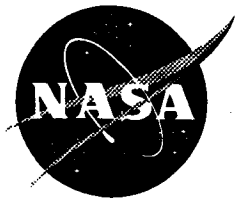
NASA/CR-2005-212873



On-Line Loss of Control Detection Using Wavelets

Peter M. Thompson, Ph.D., David H. Klyde, Edward N. Bachelder, Ph.D., and Theodore J. Rosenthal
Systems Technology, Inc.
Hawthorne, California

Under NASA Contract NAS4-01004

July 2005

The NASA STI Program Office...in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results...even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

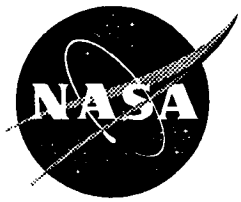
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2005-212873



On-Line Loss of Control Detection Using Wavelets

Peter M. Thompson, Ph.D., David H. Klyde, Edward N. Bachelder, Ph.D., and Theodore J. Rosenthal
Systems Technology, Inc.
Hawthorne, California

Prepared for
NASA Dryden Flight Research Center
Edwards, California
Under NASA Contract NAS4-01004

National Aeronautics and
Space Administration

Dryden Flight Research Center
Edwards, California 93523-0273

July 2005

NOTICE

Use of trade names or names of manufacturers in this document does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161
(703) 605-6000

VOLUME 1 TABLE OF CONTENTS

VOLUME 2 TABLE OF CONTENTS

VOLUME 3 TABLE OF CONTENTS

SYSTEMS TECHNOLOGY, INC

13766 S. HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250-7083 • PHONE (310) 679-2281
FAX (310) 644-3887

Technical Report No. 1341-1

ON-LINE LOSS OF CONTROL DETECTION USING WAVELETS VOLUME I: FINAL TECHNICAL REPORT

September 30, 2003

Peter M. Thompson, Ph.D.
David H. Klyde
Edward N. Bachelder, Ph.D.
Theodore J. Rosenthal

Prepared for
NASA Dryden Flight Research Center
Edwards, CA
Contract No. NAS4-01004

FOREWORD

This report documents the results of a Phase II Small Business Innovation Research (SBIR) contract to develop an on-line wavelet-based loss of control detection tool. The work was conducted by Systems Technology, Inc. (STI) in Hawthorne, CA for NASA Dryden Flight Research Center under Contract No. NAS4-01004. The period of performance for this contract was 31 January 2001 to 30 September 2003. Martin Brenner served as the Contracting Officer's Technical Representative for NASA. Peter M. Thompson was the Principal Investigator and David H. Klyde was the program manager for STI. Simulator facilities and related support were provided by BAE SYSTEMS Platform Solutions Sector in Johnson City, NY. The program manager for BAE SYSTEMS subcontract was Matthew Trouve, while the engineering effort was led by Richard Parkinson. The authors would also like to acknowledge the significant efforts of Jeffrey Gernhart, the simulator facility manager. In addition, the authors recognize the initial BAE SYSTEMS work of Andrew Corea, Timothy Horan, and Feng Liang. Finally, the authors would like to thank Douglas Thrall of BAE SYSTEMS for his support of this program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

s three volume report documents the results of a Phase II Small Business Innovation Research (SBIR) tract to develop an on-line means to predict aircraft loss of control using wavelets. Systems hnology, Inc. (STI) conducted the work for NASA Dryden Flight Research Center at Edwards, CA. overall objective of this proposed program was to apply wavelet technology in the form of a Loss of trol Analysis Tool Set or LOCATS as a means of predicting loss of control for the broad range of actory automatic and manual control system problems. These control problems are those that escape ction by typical design criteria and methodologies, surface under unusual or rare circumstances, and aten flight safety. Specific Phase II technical objectives were as follows:

- Extend the capabilities of the LOCATS to include improved wavelet formulations for real-time applications with validity measures;
- Identify time varying pilot-vehicle system characteristics that can be an indicator of impending loss of control;
- Identify, develop and validate wavelet-based time varying metrics to predict loss of control of automatic and manual flight control systems and to monitor overall flight control system health;
- Develop software and/or hardware solutions for the implementation of the real-time LOCATS;
- Provide a graphical user interface, real-time graphics displays, and networking capabilities for the prototype real-time LOCATS; and
- Evaluate the prototype real-time LOCATS in a piloted simulation that features flight control system software and hardware representative of current state-of-the-art aircraft.

evaluation of the prototype LOCATS was conducted using the flight control system hardware-in-the- p simulator facility of BAE SYSTEMS Platform Solutions Sector (here after referred to as BAE STEMS) of Johnson City, NY.

lume I of this report begins with an overview of issues involving aircraft loss of control. This is lowed by a description of the LOCATS software, detailed development of the wavelet-based loss of trol detection techniques, and a discussion of the results from the LOCATS piloted simulation luation. Supporting material is provided in the appendices. Volume II of this report provides the CATS piloted simulation test plan and detailed descriptions of the simulation checkout and formal luation. Finally, Volume III of this report contains the LOCATS User's Guide.

2. OVERVIEW OF LOSS OF CONTROL ISSUES

2.1 The Loss of Control Problem

This section provides a review the aeronautical problem that has motivated this work with wavelet analysis. In a modern aircraft flight control system (FCS) the likelihood of a catastrophic component failure (hardware and/or software) is minimized through redundancy and fault detection schemes. In fact most FCS components are designed to operate under multiple degradations and software is designed for robustness in the presence of aircraft model uncertainties. In light of these features, flight safety can still be jeopardized by loss of FCS control that can result from events originating in the instrumentation, actuation, aircraft components, or software. These events are not necessarily failures in and of themselves. Their propagation through the FCS can, however, result in an overall system failure.

Examples of such events include the X-31A crash, the JAS 39 Gripen departure and crash, Boeing 737 rudder failures, and the YF-22 PIO. On 19 January 1995, the X-31A research aircraft crashed at Edwards AFB, CA due to suspected pitot-static system icing. Although the instrument failure has been identified as the root cause of the incident, many other issues regarding safety entered into the mix as outlined in Ref. 10. In August 1993 the JAS 39 Gripen crashed during an air show when the pilot lost control of the aircraft following a steep turn. A report on the investigation of the event found that the control surfaces were driven at their rate limit and the resulting phase lag in the control loop became too large for the flight control system to maintain stability (Ref. 1). In March 1999 the National Transportation Safety Board (NTSB) determined that the Boeing 737 rudder system led to the 1991 USAir and the 1994 United Airlines crashes and one near-crash in 1996. The NTSB unanimously concluded that the onboard rudder system is not reliably redundant and that component failures and malfunctions are still possible (Ref. 2). Finally, a number of factors led to the dramatic YF-22 pilot-induced oscillation or PIO and subsequent crash. Although an Air Force report on the incident indicated that there were no failures per se, the flight control system gains set by the software when thrust vectoring is active led to a configuration that was PIO-sensitive (Ref. 3). Once the PIO was initiated the surface actuators became rate limited, introducing further control lag that worsened the oscillation.

The above examples emphasize the refractory technical problems that surround the design and development of modern flight control systems. As discussed below these technical problems include:

- Control system nonlinearities (e.g., rate limiting, command path gain shaping, etc.);
- Transitions in quadratic dipoles from favorable to unfavorable (e.g., changes in flexible mode characteristics resulting from the release of stores);
- Controller gain mismatch;
- Signal aliasing; and
- PIO tendency.

2.2 FCS Nonlinearities and Loss of Control

In conventional manual control systems the primary introduced nonlinearities are rate and position limits intrinsic to surface actuators and various preloads, thresholds, and detents intended to offset frictional and other unfavorable effects and thereby improve the threshold properties of the cockpit manipulators. Fly-by-wire/fly-by-light FCS, on the other hand, provide a variety of possibilities for deliberate nonlinearities on an easy to mechanize basis. At the outset, the more elaborate variety of system mode possibilities requires a fairly large number of nonlinear elements just to cope with FCS mode and aircraft configuration shifts, various changing interfaces, etc. The easy to mechanize features of digital control also provide a fertile field for the introduction of special situation-sensitive features intended to offset some perceived unfavorable events. There is not always, however, a comprehensive understanding and

preciation of the accompanying side effects, not the least of which can be an enhanced susceptibility to O and other loss of control scenarios.

ıe locations of common FCS nonlinearities are shown in Figure 1. These include gain shaping of the lot's input command, software and actuator rate limits, and control surface position limits. Although ɜ use of these nonlinearities is commonplace and appropriate, it is always desirable to design a system at is as linear as possible (Ref. 4) thereby avoiding rare but potentially catastrophic loss of control ents. Such events have been demonstrated in several highly visible events including the YF-22 PIO ıere inappropriate command gain shaping resulting from FCS mode switching produced severe elevator te limiting and the Gripen departure that also featured severe rate limiting (Ref. 5). Events of this ture, though not catastrophic, have also been encountered during the development of two recent FBW ınsports, the Boeing 777 and C-17.

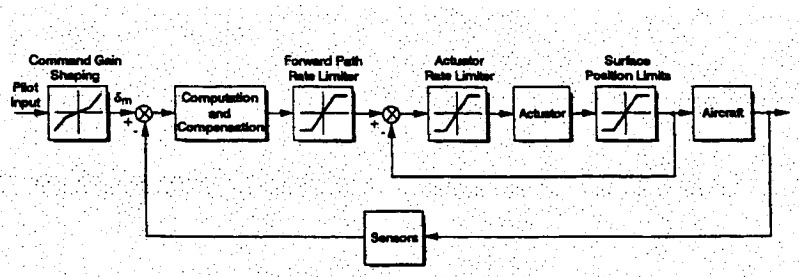


Figure 1. Location of Common Flight Control System Nonlinearities

2.1 Rate Limits

ften times software rate limits are placed in series with the actuator limit to insure that the actual ıysical limit of the actuator is not encountered, thus preventing potential damage to the flight hardware. is not unusual for these nonlinearities to be found in the feedback path as well. As described in Ref. 6 e primary difference between the software limit and the hardware limit is the significant loss in actuator ındwidth that accompanies the hardware limit when saturated. A simplified model of a rate-limited tuator is shown in Figure 2. As long as the error signal, e, remains below the saturation point, e_L, the stem behaves as a linear first order lag whose response is entirely dictated by the time constant, $T = \omega_a$, where ω_a is the actuator bandwidth. When saturation occurs the surface will move at its maximum te, V_L, until the commanded magnitude, frequency, or both are reduced. Saturation occurs when the ror signal exceeds the saturation point (i.e., $e > e_L = V_L / \omega_a$). It is also convenient to define a nonlinear stem time constant, $T_{NL} = A/V_L$, where A is the amplitude of the input command. Although this time ınstant applies only to the indicial (step) response describing function, it has been shown to have an ıportant role in characterizing the nonlinear system for other input forms (Ref. 6).

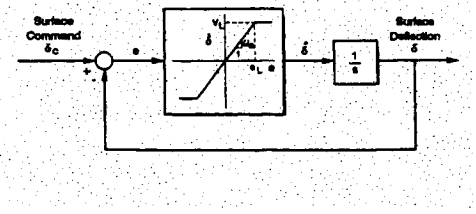


Figure 2. Simplified Actuator Model

ıgure 3 displays the closed loop actuator time responses for a saturation case with an input amplitude of 5°. The error signal still appears more or less sinusoidal even though the 2° saturation point is generally

exceeded throughout the run length. The nonlinear nature is more evident in the actuator rate response that appears box car-like for this highly saturated case. In the lower panel, the actuator output position is compared to the input command. A triangle wave output response is displayed that reverses when approximately equal to the input (i.e., when the error signal passes through zero). For the linear case, the delay between the output and input is the linear time constant or 0.05 sec. In the nonlinear case shown below, this delay has increased to 0.15 sec.

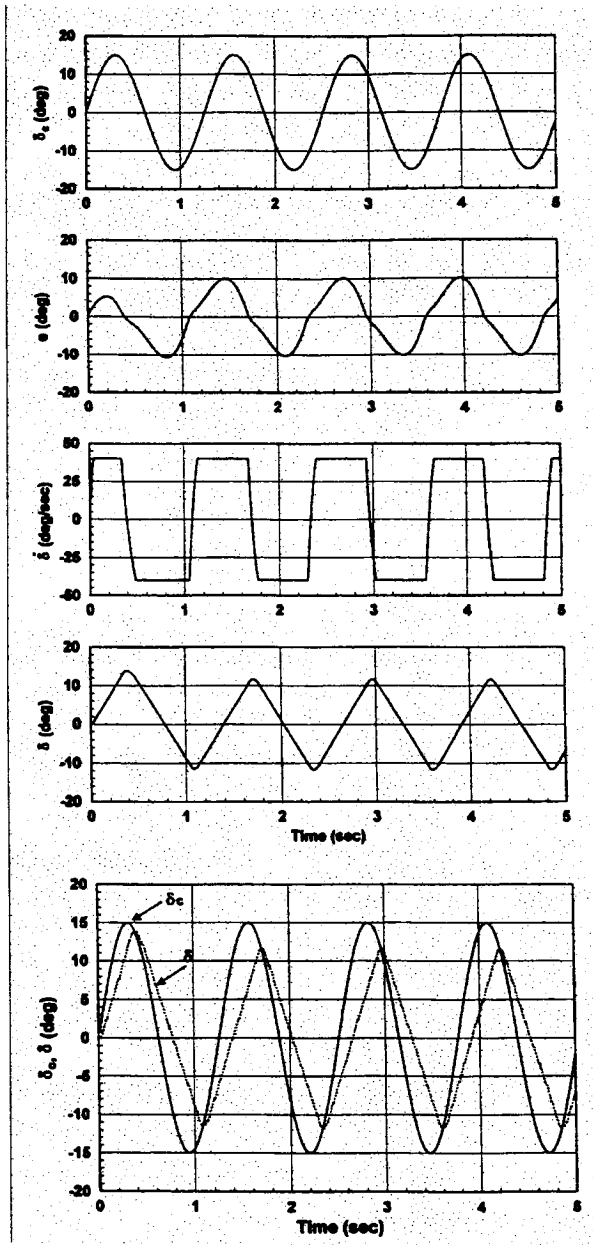


Figure 3. Rate-limited Actuator Responses to a Sinusoidal Input

In general, rate saturation results in an amplitude reduction and a significant added phase lag. These characteristics are displayed in the magnitude and phase curve families of Figure 4. Both sets of curves are plotted as a function of normalized frequency, ω/ω_a, and the linear to nonlinear time constant ratio, T/T_{NL}. In the sinusoidal input case T/T_{NL} is equal to $V_L/(\omega_a A)$. Thus, the two plots display the describing

ction magnitude and phase of the nonlinear system in terms of the actuator design parameters (V_L and and the input parameters (A and ω), all known quantities. There are several observations to note from plots. First, the $T/T_{NL} = 1$ curve represents the linear case. Second, the more highly saturated cases resented by the smaller time constant ratio curves, depart from the linear curve at a normalized quency that is equivalent to their time constant ratio. For example the $T/T_{NL} = 0.1$ curve departs from linear curve at a normalized frequency of 0.1. Another more significant result is that known design input parameters can be used to identify the added phase lag due to a rate limiting actuator.

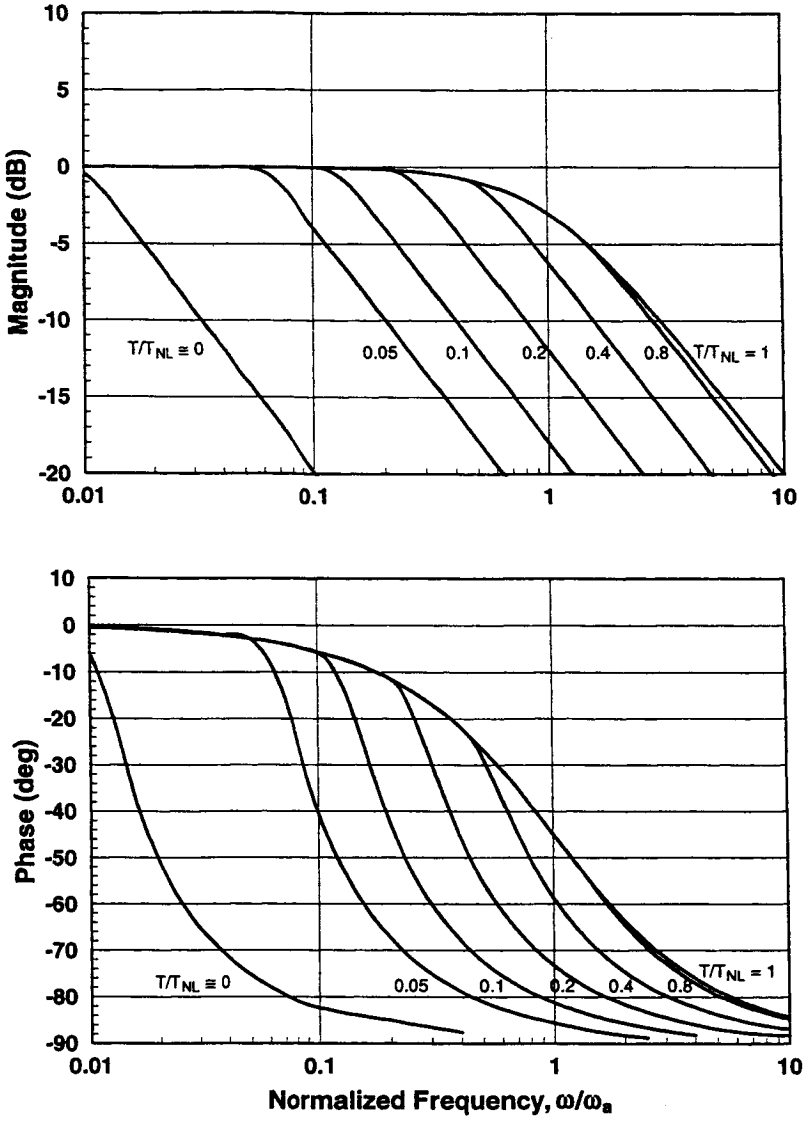


Figure 4. Frequency Response of Nonlinear Actuator Model

software rate limit can be represented by the actuator model described above where the bandwidth is ectively infinite (i.e., $T/T_{NL} \rightarrow 0$) as discussed in Ref. 7. Although Figure 4 displays both an amplitude luction and added phase lag due to rate limiting, it is the added phase lag that is the dominant effect. us when encountered in a FCS or a PVS, the primary effect of actuator or software rate saturation is to ısume available phase margin that can then lead to loss of control.

2.2.2 Nonlinear Gain Shaping

The most common deliberately introduced nonlinear feature of a FCS is a variable gain in the pilot command pathway. An example gain shaping block is shown in Figure 5. The slope around neutral, K_1, is optimized for high pilot gain (low amplitude) precision control. K_2 and often other break-up slopes are optimized to achieve full output with maximum input. The describing function for the command shaping element is N(a/A), where A is the input amplitude.

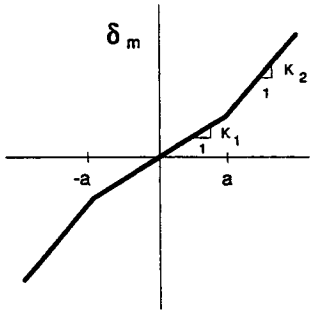


Figure 5. Example Command Gain Shaping Nonlinear Element

From Figure 5 the linear range is defined for an input amplitude $|A| \le |a|$ and the peak output amplitude is proportional to K_1 (i.e., $\delta_m = K_1 A$). For a sinusoidal input with amplitude A, the periodic output when $|A| > |a|$, will have a peak amplitude given by, $\delta_m = K_1 a + K_2(A - a)$. Using the peak value for the amplitude as an approximation of the fundamental component of the periodic output gives the describing function estimate shown below:

$$|N| \approx \frac{\delta_m(a/A)}{A} = K_2 - (K_2 - K_1)\frac{a}{A}$$

Gain shaping nonlinearities can lead to a reduction in system stability when unexpected jumps in gain occur (e.g., YF-22 PIO). The reduction in stability with increased gain may result directly from reduced gain margins or the introduction of other nonlinearities (e.g., rate limiting).

2.2.3 Mechanical FCS Nonlinearities

As described in Ref. 8 mechanical FCS nonlinearities that can influence system performance include friction and hysteresis, inceptor preload, and inceptor to surface gearing. Because friction is unavoidable, it must be minimized in the design process. Coulomb friction in a mechanical inceptor (stick, pedal, etc.) always opposes the motion of the inceptor. This will produce a displacement versus force hysteresis loop. In high gain scenarios limit cycle oscillations in the FCS and/or AFS are possible in the presence of hysteresis. Preload is used to return the inceptor to a centered position and also to reduce the dynamic consequences of hysteresis through describing function attenuation and phase reduction (Ref. 8). Inceptor to surface gearing is usually designed to provide a nearly constant gradient for trim conditions throughout the flight envelope and increased gradients near full surface deflection to prevent excessive stick deflection or force requirements while maneuvering.

2.3 Additional FCS Loss of Control Mechanisms

a. Unfavorable Quadratic Dipoles

The presence of lightly damped quadratic dipoles (i.e., a quadratic pole-zero pair) that result from various feedback control scenarios can be lead to system instability. These dipoles are especially troublesome when they occur in the frequency region of piloted control. Although there are several notorious rigid body scenarios (Ref. 8), these quadratic dipoles are also associated with flexible modes. In general, an

unfavorable and potentially unstable (closed-loop) condition arises when the numerator zero is at a higher frequency than the pole, while a favorable and stable condition arises when the pole frequency is higher. These pole-zero location effects are exemplified in the system survey plots of Figure 6. For AFS systems favorable dipole pairs (scenario "b" in Figure 6) may also become unfavorable (scenario "a" in Figure 6) as store configurations change.

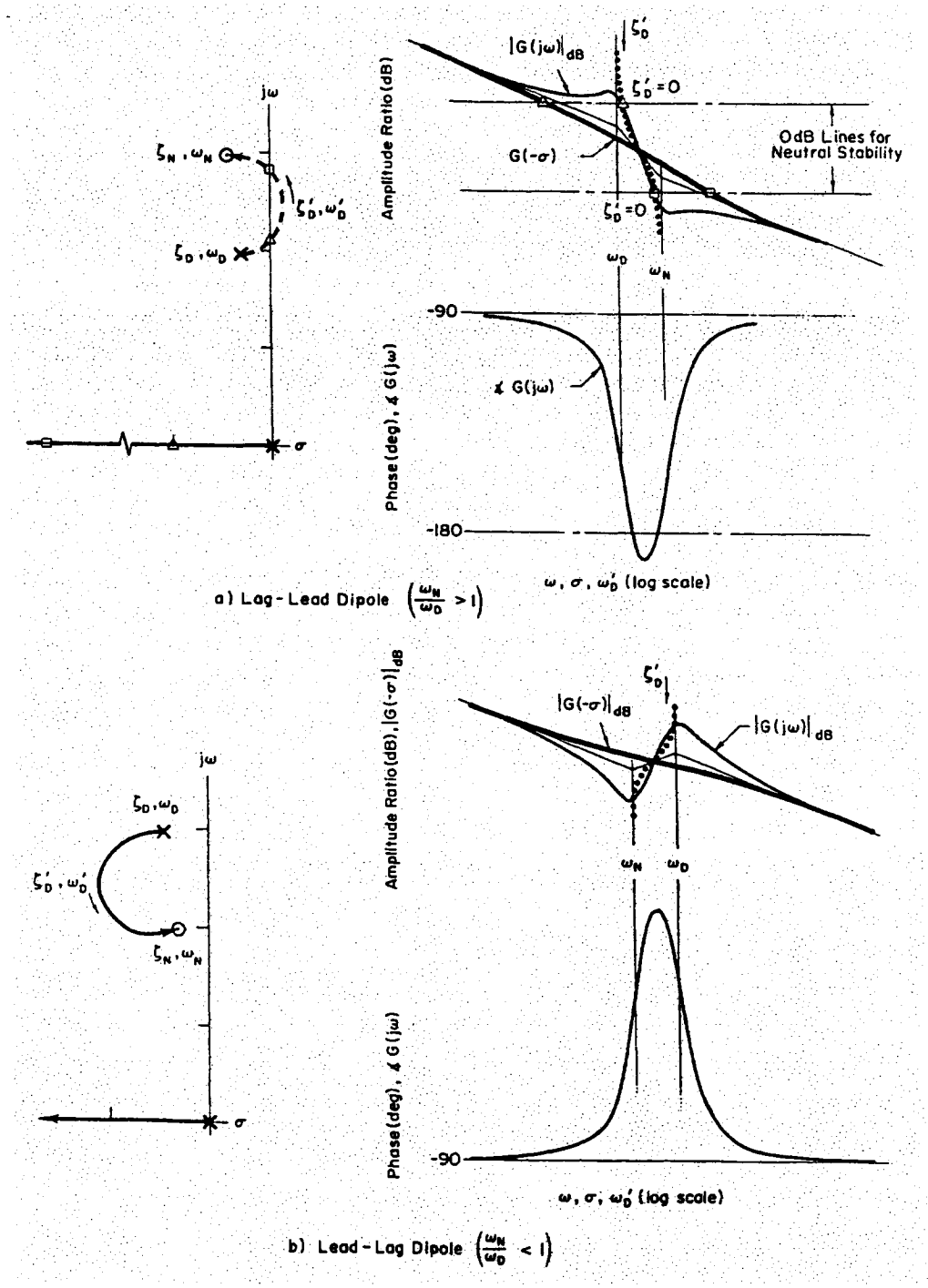


Figure 6. Quadratic Dipole System Surveys

2.3.1 Gain Mismatch

The determination of the optimum FCS controlled element gain is clearly a matter of supreme importance to assure a favorable pilot-aircraft interface, effective pilot-vehicle interactions, and an absence of PIO tendencies. With conventional center sticks, pedals, and yokes decades of past practice provide traditional answers. With the introduction of full-authority stability augmentation and fly-by-wire/fly-by-light systems the stick "inceptor" has permitted the introduction of many other options. These range from side versus center cockpit locations, force-alone versus various degrees of motion, etc. Also, the inceptors have become "subsystems" by incorporating sensors and frequency and nonlinear amplitude shaping circuits. Further, the harmonization of within- and between-axis characteristics of cockpit inceptors that share functions, such as the conventional stick as a lateral and longitudinal controller, has new dimensions. Consequently, the proper setting of controlled element gain has become a nontrivial development aspect on every new aircraft that introduces a new inceptor at the pilot-control-system interface. In the absence of an extensive background of data for these there is no basis other than experiment to determine the optimum gains. The pathway to ultimate success has often had many byways, with minor wiggles, bobbles, and ratchets, as well as occasional severe PIOs.

2.3.2 Sampling Rates and Aliasing of Signals

Sampling at different rates in a FCS can be problematic. An example problem occurred as part of the Outrider Autonomous Uninhabited Vehicle (AUV) analysis conducted at STI as part of a recent consulting effort (Ref. 9). It was desired to calculate the transfer function from input and output signals that were stored on separate recorders at different sampling rates. When the transfer function was computed an additional phase lag was present that was not easily explainable. Further investigation revealed that the different sampling rates produced the additional lag that when properly accounted for was no longer present.

Another possible source of trouble in the control system is aliasing of signals. An example of this was encountered during the development of an Active Oscillation Control System (AOCS) for the F/A-18A block aircraft that suffered from limit cycle oscillations between 5 and 6 Hz under certain store configurations (Ref. 10). Flight testing revealed that high frequency content between 35 and 40 Hz at the output of the lateral accelerometer was aliased (for a sampling rate of 40 Hz.) into the problem frequency range of 5 to 6 Hz.

2.3.3 System Failures

Other mechanisms that can lead to FCS loss of control include sensor loss, actuator failures (hydraulic or electric), software failures, and battle damage. Many of these mechanisms are routinely handled through required system redundancy.

2.4 Pilot-Vehicle Systems and Loss of Control

The range of human pilot control possibilities as part of a pilot-vehicle system (PVS) comprises three fundamental modes of dynamic behavior (Ref. 11). These correspond to the three-pathway structure shown in Figure 7. The pathways are internal organizations of the pilot's perceptions. They underlie and correspond to the following pilot control behavior patterns:

- Closed-loop (compensatory) – pilot controls outputs driven by PVS errors;
- Combined Open- and Closed-loop (pursuit) – pilot control is dependent on PVS errors plus known or induced input commands and/or system outputs.
- Open-loop (precognitive) – highly-skilled and practiced pilot programmed control outputs to command the aircraft to a new state.

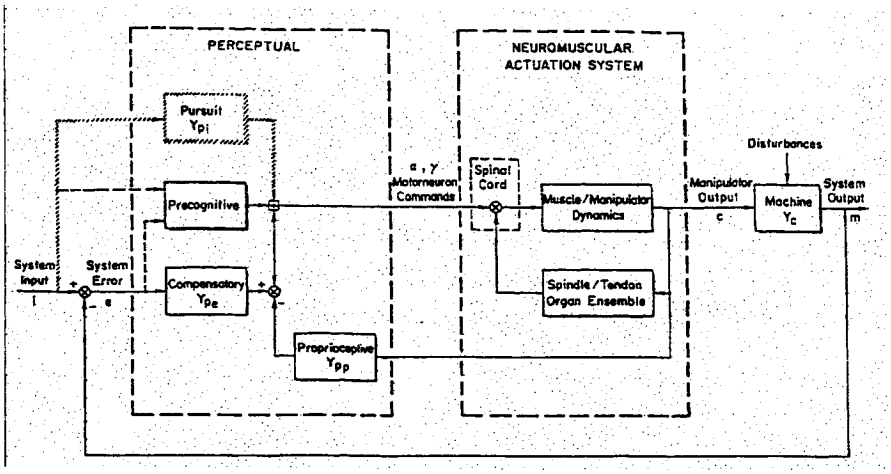


Figure 7. Three Modes of Perceptually Centered Model of Pilot Behavior

In both compensatory and pursuit operations the pilot is highly interactive with the aircraft on a continuous basis, whereas in precognitive operations the pilot's actions are essentially discrete and independent of the aircraft's response during the time period of the pilot's programmed command. Precognitive behaviors constitute the vast majority of pilot activities in nominal aircraft maneuvers where the pilot's outputs are a stored repertory that are released in a timely and appropriate fashion. These open-loop sequences may be short, as with a pseudo bang-bang control, or long, as in generating a synchronous sine wave. The aircraft simply responds without further pilot intervention. Of course, if the ultimate response is not as desired, the pilot may insert another control sequence. In compensatory and pursuit processes the pilot is part of a feedback control process, with all of the advantages and disadvantages (e.g., possible instabilities leading to loss of control) associated with this kind of control.

b. Compensatory Control Behavior

Here, the crossover model will be applied as a "truth" model for assessment of pilots performing precision tracking tasks. There is a wealth of research concerning the development of pilot behavior models including the crossover model. As detailed in Ref. 11, the crossover model is valid for single-loop compensatory control (e.g., precision tracking). A block diagram for the compensatory control scenario is shown in Figure 8. As shown in the figure, the pilot controls the system output, *m*, in response to the PVS error, *e*.

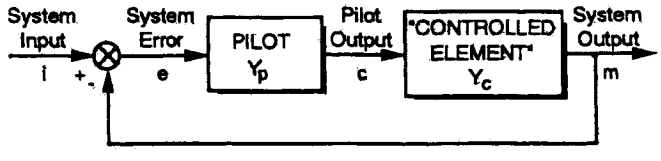


Figure 8. Compensatory Control Scenario

In short, the crossover model states that the pilot-vehicle system can be represented by the following equation:

$$Y_p(j\omega)Y_c(j\omega) = \frac{\omega_c e^{-j\omega\tau_e}}{j\omega} = \frac{m}{e}$$

In the above equation and figure Y_p is the pilot describing function, Y_c is the controlled element, ω_c is the crossover frequency, and τ_e is the effective system time delay. The key variables, ω_c and τ_e, are functions of the controlled element dynamics (airplane model), task variables (desired and adequate performance requirements), and environment (system delays, field-of-view, etc.). The crossover frequency is defined as the frequency on a Bode plot at which the pilot-vehicle system open-loop describing function amplitude ratio crosses the 0 dB line. It has been demonstrated through extensive research that those controlled elements that are most "k/s–like" in the region of crossover require the least compensation by the pilot. In turn, as pilot compensation increases, the achieved crossover frequency decreases. The effective system time delay is a function of the high frequency flight control system and aircraft dynamics (e.g., actuators, structural filters, structural modes, etc.) and pilot added time delays due to lead generation. Once again, the more "k/s–like" the controlled element is in the region of crossover, the less pilot lead compensation will be required and the smaller the effective time delay. When little or no lead generation is required by the pilot and the higher frequency dynamics are negligible, the effective time delay will consist solely of the delay in the human pilot's response. This delay has been shown to be in the neighborhood of 0.2 to 0.25 sec.

The closed-loop characteristics of the crossover model as defined in Ref. 11, allow the system phase margin to be defined in terms of the key model parameters as shown in the following equation,

$$\phi_M = \frac{\pi}{2} - \tau_e \omega_c$$

For evaluation tasks defined by known forcing functions and measured error signals (such as a sum-of-sines tracking task), key parameters such as crossover frequency are measured directly. These proven handling qualities experimental techniques will be used to evaluate and refine the metrics defined as part of the wavelet analysis techniques.

2.4.1 Pursuit Control Behavior

A block diagram of a pursuit control scenario is shown in Figure 9. In this instance the pilot is able to perceive system inputs and can then operate on both the error (Y_{p_e}) and the input forcing function (Y_{p_i}). The actual input may be shown on a display, or may be deduced by the pilot by virtue of the available cues. To clarify this point, consider the probe-and-drogue refueling task. When beginning the approach to the drogue, the pilot perceives the motions of the drogue (the input) against a background of the tanker, horizon, and any context such as clouds. As the aircraft nears the drogue, the pilot's attentional field begins to narrow or tunnel as the permissible error is reduced. In the process the pilot becomes focussed on the "error" (probe-and-drogue displacement) rather than the "input" and aircraft "output" as well as this error. The pilot's control output becomes conditioned on the error alone, and behavior changes from "pursuit" to "compensatory." In pursuit behavior the controlled element block shown in the figure remains unchanged from the previous scenario.

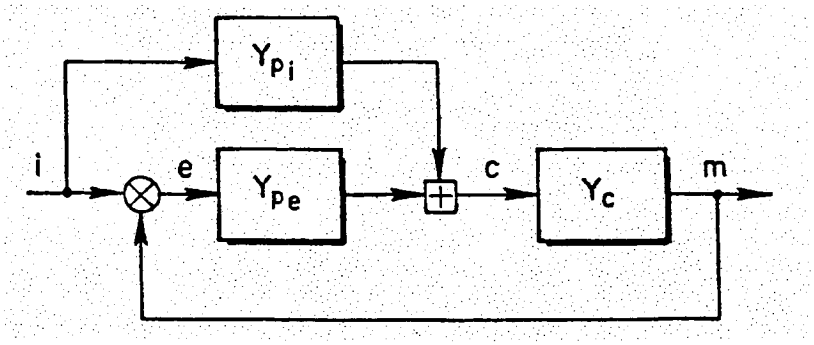


Figure 9. Pursuit Control Scenario

The output over error describing function is defined by,

$$Y_\beta = \frac{Y_c\left(Y_{p_i} + Y_{p_e}\right)}{1 - Y_c Y_{p_i}} = \frac{m}{e}$$

When the pilot responds to error alone, the above equation reduces to the compensatory case. The closed-loop and error over input describing functions are,

$$\frac{m}{i} = \frac{Y_\beta}{1+Y_\beta} = \frac{Y_c\left(Y_{p_i} + Y_{p_e}\right)}{1 + Y_c Y_{p_e}} \quad \text{and} \quad \frac{e}{i} = \frac{1}{1+Y_\beta} = \frac{1 - Y_c Y_{p_i}}{1 + Y_c Y_{p_e}}$$

As the denominator for this scenario is $1 + Y_c Y_{p_e}$, the PVS stability remains a function of only the compensatory system. A pilot performing pursuit control behavior will attempt to make Y_β very large so that the $|m|/|i| \rightarrow 1$ and $|e|/|i| \rightarrow 0$. In the limit this can approach ideal "precognitive" control where Y_{p_i} can exactly invert Y_c. Then $Y_{p_i} = 1/Y_c$ and m/i =1 and there is no closed-loop control. In any event the feed forward element does, however, improve overall system performance. In general, the PVS closed-loop bandwidth and apparent crossover frequency (when estimated by measurements of m/e) will be higher.

2.4.2 Pilot-Induced Oscillations

Pilot-induced oscillations or PIOs are perhaps the most common type of pilot-vehicle system loss of control. PIOs have been an aviation safety problem since the Wright Brothers first powered flight 100 years ago. As the state-of-the-art in aircraft technology advanced, the root causes of these unfavorable couplings has changed, yet the problem remains – even with the most advanced FBW designs. In 1997 the National Research Council published a report entitled *Aviation Safety and Pilot Control* (Ref. 5) that provided a series of recommendations to minimize the occurrence of these rare, but extremely hazardous events. In the time since this report was published, a number of severe events have occurred in commercial aviation, some of which resulted in serious injuries and fatalities. Specific examples are given in Table 1. Figure 10 shows flight data recorder time histories from the Falcon 900 PIO. The severity of the event is illustrated by the oscillations in vertical acceleration of greater than 4g and –3g. Furthermore, there have been a number of military examples of unfavorable pilot-vehicle interactions since the publishing of the NRC report. These include the shipboard landing PIO examples listed in Table 1 that could have easily resulted in lost or damaged aircraft had the tasks not been successfully aborted. These events clearly demonstrate that unfavorable pilot-vehicle interactions remain a threat to aviation safety.

Although PIOs have similar defining traits, the causal factors can be quite varied. Thus, we have come to categorize PIOs in terms of existing pilot behavior models and analysis techniques. These categories are summarized below from Ref. 5.

Category I – Essentially Linear Pilot-Vehicle System Oscillations: The effective controlled element characteristics are essentially linear, and the pilot behavior is also quasi-linear and time-stationary. The oscillations are associated with high open-loop system gain. The pilot dynamic behavior mode may be pursuit, compensatory, or synchronous.

Category II – Quasi-Linear Pilot-Vehicle System Oscillations with Series Rate or Position Limiting: These are severe PIOs, with oscillation amplitudes well into the range where software and/or actuator rate and/or position limiting in series with the pilot are present as primary nonlinearities. Rate limiting, either as a series element or as a rate-limited surface actuator, modifies the Category I situation by adding an amplitude-dependent lag and by setting the limit cycle magnitude. Other simple nonlinearities (e.g., stick command shaping and some aerodynamic characteristics) may also be present. These are the most common and easiest to analyze true limit-cycle severe PIOs.

Table 1. Recent Commercial and Military PIO Events

Date	Aircraft	Flight Details	Event Description
8 June 97	MD-11	JAL Flight 706 1 fatality	PIO Pilot attempted to manually override autopilot & deploy speed brakes
1 Jan. 99	B-767	AA Flight to UK-Heathrow	PIO on first touchdown bounce resulting in compression buckling of fuselage skin
14 Sep. 99	Falcon 900	Olympic Airways Flight to Bucharest Romania 7 fatalities	PIO followed autopilot disengage. Aircraft was dispatched with failed pitch force feel system.
8 Feb. 01	A-321	Air France to UK-Heathrow	Roll PIO followed sudden left wing drop
17 Mar. 01	A-320	NW Flight 985 from Detroit	Elevator not trimmed for takeoff. Pitch PIO resulted in aborted takeoff.
1 Feb. 99	V-22	Sea Trials	Roll PIO followed saturation of AFCS
8 Mar. 99	F/A-18 F	Follow-on Sea Trials	Roll PIO followed rate saturation of lateral actuators resulting in bolter

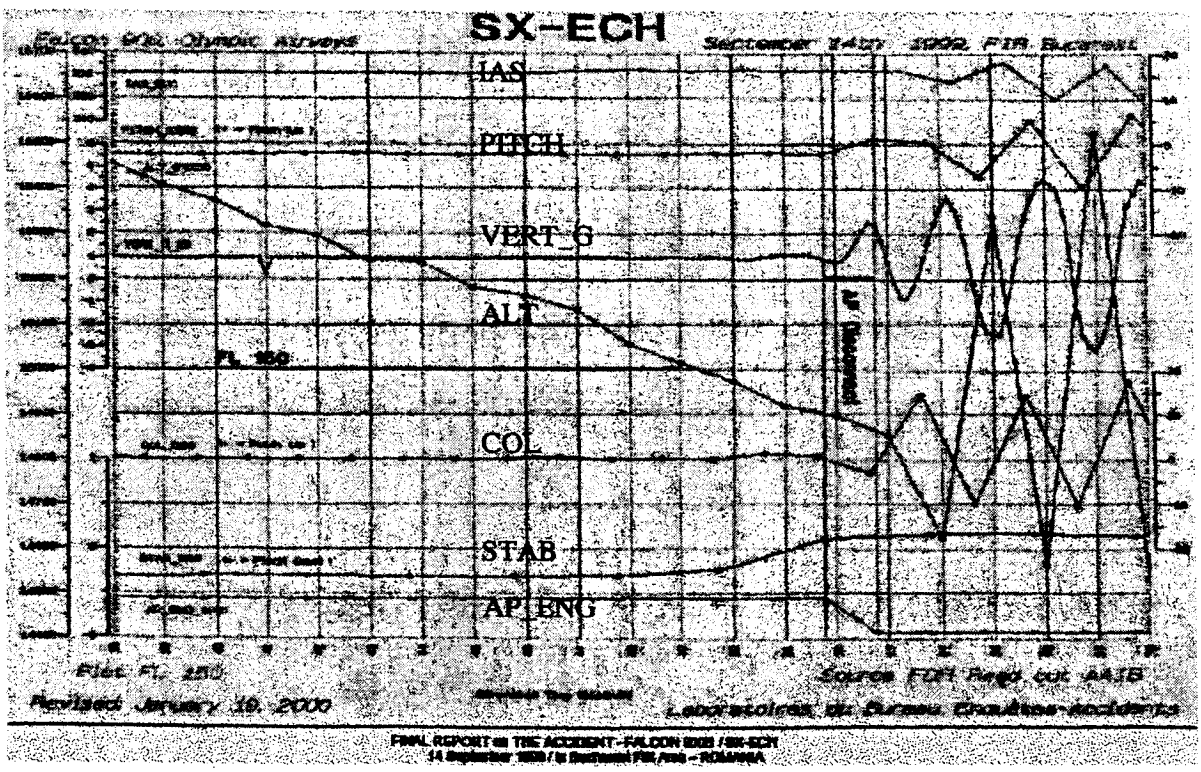


Figure 10. Flight Data Recorder Time Histories from Olympic Airways PIO (Ref. 12)

Category III – Essentially Non-Linear Pilot-Vehicle System Oscillations with Transitions: These PIOs fundamentally depend on nonlinear transitions in either the effective controlled element dynamics, or in the pilot's behavioral dynamics. The shifts in controlled element dynamics may be associated with the size of the pilot's output, or may be due to internal changes in either control system or aerodynamic/propulsion configurations, mode changes, etc. Pilot transitions may be shifts in dynamic behavioral properties (e.g., from compensatory to synchronous), from modifications in cues (e.g., from attitude to load factor), or from behavioral adjustments to accommodate task modifications.

With existing criteria the likelihood of Category I PIOs is now easily minimized in the FCS design process. Because of the presence of significant nonlinearities in the controlled element, the human operator or both, the prediction of susceptibility to Category II and III PIOs is much a more troublesome problem.

3. DESCRIPTION OF THE LOSS OF CONTROL ANALYSIS TOOL SET (LOCATS)

3.1 Overview

The LOCATS application is a real-time analysis tool that is designed to detect impending loss of control. The loss of control may originate within an automatic flight control system mode or a pilot-vehicle system. The most common form of loss of control within a pilot-vehicle system is the pilot-induced oscillation or PIO. In Phase I, the feasibility of the LOCATS concept was demonstrated using a wavelet toolbox developed by STI using Matlab (Refs. 13 and 14). In Phase II this toolbox was used to further develop and evaluate the wavelet-based estimation techniques and to assess time-varying loss of control metrics.

A program flow diagram for the LOCATS software is shown in Figure 11. All program options and data manipulation features are accessed via the graphical user interface or GUI. Program options include selection of desired wavelet-based analysis tools, loss of control metrics, and data displays. The input to the software is a set or subset of aircraft signal variables that are available from the simulator via the SBS Technologies 1553 data bus card installed in the host PC. In this case the aircraft signals are selected from those available within the structure of the BAE SYSTEMS simulator facility. From these signals a user-selected subset is sent to the wavelet analysis module from which time varying frequency responses and related computations are made. These data are then used to exercise the user-selected loss of control metrics. Finally, the results are presented graphically via the data display module. This process takes place in real-time, so that impending loss of control may be detected early enough for the pilot to receive appropriate warnings and take corrective actions.

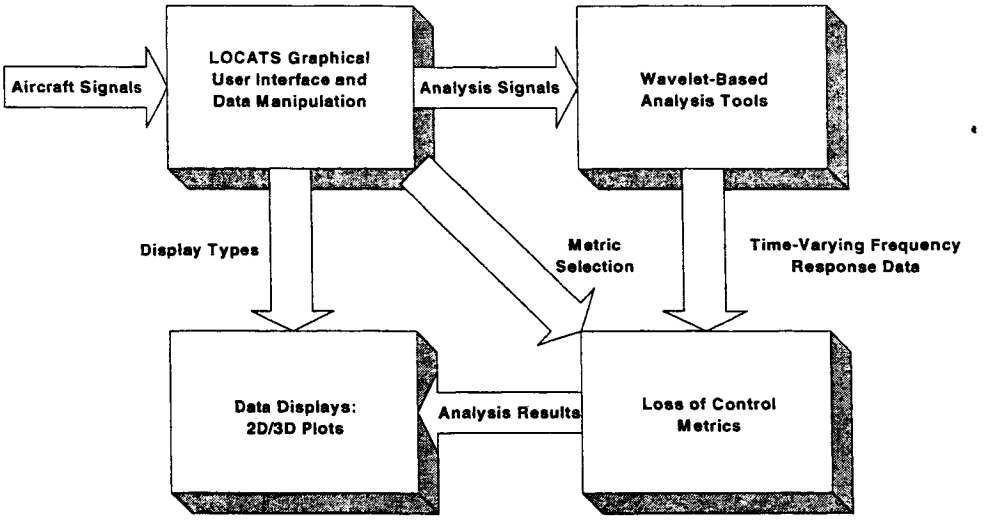


Figure 11. LOCATS Program Flow Diagram

A LOCATS User's Guide is included as Volume III of this report. The reader is directed to that volume for specific details concerning the use of the software. The following sections will provide a brief summary of the software components identified in Figure 11.

3.2 The Graphical User Interface

As mentioned above access to LOCATS options and features is provided via the GUI and details surrounding its use are featured in the Volume III User's Guide. The LOCATS main window is shown in Figure 12. Data handling features are exercised using the toolbar icons at the top of the window. The

icons from left to right, Save Collected Data, Set Configuration Options, Load Data from a File, Add New Plots, Begin Data Collection, Pause Playback, Stop Data Collection, and Plot Existing Data. A running log of LOCATS execution is provided under System Information. The wavelet-based analysis tool of choice is selected via the Analysis Method box. Because the WERA method was a more recent development, it has not yet been fully integrated into the software, and hence only one analysis method can be used at a time. The final feature of the LOCATS main window is the Active Plots listing. From this listing the various run-time plots can be configured for a given data collection scenario or off-line analyses.

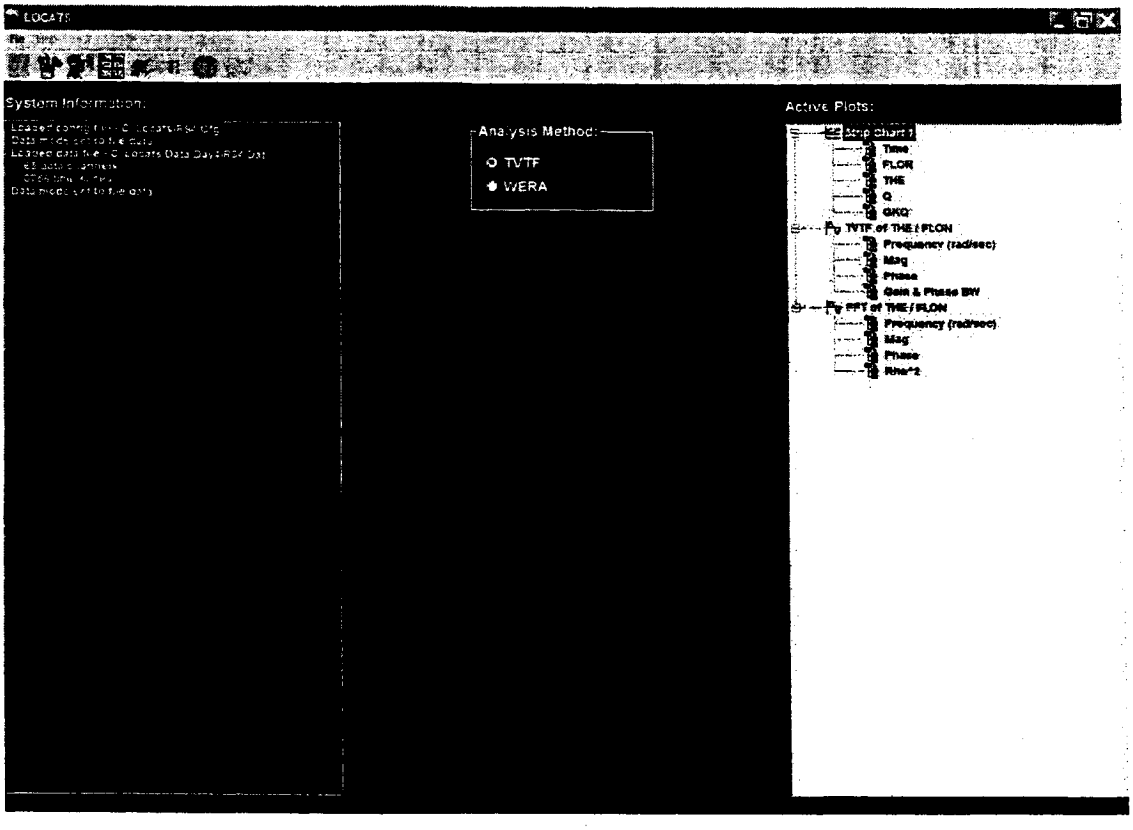


Figure 12. The LOCATS Main Window

3.3 The Wavelet-Based Analysis Tools

Two wavelet-based analysis methods have been developed for use in LOCATS. The first is the time varying transfer function (TVTF) method that was first introduced in the Phase I program (Ref. 13). The second is a wavelet eigensystem realization algorithm (WERA) method. Detailed descriptions of these methods are provided in Sections 4 and 5 of this report. Because both methods have unique strengths and weaknesses, they were both included in LOCATS for the piloted simulation assessment. Both methods provide results using a common set of loss of control metrics.

Analysis options are selected prior to a data collection run using the Analysis Options window (see Figure 13). Timing Options set the length of the transform and the interval between transforms, while fmax sets the maximum frequency. FFT Options set the binning or frequency averaging, the length of the transform, and the cosine taper fraction for the fast Fourier transform computations. CWT Options select the continuous wavelet transform and its properties, the length of the transform, and the number of frequencies for the TVTF computations. LLS Options sets the numerator and denominator order, the number of points used in the estimate, the minimum frequency, and the number of frequencies for linear

least squares time series estimations. When incorporated, the WERA Options will set the parameters for the wavelet eigensystem realization algorithm estimates.

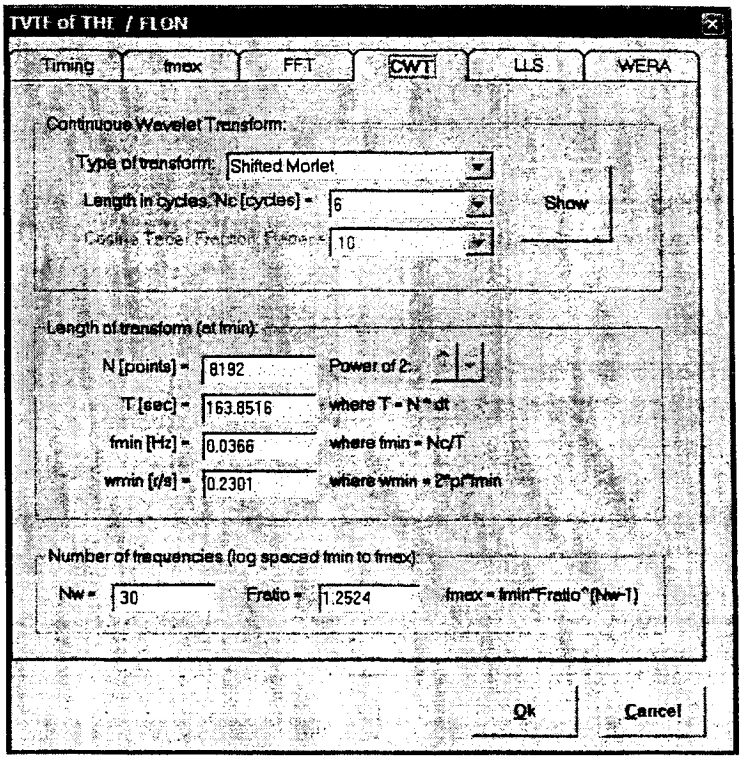


Figure 13. LOCATS Analysis Options Window

3.4 Loss of Control Metrics

3.4.1 Overview

The primary goal of this program is to develop on-line tools that can identify impending loss of control. The wavelet-based methods provide the engine for identifying time-varying systems. Once the system has been identified a wealth of metrics can employed to assess the susceptibility of the system to loss of control. These "time-varying" metrics include classical system stability measures (e.g., gain and phase margins, peak magnification ratio, etc.), multivariable system robustness measures, and the many flying qualities measures. The focus here has been on those system changes that result in a loss of phase margin and therefore greatly increase susceptibility to pilot-vehicle system instabilities such as PIO. Thus, loss of control metrics that are sensitive to changes in phase margin were emphasized.

3.4.2 Airplane Bandwidth

The airplane bandwidth handling qualities criteria has been shown to be an effective indicator of degraded handling qualities and PIO susceptibility (Ref. 15). Thus, a time-varying version of this metric is of particular interest to this project. The airplane bandwidth criterion was developed from the crossover model concept (Ref. 16). As described in Ref. 17, the attitude bandwidth is related to the premise that the maximum crossover frequency that a pure gain pilot can achieve, without threatening stability, is a valid figure-of-merit for the controlled element. Key bandwidth parameters include the bandwidth frequency (ω_{BW}), the neutral stability frequency (ω_{180}), and phase delay, a measure of the high frequency phase roll off (τ_p). As shown in Figure 14 the bandwidth parameters are obtained from an attitude to manipulator force response, because the primary control cue for the pilot is attitude and not, for example, acceleration. In this way the higher frequency effects of actuator and inceptor dynamics are included. Examples of

current PIO and handling qualities boundaries for pitch attitude bandwidth are shown in Figure 15. Similar requirements have been defined for the roll axis.

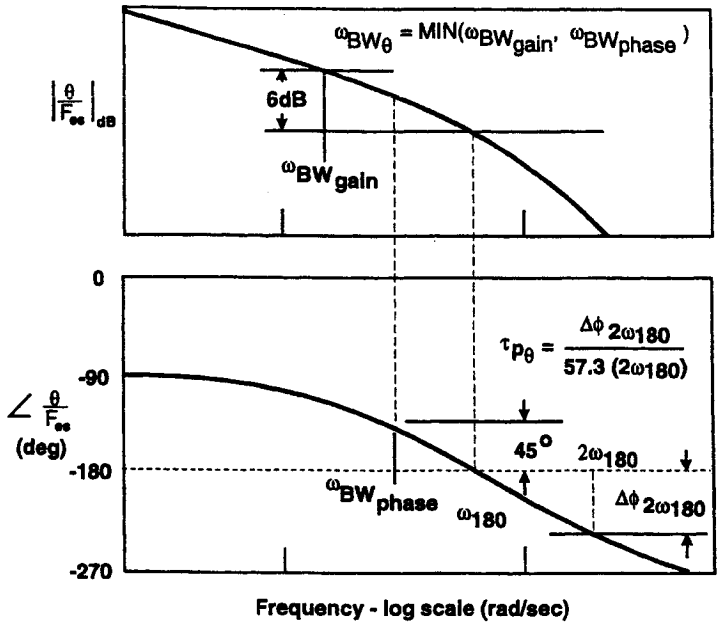


Figure 14. Definition of Pitch Attitude Bandwidth Parameters (modified from Ref. 3)

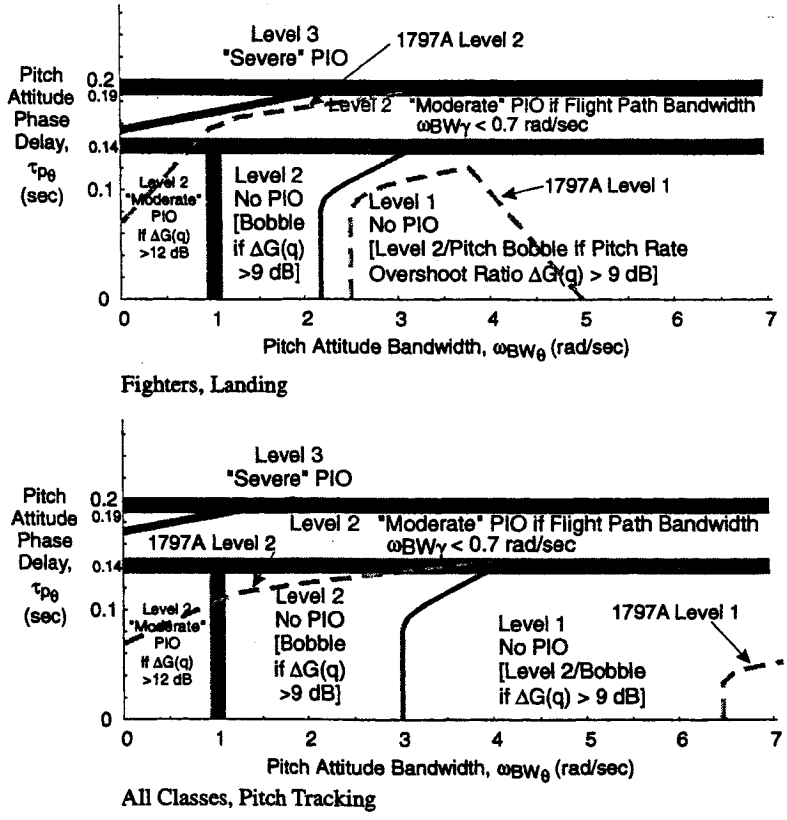


Figure 15. Example Pitch Attitude Airplane Bandwidth PIO (Thick Lines) and Handling Qualities (Thin Lines) Boundaries (Ref. 3)

An example of how the airplane bandwidth parameters may change over time is shown in Figure 3 (Ref. 18). In this example an overall Level 1 configuration is combined with increasing levels of rate saturation. Here the data are plotted against a simplified version of the Figure 15 PIO boundaries. As the saturation level increases, the phase delay increases and the bandwidth decreases, eventually resulting in a configuration that is highly susceptible to PIO.

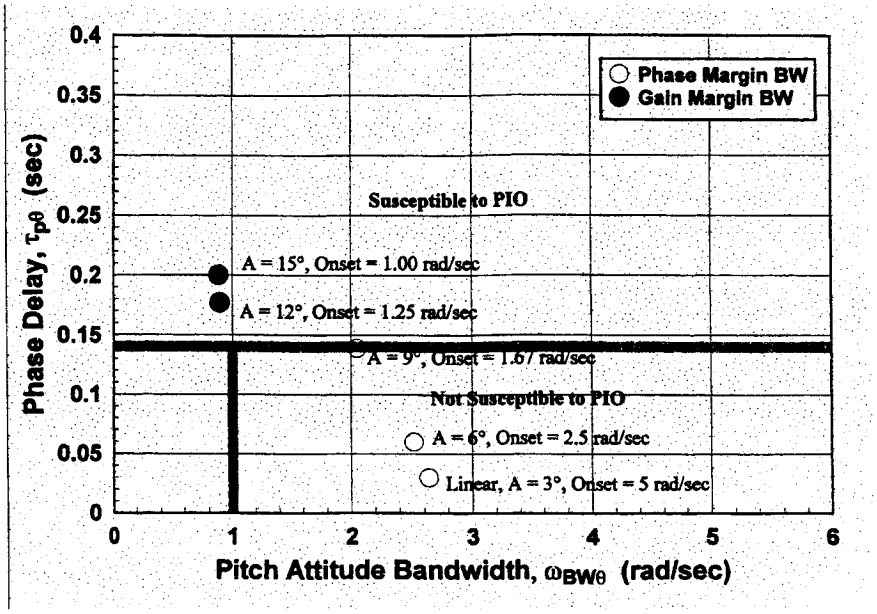


Figure 16. Example Airplane Bandwidth Variations with Increasing Rate Saturation

3.4.3 Other Metrics

Additional metrics along with airplane bandwidth have been incorporated into LOCATS. These include the stability metrics incorporated in the STI Wavelet Analysis Toolbox during the Phase I program. For example, the frequency response estimate of a loop transfer function (LTF) can be used to compute stability metrics. A block diagram is shown in Figure 17. The signals $e(t)$ and $y(t)$ can be used to estimate the LTF.

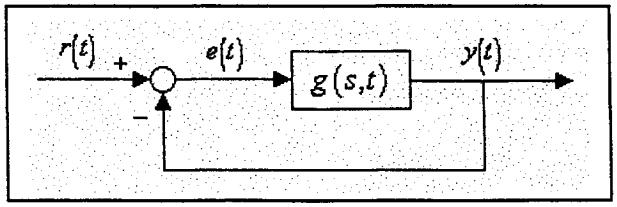


Figure 17. Loop Transfer Function Used to Estimate Time Varying Stability Metrics

Alternatively, if smoothing is available, then it can be better to also use the command signal $r(t)$ and compute:

$$\hat{g}(\omega) = \frac{\tilde{S}_{yr}(\omega)}{\tilde{S}_{er}(\omega)}$$

If the frequency responses are time varying then so are the stability metrics. In Phase I the classical stability metrics, as listed below, were implemented in the toolbox and may be easily incorporated into LOCATS:

- Phase margin (PM): For each t the frequency response $\hat{g}(\omega,t)$ is searched for a unit magnitude crossover frequency (ω_c) and the phase margin is 180° plus the phase at that frequency. Log interpolation is used between frequency data points. If there is more than one ω_c then the lowest is used. Alternatively all can be kept or only the lowest phase margin can be kept.

- Gain margin (GM): For each t the frequency response $\hat{g}(\omega,t)$ is searched for $-180°$ crossover frequency (ω_{180}) and the gain margin is the inverse gain at that frequency. Log interpolation is used between frequency data points. If there is more than one ω_{180} then the highest is used. Alternatively all can be kept or only the lowest gain margin can be kept.

- Peak magnification ratio (M_p): For each t the frequency response $\hat{g}/(1+\hat{g})(\omega,t)$ is computed from $\hat{g}(\omega,t)$ and then searched for local maxima. If there is more than one then the one occurring at the highest frequency is used.

- Multiplicative robustness: For each t the frequency response $1+\hat{g}^{-1}(\omega,t)$ is the multiplicative robustness bound, defined as the largest frequency response e_m for which $\hat{g}(1+e_m)$ can be guarantied to be stable. On three-dimensional plots peaks show up better than valleys, so the inverse bound $\hat{g}/(1+\hat{g})(\omega,t)$ is used instead (for which the maximum across ω is M_p).

These stability metrics are actually robustness measures, the distinction being not just a yes/no answer to stability but a measure of the nearness to instability. The phase and gain margins will each "go negative" when a transition to instability occurs. For automatically controlled systems this transition to instability is readily apparent, and in fact a warning of impending instability can be implemented, even with an estimate of time remaining. For manually controlled systems the transition to instability is not so clear cut. A human operator can (and does) quickly change strategy and will "back out of the loop" to stabilize the system as was shown in Refs. 13 and 14.

3.5 Data Displays

LOCATS supports a number of data displays that can be selected for use in a real-time environment or for off-line analyses. Selection of the display types is made via the Figure 18 menu.

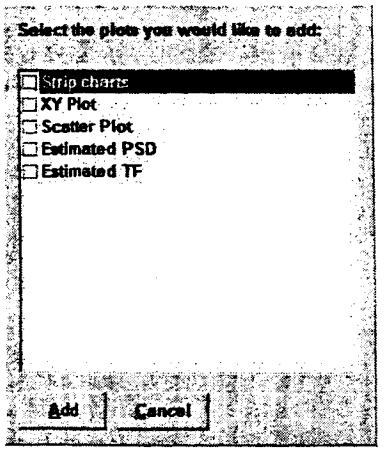


Figure 18. Selection of LOCATS Data Displays

The strip chart feature is available to view time series of critical signals. Up to four signals can be included in a single strip chart group, and multiple groups can be used simultaneously. In Figure 19 the strip chart time axis and signal selection windows are shown. In a real-time environment the time axis will scroll, as defined by user selected parameters, once the maximum time value is exceeded. Signals may be displayed using the Auto Scale feature or with user defined inputs. The strip chart feature also provides access to a wide range of advanced editing features (see Figure 19). An example four signal strip chart display is shown in Figure 20.

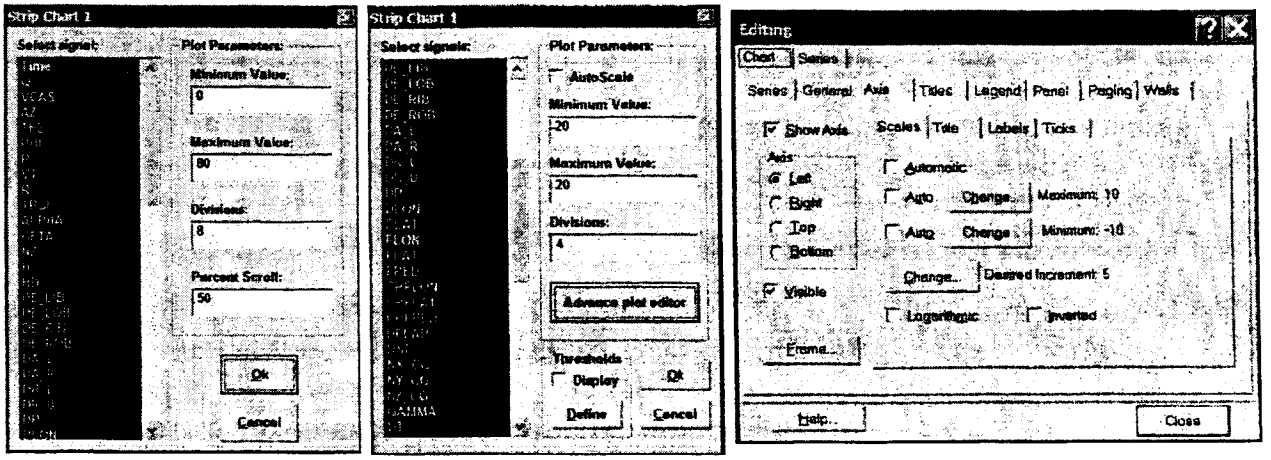


Figure 19. Strip Chart Time Axis Options, Signal Selection, and Advanced Editing Features

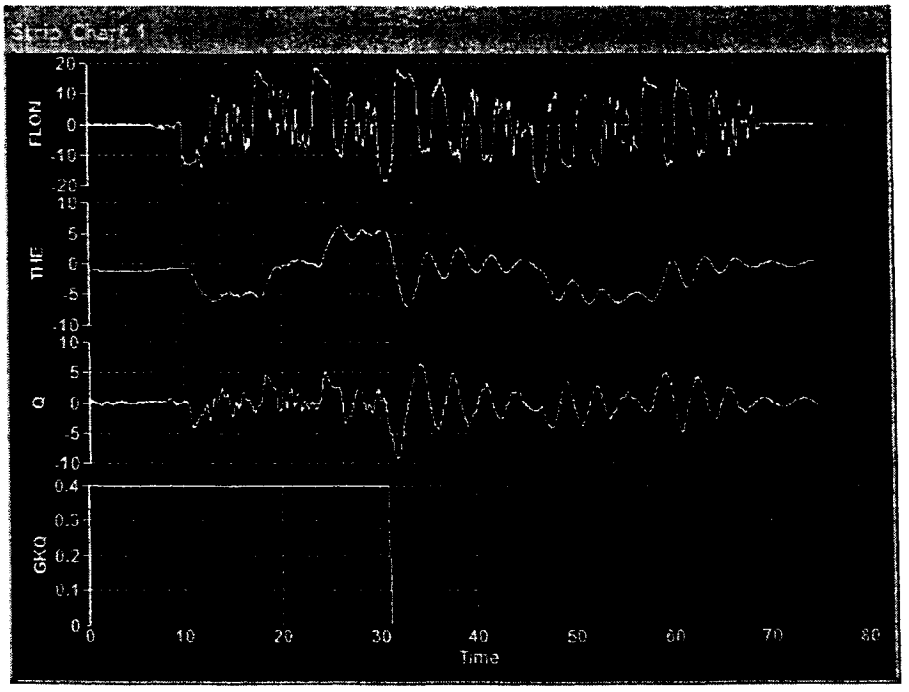


Figure 20. Example Strip Chart Display

A primary feature of LOCATS is the set of real-time analysis plots. To exercise these features, the user first selects a display type from the Figure 18 menu. Then by "clicking" on the corresponding item in the Active Plots list the Figure 21 window appears. From this window the display can be given a title and up to four data plot channels may be defined for each display. For the transfer function estimation displays,

the transform method is selected and access to the timing and setup parameters (see Figure 13) is provided.

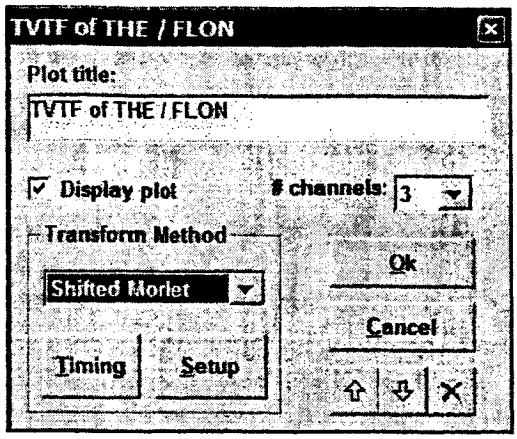


Figure 21. Estimated Transfer Function Options Menu

For the transfer function estimation displays access to the input and output signals selection and the frequency axis parameters menu shown in Figure 22 is gained by "clicking" on the Frequency label in the Active Plots listing. Corresponding plot types (e.g., magnitude Bode, phase Bode, etc.) and related plotting parameters (see the magnitude Bode example in Figure 22) are then accessed through the Channel 1 – Channel 4 labels in the Active Plots listing. The data display options can be conveniently stored in a configuration file for repeated use. An example TVTF display is shown in Figure 23. Note the persistence lines that provide a visual cue as to how the system is changing in time. The lines change from light to dark as they move back in time.

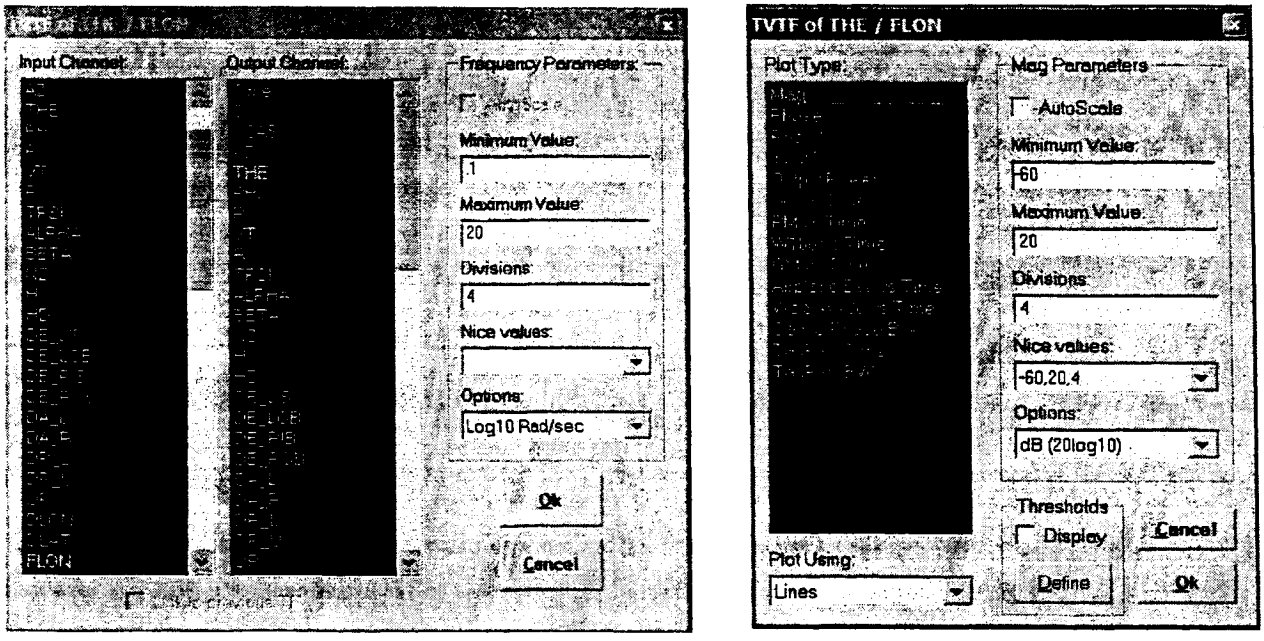


Figure 22. Example Analysis Signal Selection and Plot Option Menus

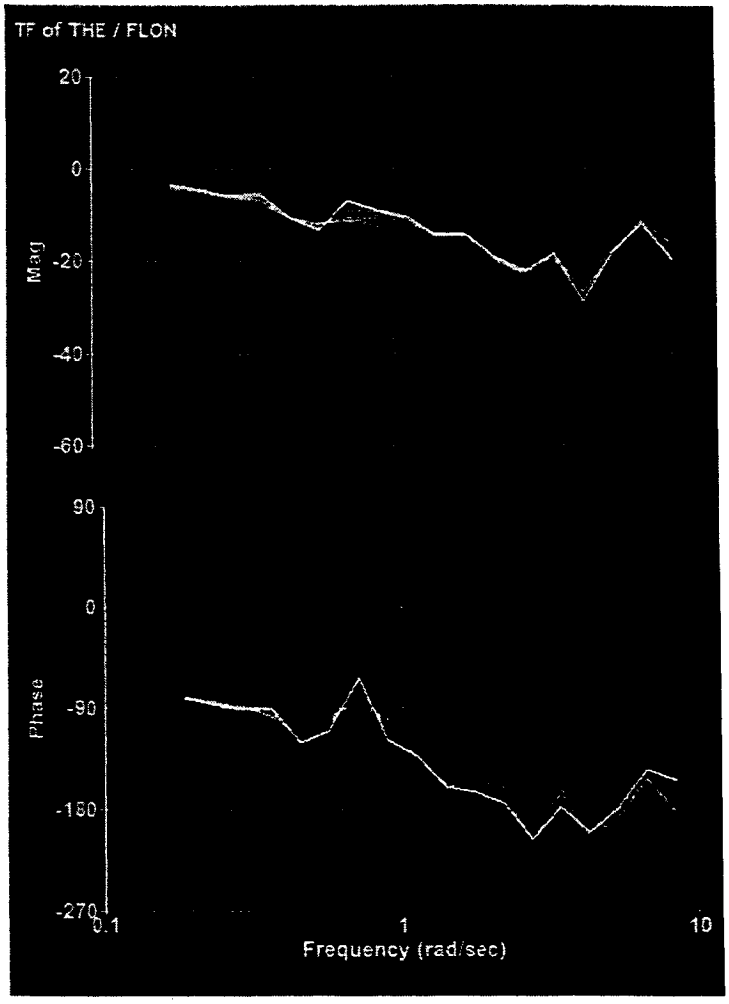


Figure 23. Example TVTF Data Display

4. TIME VARYING TRANSFER FUNCTIONS

4.1 Fourier and Wavelet Transforms

Wavelet transforms may one day become common in the systems and control community, but they are not yet so. A good way to introduce wavelet transforms is to start with the Fourier transform, ease out of this familiar territory by using windowed Fourier transforms, and then from there jump to wavelet transforms. This is the approach we now take, comparing and contrasting, and introducing the issue of causality. Further details and comparisons are included in Ref. 13.

4.1.1 Fourier Transform (FT)

A transform, any transform, of a time function is an alternative representation of that time function as a sum or an integral of basis functions. The most common transform in systems engineering is the Fourier transform, for which the basis functions are complex exponentials. The Fourier transform pair provides a means to compute the coefficients of each basis function, and a means to reconstruct the original signal:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{+j\omega t} d\omega$$

4.1.2 Windowed Fourier Transform (WFT)

The FT uses the entire time function and does not provide a means to identify where in time transient phenomena has occurred. It is meet this need for time localization that alternatives leading up to the wavelet transform are defined. The first step is the windowed Fourier transform, whereby the complex exponential is multiplied by a shifting time window:

$$F_g(\omega, u) = \int_{-\infty}^{\infty} f(t) g(t-u) e^{-j\omega t} dt$$

The WFT is then the frequency content of the input near the time $t = u$. The subscript g reminds us that a window is present. Reconstruction of the original signal is an important issue, for this or any transform, and can be done here but requires two integrations:

$$f(t) = \frac{1}{\|g\|^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F_g(\omega, u) g(t-u) e^{+j\omega t} dt du$$

For causality the time u is set at the endpoint of the window, or equivalently, $g(t)$ is nonzero only for negative time. This restriction allows the computation of $F_g(\omega, t)$ for $t \le u$ based only on $f(t)$ for $t \le u$, and also, to recover using the same restrictions.

Having just defined a transform with time localization, is this enough? For many applications perhaps yes, but the major drawback of the WFT is that the window is the same length for all of the frequencies. A long window appropriate for low frequencies phenomena will average out transients that may occur at high frequencies. Scaling the window length with frequency, so that low frequencies have long windows and high frequencies have short windows is the next logical step, and this in fact is what a wavelet transform does.

4.1.3 Wavelet Transform

The time function is decomposed into basis functions that translate in time and whose length and magnitude scale with a parameter that can be interpreted as frequency:

$$W_\phi(\omega,u) = \int_{-\infty}^{\infty} f(t)\omega^{1/2}\phi^*(\omega(t-u))dt$$

The magnitude is scaled so that the basis functions all have the same norm. All of the basis functions are derived from $\phi(t)$, the so-called "mother wavelet". The mother wavelet can be real or complex, and in many wavelet applications it is defined as a real function, but for applications such as LOCATS where phase information is important the complex exponentials are a natural choice, and therefore the following is used:

$$\phi(t) = g(t)e^{jt}$$

This brings the discussion back to a set of basis functions that look like those used for the WFT, but there is an important difference: the length (and the magnitude) of the window scales with frequency. If one measures the length of the mother wavelet in terms of wavelengths, then each of the scaled basis functions will have the same number of wavelengths. References 19, 20, 21, and 22 support the following developments.

The Fourier transform of the mother wavelet, $\hat{\phi}(\omega)$, scales and translates as follows:

$$\phi(t) \leftrightarrow \hat{\phi}(\omega)$$
$$s^{1/2}\phi(s(t-u)) \leftrightarrow s^{-1/2}e^{-jut}\hat{\phi}(\omega/s)$$

The mother wavelet as a function of time becomes taller and skinner with increasing scale, whereas its FT shrinks and gets wider. A translation in time corresponds to a phase shift.

The original time function is recovered using another double integral:

$$f(t) = \frac{1}{C}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_\phi(\omega,u)\omega^{1/2}\phi(\omega(t-u))\omega^2 dtdu \;,\; C = 2\pi\int_{-\infty}^{\infty}\frac{1}{|\omega|}\hat{\phi}(\omega)d\omega$$

The constant C is not finite unless the Fourier transform of the mother wavelet, $\hat{\phi}(\omega)$, is zero at $\omega = 0$, a condition that is equivalent to saying that $\phi(t)$ must have zero mean.

The wavelet transform, $W_\phi(\omega,t)$, can be interpreted as the frequency response of the time function, $f(t)$, in the neighborhood of the point ω,t in the time-frequency plane. The Morlet wavelet, a favorite from the Phase I effort, is defined using a Gaussian window, $g(t)$. A nice consequence is that $\hat{\phi}(t)$ also has a Gaussian window, "nice" because a Gaussian window is smooth. There is a fundamental limit to how localized the wavelet transform can be in both time and frequency, akin to the Heisenberg uncertainty principle, which will not be discussed here, except to say that another important property of the Morlet wavelet is that it minimizes the time-frequency localization.

Another way to interpret the wavelet transform is that $W_\phi(\omega,t)$ for each frequency is a convolution of $f(t)$ and $\omega^{1/2}\phi(\omega t)$. This fact can be used to calculate the wavelet transform, this is the method that was used in Phase I, and we also found it to be useful for real time calculations in Phase II.

4.2 Causal Wavelets

For causality it is required that $\phi(t)$ be nonzero only for negative time. In this case the wavelet transform $W_\phi(\omega,t)$ for $t \le u$ only depends on $f(t)$ for $t \le u$, and recovery likewise. Causality is needed for real time operation, but "causal wavelets" are not commonly used. Causality is discussed in Ref. 22, but not

even defined in other standard treatments of wavelets. In Ref. 13 several causal wavelets were investigated:

- Shifted Morlet, shifted to the left and the Gaussian tail truncated;
- Dryden's Laplace wavelet, $g(t) = e^{+at}$ for $t \leq 0$;
- Peter's Laplace wavelet, a variation that has zero mean; and
- Cosine tapered, where $g(t)$ is rectangular except first and last bit is cosine tapered.

All were implemented in the STI Wavelet Analysis Toolbox, as is the regular, non-causal Morlet. The shifted Morlet had been the preferred choice, mainly because when used with test functions the wavelet transform is smooth. The tradeoff is delay, and a better choice was needed that compromises smoothness and delay. A search for additional selections of $g(t)$ began with the one-sided probability density functions (PDFs), in particular the Gamma, Rayleigh, and Weibull PDFs that are basically skewed Gaussian and exponential functions. A description of these analytic wavelets is provided in the next section.

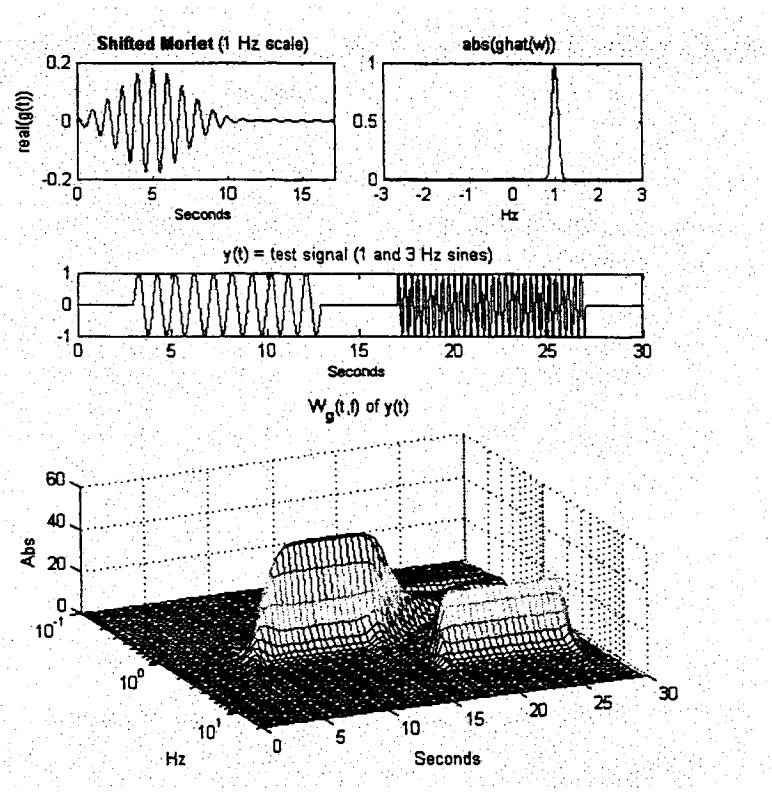


Figure 24. Shifted Morlet Wavelet Transform of a Test Signal

Figure 24 gives an example of a wavelet transform computed using the shifted Morlet basis functions. The shifted Morlet and its Fourier transform are shown in the top part of the figure. The test input $f(t)$ has two sinewaves, separated in time, at 1 and then 3 Hz. The WT shows two distinct concentrations of energy in the time-frequency plane. The WT is causal – the response along the time axis does not precede the input function. The time for the WT to reach a maximum is the same time it takes for the basis function to reach a maximum – about 3 wavelengths. This is the delay that we want to decrease. The WT response at 3 Hz is shorter by the square root of 3, and the width along the frequency axis is the same, at least when plotted as done here on a log frequency scale.

4.3 Analytic Wavelets

In the Phase I effort that is summarized in Ref. 13 the shifted Morlet wavelet was used for most of the analysis. It has nice properties, such as minimum energy localization in the time-frequency plane, and smoothness (a bell-shaped envelope) in both the time and frequency domain. This choice of wavelet suffers, however, from having a slow rise time in response to sudden changes in the signal. The additional wavelets evaluated in Phase II are defined below.

4.3.1 Shifted Morlet

The shifted Morlet wavelet is defined by,

$$w(t) = \frac{1}{\sqrt{\pi\beta}} \exp\left[j2\pi f_c (t-c) - (t-c)^2/\beta \right] \quad \text{for } t \geq 0, c > 0, \beta > 0$$

Mean = c

Variance = $\beta/2$

Number of cycles to reach maximum value (for $f_c = 1$) is $x = c$

The frequency of the wavelet is f_c Hz, and the mother wavelet is defined to have $f_c = 1$. The tail of the shifted Morlet for $t < 0$ is set to zero. The parameter values used in Phase 1 were $c = 3$ and $\beta = 2$. A larger set of values is now being considered.

Time and frequency domain plots for different parameter values are shown in Figure 25. The response of this wavelet to the test signal was shown previously in Figure 24.

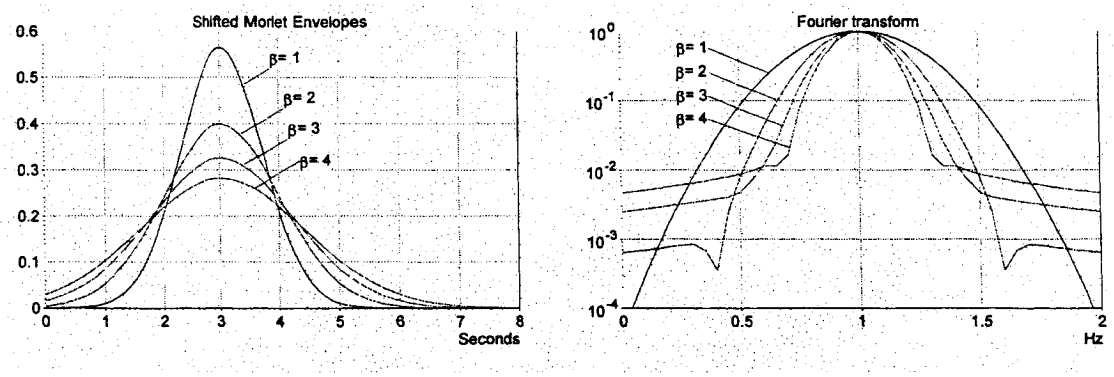


Figure 25. Shifted Morlet Wavelets and their Fourier Transforms

4.3.2 Rayleigh

The Rayleigh wavelet is defined by,

$$w(t) = \frac{2t}{\beta} \exp\left[j2\pi f_c t - t^2/\beta \right] \quad \text{for } t \geq 0, \beta > 0$$

Mean = $\sqrt{\beta\pi}$

Variance = $\beta(1-\pi/4) \approx \beta/5$

Number of cycles to reach maximum value is $x = \sqrt{\beta/2}$

The envelope is a skewed Gaussian curve. Currently we are using $\beta = 4.5$, so that the maximum is reached after 1.5 cycles.

Time and frequency domain plots for different parameter values are shown in Figure 26. The response of this wavelet to the test signal was shown previously in Figure 27.

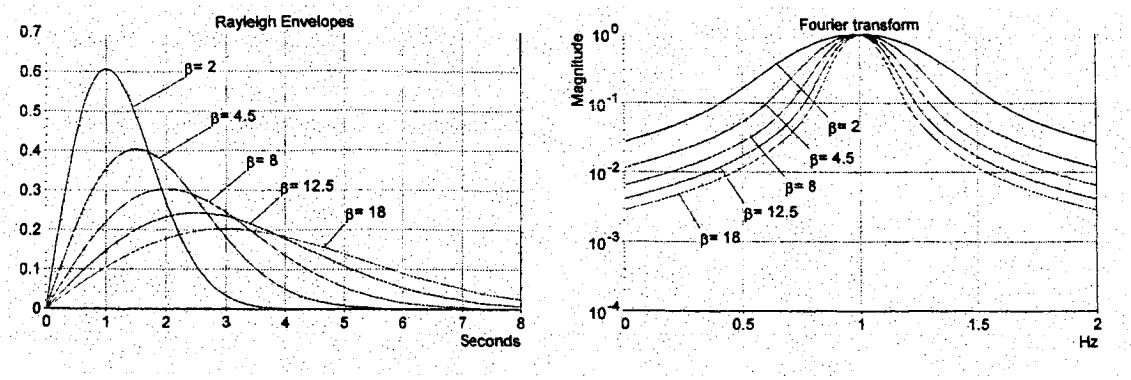


Figure 26. Rayleigh Wavelets and Their Fourier Transforms

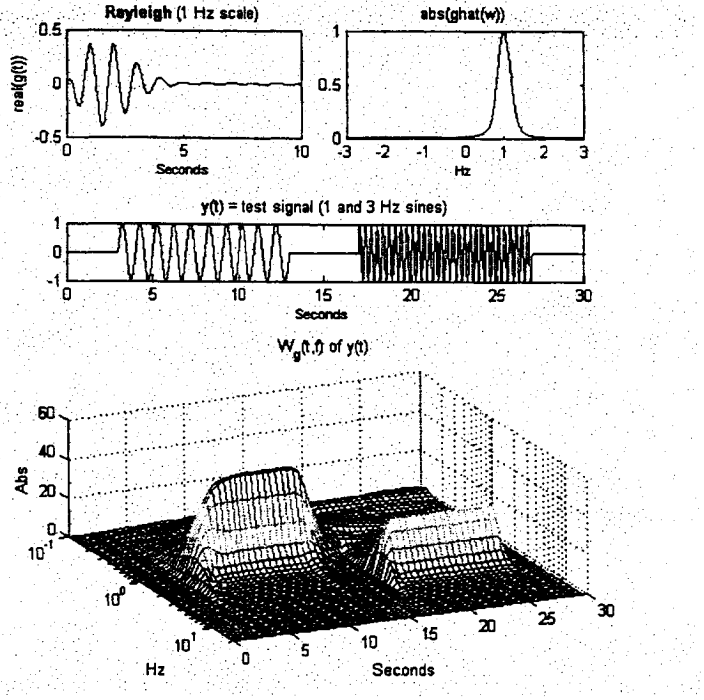


Figure 27. Rayleigh Wavelet Transform of a Test Signal (β = 4.5)

4.3.3 Erlang

The Erlang wavelet is defined by,

$$w(t) = \frac{a^n t^{n-1}}{(n-1)!} \exp[j2\pi f_c t - at] \quad \text{for } t \geq 0, a > 0, n > 1$$

Mean = n/a

Variance = n/a^2

Number of cycles to reach maximum value is $x = (n-1)/a$

The envelope is a power of t times an exponential of t. Currently we are using $n=3$ and $a=4/3$, so that the maximum is reached after 1.5 cycles.

Time and frequency domain plots for different parameter values are shown in Figure 28. The response of this wavelet to the test signal was shown previously in Figure 29.

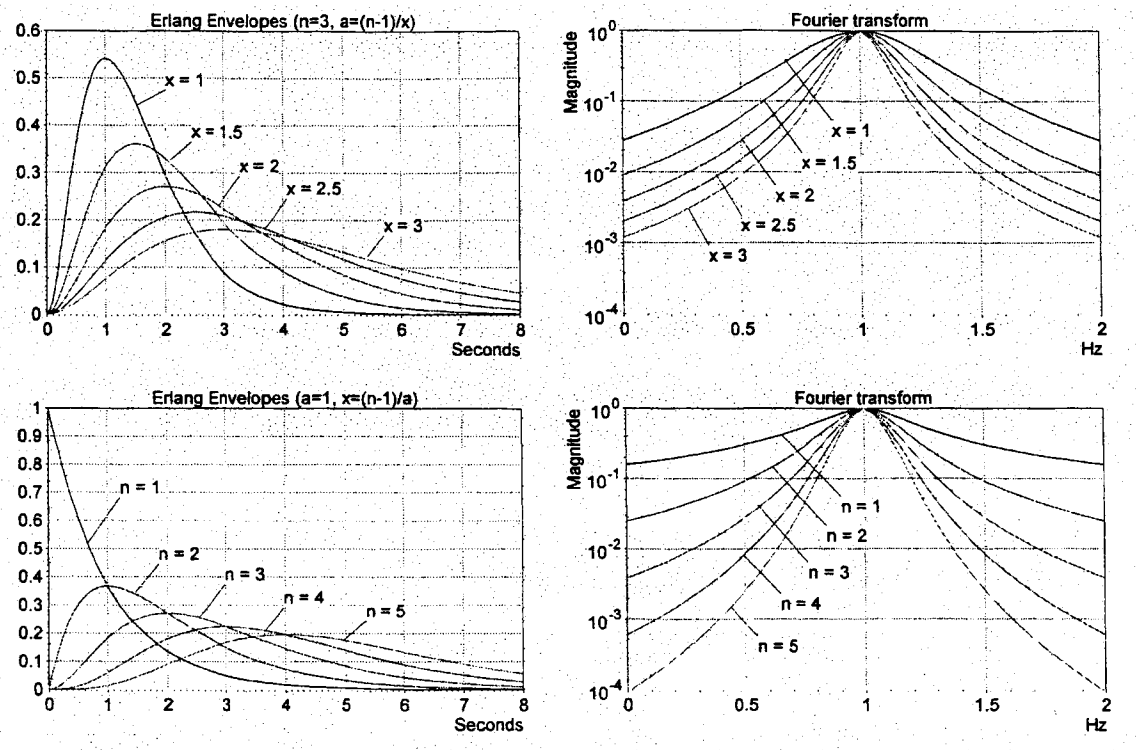


Figure 28. Erlang Wavelets and their Fourier Transforms

4.3.4 Weibull

The Weibull wavelet is defined by,

$$w(t) = abt^{b-1}\exp\left[j2\pi f_c t - at^b \right] \quad \text{for } t \geq 0,\ a>0,\ b>0$$

$$\text{Mean} = (1/a)^{(1/b)}\,\Gamma(1+1/b)\ , \text{ where } \Gamma(c) = \int_0^\infty x^{c-1}e^{-x}dx$$

$$\text{Variance} = (1/a)^{(2/b)}\left[\Gamma(1+2/b) - \Gamma^2(1+1/b)\right]$$

$$\text{Number of cycles to reach maximum value is } x = \left(\frac{b-1}{ab}\right)^{(1/b)}$$

The envelope is a power of t times an exponential of a power of t. Currently we are using $b=3$ and $x=1.5$. The expressions for the mean and variance are cumbersome, but the parameters b and x have the most meaning, from which the others can be calculated.

Time and frequency domain plots for different parameter values are shown in Figure 30. The response of this wavelet to the test signal was shown previously in Figure 31.

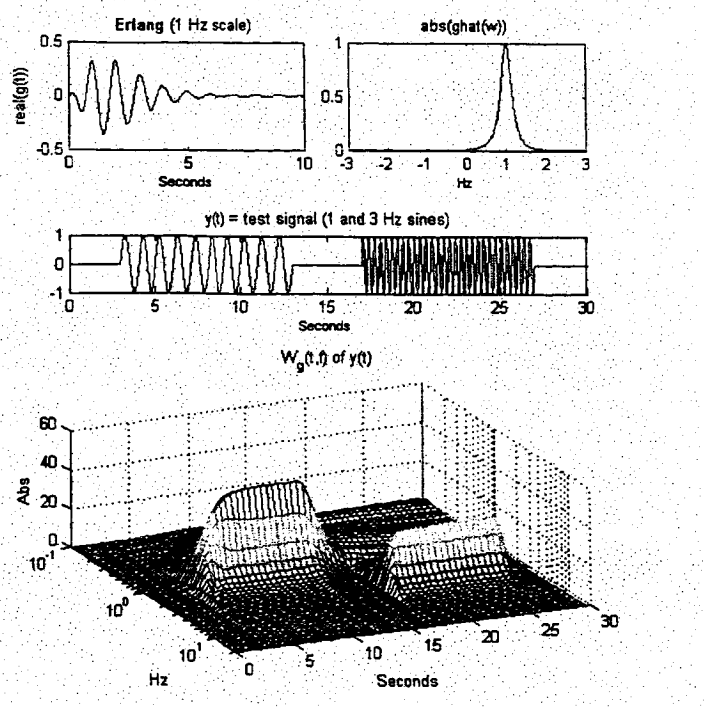


Figure 29. Erlang Wavelet Transform of a Test Signal (a = 2, n = 3)

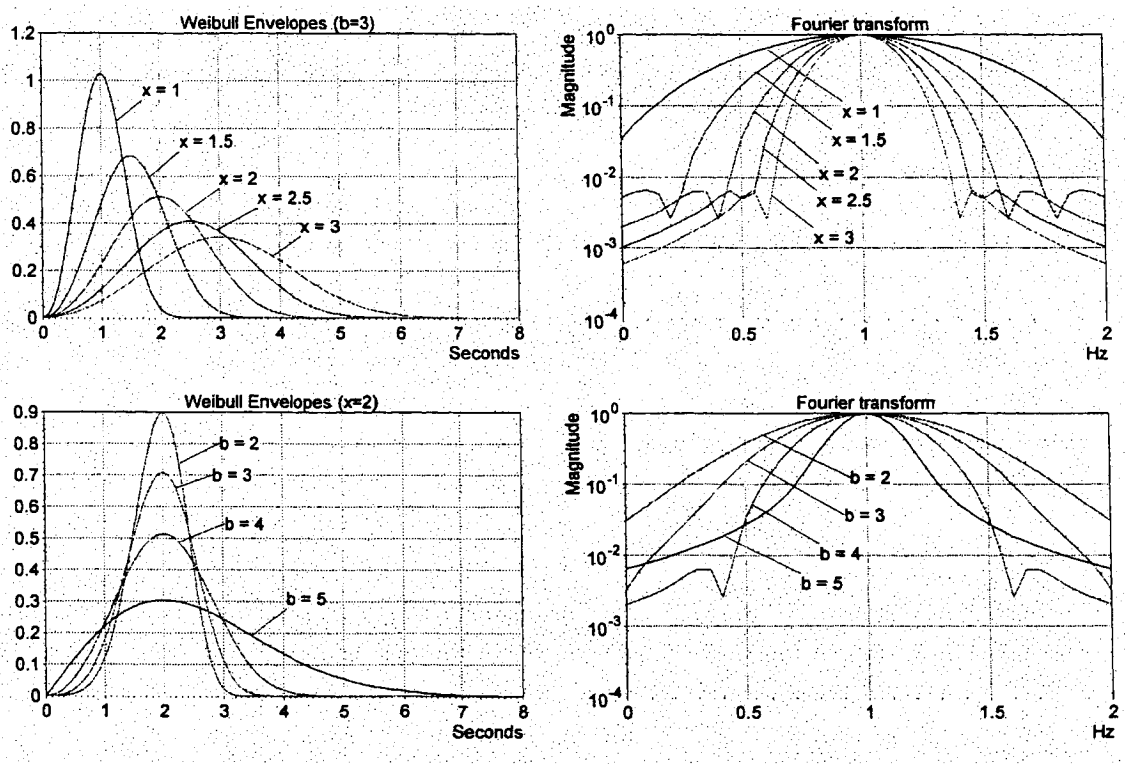


Figure 30. Weibull Wavelets and their Fourier Transforms

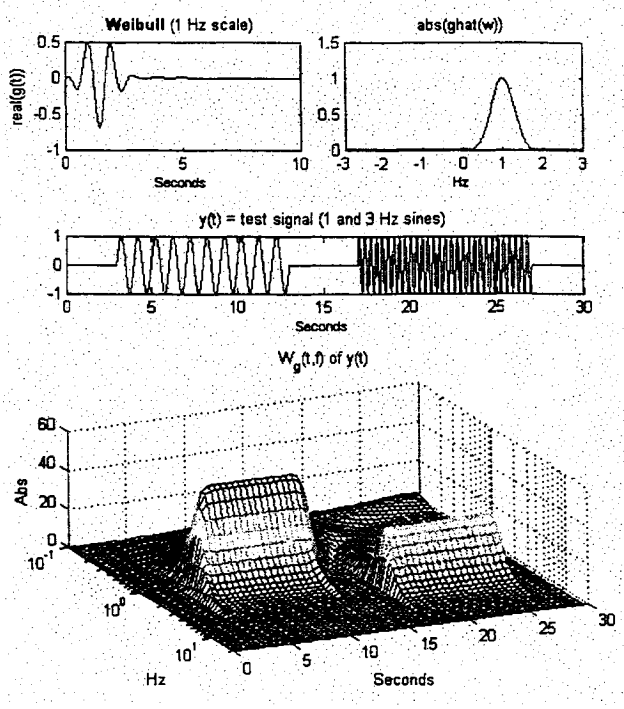


Figure 31. Weibull Wavelet Transform of a Test Signal (a = 0.1975, b = 3)

Through analysis of various data sets including the LOCATS simulation data the Rayleigh wavelet was found to possess the characteristics required for real-time detection of changes in a dynamic system. These characteristics will be demonstrated later in the report.

4.4 Discrete Wavelet Transforms

The connection between digital and continuous wavelets is one area where the vast technical literature does not provide much help. For the most part the study of the two types of wavelets are disconnected.

- The study of continuous wavelet transforms begins with the definition of the wavelet function, then the transform equation, and then the properties of the resulting wavelet transform. Families of wavelets are defined, the most famous being the Morlet wavelet.
- The study of digital wavelet transforms starts with the selection of a high and low pass filter, then the transform algorithm, and then the properties of the resulting wavelet transform. What is missing is the wavelet function. It enters as a secondary item, something that can be computed if you want to, but not really necessary, and not usually done. The filters are considered to be more fundamental. Families of digital wavelets are defined, the most famous being Daubechies wavelets, but they are really families of high and low pass filters. The Daubechies wavelets are used in the WERA method discussed in the next section.

The literature on this subject is not barren, however, and we have the example of Ref. 23 where this approximation is implemented. The objective is to understand this work and to generalize it. The main reference being used for the study of digital wavelets is the textbook by Strang and Nguyen (Ref. 22).

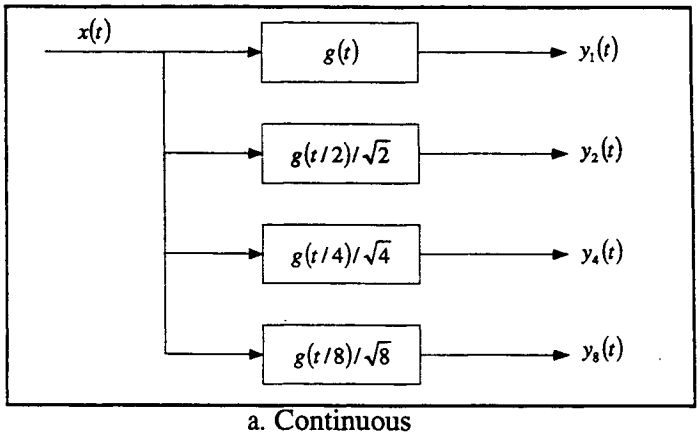
Both the continuous and digital wavelets can be interpreted as filter banks, and this is shown in Figure 32. A range of frequencies is selected, each being an octave apart. The continuous wavelet at the highest frequency is implemented as a filter with the impulse response $g(t)$. The CWT has no restrictions on the

choice of frequencies, this is a strength of the method, but for now the "dyadic" selection of frequencies that is imposed by the DWT is duplicated.

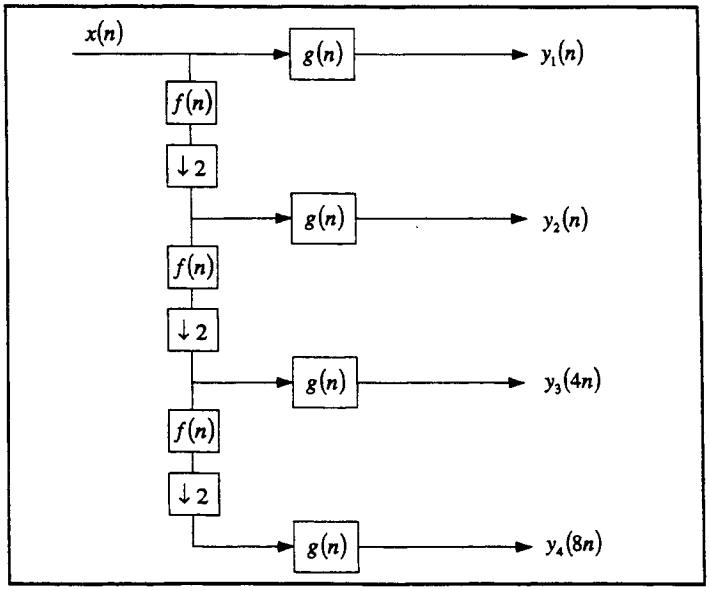
The DWT is equivalent to the filter bank using the high pass filter $g(n)$ and the low pass filter $f(n)$. After each low pass filter the signal is down-sampled, and operation that selects every other sample. The low and high pass filters are the same in each branch of the filter bank. The speed of the DWT comes from the fact that these filters are each short finite impulse response (FIR) filters that can be quickly computed. One of the ways to approximate the DWT is suggested by the notation used in Figure 32:

$$g(n) = g(t) \text{ at } t = n\Delta$$

where Δ is the sample period. In words, the digital high pass filter is set equal to the sampled continuous wavelet defined at the highest frequency.



a. Continuous



b. Digital

Figure 32. Filter Banks Based on the Continuous and Digital Wavelet Transforms

The digital high and low pass filters are inter-related and there are conditions of validity that must, or least should, be satisfied. Any low pass filter can be used in Figure 32b and implemented, but only those satisfying certain conditions provide perfect reconstruction, one of the requirements of digital wavelet

transforms. Following the development in Chapter 4 of Ref. 22, the two channel filter bank with both analysis and synthesis portions is shown in Figure 33.

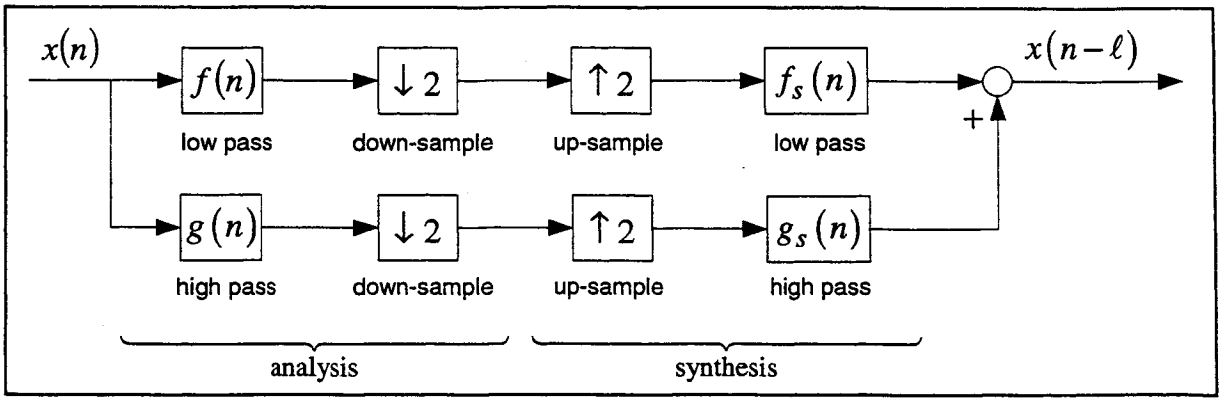


Figure 33. Two channel filter bank with both analysis and synthesis

The up-sample operation places a zero between each sample. The two requirements for perfect reconstruction are:

$$\text{Alias cancellation: } f(z)f_s(-z)+g(z)g_s(-z)=0$$

$$\text{No distortion: } f(z)f_s(z)+g(z)g_s(z)=2z^{-\ell}$$

These conditions use the z-transforms of the filters. For FIR filters, as used here, these z-transforms are polynomials in z^{-1}. Perfect reconstruction means that the original signal is recreated with only a delay. The alias cancellation requirement is satisfied with the following choices of synthesis filters:

$$\text{For alias cancellation: } f_s(z)=g(-z) \text{ and } g_s(z)=-f(-z)$$

The no distortion requirement is then:

$$f(z)g(-z)-g(z)f(-z)=2z^{-\ell}$$

With the substitution: $p(z)=f(z)g(-z)$, this requirement can be written as:

$$p(z)-p(-z)=2z^{-\ell}$$

The design of digital wavelet transforms can be succinctly stated in two steps:

- Select $p(z)$; and
- Factor into $f(z)$ and $g(-z)$.

The most famous of DWTs, the Daubechies transforms, makes the selection:

$$p(z)=\left(1+z^{-1}\right)^{2p}Q(z)$$

where $p=\ell$ is the order of the Daubechies wavelet, and the polynomial $Q(z)$ is of order $2p-2$. The no distortion requirement is used to define $2p-1$ linear equations that the coefficients of $Q(z)$ must satisfy. The polynomial $p(z)$ turns out to be all pass, meaning the inverse of each root is also a root, and spectral factorization is used to factor $p(z)$ into $f(z)$ and $g(z)=-z^{-p}f(-1/z)$.

Coming from continuous wavelets, the following method is used:

- Sample $g(t)$ at the highest frequency to define $g(n)$;
- Select the low pass filter $f(n)$ to be the same order; and
- Use the no distortion requirement to create a linear matrix equation that uniquely defines the coefficients of $f(n)$.

This has been done, but problem remain. The conditions for low, high, and band pass filters are stated in Table 2 for digital and continuous transforms.

Table 2. Conditions for Digital and Continuous Filter Transforms

	Low pass	High pass	Band pass
Digital	$g(z)=0$ at $z=-1$	$g(z)=0$ at $z=1$	$g(z)=0$ at $z=\pm 1$
Continuous	$g(s)=0$ at $s=\infty$	$g(s)=0$ at $s=0$	$g(s)=0$ at $s=0,\infty$

A digital band pass filter cannot satisfy the no distortion requirement. The no distortion requirement cannot be satisfied if $p(z)$ and $p(-z)$ have a common root, say at $z=\pm 1$. Then the requirement becomes:

$$p(z)-p(-z)=(z+1)(z-1)[p_0(z)-p_0(-z)]=2z^{-\ell}$$

The remaining factors cannot cancel these roots, because the remaining factors are polynomials. The no distortion condition cannot be satisfied, and therefore a digital band pass filter cannot be used in place of the high pass filter in the DWT.

Many of the CWTs are defined using analog band pass filters. In particular, the Morlet wavelet is equivalent to a band pass filter. The sampled Morlet wavelet is not band pass, but almost so, and is so in the limit as more and more samples are used. Even though not quite band pass, the no distortion condition is numerically ill-conditioned, resulting in a low pass filter that has coefficients for $f(z)$ that are orders of magnitude higher than the coefficients for $g(z)$. They should be the same size, ideally, both norm one. In the end, CWTs were selected for implementation in LOCATS.

4.5 Computing the Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) is:

$$W_f(\omega,u)=\int_{-\infty}^{\infty} f(t)\omega^{1/2}\phi^*(\omega(t-u))dt$$

For a given frequency this is a convolution, and the CWT is computed as a set of convolutions. The subscript on W is changed to indicate the time series (whereas before the subscript was the wavelet function).

4.5.1 Post-Processing

When using stored data the convolution is computed using FFTs. The signals $f(t)$ are $\omega^{1/2}\phi^*(\omega t)$ are each sampled over the time window of the stored data for a total of N points each. Forward FFTs of length $2N-1$ are computed of each to prevent circular convolution, multiplied, and then the inverse FFT is computed to give $W_f(\omega,u)$ at the selected frequency. This is the calculation used in the STI's Matlab Wavelet Toolbox. Further details are in Ref. 13.

4.5.2 Real Time

The real time calculations in LOCATS directly computes the convolution as an inner product of the time signal $f(t)$ and $\omega^{1/2}\phi^*(\omega t)$, however to save computation time the wavelet is computed and stored one time with N points, using N appropriate for the lowest frequency. Typical values are N equal 4096 or 8192 – though there is no time savings here for powers of two. The stored wavelet is down-sampled and/or interpolated for each of the higher frequency wavelets before computing the convolution.

The number of multiplications for each transform is:

$$N_{total} = N \sum_{i=0}^{N_w-1} \frac{\omega_0}{\omega_i}$$

where ω_0 is the lowest frequency and the ω_i are the remaining frequencies. If these frequencies are log-spaced and the highest is ω_{max}, it works out that:

$$N_{total} = N \frac{1-(\omega_0/\omega_{max})}{1-(\omega_0/\omega_{max})^{1/(N_w-1)}}$$

Typical values are $\omega_0/\omega_{\max} = 1/300$ and $N_w = 120$, in which case $N_{total} \approx 34N$, and which is significantly less than N^2 for large values of N.

4.6 Power Spectrum Estimation

The main objective is to estimate the frequency response of a system, but an important first step is to estimate the power spectrum of the input and output time series. The computed power spectra are estimates, and as with all estimates there is a mean and variance of the estimate. Determining the statistics of the estimate is the subject of Appendix A, using material mainly from Ref. 27. The main results from the appendix are summarized here.

4.6.1 Estimation Method

The objective is to estimate the power spectra $S_{xx}(f)$ of the time series $x(t)$ at a single frequency $f = f_0$ Hz. It is assumed that $x(t)$ is a stochastic processes defined as zero-mean white noise passed through a LTI system. The estimation method uses a LTI system, $H(s)$, a narrowband filter centered at f_0 Hz. The power spectra estimate is obtained as shown in Figure 34. The output of the narrowband filter is the time series $y(t)$, and the estimate of its mean square value computed over the previous T seconds is $\hat{\psi}_y^2$. The estimate is normalized by $R_{hh}(0)$, where $R_{hh}(\tau)$ is the autocorrelation of $H(s)$. The end result is the estimate of the power spectra, $\hat{S}_{xx}(f_0)$ at the frequency f_0.

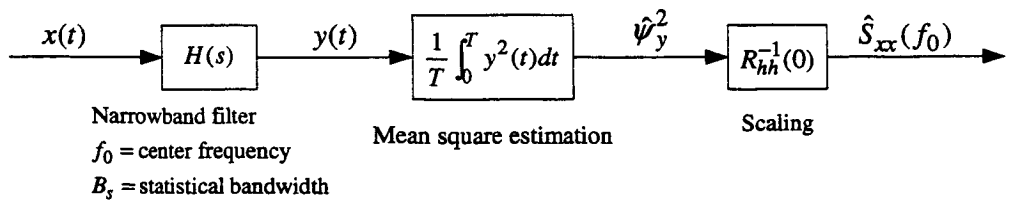


Figure 34. Power Spectra Estimate

An important parameter is the bandwidth of the narrowband filter. Qualitatively, the bandwidth is the width in frequency where the spectrum of the filter has most of its power. The particular definition of bandwidth that works the best for power spectral estimation is the so-called "statistical bandwidth:"

$$B_s = \frac{\left[\int_{-\infty}^{\infty} S_{hh}(f)df\right]^2}{2\int_{-\infty}^{\infty} S_{hh}^2(f)df} = \frac{R_{hh}^2(0)}{2\int_{-\infty}^{\infty} R_{hh}^2(\tau)\tau} = \frac{R_{hh}^2(0)}{2R_{h_1h_1}(0)}$$

Several equivalent expressions for the statistical bandwidth are given above. All have their use for theory, and it turns out the last of these, using $h_1(t) = h(t) * h(t)$ is the best for computation.

4.6.2 Statistics of the Estimate

Using the spectrum estimate in Figure 34, the mean value of the estimate is:

$$E[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}(0)}\int_{-\infty}^{\infty} S_{hh}(f)S_{xx}(f)df \approx S_{xx}(f_0)$$

This is a biased estimate, being less so as the bandwidth of $H(s)$ decreases. The variance of the estimate is:

$$\text{var}[\hat{S}_{xx}(f_0)] \approx \frac{1}{B_sT}S_{xx}^2(f_0)$$

The variance is proportional to the square of the actual value of the spectrum. The proportionality constant is an important metric, highlighted by the following normalized error, and good when the bias is small:

$$\varepsilon[\hat{S}_{xx}(f_0)] = \frac{\text{rms}[\hat{S}_{xx}(f_0)]}{S_{xx}(f_0)} = \frac{1}{\sqrt{B_sT}}$$

The bandwidth time product, B_sT, is used to compare the different transform estimation methods.

4.6.3 Power Spectrum Estimate Using Finite Time Fourier Transforms

The finite time Fourier Transform of the time series $x(t)$ is:

$$F_y(\omega_0, t) = \int_{t-T}^{t} x(\tau)e^{-j\omega_0\tau}d\tau$$

This is equivalent to convolution with the filter:

$$h(t) = e^{-j\omega_0(T-t)}$$

defined for $0 \le t < T$ and where $\omega_0 = 2\pi f_0$. The bandwidth-time constant for this filter works out to be:

$$B_sT = \frac{3}{4}$$

The terms B_s and T, and the product B_sT are all constants that do not depend on frequency.

4.6.4 Power Spectra Estimate Using Morlet Wavelet Transforms

The Morlet Wavelet Transform of the time series $x(t)$ is equivalent to the convolution:

$$W_y(\omega_0, t) = \int_{t-T}^{t} x(\tau) h(t-\tau) d\tau$$

for:

$$h(t) = \sqrt{f_0}\phi^*(f_0 t)$$

$$\phi(t) = e^{-j2\pi t} g(t)$$

$$g(t) = \frac{1}{\sqrt{\pi\beta}} e^{-t^2/\beta}$$

The statistical bandwidth for the filter $h(t)$ is:

$$B_s = \frac{f_0}{\sqrt{4\pi\beta}}$$

The length of the transform is defined to be $T = 6\sigma = N_c / f_0$, where $\sigma = (\sqrt{\beta/2})/f_0$ is the standard deviation of the Gaussian envelope, and N_c is the number of cycles of the wavelet frequency. Substituting, we get:

$$B_s = \frac{f_0}{N_c\sqrt{2\pi/9}} \approx \frac{1.2 f_0}{N_c}$$

The bandwidth-time constant for this filter is:

$$B_s T = \frac{1}{\sqrt{2\pi/9}} \approx 1.2$$

The terms B_s and T depend on frequency, but the product $B_s T$ is constant. The statistical bandwidth will change for different mother wavelets. The exact value has only been determined for the Morlet wavelet.

4.6.5 Reducing the Variance of the Power Spectrum Estimate

The variance of the power spectrum estimate is inversely proportional to the bandwidth-time product $B_s T$. Increase the size of this "rectangle" or "frame" in the time-frequency plane to reduce the variance. Comments:

- The time dimension T can be increased, but doing so is tricky because the statistical bandwidth is inversely proportional to T. The most straightforward approach (of increasing T and letting the statistical bandwidth change accordingly) does not change $B_s T$ and hence does not reduce the variance of the estimate. This is true for Fourier transform estimates, and it is also true for Wavelet transform estimates.
- Averaging power spectrum estimates computed at different times increases the "effective" value of T. Define the total time window as $T_{total} = n_t T$. If the time windows overlap then n_t is smaller then the number of individual estimates. The statistical bandwidth remains the same, and the effective bandwidth-time product increases to $n_t T B_s$. The tradeoff for $n_t T$ is reduced variance versus increased time to detect changes.

- The statistical bandwidth can be increased, but again use averaging to increase the effective bandwidth-time product. Define the total frequency window as $B_{s_{total}} = n_f B_s$, where n_f is smaller than the number of individual estimates if the bandwidths overlap. Combine both time and frequency averaging so that the effective bandwidth-time product is $n_t n_f T B_s$.

- The tradeoff for $n_f B_s$ is reduced variance versus increased bias of the mean. The estimate of the mean is average power about the center frequency, and the lightly damped resonant modes will be obscured if the effective bandwidth of the estimate is too large.

- A difficult case is power spectrum estimation of a lightly damped resonant mode. To estimate a damping ratio ζ a reasonable bound for the effective statistical bandwidth is:

$$n_f B_s = \zeta f_0$$

- Substitute $B_s = 1.2 f_0 / N_c$ for Morlet wavelets and then the center frequency f_0 drops out, leaving:

$$1.2 n_f / N_c = \zeta$$

- For example, if $\zeta = .05$ then the equation requires that $N_c / n_f = 24$. The wavelet needs to be defined with at least 24 cycles of the center frequency. The damping ratio restricts the freedom available in the frequency domain, and so the only way to reduce the variance of the power spectrum estimate is to use overlapping time windows.

To summarize, the normalized error of the power spectrum estimate is reduced by time and frequency domain averaging, with respectively n_t and n_f degrees of freedom in the time and frequency domain:

$$\varepsilon[\hat{S}_{xx}(f_0)] = \frac{\text{rms}[\hat{S}_{xx}(f_0)]}{S_{xx}(f_0)} = \frac{1}{\sqrt{n_t n_f B_s T}}$$

4.7 Time Varying Transfer Function Estimation

A system transforms an input signal to an output signal. The innovation of the work described herein is to use the wavelet transform of the input and output signals to estimate a time varying frequency response. This is done using techniques similar to those that successfully use Fourier transforms, but now with two advantages: (1) the ability to estimate time variations in the system and (2) faster convergence of the estimate. The frequency response that is estimated is called a "time varying transfer function (TVTF)." The convergence time of the estimate depends on the frequency of the wavelet basis functions – faster at high frequencies and slower at low frequencies.

4.7.1 Transfer Function Estimation Using Finite Time Fourier Transforms

In Figure 2 the open loop system is a linear time invariant (LTI) with the transfer function $g(s)$.

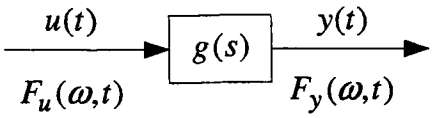


Figure 35. Transfer Function Estimation Using Fourier Transforms

The frequency response is $g(\omega)$, defined as $g(s)$ at $s = j\omega$. Under ideal conditions the frequency response estimate is:

$$\hat{g}(\omega,t) = \frac{F_y(\omega,t)}{F_u(\omega,t)}$$

The "hat" indicates an estimated frequency response. The estimate is made at time t based on input and output data over the previous T seconds. The transforms are either Finite Time Fourier Transforms (FTFT), or when using sampled data, as will always be the case, Digital Fourier Transforms (DFT). The latter are computed using the Fast Fourier Transform (FFT) algorithm.

The ratio of transforms works well when the input Single to Noise Ratio (SNR) of the input is large, and can be obtained in practice by inputs that are narrowband Gaussian noise or a sum-of-sines.

A more robust estimate uses a smoothed estimate of the cross-spectral density divided by a smoothed estimate of the input power spectra density:

$$\hat{g}(\omega,t) = \frac{\tilde{S}_{yu}(\omega,t)}{\tilde{S}_{uu}(\omega,t)}$$

The "tildes" indicate smoothing, shown below as an average over a frequency range:

$$\tilde{S}_{\alpha\beta}(\omega_{avg},t) = \frac{1}{\omega_1 - \omega_0}\int_{\omega_0}^{\omega_1} F_\alpha(\omega,t)F_\beta^*(\omega,t)d\omega$$

In the digital domain the smoothing is the average of adjacent frequency bins.

The coherence measures the relative error of the output power spectral density:

$$\rho^2(\omega,t) = \frac{\left|\tilde{S}_{yu}(\omega,t)\right|^2}{\tilde{S}_{uu}(\omega,t)\tilde{S}_{yy}(\omega,t)}$$

For linear time invariant systems the coherence is ideally one, indicating all of the output power is correlated with the input. Coherence less than one can be due to measurement noise, other inputs assumed to be zero but not actually zero, transient responses that have not yet settled, nonlinearities in the system, and/or time variations in the system.

The coherence is one part of a more complicated expression for the confidence interval for $\hat{g}(\omega)$, see Ref. 27. The confidence interval decreases as the coherence gets closer to one, as expected, but just as important is the fact that the confidence interval decreases as n, the number of degrees of freedom in the estimate increases. The degrees of freedom increase as the length of the time series increases, as more adjacent FFT bins are averaged, and as the tapering of the time series window increase. These effects are

related, but basically the tradeoff is that lower frequencies, smaller frequency resolution, and smaller confidence intervals all require longer time series.

4.7.2 Time Varying Transfer Function Estimation Using Wavelet Transforms

Start with the same LTI system and compute wavelet transforms of the input and output:

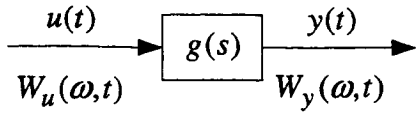


Figure 36. Transfer Function Estimation Using Wavelet Transforms

Under ideal conditions the frequency response estimate is:

$$\hat{g}(\omega,t)=\frac{W_y(\omega,t)}{W_u(\omega,t)}$$

This ratio was used in Ref. 13 with sum-of-sine inputs. The main difference with Fourier methods is the dependence of the estimate on time. At the frequency ω_0 define the length of the wavelet to be T_0, and therefore the estimate $\hat{g}(\omega_0,t)$ depends on input and output data for the previous T_0 seconds. The time dependence scales with frequency, so that at frequency ω_1 the estimate $\hat{g}(\omega_1,t)$ depends on input and output data over the previous $T_1=(\omega_0/\omega_1)T_0$ seconds. The time window is smaller at higher frequencies, hence the ability to responds more rapidly to changes at high frequencies.

A more robust estimate uses a smoothed estimate of the cross-spectral density divided by a smoothed estimate of the input power spectra density. The same notation is used for the smoothed spectra:

$$\hat{g}(\omega,t)=\frac{\tilde{S}_{yu}(\omega,t)}{\tilde{S}_{uu}(\omega,t)}$$

where the average is over both time and frequency:

$$\tilde{S}_{\alpha\beta}(\omega_{avg},t)=\frac{1}{(\omega_1-\omega_0)(t-t_0)}\int_{\omega_0}^{\omega_1}\int_{t_0}^{t}W_\alpha(\omega,t)W_\beta^*(\omega,t)dtd\omega$$

The average frequency is used and the current time. In the digital domain the smoothing in the frequency domain is the average of adjacent frequencies.

4.8 Estimating Time Varying Transfer Function Coefficients

Transform methods are used to estimate the time varying transfer function (TVTF). The frequency response is estimated at many different times, and from this set of frequency responses parameters are estimated. One such set of parameters is the numerator and denominator coefficients of a transfer function. The parameters are estimated from each frequency response, and hence the parameters are time varying. Estimating transfer function coefficients is the subject of Appendix B. The results are summarized here.

4.8.1 Problem Statement

The estimation problem starts with input and output data:

$$u_i = \text{input data}$$

$$y_i = \text{output data}$$

defined for a set of time steps, and the object is to estimate the coefficients of the z-transform:

$$G(z) = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0}$$

There are many parameter estimation methods to choose from grouped into the main categories shown in Figure 37.

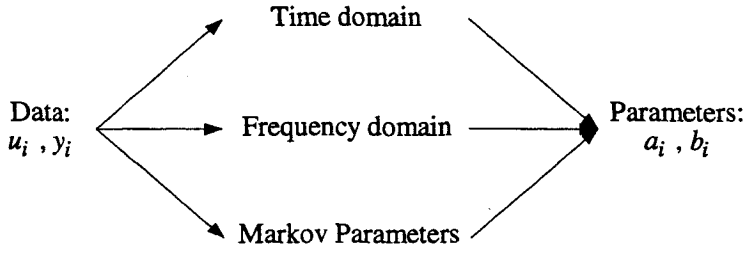


Figure 37. Transfer Function Coefficient Estimation Methods

Time domain methods work directly with the input and output data. Frequency domain methods start with a frequency response, from which the parameters of the transfer function are estimated. Markov parameters methods first estimate Markov parameters, which are the impulse response of the system, and from which the transfer function parameters are estimated.

Mathematical tools for the parameter estimation problem include linear and nonlinear least square minimization, difference equation solutions, and orthogonal polynomials. Troublesome issues include initial conditions, sensitivity to noise, determining the system order, numerical stability, and stability of the estimated system. All of these subjects are discussed in Appendix B and in the references that are listed there. The methods included in Figure 37 overlap, and conditions are derived in Appendix B where the different methods give exactly the same solution.

4.8.2 Time Domain Methods

The difference equation from which the z-transform is derived is used to define a one-step prediction:

$$\hat{y}_k = -\left(a_{n-1} y_{k-1} + \cdots + a_0 y_{k-n}\right) + \left(b_m u_{k-n+m} + \cdots + b_0 u_{k-n}\right)$$

The objective is to minimize the weighted error of this one-step prediction:

$$J = e^T W e \ , \text{ where } e = \begin{bmatrix} e_0 \\ \vdots \\ e_{N-1} \end{bmatrix} \text{ and } e_k = \hat{y}_k - y_k$$

The error depends linearly on the transfer function coefficients, and the solution is found by linear least square minimization:

$$\theta = (WF)^+ W y$$

The matrix F and vector y are created from input and output measurements, the superscript + indicates a pseudo-inverse, and the vector θ contains the transfer function coefficients. A variation on this problem is to minimize the simulation error, which makes this a nonlinear minimization problem.

The time domain linear least square parameter estimation problem is a classic problem in system theory. Iterative solutions can be used to speed calculations and to embed the estimated parameters into adaptive control. A nice feature is that the time domain solution does not depend on the initial conditions, and in particular the initial conditions do not have to be zero. Also, this method is not sensitive to constant offsets in the data. Time domain methods, however, are sensitive to noise, nonlinearity, data skewing and other practical problems, and reliability is a serious issue. We have included time domain methods mainly as a basis of comparison, something to do better than. A good way to increase the reliability of the estimated parameters is to switch to the frequency domain.

4.8.3 Frequency Domain Methods

The error to be minimized is the frequency response, obtained using either Fourier or wavelet methods. To define a frequency domain error, multiply the z-transform through by the dominator:

$$\left(z^n + a_{n-1}z^{n-1}\cdots + a_0\right)G(z) = \left(b_m z^m + \cdots + b_0\right)$$

Define $z_i = e^{j\omega_i h}$ and G_i respectively as digital frequencies and frequency responses, where h is the sample period. The objective is to minimize the weighted error:

$$J = e^H We \ , \text{ where } e_k = \left(z_k^n + a_{n-1}z_k^{n-1}\cdots + a_0\right)G_i - \left(b_m z_k^m + \cdots + b_0\right)$$

The error is complex, and the superscript H is the complex-conjugate transpose. The error is the frequency response error of just the numerator, a quantity that is not usually important, if at all, but this error has the distinct advantage of being linear in the coefficients. To solve the problem define F and y in terms from the data, and the linear least square solution takes to form:

$$\theta = \begin{pmatrix} \text{Re}[WF] \\ \text{Im}[WF] \end{pmatrix}^+ \begin{pmatrix} \text{Re}[Wy] \\ \text{Im}[Wy] \end{pmatrix}$$

Several variations of the frequency domain parameter estimation have been implemented in the wavelet analysis toolbox. These include:

- An iterative version of the problem is defined whereby weight depends on the previous estimate of the coefficients and is used to define a relative error:

$$W = \text{diag}(w_i) \ , \text{ where } w_i = \left|b_m z_i^m + \cdots + b_0\right|^{-2}$$

- The relative error gives equal weight across the frequency range, and in particular does a better job at estimating the transfer function in the frequency range where the gain is low.
- To directly minimize the frequency response use the error:

$$J = e^H We \ , \text{ where } e_k = G_k - \left(b_m z_k^m + \cdots + b_0\right) / \left(z_k^n + a_{n-1}z_k^{n-1}\cdots + a_0\right)$$

- The error now depends nonlinearly on the parameters, and an iterative solution is required. Use the linear solution, or even better the iterative linear solution as the

initial guess. The diagonal weight with $w_i = |G_i|^{-2}$ turns the error into a relative error.

- The coefficients of a Laplace transform can be directly estimated, where ω_i and G_i are respectively the analog frequencies and the frequency response. The large dynamic range of the parameters can cause numerical stability, and this problem is alleviated by using normalized frequencies, ω_i / ω_{max}, and by converting the numerator and denominator polynomials to orthogonal polynomials.

- A delay can be included in the problem, z^{-d} in the digital world and $e^{-s\tau}$ in the analog world. The delays can be estimated iteratively and/or as part of the nonlinear problem.

4.8.4 Markov Parameter Methods

The Markov parameters for digital systems are the digital impulse response. The basic idea is to estimate the Markov parameters from the input and output time series, and then use the parameters either to determine the transfer function coefficients, or via the Eigensystem Realization Algorithm (ERA) determine a state space system. Advantages of the Markov parameter methods are that they can be used to determine the system order, they directly estimate state space systems, and they have been shown (by us and others) to work well with high order (50 and above) systems.

The difficult part of this method is finding the Markov parameters. The parameters are sensitive to the initial conditions, to noise, and to offsets in the data. One can even say, very sensitive. One approach to reducing this sensitivity (Ref. 24) is to use the time domain parameter estimation problem as a first step, from which the Markov parameters are determined. Another way to reduce sensitivity (Ref. 25) is to start first in the frequency domain, and to use the transfer function coefficients so estimated to determine the Markov parameters.

As part of this study we have implemented and further developed a method whereby wavelet transforms are used as an intermediate step in finding the Markov parameters (Ref. 26). The wavelet transforms of the input and output time series are used to estimate the Markov parameters in the transform domain. Selective use of the transform coefficients reduces sensitivity to noise and to offsets. Estimating the initial conditions and then subtracting off the forced response reduces sensitivity to non-zero initial conditions. The original Markov parameters are recovered using the inverse wavelet transform. The combined method is called the Wavelet Eigensystem Realization Algorithm (WERA). The Markov parameter estimation methods and WERA are further discussed in Section 5 and Appendix B.

4.8.5 Summary

We have used frequency domain and WERA parameter estimate methods extensively in this study. Many examples are presented in Section 6.

The number of variations in the parameter estimation methods quickly becomes a burden to the user and automation is needed. The variations of the frequency domain estimation method that have been most productive, and automated, are briefly summarized here. Laplace transform coefficients are estimated including a parameter for delay. The linear least square solution is used as the initial guess, followed by several iterations where the relative error is minimized, followed by several iterations with different delays used to obtain a starting guess of delay, and finished with nonlinear minimization.

Any method works well for low order problems using data generated by simulation, but the combined ill-effects of "real world" data makes single-step linear least square methods unreliable. The alternative is to "get close," and finish with nonlinear minimization. Even better, and this option can be switched on if

computation time is not an issue, is to use many starting guess, with different order systems, and other variations, and then choose the global minimum.

4.9 Analysis Example (F-15 Active Model)

Time varying transfer functions (TVTFs) are demonstrated using a model of the F-15 Active aircraft. A change is made to the longitudinal dynamics by suddenly inserting an extra delay. The exact frequency responses are known before and after this change, and which are used to compare the estimated frequency responses using wavelet methods. A model of a pitch SCAS is shown in Figure 38. The SCAS is needed because the open loop vehicle is unstable. The input δ_e is the effective elevator input made by combining stabilator, canard, and thrust vector inputs. Ideal actuators and sensors are used, and the aircraft dynamics use a short period approximation. Open and closed loop transfer functions are shown in the Figure 38. The delay is modeled using a second order Padé approximation. Of particular note is that the 120 msec delay results in a lightly damped closed loop poles at 10.2 rad/sec with a damping ratio of 0.0868.

Simulation results are shown in Figure 39. The closed loop input is the output of a bandpass filter of 0.1 to 20 rad/sec driven by unit intensity white noise. The run length is 60 seconds with 3 seconds of zero input at the start and end. The sample period in the simulation is 10 msec. The presence of the lightly damped mode is apparent in the pitch response in the second half of the simulation.

Given the time series the objectives are to estimate the time varying frequency response and to detect the change in delay. The TVTF is shown different ways in Figure 40:

Part a) contains the exact frequency responses, used to verify the estimates. Notable features are the peaked magnitude response and the resulting sudden drop-off of phase. The time for the magnitude peak to cross the 0 dB line is used as the detection time.

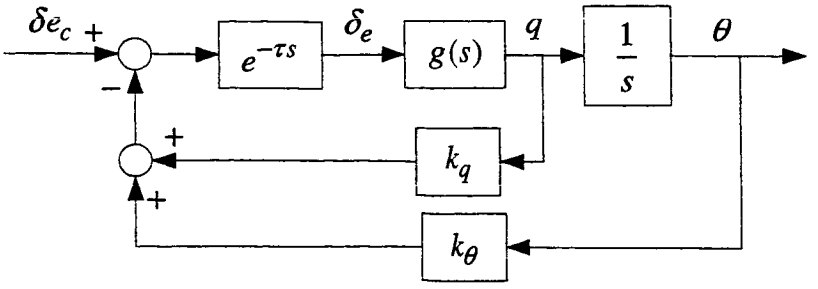
Part b) is a 2D plot of the TVTF, showing 10 seconds of data ending at 37 seconds. The last frequency response is the bold dotted line, and the previous frequency responses fade to white. The peak in the magnitude is seen to build up, and the phase changes from a gradual to a sudden increase. The coherence drops off when the change in delay occurs.

Part c) is a 3D plot of the same TVTF. The magnitude and phase changes are visually apparent, but numerical results are better obtained from the 2D slices of this figure.

How long does it take to detect the change in delay? The magnitude peak crosses the 0 dB line in 7 seconds. The detection time depends on how the TVTF is computed, and different methods and parameters are shown in Figure 41 and summarized in Table 3. The left hand column in the figure is the 3D display of TVTFs looking almost straight down the frequency axis, a view that highlights the magnitude change. The right hand column is a single frequency versus time, from which the detection times are obtained, varying from 5 to 15 seconds. Part a) of the figure uses the same TVTF parameters as the first part of the example. The detection time is increased in Part b) by using fewer cycles of the Erlang wavelet and by averaging fewer adjacent times. The variance of the frequency response estimate increases accordingly. Parts c) and d) use finite time Fourier transforms with sliding windows, with window lengths of 60 and 5 seconds. The smaller window matches the 5 second detection time.

Table 3. F-15 Active Example Detection Times

Method	Parameters	Detection Time
Erlang wavelet	5 cycles, 21 adjacent times averaged	7 secs
Erlang wavelet	12 cycles, 120 frequencies log spaced from 0.1 to 30 rad/sec, 3 adjacent freqs averaged, 300 transforms one each 0.2 secs, 41 adjacent times averaged	5 secs
Finite Time Fourier Transform	Frequency smoothing with adjacent frequency ratio of 1.05 and at least 3 bins averaged, 41 adjacent times averaged, 60 second sliding window	15 secs
Finite Time Fourier Transform	21 adjacent times averaged, 5 second sliding window	5 secs



Open Loop:

$$\frac{q}{\delta_e} = \frac{\text{-}21.9(0.993)}{(\text{-}1.8)(3.53)}$$

$$K_q = -0.435$$

$$K_\theta = -0.435 \times 3.5$$

Closed Loop:

$$\frac{\theta}{\delta_{e_c}} = \frac{\text{-}21.9(0.993)}{(1.55)(3.39)(6.32)} \quad \text{with} \quad \tau = 0$$

$$\frac{\theta}{\delta_{e_c}} = \frac{\text{-}21.9(0.993)[0.866, 28.9]}{(1.4)(3.46)[0.0868, 10.2](54.6)} \quad \text{with} \quad \tau = 0.12$$

Figure 38. F-15 Active Pitch SCAS

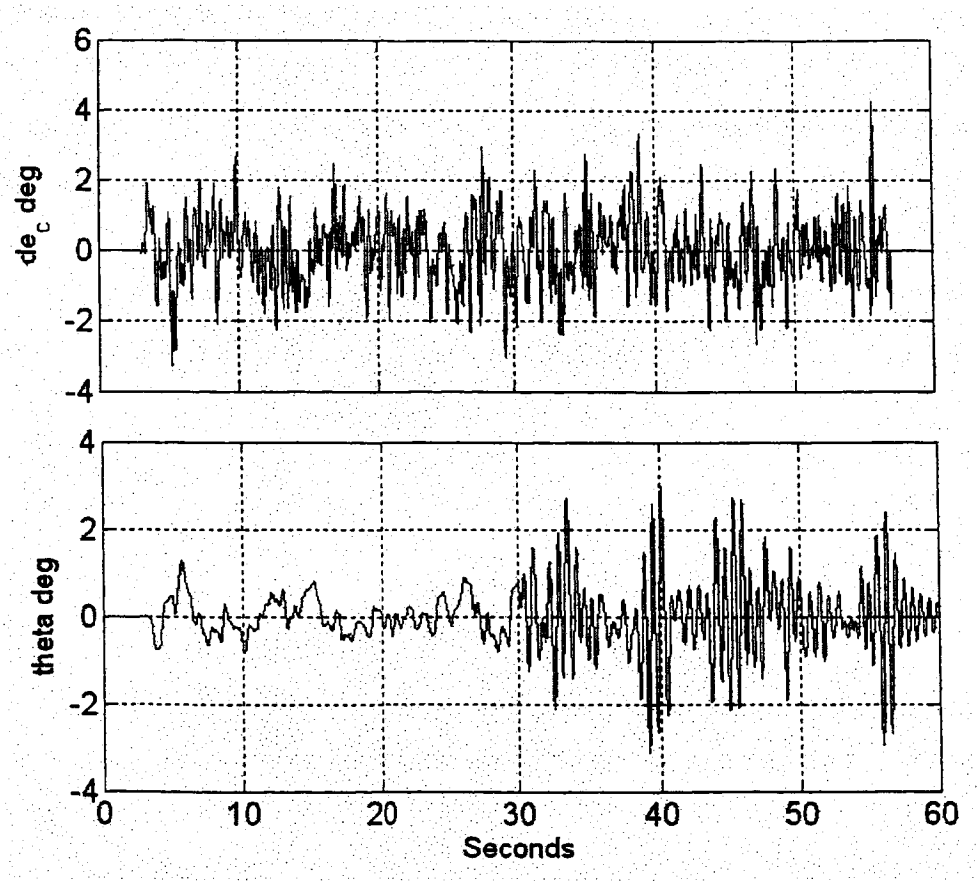
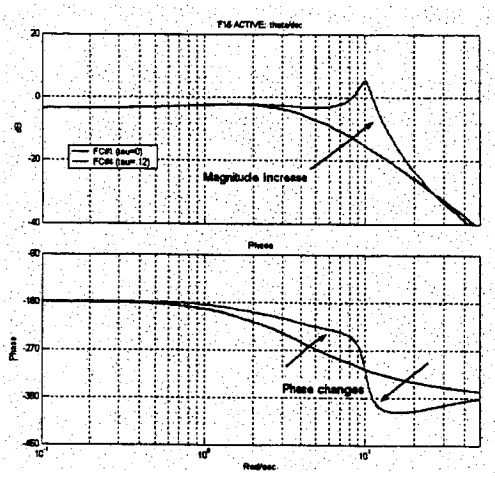
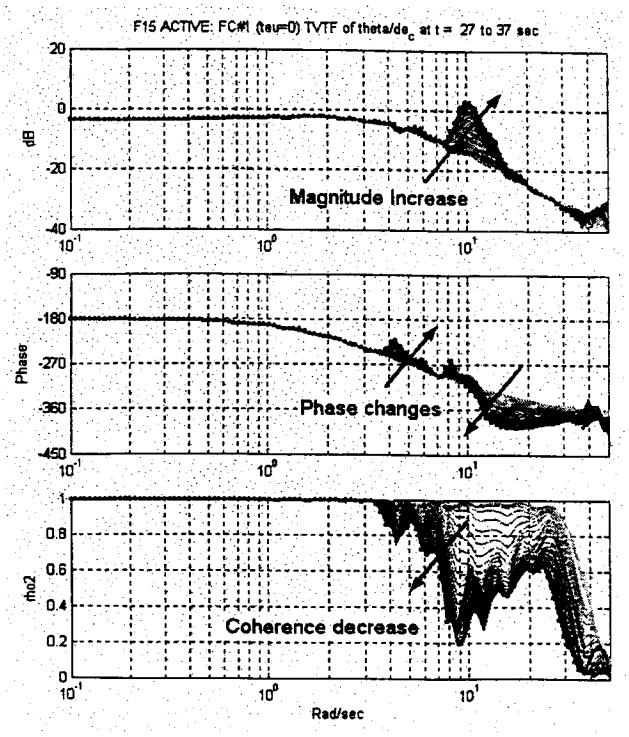


Figure 39. F-15 Active Simulation (the delay changes from $\tau = 0$ to $\tau = 0.12$ at 30 seconds)

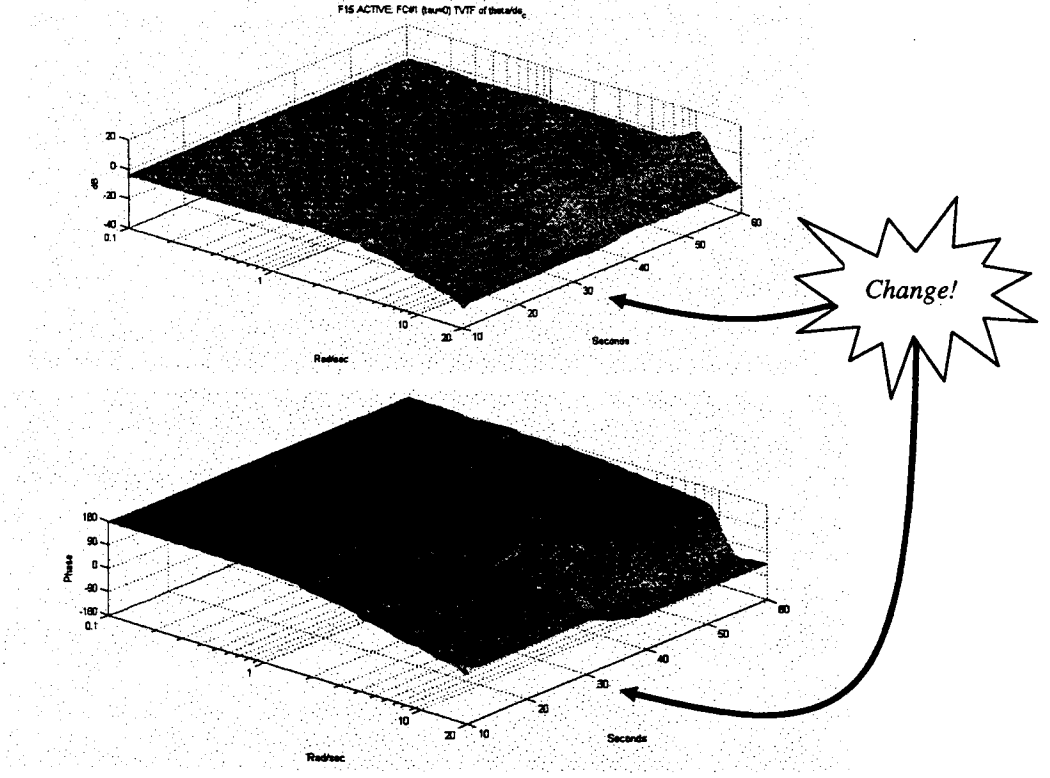


a) Exact frequency responses before and after change at 30 seconds of τ from 0 to 0.12 seconds



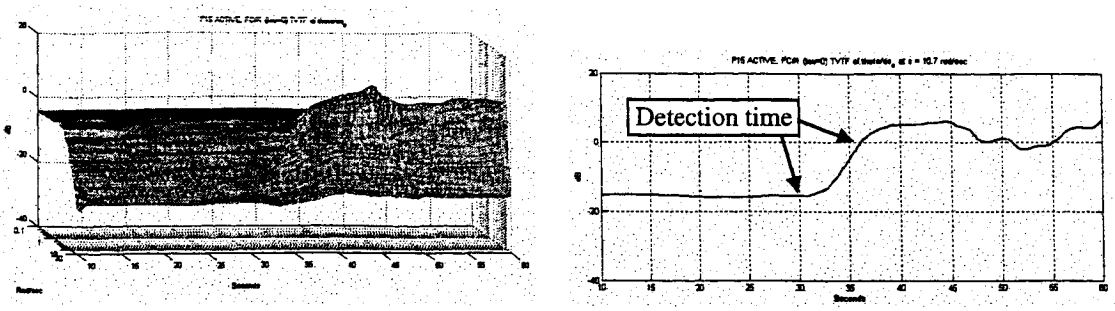
b)

TVTF (2D display with persistence)

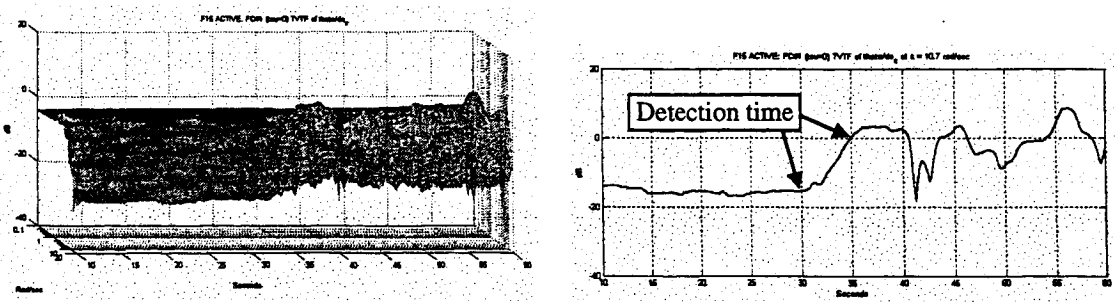


c) TVTF (3D Display of Theta/dec)

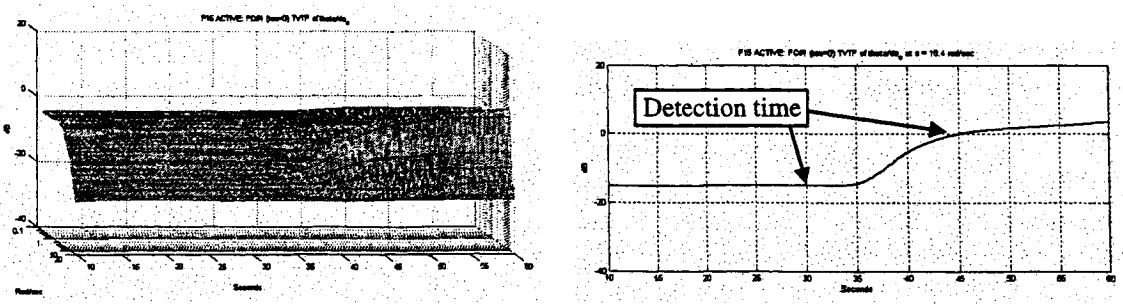
Figure 40. TVTF Example (F-15 Active with sudden additional delay)



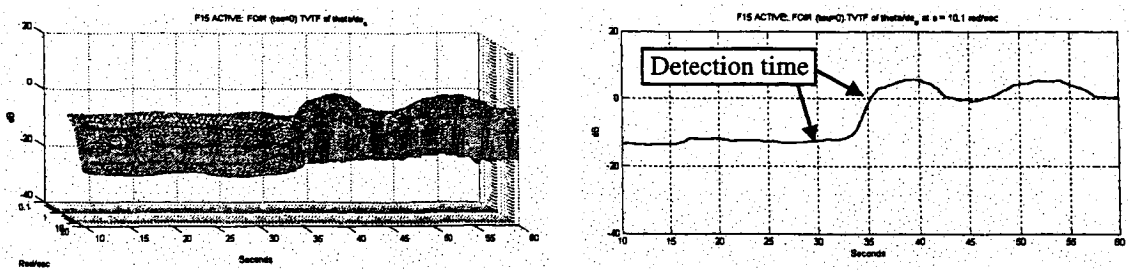
a) Moderate detection (Erlang 12 cycle wavelet, 41 time bins averaged, 7 seconds to 0 dB line)



b) Fast detection (Erlang 5 cycle wavelet, 21 time bins averaged, 5 seconds to cross 0 dB line)



c) Slow Detection (Finite time FFT, 60 second sliding window, 21 time bins averaged, 15 seconds to 0 dB line)



d) Fast Detection (Finite time FFT, 5 second sliding window, 21 bins averaged, 5 seconds to 0 dB line)

Left plots are 3D Theta/dec magnitude plots; right plots are 2D displays of a single frequency versus time

Figure 41. Detection Time Example (F-15 Active with sudden additional delay)

4.10 Analysis Example (Large Transport Fixed Base Simulator Data)

TVTF estimation, transfer function coefficient estimation, and stability metric estimation are all demonstrated in this example. The data is from a hardware-in-the-loop manned flight simulator for a large military transport, Run 69 from the LOCATS database. The experiment used to collect this data is described in Volume II of this report.

The fixed base, manned simulator is represented in the Figure 42 block diagram. The pilot inputs are DLON and DLAT, the longitudinal and lateral stick positions in inches. There is a turbulence input, turned on for this run, and there is also an auto-throttle loop turned on for this run. The auto-throttle loop is an internal feedback inside the main block. Here the interest is the closed loop response from DLON to THE, the longitudinal stick to pitch attitude. The simulator is 6 DOF nonlinear, with a complete aerodynamic model and nonlinear actuator models, the actual flight control computers, and a representative stick. The stick has a deadband, friction, and preload that is prevalent primarily in the lateral axis (see Volume II of this report). We have not tried to extract a linear model from the simulator documents, and so we do not have a "truth" model for this example. The flight condition for this run is lightweight cruise.

Time series data from this run is shown in Figure 43. The pilot is performing a pitch attitude capture and hold (PACH) task. At 27 seconds into the run the pitch SCAS is turned off, changing the controlled element from a well damped, closed loop vehicle to a lower bandwidth, more lightly damped, open loop vehicle. In addition, throughout the run extra delay is included in the pilot input, in order to make the piloting task more difficult.

The TVTF is computed using the Matlab Wavelet Toolbox developed by us as part of this study. The wavelet is the Erlang wavelet with 5 cycles, computed every 0.3 seconds, with 200 frequencies log-spaced from 0.1 to 20 rad/sec. Smoothing is performed on 41 adjacent times and 11 adjacent frequencies. The TVTF is shown in Figure 44, which has the magnitude, phase, and coherence. The change in the TVTF at 27 seconds is visually not very apparent, making this a harder case to analyze. The change is easier to see on a color display, where the color changes show that the open loop vehicle has a more rapid roll-off in both magnitude and phase. Another "feature" is the ridges in phase above 10 rad/sec, indicating a large variance in the phase estimate, and due mainly to the lack of input power above 10 rad/sec.

The example continues with the estimation of three stability metrics:

ω_{phase} = phase bandwidth, the frequency at which the phase crosses –135 degrees. A pure gain pilot closing the loop at ω_{phase} will result in 45 degrees of phase margin.

ω_{gain} = gain bandwidth, the frequency at which the magnitude is 6 dB above the magnitude at ω_{180}, where ω_{180} is the frequency where the phase crosses –180 degrees. A pure gain pilot closing the loop at ω_{gain} will result in 6 dB of gain margin.

τ_p = phase delay, defined as one-half the slope of the phase between the ω_{180} and $2\omega_{180}$ frequencies. Large phase delays indicate a rapid phase roll-off, which in turn results in a higher pilot workload. The extra phase roll-off is due to pure delay, higher order dynamics that act like an effective delay, and nonlinearities such as rate limits.

Each of these metrics can be estimated from a frequency response by searching for magnitude and phase crossovers, and the toolbox contains options to do just that. The advantage of using the "raw" frequency response is that a parametric model of the controlled element is not needed. The disadvantage is that the variance in the frequency response estimate can result in large errors in the estimate of the metrics, in other words, the stability metrics are more sensitive than the frequency response. This is particularly a problem for the gain bandwidth and the phase delay, because they are based on crossover frequencies

above 4 rad/sec, and therefore up the in frequency range where the input does not have much power and the coherence is low.

To reduce the sensitivity of the stability metrics, the raw frequency responses are used to estimate coefficients of a Laplace transform, and then the stability metrics are estimated from the frequency responses of this fitted Laplace transform. This process is demonstrated in parts a) and b) of Figure 45; where the raw "TVTF" and "Fitted" frequency responses are plotted before and after the change in the pitch SCAS. The frequency response of the fitted Laplace transform is seen to smoothly cut through the bobbles in the raw frequency response.

The stability metrics are computed for each time slice of the TVTF and plotted versus time in part c) of Figure 45. A significant reduction in the phase bandwidth occurs after the pitch SCAS is turned off at 27 seconds. We have not automated detection times, but a plausible detection time of 13 seconds for the phase bandwidth change is overlaid on the plot. The gain bandwidth and phase delay each have significant transient changes when the pitch SCAS is turned off, but then settle back to their original values.

How good are the estimated stability metrics? This is a difficult question. We do not have "truth" models available to us, but one way to check Figure 45c is to compare the estimates with values obtained from longer sections of the data. The stability metrics estimated this way are listed in Table 4, using 25 seconds of data before the change, and 60 seconds after. There is a good match with the time varying metrics in Figure 45c.

Table 4. Estimated TF and Stability Metrics for LOCATS Run 69

	With Pitch SCAS (0 to 25 seconds)	Without Pitch SCAS (30 to 90 seconds)	
Estimated tf	$\frac{\theta}{\delta_{e_c}} = \frac{-5.64(-11.3)e^{-0.198s}}{(1.06)[0.536,5.54]}$	$\frac{\theta}{\delta_{e_c}} = \frac{-2.04(-7.52)e^{-0.138s}}{[0.614,1.1](4.92)}$	
ω_{phase} [rad/sec]	2.4	1.2	⇐ Significant change
ω_{gain} [rad/sec]	1.3	1.2	
ω_{180} [rad/sec]	3.5	1.8	⇐ Significant change
τ_p [sec]	0.32	0.33	

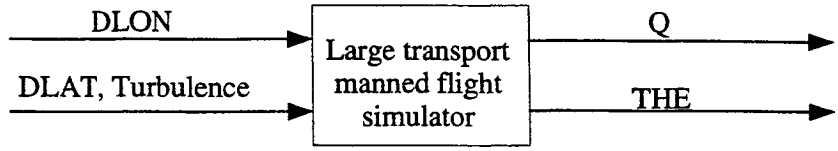


Figure 42. Large Transport Fixed Base Flight Simulator

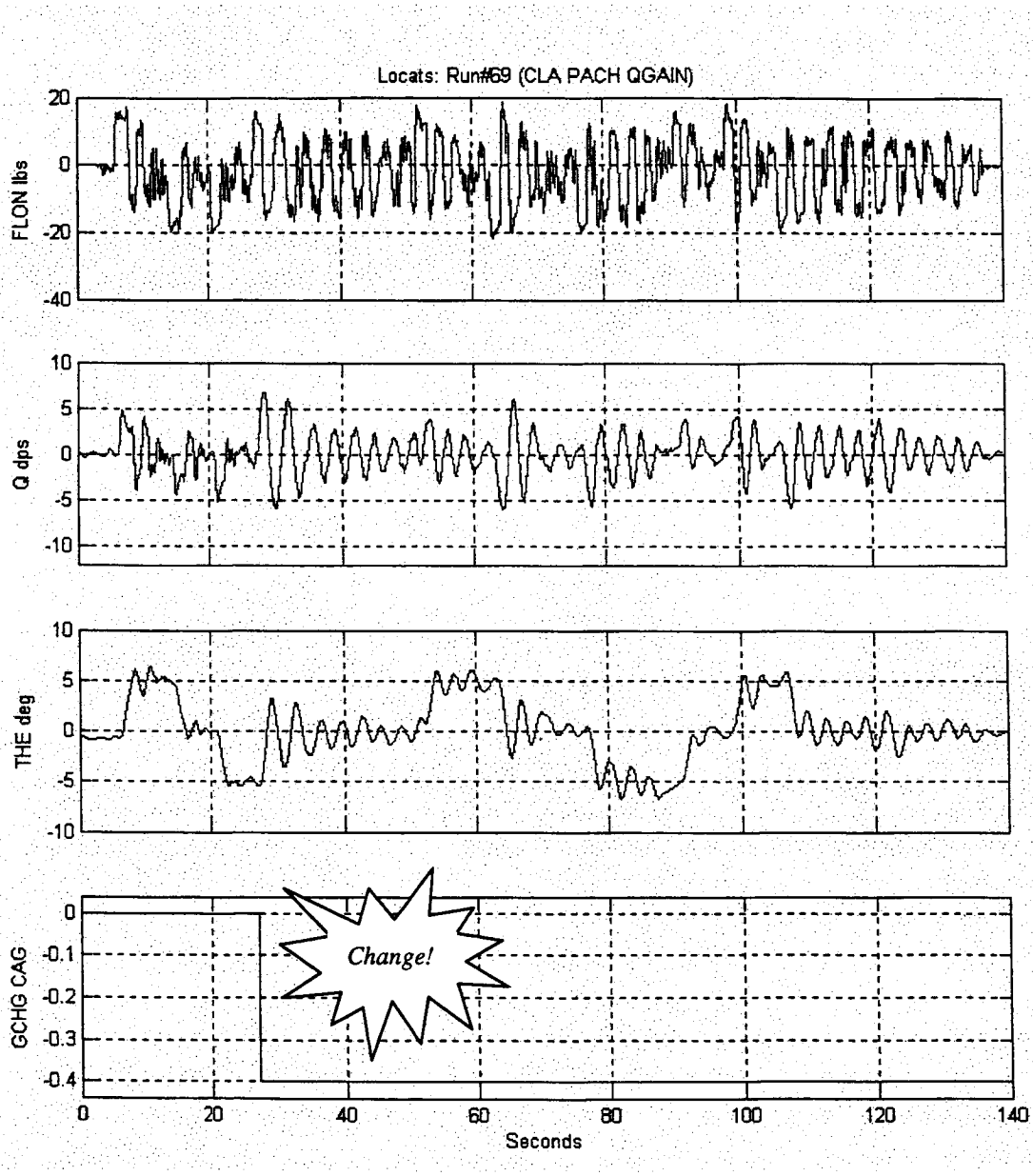


Figure 43. Time Series from LOCATS Run 69 Pitch Tracking Task

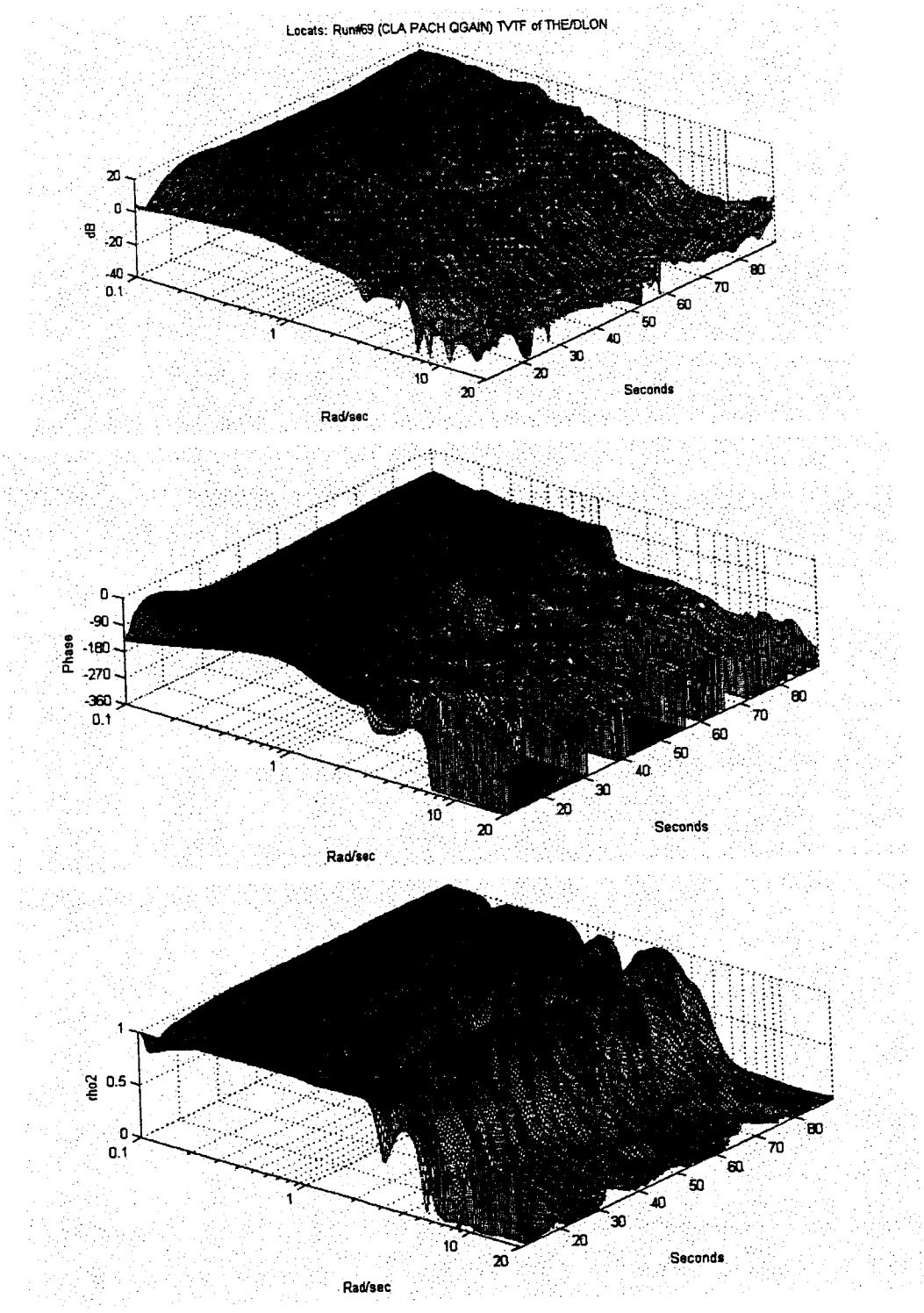
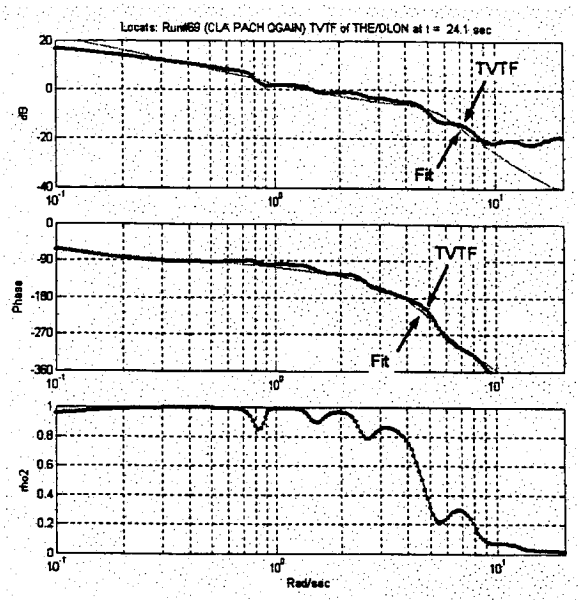
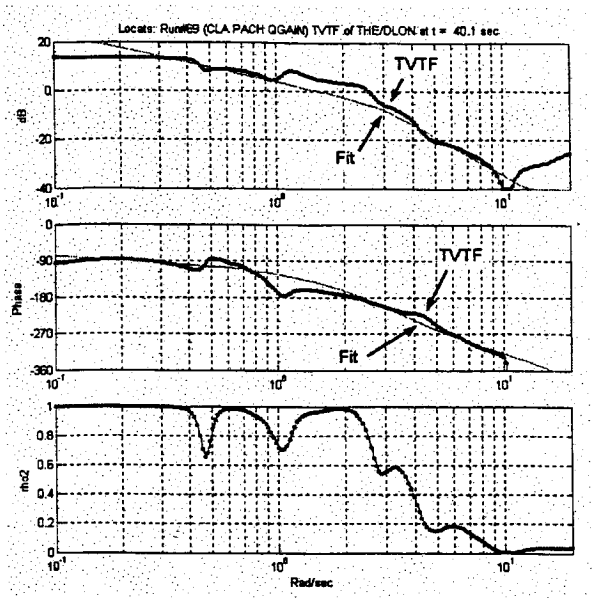


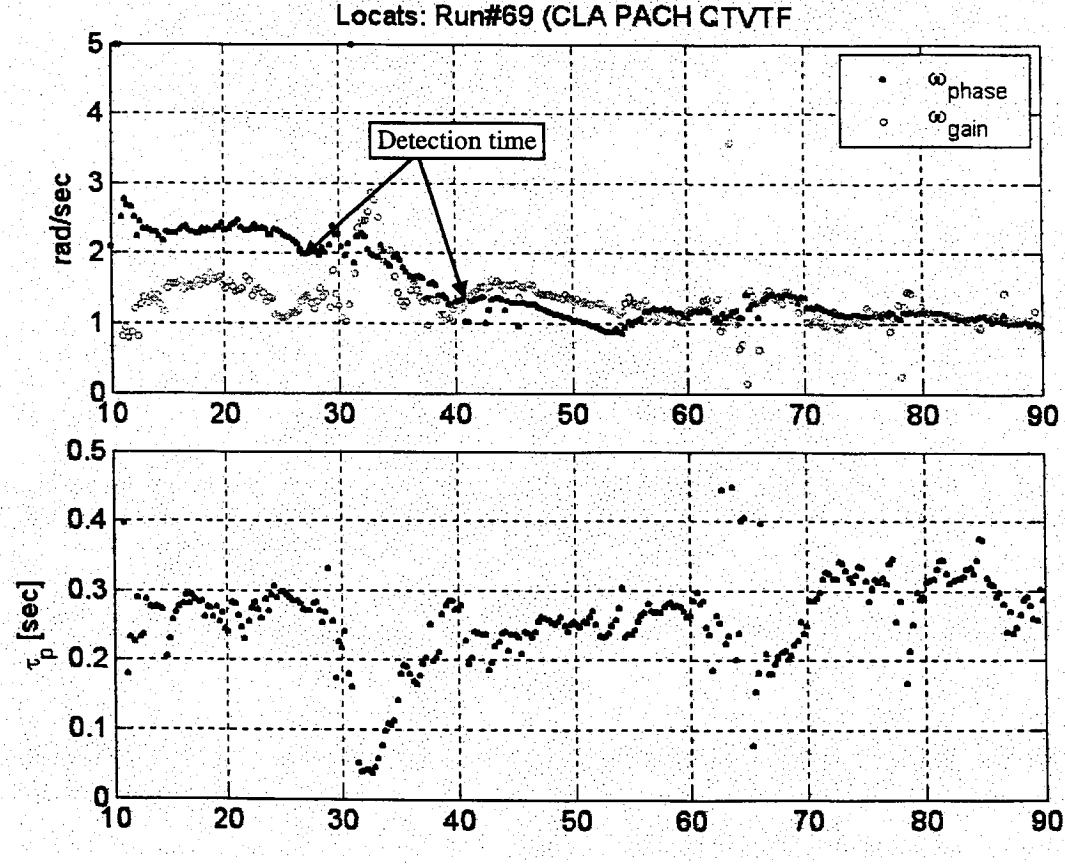
Figure 44. 3D TVTF from LOCATS Run 69



a) TF fit before SCAS change



b) TF fit after SCAS change



c) Phase and Gain Bandwidths and TauP Metrics

Figure 45. Stability Metric Estimation from LOCATS Run 69

5. ERA-BASED SYSTEM IDENTIFICATION USING WAVELETS

5.1 Introduction

The experimental identification of physical vibration modes and mode shapes can be done with impulse response functions that are extracted from measured vibration records. In order to extract these impulse response functions, the Fast Fourier Transform (FFT) has often been used in conjunction with repeated data filtering and windowing (Ref. 27). These techniques require input signals that are rich enough to excite frequencies of interest – this does not appear to be the case with wavelet analysis. Moreover, with its more intuitive decomposition of data, wavelet analysis allows identification of time-varying system parameters.

The system realization, based on a first-order state space model, can be represented as:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y = Cx(t) + Du(t)$$

In the time domain, the solution at time t_n to the above equation can be expressed as:

$$y(t_n) = \int_{-\infty}^{n} h(t_n - \tau)u(\tau)d\tau$$

where $h(t)$ is the temporal impulse response function. This convolution formula can be expressed in matrix form by:

$$Y = hU$$

In this equation, Y represents the output matrix, U is the input matrix, and h is the time-domain impulse response matrix. Once h is known the output of our system can be determined for any arbitrary input. An accurate extraction of $h(t)$, often referred as the Markov parameters, will identify a system (Ref. 28) since

$$h(t) = Ce^{At}B$$

One could try to obtain $h(t)$ directly by applying a unit impulse input $u(t)$. In this case, $y(t)$ would theoretically be identical to the desired $h(t)$. However, this is practically infeasible. Looking at the equation $Y = hU$, if the input signals are not rich in frequency content, or the sampling size is too large, U can become ill-conditioned and the corresponding impulse response function cannot be computed accurately. Moreover, the computations associated with $h(t)$ can be intensive. For these reasons, FFT-based extraction of the impulse response function is widely used (Ref. 27) since it has high computational efficiency and reasonable robustness provided the input data is rich in frequency content. However, if the input load u consists of only a single or a few frequencies, the temporal impulse response data tends to be erratic and badly conditioned. Wavelet analysis appears to offer a more robust alternative.

5.2 Computational Technique

5.2.1 Wavelet Theory

The wavelet transform allows any arbitrary signal $f(x)$ to be decomposed into an infinite summation of wavelets at different scales according to the expansion (Ref. 29):

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} W(2^j x - k)$$

The $W(2^j x-k)$ functions are the wavelets and provide a local basis along the time axis. To see how wavelets can be generated from the so-called dilation equation and their relation to the scaling functions

$\varphi(x)$, the reader is directed to Ref. 22. Because of the way in which the wavelets are defined, when j is negative, the wavelets $W(2^j x-k)$ can be expressed as a sum of terms like $\varphi(x-k)$. The corresponding wavelet transform can then be expressed as follows:

$$f(x)=\sum_{k=-\infty}^{\infty} c_{\varphi,k}\varphi(x-k)+\sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} c_{j,k}W(2^j x-k)$$

In this report, the Daubechies coefficients (Refs. 30 and 31) are used to generate these wavelets. Figure 46 shows a family of Daubechies' wavelets. Note that *db2*, which is used in this analysis, spans a bit less than 3 units over the x-axis. The Daubechies' family of wavelets satisfies two crucial requirements: 1) orthogonality of local basis functions, and 2) second or higher-order accuracy, depending on the dilation expression adopted (Ref. 32).

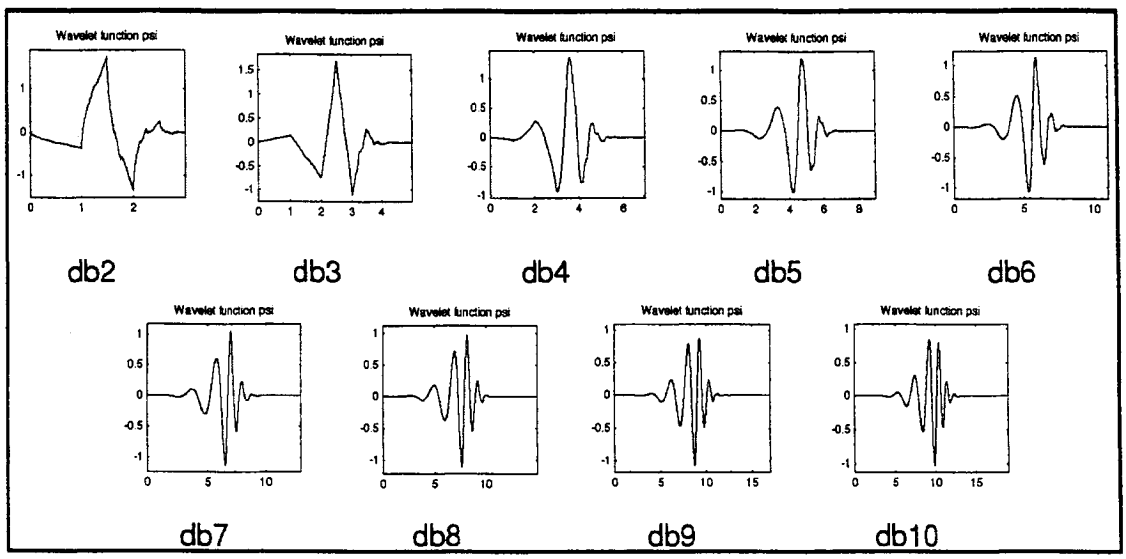


Figure 46. Daubechies Family of Wavelets

In order to use the wavelet transformation in practical applications, a way to define a discrete version of the transformation must be found. For this purpose, the range of the independent variable x is limited to the unit interval [0,1), and a wavelet series expansion is performed over that interval (Ref. 33). A complication arises since some of the wavelets $W(2^j x-k)$ overlap the edges of the interval. For this reason, it is convenient to assume that $f(x)$ is one period of a periodic signal exactly repeated in the adjacent unit intervals. Note that with the D4 wavelet family $W(x)$ spans almost three units. Over the interval [0,1) there are contributions from three bases: from the first third of $W(x)$, from the middle third of $W(x+1)$, and from the last third of $W(x+2)$. This is equivalent to $W(x)$ being wrapped around the unit interval. Thus the wavelet expansion of $f(x)$ in the [0,1) interval can be written as:

$$f(x)=a_0\varphi(x)+a_1W(x)+\begin{bmatrix}a_2 & a_3\end{bmatrix}\begin{bmatrix}W(2x)\\W(2x-1)\end{bmatrix}+\begin{bmatrix}a_4 & a_5 & a_6 & a_7\end{bmatrix}\begin{bmatrix}W(4x)\\W(4x-1)\\W(4x-2)\\W(4x-3)\end{bmatrix}+\ldots$$

$$+a_{2^j+k}W(2^j x-k)+\ldots$$

The coefficients a_1, a_2, a_3, a_4, ... give the amplitudes of the contributing wavelets (after wrapping) to one cycle of the periodic function in the [0,1) interval. Because of orthogonality conditions, the general wavelet transform coefficients can be found by

$$a_{2^j+k} = 2^j \int_0^1 f(x)W(2^j x - k)dx, \qquad a_0 = \int_0^1 f(x)\varphi(x)dx$$

The discrete wavelet transform (DWT) is an algorithm for computing the above equations when $f(x)$ is sampled at equally spaced intervals over [0,1). As stated before, it is assumed here that $f(x)$ is one period of a periodic signal and that the scaling and wavelet functions wrap around the interval [0,1). A remarkable feature of the DWT algorithm is that there is no need to compute $\varphi(x)$ or $W(2^j x-k)$ explicitly. An Matlab-based implementation of this DWT algorithm, found in Appendix 7 of Ref. 33, was developed by Mallat, and is often referred by as the *Mallat's Pyramid Algorithm.*

5.2.2 DWT-Based Extraction of the Markov Parameters

The DWT method starts with the convolution integral,

$$y(t_n) = \int_{-\infty}^{t_n} h(t_n - \tau)u(\tau)d\tau = \int_0^1 h(\theta)u(t_n - \theta)d\theta$$

Note that $h(\theta)$ is the temporal impulse response function. The impulse response function is expanded in terms of the wavelet basis functions for the entire time interval, $0 \le \theta < 1$;

$$h(\theta) = h_0^{DWT} + \sum_j \sum_k h_{2^j+k}^{DWT} W(2^j\theta + k)$$

where the h^{DWT} terms are the expansion coefficients. For the DWT characterization of $u(t_n-\theta)$, first $u(\theta)$ is reversed in time to obtain $u(-\theta)$, then it is shifted toward the positive time axis by an amount t_n. Following this, $u(t_n-\theta)$ can be expressed as

$$u(tn - \theta) = u_0^{DWT} + \sum_i \sum_j u_{2^j+k}^{DWT} W(2^j\theta + k)$$

Substituting and making use of the orthogonality conditions, yields the following (Ref. 33):

$$y(t_n) = h^{DWT} u^{DWT}(t_n)$$

For the entire data sample, the inputs and outputs are arranged in the form

$$Y = h^{DWT} U^{DWT}$$

Solving for h^{DWT} (Ref. 32),

$$h^{DWT} = Y U^{DWT^T} (U^{DWT} U^{DWT^T})^{-1}$$

The desired temporal impulse response data is finally obtained by taking the inverse wavelet transform of h^{DWT}. (See Ref. 33 for details on how to implement the inverse wavelet transform.)

$$h(t) = IDWT(h^{DWT})$$

It is important to note that, in the h^{DWT} equation, the term $U^{DWT}(U^{DWT})^T$ is invertible. This is because the wavelet transform coefficients consist of a set of orthogonal local basis functions. This represents a significant improvement over direct time-domain methods – they find the Markov parameters by $h=U^{-1}Y$. Unfortunately U is often ill-conditioned; and one is forced to take its pseudo-inverse instead. With FFT-based methods, an ill-conditioned U matrix also negatively affects the low-mode or static deformation

amplitudes and phases. The proposed wavelet transform does not induce any rank-deficient basis function.

5.2.3 Response Decoupling

The limits of the convolution integral, $y(t_n) = \int_{-\infty}^{t_n} h(t_n-\tau)u(\tau)d\tau$, can be changed to $y(t_n) = \int_{0}^{t_n} h(t_n-\tau)u(\tau)d\tau$ if $h(t)$ is defined to be zero for time less than zero. This then represents the forced response of a system to an input $u(t)$, where all initial conditions (at $t_0 = 0$) are zero. When a data sample is taken after system start-up at $t_0 > 0$, the convolution integral is generally no longer valid, as the sample usually introduces non-zero initial conditions $y(t_0), \dot{y}(t_0)$, etc. Thus the general response consists of a forced component and free component,

$$y(t_n) = y_{forced}(t_n) + y_{free}(t_n)$$

Using a state-space representation, the free response can be written as

$$y_{free}(t_n) = Ce^{(t_n-t_0)A}x_{t_0}$$

The convolution integral can then be written as

$$y_{forced}(t_n) = \int_{0}^{t_n} h(t_n-\tau)u(\tau)d\tau = h(t_{t_n}) * u(t_{t_n})$$

The above equation can be rearranged so that

$$h(t_{t_n}) * u(t_{t_n}) = y(t_n) - Ce^{(t_n-t_0)A}x_{t_0}$$

Discretizing the free response with a sampling period of T (Ref. 34),

$$e^{TA} = I + \frac{T}{1!}A + \frac{T^2}{2!}A^2 + \ldots$$

For a sufficiently small T, only two or three terms need to be carried out and the free response equation can then be written using the difference equation

$$x_0(k+1) = e^{TA}x_0(k),$$

$$y_{free}(k+1) = Ce^{TA}x_0(k)$$

The discrete wavelet transform of the impulse response function can then be written as

$$h^{DWT} = y_{forced}U^{DWT^T}(U^{DWT}U^{DWT^T})^{-1},$$

where

$$y_{forced}(k) = y_{obs}(k) - Ce^{TA}x_0(k)$$

Finally, the impulse response function in the time-domain is computed by taking the inverse discrete wave transform of the above equation,

$$h(t) = IDWT\{y_{forced}U^{DWT^T}(U^{DWT}U^{DWT^T})^{-1}\}$$

Computation of the system parameters is done with two methods that make use of the observed impulse response $h(t)$. The first method employs the Eigensystem Realization Algorithm (ERA) method (discussed in Section 5.5) that uses least squares to estimate the discrete state space realization of the observed impulse response. The second method takes the Fourier transform of $h(t)$,

$$H(\omega) = \int_0^{t_n - t_0} h(t) e^{-i\omega t} dt$$

$$\Downarrow$$

$$H(\omega) = \sum_{k=0}^{k=N} h(kT) e^{-i\omega kT} T$$

Looking at the forced response equation, the unknowns are A, C, and x_0. Any single-input, single-output system that has a state-space model has a rational, proper transfer function of the form

$$\frac{y(s)}{u(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots b_n}{s^n + a_1 s^{n-1} + \cdots a_n}$$

If $b_0 \neq 0$, it is convenient to divide the denominator into the numerator to rewrite as

$$\frac{y(s)}{u(s)} = b_0 + \frac{c_1 s^{n-1} + \cdots c_n}{s^n + a_1 s^{n-1} + \cdots a_n},$$

where

$$c_i = b_i - b_0 a_i, \quad i = 1, \cdots n$$

In order for the states x to be observable from the output y, the state-space equations are written in *observable form*, i.e.

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & & & -a_n \\ 1 & & & -a_{n-1} \\ & \ddots & & \vdots \\ & & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + b_0 u$$

The identification process is initiated by assuming initial values for the poles and zeros of the system's transfer function, which in turn determines the initial state-space representation. From the above equation $y = x_n + b_0 u$, and the states $x_{n-1}, x_{n-2}, \cdots, x_1$ can be solved from the $(n\text{-}1)$ equations available from the A and B matrices. Thus the initial states x_0 are estimated based on the initial condition of y (i.e., the first value of each data sample) and its higher derivatives. This will allow an estimate of the free response to be subtracted from the observed response, and the resulting impulse response function is compared with the impulse function arising from the estimated A and B matrices. This comparison is conveniently done in the frequency domain, where a least-squares fit of the observed impulse response to poles and zeros is performed.

5.2.4 Example Results

A simulation was conducted using a sum-of-sines input forcing function (7 frequencies, 10 rad/sec bandwidth), where the magnitude of the sine wave amplitude tapered with increasing frequency. The input/output transfer function to be identified was

$$\frac{y(s)}{u(s)} = \frac{(s+.5)}{(s+3)(s+5)} = \frac{s+.5}{s^2+8s+15}$$

Data was sampled at 100 Hz, and a time window of 2.56 seconds was used. A D4 Debauchies wavelet with 7 levels (2^7 by 2^7 input matrix U^{DWT}) was also employed. Figure 47 compares the observed and simulated impulse response functions, where the initial estimate of the system poles and zeros matched the simulated values. The data sample was taken 20 seconds into the simulation run.

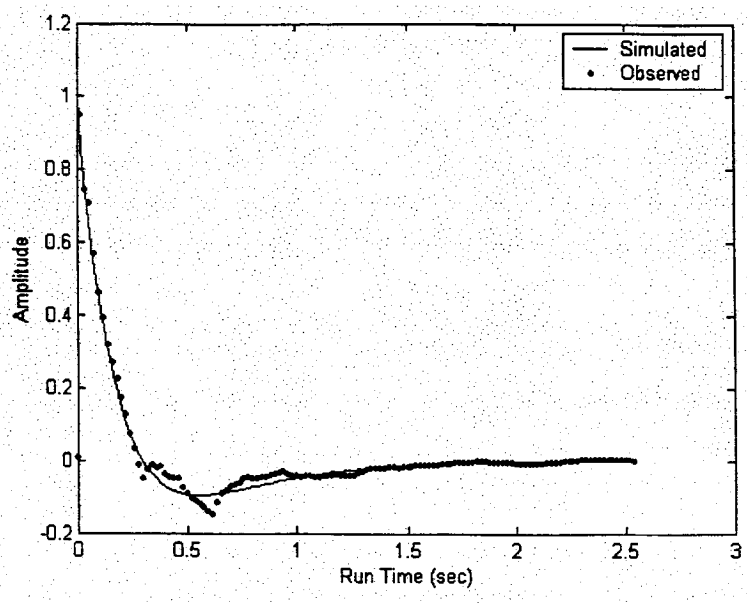


Figure 47. Comparison between Observed and Simulated Impulse Response Function

In Figure 48 the frequency response of the two impulse functions is shown. The observed computation is a near-match with the simulated response.

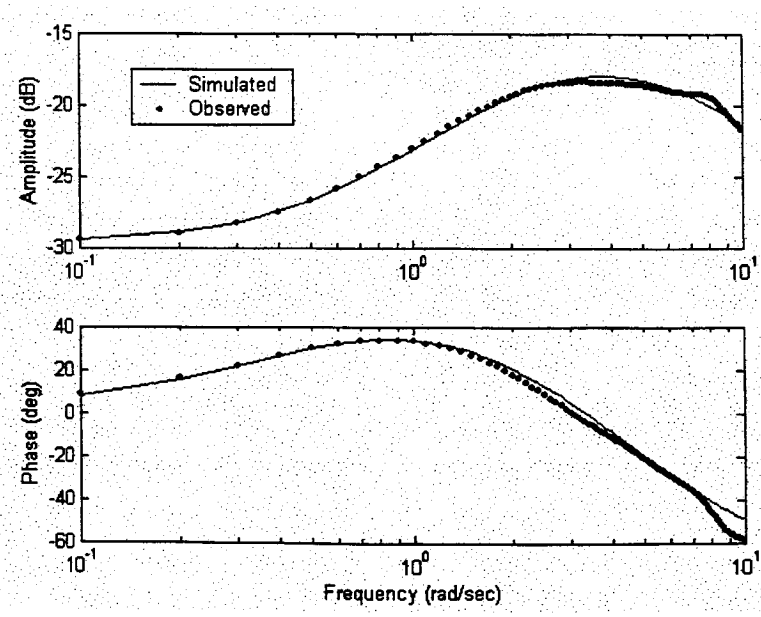


Figure 48. Comparison between Observed and Simulated Frequency Response

In Figure 49 the initial estimates of the zero and two poles were placed 0.5 rad/sec higher than actual (which affects the free response that is subtracted from the observed response). Essentially, the observed response precisely indicates the simulated zero and poles, showing robustness to zero and pole estimate error. What this indicates is that whenever an update is performed with a new data sample, the zeros and poles computed for the previous data sample will likely be a suitable initial estimate for the update (i.e., the solution is convergent).

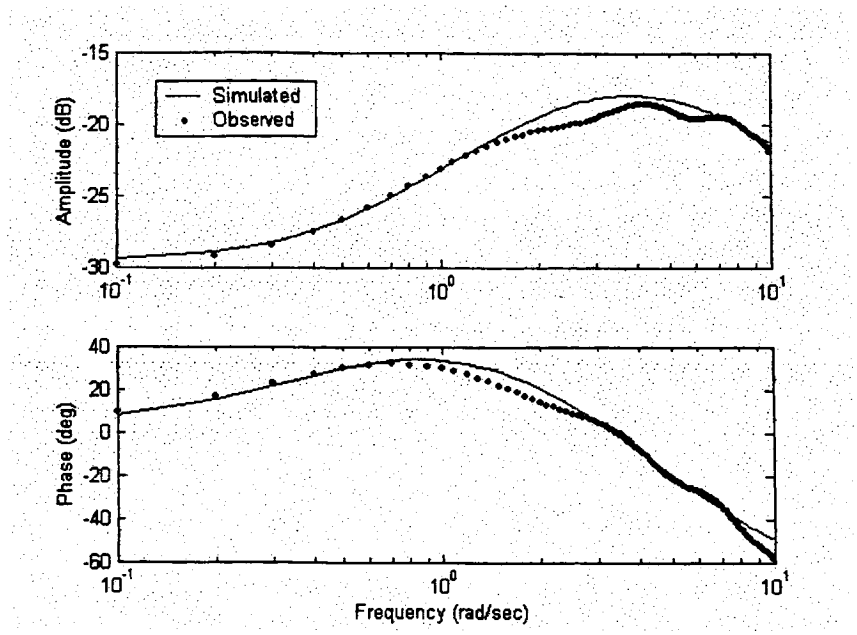


Figure 49. Comparison between Observed and Simulated Frequency Response with Initial Zero and Pole Estimates Shifted 0.5 rad/sec Higher than Actual

5.3 Estimation of Pilot Describing Function Parameters

Using the discrete wavelet transform techniques described above, estimates of time-varying pilot describing function parameters can be made. To represent a transfer function in state-space form, the transfer function must be realizable (i.e., the order of the numerator must be no higher than the denominator's order). Assuming that the pilot's describing function Y_p takes the form of lead compensation and gain, $Y_p(t) = K_p(t)(s + z(t))$, it then becomes necessary to run the pilot stick output through a filter to create a pseudo, realizable transfer function Y_p'. Experimenting with a first and second order filter, it was determined that a second order filter yielded the best results. Furthermore, a natural frequency ω_n chosen well above the pilot's zero location and a damping ratio ζ above critical damping provided good results. Figure 50 illustrates the filtering process, where the filter parameters chosen were $\omega_n = 15$ rad/sec, and $\zeta = 1$.

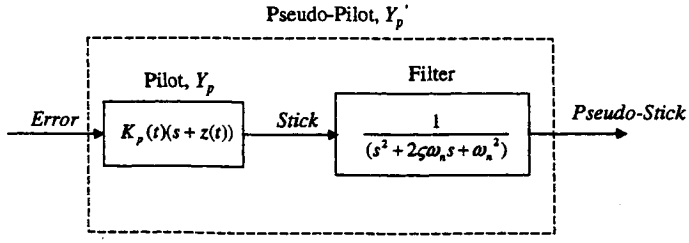


Figure 50. Filtering of Pilot Output to Produce a Realizable Transfer Function

A simulation was created where the controlled plant was a second order system (ω_n = 1 rad/sec, and ζ = 1). The pilot's describing function parameters were changed step-wise at 10 seconds and 20 seconds such that the open-loop crossover frequency (ω_c) changed step-wise as shown in Table 5.

Table 5. Pilot Describing Function Parameters

Time	K_p	z	ω_c
t≤10	4.46	3	5
10<t≤20	3.16	1	3
20<t≤30	4.46	3	5

Figure 51 shows the components of the simulation. The input to the pilot transfer function is a sum-of-sine waves generated at 100 Hz. A time window of 1.28 seconds was used for the discrete wavelet computations, and parameter values were updated at 10 Hz. In this simulation the plant's parameters are assumed rather than estimated for the computation of ω_c.

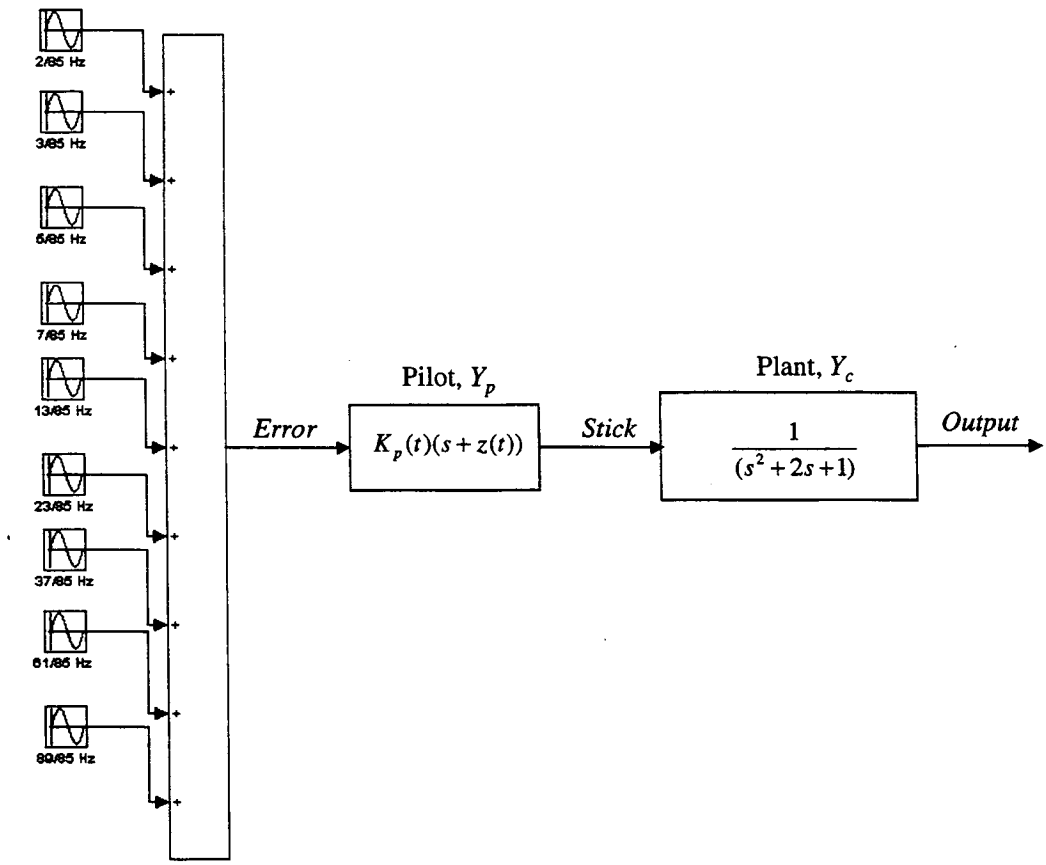


Figure 51. Simulation Components

Figure 52 shows the actual and estimated pilot parameters. At 10 seconds the estimate of K_p is seen to briefly rise before rapidly decreasing, arriving at a stable value approximately 2 seconds later. At 20 seconds K_p (estimate) immediately increases and arrives at a stable value approximately 1 second later. Estimated values of z achieve stable values at roughly the same times as the estimates of K_p, but at 10 seconds z (estimate) immediately decreases.

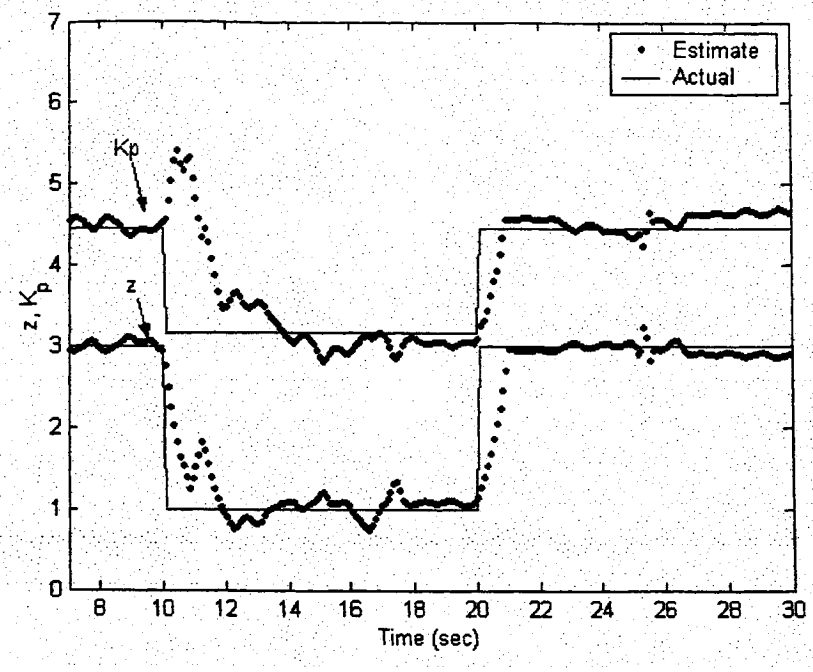


Figure 52. Time-Varying Pilot Describing Function Parameters

In Figure 53 the estimate of ω_c is compared with the actual values – there is good agreement between the two. As with the pilot describing function parameters, a decrease in actual parameter value requires approximately twice the time required to achieve steady-state compared to when the parameter experiences the same increase in absolute value.

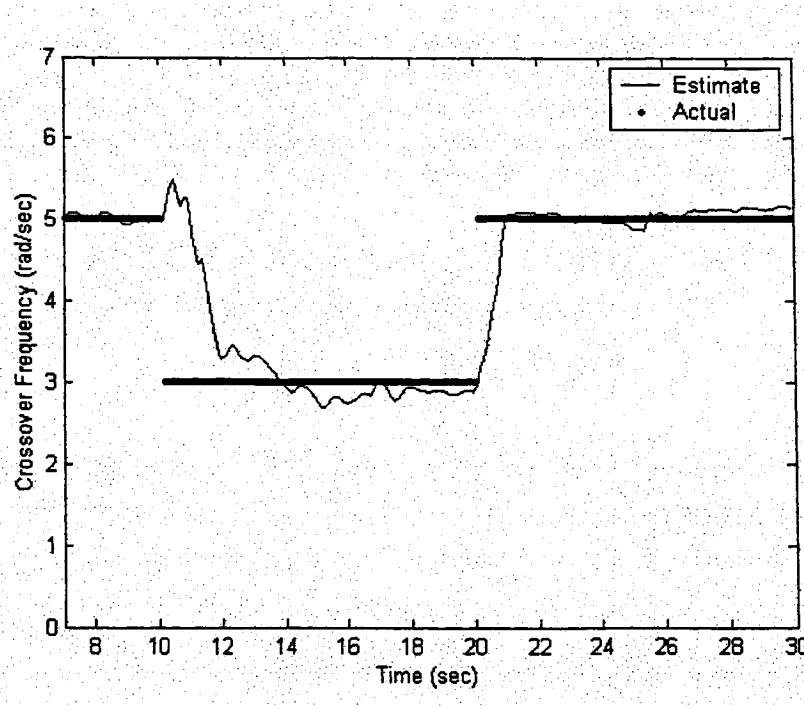


Figure 53. Time-Varying Crossover Frequency

In Figure 54 the estimated phase margin matches well with the actual values.

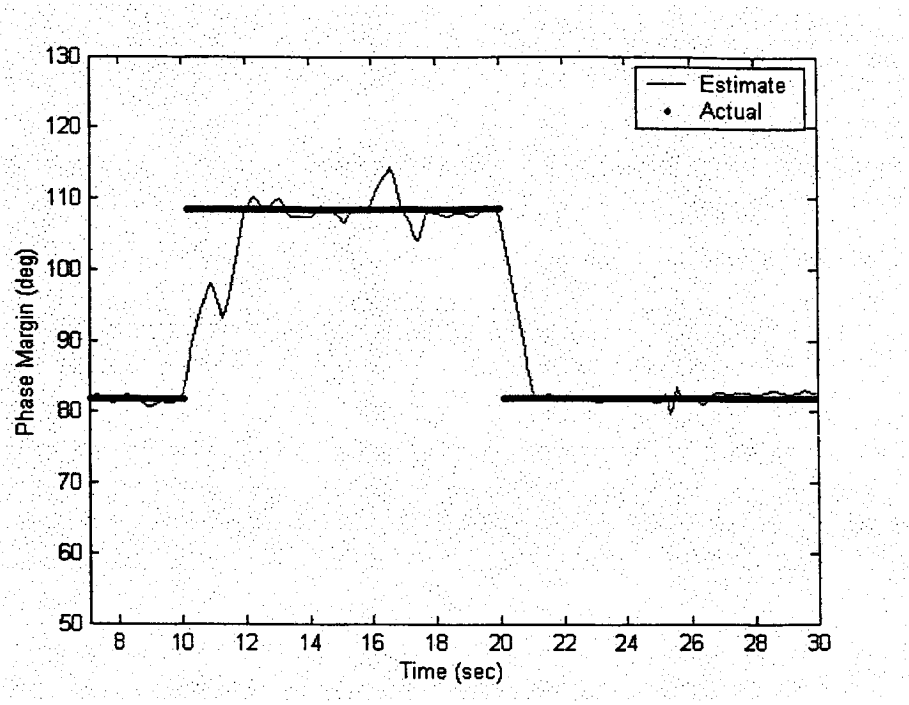


Figure 54. Time-Varying Phase Margin

5.4 Time Delay Identification

5.4.1 Time Domain Methods

Frequency domain methods for determining gain response of an element generally yield well-behaved results. However, the phase response computed using the same methods tends to be much more spurious, making stability analysis difficult. When pure time delays (or effective pure time delays arising from high frequency dynamics) are involved, time domain methods appear to be a promising alternative to frequency methods for identifying phase loss due to time delays.

Figure 55 represents the open-loop pilot response during a compensatory tracking task. In Figure 56 the error is modeled to be a sine wave, and the pilot is modeled as a pure time delay.

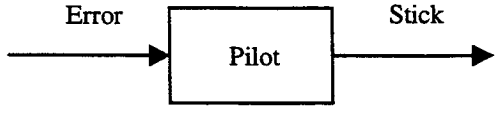


Figure 55. Open-Loop Pilot Response

5.4.2 Cross-Correlation Method

One of the methods explored for measuring the time delay was cross-correlation, illustrated in Figure 57. Within a give time window (containing an integer number of cycles), cross-correlation is computed. The stick signal is then shifted in time (the error signal remains fixed) and the computation repeated. The time delay estimate is the shift corresponding to the maximum cross-correlation value. As the time window progresses along the two signals in time, the maximum cross-correlation will correspond to varying time shifts (for a constant simulated time delay).

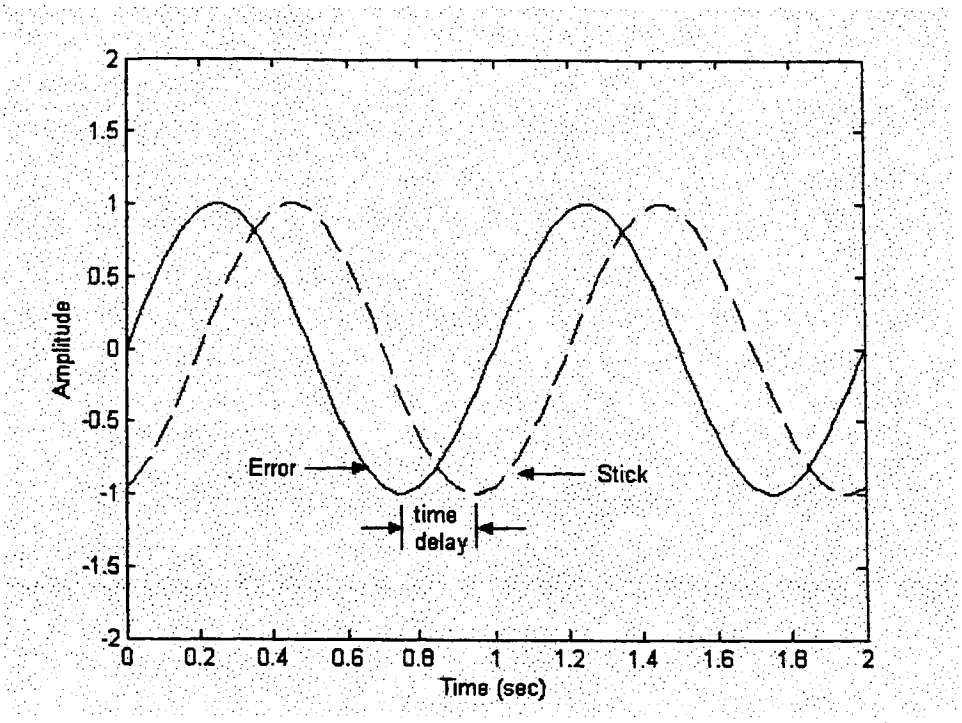


Figure 56. Pilot Modeled as a Pure Time Delay

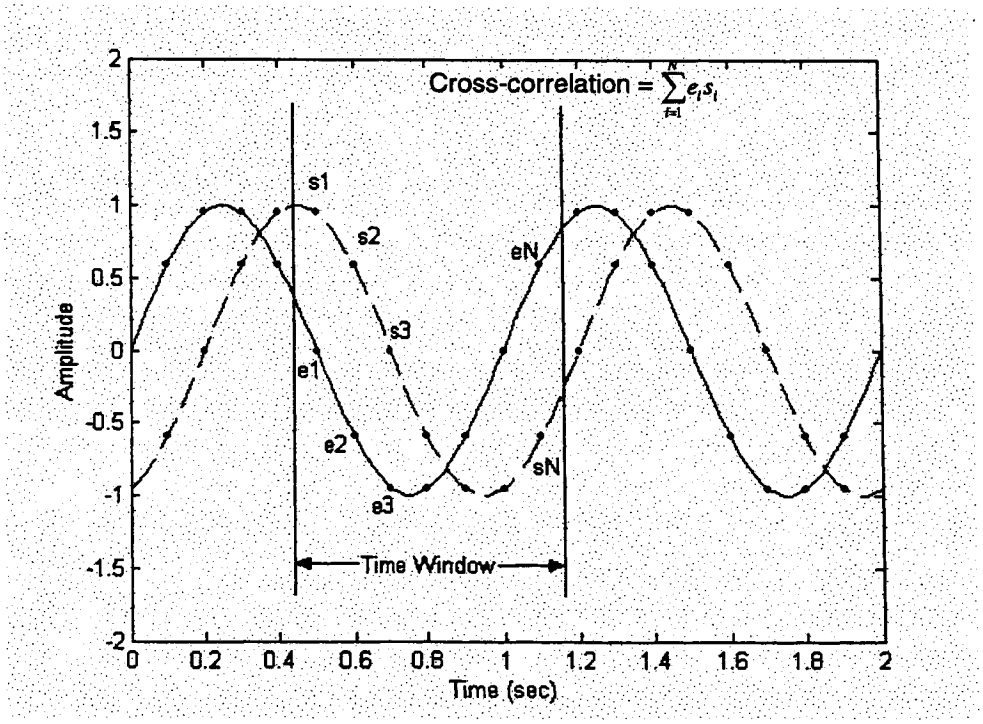


Figure 57. Cross-Correlation Technique

The result is an estimate that oscillates about a mean that is lower than actual (see Figure 58). If the time window is lengthened to include more cycles, the oscillation reduces and the mean improves, but there is a tradeoff with resolution.

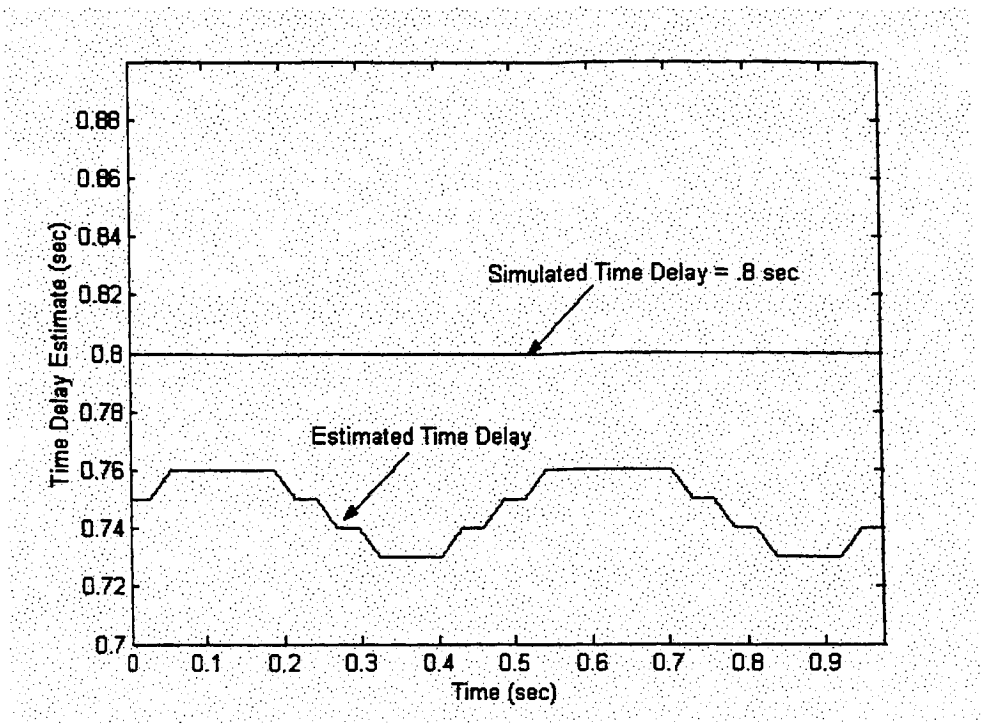


Figure 58. Bias and Oscillating Estimate Using Cross-Correlation

A way to improve the time delay estimate while limiting the window length to one cycle of the forcing function is to take the windowed data for any given point in time and concatenate the data (i.e., attach back-to-front) an arbitrary number of times. Thus a synthetic signal can be created that is of sufficient length for cross-correlation to yield a smooth and accurate estimate. However, this only is successful if the original window contains (approximately) an integer number of cycles of the error signal, and if two or more discrete frequencies are captured in this signal. This imposes four conditions: 1) the error signal must be composed of a sum-of-sine waves; 2) the error frequencies cannot be simply non-harmonics of one another – at least two frequencies must nearly fit inside the chosen time window of the and the time window; 3) one or more filters must be employed to pass only the integer-cycle frequencies; and 4) the time window cannot be arbitrarily selected, as it must accommodate at least two of the sine error frequencies. These four limitations greatly restrict the utility of this method, in terms of experimental and operational environments as well as interoperability with other analysis methods.

5.4.3 Least Squares Method

A technique that offers considerably more flexibility makes use of least squares, shown in Figure 59. Within a given time window the sum of the squared differences between the error and stick signals is computed. The stick signal is then shifted in time (the error signal remains fixed) and the computation repeated. The time delay estimate is the shift corresponding to the minimum squared difference summation.

Matlab was used to verify this time delay estimation method using an error signal composed of a sum-of-sines, shown in Figure 60. A time window of 3.7 seconds yielded a good balance between performance and resolution for the given forcing function. Using a window less than 3.7 seconds gave an increasingly noisy estimate. It was observed that the estimate suffered if frequencies lower than 2π/(time window) rad/sec were not filtered out, as well as frequencies higher than approximately 1/10 of the sampling rate. Thus the error and stick signals were band-passed within these two bounds. It is suggested that for an

actual closed-loop system the time window is chosen based on the open-loop crossover frequency (ω_c), Time Window $\approx 2\pi/\omega_c$, since crossover defines the region that most affects performance.

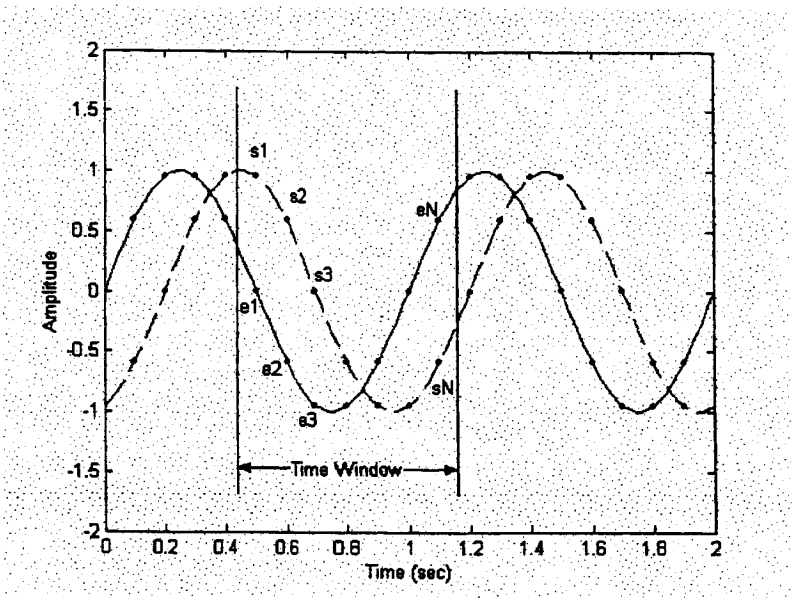


Figure 59. Least Squares Time Delay Estimation

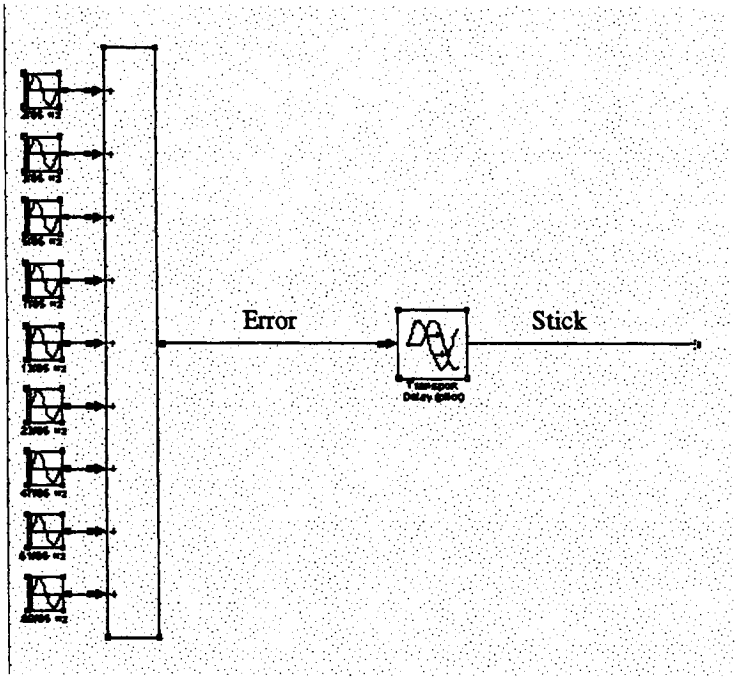


Figure 60. Sum-of-Sine Waves Used for Least Squares Verification

Three time delay conditions were simulated: 1) constant (no variation over time), 2) step changes at discrete points in time, and 3) sinusoidal variations over time. Steady-state performance is seen to be excellent in Figure 61, and estimation of discrete jumps in time delay in Figure 62 is likewise excellent. Figure 63 shows estimation of dynamic variations of time delay to be very good, where a shift due to the time window clearly discernible. Note in Figure 63 there is very little gain reduction (forcing function frequency is 1cycle/10 seconds).

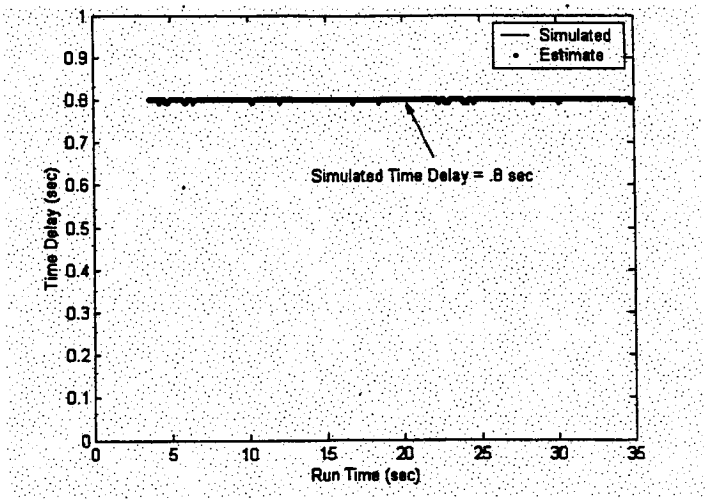


Figure 61. Simulated Constant Time Delay (Time Window = 3.7 seconds)

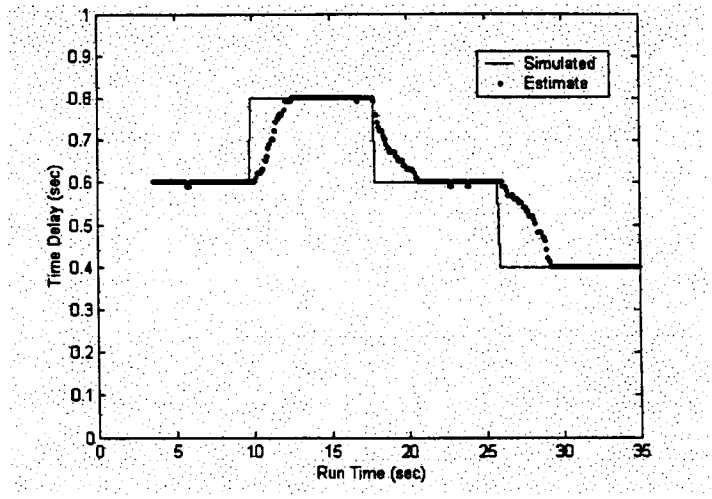


Figure 62. Simulated Step Time Delay (Time Window = 3.7 seconds)

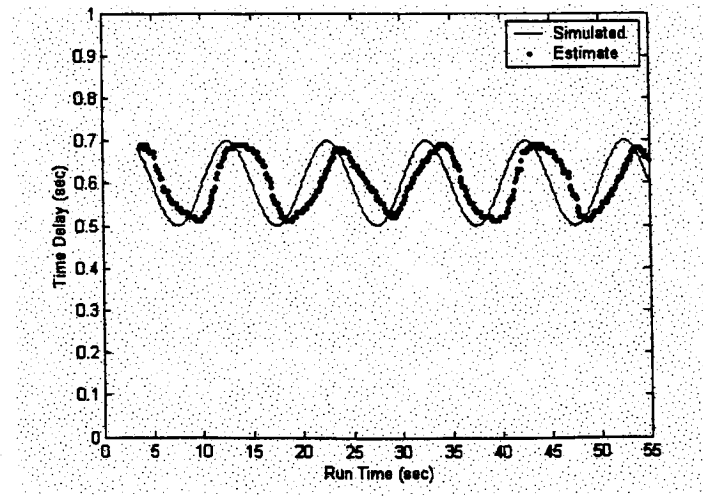


Figure 63. Simulated Sinusoidal Time Delay (Time Window = 3.7 seconds)

5.4.4 Time Delay Estimation Using Inverse Transfer Function Method

When the bandpass filter and the estimated system model does not take into account initial conditions, the time window must be large enough so that the transients arising from zero-initial conditions are no longer a factor. By taking into account initial conditions in the filter and system model, the time window can be appreciably reduced. Figure 64 shows the conditioning processes for the system input and output signals, where the system output is run through the inverse of the estimated system to give a pseudo-input that ideally should match the actual input except for a pure time delay. A confidence measure is computed by taking the linear correlation of the system input with the pseudo-system input.

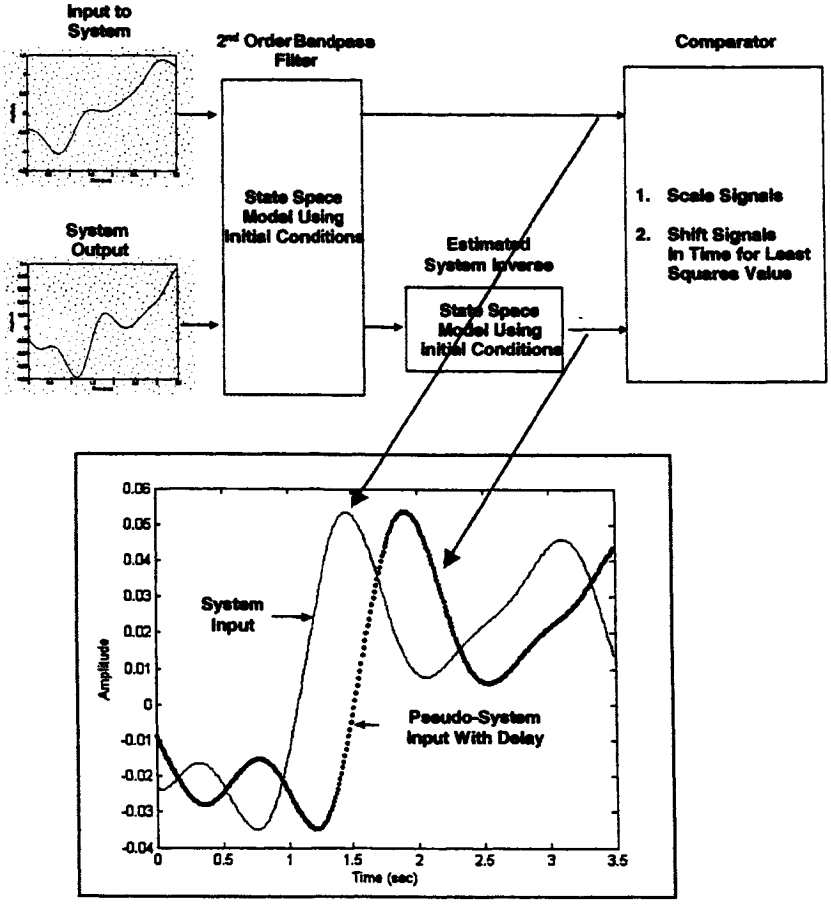


Figure 64. Processes Used for Estimating Operator Pure Time Delay

Figure 65 shows a Matlab-Simulink model for varying pilot time delay using low-passed white noise. This model results in a pure time delay that changes over the entire time window that is used to compute the time delay, so that there will always be a degree of uncertainty associated with this measurement. A sum-of-sine waves signal was run through a second-order system (first-order lead) having the time-varying pure time delay, and the results are shown in Figure 66. In Figure 67 step changes are introduced in damping, natural frequency and gain, and the exact values of these parameters are used in the calculation of equivalent time delay.

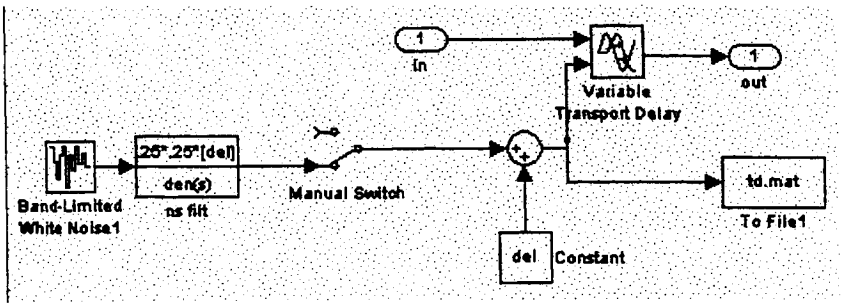


Figure 65. Time-Varying Operator Delay Generator

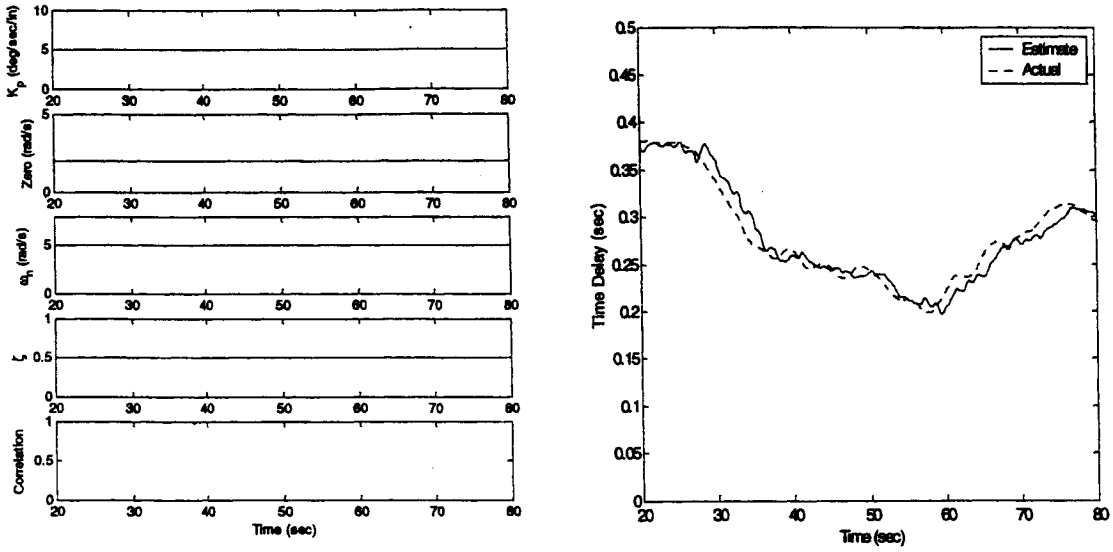


Figure 66. Time-Varying System Delay, Constant Parameters, Sum-of-Sines Input

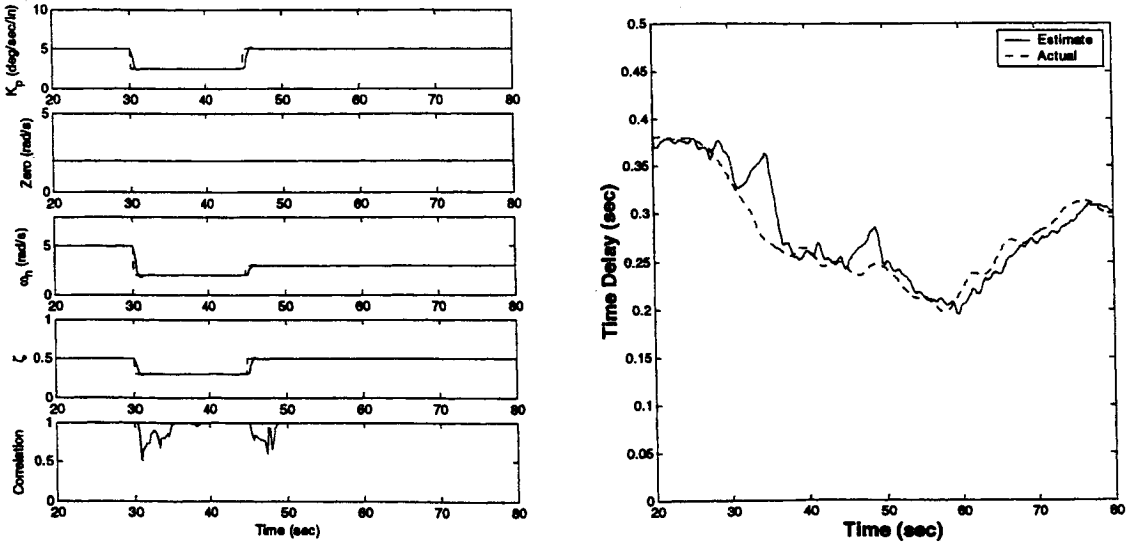


Figure 67. Time-Varying System Delay with Parameter Step Changes, Sum-of-Sine Waves Input

5.5 The WERA Method for Parameter Identification

5.5.1 Computational Procedures

The discrete Wavelet transform/Eigensystem Realization Algorithm (WERA) running time method assumes the system order and constructs from this a discrete state-space model based on an initial model estimate. Using this model, the state initial conditions are computed and the free (unforced) solution calculated for the length of the measurement window. This free response is then subtracted from the observed response, leaving the forced response. This is done because the convolution integral is valid (for practical applications) only when the initial conditions are zero. Obtaining the forced response allows use of the convolution integral in conjunction with the discrete wavelet transform. This produces the impulse response (Markov parameters), which is used by the ERA to compute the numerator and denominator of the system whose order is specified. This method works well for identifying key, but not all, parameters of a second-order system. The remaining parameters are obtained using time-domain methods.

The following aircraft short period (SP) approximation was used to model the pitch rate response dynamics of the aircraft model used in the LOCATS simulation evaluation:

$$\frac{\dot{\theta}(s)}{\delta_\theta(s)} = K\frac{(s+z)}{(s^2+2\zeta\omega_n s+\omega_n^{\,2})}e^{-\tau}$$

The WERA method was used to obtain the Bode gain and natural frequency by:

- Selecting a window length based on the current natural frequency estimate. As the natural frequency decreases the window length must increase for satisfactory identification.
- Lowpass filtering the input/output (I/O) signals (7 rad/sec) based on the range of frequencies the short period approximation is valid over. Identification of the pitch dynamics using a static version of the WERA method (where the initial conditions were in fact zero) yielded dynamics that departed substantially from the SP approximation at frequencies above 10 rad/sec, due to stick, actuator, and control augmentation.

An impulse response of the state-space estimate using ERA was produced, and this was compared to the observed impulse response (obtained via the DWT). The differences between the two were squared and normalized, and the inverse was used to weight how much the previous estimate could be incremented in the direction of the currently computed parameters.

The zero location of the SP approximation was computed in a similar fashion, except that the output response was filtered using a transfer function whose numerator was the estimated denominator of the system (thus canceling it out). The second-order denominator was given a natural frequency above the lowpass frequency (10 rad/sec) to separate the zero and natural frequency dynamics.

The effective time delay of the system was computed using the following steps:

- Bandpass the I/O, (0.85 to 10 rad/sec) using a first-order butterworth filter constructed using a state-space model of the filter and then computing the initial conditions of the filter states. This minimizes the transients that occur due to filtering, thus reducing the window length that is required to identify the time delay.
- Construct a transfer function that is the inverse of the assumed system model, and make it realizable by convolving the denominator (which is first-order) with a high-frequency pole (20 rad/sec). A state space representation of this transfer function is constructed, and the bandpass-filtered response of the aircraft (pitch rate) is fed

through this system using computed initial conditions. The bandpass-filtered input to the aircraft, stick position, is run through a first-order transfer function whose denominator is the same high-frequency pole (20 rad/sec) mentioned above, once again using initial conditions.

- The conditioned output is scaled with the conditioned input using the ratio of the maximum values of those signals observed within the time window (3.5 seconds). The amount that this ratio can change from one update to the next is bounded, so as to minimize perturbations caused by sudden system parameter changes.
- The output is shifted in time and compared with the input signal, and the time shift associated with the minimum sum of the squared differences corresponds with the time delay estimate.
- A linear correlation is computed between the shifted input and output signals, and used to weight the amount by which a time delay estimate can be incremented.

The damping of the SP approximation is computed by constructing a discrete state-space model of the system, and incrementing damping values over a limited range. The damping producing the minimum sum of the squared differences (between the observed and trial response) is selected.

The roll rate response dynamics of the simulator aircraft model were modeled using a first-order approximation along with an effective time delay:

$$\frac{\dot{\phi}(s)}{\delta_{\phi}(s)} = \frac{K}{(s+p)} e^{-\tau s}$$

The WERA method was used to compute both the gain and break frequency in a single step (the I/O signals were lowpass-filtered at 15 rad/sec), and the time delay was computed the same way as for the pitch rate response.

When there are relatively prolonged periods of quiescence in the input, the identification process degrades. Rather than depend on metrics such as linear correlation to indicate when this has occurred (by which time the identification results are already deteriorating), the stick activity is measured so as to anticipate poor identification conditions. To do this, the low-frequency gain of the system is computed based on current parameter estimates, and multiplied by the standard deviation of the stick over the last 20 milliseconds. This provides a metric (μ) that is essentially independent of the system it is used with. It was found that a μ value of 0.025 and above yielded satisfactory results. If μ dropped below this threshold, the parameters were frozen at their current values until μ again rose above 0.025.

5.5.2 Parameter Identification from a Simulation Model

A sum-of-sine waves input was fed through a time-varying transfer function (short period approximation with an equivalent time delay) whose actual parameters are shown in Figure 68 by dashed lines. These parameters were estimated with the WERA method described above, and the correlation between the actual input and the pseudo-input (using the inverse of the estimated transfer function) is seen to be excellent except for the transient period following step changes. In Figure 69 the input driving the system is band-passed white noise, and identification in general is good. When the damping is increased at 45 seconds, however, the damping estimate fails to track the change.

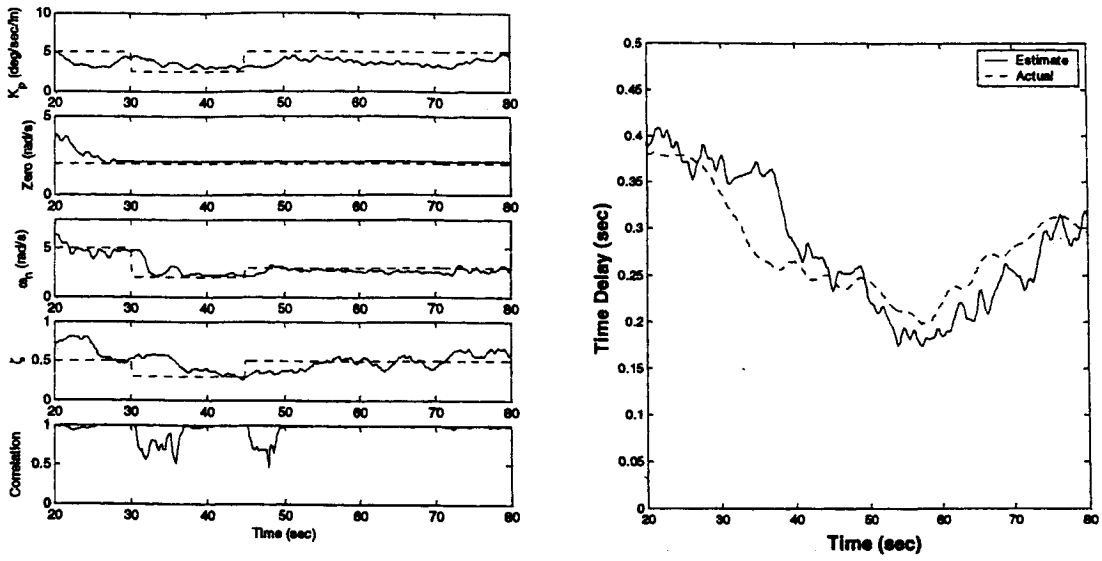


Figure 68. Time-Varying System Delay with Estimated Parameter Changes, Sum-of-Sine Waves Input

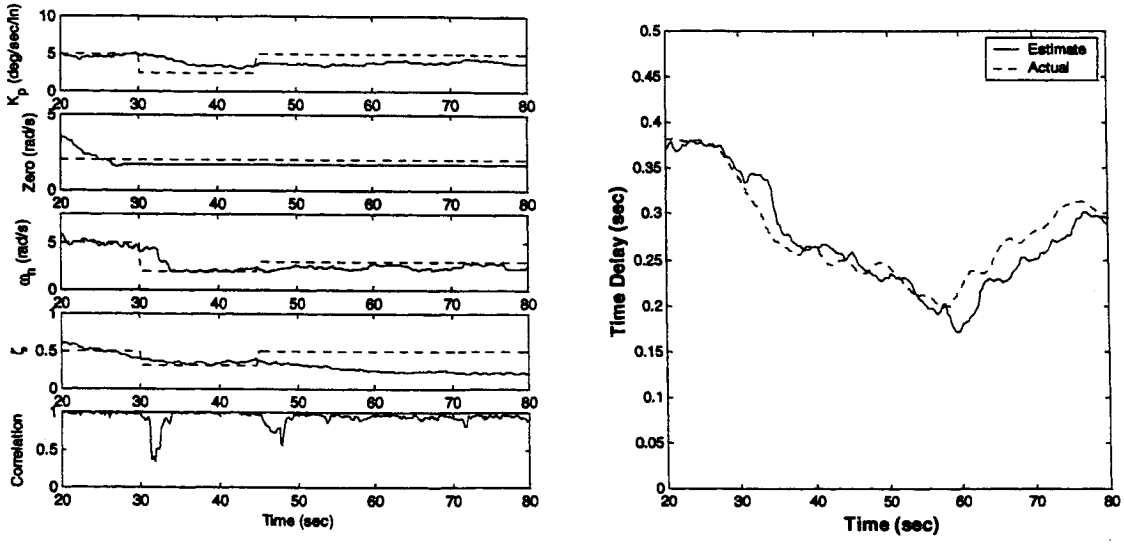


Figure 69. Time-Varying System Delay with Estimated Parameter Changes, Band-Passed White Noise Input

5.5.3 Parameter Identification from Flight Test Data

From October 1990 to March 1991 the U.S. Navy conducted a flying qualities evaluation of the F-14 with simulated dual hydraulic failure (Ref. 35). In this study, the back-up flight control module or BUFCM was evaluated to define areas of operation for in-flight refueling and landing. The BUFCM has two modes that are available to the pilot. The BUFCM-HIGH mode features a maximum stabilator rate of 10 deg/s, while BUFCM-LOW has a 5 deg/s maximum rate. Although the aircraft demonstrated good handling qualities using the BUFCM in formation flight with a tanker, a number of PIOs were encountered during in-flight refueling, drogue tracking, and offset field landings. Because the F-14 was fully instrumented, a valuable PIO database was created.

An analysis of this database was previously undertaken using Fast Fourier Transform techniques as reported in Ref. 36. Although the entire database is of interest, Run J_09 is of particular interest to this

project, because as shown in Figure 70 it features intermittent pilot-induced oscillations. That is, there is an initial PIO that occurs between 12 and 22 seconds, followed by a quiescent period up to 42 seconds or so, when a second PIO of smaller amplitude occurs. In the two PIO regions the rate limiting associated with the BUFCM-HIGH configuration is clearly visible in the triangle wave-like output evident in the stabilizer response.

The pitch rate response to stick position were modeled with the short period approximation and an equivalent time delay:

$$\frac{\dot{\theta}(s)}{\delta_\theta(s)} = K\frac{(s+z)}{(s^2+2\zeta\omega_n s+\omega_n{}^2)}e^{-\tau s}$$

Figure 71 shows the estimated equivalent time delay and time-varying airplane bandwidth parameters for the drogue tracking run. In general all three metrics are in consonance with the occurrence of PIO, although the changes are relatively small. That is, the aircraft system is always "bad" in terms of the airplane bandwidth criteria, but it is "worse" in the two PIO regions. The changes in the bandwidth parameters correlate well with the changes in pilot gain as seen by the amplitude changes in the Figure 70 stick force response. Specifically, the lower stick amplitudes in the non PIO region reduced the severity of the rate limiting. The pilot then attempts to track the drogue again, the rate limiting becomes more severe, and the second PIO results.

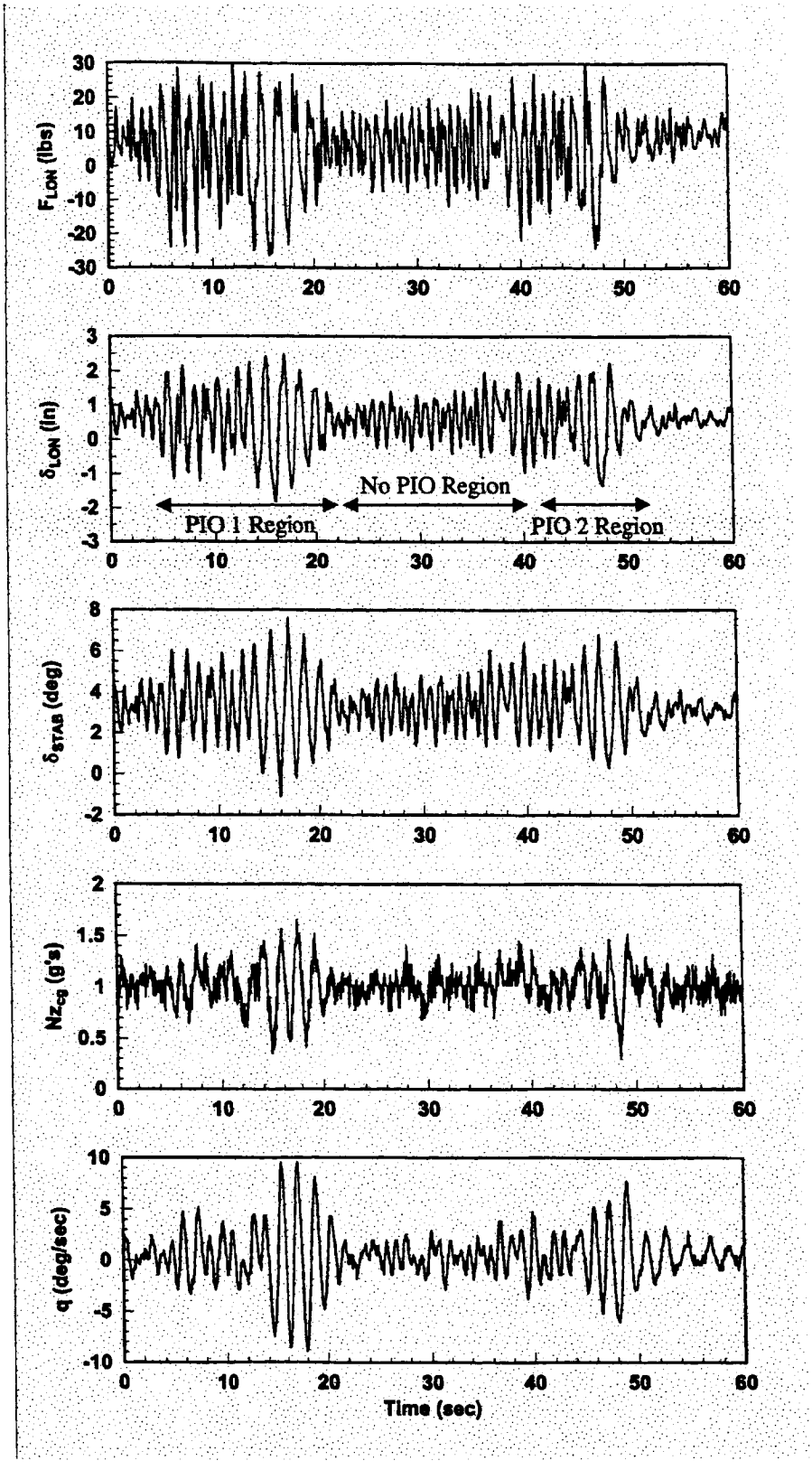


Figure 70. F-14 BUFCM Drogue Tracking Time Histories with Intermittent PIO

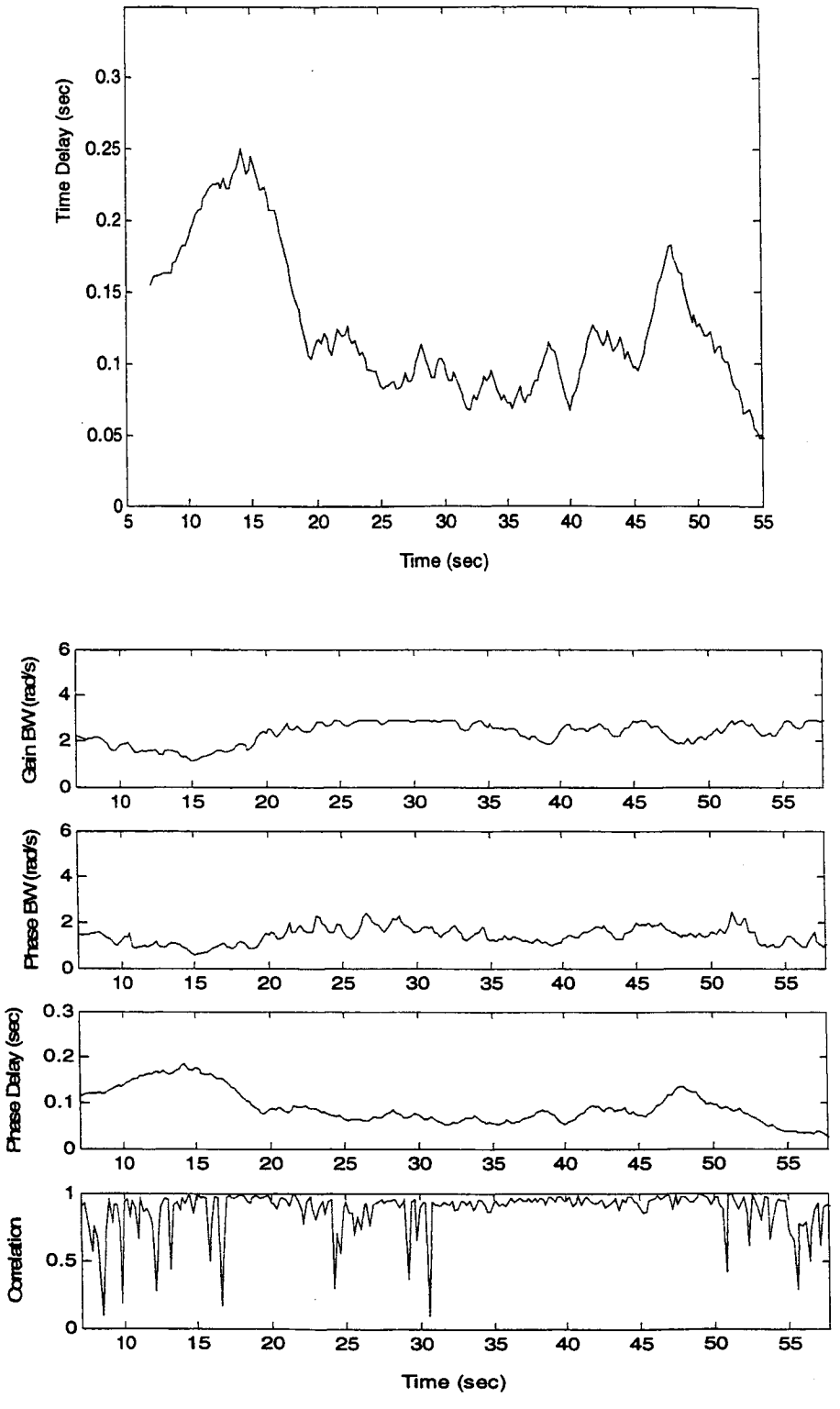


Figure 71. Equivalent Time Delay and Airplane Bandwidth Parameters

6. LOCATS SIMULATION EVALUATION RESULTS

6.1 Background

6.1.1 LOCATS Evaluation Success Criteria

There are two fundamental factors that will determine the ultimate success of the LOCATS concept. First, the LOCATS software must be able to detect impending loss of control events while minimizing "false alarms" and/or missed events. Second, the time to detection must be minimized, so that the pilot or automatic flight control system can react to the situation. Thus for a given flight control system failure scenario, the following questions will be addressed:

- Can the wavelet-based computations run in a real-time environment?
- Was loss of control or impending loss of control observed?
- If yes, what was the triggering event?
- If yes, was the event detected by the LOCATS software?
- If no, was a non-event detected by the LOCATS software?
- What metric(s) provided the detection mechanism?
- What was the detection time?
- Can the detection time be improved?

Later versions of the LOCATS software may include recommended courses of action that could be provided to the pilot in the form of a flight director or as direct commands to flight control system.

6.1.2 Flight Conditions, Aircraft Configurations, and Pilot Evaluation Tasks

6.1.2.1 Flight Conditions

The piloted evaluations will be conducted in both up-and-away and terminal flight conditions. The specific flight condition variations used in the LOCATS formal evaluation are identified in Table 6. Each evaluation began with the autopilot and autothrottle on and the aircraft trimmed. The steep approach flight conditions defined in the original test plan (see Volume II) were not used in the formal evaluations.

Table 6. Flight Conditions for the Piloted Simulation

No.	ID	Description	Altitude (1000 ft)	Airspeed (KCAS)	Turbulence	Crosswinds (knots)
1	C	Cruise	20	270	None	n/a
2	CM	Cruise with Turbulence	20	270	Moderate	n/a
3	CH	High Altitude Cruise	40	200	None	n/a
4	A	Approach	0.2	130	n/a	None
5	AX	Approach with Crosswind	0.2	130	n/a	20, R to L

6.1.2.2 Aircraft Configurations

As mentioned above the piloted evaluations were conducted in both up-and-away and terminal flight conditions. The four aircraft configurations used in the LOCATS formal evaluation are identified in Table 7. These represent a subset of the ten configurations that were defined as part of the test plan (see Volume II).

Table 7. Aircraft Configurations for the Piloted Simulation

No.	ID	Description	Gross Weight (1000 lb)	CG (%chord)	Flaps (deg)	Gear
1	CHF	Cruise/Heavy/Forward CG	580	0.3	Retracted	Up
2	CLF	Cruise/Light/Forward CG	300	0.27	Retracted	Up
3	CLA	Cruise/Light/Aft CG	300	0.42	Retracted	Up
4	ALA	Approach/Light/Aft CG	300	0.42	27.4	Down

6.1.2.3 Pilot Evaluation Tasks

The pilot evaluation tasks used in the LOCATS evaluation were as follows:

- Pitch Attitude Capture and Hold (PACH): From steady level flight the pilot rapidly captured a pitch attitude of at least 5° above trim and maintain this attitude within the specified tolerances for 5 seconds or until stable. The target attitude was defined by a clearly displayed reference on the HUD pitch ladder. The pilot then performed subsequent 5° captures to complete the maneuver. Before proceeding to the next capture, the pilot maintained each attitude within the specified tolerances for 5 seconds or until stable.
- Bank Angle Capture and Hold (BACH): From steady, wings level flight the pilot rolled the aircraft and captured a bank angle of 30° and maintained this bank angle within the specified tolerance for 5 seconds or until stable. Then the pilot made subsequent capture and holds of 0° and -30°. In some cases 20° bank angle captures were used to increase the aggressiveness of the task.
- Pitch and Roll Sum-of-Sines Tracking (SOS): The pilot aggressively tracked error bars displayed on the HUD and attempted to keep the bars within the displayed tolerance reticles. The pitch and roll error bars were driven by signals representing the difference between the commanded attitude and the actual attitude. In this case the commanded attitude was driven by the sum of seven sine waves as defined in Volume II of this report. This task could be run in single or dual axis mode.
- Pitch and Roll Discrete Tracking (DT): Pitch and Roll Sum-of-Sines Tracking (SOS): The pilot aggressively tracked error bars displayed on the HUD and attempted to keep the bars within the displayed tolerance reticles. The pitch and roll error bars were driven by signals representing the difference between the commanded attitude and the actual attitude. Here the commanded attitude was driven by a set of discrete steps and ramps as defined in Volume II of this report. This task is only effective in dual axis mode.

Detailed maneuver descriptions and related background material are provided in Volume II of this report.

6.1.3 Simulated Flight Control System Failures

Flight control failures were introduced into the simulator at the discretion of the operator using the simulator change-a-gain feature. This feature allowed flight control system parameters to be varied from a default value to some new value, while the simulator was operating. The change-a-gain feature provided a choice of eight parameter values including the default value. Because of the interest in inducing loss of control, only the most severe changes from the default value were used. Table 8 provides a list of simulated flight control system failures based on the results of the simulation checkout as described in Volume II of this report. In addition, a new scenario was introduced involving the roll axis. As the formal evaluations progressed it became clear that loss of control in the roll axis (i.e., PIO or

otherwise) could not be induced with the available flight control system changes. To help "loosen" up the roll axis, a new change-a-gain option was made available that eliminated the sideslip rate feedback in the roll axis via the GBDEST gain. Eliminating this gain alone did not produce a noticeable change to the pilot. When used in conjunction with the elimination of the roll rate feedback, however, it was possible to generate at a minimum mild PIO in the roll axis. To coordinate its use in the evaluation process, this gain change was only used in conjunction with a PGain change. As mentioned in the previous section, multiple key strokes are required to make a single change. Within the given time constraints, it was not possible to add the GBDEST to the saved signal list. Thus, the BGain change was always made first, followed by the PGain. Table 9 lists the change-a-gain values used in the LOCATS formal evaluation.

Table 8. Simulated Flight Control System Failures

No.	ID	Description	Demonstration Maneuvers
1	PitchTC	Alter pitch stick filter time constant via change-a-gain	All pitch axis and dual axis tasks
2	PitchTD	Inject pitch stick time delay	DT, SOS, PACH
3	RollTD	Inject roll stick time delay	DT, SOS, BACH
4	QGain	Alter pitch rate feedback via change-a-gain	All pitch axis and dual axis tasks
5	PGain	Alter roll rate feedback via change-a-gain	All roll axis and dual axis tasks
6	BGain	Alter GBDEST via change-a-gain (only used in conjunction with PGain)	All roll axis and dual axis tasks
7	QLIM	Reduce pitch rate command path limiter	DT, SOS, PACH
8	PLIM	Reduce roll rate command path limiter	DT, SOS, BACH

Table 9. Change-a-Gain Values for Simulated Flight Control System Failures

No.	ID	Default Value	Changed Value
1	PitchTC	0 sec	0.25 sec
2	PitchTD	0 msec	240 msec
3	RollTD	0 msec	240 msec
4	QGain	0.4	0
5	PGain	1.0	0
6	BGain	1.2	0.2
7	QLIM	25 deg/sec	7.5 deg/sec
8	PLIM	120 deg/sec	36 deg/sec

6.2 Piloted Simulation

6.2.1 Overview

During the formal evaluation of the LOCATS system 122 analysis runs were completed. Of these 41 were diagnostic maneuvers that are used to characterize the configurations used in the formal evaluation or to examine the real-time properties of the wavelet-based computations. The diagnostic maneuvers included pitch and roll axis doublets and pitch and roll axis frequency sweeps. The maneuvers were conducted with autothrottle on and off. Throttle frequency sweeps were conducted with the autopilot on and off. The remaining 81 runs featured the flight control system failure scenarios defined in Table 8. Because of the limitations of the types of failures that could be introduced via the change-a-gain feature, the aircraft

could not be made to depart. It was possible, however, to repeatedly produce significant pilot-induced oscillations. A subset of these runs is discussed below as part of the LOCATS assessment. A complete run log is provided in Volume II of this report.

As mentioned above a number of diagnostic maneuvers were conducted as part of the formal evaluation. The pitch and roll axis frequency sweep runs were used to obtain the airplane bandwidth parameters for the baseline configurations. The results are tabulated in Volume II of this report. In short, the pitch attitude tracking handling qualities of the baseline configurations fall solidly in the Level 2 region for bandwidth frequency. However, when the phase delay is taken into account, the configurations move into the Level 3 region. The roll attitude bandwidth frequencies of the baseline configurations fall into the Level 1 region, but the phase delay values, once again, move the configurations into the Level 3 region. The analyses documented in Volume II of this report reveal that a significant portion of the high phase delay values result from the dynamics of the pilot control stick.

These results would seem to indicate that the aircraft model used in the LOCATS evaluation should be highly susceptible to PIO. It has been shown time and again, however, that it is difficult to induce PIO in a fixed base simulator with configurations and tasks that replicate flight test conditions (Refs. 5, 37 and 38). As discussed in Ref. 5, higher gain tasks or other artificial means must be employed when using a fixed base simulator to induce PIO or to replicate PIO from flight. In the Ref. 38 simulator experiment, added time delays, increased stick gains, and added wind gusts and turbulence were used to better replicate flight test results. Although the results were not wholly satisfactory to the authors since the "good" handling configurations were affected by the changes just as the "bad" handling configurations were, PIO was more easily induced. It should be no surprise, therefore, to learn that the addition of significant added time delay was required to consistently generate PIO in the BAE SYSTEMS simulator. The addition of moderate turbulence was also useful in this regard. For the LOCATS assessment an added control stick time delay (pitch and/or roll) of 240 msec was used. This large added delay when used in conjunction with the Table 8 FCS failures regularly produced PIO.

The following sections describe the simulator results for the four piloted evaluation tasks used in the formal LOCATS evaluation. To assess the TVTF and WERA analysis methods, a subset of the 81 evaluation runs that represent a cross section of no PIO, mild PIO, and sustained or severe PIO cases were selected. To make best use of the available tools including the more recently incorporated transfer function estimation techniques (see Appendix B), the analyses were conducted with the STI wavelet toolbox. An assessment of the real time computation capabilities of the wavelet-based algorithms was made using LOCATS and is also reported on herein.

6.2.2 Bank Angle Capture and Hold (BACH) Evaluations

6.2.2.1 Selected Evaluation Runs

A log of the selected BACH evaluation runs is provided in Table 10. The table features the same format used for the complete run log that is provided in the Volume II report. The flight condition, aircraft configuration, and flight control system failure acronyms correspond to those defined previously in Table 6, Table 7, and Table 8, respectively. As mentioned above, elimination of the roll rate feedback (PGain) alone was not enough to induce PIO. This is reflected in the R120 case that is included with the no PIO runs. Time histories and TVTF and WERA analysis results for all of the Table 10 runs are provided in Volume II. The more detailed analysis of an example no PIO, mild PIO, and sustained or severe PIO run is provided below.

Table 10. Selected BACH Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R77	A	ALA	BACH	n/a	
R80	AX	ALA	BACH	n/a	
R120	CH	CLA	BACH	PGain	
Mild PIO Runs					
R28	C	CLA	BACH	BGain + PGain	Mild, damped oscillations
R79	A	ALA	BACH	BGain + PGain	RollTD whole run, Large overshoots
R122	CH	CLA	BACH	BGain + PGain	RollTD whole run, mild PIO in middle or run
Sustained or Severe PIO Runs					
R29	C	CLA	BACH	BGain + PGain + RollTD	Large overshoots follow failure and lead to PIO, oscillations damp out
R82	AX	ALA	BACH	BGain + PGain	RollTD whole run, mild PIO transitions to severe, sustained oscillations

6.2.2.2 Analysis of a No PIO Run (R77)

Lateral stick position and roll attitude time histories for R77 are shown in Figure 72. The solid horizontal lines on the roll attitude plot indicate the target attitude for the captures. Not that although no PIO is evident in the run, several captures have at least one significant overshoot. This may be in part due to the dynamic characteristics of the control stick in roll that are documented in Volume II of this report.

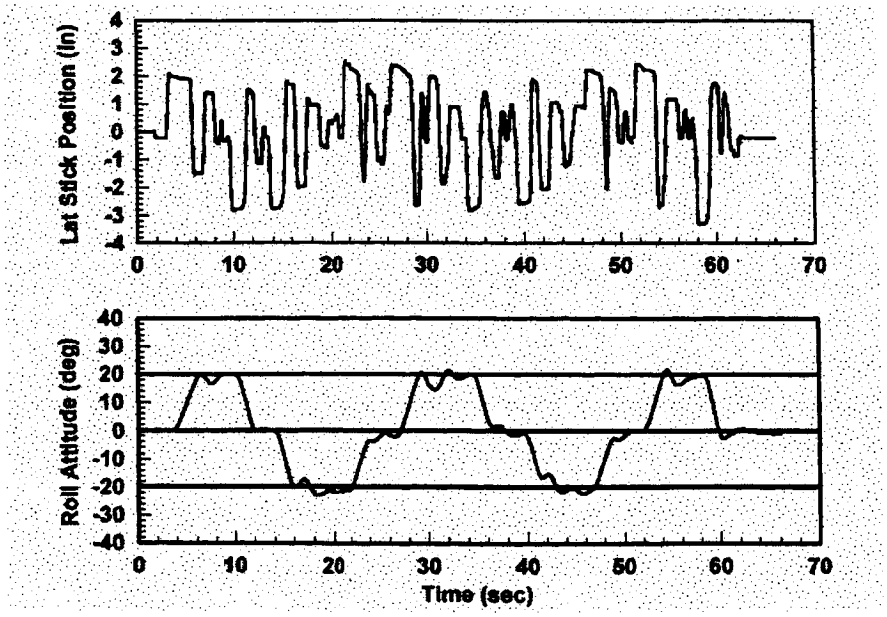
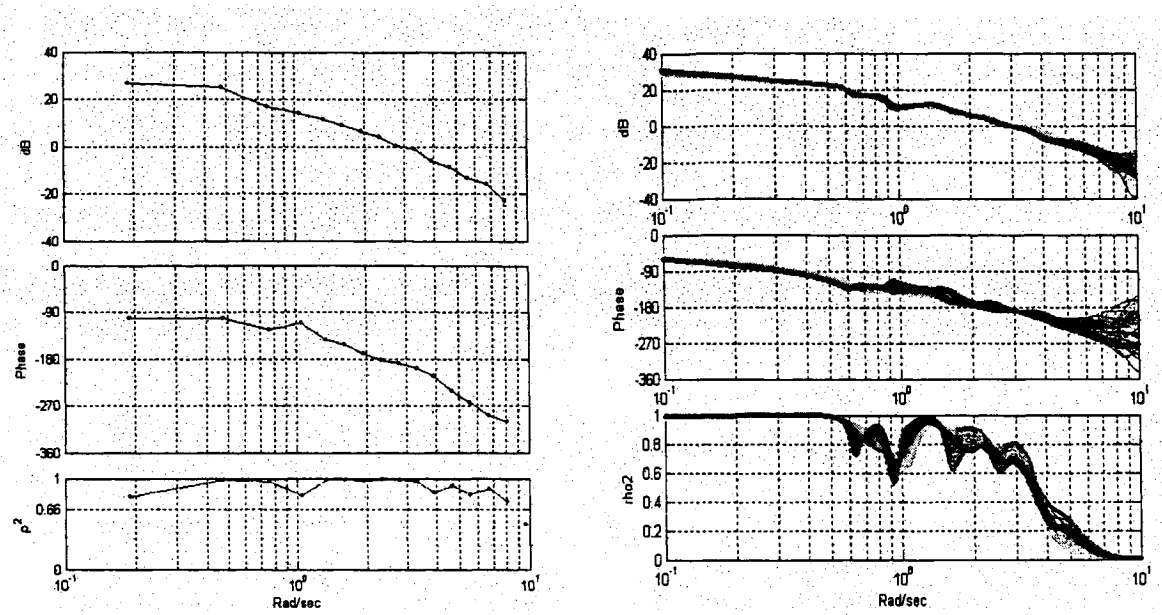


Figure 72. R77 BACH, No PIO

Evaluation runs, such as R77, that did not involve a flight control system failure were used as control runs to compare the wavelet-based transfer function estimation methods with more traditional estimation techniques. Although not shown here comparisons were also made with results generated from the diagnostic maneuvers documented in Volume II of this report. A comparison of the TVTF and FFT roll attitude to lateral stick position transfer functions is provided in Figure 73. The TVTF plot is a snapshot in time (taken at 32.6 sec) with persistence lines that fade from dark to light over 10 seconds. There is good agreement between the two methods particularly in the important 0 dB magnitude crossover region and the -90 to -180 deg phase region.

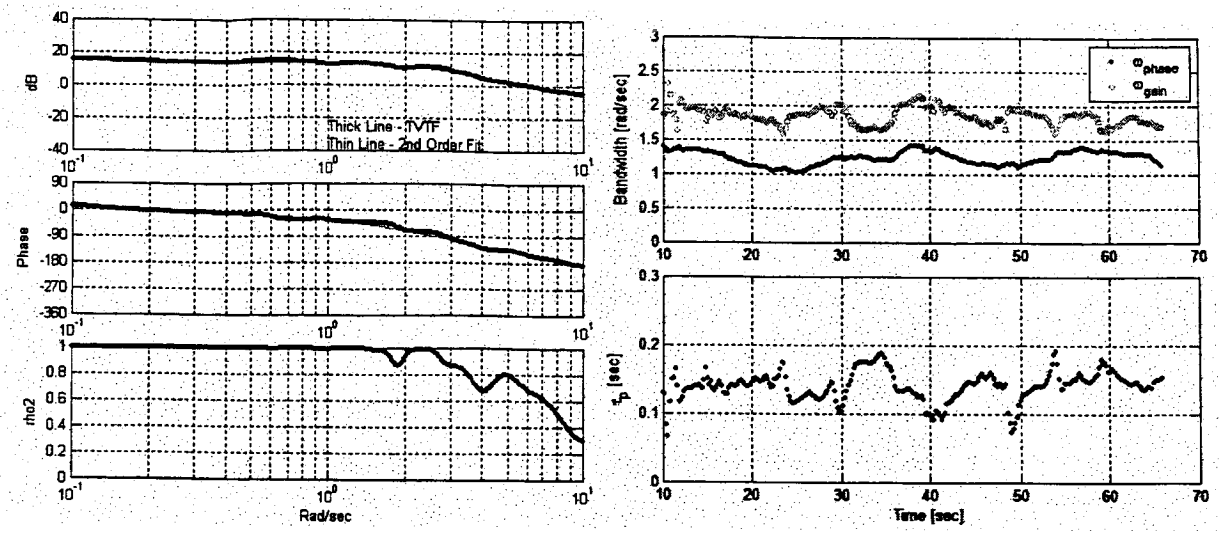


a) FFT b) TVTF at 32.6 sec

Figure 73. R77 PHI/DLAT Transfer Function Estimation

Using the techniques described in Sections 4 and 5 time varying airplane bandwidth parameters were computed. The TVTF estimates were made using a 5 cycle Erlang mother wavelet with 21 point time averaging. (Note that the 5 cycle Erlang mother wavelet was used for all of the subsequent TVTF analyses described herein.) Smoother results were obtained with 41 point time averaging, but as one would expect, the increased time averaging increases the detection time. Analysis plots for all of the selected runs with both levels of time averaging are provided in Volume II of this report. For the roll axis cases, 2nd order fits with an effective time delay were made to the TVTF roll rate to lateral stick frequency responses. The rate signal was used instead of attitude because as indicated in Figure 74a improved TVTF coherence was obtained with this signal. The fits were then used to compute time varying airplane bandwidth parameters. The R77 results are shown in Figure 74b and as expected are fairly uniform over the length of the run.

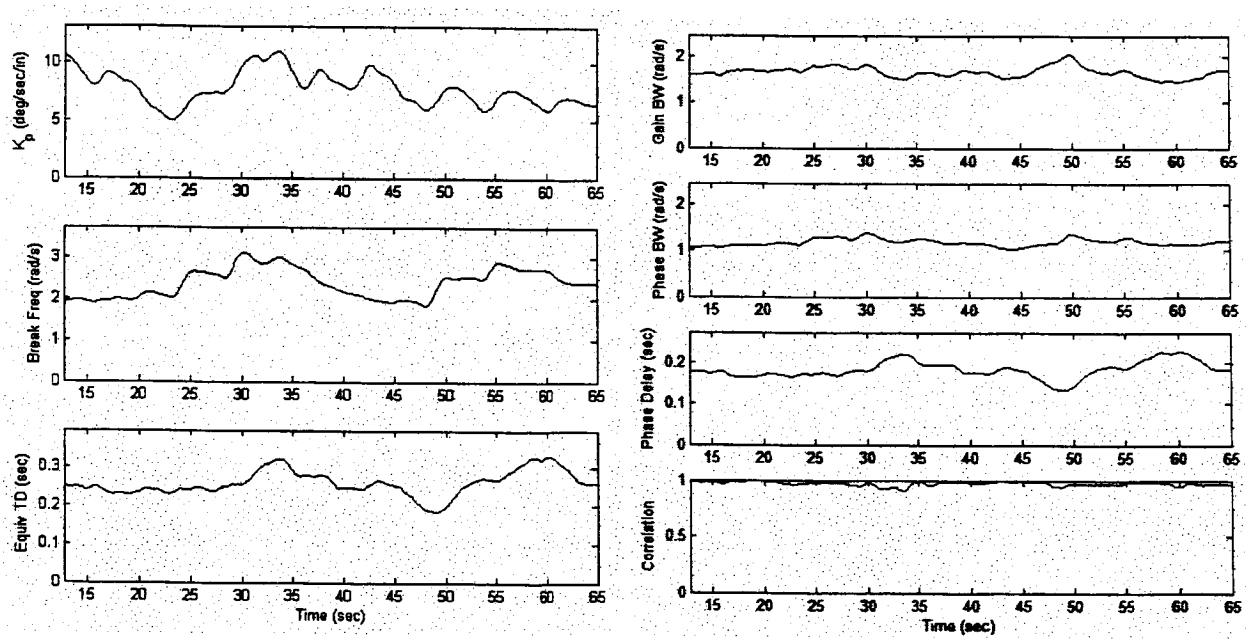
For the roll axis cases the WERA method assumes a first order model with an equivalent time delay. The time varying model parameters for R77 are given in Figure 75a. Time varying bandwidth parameters are then computed from the frequency response generated from the identified model. The results which compare favorably with the TVTF are shown in Figure 75b.



a) P/DLAT Fit to TVTF at 32.6 sec

b) Time Varying Airplane Bandwidth

Figure 74. R77 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 75. R77 WERA Analysis

6.2.2.3 Analysis of Mild PIO Run

The lateral stick position, roll attitude time histories for the R122 BACH are shown in Figure 76. Also included in the figure is a time history of the roll rate feedback gain. Note that at approximately 48 seconds, the feedback augmentation is removed. Following the FCS failure, increased overshoots and oscillations are seen in the roll attitude response, particularly around 80 seconds. After an initial large overshoot following the failure, the pilot copes well, and only mild PIO characteristics are observed. The primary effect is to degrade task performance.

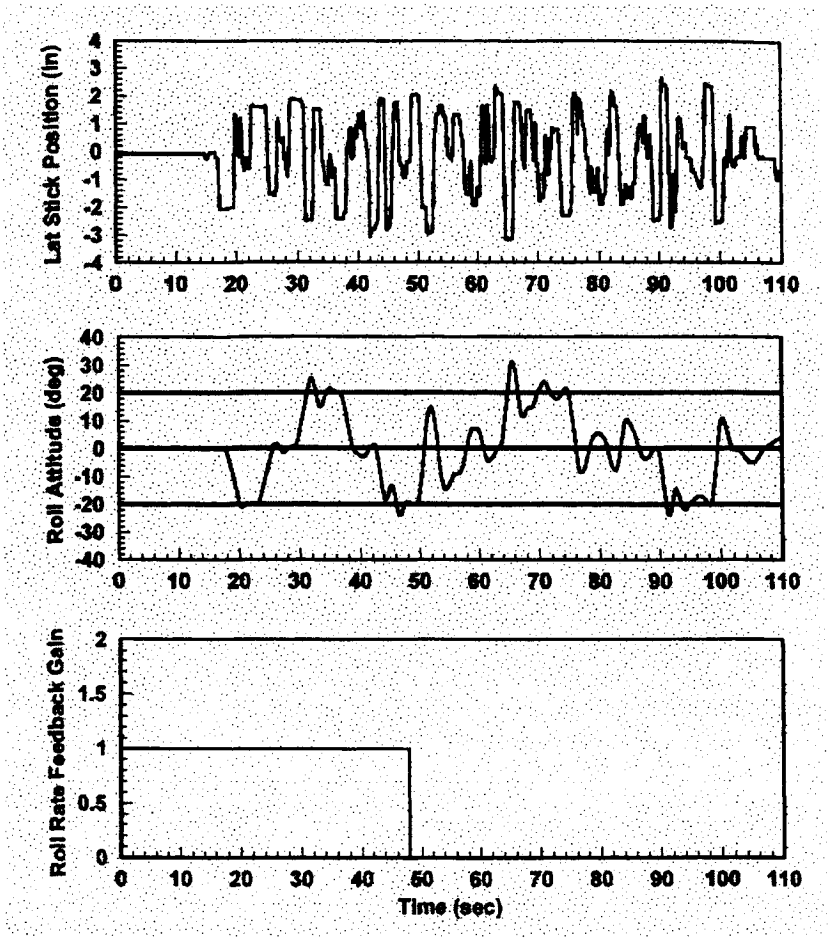
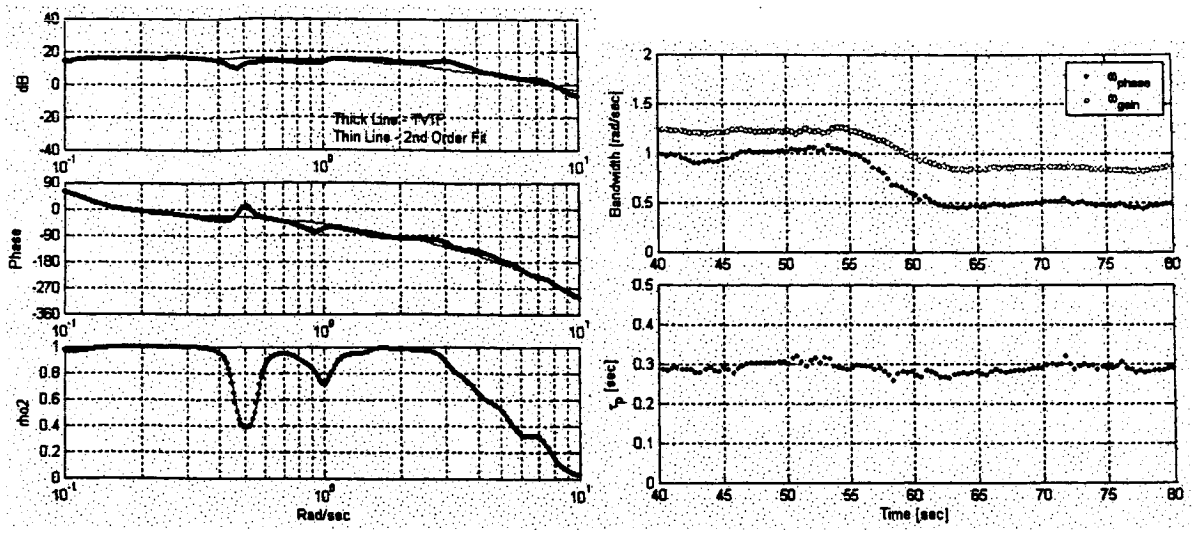


Figure 76. R122 BACH, Mild PIO

TVTF analysis results of this run are shown in Figure 77. In Figure 77a a roll rate to lateral stick position TVTF frequency response is shown for a time of 57.3 sec into the run. With the exception of the dip at 5 rad/sec, the coherence is high in the frequency range associated with pilot control (i.e, the range in which the airplane bandwidth parameters are computed). The 2nd order plus equivalent time delay fit to the TVTF is represented by the thin line. The corresponding airplane bandwidth parameters that were computed from the fit are provided in Figure 77b. The primary effect of the roll rate FCS failure is to reduce the system bandwidth. As indicated in Table 10 this run featured the added 240 msec of roll stick time delay for the entire run. This added time delay produces the high phase delay values that remain essentially constant, even after the FCS failure. This result is not unexpected as significant phase delay changes are not expected with failures that reduce the amount of augmentation in the system (i.e., changes that make the system more like a bare airframe). In the end the system is by around 80 seconds characterized by a low phase bandwidth with a corresponding large phase delay that results in the mild PIO observed in the time responses.

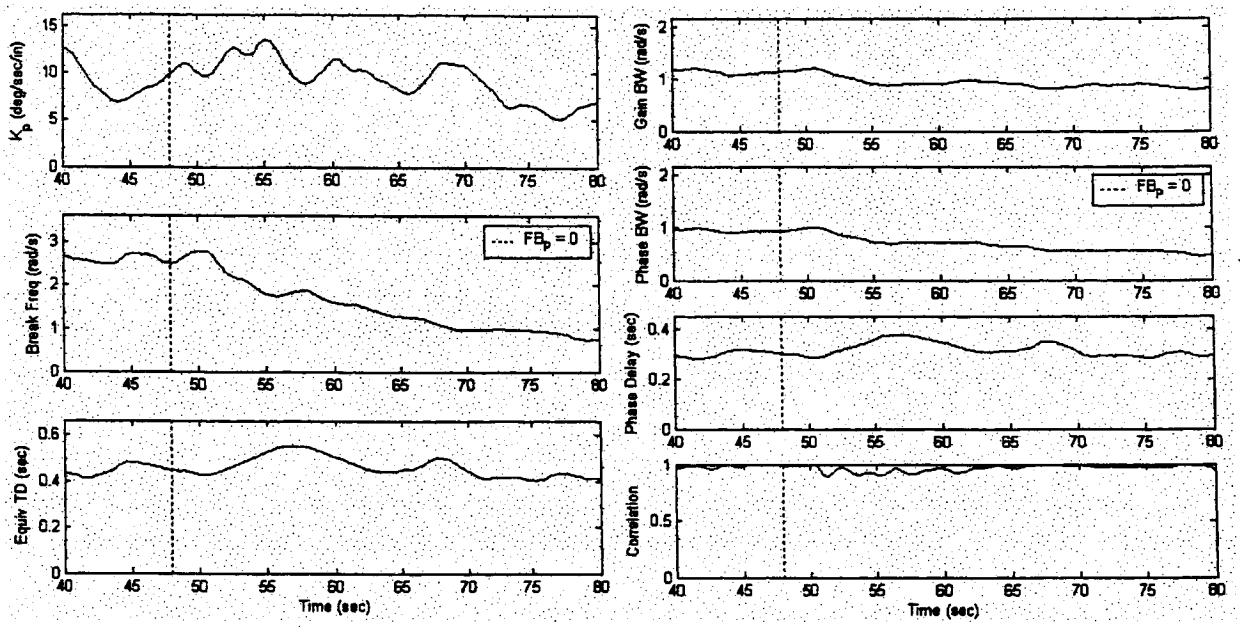
The corresponding WERA results for this run are shown in Figure 78. Figure 78a displays the time varying model parameters used to generate the frequency response from which the bandwidth parameters are computed. In this example, the equivalent time delay remains fairly constant, while the gain and break frequency wander. The bandwidth results seen in Figure 78b are in good agreement with the TVTF results. Here though the detection of the FCS failure is more rapid.



a) P/DLAT Fit to TVTF at 57.3 sec

b) Time Varying Airplane Bandwidth

Figure 77. R122 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 78. R122 WERA Analysis

6.2.2.4 Analysis of Sustained or Severe PIO Run

Time histories for the R29 severe PIO example are shown in Figure 79. This run was characterized by two FCS failures. First, the roll rate augmentation is removed just after 40 seconds. This produces some increased overshoots, but no sustained oscillations. At just after 80 seconds, the roll stick time delay is added. This results in multiple overshoots of nearly double the target attitude. Thus, even though the oscillations are not sustained, the magnitude of the oscillations is severe, and prohibits the pilot from satisfactorily completing the capture and hold task.

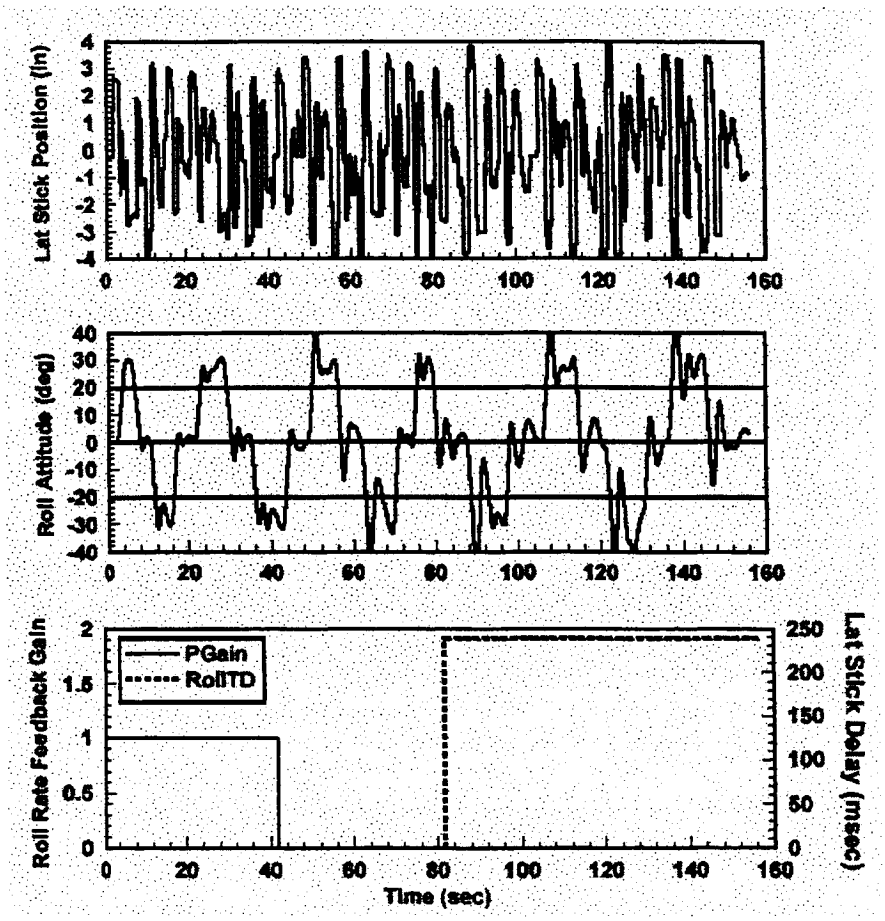
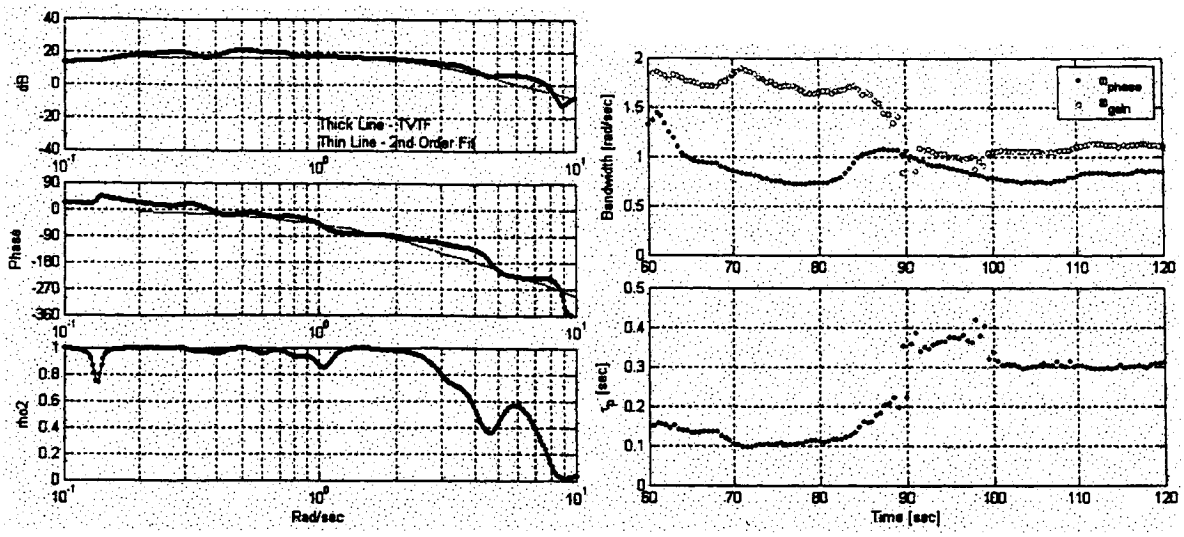


Figure 79. R29 BACH, Sustained or Severe PIO

A snapshot of the TVTF frequency response with the corresponding fit at a time of 95 sec is displayed in Figure 80a. The corresponding bandwidth parameters are shown in Figure 80b. Although there is a reduction in bandwidth associated with the roll rate feedback failure, it is the added stick delay that degrades the system to the extent that a severe PIO results. The effect of the added stick delay is to reduce the gain bandwidth (although the phase bandwidth remains the lower of the two) and to significantly increase the phase delay. The rapid rise in phase delay corresponds with the large roll attitude overshoot seen at 85 sec.

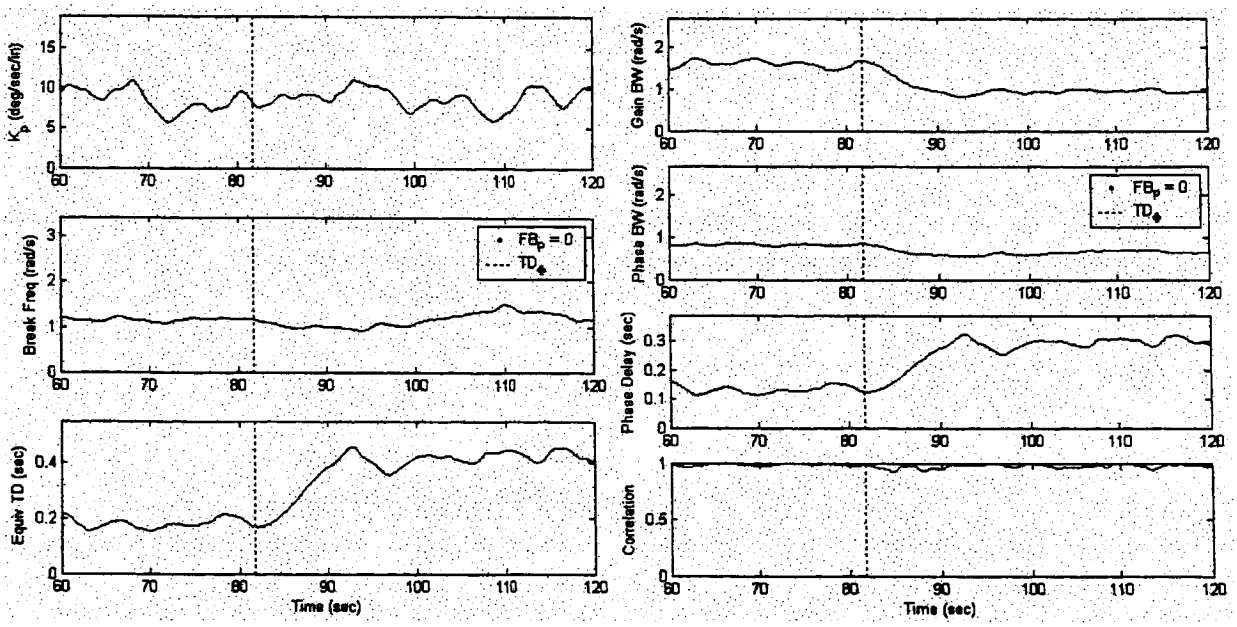
The WERA analysis results are presented in Figure 81 and once again excellent agreement is seen with the TVTF results. In contrast to the R122 results, in this case the gain and break frequency parameters remain fairly constant, while the significant model variation is seen in the equivalent time delay. Of course, this is the expected result when the system change is associated with the insertion of a large pure time delay. In this example, the detection times are again similar, although the TVTF captures the effect of the added delay more rapidly.



a) P/DLAT Fit to TVTF at 95 sec

b) Time Varying Airplane Bandwidth

Figure 80. R29 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 81. R29 WERA Analysis

6.2.3 Pitch Attitude Capture and Hold (PACH) Evaluations

6.2.3.1 Selected Evaluation Runs

A log of the selected PACH evaluation runs is provided in Table 11. The table features the same format used for the complete run log that is provided in the Volume II report. The flight condition, aircraft configuration, and flight control system failure acronyms correspond to those defined previously in Table 6, Table 7, and Table 8, respectively. For the PACH runs both the pitch rate feedback failure and a reduction in the pitch rate command rate limit produced PIO, especially when the added pitch stick time delay was present from the start of the run.

Table 11. Selected PACH Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R68	CM	CLA	PACH	n/a	No change
Mild PIO Runs					
R95	CM	CLA	PACH	QLIM	Oscillations after failure
R96	CM	CLA	PACH	QLIM	PitchTD whole run, some PIO
R115	CH	CLA	PACH	QGain	Three cycle PIO following failure
R117	CH	CLA	PACH	QLIM	PitchTD whole run, small amplitude but sustained PIO
Sustained or Severe PIO Runs					
R38	CM	CLA	PACH	QGain	PitchTD from start, autothrottle disengage triggers sustained PIO
R69	CM	CLA	PACH	QGain	PitchTD whole run, PIO
R94	CM	CLA	PACH	QGain	PitchTD whole run, significant PIO after failure
R116	CH	CLA	PACH	QGain	PitchTD whole run, PIO

6.2.3.2 Analysis of No PIO Run

Longitudinal stick position and pitch attitude time histories from the R68 no PIO case are shown in Figure 82. Here again the solid lines on the pitch attitude plot refer to the target attitude for the capture and hold task. With the baseline aircraft, the pilot easily performed the task within the desired performance requirements (see complete task description in Volume II of this report).

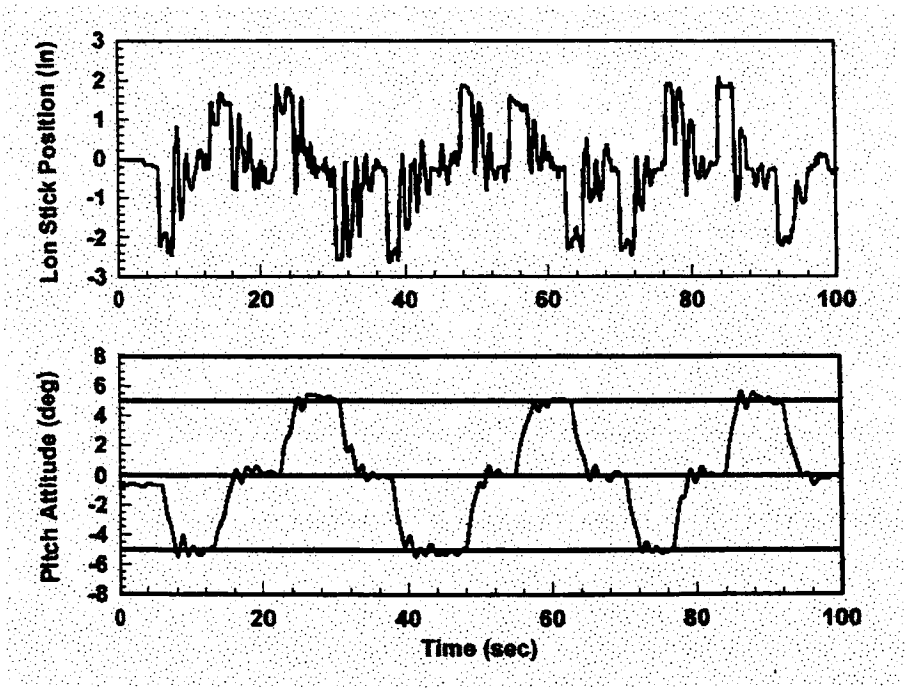
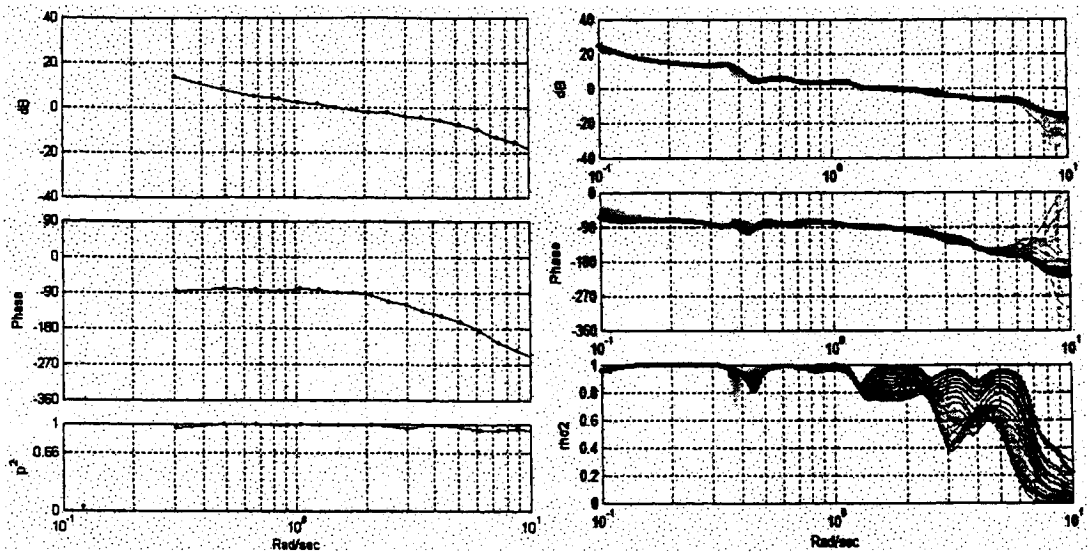


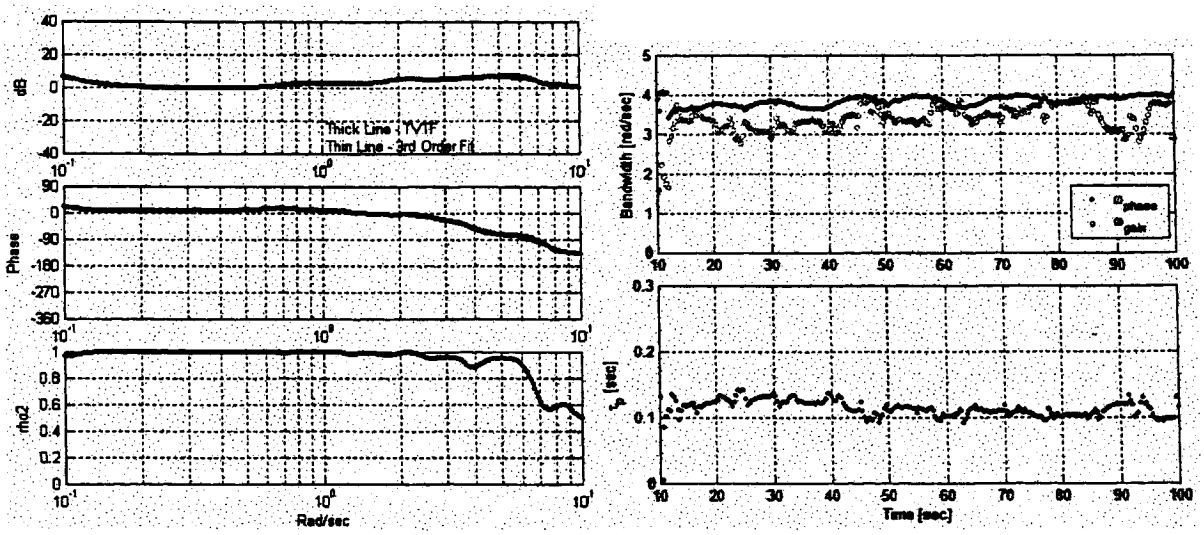
Figure 82. R68 PACH, No PIO

Since there are no changes introduced, comparisons between the TVTF estimation and the traditional FFT can be made. The results that are shown in Figure 83 indicate good agreement between the two methods. One change of note from the roll axis results discussed previously (see Figure 73) is the improved coherence observed in this example. Specifically, the high coherence frequency region extends beyond that seen in the roll axis. A TVTF snapshot at 55.1 sec with a corresponding 3rd order including an equivalent time delay is displayed in Figure 84a. The fits to the complete TVTF are then used to compute the time varying airplane bandwidth parameters shown in Figure 84b. The corresponding WERA results are provided in Figure 85. There is good agreement between the two methods for the phase bandwidth and phase delay parameters. For this example, however, the gain bandwidth computed from the WERA frequency response tends to oscillate over the length of the run.



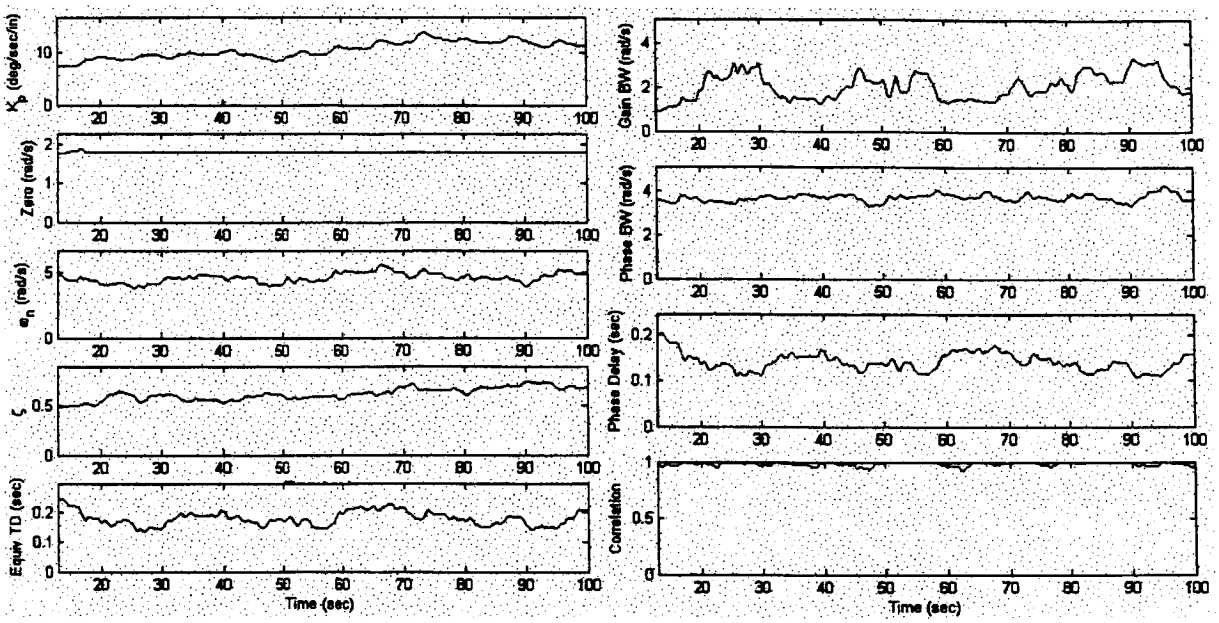
a) FFT b) TVTF at 55.1 sec

Figure 83. R68 THE/DLON Transfer Function Estimation



a) Q/DLON Fit to TVTF at 55.1 sec b) Time Varying Airplane Bandwidth

Figure 84. R68 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters b) Time Varying Airplane Bandwidth

Figure 85. R68 WERA Analysis

6.2.3.3 Analysis of Mild PIO Run

Pitch axis time histories for the R115 mild PIO run are shown in Figure 86. For this run the pitch rate feedback was removed just before the 45 sec point. The failure produced a mild PIO on the next capture that was characterized by a 3 cycle damped oscillation. Smaller magnitude oscillations are seen on the subsequent capture.

Results of the TVTF analysis for this run are shown in Figure 87. The TVTF snapshot of Figure 87a displays good coherence for the primary frequency range of interest and an excellent fit. The corresponding time varying bandwidth parameters are given in Figure 87b. As was observed with the roll rate feedback failure, the primary effect of the change in pitch rate feedback is to reduce the bandwidth. Elimination of the augmentation produces no significant change in phase delay, a result verified by the analysis of the frequency sweep runs (see Volume II) with and without the pitch rate feedback. There is a slight increase in the gain bandwidth that coincides with a slight dip in the phase delay following the failure. It is the reduction in phase bandwidth, however, that provides the indicator of the degraded characteristics of the aircraft configuration.

Results from the R115 WERA analysis are displayed in Figure 88. For this example the equivalent time delay (see Figure 88a) continues to increase following the failure, while the other model parameters remain more or less constant. This produces the steadily increasing phase delay of Figure 88b. The gain bandwidth parameter also appears to be too low, even prior to the failure. The apparent inconsistency of these results, at least following the failure, should be expected given that the correlation parameter (see Figure 88b) falls well below the desired value of 1 beyond 46 sec.

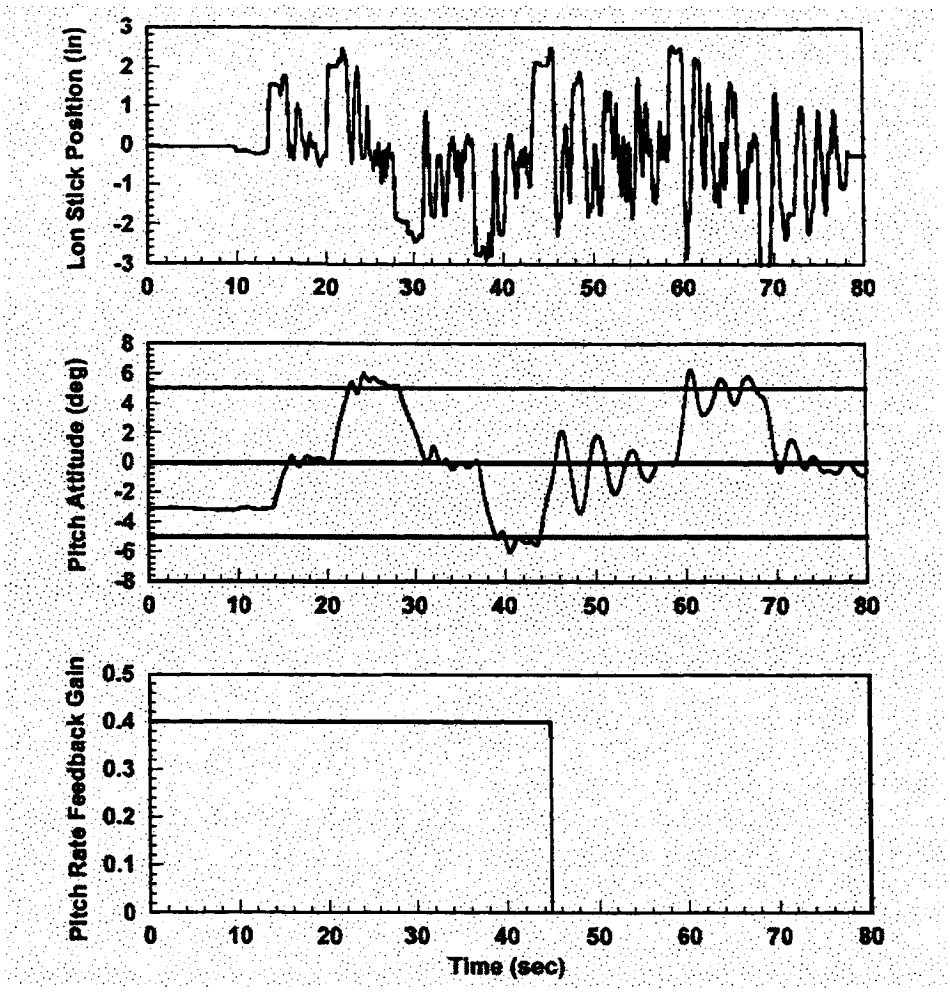
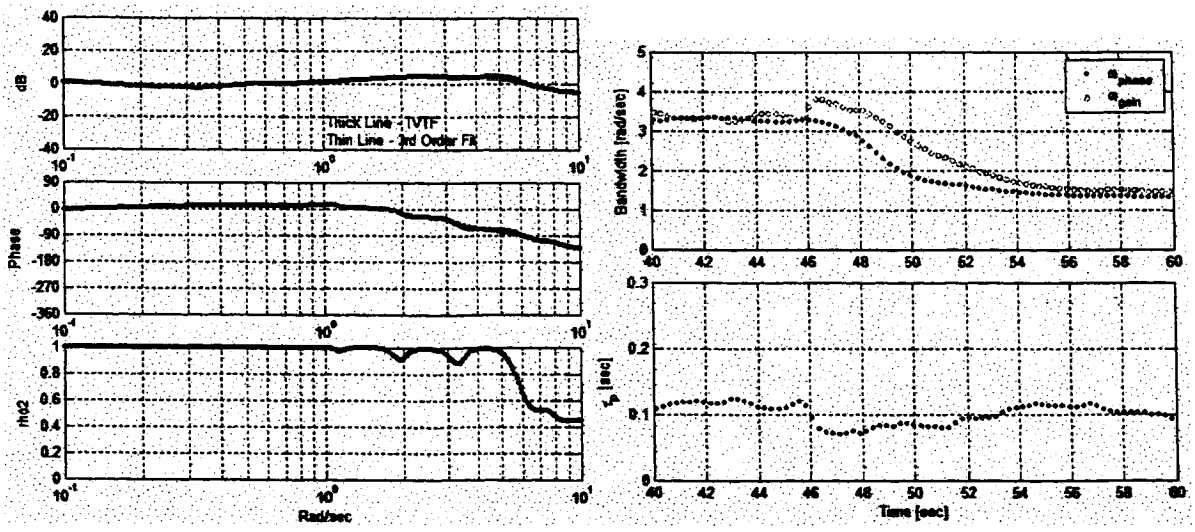


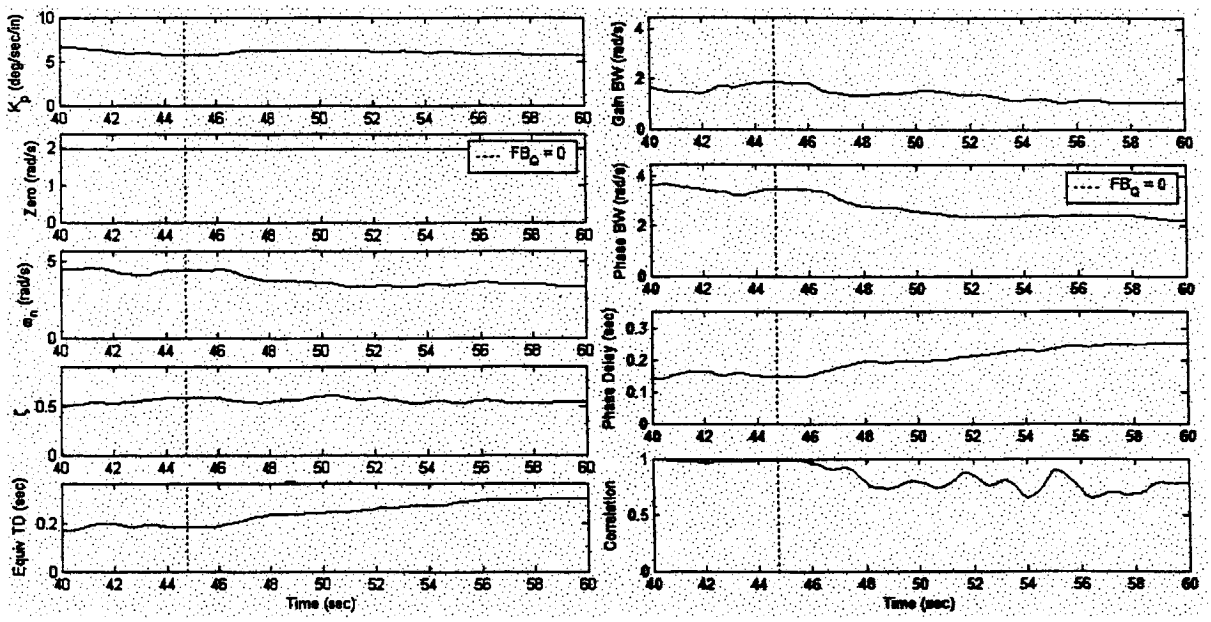
Figure 86. R115 PACH, Mild PIO



a) Q/DLON Fit to TVTF at 46.3 sec

b) Time Varying Airplane Bandwidth

Figure 87. R115 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters b) Time Varying Airplane Bandwidth

Figure 88. R115 WERA Analysis

6.2.3.4 Analysis of Sustained or Severe PIO Run

Time histories for the R69 sustained or severe PIO run are found in Figure 89. This run featured the added stick time delay for the entire run length and a failed pitch rate feedback at just under 24 seconds. Following the failure there are sustained oscillations that dominate each of the subsequent capture attempts. Although the oscillations are damped they do prevent the pilot from completing the task.

In Figure 90 the TVTF analysis results of this severe PIO case are given. Once again the third order plus equivalent time delay fit provides an excellent match to the TVTF frequency response following the failure at the 27.2 sec mark in the run (see Figure 90a). Just as with the mild PIO run, the primary effect of the failure is to significantly reduce the phase bandwidth as shown in Figure 90b. On the other hand, the gain bandwidth remains essentially unchanged, while there is a slight reduction in the corresponding phase delay.

WERA results for this run are given in Figure 91. As was the case with the other two pitch cases discussed thus far, the gain bandwidth parameter appears to be too low in value throughout the run. The phase bandwidth, on the other hand, agrees well with the TVTF results as does the phase delay parameter with the exception of the region following the failure where the correlation measure dips.

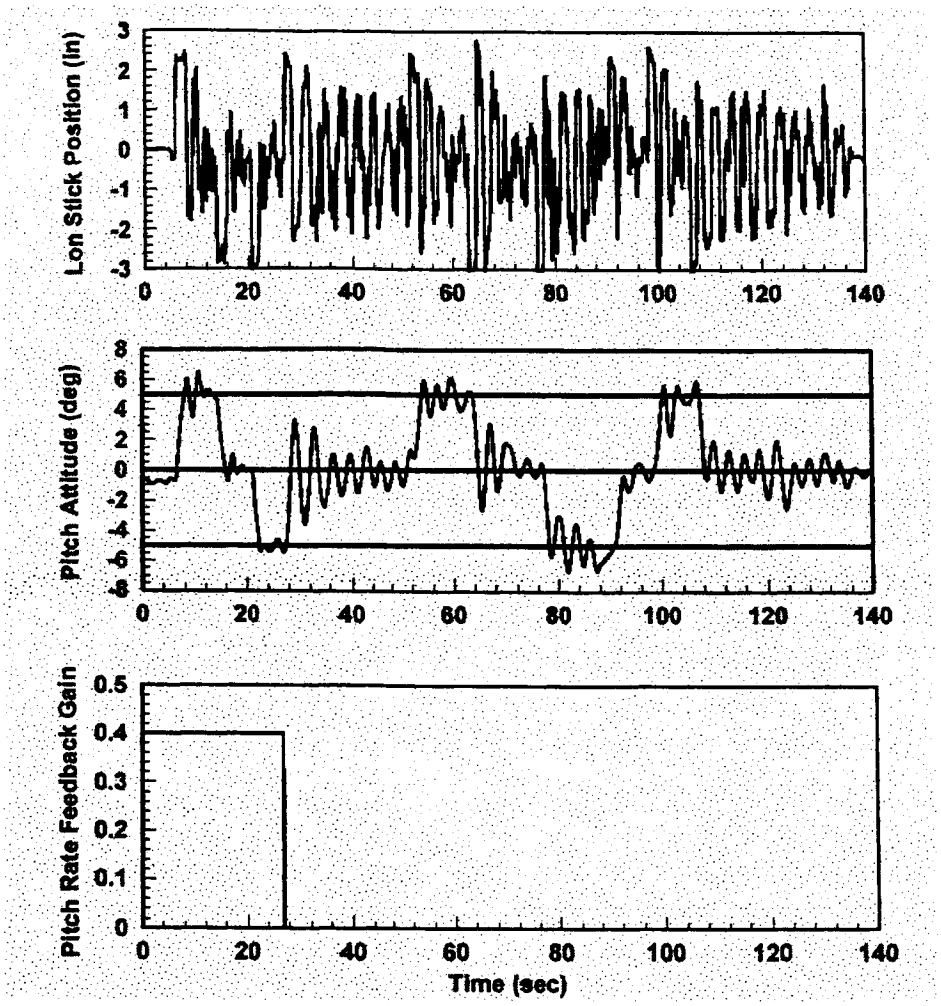
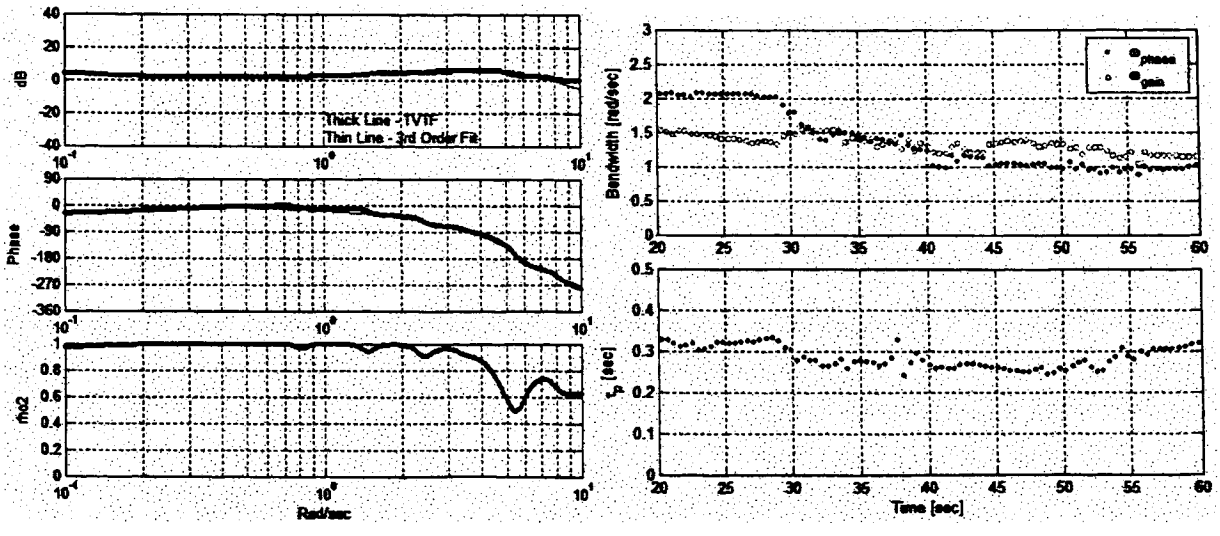


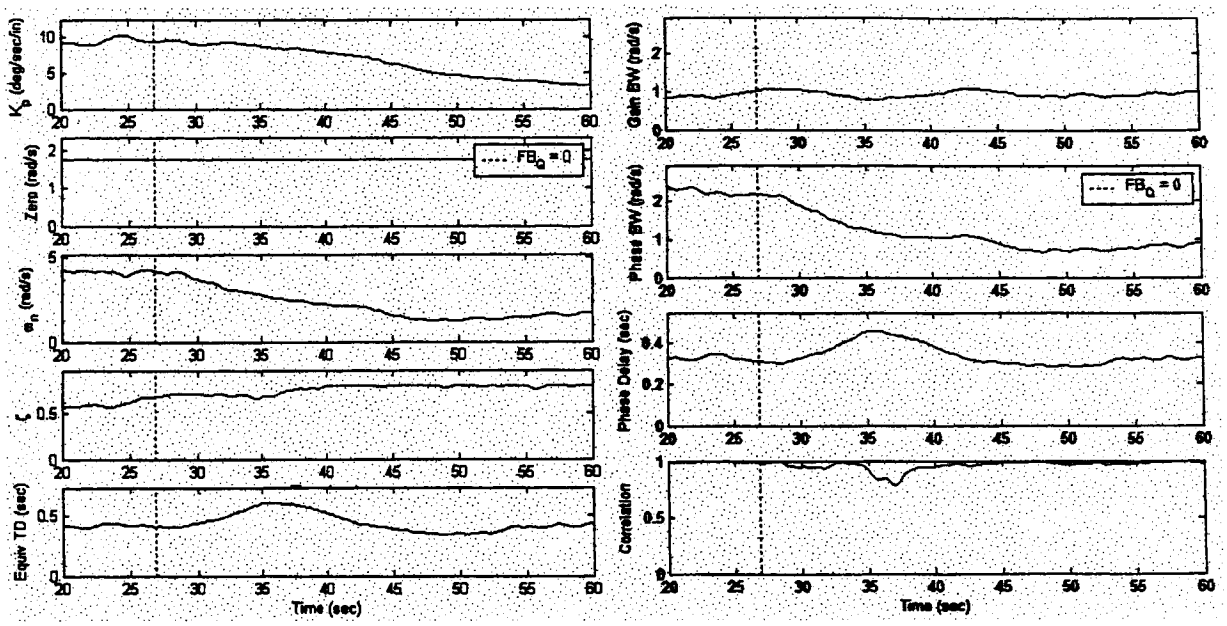
Figure 89. R69 PACH, Sustained or Severe PIO



a) Q/DLON Fit to TVTF at 27.2 sec

b) Time Varying Airplane Bandwidth

Figure 90. R69 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters b) Time Varying Airplane Bandwidth

Figure 91. R69 WERA Analysis

6.2.4 Sum-of-Sines Tracking (SOS) Evaluations

6.2.4.1 Selected Evaluation Runs

A log of the selected SOS evaluation runs is provided in Table 12. The table features the same format used for the complete run log that is provided in the Volume II report. The flight condition, aircraft configuration, and flight control system failure acronyms correspond to those defined previously in Table 6, Table 7, and Table 8, respectively. Selected runs feature two no PIO cases, one pitch axis and one roll axis. Two corresponding sustained or severe PIO cases are also included. Note that the added stick time delays were included over the entire run lengths for the PIO cases.

Table 12. Selected SOS Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R118	CH	CLA	SOS	n/a	Pitch axis only
R123	CH	CLA	SOS	n/a	Roll axis only
Sustained or Severe PIO Runs					
R119	CH	CLA	SOS	QGain	Pitch axis only, PitchTD whole run, sustained PIO follows failure
R124	CH	CLA	SOS	BGain + PGain	Roll axis only, RollTD whole run, sustained oscillations

6.2.4.2 Analysis of Pitch Axis No PIO Run

Longitudinal stick position and pitch attitude time histories for the R118 no PIO case are given in Figure 92. No significant observations are made here other than to say the pilot was able to perform the task well with the baseline aircraft.

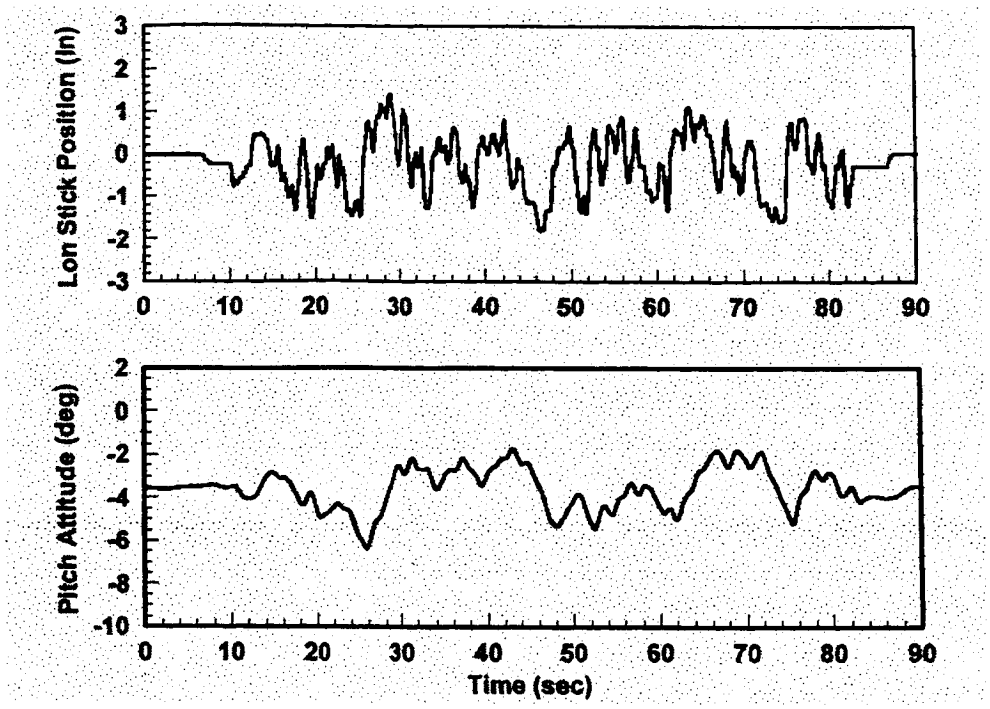
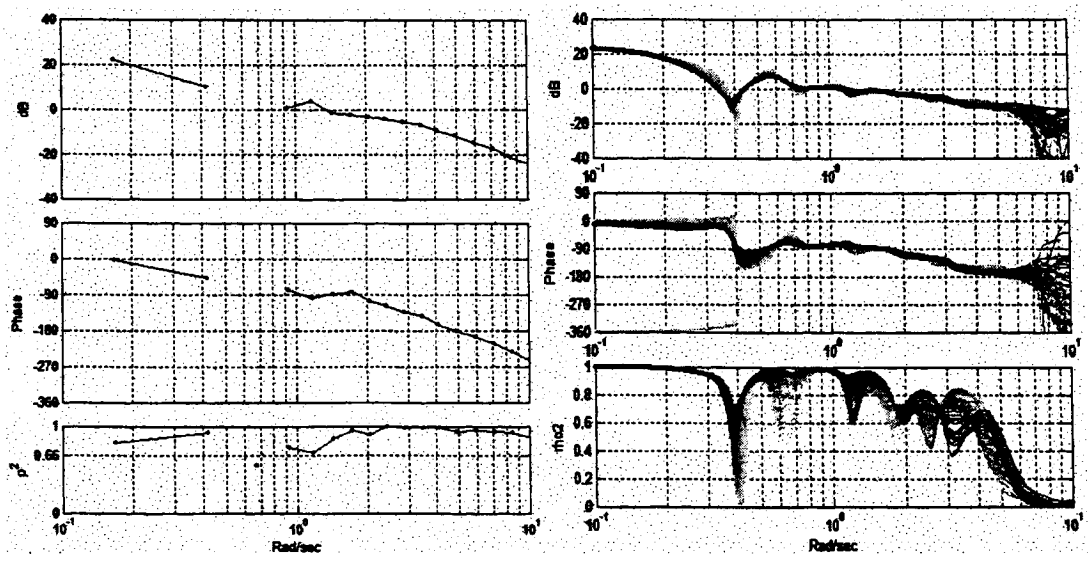


Figure 92. R118 Pitch Axis SOS, No PIO

FFT and TVTF comparisons for R118 are provided below in Figure 93. Here again the two techniques compare favorably in the frequency range of interest. Note that both frequency responses display the dip in coherence in the neighborhood of 0.4 rad/sec. Both responses also indicate a -180 deg crossing at approximately 5 rad/sec and a 0 dB crossing around 1.2 rad/sec.

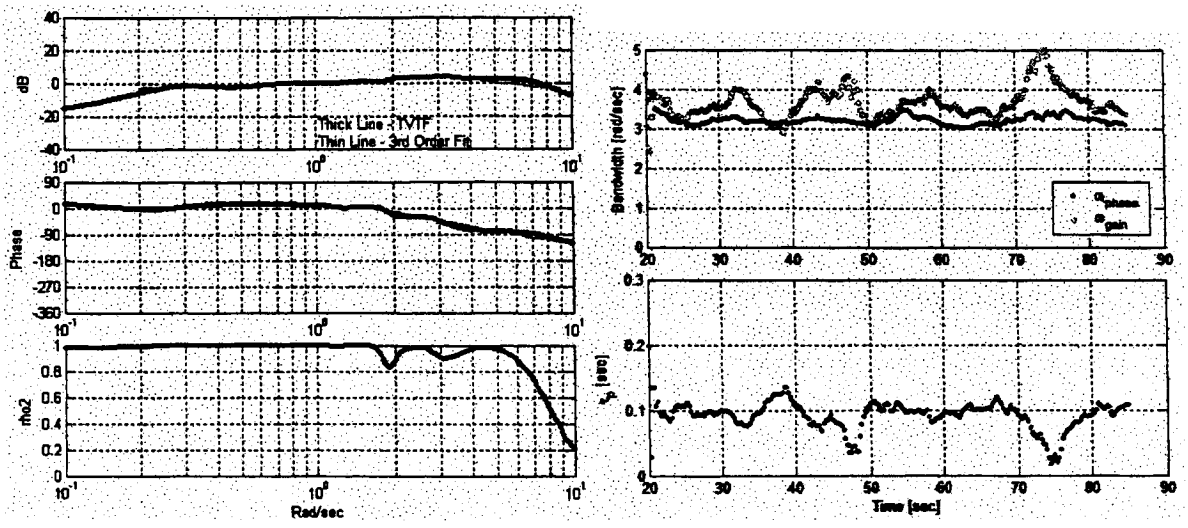


a) FFT b) TVTF at 45.6 sec

Figure 93. R118 THE/DLON Transfer Function Estimation

TVTF results for this run are shown in Figure 94. Although the fit to the TVTF frequency response is good (see Figure 94a), this run appears to suffer somewhat from the drop in coherence above 6 rad/sec. This is most evident in the gain bandwidth and phase delay parameters, both of which are tied to the -180 deg frequency point, that oscillate somewhat about a nominal value. The phase bandwidth parameter that is obtained in the high coherence region, on the other hand, has a much more uniform value.

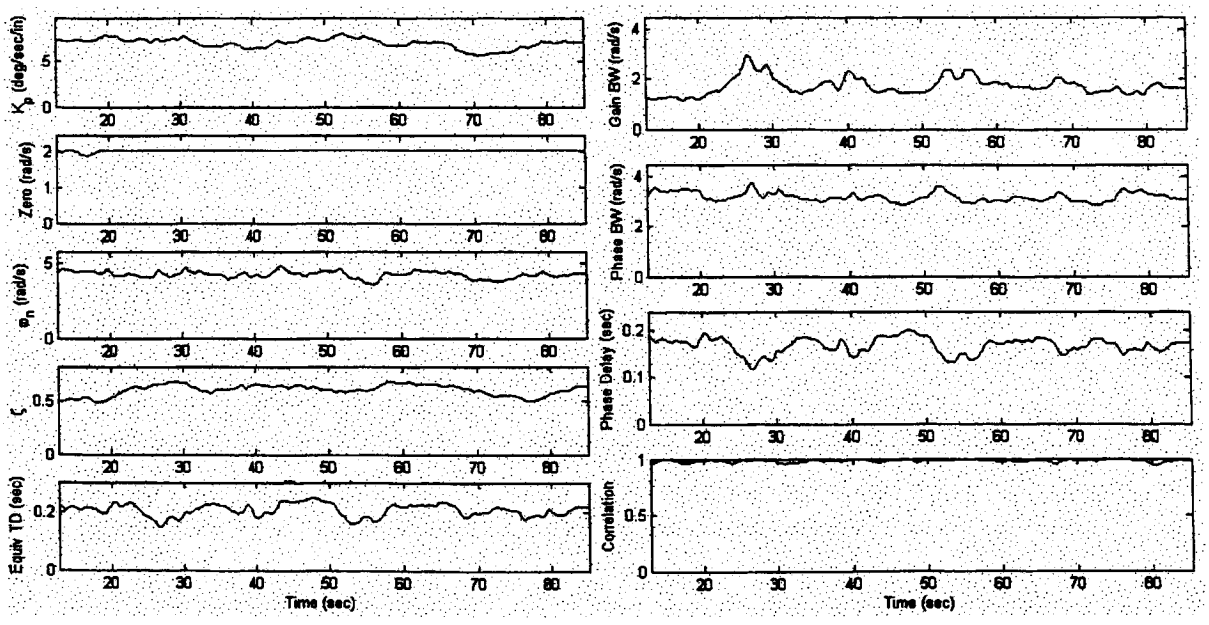
The phase bandwidth results from the WERA analysis are also quite smooth (see Figure 95) and agree well with the TVTF results. The issues previously noted with the gain bandwidth and phase delay parameters are again noted here.



a) Q/DLON Fit to TVTF at 45.6 sec

b) Time Varying Airplane Bandwidth

Figure 94. R118 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 95. R118 WERA Analysis

6.2.4.3 Analysis of Sustained or Severe Pitch Axis PIO Run

Time histories for the sustained or severe PIO example are shown in Figure 96. Note that when the pitch rate feedback fails at just under 48 sec, the pilot enters a sustained PIO and all but abandons the task for the next 70 sec.

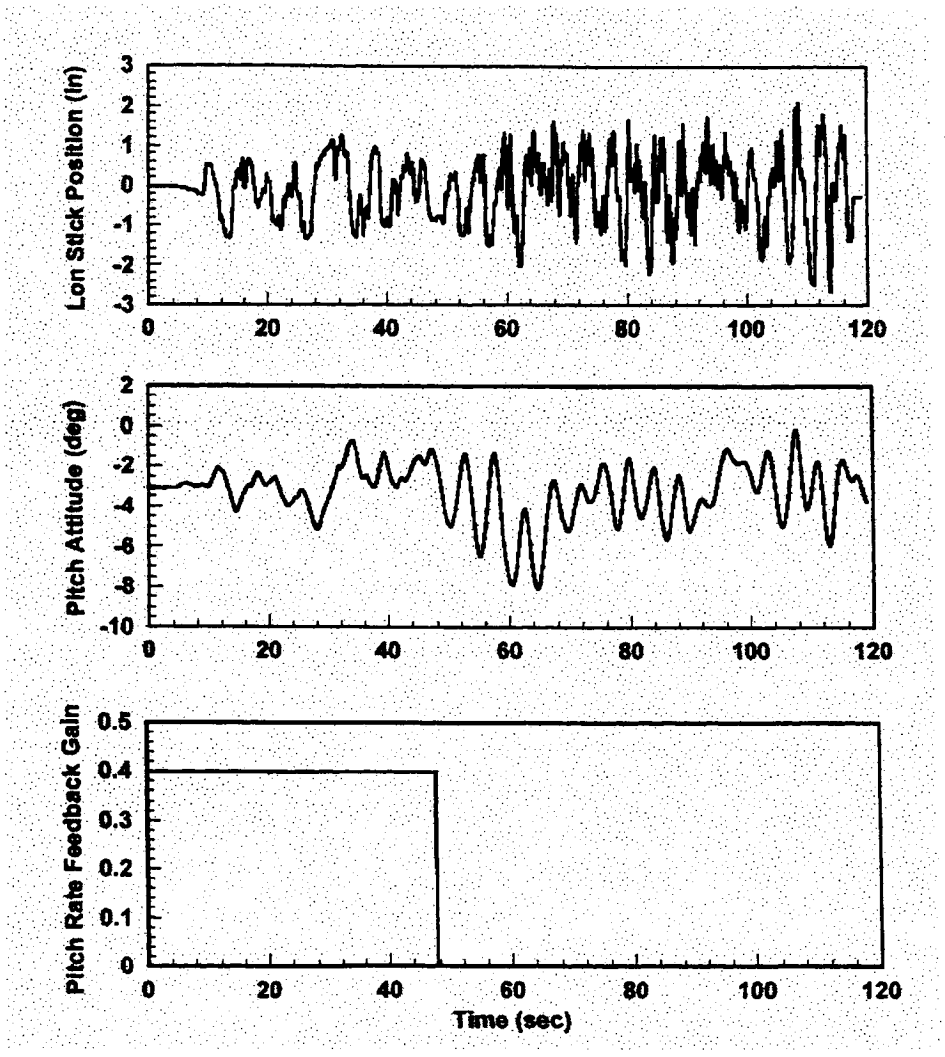
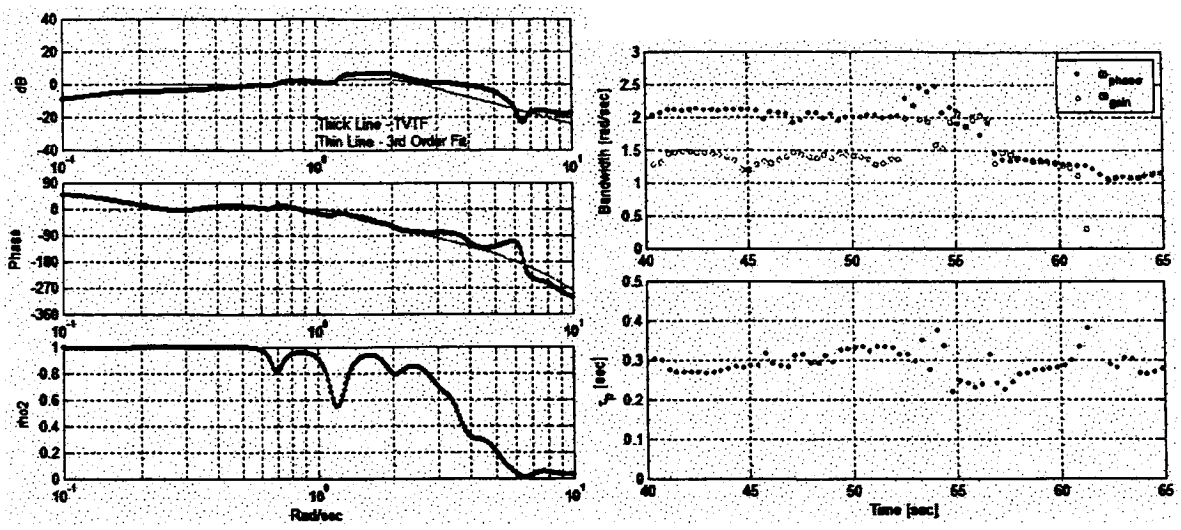


Figure 96. R119 Pitch Axis SOS, Sustained or Severe PIO

The TVTF analysis plots for this run are given in Figure 97. The bandwidth parameters (see Figure 97b) show a region of scatter in the neighborhood of 55 sec. Note that the fit to the TVTF frequency response suffers in this region as seen in Figure 97a. Despite this issue, the parameters overall respond to the trends established in the previous pitch axis PIO examples.

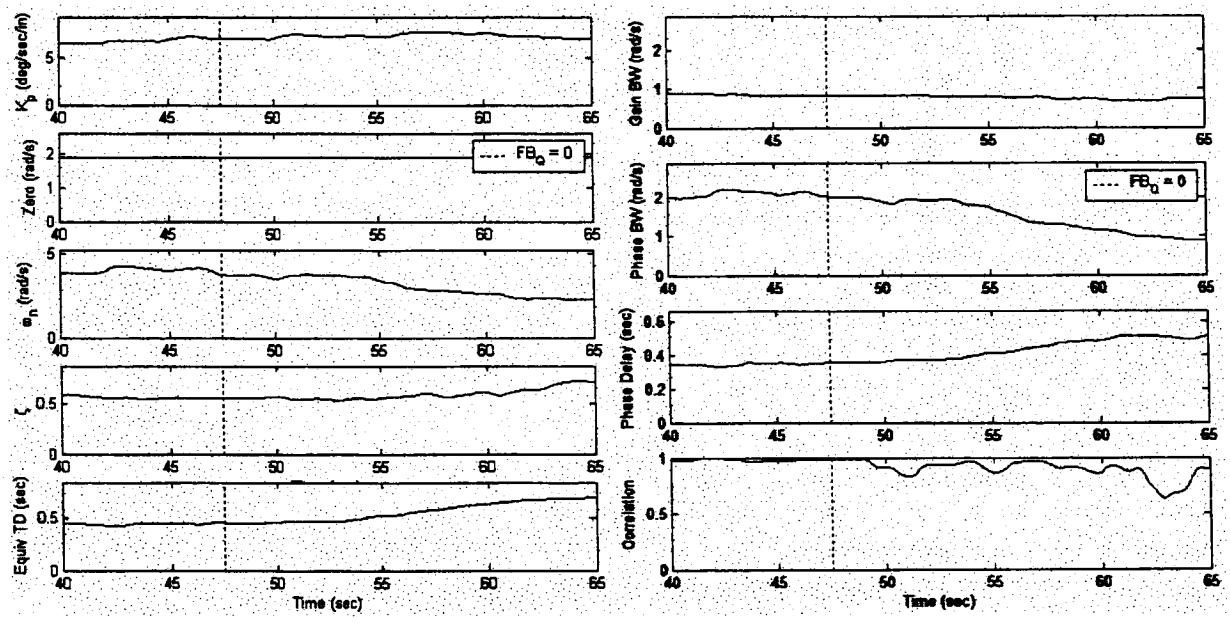
For this example the WERA analysis as shown in Figure 98 follows previously established trends although there is better agreement with the TVTF results. The phase delay parameter also begins a slow and steady rise when the correlation drops below 1 after the 50 sec mark.



a) Q/DLON Fit to TVTF at 55.2 sec

b) Time Varying Airplane Bandwidth

Figure 97. R119 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 98. R119 WERA Analysis

6.2.4.4 Analysis of Roll Axis No PIO Run

Lateral stick position and roll attitude time histories for the R123 no PIO case are given in Figure 99. No significant observations are made here other than to say the pilot was able to perform the task reasonably well with the baseline aircraft. FFT and TVTF comparisons are found in Figure 100. There is good agreement between the two methods including a similar drop in coherence between 1 and 2 rad/sec.

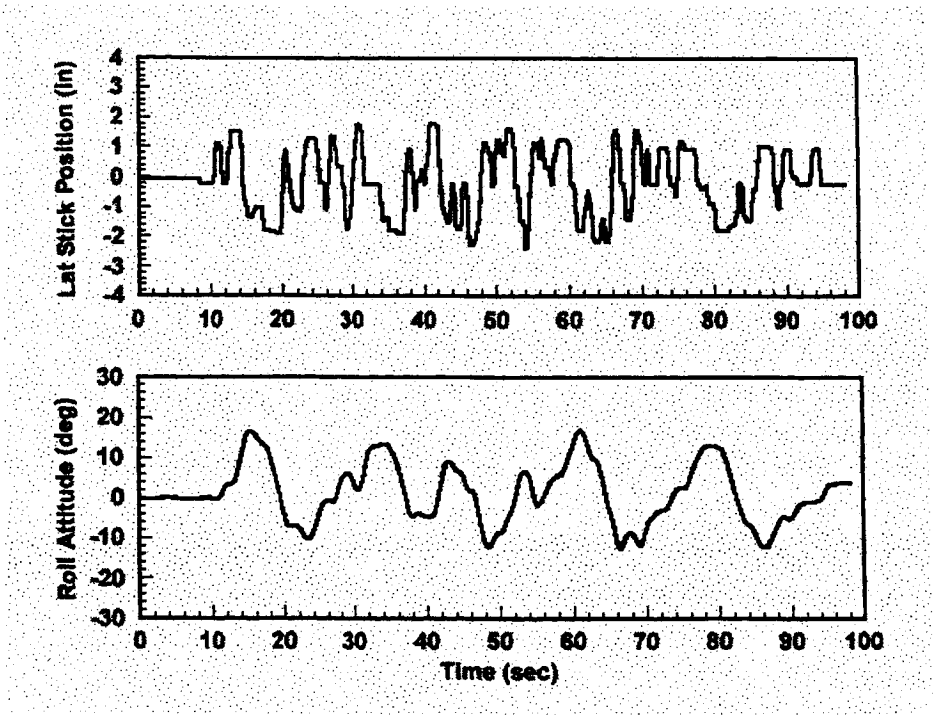
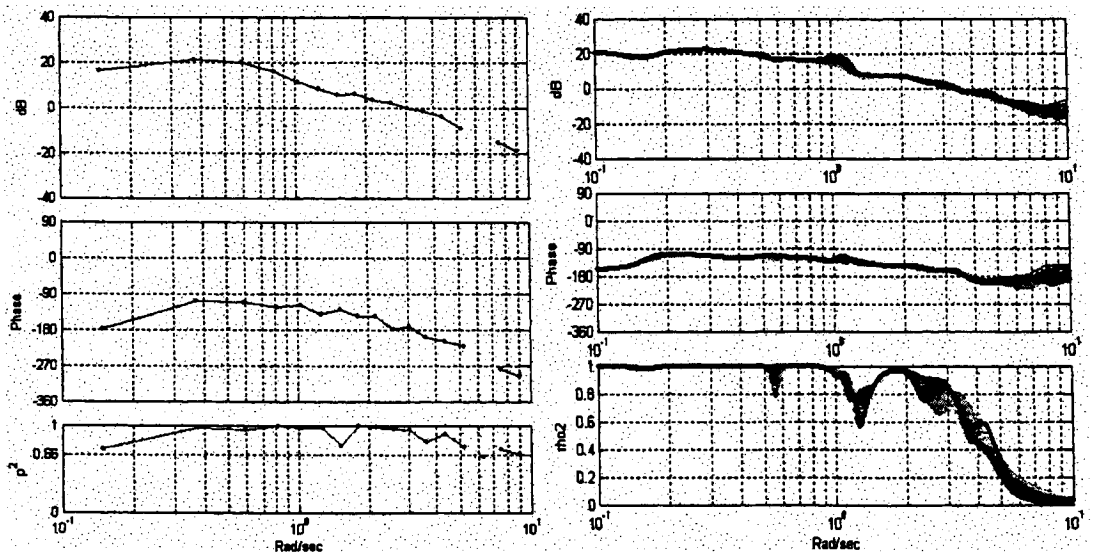


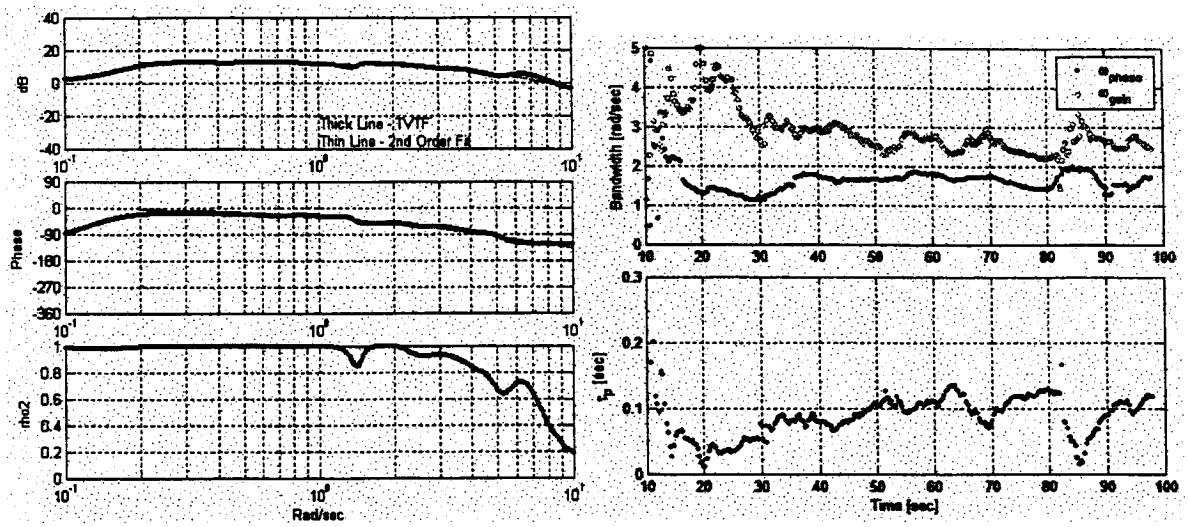
Figure 99. R123 Roll Axis SOS, No PIO



a) FFT b) TVTF at 45.6 sec

Figure 100. R118 THE/DLON Transfer Function Estimation

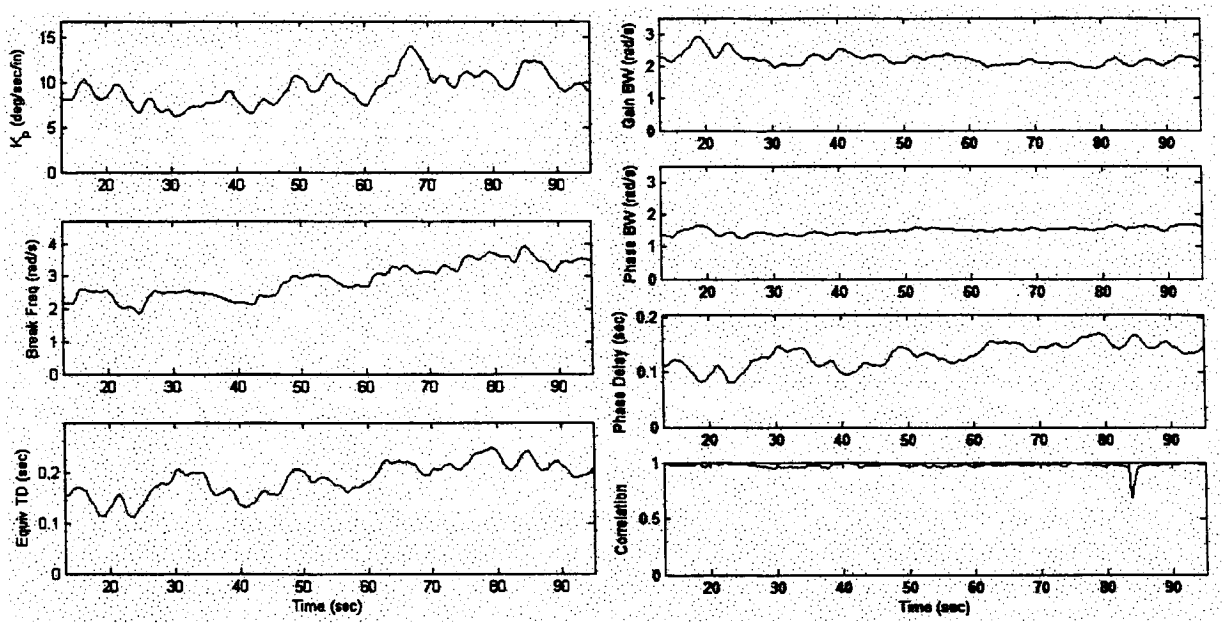
Following the first 30 sec of the run the TVTF provides fairly smooth results as shown in Figure 101. Equivalent results were obtained using the WERA technique (see Figure 102).



a) P/DLAT Fit to TVTF at 45.6 sec

b) Time Varying Airplane Bandwidth

Figure 101. R123 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 102. R123 WERA Analysis

6.2.4.5 Analysis of Roll Axis Sustained or Severe PIO Run

Time histories for the sustained or severe PIO example are shown in Figure 103. Note that when the roll rate feedback fails at just over 41 sec, the pilot enters a sustained PIO and all but abandons the task for the next 70 sec. Both the TVTF (Figure 104) and WERA (Figure 105) results provide similar results, where the primary effect of the FCS failure is to reduce the phase bandwidth.

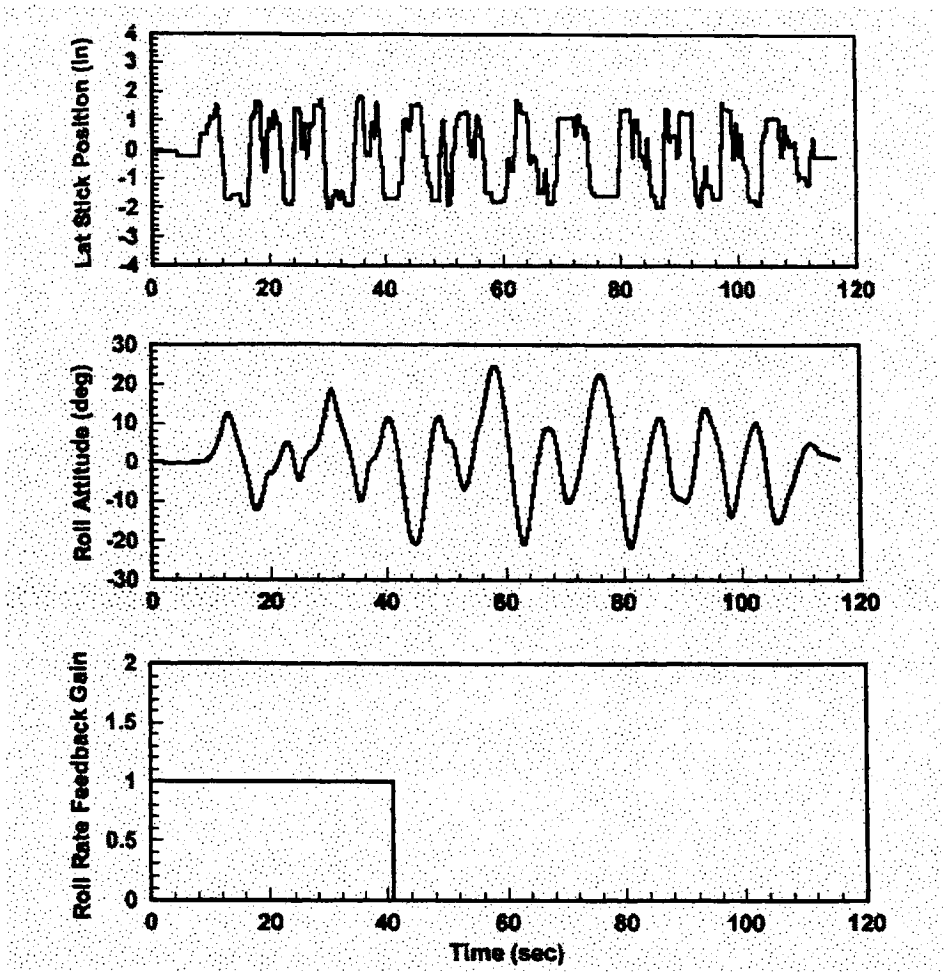
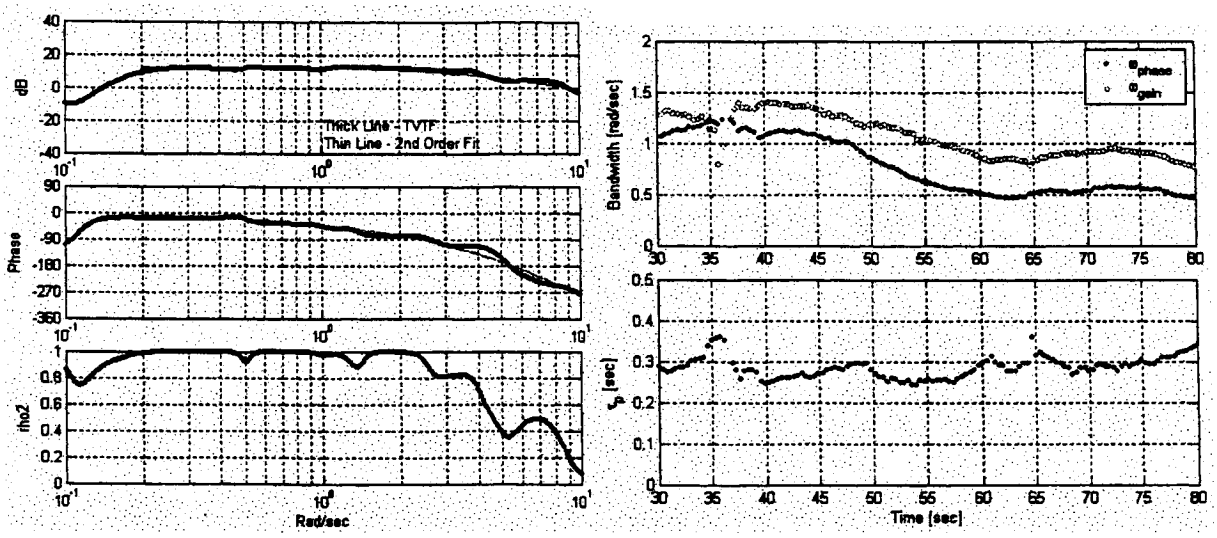


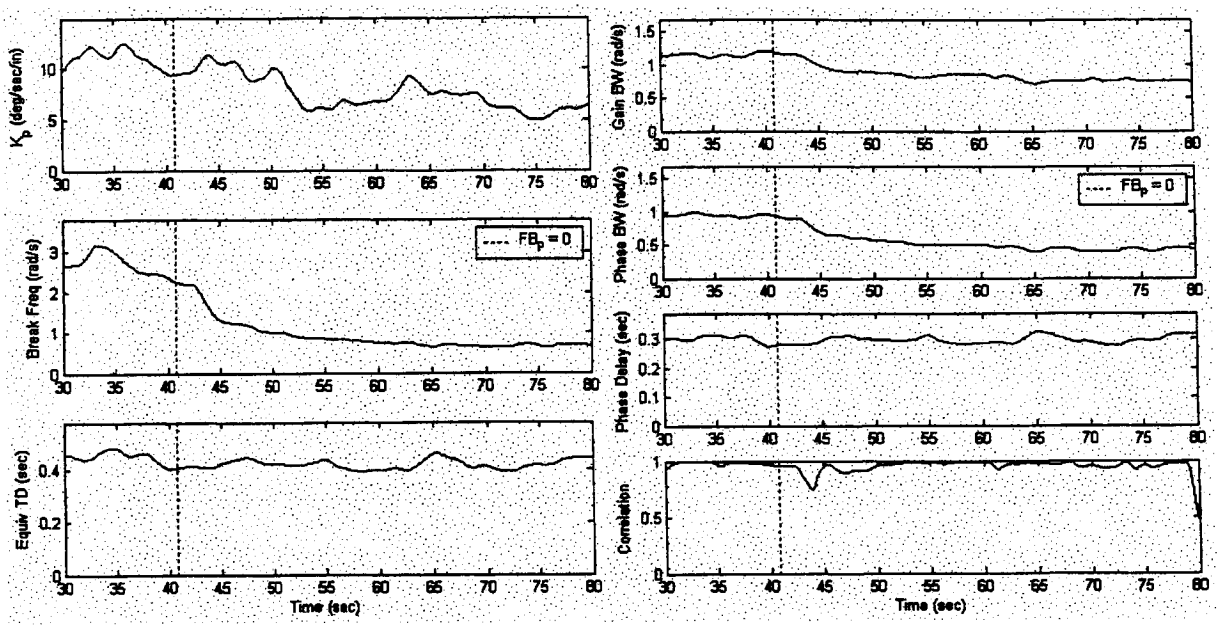
Figure 103. R124 Roll Axis SOS, Sustained or Severe PIO



a) P/DLAT Fit to TVTF at 42.2 sec

b) Time Varying Airplane Bandwidth

Figure 104. R124 TVTF Analysis



a) Time Varying Model Parameters b) Time Varying Airplane Bandwidth

Figure 105. R124 WERA Analysis

6.2.5 Discrete Tracking (DT) Evaluations

6.2.5.1 Selected Evaluation Runs

A log of the selected pitch and roll DT evaluation runs is provided in Table 13. The table features the same format used for the complete run log that is provided in the Volume II report. The flight condition, aircraft configuration, and flight control system failure acronyms correspond to those defined previously in Table 6, Table 7, and Table 8, respectively.

Table 13. Selected DT Evaluation Runs

Run#	FC	Configuration	Task	LOC	Comments
No PIO Runs					
R15	C	CHF	DT	n/a	
R18	C	CHF	DT	PLIM	
Mild PIO Runs					
R70	CM	CLA	DT	QGain + BGain + PGain	Mild pitch PIO
R71	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, pitch PIO
R99	CM	CLA	DT	QGain	Mild pitch PIO
R100	CM	CLA	DT	QGain	PitchTD whole run, large overshoot following failure, mild PIO
Run#	**FC**	**Configuration**	**Task**	**LOC**	**Comments**

R106	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, multi-axis failures
Sustained or Severe PIO Runs					
R128	CH	CLA	DT	BGain + PGain	PitchTD and RollTD whole run, sustained roll PIO

6.2.5.2 Analysis of No PIO Run

Pitch and roll axis time histories for the no PIO run, R15, are provided in Figure 106. Note that the DT task features bank angle captures of nearly 60 deg.

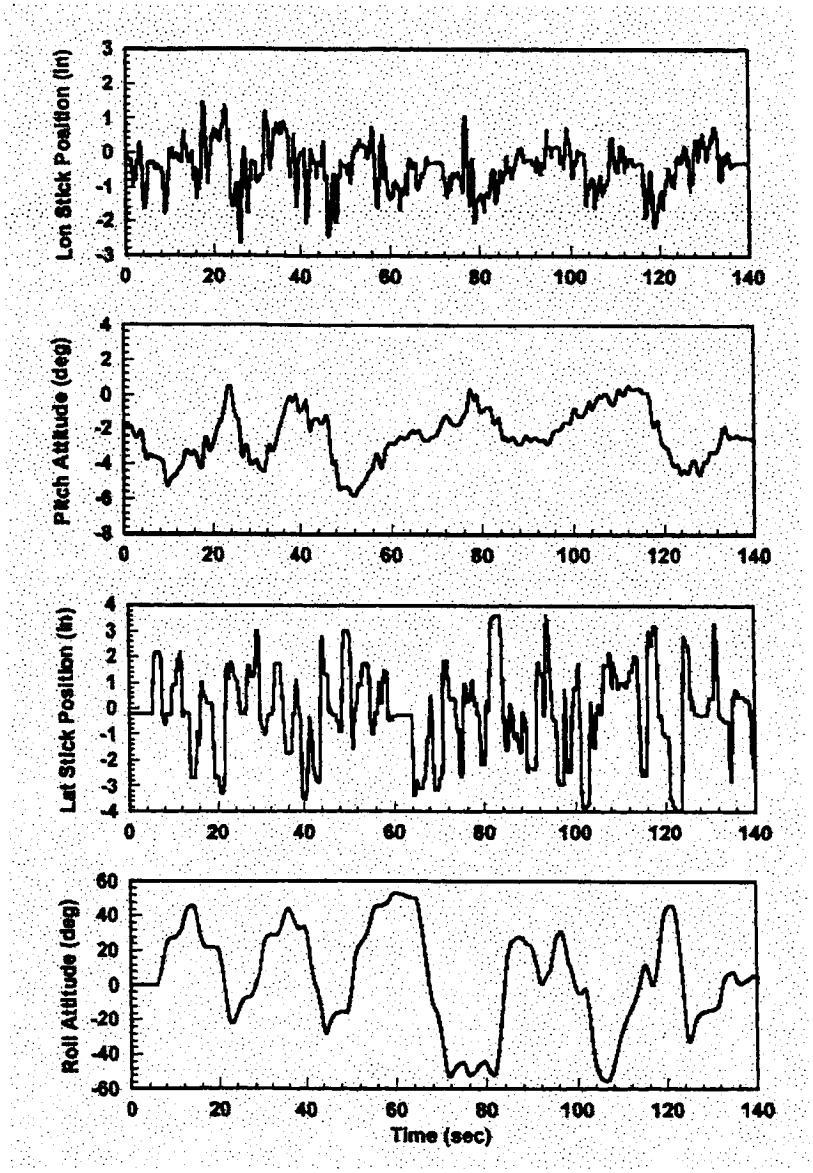
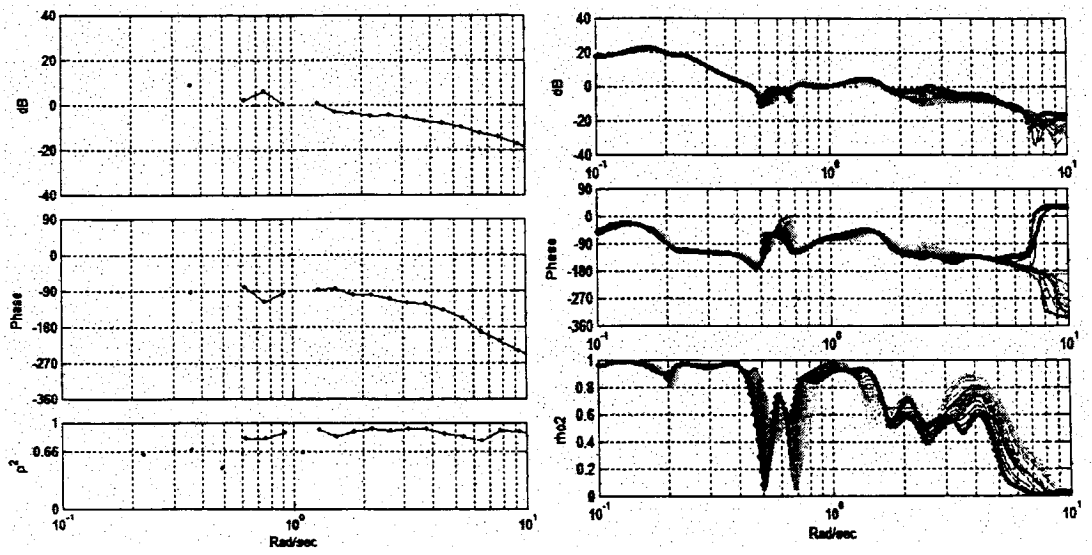


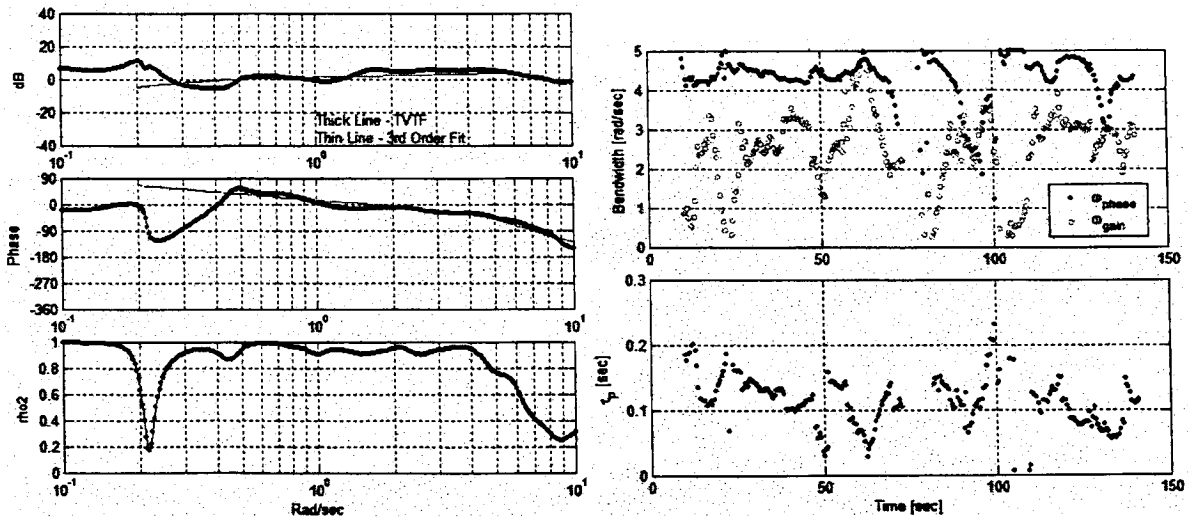
Figure 106. R15 DT, No PIO

As indicated by the FFT and TVTF comparisons (Figure 107) and the subsequent TVTF (Figure 108) and WERA (Figure 109) analysis plots, this run did not yield clean pitch axis results. The TVTF problems are indicated by the significant lack of coherence across the frequency range of interest. Similarly, the WERA analysis is characterized by a significant drop in correlation throughout the run.



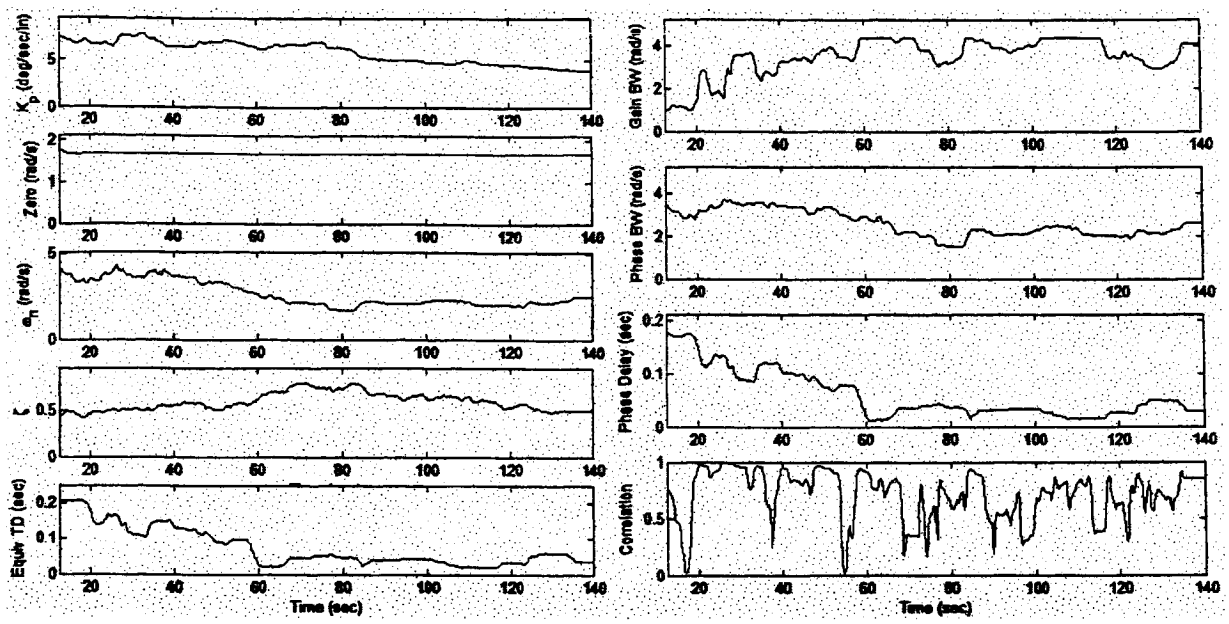
a) FFT b) TVTF at 92.5 sec

Figure 107. R15 THE/DLON Transfer Function Estimation



a) Q/DLON Fit to TVTF at 61.2 sec b) Time Varying Airplane Bandwidth

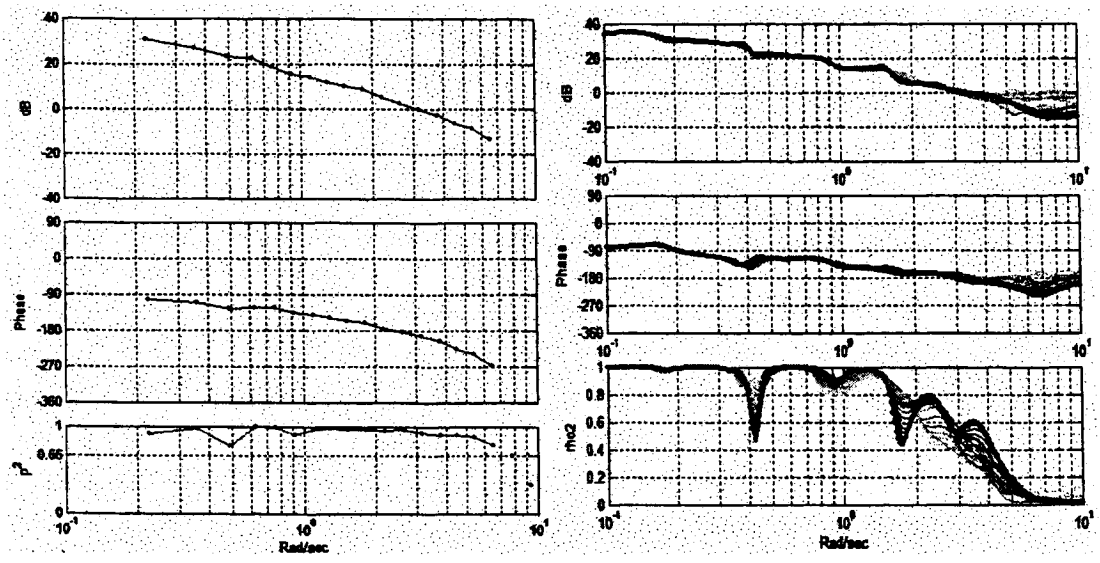
Figure 108. R15 Pitch Axis TVTF Analysis



a) Time Varying Model Parameters b) Time Varying Airplane Bandwidth

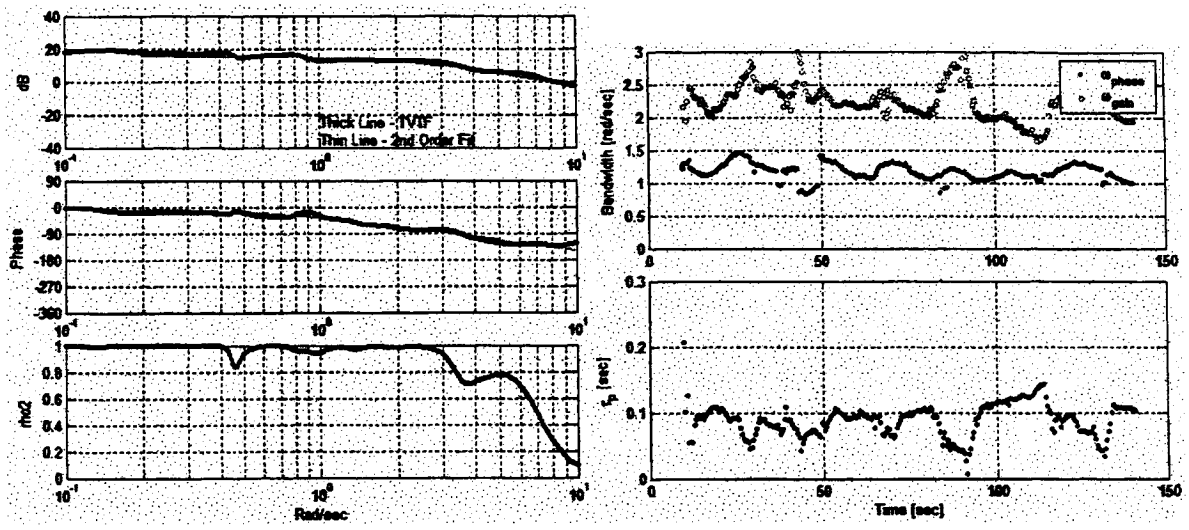
Figure 109. R15 Pitch Axis WERA Analysis

The roll axis results, although not as clean as seen previously, are much improved in comparison to the pitch axis. This is seen first in the good agreement between the FFT and TVTF methods as shown in Figure 110. Next, there is also good agreement between the TVTF (Figure 111) and WERA (Figure 112) analyses. Note also the improved coherence Figure 111a that is achieved using the roll rate signal as compared to the roll attitude signal as seen in Figure 110b.



a) FFT b) TVTF at 101 sec

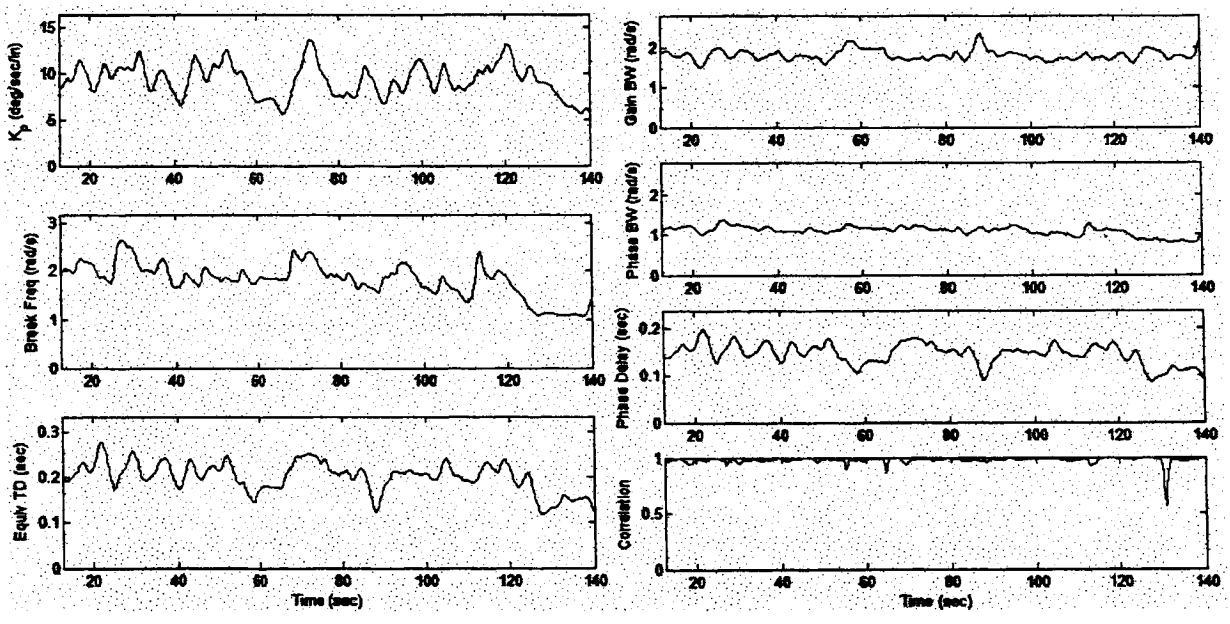
Figure 110. R15 PHI/DLAT Transfer Function Estimation



a) P/DLAT Fit to TVTF at 55.7 sec

b) Time Varying Airplane Bandwidth

Figure 111. R15 Roll Axis TVTF Analysis



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 112. R15 Roll Axis WERA Analysis

6.2.5.3 Analysis of Mild Pitch Axis PIO Run

Time histories for the mild pitch axis PIO of R99 are shown in Figure 113. In this example the pitch rate feedback fails at just over 44 sec, and mild pitch oscillations result. Better results for the TVTF (Figure 114) and WERA (Figure 115) analyses were obtained with this run. As demonstrated previously, the primary effect of the FCS failure is to decrease the phase bandwidth. Both methods demonstrate this result effectively. The TVTF technique displays a sensitivity in gain bandwidth that seems related to issues with the estimated frequency response below 0.75 rad/sec that were unique to this task. The WERA results, on the other hand, again show the previously discussed pitch axis issues with gain

bandwidth and phase delay. There is also a drop in correlation at 42 sec that persists for the rest of the displayed run length.

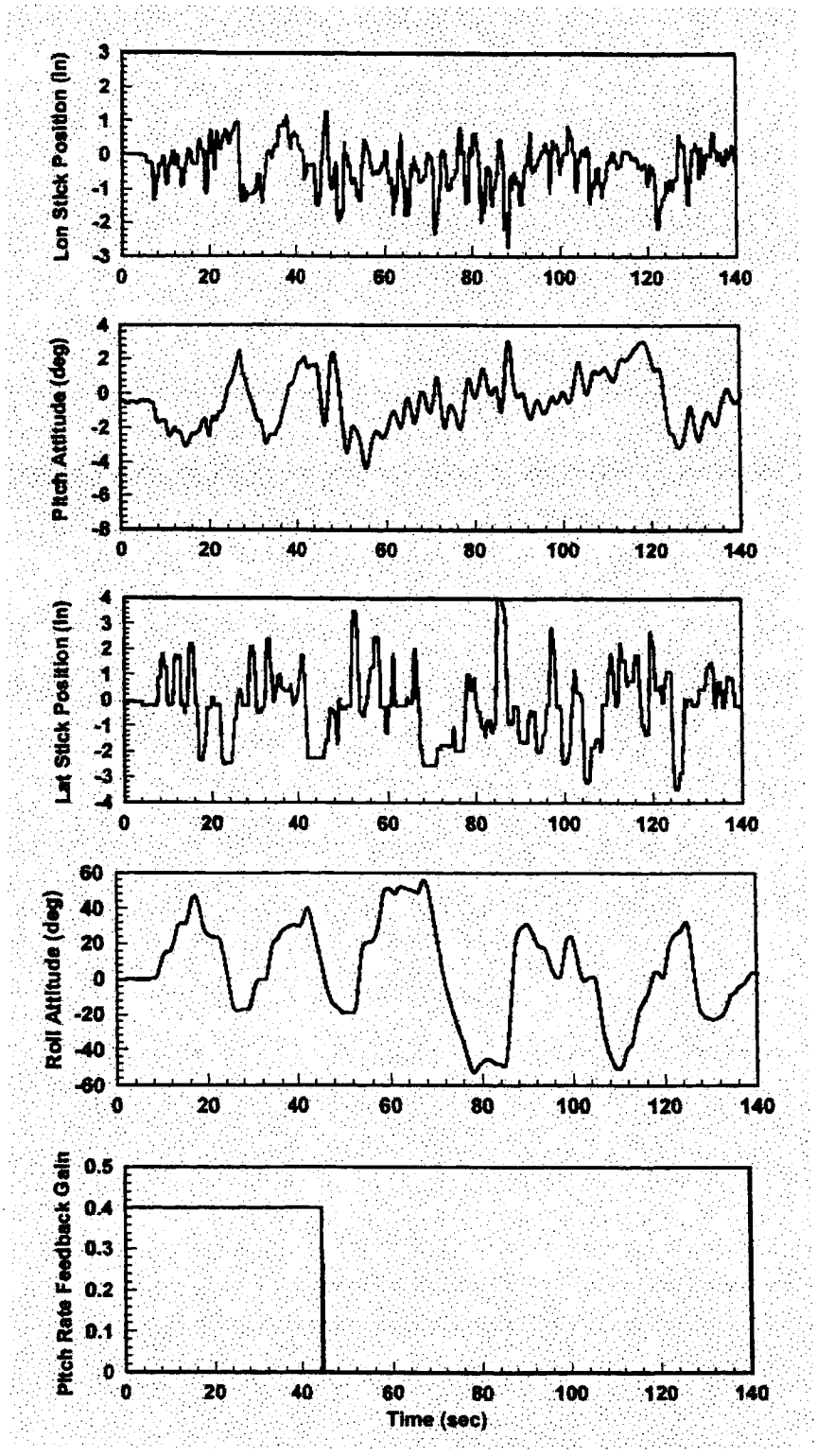
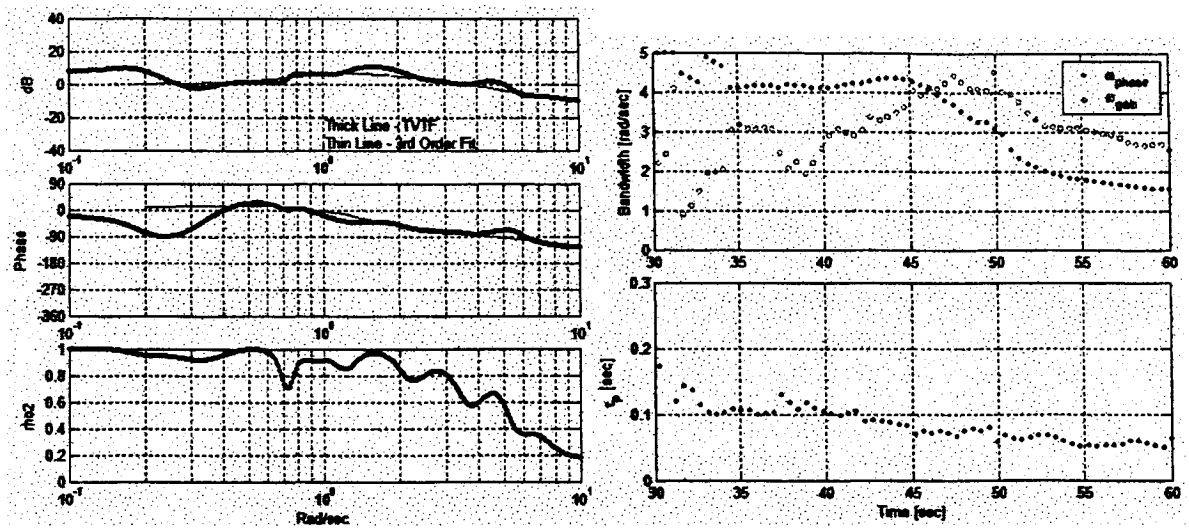


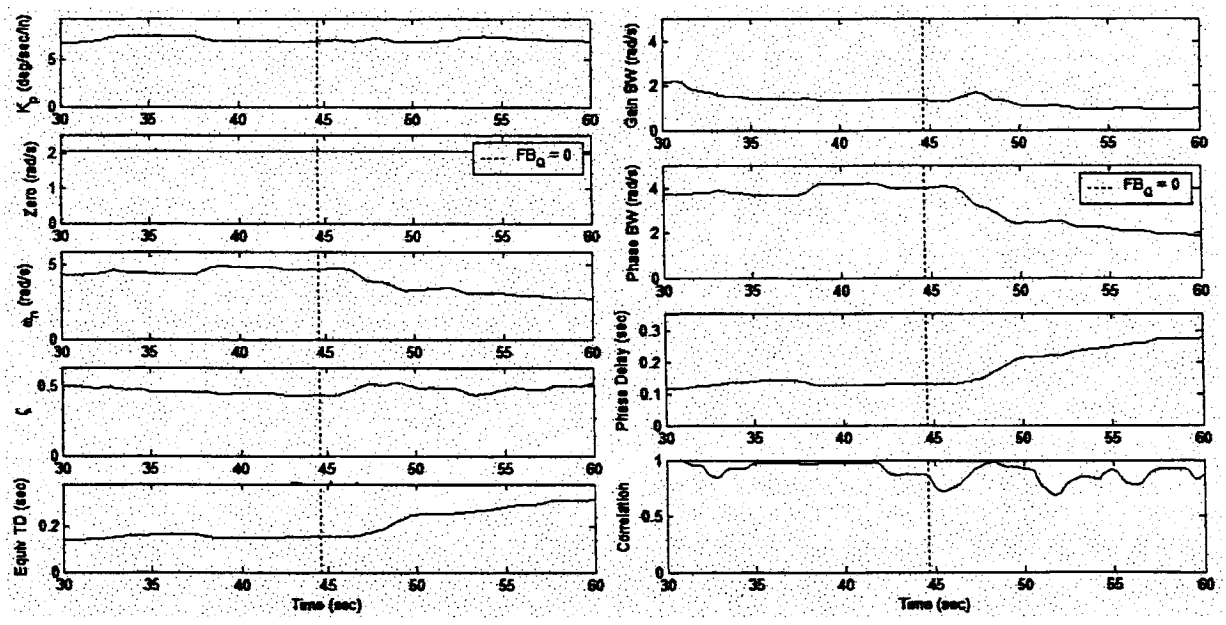
Figure 113. R99 DT, Mild Pitch PIO



a) Q/DLON Fit to TVTF at 54.8 sec

b) Time Varying Airplane Bandwidth

Figure 114. R99 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 115. R99 WERA Analysis

6.2.6 Analysis of Sustained or Severe Roll Axis PIO Run

Time histories for a sustained or severe roll axis PIO case are shown in Figure 116. As shown in the figure, following the roll rate feedback failure at just past 70 sec, significant roll overshoots develop that lead to sustained oscillations after the tracking task ends at roughly 140 sec. As was demonstrated with the no PIO DT run, the roll axis TVTF (Figure 117) and WERA (Figure 118) analyses yield much cleaner results than those shown for the pitch axis. There is also good agreement between the two methods.

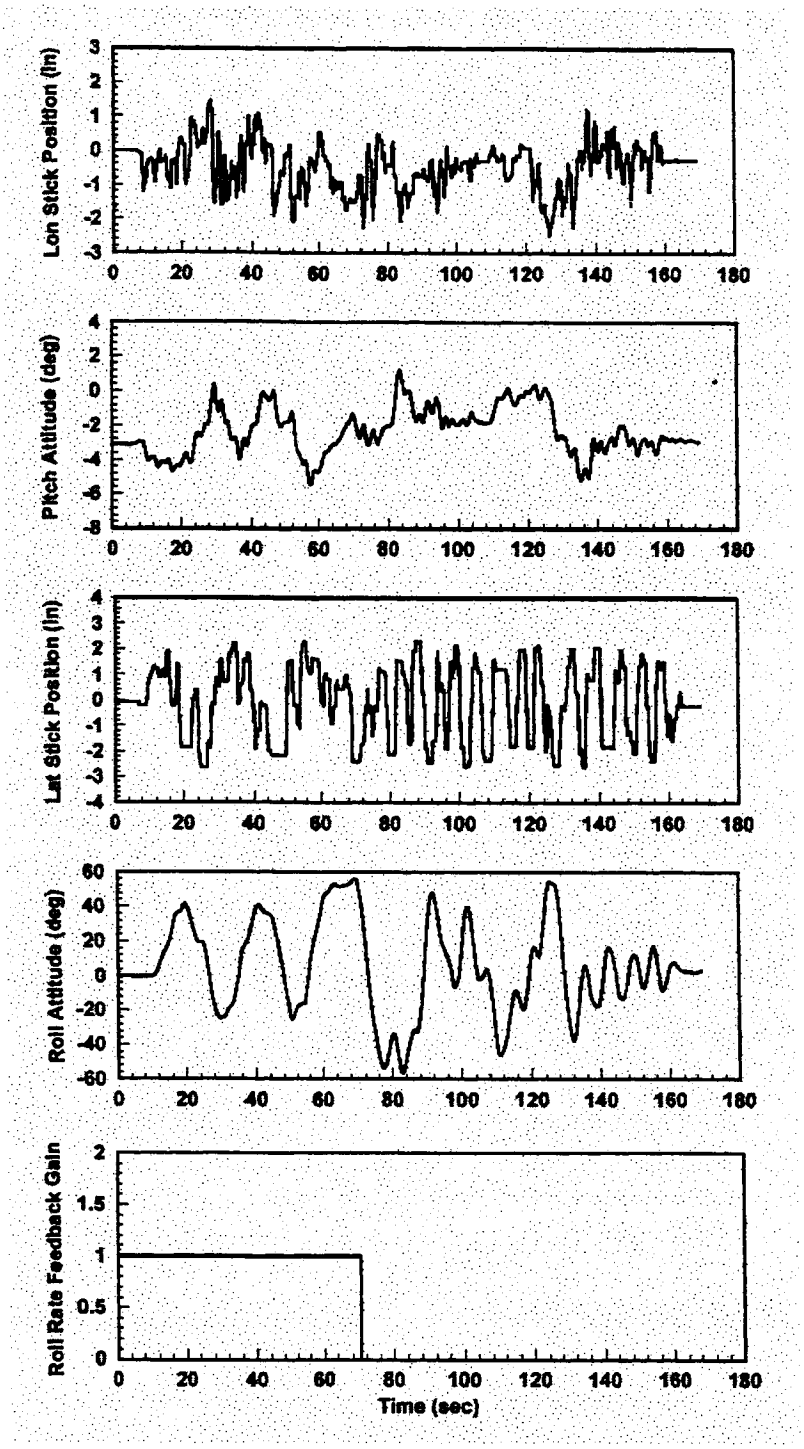
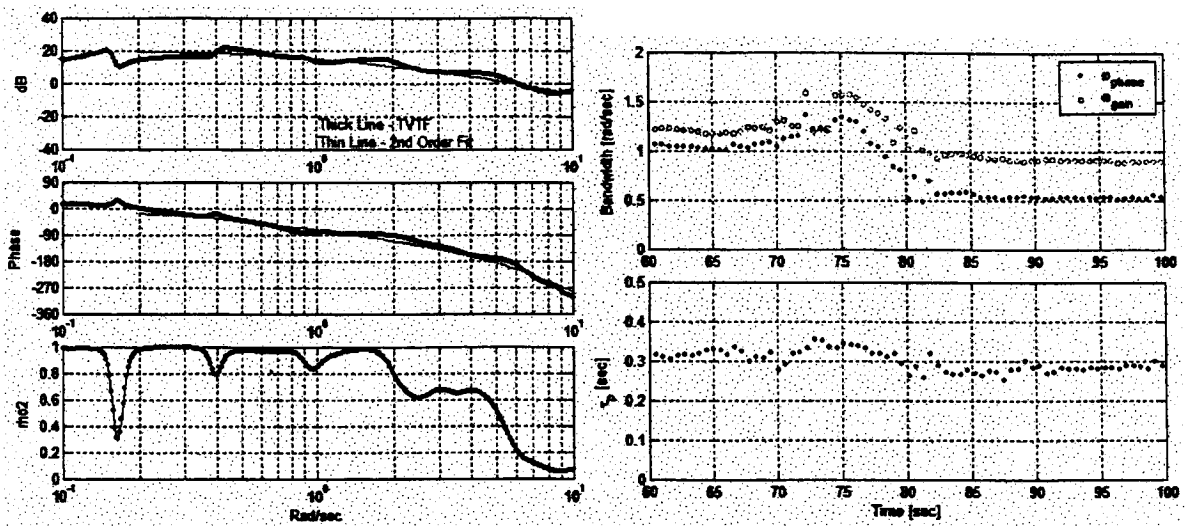


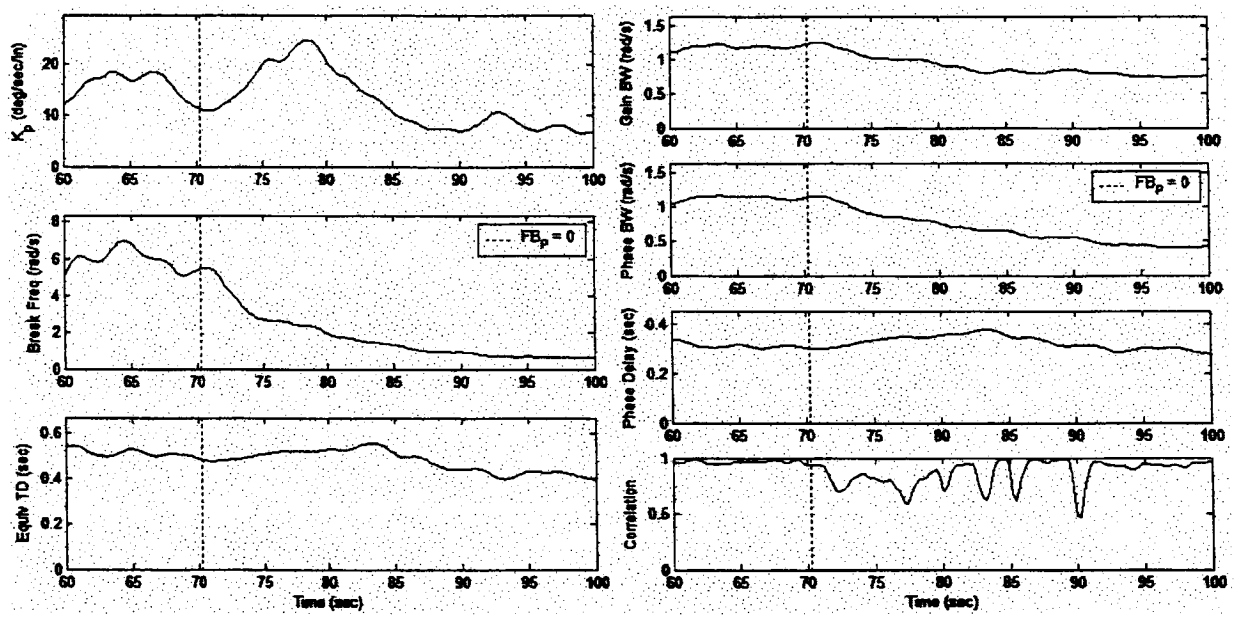
Figure 116. R128 DT, Sustained or Severe Roll PIO



a) P/DLAT Fit to TVTF at 88 sec

b) Time Varying Airplane Bandwidth

Figure 117. R128 TVTF Analysis (21 Point Time Averaging)



a) Time Varying Model Parameters

b) Time Varying Airplane Bandwidth

Figure 118. R128 WERA Analysis

6.3 Assessment of LOCATS Real-Time Computation Capabilities

6.3.1 Timing Loop

The real time signals used by LOCATS are obtained from the Mil 1553 data bus via software data requests. These requests are inside a timing loop as follows:

Top of timing loop
- Start 20 msec timer and record current time
- Request signals
- Store signals

Transform calculations (if any)
Plotting (if any)
Loop to use up remaining part of 20 msec (if any)
End of timing loop

The calculations depend on the choice of plots. Stripcharts are updated each cycle. The time series for PSD and TF plots are stored every cycle, but the transform calculations and plotting are updated every Nincr samples, where Nincr is a specified number of increments. For example, if one transform has Nincr = 5 and a second has Nincr = 7, then they are computed and plotted respectively every 5 and 7 samples, overlapping every 35 samples.

Another factor in the timing is the number of lines drawn on each plot. The PSD and TF plots can have multiple lines, as determined by the "persistence" parameters. Drawing multiple lines turned out to have a larger timing penalty then computing transforms.

If the calculations and plotting take longer than 20 msecs then the final part of the timing loop is bypassed, and the next cycle starts by resetting the timer to zero and requesting the next batch of signals. The time at the start of each cycle is stored and the delta time indicates whether or not a slip occurs. There is not a master clock with 20 msec increments. The 20 msec timer starts when the previous cycle is completed.

6.3.2 Timing and Quantization of Buffered Signals

The signals on the Mil 1553 bus are updated according to the simulator software and are not under the control of LOCATS. The simulator consists of several computers including the flight control system computers, and their update cycles are not synchronized with LOCATS. The update rate turned out to be different depending on the signal, and for a given signal the update rate was not always uniform. The Mil 1553 bus holds the previous value of each signal until it is updated.

A detail of signals from LOCATS Run 41 is shown in Figure 119 and Figure 120. Each dot is a recorded value for that signal. The signal timing and quantization can be determined from these plots. Comments:

- The first line is the delta time between successive signals, and in this 2.5 second segment of data the delta time holds steady at 20 msec indicating there is no slippage.
- Each sample of the Q signal is held for 2 cycles, indicating the update rate is about half of the 50 Hz rate set for LOCATS.
- Each sample of the VCAS signal is held for at least 4 cycles, indicating an update rate about one-forth of 50 Hz. Sometime the signal is held for more than 4 cycles, due to quantization.
- Most samples of the THE signal are held for 1 cycle, and occasionally 2 cycles, indicating an update rate of about 50 Hz but not quite synchronized with the LOCATS update.
- The DLON samples are all held for one cycle, perfect synchronization over this segment of data.
- The lateral axis signals (PHI, P, and DLAT) are all held for many cycles, due mainly to quantization. The pilot input is primarily pitch for this run.

6.3.3 Timing Tests

LOCATS requests data from the Mil 1553 bus every 20 msec, but slips when the calculations and/or plotting takes more than this amount. The time of each sample is recorded from a precision timer built into the PC, and the delta time can therefore be used to test the computation time.

Several LOCATS runs were used primarily to record calculation times. The type of input is not relevant to the timing, but for the record each of these timing test runs was a single pitch doublet with data collected for 20 to 30 seconds. The number of plots and the length of the transform calculations were varied for each run. The transform calculations were either FFTs or wavelets, computed every Nincr samples, one time for each plot. The "persistence" is the number of lines drawn for each PSD plot, a persistence of 10, for example, plots the current PSD line and the 9 previously calculated PSD lines.

The results of the timing tests are summarized in Table 14. The delta time plots for most of the runs are included in Figure 121 and Figure 122 (the runs not included have flat, 20 msec delta time plots).

Conclusions:

- No slippage with just stripcharts.
- No slippage for a single PSD plot using either FFTs or wavelets up to length 4096. The number 4096 can be increased some, but the tests are not sufficient to determine an exact bound.
- Plotting with persistence results in severe timing slippage, taking as much as 10 times longer than the transform calculations.

Recommendations:

- Use the current version of LOCATS with just stripcharts. This is the safest way to obtain data with no slippage (this is what was done for runs above R8).
- Change the LOCATS software to separate the data collection from the calculations and plotting. Do this either with a single computer using interrupt driven multiple threads, or with separate computers. Writing this type of software was beyond the scope of the current project.

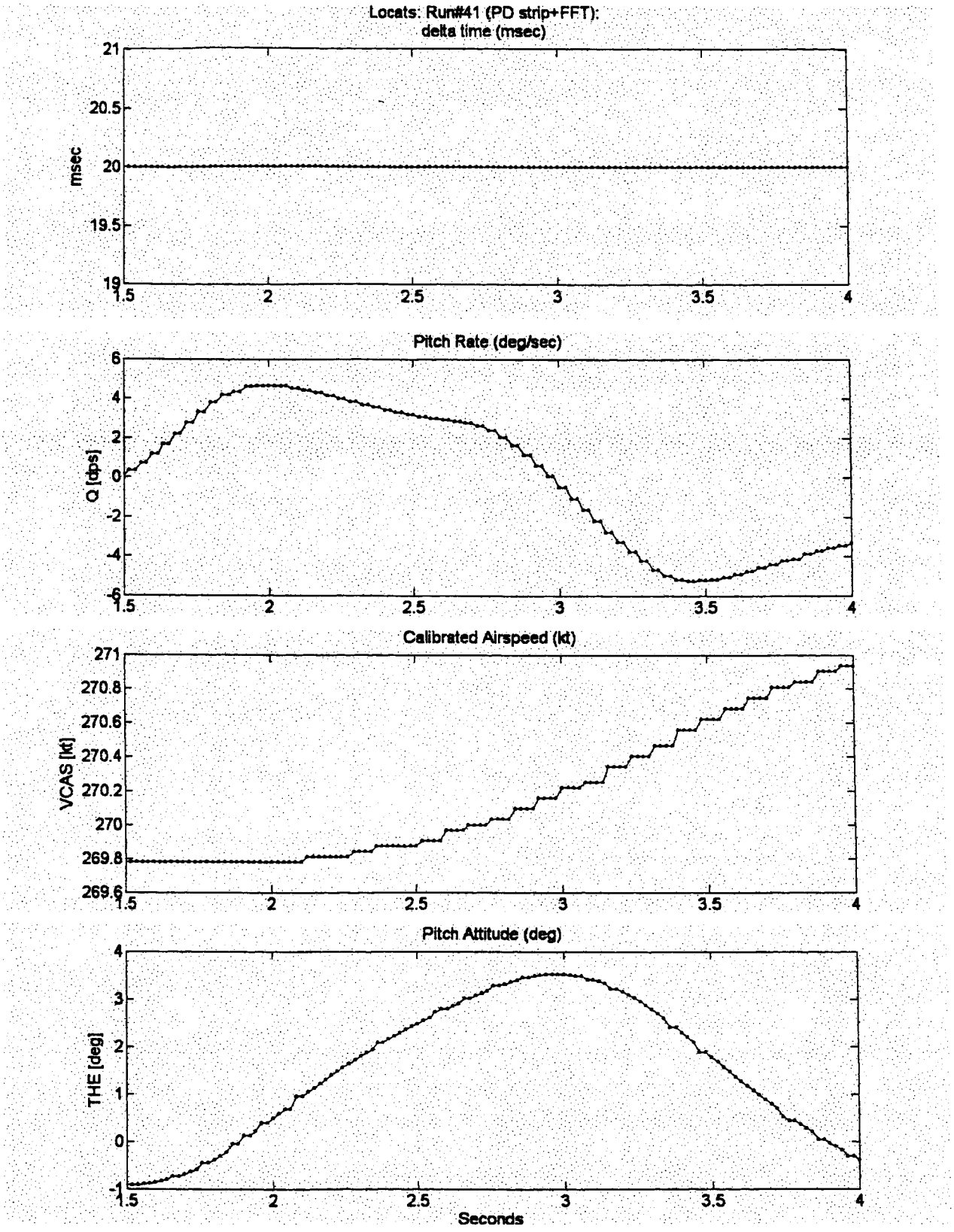


Figure 119. (Part 1 of 2) Detail of Signals Showing Timing and Quantization

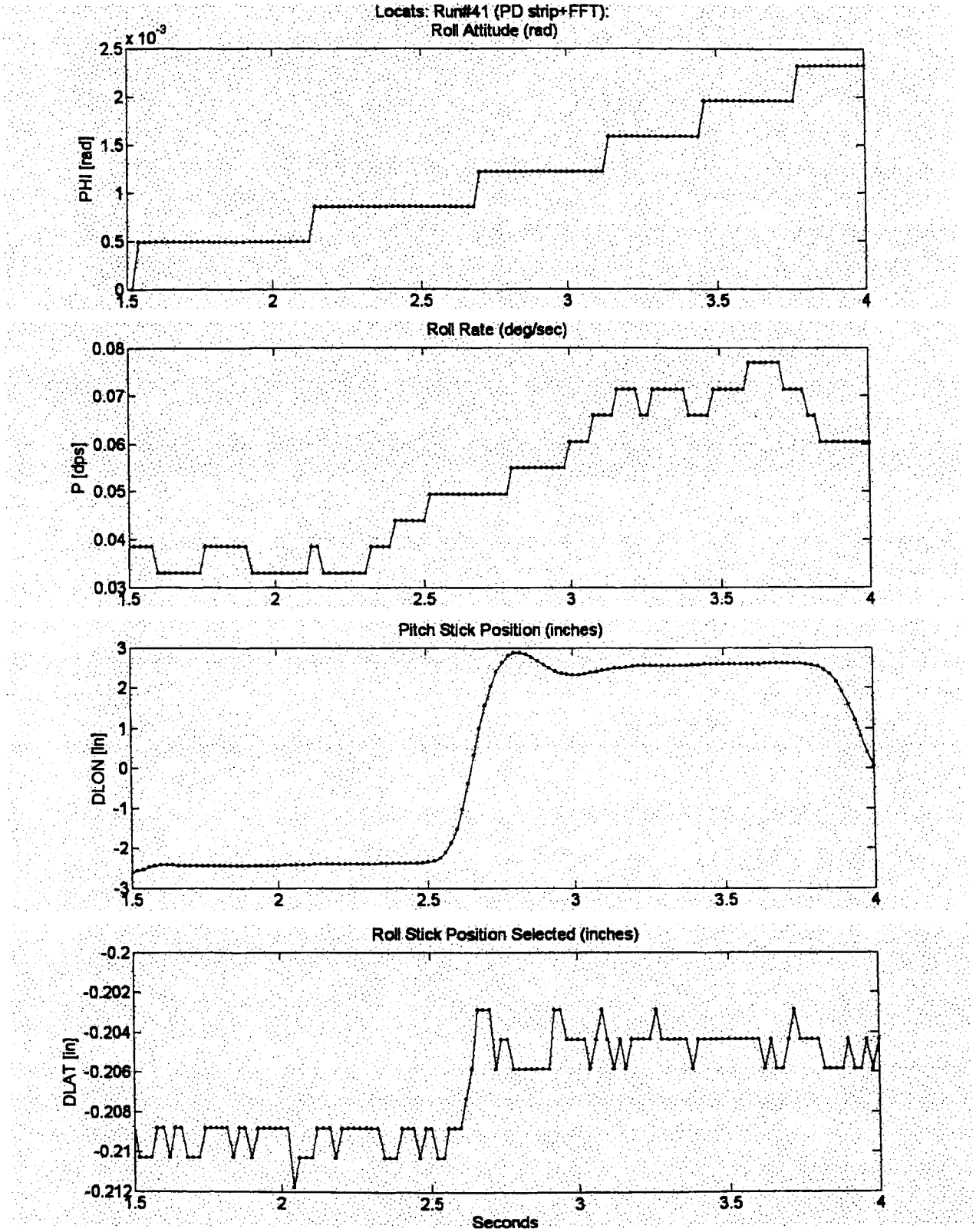


Figure 120. (Part 2 of 2) Detail of Signals Showing Timing and Quantization

Table 14: LOCATS Timing Tests

LOCATS Run #	Description Of Run	Real Time Plots	# Samples out of first 1000 that slip more than 0.1 msec (nominal value is 20 msec)
39	Pitch doublet	Stripchart only	0
41	Pitch doublet	Stripchart 4096 point FFT (Persistence = 1)	0
42	Pitch doublet	Stripchart 4096 point FFT (Nincr = 6, Persistence = 1) 4096 point Wavelet (Nincr = 8, Persistence=1))	158 (slips to about 30 msec every 24 samples when transforms line up)
43	Pitch doublet	Stripchart 4096 point FFT (Nincr = 6, Persistence = 10) 4096 point wavelet (Nincr = 8, Persistence = 10)	196 (slips to 40 to 60 msec for each transform plot, slips to 140 to 160 msec every 24 samples when transform plots line up)
60	Pitch doublet	Stripchart 8192 point wavelet (Nincr = 6, Persistence = 1)	136 (slips 0 to 4 msec for each transform)
61	Pitch doublet	Stripchart 4096 point wavelet (Nincr = 6, Persistence = 1)	0
62	Pitch doublet	Stripchart 6144 point wavelet (Nincr = 6, Persistence = 1)	3 (slips more than 0.1 msec 3 times, with a max slip of 0.6 msec)
63	Pitch doublet	Stripchart 8196 point FFT (Nincr = 6, Persistence = 1) 6144 point wavelet (Nincr = 7, Persistence = 1)	35 (slips 6 to 10 msec every 42 samples when transform line up)

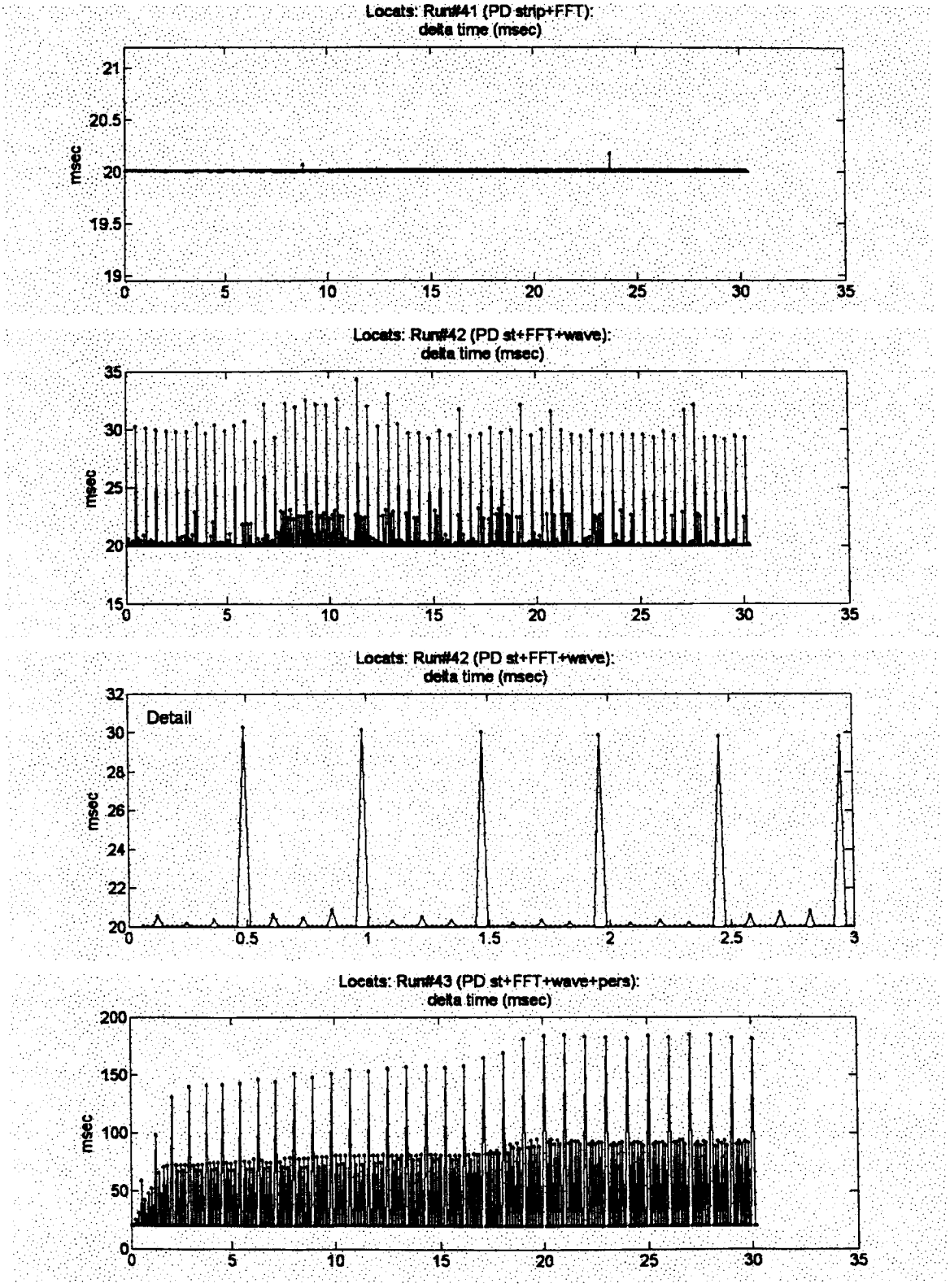


Figure 121. (Part 1 of 2) LOCATS Timing Tests

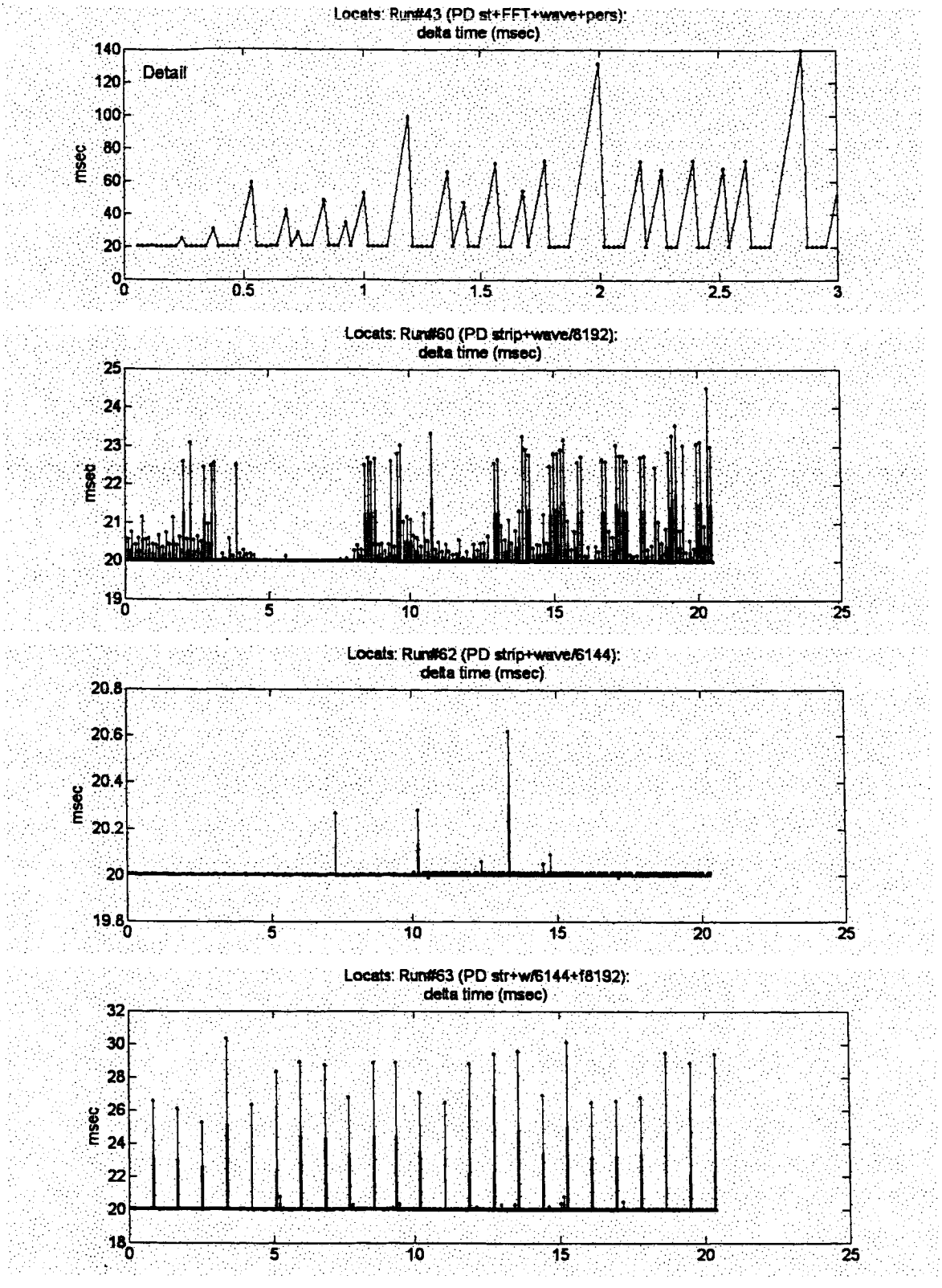


Figure 122. (Part 2 of 2) LOCATS Timing Tests

6.4 Assessment of LOCATS Failure Detection

Changes are made midway through a run to the Stability Control and Augmentation System (SCAS) in the manned simulator. Most of the changes are to zero out selected gains, effectively disconnecting the SCAS so that the pilot suddenly has to control the open loop vehicle. Other changes are to suddenly add significantly more delay. The SCAS changes are considered failures, the estimated stability metrics change as a result of these failures, and the question is how long does it take to detect the change in the stability metrics?

The detection is not automated in real time, and the detection times listed in Table 15 are computed by post-processing of the data. The stability metric that changes the most is selected, initial and final values on either side of the failure are measured, and the detection time is defined as the time it takes for the metric to cross percentage thresholds between the initial and final values. The process for measuring detection time is illustrated for Run 122 in Figure 123. In this example it takes 11.1 seconds for the phase bandwidth to cross the 60% threshold line.

Each of the failures in Table 15 results in PIO. The period of the dominant frequency in the oscillation is reported in Table 15. Scalograms (i.e., wavelet transforms) of the output response are used to identify the dominant frequency. An example for Run 122 is shown in Figure 124. The concentration of power at the PIO frequency is nicely displayed in this plot.

Upon reviewing Table 15 the following comments are made:

- The 60% threshold is a reasonable value to use for detection, the tradeoff being false alarms and detection time.
- The detection time in the pitch axis (for the 60% threshold) is consistently around 6.5 seconds.
- The detection time in the roll axis (for the 60% threshold) has a larger range of values, from 5.1 to 11.5 seconds.
- The detection time as a ratio of the PIO frequency is a good figure of merit. These ratios are included in Table 15. Detection in less than 2 cycles is considered good.
- The detection time in the pitch axis (for the 60% threshold) ranges from 1.3 to 1.9 cycles.
- The detection time in the roll axis (for the 60% threshold) ranges from 0.9 to 2.1 cycles.

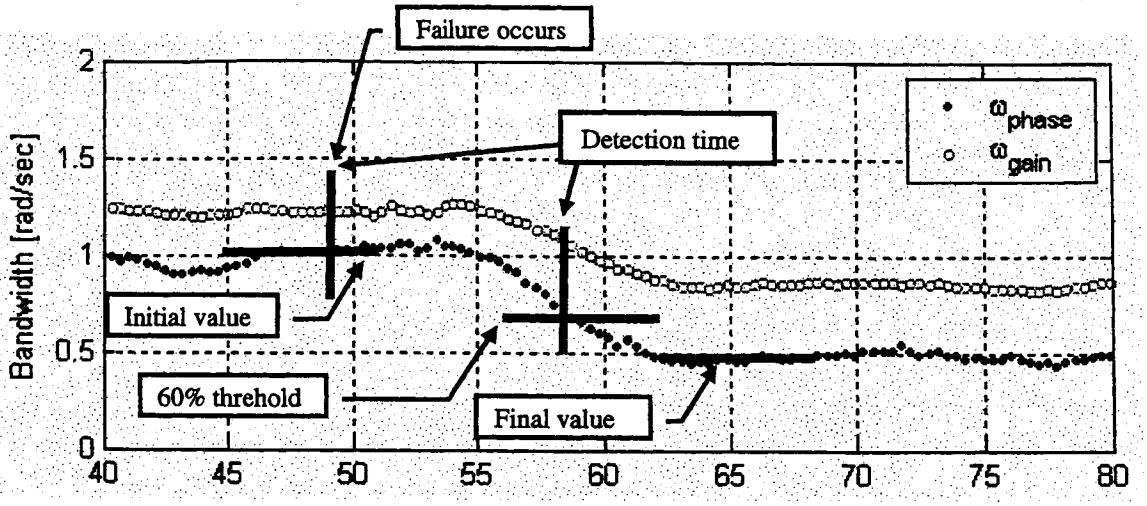


Figure 123. Example (Run 122) Showing How Detection Time is Determined

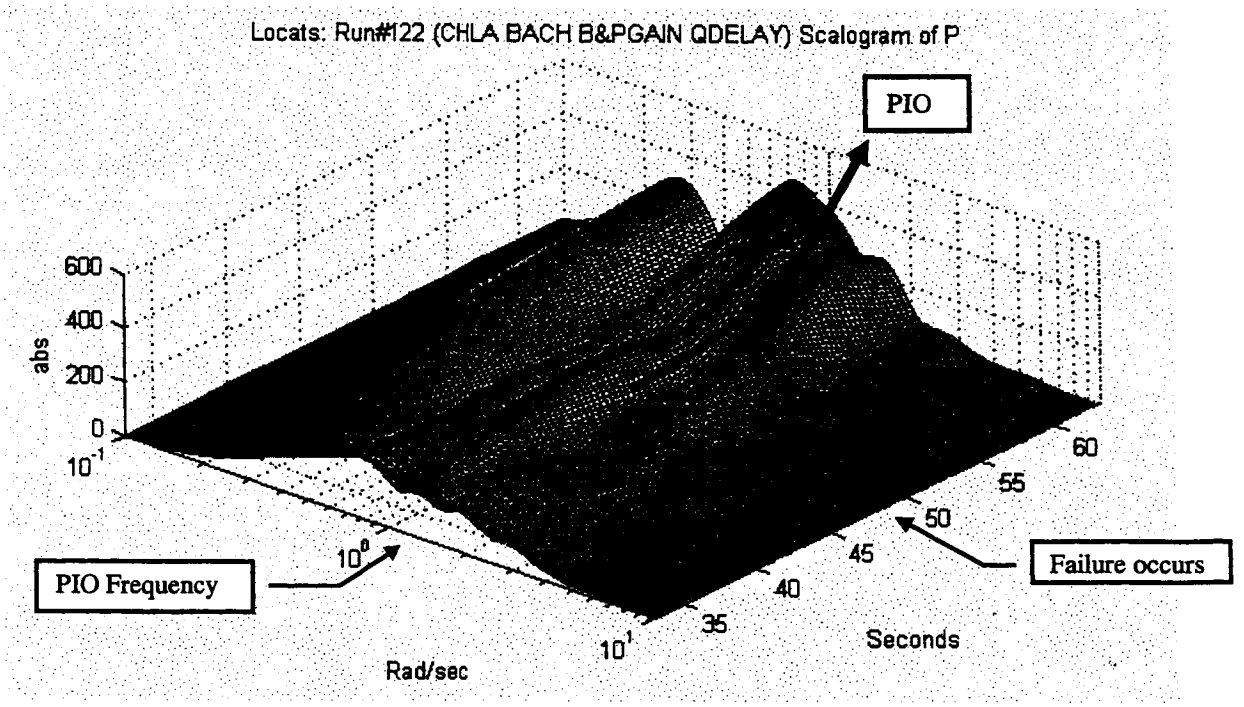


Figure 124. Example (Run 122) Sh.owing Wavelet Transform of PIO Response

Table 15. Failure Detection Times

Run	FC Config	Task	SCAS change	Comment	PIO period [sec]	Stability metric change	Detection time for stability metric change [sec] (ratio with PIO period)		
							30%	60%	90%
R122	CH CLA	BACH	BGain and PGain zeroed	Roll TD whole run, mild PIO	5.3	ω_{phase} 0.96 → 0.48 r/s	9.7 (1.8)	11.1 (2.1)	13.6 (2.5)
R29	C CLA	BACH	Roll TD increased	Large overshoots and PIO	4.2	τ_p 0.11 → 0.30 sec	4.5 (1.1)	6.5 (1.5)	7.5 (1.8)
R115	CH CLA	PACH	QGain zeroed	Three cycle PIO	3.7	ω_{phase} 3.3 → 1.4 r/s	5.0 (1.4)	6.3 (1.7)	8.4 (2.3)
R69	CM CLA	PACH	QGain zeroed	Pitch TD whole run, PIO	3.5	ω_{phase} 2.1 → 1.1 r/s	6.6 (1.9)	6.6 (1.9)	9.3 (2.6)
R119	CH CLA	SOS (Pitch)	QGain zeroed	Pitch TD whole run, sustained PIO	5.0	ω_{phase} 2.1 → 1.1 r/s	6.4 (1.3)	6.4 (1.3)	14.0 (2.8)
R124	CH CLA	SOS (Roll)	BGain and PGain zeroed	Roll TD whole run, sustained oscillations	8.9	ω_{phase} 1.1 → 0.48 r/s	8.1 (0.9)	11.5 (1.3)	17.2 (1.9)
R99	CM CLA	DT (Pitch)	QGain zeroed	Mild PIO	4.4	ω_{phase} 4.2 → 1.4 r/s	4.5 (1.0)	6.9 (1.6)	11.2 (2.5)
R128	CH CLA	DT (Roll)	BGain and PGain zeroed	Pitch TD and roll TD whole run, sustained roll PIO	5.5	ω_{phase} 1.1 → 0.58 r/s	4.6 (0.8)	5.1 (0.9)	5.1 (0.9)

Abbreviations: C = cruise, CM = cruise with turbulence, CH = high altitude cruise, CLA = cruise light aft cg, BACH = bank angle capture and hold, PACH = pitch angle capture and hold, SOS = sum of sines, DT = discrete tracking, PIO = pilot induced oscillation, TD = time delay

7. SUMMARY AND CONCLUSIONS

7.1 Program Summary

Wavelet transforms are used for on-line detection of aircraft loss of control. Wavelet transforms are compared with Fourier transform methods and shown to more rapidly detect changes in the vehicle dynamics. This faster response is due to a time window that decreases in length as the frequency increases. New wavelets are defined that further decrease the detection time by skewing the shape of the envelope.

The wavelets are used for power spectrum and transfer function estimation. Smoothing is used to tradeoff the variance of the estimate with detection time. Wavelets are also used as front-end to the eigensystem reconstruction algorithm. Stability metrics are estimated from the frequency response and models, and it is these metrics that are used for loss of control detection.

A Matlab toolbox was developed for post-processing simulation and flight data using the wavelet analysis methods. A subset of these methods was implemented in real time and named the Loss of Control Analysis Tool Set or LOCATS. A manual control experiment was conducted using a hardware-in-the-loop simulator for a large transport aircraft, in which the real time performance of LOCATS was demonstrated. The next step is to use these wavelet analysis tools for flight test support.

7.2 Strengths and Weaknesses of the Wavelet-based Analysis Methods

7.2.1 TVTF

Real time computation of the Time Varying Transfer Functions using wavelet transforms was demonstrated using the LOCATS software. The wavelet transforms were computed using convolution summations with stored and sampled mother wavelets.

The main advantage of the wavelet based method versus Fourier methods is the faster detection of changes in the frequency response. This was expected based on the theory and the Phase I feasibility study, and the faster detection was verified in real time using LOCATS.

Another advantage of wavelet methods is they are non-parametric, meaning a model of the system is not needed in order to estimate the frequency response.

A weakness of wavelet methods is the input must be consistently exciting in the frequency ranges of interest. This is true of sum-of-sine inputs and Guassian inputs, but not true of chirps and discrete inputs.

The variance of the estimated frequency response is significantly reduced by smoothing in time and frequency, but this remains an issue that needs improvement.

The detection of frequency response changes and stability metric changes was not automated. The faster detection of wavelets versus Fourier methods is based on post-processing the data and also by qualitative assessments made by viewing the data.

7.2.2 WERA

Wavelets can be used as a front end for the Eigensystem Realization Algorithm. The Markov parameters are estimated in the transform domain and then converted back to the time domain. This methodology was demonstrated on simulated data and by post-processing the experimental data, but was not implemented in real time.

The main advantage of WERA is the reduced sensitivity to measurement noise. This advantage was demonstrated using the experimental data.

The WERA method was further modified to reduce sensitivity to the initial conditions, and this was also demonstrated. The current development, however, is sensitive to modeling assumptions and limited to second order models.

The WERA is also sensitive to offsets in the data, as is typical of aircraft data. Currently this remains an open problem.

7.3 Assessment of the LOCATS Software

7.3.1 Real Time Performance

The longest wavelet transform demonstrated in real time had 4096 points for the lowest frequency (the number of points decreases as the frequency increases). The longest Fourier transform demonstrated in real time also had 4096 points. Multiple transforms and longer transforms result in slippage in the data collection with resulting non-uniform samples.

The transform plots have a "persistence" parameter that allows multiple lines on each plot, so that current and previous transforms can be compared. The plotting package was slow in doing this, however, and the use of persistence caused slippage in the data collection, more serious than the use of transforms of 16K points.

With the current implementation of LOCATS it is recommended that data be collected using just strip charts and a single power spectra calculation with less than 4096 points and with no persistence.

The main limitation in performance is that all of the transform calculations and plotting needs to take place within a single sample period, set at 50 msec for the experimental data collection.

A multi-processor or multi-thread implementation is needed to improve the performance, so that transform calculations and plotting can extend beyond a single sample period with slippage in the data collection. For example, if the transforms are computed every 10 samples then 500 msec would be available for calculation and plotting.

The convolutions used to compute the continuous wavelet transforms are estimated to take about 30N multiplications, where N is the number of points, considerably less than N-squared for large N, but not as fast as the pyramid algorithm for digital wavelets. Digital wavelets require multiple voices to meet frequency resolutions requirements for systems analysis, nevertheless there are improvements to be made in the real time calculation.

7.3.2 LOCATS Success Criteria

Was loss of control or impending loss of control observed?

- Yes, numerous PIO events were encountered. They were not divergent.

If yes, what was the triggering event?

- The triggering event was primarily a FCS rate feedback gain change in the presence of a significant added time delay.
- Inadvertent autothrottle disengage at a FCS failure point triggered a PIO.
- Changes in command path rate limits triggered mild PIO.

If yes, was the event detected by the LOCATS software?

- Changes in frequency response and airplane bandwidth were measured in real-time.
- Automated real-time detection mechanisms were not implemented.

If no, was a non-event detected by the LOCATS software?

- Time and frequency domain smoothing reduces variance and provides coherence measures.
- Smoothing reduces probability of false alarms.

What metric(s) provided the detection mechanism?

- Airplane bandwidth was the primary metric.

What was the detection time?

- Definition of detection time is sensitive to type of FCS failure.
- For feedback gain changes and insertion of added time delay, detection time can be measured from time of failure.
- For rate limiting, detection time should be measured from the saturation point.
- Percentage changes of metrics such as phase bandwidth were used to determine failure detection times. Typical detection times (for 60% changes) ranged from 5 to 12 seconds. Failures that resulted in PIO were detected in less than 2 cycles of the oscillation.

Can the detection time be improved?

- Yes, the technology has not yet matured. Reduced sensitivity to noise and initial conditions and automation are needed.

Was information provided to the pilot?

- No pilot alerting mechanism was implemented.

7.4 Recommendations for Further Work

Recommendations for improving wavelet theory include:

- Improved coherence measures;
- Improved Least Square/WERA parameter identification;
- Multivariable identification;
- Noise reduction; and
- Reduced sensitivities to nonlinearities.

Recommendations for improving wavelet applications include:

- Implement digital wavelets;
- Additional metrics;
- Automated detection;
- Pilot alerting;
- Control room real time assessment; and
- On-board simulator and flight test assessment.

APPENDIX A – ESTIMATION OF POWER SPECTRA AND TRANSFER FUNCTIONS USING WAVELETS

1. OVERVIEW

Wavelet transforms can be used to estimate power spectra and transfer functions. Here we give formulas for the mean and variance of these estimates. Section 2 is a review of estimation of the statistics of random variables. Section 3 defines random processes and presents statistics of random processes that are used for power spectral estimation. Section 4 then review estimation of the statistics of a random process, including the auto-spectrum of the random process. The review material follows the treatment in Ref. 27.

2. ESTIMATING STATISTICS OF RANDOM VARIABLES

2.1 Single Random Variable

A random variable x has a probability density function (pdf) $p(x)$. The mean, mean square, and variance are functions of the random variable defined respectively by:

$$\mu = E[x] = \int_{-\infty}^{\infty} x p(x) dx$$

$$\psi^2 = E[x^2] = \int_{-\infty}^{\infty} x^2 p(x) dx$$

$$\sigma^2 = E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx = \psi^2 - \mu^2$$

The standard deviation is σ. The moment generating function is:

$$M(s) = \int_{-\infty}^{\infty} e^{sx} p(x) dx$$

from which the moments can be found by:

$$E[x^i] = \int_{-\infty}^{\infty} x^i p(x) dx = M^{(i)}(0)$$

2.1.1 Gaussian Distribution

The most common assumption that is made about random variables in systems theory is that they have a Gaussian distribution, also called the normal distribution. The pdf is:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)/(2\sigma^2)}$$

For shorthand use $x \sim N(\mu, \sigma)$, where μ is the mean and σ is the standard deviation. The standardized random variable is $z = (x-\mu)/\sigma$, which is $z \sim N(0,1)$. The probability distribution of z is:

$$P(z_\alpha) = \text{Prob}(z \le z_\alpha) = \int_{-\infty}^{z_\alpha} p(z) dz = 1 - \alpha$$

which is usually tabulated as the area under the right-hand-side tail:

$$1 - P(z_\alpha) = \int_{z_\alpha}^{\infty} p(z) dz = \text{Prob}(z > z_\alpha) = \alpha$$

where z_α is called the 100α percentage point. The error function is sometimes used:

$$\text{erf}(z)=\frac{2}{\sqrt{\pi}}\int_0^z e^{-u^2}du\ ,\ P(z_\alpha)=\frac{1}{2}\left[1+\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right]$$

The area within one, two, and three standard deviations is:

$$\text{Prob}(\mu\le z<\mu)=0.6827$$
$$\text{Prob}(-2\sigma\le z<2\sigma)=0.9545$$
$$\text{Prob}(-3\sigma\le z<3\sigma)=0.9973$$

So less than 1% of the samples will be in the tails beyond 3σ. The moment generating function for x is:

$$M(s)=e^{sx+s^2x^2/2}$$

In the estimation problems we will use up to fourth order moments. The moment generating function (6) is used to derive the first four moments listed in Table 16. The remaining entries pertain to the case of two random variables.

2.1.2 Chi-Square Distribution

The sum squared of n random variables $z_i \sim N(0,1)$ is chi-square with n degrees of freedom:

$$\chi_n^2=z_1^2+z_2^2+\cdots+z_n^2$$

This distribution is used in the estimation of variance. The pdf is:

$$p(\chi^2)=[2^{n/2}\Gamma(n/2)]^{-1}e^{-\chi^2/2}(\chi^2)^{n/2-1}\quad \chi^2\ge 0$$

where $\Gamma(n/2)$ is the gamma function. The mean, variance, and the 100α percentage point are:

$$\mu_{\chi^2}=n$$

$$\sigma^2_{\chi^2}=2n$$

$$\int_{\chi^2_{n;\alpha}}^{\infty}p(\chi^2)d\chi^2=\text{Prob}(\chi^2\ge\chi^2_{n;\alpha})=\alpha$$

For n greater than about 30 it follows that $\sqrt{2\chi_n^2}\sim N\left(\sqrt{2n-1},1\right)$.

2.1.3 The t Distribution

Define $y\sim\chi_n^2$ and $z\sim N(0,1)$ to be independent random variables and define the new random variable:

$$t_n=\frac{z}{\sqrt{y/n}}$$

This is called the Student's t distribution with n degrees of freedom. It is used in the estimation of the mean when the variance is not known beforehand. The pdf is:

$$p(t)=\frac{\Gamma[(n+1)/2]}{\sqrt{\pi n}\,\Gamma(n/2)}\left[1+t^2/n\right]^{-(n+1)/2}$$

The mean, variance, and the 100α percentage point are:

$$\mu_t = 0 \quad \text{for } n > 1$$

$$\sigma_t^2 = \frac{n}{n-2} \quad \text{for } n > 2$$

$$\int_{t_{n;\alpha}}^{\infty} p(t)dt = \text{Prob}(t_n \geq t_{n;\alpha}) = \alpha$$

The Student's t distribution is very nearly $N(0,1)$ for n greater than about 30.

Table 16. Gaussian Moments

Moment	Value	Zero mean
For $x \sim N(\mu,\sigma)$		
$E[x]$	μ	0
$E[x^2]$	$\psi^2 = \sigma^2 + \mu^2$	σ^2
$E[x^3]$	$3\sigma^2\mu + \mu^3$	0
$E[x^4]$	$3\sigma^4 + 6\sigma^2\mu^2 + \mu^4$	$3\sigma^4$
For $x \sim N(\mu_x,\sigma_x)$, $y \sim N(\mu_y,\sigma_y)$		
$E[xy]$	$R_{xy} = C_{xy} + \mu_x\mu_y$	0
$E[x^2y]$	$\psi_x^2\mu_y + 2C_{xy}\mu_x =$ $(\sigma_x^2 - \mu_x^2)\mu_y + 2R_{xy}\mu_x$	0
$E[x^3y]$	$3C_{xy}\psi_x^2 + (3\sigma_x^2 + \mu_x^2)\mu_x\mu_y =$ $3R_{xy}\psi_x^2 + 6\mu_x^3\mu_y$	$3R_{xy}\sigma_x^2$
$E[x^2y^2]$	$\psi_x^2\psi_y^2 + 2(C_{xy}^2 + 2C_{xy}\mu_x\mu_y) =$ $\psi_x^2\psi_y^2 + 2(R_{xy}^2 - \mu_x^2\mu_y^2)$	$\sigma_x^2\sigma_y^2 + 2R_{xy}^2$
For $x_1, x_2, x_3, x_4 \sim N(\mu,\sigma)$		
$E[x_1x_2x_3x_4]$	$C_{12}C_{34} + C_{13}C_{24} + C_{14}C_{23} + \mu^4 +$ $\mu^2(C_{12} + C_{13} + C_{14} + C_{23} + C_{24} + C_{34}) =$ $R_{12}R_{34} + R_{13}R_{24} + R_{14}R_{23} - 2\mu^4$	$R_{12}R_{34} + R_{13}R_{24} +$ $R_{14}R_{23}$
$E[x_1^2x_2x_3]$	$\sigma^2C_{23} + 2C_{12}C_{13} + \mu^4 +$ $\mu^2(\sigma^2 + 2C_{12} + 2C_{13} + C_{23}) =$ $\psi^2R_{23} + 2R_{12}R_{13} - 2\mu^4$	$\sigma^2R_{23} + 2R_{12}R_{13}$

If x and y are independent then $C_{xy} = 0$ and $R_{xy} = \mu_x\mu_y$.

2.2 Two Random Variables

Two random variables x and y have the joint pdf $p(x,y)$. It follows that

$$p(x) = \int_{-\infty}^{\infty} p(x,y)dy \quad , \quad p(y) = \int_{-\infty}^{\infty} p(x,y)dx$$

from which the moments of x and y can be determined. The correlation and covariance function are defined respectively by:

$$R_{xy} = E[xy] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyp(x,y)dxdy$$

$$C_{xy} = E[(x-\mu_x)(y-\mu_y)] = R_{xy} - \mu_x\mu_y$$

The correlation coefficient is an important parameter defined by:

$$\rho_{xy} = \frac{C_{xy}}{\sigma_x\sigma_y}$$

If the random variables x and y are independent then $p(x,y) = p(x)p(y)$, and hence $R_{xy} = \mu_x\mu_y$, $C_{xy} = 0$, and $\rho_{xy} = 0$. A weaker condition is if x and y are uncorrelated, defined by $\rho_{xy} = 0$. It is a weaker condition because uncorrelated does not imply independence, with the exception of joint Gaussian random variables. The moment generating function is:

$$M(s,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{sx+ty} p(x,y)dxdy$$

from which:

$$E[x^i y^j] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^i y^j p(x,y)dxdy = M^{(i)(j)}(0,0)$$

2.2.1 Joint Gaussian Distribution

The joint or bivariate Gaussian pdf is a three dimensional bell-shaped curve:

$$p(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}} \exp\left\{-\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho_{xy}\left(\frac{y-\mu_y}{\sigma_y}\right)\left(\frac{x-\mu_x}{\sigma_x}\right) + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right] \Big/ 2\left(1-\rho_{xy}^2\right)\right\}$$

The joint pdf expression is complicated, but fortunately we will not need to us it for any derivations. The moment generating function is more useful:

$$M(s,t) = e^N \quad , \text{ where } N = \mu_x s + \mu_y t + \frac{1}{2}(\sigma_x^2 s^2 + 2stC_{xy} + \sigma_y^2 t^2)$$

From which joint moments in Table 16 are derived. When x and y are independent it follows that $\rho_{xy} = 0$, $C_{xy} = 0$, $R_{xy} = \mu_x\mu_y$, and $E[x^2y^2] = \psi_x^2\psi_y^2$.

2.3 Estimation

Different parameters of the random variable can be estimated from N samples x_i of the random variable. It is assumed that the samples are independent and identically distributed (iid). Usually it is the moments of the random variable that are to be estimated. Call the parameter to be estimated S, and the

estimate $\hat{S}$, using the hat to indicate the estimate. The estimate is itself a random variable with a mean, variance and distribution. The estimate is *unbiased* if the mean of the estimate equals the parameter, and *consistent* if the variance of the estimate approaches zero as the number of samples approaches infinity. The distribution is used to give confidence intervals to the estimate.

The moments and errors used for parameter estimation are defined in Table 17. Note that the he mean square error subtracts off the parameter, whereas the variance subtracts off the expected value of the parameter. This makes the mean square error equal to the sum of the variance plus the bias squared. The root-mean-square (rms) is the square root of the mean square error, and the standard deviation equals the rms error when the estimate is unbiased.

Table 17. Parameter Estimation, Definitions

Mean	$E[\hat{S}]$
Bias	$b[\hat{S}] = E[\hat{S}] - S$
Variance	$\text{var}[\hat{S}] = E[(\hat{S} - E[\hat{S}])^2] = E[\hat{S}^2] - E^2[\hat{S}]$
Mean square error	$E[(\hat{S} - S)^2] = \sigma^2[\hat{S}] + b^2[\hat{S}]$
Root mean square error	$\text{rms}[\hat{S}] = \sqrt{\sigma^2[\hat{S}] + b^2[\hat{S}]}$
Normalized error	$\varepsilon[\hat{S}] = \text{rms}[\hat{S}] / S$

2.4 Estimate of the Mean

The random variable x has mean μ, mean square ψ^2, and variance $\sigma^2 = \psi^2 - \mu^2$. The random variable can have any distribution, except where explicitly specified as Gaussian. The estimate of the mean is:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

The moments and errors of this estimate are listed in Table 18.

Table 18. Estimate of the Mean of a Random Variable

Mean	$E[\hat{\mu}] = \mu$
Bias	$b[\hat{\mu}] = 0$ (unbiased)
Variance	$\text{var}[\hat{\mu}] = \sigma^2 / N$ (consistent)
Rms error	$\text{rms}[\hat{\mu}] = \sigma / \sqrt{N}$
Normalized error	$\varepsilon[\hat{\mu}] = (\sigma / \mu) / \sqrt{N}$

2.4.1 Derivations

The expected value of the estimate is computed as follows:

$$E[\hat{\mu}] = E\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N} E[x_i] = \frac{1}{N}\sum_{i=1}^{N} \mu = \mu$$

The expected value equals the parameter, and hence the estimate is unbiased. Similarly, the mean square of the estimate is:

$$E[\hat{\mu}^2] = E\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right]^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} E[x_i x_j]$$

It follows from the assumption of independence of the samples that

$$E[x_i x_j] = \begin{cases} E[x_i]E[x_j] = \mu^2 & \text{if } i \neq j \\ E[x_i^2] = \psi^2 & \text{if } i = j \end{cases}$$

Hence:

$$E[\hat{\mu}^2] = \frac{1}{N^2}\sum_{i=1}^{N}\left(\psi^2 + (N-1)\mu^2\right) = \frac{1}{N}\psi^2 + \frac{N-1}{N}\mu^2 = \mu^2 + \frac{1}{N}\sigma^2$$

The variance of the estimate follows after substitution:

$$\text{var}[\hat{\mu}] = E\left[\left(\hat{\mu} - E[\hat{\mu}]\right)^2\right] = E[\hat{\mu}^2] - E^2[\hat{\mu}] = \frac{1}{N}\sigma^2$$

The variance asymptotically approaches zero as $N \rightarrow \infty$, and hence the estimate is consistent.

2.4.2 Distribution (Known Variance)

Assume the random variable x has a Gaussian distribution. The distribution of the estimated mean depends on whether or not the variance of the random variable is known. If the variance is known, it follows that:

$$\frac{\hat{\mu} - \mu}{\sigma/\sqrt{N}} = z \sim N(0,1)$$

Thus, the normalized expected value has the standardized Gaussian distribution. The Law of Large Numbers states that the same result is true when x has any distribution, or even different distributions, for large enough N. The distribution makes it possible to make statements like:

$$\text{Prob}\left[\hat{\mu} > \frac{s z_\alpha}{\sqrt{N}} + \mu\right] = \alpha$$

This sets an upper limit for the estimate of the mean with a probability α that the estimate is above this limit. Common values are $\alpha = 0.10, 0.05, 0.01$ respectively with $z_\alpha = 1.28$, 1.65, 2.33. The confidence interval is:

$$\left[\hat{\mu} - \frac{\sigma z_{\alpha/2}}{\sqrt{N}} \leq \mu < \hat{\mu} + \frac{\sigma z_{\alpha/2}}{\sqrt{N}}\right]$$

The true mean will be in this interval with $100(1-\alpha)\%$ confidence, in other words, the probability is $1-\alpha$ that the true mean is inside this interval.

2.4.3 Distribution (Unknown Variance)

More commonly the variance is not known. Use instead the following estimate of the variance:

$$\hat{\sigma} = \frac{1}{n}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 \quad , \text{ where } n = N - 1$$

Assume the random variable x has a Gaussian distribution, and it follows that:

$$\frac{\hat{\mu} - \mu}{\hat{\sigma}/\sqrt{N}} = t_n$$

Thus, the normalized estimate of the mean has a Student's t distribution with n degrees of freedom. An upper limit of the estimate is established by:

$$\text{Prob}\left[\hat{\mu} > \frac{\hat{\sigma} t_{n;\alpha}}{\sqrt{N}} + \mu\right] = \alpha$$

And the confidence interval for the true mean is:

$$\left[\hat{\mu} - \frac{\sigma t_{n;\alpha/2}}{\sqrt{N}} \le \mu < \hat{\mu} + \frac{\sigma t_{n;\alpha/2}}{\sqrt{N}}\right]$$

2.5 Estimate of the Mean Square

The estimate of the mean square is:

$$\hat{\psi}^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$$

The moments and errors of this estimate are listed in Table 19. The normalized error for the expected value of the mean is $(\sigma/\mu)\sqrt{1/N}$, and here: $\sqrt{2/N}$. Set $\sigma/\mu = 1$, and we note in this case it takes twice as many samples for the same normalized error when estimating the mean square, as opposed to estimating the mean.

Table 19. Estimate of the Mean Square of a Random Variable

Mean	$E[\hat{\psi}^2] = \psi^2$
Bias	$b[\hat{\psi}^2] = 0$ (unbiased)
Variance (Gaussian)	$\text{var}[\hat{\psi}^2] = 2(\sigma^4 + 2\sigma^2\mu^2)/N$ (consistent)
Variance (Gaussian, zero mean)	$\text{var}[\hat{\psi}^2] = 2\sigma^4/N$ (consistent)
Rms error (Gaussian, zero mean)	$\text{rms}[\hat{\psi}^2] = \sigma^2\sqrt{2/N}$
Normalized error (Gaussian, zero mean)	$\varepsilon[\hat{\psi}^2] = \sqrt{2/N}$

2.5.1 Derivations

The expected value of the estimate is computed as follows:

$$E[\hat{\psi}^2] = E\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2\right] = \frac{1}{N}\sum_{i=1}^{N} E[x_i^2] = \frac{1}{N}\sum_{i=1}^{N} \psi^2 = \psi^2$$

The expected value equals the parameter, and hence the estimate is unbiased. The mean square of the estimate is:

$$E[\hat{\psi}_x^4] = E\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2\right]^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} E[x_i^2 x_j^2]$$

This is as far as we can go in general, and so we assume $x_i \sim N(\mu,\sigma)$. From Table 16:

$$E[x_i^2 x_j^2] = \begin{cases} 3\sigma^4 + 6\sigma^2\mu^2 + \mu^4 & i = j \\ \psi^4 & i \neq j \end{cases}$$

hence:

$$E[\hat{\psi}^4] = \frac{1}{N}\left(3\sigma^4 + 6\sigma^2\mu^2 + \mu^4\right) + \frac{N-1}{N}\psi^4$$

and after some rearrangement:

$$E[\hat{\psi}^4] = \psi^4 + \frac{2}{N}(\sigma^4 + 2\sigma^2\mu^2)$$

The variance of the estimate is:

$$\text{var}[\hat{\psi}^2] = E[\hat{\psi}^4] - E^2[\hat{\psi}^2] = E[\hat{\psi}^4] - \psi^4$$

Again assume Gaussian, and then substitute to get:

$$\text{var}[\psi^2] = \frac{2}{N}(\sigma^4 + 2\sigma^2\mu^2)$$

2.6 Estimate of the Variance

Dividing by N gives a biased estimate of the variance. Dividing by $n = N-1$ fixes the problem. It is easy to unbias the estimate when working with random variables, but not so easy when working with random processes, so to provide a better analogy we present the estimates here both ways. If the mean of the random variable is known, divide by N, and there is no bias. If the mean is known to be zero, use the results for the mean square.

2.6.1 Biased Estimate

The biased estimate of the variance is:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu}_x)^2$$

The moments and errors of this estimate are listed in Table 20.

Table 20. Biased Estimate of the Variance of a Random Variable

Mean	$E[\hat{\sigma}^2] = \sigma^2(N-1)/N$
Bias	$b[\hat{\sigma}^2] = -\sigma^2/N$ (biased)
Variance (Gaussian)	$\text{var}[\hat{\sigma}^2] = 2\sigma^4(N-1)/N^2$ (consistent)
Rms error (Gaussian)	$\text{rms}[\hat{\sigma}^2] = \sigma\sqrt{(2N-1)}/N$
Normalized error (Gaussian)	$\varepsilon[\hat{\sigma}^2] = \sqrt{(2N-1)}/N$

2.6.2 Derivations for Biased Estimate

Add and subtract μ in the definition of $\hat{\sigma}^2$ and it follows that:

$$E[\hat{\sigma}^2] = E\left[\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu + \mu - \hat{\mu})^2\right] = \sigma^2 - \text{var}[\hat{\mu}]$$

We already know $\text{var}[\hat{\mu}_x] = \sigma^2/N$, and therefore $E[\hat{\sigma}^2] = \sigma^2(N-1)/N$. It is considerably more difficult to determine the variance. The variance expands to:

$$\text{var}[\hat{\sigma}^2] = E[\hat{\sigma}^4] - E^2[\hat{\sigma}^2]$$

where:

$$E[\hat{\sigma}^4] = E[(\hat{\psi}^2 - \hat{\mu}^2)^2] = E[\hat{\psi}^4] - 2E[\hat{\psi}^2\hat{\mu}^2] + E[\hat{\mu}^4]$$

There are many terms to keep track of. It follows non-easily that:

$$E[\hat{\psi}^4] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}E[x_i^2 x_j^2] = \frac{N+2}{N}\sigma^4 + \frac{2(N+2)}{N}\sigma^2\mu^2 + \mu^4$$

$$E[\hat{\psi}^2\hat{\mu}^2] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}E[x_i^2 x_j x_k] = \frac{N+2}{N^2}\sigma^4 + \frac{N+5}{N}\sigma^2\mu^2 + \mu^4$$

$$E[\hat{\mu}^4] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{\ell=1}^{N}E[x_i x_j x_k x_\ell] = \frac{3}{N^2}\sigma^4 + \frac{6}{N}\sigma^2\mu^2 + \mu^4$$

Add these up, most terms cancel, to get:

$$E[\hat{\sigma}^4] = \frac{N^2-1}{N^2}\sigma^4$$

Subtract $E^2[\hat{\sigma}^2]$, and finally:

$$\text{var}[\hat{\sigma}^4] = \frac{N^2-1}{N^2}\sigma^4 - \frac{(N-1)^2}{N^2}\sigma^4 = \frac{2(N-1)}{N^2}\sigma^4$$

2.6.3 Unbiased Estimate

The unbiased estimate of the variance is:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{N}(x_i - \hat{\mu}_x)^2 \quad , \text{ where } n = N-1$$

The moments and errors of this estimate are listed in Table 21.

Table 21. Unbiased Estimate of the Variance of a Random Variable

Mean	$E[\hat{\sigma}^2] = \sigma^2$
Bias	$b[\hat{\sigma}^2] = 0$ (unbiased)
Variance (Gaussian)	$\text{var}[\hat{\sigma}^2] = 2\sigma^4 / n$ (consistent)
Rms error (Gaussian)	$\text{rms}[\hat{\sigma}^2] = \sigma^2\sqrt{2/n}$
Normalized error (Gaussian)	$\varepsilon[\hat{\sigma}^2] = \sqrt{2/n}$

2.6.4 Derivations for Unbiased Estimate

Because $\hat{\sigma}^2_{unbiased} = (N/n)\hat{\sigma}^2_{biased}$ it follows that:

$$E[\hat{\sigma}^2_{unbiased}] = (N/n)E[\sigma^2_{unbiased}] = \sigma^2$$

$$\text{var}[\hat{\sigma}^2_{unbiased}] = (N/n)^2\,\text{var}[\hat{\sigma}^2_{biased}] = 2\sigma^4/n$$

The second of these relies on the assumption that the distribution is Gaussian.

2.6.5 Distribution of the Unbiased Estimate

Assume that the random variable x has a Gaussian distribution, and then it follows that the unbiased estimate of the variance has the following distribution:

$$\frac{n\hat{\sigma}^2}{\sigma^2} = \chi^2_n$$

In words, the sample variance, normalized, is chi-square with n degrees of freedom. The confidence interval is:

$$\left[\frac{n\hat{\sigma}^2}{\chi^2_{n;\alpha/2}} \le \sigma^2 < \frac{n\hat{\sigma}^2}{\chi^2_{n;1-\alpha/2}}\right]$$

We can use this chi-square distribution to back into the mean and variance results listed in Table 21. This is a good check on the results more laboriously derived using summations. The expected value and variance of the chi-square distribution are respectively n and $2n$, hence the mean of $\hat{\sigma}^2$ is σ^2/n times n, and the variance is $(\sigma^2/n)^2$ times $2n$.

3. STATIONARY RANDOM PROCESSES

3.1 Definitions

3.1.1 First and Second Order Moments

The signals $x(t)$ and $y(t)$ are defined as real, random processes. At each point in time these signals are random variables with a pdf. The mean of $x(t)$, for example, is:

$$\mu_x(t) = E[x(t)] = \int_{-\infty}^{\infty} x(t)\, p_x(x,t)dx$$

The most common assumption in system theory about this distribution is that is it a Gaussian distribution. What makes random processes different from random variables is that the pdf, and hence the moments, can change with time. In addition, and more importantly, the time signal $x(t)$ at different times is related by a joint pdf, which in general depends on the different times. The auto-correlation of $x(t)$, for example, is defined by:

$$R_{xx}(t,u) = E[x(t)x(u)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 p_{x_1,x_2}(x_1,x_2,t,u)dx_1 dx_2$$

It the random process is stationary then the pdf does not depend on time, and the second order joint pdf only depends on the time difference. A good way to think of stationary random processes is that the statistics do not depend on the start time of the time signal. The first and second order statistics of a stationary random process $x(t)$ are defined by:

$$\text{Mean: } \mu_x = E[x(t)] = \int_{-\infty}^{\infty} x p_x(x)dx$$

$$\text{Auto-correlation: } R_{xx}(\tau) = E[x(t)x(t+\tau)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 p_{x_1,x_2}(x_1,x_2,\tau)dx_1 dx_2$$

$$\text{Auto-covariance: } C_{xx}(\tau) = E[(x(t)-\mu_x)(x(t+\tau)-\mu_x)] = R_{xx}(\tau) - \mu_x^2$$

$$\text{Mean square: } \psi_x^2 = R_{xx}(0)$$

$$\text{Variance: } \sigma_x^2 = C_{xx}(0)$$

A less restrictive assumption is that the random process is wide sense stationary (WSS), which means that the mean does not depend on time, and the correlation only depends on the time difference, but no such restrictions are placed on the pdfs and higher moments. For theGaussian random processes being WSS is equivalent to being stationary. WSS random processes are defined by two statistics, the mean and the auto-correlation. The other second order moments can be determined from these two statistics.

Now consider two random processes, $x(t)$ and $y(t)$. Similar definitions to those above apply to $y(t)$ when considered by itself. What is new are the cross-correlation and covariance functions, defined respectively by:

$$R_{xy}(\tau) = E[x(t)y(t+\tau)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy p_{xy}(x,y,\tau)dxdy$$

$$C_{xy}(\tau) = E[(x(t)-\mu_x)(y(t+\tau)-\mu_x)] = R_{xy}(\tau) - \mu_x\mu_y$$

Three properties of the auto- and cross-correlation functions are of interest.

$$|R_{xx}(\tau)| \le R_{xx}(0)$$

$$R_{xx}(-\tau) \le R_{xx}(\tau)$$

$$\left|R_{xy}(\tau)\right|^2 \le R_{xx}(0)R_{yy}(0)$$

In words, the maximum value of the auto-correlation is $\tau = 0$, the auto-correlation is a symmetric function, and a bound can be placed on the cross-correlation. The same three properties are true about covariance functions. Replace R by C in the above equations. Define the correlation coefficient, which depends only on the time difference and is defined by:

$$\rho_{xy}(\tau) = \frac{C_{xy}(\tau)}{\sigma_x \sigma_y}$$

3.1.2 Complex and Vector Random Processes

The random processes $x(t)$ and $y(t)$ can be complex, in which case the second order moments are defined using complex conjugation, for example

$$\text{Cross-correlation: } R_{xy}(\tau) = E[x(t)y^*(t+\tau)]$$

$$\text{Cross-covariance: } C_{xy}(\tau) = R_{xy}(\tau) - \mu_x \mu_y^*$$

The superscript * indicates complex conjugation. The auto-correlation and auto-covariance are real, but the cross-versions are in general complex. If the random processes $x(t)$ and $y(t)$ are vectors, real or complex, then:

$$\text{Cross-correlation: } R_{xy}(\tau) = E[x(t)y'(t+\tau)]$$

$$\text{Cross-covariance: } C_{xy}(\tau) = R_{xy}(\tau) - \mu_x \mu_y'$$

The first order moments are vectors and the second order moments are matrices. The apostrophe y' is defined as the transpose of the complex conjugate, also known as the Hermitian. For the real case the Hermitian is the same as the transpose. We use the apostrophe so that the same notation works for either case. (And this is consistent with how Matlab treats the apostrophe). Note that the same notation also works for complex numbers, since $y^* = y'$.

3.2 Spectral Density Functions

The energy in time signals can be converted to the frequency domain, leading to the definitions of the auto-spectra and cross-spectral density functions.

3.2.1 Definition and Properties

The auto-spectra and cross-spectra are the Fourier transforms of their respective correlation functions:

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau$$

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j2\pi f\tau} d\tau$$

The auto-spectra is real, non-negative, and symmetric (positive semi-definite in the vector case). The cross-spectra is complex, non-negative, and is symmetric if the indices are swapped:

$$S_{xx}(-f) = S_{xx}^*(f) = S_{xx}(f) \ge 0$$

$$S_{xy}(-f) = S_{xy}^{*}(f) = S_{yx}(f) \geq 0$$

The inverse transform relationships are:

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx}(f) e^{j2\pi f\tau} df$$

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} S_{xy}(f) e^{j2\pi f\tau} df$$

Set $\tau = 0$ in the inverse transform and it follows that:

$$\psi_x^2 = R_{xx}(0) = \int_{-\infty}^{\infty} S_{xx}(f) df$$

3.2.2 Non-Zero-Mean Case

The definitions for the auto- and cross-spectra are true whether or not the underlying random processes have zero mean. It is helpful nonetheless to explicitly include the non-zero-mean in the equations, and what will be seen is that a random process with a non-zero mean will have an auto-spectrum with an impulse. The auto-correlation in the non-zero-mean case will be of the form:

$$R_{xx}(\tau) = C_{xx}(\tau) + \mu_x^2$$

where μ_x is the mean and the auto-covariance $C_{xx}(\tau)$ has zero-mean. The auto-spectra is:

$$S_{xx}(f) = S_{xx_0}(f) + \mu_x^2 \delta(f)$$

where $S_{xx_0}(f)$ is computed directly from the auto-covariance:

$$S_{xx_0}(f) = \int_{-\infty}^{\infty} C_{xx}(\tau) e^{-j2\pi f\tau} d\tau$$

It is for this reason that some of the upcoming results will use the auto-covariance in place of the auto-correlation. The variance is:

$$\sigma_x^2 = C_{xx}(0) = \int_{-\infty}^{\infty} S_{xx_0}(f) df$$

Of course all this bother goes away if the random process is assumed to be zero mean.

3.2.3 One-Sided Spectra

The one-sided spectrum is defined just for positive frequency, and is twice as high so that the total area is the same:

$$G_{xx}(f) = \begin{cases} 2S_{xx}(f) & f > 0 \\ S_{xx}(f) & f = 0 \\ 0 & f < 0 \end{cases}$$

Some authors favor the one-sided spectrum, and so the definition is included here. We favor the two-sided version, and will use it outside of this sub-section. It follows via substitution that:

$$G_{xx}(f)=\begin{cases}2\int_{-\infty}^{\infty}R_{xx}(\tau)e^{-j2\pi f\tau}d\tau & f>0\\ \int_{-\infty}^{\infty}R_{xx}(\tau)d\tau & f=0\\ 0 & f<0\end{cases}$$

Not having the "2" in the definitions when $f=0$ is needed to properly handle impulses. In the non-zero-mean case it follows that:

$$G_{xx}(f)=\begin{cases}2S_{xx}(f) & f>0\\ \mu_x^2\delta(f) & f=0\\ 0 & f<0\end{cases}$$

The inverse transform relationship is:

$$R_{xx}(\tau)=\int_0^{\infty}G_{xx}(f)e^{j2\pi f\tau}df$$

The mean square value of $x(t)$ is:

$$\psi_x^2=R_{xx}(0)=\int_0^{\infty}G_{xx}(f)df$$

The definitions are similar for the one-sided $G_{yy}(f)$ and $G_{xy}(f)$.

3.2.4 Spectra Via Finite Fourier Transforms

The spectra will be estimated, in fact must be estimated, from finite records. From the sample record $x(t)$, and similarly for $y(t)$, compute the finite time Fourier transform:

$$X(f,T)=\int_0^T x(t)e^{-j2\pi ft}dt$$

Define the following product:

$$S_{xy}(f,T)=\frac{1}{T}X(f,T)Y^*(f,T)$$

The cross-spectra results from the following limit:

$$S_{xy}(f)=\lim_{T\to\infty}E[S_{xy}(f,T)]$$

It is important to note that the limit is written in terms of the expected value. The above equality needs to be proven. The expectation moves inside the integrals:

$$E[S_{xy}(f,T)]=\frac{1}{T}\int_0^T\int_0^T E[x(\alpha)x(\beta)]e^{-j2\pi f(\beta-\alpha)}d\alpha d\beta$$

Set $\tau=\beta-\alpha$ and it follows after some non-obvious machinations that:

$$E[S_{xy}(f,T)]=\int_{-T}^{T}\left(1-\frac{|\tau|}{T}\right)R_{xy}(\tau)e^{-j2\pi f\tau}d\tau$$

And then in the limit as $T \to \infty$ this integral becomes $S_{xy}(f)$. Similar results hold for the auto-spectra $S_{xx}(f)$ and $S_{yy}(f)$.

3.2.5 Statistical Bandwidth

There are many ways to define bandwidth. Qualitatively, the bandwidth is the width in frequency where the auto-spectrum has most of its power. When defining bandwidth we restrict attention here to a scalar (non-vector) random process. We also assume that the random process has zero mean. The particular definition of bandwidth that works the best for power spectral estimation is the so-called "statistical bandwidth:"

$$B_s = \frac{\left[\int_{-\infty}^{\infty} S_{xx}(f)df\right]^2}{2\int_{-\infty}^{\infty} S_{xx}^2(f)df}$$

This equals:

$$B_s = \frac{R_{xx}^2(0)}{2\int_{-\infty}^{\infty} R_{xx}^2(\tau)\tau}$$

That the numerators are equal follows from the very definition of the auto-spectrum. It is more complicated to show that the denominators are equal:

$$\begin{aligned}\int S_{xx}^2(f)df &= \iiint R_{xx}(\alpha)R_{xx}(\beta)e^{-j2\pi f(\alpha+\beta)}d\alpha d\beta df \\ &= \iint R_{xx}(\alpha)R_{xx}(\beta)\delta(\alpha+\beta)d\alpha d\beta \\ &= \int R_{xx}^2(\tau)d\tau\end{aligned}$$

All of the limits of integration are from minus to plus infinity, and the integral with respect to f is first computed, resulting in the delta function. The next expression for the statistical bandwidth works whether or not the system is zero mean (which we state without proof):

$$B_s = \frac{C_{xx}^2(0)}{2\int_{-\infty}^{\infty} C_{xx}^2(\tau)\tau}$$

We give one more equivalent expression for the statistical bandwidth, which turns out to be the best for computation. Define the random process $x_1(t) = x(t) * x(t)$, which is $x(t)$ convolved with itself. Then:

$$B_s = \frac{R_{xx}^2(0)}{2R_{x_1x_1}(0)}$$

We are not quite ready to prove this, and we do so after better understanding the problem of passing a random process through a linear system.

3.3 Linear System Response

The input of a stable, linear system is defined as a random process, and the question is what can be said about the output. There are several important points:

- The output is also a random process;

- The output spectrum is the input spectrum multiplied by the square of the system frequency response;
- The output variance can be computed using a Lyapunov equation; and
- So can the statistical bandwidth.

Complicating factors are whether or not the input is white noise, has zero mean, and whether or not the system has zero initial conditions. We now present the full story, starting with the definition of white noise.

3.3.1 Definition of White Noise

A white noise random process $w(t)$ has no correlation between $w(t)$ and $w(t+\tau)$, no matter how small the time difference. Think of it this way: if you know the response at time t, this knowledge provides no information about the response at time $t+\tau$, no matter how small the time difference. All that is known at any given time is the mean and variance of the signal. The random process has mean μ_w and auto-correlation:

$$R_{ww}(\tau) = W\delta(\tau) + \mu_w^2$$

The quantity W is called the "intensity," so a quick way to specify the random process is "white noise with mean μ_w and intensity W." The auto-spectra is:

$$S_{xx}(f) = W + \mu_w^2\delta(f)$$

The spectral bandwidth is:

$$B_s = \infty$$

White noise as infinite power, and hence is unrealistic, but white noise passed through a linear system nevertheless turns out to be a very useful and practical model for random processes.

3.3.2 White Noise through a Linear System

The input is white noise, non-zero-mean, and the initial conditions of the system are zero. The statistics of the output are listed in Table 22. The problem is defined above the double lines in Table 22, and the results are below the double lines. The results in Table 22 are a special case of the more general problem, presented next.

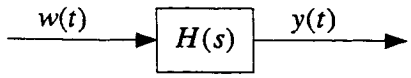
3.3.3 Non-White Noise through a Linear System

The input is a random process, non-zero-mean, and the initial conditions are zero. The statistics of the output are listed in Table 23. The derivation is now presented.

The linear system has impulse response $h(t)$ and Laplace transform $H(s)$. It is traditional to present block diagrams using the Laplace transform, and we will follow this tradition. The Fourier transform $H(f)$ is the Laplace transform evaluated at $s = j2\pi f$. It is with very little extra complication that we define the system very generally as being stable, non-causal, multivariable, and complex. (The complications are basically to keep the order of terms correct and to use an apostrophe now and then to indicate the complex conjugate transpose).

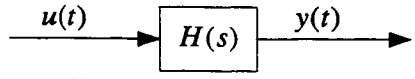
The initial conditions are set to zero. A system with zero initial conditions is equivalent to one that has been operating for a long time, ideally since minus infinity, so that the initial condition response has decayed to zero. Only in this case is the output a stationary random process, and only in this case can the output spectrum even be defined.

Table 22. White Noise Through a Linear System



Input mean:	μ_w
Input auto-correlation:	$R_{ww}(\tau) = W\delta(\tau) + \mu_x\mu_x'$
Input auto-spectra:	$S_{ww}(f) = W + \mu_x\mu_x'\delta(f)$
Output mean:	$\mu_y = H(0)\mu_w$
Output auto-correlation:	$R_{yy}(\tau) = \int_{-\infty}^{\infty} h(\alpha)Wh'(\tau+\alpha)d\alpha + \mu_w\mu_w'$
Output auto-spectra:	$S_{yy}(f) = H(f)WH'(f) + \mu_y\mu_y'\delta(f)$
Cross-correlation:	$R_{yw}(\tau) = h(\tau)W + \mu_y\mu_w'$
Cross-spectra:	$S_{yw}(f) = H(f)W + \mu_w\mu_w'\delta(f)$

Table 23. Non-White Noise Through a Linear System



Input mean:	μ_u
Input auto-correlation:	$R_{uu}(\tau)$
Input auto-spectra:	$S_{uu}(f)$
Output mean:	$\mu_y = H(0)\mu_u$
Output auto-correlation:	$R_{yy}(\tau) = \int_0^{\infty}\int_0^{\infty} h(\alpha)R_{uu}(\tau+\alpha-\beta)h'(\beta)d\alpha d\beta$
Output auto-spectra:	$S_{yy}(f) = H(f)S_{uu}(f)H'(f)$
Cross-correlation:	$R_{yu}(\tau) = \int_{-\infty}^{\infty} h(\alpha)R_{uu}(\tau-\alpha)d\alpha$
Cross-spectra:	$S_{yu}(f) = H(f)S_{uu}(f)$

The statistics of the output $y(t)$ follow from the convolution integral:

$$y(t) = \int_{-\infty}^{\infty} g(\alpha)u(t-\alpha)d\alpha$$

The limits of integration start at minus infinity if the system is non-causal, and start at zero if the system is causal. The limits end at plus infinity because the input is assumed to start at minus infinity. The mean of the output is:

$$\mu_y = E[y(t)] = E\left[\int_0^{\infty} h(\alpha)u(t-\alpha)d\alpha\right]$$

The expectation operator moves inside the integral, and it follows that:

$$\mu_y = \int_0^\infty h(\alpha)E[u(t-\alpha)]d\alpha = \int_0^\infty h(\alpha)\mu_u d\alpha = H(0)\mu_u$$

The auto-correlation of the output is:

$$R_{yy}(\tau) = E[y(t)y'(t+\tau)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\alpha)E[u(t-\alpha)u'(t+\tau-\beta)]h'(\beta)d\alpha d\beta$$

The limits of each of the two integrations start at minus infinity if the system is non-causal, and start at 0 if the system is causal. This completes the derivation of the auto-correlation for the general case. If the input is white noise $u(t) = w(t)$ then:

$$\begin{aligned} R_{yy}(\tau) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\alpha)[W\delta(\tau+\alpha-\beta) - \mu_w\mu_w']h'(\beta)d\alpha d\beta \\ &= \int_{-\infty}^{\infty} h(\alpha)Wh'(\tau+\alpha)d\alpha - \mu_y\mu_y' \end{aligned}$$

The limits of integration start at minus infinity if the system is non-causal, and start at $\min(0,\tau)$ if the system is causal. Note that the auto-correlation in the white noise case is $h(t)W$ convolved with $h'(-t)$.

The auto-spectra result follows in a laborious but straightforward manner using the definition of the Fourier transform. The cross-results follow similarly and more easily using the convolution and the Fourier transform.

3.3.4 Finite Time Response Using State Space Systems

Next treat the more realistic case where a random process is applied to a system with non-zero initial conditions and the system is allowed to run for just a finite time. What we will find out is that the state and the output are random processes, but non-stationary random processes. The results are shown in Table 24. Only in the limit as the time interval extends to infinity do they revert to stationary random processes. In what follows the statistics of the state and output are derived. Probably the most important thing to note is that the steady state variance is the solution of a Lyapunov equation, and this is the best way to actually compute the variance.

It is with no loss of generality that the input is assumed to be white noise. If not, model the non-white noise as the output of white noise through the system $F(s)$, and change the problem to the cascaded system $H(s)F(s)$ driven by white noise. Assuming that the input has zero mean restricts the problem, which is unfortunate, but the many extra complications that arise from removing this restriction do not outweigh the benefits.

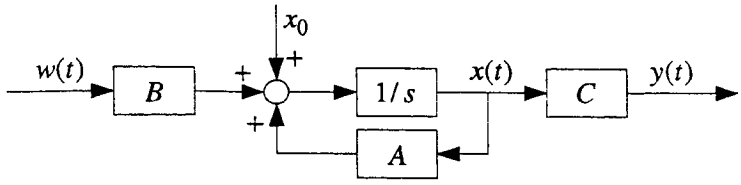
Model the system using state space equations, so that the initial condition can be more easily defined. The state space model is:

$$\begin{aligned} \dot{x} &= Ax + Bw \\ y &= Cx \end{aligned}$$

The Laplace transform of the system is $F(s) = C(sI - A)^{-1}B$. The initial condition is $x(t_0) = x_0$. The state at some time later is found using the Variations of Parameter solution to the state space equation:

$$x(t_1) = e^{A(t_1-t_0)}x(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}Bw(\tau)d\tau$$

Table 24. White Noise Through a State Space System



Input mean:	$\mu_w = 0$
Input auto-correlation:	$R_{ww}(\tau) = W\delta(\tau)$
Initial condition mean:	$\mu_x(t_0)$
Initial condition variance:	$Q_x(t_0)$
Independence:	$C_{xw}(t_0, t_1) = 0$ for $t_1 \geq t_0$
State mean:	$\mu_x(t_1) = e^{A(t_1 - t_0)}\mu_x(t_0)$
State variance:	$\dot{Q}_x(t) = AQ_x(t) + Q_x(t)A' + BWB'$
State auto-covariance	$C_{xx}(t_1, t_2) = \begin{cases} Q_x(t_1)e^{A'(t_2 - t_1)} & t_2 \geq t_1 \\ e^{A(t_1 - t_2)}Q_x(t_1) & t_2 < t_1 \end{cases}$
Output mean:	$\mu_y(t_1) = C\mu_x(t_1)$
Output variance:	$Q_y(t) = CQ_x(t)C'$
Output auto-covariance:	$C_{yy}(t_1, t_2) = CC_{xx}(t_1, t_2)C'$

Note that once the state statistics are known the output statistics follow immediately using matrix multiplication, and hence we need only derive the results for the state. After some algebra it works out that the mean and auto-correlation of the state are respectively:

$$\mu_x(t_1) = e^{A(t_1 - t_0)}\mu_x(t_0)$$

$$C_{xx}(t_1, t_2) = e^{A(t_1 - t_0)}Q_0 e^{A'(t_2 - t_0)} + \int_{t_0}^{\min(t_1, t_2)} e^{A(t_1 - \tau)}BWB'e^{A'(t_2 - \tau)}d\tau$$

The mean changes with time, and hence the random process is non-stationary. Similarly, the auto-covariance depends on both t_1 and t_2, and not just on $t_2 - t_1$. To find the variance set $t = t_1 = t_2$ and then take the derivative with respect to t, and it follows that the variance satisfies the differential equation:

$$\dot{Q}_x(t) = AQ_x(t) + Q_x(t)A' + BWB'$$

with initial condition $Q_x(t_0)$. Substitute back into the integral equation and then:

$$C_{xx}(t_1, t_2) = \begin{cases} Q_x(t_1)e^{A'(t_2 - t_1)} & t_2 \geq t_1 \\ e^{A(t_1 - t_2)}Q_x(t_1) & t_2 < t_1 \end{cases}$$

These results give a way to compute the non-stationary statistics. In the limit as $t \to \infty$ it follows that derivative $\dot{Q}_x(t) = 0$ the state variance settles to a constant:

$$\bar{Q}_x = \int_0^\infty e^{A\tau} BWB' e^{A'\tau} d\tau$$

The variance can be computed from this integral, but a much better way that follows from the differential equation is to compute the variance as the solution of the Lyapunov equation:

$$0 = A\bar{Q}_x + \bar{Q}_x A' + BWB'$$

The input was defined to have zero mean, and the system is stable, so in the limit as $t \to \infty$ it follows that $\mu_x = 0$. The auto-correlation equals the auto-covariance, which is:

$$R_{xx}(\tau) = C_{xx}(\tau) = \begin{cases} \bar{Q}_x e^{A'\tau} & \tau \geq 0 \\ e^{-A\tau} \bar{Q}_x & \tau < 0 \end{cases}$$

Just to be clear we note the different ways to write the variance of the state:

$$\psi_x^2 = R_{xx}(0) = C_{xx}(0) = \bar{Q}_x$$

The variance at the output is:

$$\psi_y^2 = R_{yy}(0) = C_{yy}(0) = \bar{Q}_y = C\bar{Q}_x C'$$

3.3.5 Computing the Statistical Bandwidth

Starting with a state space description of a linear system, Lyapunov equations can be used to compute the variance of the output. Similarly, Lyapunov equations can be used to compute the output variance of the system cascaded with itself, and then these two variances are used to compute the statistical bandwidth. The cascaded system is shown in Figure 125. The input is zero-mean white noise with auto-correlation $R_{ww}(\tau) = W\delta(\tau)$. Set $W = 1$. The statistical bandwidth has only been defined for SISO system, and so we assume that $H(s)$ is such. The claim is that:

$$B_s = \frac{R_{hh}^2(0)}{2R_{h_1 h_1}(0)}$$

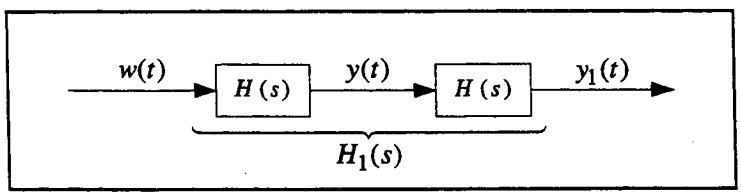


Figure 125. Cascaded System

To prove this claim it needs to be shown that:

$$R_{h_1 h_1}(0) = \int_{-\infty}^{\infty} R_{hh}^2(\tau) d\tau$$

Now we know that $R_{hh}(\tau) = h(\tau) * h'(-\tau)$, where $*$ is the convolution operator, and we know that $h_1(\tau) = h(\tau) * h(\tau)$, and hence:

$$R_{h_1 h_1}(\tau) = h_1(\tau) * h_1'(-\tau) = h(\tau) * h(\tau) * h'(-\tau) * h'(-\tau)$$

Change the order of the convolutions, which can only be done for SISO systems, and then:

$$R_{h_1 h_1}(\tau) = R_{hh}(\tau) * R_{hh}(\tau) = \int_{-\infty}^{\infty} R_{hh}(\alpha) R_{hh}(\tau - \alpha) d\alpha$$

Set $\tau = 0$ and the claim is proved. To finish the computational method a state space realization for $H_1(s)$ is needed. One such realization is shown below:

$$\begin{aligned} \dot{x}_1 &= A_1 v + B_1 w \\ y_1 &= C_1 w \end{aligned} \quad \text{, where} \quad A_1 = \begin{bmatrix} A & 0 \\ BC & A \end{bmatrix} \quad B_1 = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad C_1 = [0 \quad C]$$

The two Lyapunov equations are:

$$0 = AQ + QA' + BB'$$

$$0 = A_1 Q_1 + Q_1 A_1' + B_1 B_1'$$

and finally:

$$B_s = \frac{(CQC')^2}{2 C_1 Q_1 C_1}$$

3.4 Random Process Examples

All of the examples are white noise through a linear time invariant filter. The general case is presented first, followed by two groups of filters. The first group is low pass filters, and the second group is bandpass filters.

3.4.1 Filtered White Noise

White noise is passed through a filter. The filter is linear-time-invariant, stable, SISO, in general non-causal, and complex, with impulse response $h(t)$, Laplace transform $H(s)$, and Fourier transform $H(f)$. The white noise has zero-mean and unit-intensity. The statistics for the general problem are listed in Table 22. The statistics for this version of the problem (SISO, zero mean, unit-intensity) are shown in Table 25. The same table also includes the statistics for a scaled version of the system.

3.4.2 Bandwidth Limited White Noise

The auto-spectrum is an ideal low-pass filter with a bandwidth of B Hz. The mean of the random process can be non-zero. The corresponding filter is non-causal. The statistics of interest are listed in Table 26. The figures in the table use parameter values $\sigma = 1, B = 1, \mu = 0$.

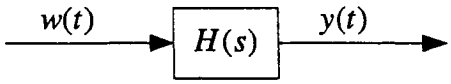
3.4.3 First Order Low Pass Filter

Now consider a causal filter, the easiest being a first order filter:

$$h(s) = \frac{b}{s + a}$$

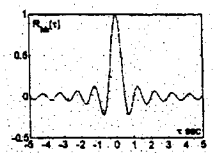
This is a low-pass filter with a filter bandwidth (–3 dB frequency) of $B = a/(2\pi)$ Hz. The input is white noise with zero mean and unit intensity. The statistical quantities of interest are listed in Table 27. The figures in the table use parameter values $B = 1$ and $a = b = 2\pi B$.

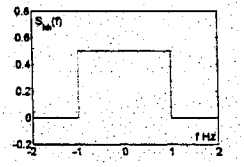
Table 25. Filtered White Noise (SISO system, zero-mean, unit-intensity)



	For the reguar system $H(s)$	For the scaled system $H_1(s)=\beta H(\alpha s)$
Impulse response	$h(t)$	$h_1(t)=\frac{\beta}{\alpha}h\left(\frac{t}{\alpha}\right)$
Auto-correlation	$R_{hh}(\tau)=h(\tau)*h'(-\tau)$	$R_{h_1h_1}(\tau)=\frac{\beta^2}{\alpha}R_{hh}\left(\frac{\tau}{\alpha}\right)$
Auto-spectra	$S_{hh}(f)=\lvert H(f)\rvert^2$	$S_{h_1h_1}(f)=\beta^2 S_{hh}(\alpha f)$
Variance	$\psi_h^2=R_{hh}(0)=\int_{-\infty}^{\infty}S_{hh}(f)df$	$\psi_{h_1}^2=\frac{\beta^2}{\alpha}\psi_h^2$
Dc power	$S_{hh}(0)=H^2(0)=\int_{-\infty}^{\infty}R_{hh}(\tau)d\tau$	$S_{h_1h_1}(0)=\beta^2 S_{hh}(0)$
Statistical bandwidth	$B_s=\frac{R_{hh}^2(0)}{2\int_{-\infty}^{\infty}R_{hh}^2(\tau)d\tau}$	$B_{s_{h_1}}=\frac{1}{\alpha}B_{s_h}$

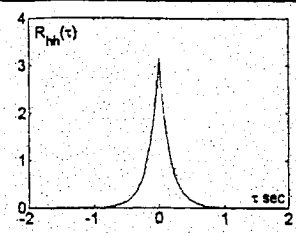
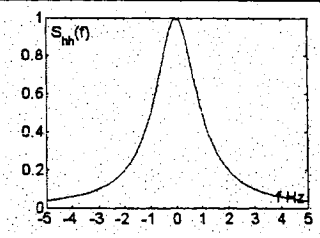
Table 26. Bandwidth Limited White Noise





Impulse response	$h(t)=\sigma\sqrt{2B}\left(\frac{\sin 2\pi Bt}{2\pi B\tau}\right)+\mu$
Auto-correlation	$R_{hh}(\tau)=\sigma^2\left(\frac{\sin 2\pi B\tau}{2\pi B\tau}\right)+\mu^2$
Auto-spectra	$S_{hh}(f)=\begin{cases}\sigma^2/(2B) & 0<\lvert f\rvert\le B\\ \mu^2\delta(f) & f=0\\ 0 & \text{otherwise}\end{cases}$
Variance	$C_{xx}(0)=\sigma^2$
Dc power	$\int_{-\infty}^{\infty}C_{xx}(\tau)d\tau=\frac{\sigma^2}{2B}$
Statistical bandwidth	$B_s=B$

Table 27. First Order Lowpass Filter

	
Impulse response	$h(t)=\begin{cases} be^{-at} & t\geq 0 \\ 0 & t<0 \end{cases}$
Auto-correlation	$R_{hh}(\tau)=\frac{b^2}{2a}e^{-\lvert a\rvert\tau}$
Auto-spectra	$S_{hh}(f)=\frac{b^2}{-s^2+a^2}$ at $s=j2\pi f$
Variance	$R_{hh}(0)=\frac{b^2}{2a}$
Dc power	$S_{hh}(0)=\frac{b^2}{a^2}$
Statistical bandwidth	$B_s=\pi B$

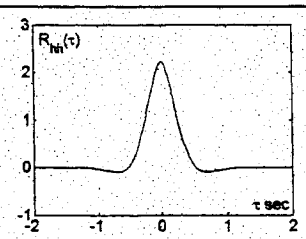
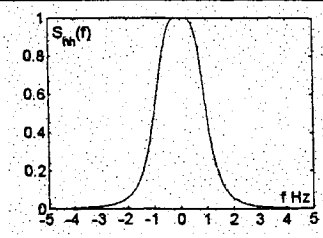
3.4.4 Second Order Low Pass Filter (Complex Poles)

The exact solutions for the statistical quantities become cumbersome beyond just the first order filter. The solution is given nonetheless for the case of a second order lowpass filter with complex poles. The Laplace transform is:

$$h(s)=\frac{\omega^2}{s^2+2\zeta\omega s+\omega^2}$$

The damping ratio is in the range $0<\zeta\leq 1$, being $\zeta=1/\sqrt{2}$ for a Butterworth filter and $\zeta=\sqrt{3}/2$ for a Bessel filter. The bandwidth is $B=a/(2\pi)$ Hz. The statistical quantities of interest are listed in Table 28. The parameter values used for the figures in this table are $B=1$ Hz, $\omega=2\pi B$, and $\zeta=1/\sqrt{2}$.

Table 28. Second Order Lowpass Filter (Complex Poles)

	
Impulse response	$h(t)=\begin{cases}\dfrac{\omega}{\sqrt{1-\zeta^2}}e^{-\zeta\omega t}\sin\left[\omega t\sqrt{1-\zeta^2}\right] & t\geq 0\\ 0 & t<0\end{cases}$
Auto-correlation	$R_{hh}(\tau)=\dfrac{\omega}{4\zeta\sqrt{1-\zeta^2}}e^{-\zeta\omega\lvert t\rvert}\times \cos\left(\omega t\sqrt{1-\zeta^2}-\text{sign}(t)\times\arctan\dfrac{\zeta}{\sqrt{1-\zeta^2}}\right)$
Auto-spectra	$S_{hh}(f)=\left\lvert\dfrac{\omega^2}{s^2+2\zeta\omega s+\omega^2}\right\rvert$ at $s=j2\pi f$
Variance	$R_{hh}(0)=\dfrac{\omega}{4\zeta}$
Dc power	$S_{hh}(0)=1$
Statistical bandwidth	$B_s=\dfrac{2\pi\zeta}{1+4\zeta^2}B$ ($=1.481B$ for Butterworth)

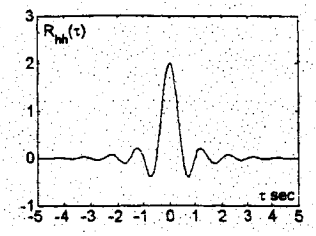
3.4.5 Butterworth Filters

High order Butterworth filters are used for lowpass filters with a sharp cutoff. As the order becomes higher, the Butterworth filter approaches the ideal lowpass filter of the same bandwidth, and therefore we expect the statistical bandwidth to approach the filter bandwidth. Indeed it does, as we see from the values listed in Table 29. The figures in the table are for a 10^{th} order Butterworth filter with $B=1$ Hz.

The Matlab code (Version 6.5 with the control toolbox) used to compute the values in Table 29 is listed below:

```
for n=[1:10,15,20]
  [a,b]=butter(n,2*pi,'s');
  sys=ss(tf(a,b));
  sys2=sys*sys;
  Rhh0=sys.c*lyap(sys.a,sys.b*sys.b')*sys.c';
  Rh2h20=sys2.c*lyap(sys2.a,sys2.b*sys2.b')*sys2.c';
  Bs=Rhh0^2/(2*Rh2h20);
  fprintf('n=%d, Rhh0=%g, Bs=%g\n',n,Rhh0,Bs)
end
```

Table 29. Butterworth Filters



Butterworth order	$R_{hh}(0)$	B_s Hz
1	3.14159 B	3.14159 B
2	2.22144 B	1.48096 B
3	2.0944 B	1.25664 B
4	2.05234 B	1.17277 B
5	2.03328 B	1.1296 B
6	2.02303 B	1.10347 B
7	2.01688 B	1.08601 B
8	2.01291 B	1.07355 B
9	2.01019 B	1.06422 B
10	2.00825 B	1.05697 B
15	2.00366 B	1.03638 B
26	2.00206 B	1.0267 B

3.4.6 Bandpass White Noise

The auto-spectrum is an ideal band-pass of width B Hz centered at f_0 Hz (where $f_0 > B/2$). The mean of the random process can be non-zero. The corresponding filter is non-causal. The statistics of interest are listed in Table 30. The figures in the table use parameter values $\sigma = 1$, $f_0 = 2$, $B = 1$, and $\mu = 0$.

3.4.7 Finite Time Fourier Transform (Rectangular Window)

For the time series $u(t)$, the Fourier Transform computed for the previous T seconds and at the frequency f_0 Hz is:

$$U(f_0,t) = \int_{t-T}^{t} u(\alpha) e^{-j2\pi f_0 \alpha} d\alpha$$

This calculation is equivalent to the convolution $U(f_0,t) = u(t) * h(t)$ for:

$$h(t) = \begin{cases} e^{-j2\pi f_0 (T-t)} & \text{for } 0 \le t < T \\ 0 & \text{otherwise} \end{cases}$$

The statistical properties of this filter are therefore important for studying estimation using the finite time Fourier transform. This transform is approximated by the FFT, and therefore this filter is also of interest for studying estimation using the FFT. This filter is called a "rectangular window" because the envelope of this filter is rectangular, begin equal to one over the time interval $0 \le t < T$ and zero outside this time

interval. The statistical quantities of interest are shown in Table 31. The figures in the table use the parameters $T = 1$ and $f_0 = 2$.

Table 30. Bandpass White Noise

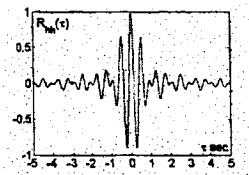
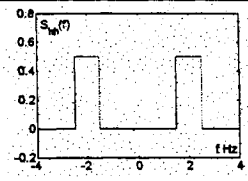
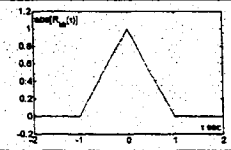
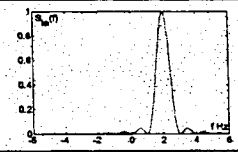
	
Impulse response	$h(t) = \sigma\sqrt{2B}\left(\frac{\sin \pi B t}{\pi B \tau}\right)\cos 2\pi f_0 t + \mu$
Auto-correlation	$R_{hh}(\tau) = \sigma^2\left(\frac{\sin \pi B \tau}{\pi B \tau}\right)\cos 2\pi f_0 \tau + \mu^2$
Auto-spectra	$S_{hh}(f) = \begin{cases} \sigma^2/(2B) & \lvert f \pm f_0 \rvert \le B/2 \\ \mu^2 \delta(f) & f = 0 \\ 0 & \text{otherwise} \end{cases}$
Variance	$C_{xx}(0) = \sigma^2$
Dc power	$\int_{-\infty}^{\infty} C_{xx}(\tau) d\tau = 0$
Statistical bandwidth	$B_s = B$

Table 31. Finite Time Fourier Transform (Rectangular Window)

	
Impulse response	$h(t) = \begin{cases} e^{-j2\pi f_0 (T-t)} & \text{for } 0 \le t < T \\ 0 & \text{otherwise} \end{cases}$
Auto-correlation	$R_{hh}(\tau) = \begin{cases} e^{+j2\pi f_0 \tau}(T - \lvert\tau\rvert) & \text{for } \lvert\tau\rvert \le T \\ 0 & \text{otherwise} \end{cases}$
Auto-spectra	$S_{hh}(f) = \left[\frac{\sin \pi (f - f_0) T}{\pi (f - f_0)}\right]^2$
Variance	$R_{hh}(0) = T$
Dc power	$S_{hh}(0) = \left[\frac{\sin \pi f_0 T}{\pi f_0}\right]^2 = \begin{cases} T^2 & \text{if } f_0 = 0 \\ 0 & \text{if } f_0 T = \text{integer} \neq 0 \end{cases}$
Statistical bandwidth	$B_s = \frac{3}{4T}$

The entries in Table 31 are derived. Start with the Fourier transform of just the envelope of the rectangular window:

$$r(t)=\begin{cases}1 & 0\le t\le T\\ 0 & \text{otherwise}\end{cases}$$

From which it follows that:

$$R(f)=e^{-j\pi fT}\frac{\sin\pi fT}{\pi f}$$

$$R_{rr}(\tau)=\begin{cases}T-|\tau| & 0\le\tau\le T\\ 0 & \text{otherwise}\end{cases}$$

The filter impulse response is

$$h(t)=\alpha e^{j2\pi f_0 t}r(t)\quad\text{for}\quad\alpha=e^{-j2\pi f_0 T}$$

Use the transform identity:

$$H(f)=\alpha R(f-f_0)$$

and hence:

$$H(f)=e^{-j\pi(f+f_0)T}\frac{\sin\pi(f-f_0)T}{\pi(f-f_0)}$$

The auto-spectrum is $S(f)=|H(f)|^2$ and the phase term drops out. Use the same transform identity to find the auto-correlation:

$$R_{hh}(\tau)=|\alpha|^2 e^{j2\pi f_0\tau}R_{rr}(\tau)$$

Finally note that:

$$\int_{-\infty}^{\infty}|R_{hh}(\tau)|^2 d\tau=2\int_0^T(T-\tau)^2 d\tau=\frac{2T^2}{3}$$

and hence:

$$B_s=\frac{R_{hh}^2(0)}{2\int_{-\infty}^{\infty}|R_{hh}(\tau)|^2 d\tau}=\frac{3}{4T}$$

3.4.8 Morlet Wavelet Transform

Complex wavelets:

For the time series $u(t)$, the Wavelet Transform is defined by the convolution:

$$U(f_0,t)=\int_{-\infty}^{\infty}u(\alpha)h(t-\alpha)d\alpha$$

where $h(t)=\psi'(t)$ and $\psi(t)$ is a scaled version of the mother wavelet $\phi(t)$:

$$\psi(t)=\sqrt{f_0}\times\phi(f_0 t)$$

Complex wavelets take the form:

$$\phi(t) = e^{-j2\pi t} g(t)$$

where $g(t)$ is an envelope. The statistics of the filter $h(t)$, which ultimately depend on $g(t)$ and the center frequency f_0, play an important part of estimation using wavelets.

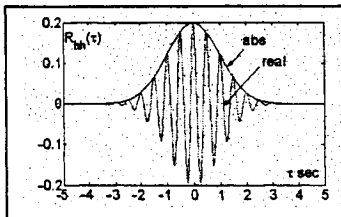
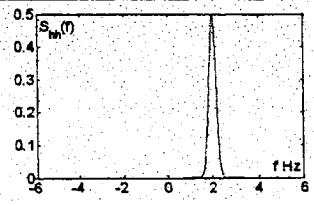
Morlet Wavelet:

The envelope for the Morlet wavelet is the bell-shaped curve:

$$g(t) = \frac{1}{\sqrt{\pi\beta}} e^{-t^2/\beta}$$

This is a non-causal wavelet. The statistics are presented in Table 32. The figures in the table using the parameters $f_0 = 2$ Hz and $\beta = 4$.

Table 32. Morlet Wavelet

	
Impulse response	$h(t) = \sqrt{\frac{f_0}{\pi\beta}} e^{j2\pi f_0 t} e^{-(f_0 t)^2/\beta}$
Auto-correlation	$R_{hh}(\tau) = \frac{1}{\sqrt{2\pi\beta}} e^{j2\pi f_0 \tau} e^{-(f_0\tau)^2/(2\beta)}$
Auto-spectra	$S(f) = \frac{1}{f_0} \exp\left[-2\pi\beta\left(\frac{f - f_0}{f_0}\right)^2\right]$
Variance	$R_{hh}(0) = 1/\sqrt{2\pi\beta}$
Dc power	$S(0) = \exp(-2\pi\beta)/f_0$
Statistical bandwidth	$B_s = f_0/\sqrt{4\pi\beta}$

Derivation:

The derivations of the Morlet wavelet statistics are now presented. Using a Fourier transform table we find that the Fourier transform of the envelope is:

$$G(f, \beta) = e^{-\beta(\pi f)^2}$$

Start with some transform identities:

$$H(f,\beta)=\frac{1}{\alpha}G\left(\frac{f-f_0}{\alpha^2},\beta\right)$$

$$S(f)=\frac{1}{\alpha^2}\left|G\left(\frac{f-f_0}{\alpha^2}\right)\right|^2$$

$$R_{hh}(\tau)=\alpha^2 e^{j2\pi\alpha^2\tau}g(\alpha^2 t)*g(-\alpha^2 t)$$

$$R_{hh}(0)=\int_{-\infty}^{\infty}g^2(\tau)d\tau$$

The β parameter is called out as a separate parameter, and $\alpha=\sqrt{f_0}$ is defined to reduce the notational clutter. The mean square value is computed just from the envelope:

$$R_{hh}(0)=\frac{1}{\pi\beta}\times\int_{-\infty}^{\infty}e^{-2t^2/\beta}d\tau=\frac{1}{\pi\beta}\times\sqrt{\frac{\beta}{2}}\times\underbrace{\int_{-\infty}^{\infty}e^{-s^2}d\tau}_{\sqrt{\pi}}=\frac{1}{\sqrt{2\pi\beta}}$$

By substitution it follows that:

$$h(t,\beta)=\frac{\alpha}{\sqrt{\pi\beta}}e^{j\alpha^2 t}e^{-(\alpha^2 t)^2/\beta}$$

$$H(f,\beta)=\frac{1}{\alpha}\exp\left[-\beta\pi^2\left(\frac{f-f_0}{\alpha^2}\right)^2\right]$$

The auto-spectrum works out to be:

$$S(f)=\left|H(f,\beta)\right|^2=\frac{1}{\alpha}H(f,2\beta)$$

This simplification is thanks to the properties of the exponential. The auto-correlation can therefore be found in terms of the filter impulse response:

$$R_{hh}(\tau)=\frac{1}{\alpha}h(t,2\beta)=\frac{1}{\sqrt{2\pi\beta}}e^{j\alpha^2\tau}e^{-(\alpha^2\tau)^2/(2\beta)}$$

More work is needed to find the statistical bandwidth. It follows that:

$$R_{h^2h^2}(\tau)=F^{-1}\left[S^2(f)\right]=F^{-1}\left[H^4(f)\right]=\frac{1}{\alpha^3}H(f,4\beta)$$

from which:

$$R_{h^2h^2}(0)=\frac{1}{f_0\sqrt{4\pi\beta}}$$

and the statistical bandwidth works out to be:

$$B_s=\frac{R_{hh}^2(0)}{2R_{h^2h^2}(0)}=\frac{1}{\sqrt{4\pi\beta}}f_0$$

This finishes the derivation of the results in Table 32.

Bandwidths and Standard Deviations:

The filter impulse response $h(t)$ and its transform $H(f)$ are both bell-shaped curves, and it helps to understand curves by examining their standard deviations:

$$\sigma[h(t)] = \sqrt{\frac{\beta}{2}} \times \frac{1}{f_0}$$

$$\sigma[H(f)] = \sqrt{\frac{2}{\beta}} \times \frac{f_0}{2\pi}$$

$$\sigma[S_{hh}(f)] = \sqrt{\frac{1}{\beta}} \times \frac{f_0}{2\pi}$$

Interpretations:

- In the time domain, the number of cycles in one standard deviation of $h(t)$ is $\sqrt{\beta/2}$. For example, if $\beta = 4$ there are two cycles in one standard deviation.
- In the frequency domain, the center frequency of $H(f)$ is f_0, and the standard deviation is f_0 times $\sqrt{2/\beta}/(2\pi)$. For example, if $\beta = 4$ and $f_0 = 10$ then the standard deviation is $f_0/(2\pi\sqrt{2}) \approx 0.11 f_0 = 1.1$ Hz.
- The statistical bandwidth is $B_s = \sqrt{\pi/2} \times \sigma[H(f)]$, which is about 25% larger than one standard deviation of $H(f)$. For example, if $\beta = 4$ and $f_0 = 10$ then the statistical bandwidth is about 1.4 Hz.
- The center frequency of the auto-spectrum $S(f)$ is the same frequency, f_0, but being the square of $H(f)$, the bandwidth is narrower, equal to f_0 times $\sqrt{1/\beta}/(2\pi)$.
- The standard deviation in the frequency domain, and likewise the statistical bandwidth, scales with frequency. This is one way to distinguish wavelets from Fourier transforms. The bandwidth for the finite time Fourier transform does not depend on frequency.

4. ESTIMATING STATISTICS OF RANDOM PROCESSES

Estimates are made of the mean, the mean square, and the variance of a random process. Each of these is a random variable with its own statistics. A narrowband filter is then used to estimate the auto-spectrum of the random process.

4.1 Estimate of the Mean

The stationary, real random process $x(t)$ has mean μ and auto-correlation $R_{xx}(\tau)$. The estimate of the mean is:

$$\hat{\mu} = \frac{1}{T} \int_0^T x(t) dt$$

The moments and errors of this estimate are listed in Table 33.

4.1.1 Derivations

The expectation operation moves inside the integral, so that the expected value of the estimated mean is:

$$E[\hat{\mu}] = E\left[\frac{1}{T}\int_0^T x(t)dt\right] = \frac{1}{T}\int_0^T E[x(t)]dt = \frac{1}{T}\int_0^T \mu dt = \mu$$

Table 33. Estimate of the Mean of a Stationary Random Process

In general:	
Mean	$E[\hat{\mu}] = \mu$
Bias	$b[\hat{\mu}] = 0$ (unbiased)
Variance	$\text{var}[\hat{\mu}] \approx \frac{1}{T}\int_{-\infty}^{\infty} C_{xx}(\tau)d\tau$ (consistent)
Variance (zero mean)	$\text{var}[\hat{\mu}] \approx S_{xx}(0)/T$ (consistent)
Rms error (zero mean)	$\text{rms}[\hat{\mu}] = \sqrt{S_{xx}(0)/T}$
Normalized error (zero mean)	$\varepsilon[\hat{\mu}] = (1/\mu)\sqrt{S_{xx}(0)/T}$
For bandwidth limited white noise:	
Variance	$\text{var}[\hat{\mu}] \approx \sigma^2/(2BT)$
Normalized error:	$\varepsilon[\hat{\mu}] \approx (\sigma/\mu)/\sqrt{2BT}$
For bandpass white noise:	
Variance	$\text{var}[\hat{\mu}] \approx 0$
Normalized error:	$\varepsilon[\hat{\mu}] \approx 0$

The mean square of the estimated mean is:

$$E[\hat{\mu}^2] = E\left[\frac{1}{T}\int_0^T x(t)dt\right]^2 = \frac{1}{T^2}\int_0^T\int_0^T E[x(t)x(u)]dudt = \frac{1}{T^2}\int_0^T\int_0^T R_{xx}(u-t)dudt$$

Set $\tau = u - t$ and it follows non-obviously that:

$$E[\hat{\mu}^2] = \frac{1}{T}\int_{-T}^{T}\left(1 - \frac{|\tau|}{T}\right)R_{xx}(\tau)d\tau$$

This is the exact solution, which it approximated for large T by:

$$E[\hat{\mu}^2] \approx \frac{1}{T}\int_{-T}^{T} R_{xx}(\tau)d\tau = \frac{1}{T}S_{xx}(0)$$

To determine the variance of the estimated mean use a similar derivation but with $C_{xx}(\tau)$:

$$\text{var}[\hat{\mu}^2] = \frac{1}{T}\int_{-T}^{T}\left(1-\frac{|\tau|}{T}\right)C_{xx}(\tau)d\tau \approx \frac{1}{T}\int_{-T}^{T}C_{xx}(\tau)d\tau$$

The results for bandwidth limited white noise follows because $S_{xx}(0) = \sigma^2/(2B)$.

4.2 Estimate of the Mean Square

The stationary, real random process $x(t)$ has mean μ, mean square $\psi^2 = R_{xx}(0)$, and auto-correlation $R_{xx}(\tau)$. The estimate of the mean square is:

$$\hat{\psi}^2 = \frac{1}{T}\int_0^T x^2(t)dt$$

The moments and errors of this estimate are listed in Table 34.

Table 34. Estimate of the Mean Square of a Stationary Random Process

In general:	
Mean	$E[\hat{\psi}^2] = \psi^2$
Bias	$b[\hat{\psi}^2] = 0$ (unbiased)
Variance (Gaussian)	$\text{var}[\hat{\psi}^2] \approx \frac{2}{T}\int_{-\infty}^{\infty}\left[C_{xx}^2(\tau) + 2\mu^2 C_{xx}(\tau)\right]d\tau$ (consistent)
Variance (Gaussian, zero mean)	$\text{var}[\hat{\psi}^2] \approx \frac{2}{T}\int_{-\infty}^{\infty}S_{xx}^2(f)df = \frac{2}{T}\int_{-\infty}^{\infty}C_{xx}^2(\tau)d\tau$
Rms error (Gaussian, zero mean)	$\text{rms}[\hat{\psi}^2] = \sqrt{\text{var}[\hat{\psi}^2]}$
Normalized error (Gaussian, zero mean)	$\varepsilon[\hat{\psi}^2] = 1/\sqrt{B_s T}$, $B_s = C_{xx}^2(0)/[2\int_{-\infty}^{\infty}C_{xx}^2(\tau)d\tau]$
For bandwidth limited white noise:	
Variance	$\text{var}[\hat{\psi}^2] \approx (\sigma^4 + 2\mu^2\sigma^2)/(BT)$
Normalized error (zero mean)	$\varepsilon[\hat{\mu}] \approx 1/\sqrt{BT}$
For bandpass white noise:	
Variance	$\text{var}[\hat{\psi}^2] \approx \sigma^4/(BT)$
Normalized error (zero mean)	$\varepsilon[\hat{\mu}] \approx 1/\sqrt{BT}$

4.2.1 Derivations

The expected value of the estimated mean square is:

$$E[\hat{\psi}^2] = E\left[\frac{1}{T}\int_0^T x^2(t)dt\right] = \frac{1}{T}\int_0^T E[x^2(t)]dt = \frac{1}{T}\int_0^T \psi^2 dt = \psi^2$$

The estimate is unbiased. The mean square of the estimate is:

$$E[\hat{\psi}^4] = E\left[\frac{1}{T}\int_0^T x^2(t)dt\right]^2 = \frac{1}{T^2}\int_0^T\int_0^T E[x^2(t)x^2(u)]dudt$$

At this point assume the random process is Gaussian, and then from Table 16:

$$E[x^2(t)x^2(u)] = \psi^4 + 2[C_{xx}^2(u-t) + 2C_{xx}(u-t)\mu^2]$$

Substitute, set $\tau = u - t$, and it follows that:

$$E[\hat{\psi}^4] = \frac{2}{T}\int_{-T}^{T}\left(1 - \frac{|\tau|}{T}\right)\left[C_{xx}^2(\tau) + 2C_{xx}(\tau)\mu^2\right]d\tau + \psi^4$$

This is the exact solution, which is approximated for large T by:

$$E[\hat{\psi}^4] \approx \frac{2}{T}\int_{-\infty}^{\infty}[C_{xx}^2(\tau) + 2C_{xx}(\tau)\mu^2]d\tau + \psi^4$$

Next the variance:

$$\text{var}[\hat{\psi}^2] = E[\hat{\psi}^4] - E^2[\hat{\psi}^2]$$

And since $E^2[\hat{\psi}^2] = \psi^4$ it follows that:

$$\text{var}[\hat{\psi}^2] \approx \frac{2}{T}\int_{-\infty}^{\infty}[C_{xx}^2(\tau) + 2C_{xx}(\tau)\mu^2]d\tau$$

4.3 Estimate of the Variance

The stationary, real random process $x(t)$ has mean μ, variance $C_{xx}(0)$, and auto-correlation $R_{xx}(\tau)$. The estimate of the variance is:

$$\hat{\sigma}^2 = \frac{1}{T}\int_0^T [x(t) - \hat{\mu}]^2 dt$$

The moments and errors of this estimate are listed in Table 35.

Table 35. Biased Estimate of the Variance of a Stationary Random Process

In general:	
Mean	$E[\hat{\sigma}^2] = \sigma^2 - \text{var}[\hat{\mu}]$
Bias	$b[\hat{\sigma}^2] = -\text{var}[\hat{\mu}]$ (biased)
Variance (Gaussian)	$\text{var}[\hat{\sigma}^2] \approx \frac{2}{T}\int_{-\infty}^{\infty} C_{xx}^2(\tau)d\tau - 2\text{var}^2[\hat{\mu}]$ (consistent)
Rms error (Gaussian)	$\text{rms}[\hat{\sigma}^2] = \sqrt{\text{var}[\hat{\sigma}^2]}$
Normalized error (Gaussian)	$\varepsilon[\hat{\sigma}^2] = 1/\sqrt{B_s T}$, $B_s = C_{xx}^2(0)/[2\int_{-\infty}^{\infty} C_{xx}^2(\tau)d\tau]$
For bandwidth limited white noise:	
Variance	$\text{var}[\hat{\sigma}^2] \approx \sigma^4/(BT) - \sigma^4/[2(BT)^2]$

Normalized error (zero mean)	$\varepsilon[\hat{\mu}] \approx 1/\sqrt{BT}$
For bandpass white noise:	
Variance	$\text{var}[\hat{\psi}^2] \approx \sigma^4/(BT)$
Normalized error (zero mean)	$\varepsilon[\hat{\mu}] \approx 1/\sqrt{BT}$

4.3.1 Derivations

The expected value of the estimated variance is:

$$E[\hat{\sigma}^2] = E\left[\frac{1}{T}\int_0^T [x(t) - \mu + \mu - \hat{\mu}]^2 dt\right] = \sigma^2 - \text{var}[\hat{\mu}]$$

The estimate is biased. The variance of the estimate is more difficult. Follow the method used earlier for random variables. The variance expands to:

$$\text{var}[\hat{\sigma}^2] = E[\hat{\sigma}^4] - E^2[\hat{\sigma}^2]$$

where:

$$E[\hat{\sigma}^4] = E[(\hat{\psi}^2 - \hat{\mu}^2)^2] = E[\hat{\psi}^4] - 2E[\hat{\psi}^2\hat{\mu}^2] + E[\hat{\mu}^4]$$

It follows non-easily that:

$$E[\hat{\psi}^4] = \frac{1}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[x^2(\alpha)x^2(\beta)]d\alpha d\beta$$

$$= \frac{2}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C_{xx}^2(\beta - \alpha)d\alpha d\beta + 4\,\text{var}[\hat{\mu}]\mu^2 + \psi^2$$

$$E[\hat{\psi}^2\hat{\mu}^2] = \frac{1}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[x^2(\alpha)x(\beta)x(\gamma)]d\alpha d\beta d\gamma$$

$$\approx 2\,\text{var}^2[\hat{\mu}] + (\sigma^2 + 5\mu^2)\,\text{var}[\hat{\mu}] + \mu^4 + \mu^2\sigma^2$$

$$E[\hat{\mu}^4] = \frac{1}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[x(\alpha)x(\beta)x(\gamma)x(\delta)]d\alpha d\beta d\gamma d\delta$$

$$= 3\,\text{var}^2[\hat{\mu}] + \mu^4 + 6\mu^2\,\text{var}[\hat{\mu}]$$

The approximation in $E[\hat{\psi}^2\hat{\mu}^2]$ is:

$$\frac{2}{T^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C_{xx}(\beta - \alpha)C_{xx}(\gamma - \alpha)d\alpha d\beta d\gamma \approx 2\,\text{var}^2[\hat{\mu}]$$

Add the terms to get:

$$E[\hat{\sigma}^4] = \sigma^4 + \frac{2}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C_{xx}^2(\beta - \alpha)d\alpha d\beta - 2\psi^2\,\text{var}[\hat{\mu}] - \text{var}^2[\hat{\mu}]$$

Subtract $E^2[\hat{\sigma}^2]$, and finally:

$$\text{var}[\hat{\sigma}^4] = \frac{1}{T^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C_{xx}^2(\beta - \alpha)d\alpha d\beta - 2\,\text{var}^2[\hat{\mu}]$$

4.4 Estimating the Auto-Spectrum

The objective is to estimate the auto-spectrum of the random process $x(t)$. The method use to do this is shown in Figure 126. The statistics of the estimate $\hat{S}_{xx}(f_0)$ are listed in Table 36. There is a fundamental tradeoff involving the statistical bandwidth B_s of the narrowband filter. Reducing B_s reduces the bias and increases the variance.

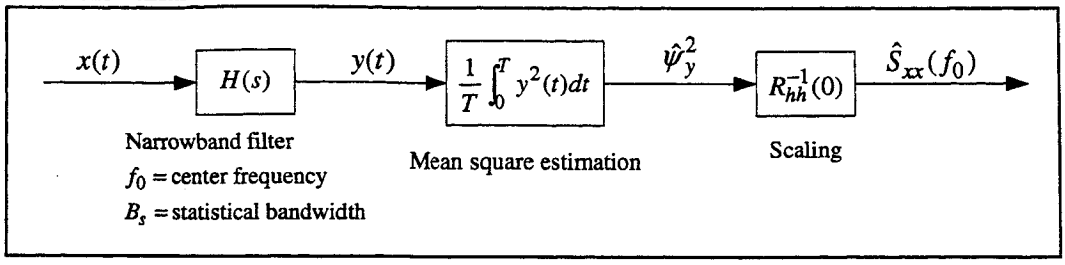


Figure 126. Auto-Spectrum Estimation

Table 36. Auto-Spectrum Estimate

Mean	$E[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}(0)} \int_{-\infty}^{\infty} S_{hh}(f) S_{xx}(f) df$
Bias	$b[\hat{S}_{xx}(f_0)] \approx \frac{B_s^2}{24} S''(f_0)$ (biased)
Variance (Gaussian)	$\text{var}[\hat{S}_{xx}(f_0)] \approx \frac{1}{B_s T} S_{xx}^2(f_0)$ (consistent)
Rms error (Gaussian)	$\text{rms}[\hat{S}_{xx}(f_0)] \approx \sqrt{\frac{1}{B_s T} S_{xx}^2(f_0) + b^2[\hat{S}_{xx}(f_0)]}$
Normalized error (Gaussian)	$\varepsilon[\hat{S}_{xx}(f_0)] \approx \sqrt{\frac{1}{B_s T} + \frac{b^2[\hat{S}_{xx}(f_0)]}{S_{xx}^2(f_0)}}$

4.4.1 Derivations

Overview:

The method for estimating the auto-spectrum is shown in Figure 126 and defined by the following steps:

- $x(t)$ is a zero-mean random process with auto-correlation $R_{xx}(\tau)$ and auto-spectrum $S_{xx}(f)$. It is $S_{xx}(f)$ that is to be estimated at f_0 Hz.
- $h(t)$ is the impulse response of a narrowband filter, with Fourier transform $H(f)$, centered at f_0 Hz and with a statistical bandwidth of B_s Hz.
- The output of the narrowband filter is the random process $y(t)$, which will have zero mean, auto-spectrum $S_{yy}(f) = |H(f)|^2 S_{xx}(f)$, and mean square estimate:

$$\hat{\psi}_y^2 = \frac{1}{T} \int_0^T |y(t)|^2 dt$$

- The estimate of the auto-spectrum at the center frequency is:

$$\hat{S}_{xx}(f_0) = \frac{1}{R_{hh}(0)} \hat{\psi}_y^2$$

Expected Value:

We need to show that $\hat{S}_{xx}(f_0)$ is indeed an estimate of $S_{xx}(f_0)$. The expected value of the estimate is:

$$E[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}(0)} E[\hat{\psi}_y^2]$$

where:

$$E[\hat{\psi}_y^2] = \psi_y^2 = R_{yy}(0) = \int_{-\infty}^{\infty} S_{yy}(f)df = \int_{-\infty}^{\infty} S_{hh}(f)S_{xx}(f)df$$

Substituting:

$$E[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}(0)} \int_{-\infty}^{\infty} S_{hh}(f)S_{xx}(f)df$$

And therefore we see that $\hat{S}_{xx}(f_0)$ is an unbiased estimate of this integrated and scaled version of $S_{xx}(f)$. Approximate the input auto-spectrum as a constant around the center frequency of $H(f)$:

$$S_{xx}(f) \approx S_{xx}(f_0)$$

Then:

$$E[\hat{S}_{xx}(f_0)] \approx S_{xx}(f_0) \frac{1}{R_{hh}(0)} \int_{-\infty}^{\infty} S_{hh}(f)df$$

The mean square value $R_{hh}(0)$ equals the remaining integral of $S_{hh}(f)$, and therefore these terms cancel, leaving:

$$E[\hat{S}_{xx}(f_0)] \approx S_{xx}(f_0)$$

And so we have shown that $\hat{S}_{xx}(f_0)$ is a biased estimate of $S_{xx}(f_0)$.

Bias: The bias is defined by:

$$b[\hat{S}_{xx}(f_0)] = E[\hat{S}_{xx}(f_0)] - S_{xx}(f_0)$$

which equals:

$$b[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}(0)} \int_{-\infty}^{\infty} S_{hh}(f)S_{xx}(f)df - S_{xx}(f_0)$$

This is an exact expression, and we look for an approximation. Set $H(f)$ to the ideal bandpass filter, as defined in Table 30, which has $S_{hh}(f) = \sigma^2/(2B)$ over then bandpass frequencies, $R_{hh}(0) = \sigma^2$, and statistical bandwidth $B_s = B$. In this case:

$$b[\hat{S}_{xx}(f_0)] = \frac{1}{B_s} \int_{f_0 - B_s/2}^{f_0 + B_s/2} S_{xx}(f)df - S_{xx}(f_0)$$

Replace $S_{xx}(f)$ with a third order Taylor series approximation:

$$S_{xx}(f) \approx S_{xx}(f_0) + S'_{xx}(f_0)(f - f_0) + S''(f_0)(f - f_0)^2 / 2$$

All of the terms drop out except:

$$b[\hat{S}_{xx}(f_0)] \approx \frac{1}{2B_s} S''_{xx}(f_0) \int_{f_0 - B_s/2}^{f_0 + B_s/2} (f - f_0)^2 df$$

from which:

$$b[\hat{S}_{xx}(f_0)] \approx \frac{B_s^2}{24} S''_{xx}(f_0)$$

We note that the bias error is small where the auto-spectrum is smooth, and possibly quite large where the auto-spectrum has sharp peaks and valleys.

Variance:

The variance of the estimate $\hat{S}_{yy}(f_0)$ follows directly from the variance of the estimate $\hat{\psi}_y^2$. No new derivations are needed. The exact expression is:

$$\text{var}[\hat{S}_{xx}(f_0)] = \frac{1}{R_{hh}^2(0)} \times \text{var}[\hat{\psi}_y^2] = \frac{1}{R_{hh}^2(0)} \times \frac{2}{T} \int_{-T}^{T} \left(1 - \frac{|\tau|}{T}\right) R_{yy}^2(\tau) d\tau$$

Again we look for an approximation. Let the time get large, so that:

$$\text{var}[\hat{S}_{xx}(f_0)] \approx \frac{1}{R_{hh}^2(0)} \times \frac{2}{T} \int_{-\infty}^{\infty} R_{yy}^2(\tau) d\tau = \frac{1}{R_{hh}^2(0)} \times \frac{2}{T} \int_{-\infty}^{\infty} S_{hh}^2(f) S_{xx}^2(f) df$$

Approximate the input auto-spectrum as a constant around the center frequency of $H(f)$:

$$S_{xx}(f) \approx S_{xx}(f_0)$$

so that:

$$\text{var}[\hat{S}_{xx}(f_0)] \approx \frac{1}{R_{hh}^2(0)} \times \frac{2}{T} \int_{-\infty}^{\infty} S_{hh}^2(f) df \times S_{xx}^2(f_0)$$

What is left is more succinctly written in terms of the statistical bandwidth:

$$\text{var}[\hat{S}_{xx}(f_0)] \approx \frac{1}{B_s T} S_{xx}^2(f_0)$$

APPENDIX B – TRANSFER FUNCTION ESTIMATION

1. INTRODUCTION

Using data to estimate a transfer function, or a state space system, is one of the great problems of system theory, which you might think is by now a solved problem, but actually is still the subject of a considerable amount of research. Whatever works for one example, doesn't work for the next, or is not fast enough, or is very sensitive, or falls apart when the problem becomes large, or does not adapt to changes in the system, and so further research is needed. You quickly get to the point where a set of estimation methods is needed, with some knowledge about when one method should be used, and not the other.

The problem of estimating the parameters of a transfer function is treated here in a comprehensive way. Estimation of state space systems is also covered, but the major focus here is on transfer functions. If we can get that right, getting to the state space system requires just a slight modification of the method. We cover just the single-input single-output problem, which is difficult enough, and again if done right, the multivariable extensions are just another detail.

The development starts with some background on the mathematical tools. The use of the pseudo-inverse to solve linear least square minimization is a standard part of linear algebra (e.g., Ref. 39). The extension of the problem to errors that have a nonlinear dependence on parameters uses Gauss-Newton-Levenberg-Marquardt methods as detailed in Ref. 40. The background on orthogonal polynomials is from Ref. 41.

The estimation problem starts with input and output data:

$$u_i = \text{input data}$$

$$y_i = \text{output data}$$

defined for a set of time steps, and the object is to estimate the parameters of the z-transform:

$$G(z) = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0}$$

Different parameter estimation methods are used from the main categories shown in Figure 127.

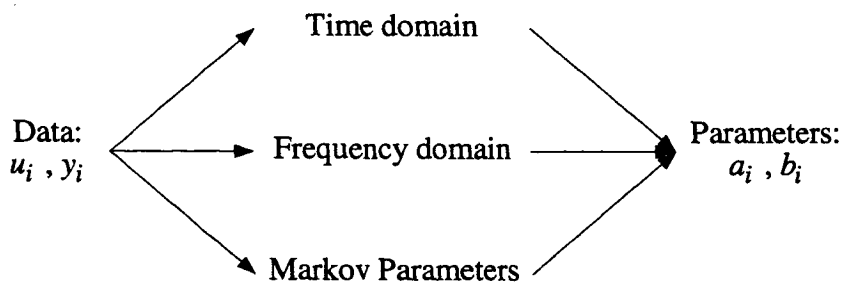


Figure 127. Estimation Methods

Time domain methods work directly with the input and output data. This is where the development starts, using the one-step prediction error method (e.g., Ref. 42). Frequency domain methods start with a frequency response, from which the parameters of the transfer function are estimated. A good set of papers that cover this version of the problem is Refs. 25, 43, 44. Markov parameters methods first estimate Markov parameters, which are the impulse response of the system, and from which the transfer function parameters are estimated. Some background material on Markov parameters is from Ref. 45, including their use to estimate transfer function parameters. Using Markov parameters to estimate state

space systems using the Eigensystem Realization Algorithm is covered in Refs. 24 and 46. Using discrete wavelets to estimate Markov parameters from time series data is discussed in Refs. 26 and 47.

The different approaches are unified here and we show conditions where the time, frequency and Markov parameter approaches give exactly the same answer. They all have the same goal, so there ought to be relationships, but these relationships are not an aspect of the problem that receives much attention in the literature. The unifying concept is the minimization of an error, and the different methods basically amount to different weights on the error.

The treatment here concentrates on several aspects of the estimation problem that often cause difficulty: estimation of the initial conditions, conditioning of the signals, and estimation of system order. Usually the assumption is made that the system starts with zero initial conditions, and when that is not actually the case, as it usually is not, answers go awry. One way around this difficulty is to estimate the initial conditions along with the system parameters, a harder problem, but solvable. Trouble often happens when signals are corrupted by noise, or by the influence of other inputs, or by the selection of sample size, or by the length of the time series. Some methods are not sensitive to these affects, and others, well, quickly go south. Signal conditioning includes filtering, or the use of Fourier transforms, or more recently and one of our areas of interest, the use of Wavelet transforms.

2. MATHEMATICAL TOOLS

The parameter estimation problem is posed as a problem where an error is minimized. The error is a vector, and it is the Euclidean norm of the vector that is minimized. The Euclidean norm is the square root of the sum of the elements squared, and hence, the parameter estimation problem is a "least square" problem. The mathematical tools that are reviewed here are:

- **Difference Equations:** The parameters to be estimated are coefficients of a linear shift invariant difference equation, and we start by reviewing the general solution of this equation.
- **Linear Least Square:** The error is defined as a linear function of the parameters to be estimated. The solution proceeds in one step by computing the pseudo-inverse of a matrix. The Singular Value Decomposition (SVD) is used to compute the pseudo-inverse.
- **Nonlinear Least Square:** The error is a nonlinear function of the parameters, and an iterative minimization of the nonlinear error function is needed. The first derivative of the error with respect to each parameter is used in the solution. The Linear Least Square problem is a special case.
- **Orthogonal Polynomials:** A linear combination of the parameters can be estimated, from which original parameters can be determined. One way to do this is to use orthogonal polynomials. This is done to reduce numerical errors.

There are additional mathematical tools needed to help with the estimation problem, which are used to define a weighted error. These additional tools include filtering, Fourier transforms, and wavelet transforms.

2.1 Difference Equations

The linear, shift invariant difference equation is:

$$y_k = -a_{n-1}y_{k-1} - \cdots - a_0 y_{k-n} + b_m u_{k-n+m} + \cdots + b_0 u_{k-n}$$

The output y_k depends on previous inputs and outputs. Assume the coefficients are known and the objective is to determine the solution of the difference equation. We are interested in the one-sided

solution, y_k for $k \geq 0$, given the input u_k for $k \geq 0$ and the initial conditions defined as y_k for $-n \leq k < 0$. Problems that do not start at zero can be shifted to zero. The total solution equals a forced solution plus an initial condition solution:

$$\underbrace{y_k}_{total} = \underbrace{y_{f_k}}_{forced} + \underbrace{y_{ic_k}}_{\substack{initial \\ condition}} \quad \text{for } k \geq 0$$

The forced solution solves the difference equation as stated above with zero initial conditions. The initial condition solution solves just the homogenous (no input) part of the difference equation:

$$y_k = -a_{n-1}y_{k-1} - \cdots - a_0 y_{k-n}$$

The initial conditions can be defined by n values of the output at any set of n times, not necessarily the times before the input is applied. We do so, however, because it is appropriate for simulation. The initial conditions can be packed into a vector:

$$y_{ic} = \begin{bmatrix} y_{-n} \\ \vdots \\ y_{-1} \end{bmatrix}$$

Qualitatively, the output for $k \geq 0$ depends on the input from that time onwards plus an unforced part that somehow carries over from what happened before the input was applied.

2.1.1 Solution using z-Transforms

The use of z-transforms is usually accompanied with an assumption of zero initial conditions, but they can be used to determine the total solution. To find the forced solution, make the assumption of zero initial conditions and then the z-transform of the system is:

$$\frac{y(z)}{u(z)} = G(z) = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1}z^{n-1} \cdots + a_0}$$

The z-transform of the input time series is:

$$u(z) = Z\{u_k\} = \sum_{k=0}^{\infty} u_k z^{-k}$$

Multiply $y(z) = g(z)u(z)$ and compute its inverse z-transform:

$$y_k = Z^{-1}\{y(z)\}$$

To find the initial condition response, use the z-transform identity for shifted signals:

$$Z\{y_{k-1}\} = \sum_{k=0}^{\infty} y_{k-1}z^{-k} = y_{-1} + z^{-1}Y(z)$$

$$Z\{y_{k-2}\} = \sum_{k=0}^{\infty} y_{k-2}z^{-k} = y_{-2} + z^{-1}y_{-1} + z^{-2}Y(z)$$

$$\vdots$$

$$Z\{y_{k-n}\} = \sum_{k=0}^{\infty} y_{k-n} z^{-k} = y_{-n} + z^{-1} y_{-n+1} + \cdots + z^{-n+1} y_{-1} + z^{-n} Y(z)$$

Substitute this expansion into the homogenous difference equation. For a third order system:

$$y(z) = \frac{-(a_2 + a_1 z^{-1} + a_0 z^{-2}) y_{-1} - (a_1 + a_0 z^{-1}) y_{-2} - a_0 y_{-3}}{1 + a_2 z^{-1} + a_1 z^{-2} + a_0 z^{-3}}$$

Compute the inverse z-transform of this expression to find the unforced solution. In general:

$$y(z) = \frac{-n(z)}{1 + a_{n-1} z^{-1} + \cdots + a_0 z^{-n}}$$

where

$$n(z) = \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-n} \end{bmatrix} \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ 0 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & a_0 \end{bmatrix} \begin{bmatrix} y_{-n} \\ y_{-n+1} \\ \vdots \\ y_{-1} \end{bmatrix}$$

2.1.2 Solution Using State Space Realizations

The general solution is much easier using state space methods. Convert the z-transform to the state space system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

The solution is written almost by inspection:

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-1-i} B u_i$$

For any state space realization, the initial conditions of the difference equation defined by the previous n output values are converted to x_0 by:

$$x_0 = A^n \begin{bmatrix} C \\ \vdots \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_{-n} \\ \vdots \\ y_{-1} \end{bmatrix}$$

The initial condition vector simplifies considerably when the observable canonical form realization is used. For $n = 3$ and $m = 2$ the state space realization is:

$$A = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

The initial condition equation simplifies to:

$$x_0 = \begin{bmatrix} -a_0 & -a_1 & -a_2 \\ 0 & -a_0 & -a_1 \\ 0 & 0 & -a_0 \end{bmatrix}^{-1} \begin{bmatrix} y_{-3} \\ y_{-2} \\ y_{-1} \end{bmatrix}$$

2.2 The Linear Least Square Problem

2.2.1 Problem Statement

The error vector is defined by:

$$e = F\theta - y$$

The objective is the find the minimum norm parameter vector θ that minimizes the weighted cost function:

$$J = e^H We$$

The variables are:

$e = N \times 1$ real or complex vector with components e_i

$\theta = p \times 1$ real parameter vector with components θ_i

$y = N \times 1$ real or complex data with components y_i

$F = N \times p$ real or complex matrix

$W \geq 0 = N \times N$ real or complex positive semi-definite weighting matrix

$J \geq 0 =$ real, scalar cost

Comments:

- The variables F and y are constructed from data, which can be real or complex. Estimation methods based in the time domain use real data and those in the frequency domain use complex data. In all cases the parameter vector θ is restricted to be real.
- The superscript H used in the cost function is the complex conjugate transpose, also called the Hermitian. In the real problem $e^H = e^T$, the regular transpose.
- The notation $W \geq 0$ means $e^H We \geq 0$ for all vectors. Because $\geq$ is used, W may not be invertible. The weight may be real or complex, and is always symmetric ($W = W^H$). The diagonal elements must always be real, (otherwise $W \neq W^H$) so if there are complex elements they are always on the off-diagonals.

2.2.2 Solution to the Real Problem

The parameter vector that minimizes the cost function is:

$$e = (WF)^+ Wy$$

where $(WF)^+$ is the pseudo-inverse. The SVD is used to compute the pseudo-inverse. There are several variations to the problem depending on the dimension and rank of WF. The pseudo-inverse works for each of these variations, but each is still worth considering. It is noted that the rank of WF and not just

F needs to be considered, because $W \geq 0$, and hence in general the rank of WF may be less than the rank of F.

Square:

WF is $N \times N$ and rank N. The inverse exists, the error is zero, the weight has no effect, and the unique solution is:

$$\theta = F^{-1} y$$

Tall, Thin:

WF is $N \times p$ with $p < N$ and WF has full column rank p. This is the usual problem we will encounter and it is the regression problem. The error is non-zero and the unique solution is:

$$\theta = (F^T WF)^{-1} WF^T y$$

Short, Fat:

WF is $N \times p$ with $N < p$ and has full row rank N. (This is not a problem we will encounter, but ...) the error is zero, the weight has no effect, the parameter vector that achieves zero error is not unique, and the parameter vector with minimum norm is:

$$\theta = F^T (FF^T)^{-1} y$$

Worst Case:

Sometimes the regression problem degenerates to this worst case. WF is $N \times p$ and the rank of WF is less than both N and p. The error is nonzero and the parameter vector that minimizes the error is not unique. Neither $F^T WF$ nor FF^T are invertible, and the SVD is the only way to go.

2.2.3 Solution to the Complex Problem

This is not a problem that is encountered here, but for the record the solution is:

$$\theta = (WF)^{+} Wy$$

The notation is the same as for the real problem, but now the parts are complex.

2.2.4 Solution to the Mixed Real and Complex Problem

The error can be complex, but the parameter vector is restricted to be real. The problem is converted to a real problem with the error:

$$e = \begin{bmatrix} \mathrm{Re}(F\theta - y) \\ \mathrm{Im}(F\theta - y) \end{bmatrix}$$

The solution is:

$$\theta = \begin{bmatrix} \mathrm{Re}(WF) \\ \mathrm{Im}(WF) \end{bmatrix}^{+} \begin{bmatrix} \mathrm{Re}(Wy) \\ \mathrm{Im}(Wy) \end{bmatrix}$$

This is not the same as the real part of the complex solution.

2.2.5 The Singular Value Decomposition

The pseudo-inverse is computed using the Singular Value Decomposition (SVD), which is now defined. The $N \times p$ matrix WF has rank r and can be real or complex. The matrix WF is decomposed into three parts:

$$WF = U \Sigma V^H$$

Each of the parts can be expanded into blocks with the dimensions as shown:

$$\underset{N}{\overset{p}{[WF]}} = \underset{N}{\overset{r \;\; N-r}{[U_1 U_2]}} \times \underset{N-r}{\overset{r \;\; p-r}{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}}} \times \underset{p-r}{\overset{p}{\begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix}}}$$

The block matrices are:

U_1 = orthonormal basis for range of WF

U_2 = orthonormal basis for null space of $(WF)^H$

V_1 = orthonormal basis for range of $(WF)^H$

V_2 = orthonormal basis for null space of (WF)

Σ_1 = diagonal matrix of singular values

The singular values are nonnegative real numbers ordered high to low: $\sigma_1 \geq \cdots \geq \sigma_r > 0$. The linear least square problem is solved using the pseudo-inverse:

$$\underset{p}{\overset{N}{\left[(WF)^+\right]}} = \underset{p}{\overset{r}{[V_1]}} \times \underset{r}{\overset{r}{\left[\Sigma_1^{-1}\right]}} \times \underset{r}{\overset{N}{\left[U_1^H\right]}}$$

2.2.6 Using the Null Space

The pseudo-inverse gives the minimum norm parameter vector, which is a nice way to resolve ambiguity, but we will encounter "worst cases" where this is not the desired solution. All of the solutions that minimize the error are given by:

$$\theta = (WF)^+ Wy + [\text{Null}(WF)]\alpha$$

where the vector α can be any $r \times 1$ vector. Equivalently, using the parts of the SVD:

$$\theta = (V_1 \Sigma_1^{-1} U_1^T) Wy + V_2 \alpha$$

The parameter vector θ is minimum norm when $\alpha = 0$, and this is the only possible value for α when the null space has dimension zero.

A variation of the linear least square problem uses just the null space. Consider the problem:

$$F \begin{bmatrix} \theta \\ \eta \end{bmatrix} = 0$$

A non-zero solution exists if the null space V_2 of F is non-trivial, in which case there must exist an α such that:

$$\begin{bmatrix}\theta\\ \eta\end{bmatrix}=\begin{bmatrix}V_{21}\\ V_{22}\end{bmatrix}\alpha$$

Here the basis for the null space is further decomposed into blocks. Now if η is fixed and has the same dimension as the null space, and if the inverse of V_{22} exists, then it follows that:

$$\theta = V_{21}V_{22}^{-1}\eta$$

2.3 Nonlinear Function Minimization

When the error is a nonlinear function of the parameters the minimization problem is more difficult, but hopefully worth the effort. One such complication occurs when the initial conditions needs to be estimated along with the parameters. The solution requires a good initial guess, usually found using a linear problem, partial derivatives need to be computed, iteration is required, and a stopping criterion is needed. It will be shown that each step of the iteration is not that different from the LLS problem just solved.

2.3.1 Problem Statement

The objective is to find the parameter vector θ that minimize the error function:

$$J(\theta)=\sum_{i=1}^{N} w_i e_i^2$$

where:

$$e_i = y_i - f(\theta,\psi_i)$$

The w_i's are weights, and the y_i's and ψ_i's are measurements. All of the variables are real. The error function can be equivalently written as:

$$J(\theta)=e^T We$$

where:

$$e=\begin{bmatrix}e_1\\ \vdots\\ e_N\end{bmatrix},\quad W=\begin{bmatrix}w_1 & & 0\\ & \ddots & \\ 0 & & w_N\end{bmatrix}$$

2.3.2 Taylor Series Expansion

The solution method is to start with an initial guess θ, and then to use the Taylor series expansion of $J(\theta)$ to compute an increment $\delta\theta$. The error function is expanded in two steps of a Taylor series:

$$J(\theta+\delta\theta)=J(\theta)-\delta\theta^T\beta+\frac{1}{2}\delta\theta^T A\delta\theta$$

The first derivatives β_k of $J(\theta)$ are packed into the vector β, and the second derivatives $\alpha_{k\ell}$ are packed into the matrix A. The components of the first derivative are:

$$\beta_k=\frac{-\partial J}{\partial\theta_k}=\sum_{i=1}^{N}\frac{-\partial}{\partial\theta_k}w_i e_i^2=2\sum_{i=1}^{N}w_i e_i\frac{-\partial e_i}{\partial\theta_k}=2\sum_{i=1}^{N}w_i e_i\frac{\partial f(\theta,\psi_i)}{\partial\theta_k}$$

Say that the number of parameters is p, and then the index is $k = 1,\ldots,p$. Equivalently, using matrices and vectors, it follows that:

$$\beta = 2F^T We$$

where the partial derivatives are packed as follows into the $N \times p$ matrix F:

$$F = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_N^T \end{bmatrix} \quad , \quad \phi_i^T = \begin{bmatrix} \dfrac{\partial f(\theta,\psi_i)}{\partial \theta_1} & \cdots & \dfrac{\partial f(\theta,\psi_i)}{\partial \theta_p} \end{bmatrix}$$

Next expand the second derivatives $\alpha_{k\ell}$, where each of the indices is defined for 1 to p:

$$\alpha_{k\ell} = \frac{\partial J}{\partial \theta_k \partial \theta_\ell} = \sum_{i=1}^{N} \frac{\partial}{\partial \theta_k \partial \theta_\ell} w_i e_i^2 = 2\sum_{i=1}^{N} w_i \left[\frac{\partial e_i}{\partial \theta_k}\frac{\partial e_i}{\partial \theta_\ell} + e_i \frac{\partial e_i}{\partial \theta_k \partial \theta_\ell} \right]$$

Note that the second derivatives of J use both the first and second derivatives of e_i. Do not use the second derivatives of e_i, partly to simplify the problem, and partly because Ref. xx argues that this is a good thing, leaving:

$$\alpha_{k\ell} \approx 2\sum_{i=1}^{N} w_i \frac{\partial e_i}{\partial \theta_k}\frac{\partial e_i}{\partial \theta_\ell} = 2\sum_{i=1}^{N} w_i \frac{\partial f(\theta,\psi_i)}{\partial \theta_k}\frac{\partial f(\theta,\psi_i)}{\partial \theta_\ell}$$

The summation can be more easily computed, it turns out, using matrices that have already been defined:

$$A = 2F^T WF$$

The problem essentially amounts to is computing the first derivatives of $f(\theta,\psi_i)$ and lining these derivatives up properly in the matrix F.

2.3.3 Iterations

Having defined the derivatives, the iteration used to update the parameter vector needs to be defined. There are many ways to do so, of which we note three:

Steepest Descent:

Just use the first derivative, and slowly march down it:

$$\delta a = \beta / \lambda$$

The parameter λ determines the step size. Setting this parameter is always troublesome. Too high? Too slow. Too low? Divergence.

Newton-Raphson:

If the current value of the parameter is close to a local minimum then a fast convergence is achieved using:

$$\delta a = A^{-1}\beta$$

What is close is not easy to know. Not close? Divergence.

Levenberg-Marquardt:

A balance between the previous two approaches is defined by:

$$\delta a = (\lambda I + A)^{-1} \beta$$

The iteration proceeds by:

- Initialize λ=1e6 (or some other value, large forces the initial steps to be small).
- Compute $\delta\theta$ and $J(\theta + \delta\theta)$.
- Stop?
- If $J(\theta + \delta\theta) > J(\theta)$, replace λ with 3λ (or some other factor) and jump to second step.
- Otherwise, replace θ with $\theta + \delta\theta$, replace λ with $\lambda/3$ (or some other factor) and jump to second step.

Stopping Criterion:

For all of these methods:

- Stop after a fixed number of steps.
- And/or stop when $\left|\chi^2(a + \delta a) - \chi^2(a)\right|$ is less than some threshold.
- And/or stop when $|\delta a|$ is less than some threshold.

Local Minima:

Local minima are always a problem, and there is no cure. Try a different initial condition, and/or smaller steps. Other minimization methods, such as genetic programming and simulated annealing, promise to do a better job of finding a global minimum. Well, maybe.

Summary:

SD: $\delta a = \beta / \lambda = F^T W e / \lambda$

NR: $\delta a = A^{-1}\beta = (F^T W F)^{-1} F^T W e$

LM: $\delta a = (\lambda I + A)^{-1}\beta = (\lambda I + F^T W F)^{-1} F^T W e$

2.3.4 The LLS Problem as a Special Case

The LLS estimation problem should be a special case. Let us show that. The error function is:

$$e_i = y_i - f(\theta, \psi_i) = y_i - \phi_i^T \theta$$

The matrix F of first derivatives is exactly the same matrix used for the LLS estimation problem. Start with an initial guess, it turns out any guess will do, but for simplicity set $\theta_0 = 0$, in which case the error of the initial guess is $e = y$. Use Newton-Raphson to update this guess:

$$\delta\theta_0 = A^{-1}\beta = (F^T W F)^{-1} W F^T y$$

We have shown that the LLS estimation problem, when posed in the more general form as a nonlinear minimization problem, converges in one step to the linear solution.

2.3.5 Mixed Real and Complex Problem

For time domain estimation all of the variables in the minimization problem are real. The problem needs to be extended to a mixed set of real and complex variables when used for frequency domain estimation. The solution at a high level is the same, but we go through the steps again being careful to keep track of what is real and what is imaginary. The statement of the minimization problem is:

$$J(\theta) = \sum_{i=1}^{N} w_i |e_i|^2$$

where:

$$e_i = y_i - f(\theta, \psi_i)$$

The parameter vector θ is real, the weights w_i are real numbers, and the function $J(\theta)$ outputs a real number. The data y_i and ψ_i can be complex, the function $f(\theta, \psi_i)$ outputs a complex number, and the error e_i is also complex. The cost function can be rewritten:

$$J(\theta) = e^H W e$$

To solve this mixed real and complex problem expand the cost function as before using two terms of the Taylor series:

$$J(\theta + \delta\theta) = J(\theta) - \delta\theta^T \beta + \frac{1}{2}\delta\theta^T A \delta\theta$$

The components of the first derivative for $k = 1, \ldots, p$ are:

$$\begin{aligned}
\beta_k &= \frac{-\partial J}{\partial \theta_k} \\
&= \sum_{i=1}^{N} \frac{-\partial}{\partial \theta_k} w_i e_i^H e_i \\
&= -\sum_{i=1}^{N} w_i \left[e_i^H \frac{\partial e_i}{\partial \theta_k} + \frac{\partial e_i^H}{\partial \theta_k} e_i \right] \\
&= \sum_{i=1}^{N} w_i \left[e_i^H \frac{\partial f(\theta, \psi_i)}{\partial \theta_k} + \frac{\partial [f(\theta, \psi_i)]^H}{\partial \theta_k} e_i \right] \\
&= 2\sum_{i=1}^{N} w_i \operatorname{Re}\left[\left(\frac{\partial f(\theta, \psi_i)}{\partial \theta_k} \right)^H e_i \right]
\end{aligned}$$

The last step follows because for the real derivative of a complex function the Hermitian and differential operations commute. To show this is the case:

$$\frac{\partial [f(x)]^H}{\partial x} = \frac{\partial [f_R(x) - j f_I(x)]}{\partial x} = \frac{\partial f_R(x)}{\partial x} - j\frac{\partial f_I(x)}{\partial x} = \left(\frac{\partial f(x)}{\partial x} \right)^H$$

The subscript R and I respectively denote the real and imaginary parts. Next, write the first derivative using matrices and vectors as follows:

$$\beta = 2\mathrm{Re}(F^H We)$$

where the $N \times p$ complex matrix F contains the partial derivatives:

$$F = \begin{bmatrix} \phi_1^H \\ \vdots \\ \phi_N^H \end{bmatrix} \quad , \quad \phi_i^H = \begin{bmatrix} \frac{\partial f(\theta,\psi_i)}{\partial \theta_1} & \cdots & \frac{\partial f(\theta,\psi_i)}{\partial \theta_p} \end{bmatrix}$$

The second derivatives of the cost function are:

$$\begin{aligned} \alpha_{k\ell} &= \frac{\partial J}{\partial \theta_k \partial \theta_\ell} \\ &= \sum_{i=1}^{N} \frac{\partial}{\partial \theta_k \partial \theta_\ell} w_i e_i^H e_i \\ &= 2\sum_{i=1}^{N} w_i \,\mathrm{Re}\left[\frac{\partial e_i^H}{\partial \theta_k} \frac{\partial e_i}{\partial \theta_\ell} + e_i^H \frac{\partial e_i}{\partial \theta_k \partial \theta_\ell} \right] \end{aligned}$$

Don't use the second derivatives of the error, so that:

$$\alpha_{k\ell} \approx 2\sum_{i=1}^{N} w_i \,\mathrm{Re}\left[\left(\frac{\partial f(\theta,\psi_i)}{\partial \theta_k} \right)^H \frac{\partial f(\theta,\psi_i)}{\partial \theta_\ell} \right]$$

In matrix form it follows that the second derivatives can be packed into the $p \times p$ matrix as follows:

$$A = 2\mathrm{Re}\left[F^H W F \right]$$

The iterative solution to the nonlinear problem follows as before. For example, the update for the Newton-Raphson method is:

$$\delta\theta = A^{-1}\beta$$

which can be expanded into several different ways:

$$\begin{aligned} \delta\theta &= \left(\mathrm{Re}\left[F^H W F \right] \right)^{-1} \mathrm{Re}\left[F^H W e \right] \\ &= \left(F_R^T W F_R + F_I^T W F_I \right)^{-1} \left(F_R^T W e_R + F_I^T W e_I \right) \\ &= \left(\begin{bmatrix} F_R \\ F_I \end{bmatrix}^T \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} F_R \\ F_I \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} F_R \\ F_I \end{bmatrix}^T \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} e_R \\ e_I \end{bmatrix} \right) \end{aligned}$$

2.4 Orthogonal Polynomials

Transforms are defined using polynomials. A problem with any polynomial is the powers are not orthogonal. The high order terms become almost linearly dependent, and hence high order problems are numerically ill-conditioned. It is better to not directly estimate the coefficients of the polynomial but to instead to estimate the coefficients of orthogonal polynomials. Define the polynomial:

$$n(x) = a_n x^n + \cdots + a_0$$

The same polynomial can be expressed as a sum of orthogonal polynomials:

$$n(x) = f_n p_n(x) + \cdots + f_0 p_0(x)$$

Any set of orthogonal polynomials will do, but the set usually recommended for estimation problems is the Chebyshev polynomials. This set of polynomials is optimal in the sense that reduced order polynomials minimize the maximum absolute value of the error. The Chebyshev polynomials are defined by the recurrence relation:

$$p_0(x) = 1$$

$$p_1(x) = x$$

$$p_k(x) = 2x p_{k-1}(x) - p_{k-2}(x)$$

The series continues:

$$p_2(x) = 2x^2 - 1$$

$$p_3(x) = 4x^3 - 3x$$

$$p_4(x) = 8x^4 - 8x^2 + 1$$

$$p_5(x) = 16x^5 - 20x^3 + 5x$$

$$p_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

The Chebyshev polynomials are orthogonal as defined by the weighted integral:

$$\int_{-1}^{1} (1-x^2)^{-1/2} p_m(x) p_n(x) dx = \begin{cases} 0 & m \neq n \\ \pi & m = n = 0 \\ \pi/2 & m = n \neq 0 \end{cases}$$

An important point is the integral has a finite range. The polynomials used in the transforms should be scaled so that the transform variable at least approximately has this range.

Matrix operations can be used to convert between regular and Chebyshev coefficients:

$$n(x) = a_n x^n + \cdots + a_0 = f_n p_n(x) + \cdots + f_0 p_0(x)$$

where:

$$\begin{bmatrix} a_n \\ \vdots \\ a_0 \end{bmatrix} = T_n \begin{bmatrix} f_n \\ \vdots \\ f_0 \end{bmatrix}$$

For $m = 3$:

$$\begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_3 \\ f_2 \\ f_1 \\ f_0 \end{bmatrix}$$

The transformation matrix is computed using the Matlab commands:

```
n=10;
T=zeros(n,n);
T(n-1:n,n-1:n)=eye(2);
for i=n-2:-1:1
  T(i:n,i)=[2*T(i+1:n,i+1);0]-T(i:n,i+2);
end
```

The transformation matrix is lower triangular, as is its inverse.

3. TIME DOMAIN ESTIMATION

3.1 Linear Least Square Estimation (One Step Prediction)

3.1.1 Problem Statement

Given the data:

$$u_k = \text{input data, for } k = 0,\ldots,N-1$$

$$y_k = \text{output data, for } k = 0,\ldots,N-1$$

and the initial conditions:

$$y_k = \text{initial conditions, for } k = -n,\ldots,-1$$

the problem is to estimate the parameters of the transfer function:

$$G(z) = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0}$$

which minimize the cost function:

$$J = e^T W e$$

and where each element of the error vector is defined by:

$$e_k = y_k - \hat{y}_k$$

The estimated output $\hat{y}_k$ is defined by:

$$\hat{y}_k = -a_{n-1} y_{k-1} - \cdots - a_0 y_{k-n} + b_m u_{k-n+m} + \cdots + b_0 u_{k-n}$$

This estimated output is the one-step prediction based on measured values of the previous inputs and outputs.

3.1.2 Solution

Convert the problem to standard form by sorting the data, initial conditions, and parameters into vectors and matrix. The difference equation is written more compactly using a vector inner product:

$$\hat{y}_k = \phi_k^T \theta$$

where the ϕ_k^T vector contains data points:

$$\phi_k^T = \begin{bmatrix} -y_{k-1} & \cdots & -y_{k-n} & u_{k-n+m} & \cdots & u_{k-n} \end{bmatrix}$$

and the θ vector contains the parameters:

$$\theta^T = [a_{n-1} \quad \cdots \quad a_0 \quad b_m \quad \cdots \quad b_0]$$

The error vector, called the prediction error, is equal to:

$$e = y - F\theta$$

where the ϕ_k vectors are rows of the matrix F :

$$F = \begin{bmatrix} \phi_0^T \\ \vdots \\ \phi_{N-1}^T \end{bmatrix}$$

The structure is best seen with an example, where for $n = 3$ and $m = 2$:

$$e = y - F\theta = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} -y_{-1} & -y_{-2} & -y_{-3} & u_0 & 0 & 0 \\ -y_0 & -y_{-1} & -y_{-2} & u_1 & u_0 & 0 \\ -y_1 & -y_0 & -y_{-1} & u_2 & u_1 & u_0 \\ -y_2 & -y_1 & -y_0 & u_3 & u_2 & u_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & -y_{N-2} & -y_{N-3} & u_N & u_{N-1} & u_{N-2} \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix}$$

The estimation problem is a linear least square minimization problem, for which the solution is:

$$\theta = (WF)^+ Wy$$

3.1.3 Discussion

Topics are addressed which will be covered in more depth in subsequent sections.

System Order:

If the system order n is known, then of course use it, but often this is not the case. Even if it is known, or at least approximately known, a lower order may be desired in order to obtain a reduced order model, or a higher order may be desired just to make sure the poles of the system are represented. Determining the order is an important practical problem. Four approaches that we have explored:

- Solve the LLS problem for different orders, stopping when the cost drops significantly.
- Use the SVD to determine the rank of F. If the input columns have full rank, and the matrix has column rank r, then the system order is $n = r - (m+1)$.
- From the a and b coefficients compute the Markov parameters, group these into a Hankel matrix, and use the SVD to determine the rank of the Hankel matrix, which is the system order.
- From the Markov parameters form a high order state space realization, compute a balanced realization of this state space system, and then the rank of the Grammian product is the system order. This is the best way to obtain a reduced order model due to guarantees on stability and due to error bounds.

Any one of these methods will work just fine in the ideal case of data generated from models and uncorrupted by noise. The last of these is least sensitive to noise.

Numerator Order:

The numerator order m must also be selected. Usually it suffices just to set $m=n$ or $m=n-1$. A digital system formed by sampling an analog systems preceded by a zero-order-hold, a very common occurrence, will have $m=n-1$. If the actual numerator order is less than the selected m, then some of the higher order coefficients of the numerator will be estimated to be zero, or close to it, which is of little consequence.

Number of Data Points:

There are $p=n+m+1$ parameters to be estimated. The minimum number of pairs of input and output data point needed is to estimate these parameters is p, plus the n initial conditions (u_k for $k=0,\ldots,p$ and y_k for $k=-n,\ldots,p$). If the initial condition are not known then $p+n$ pairs of data points are needed (u_k and y_k for $k=0,\ldots,p+n$). Using the minimum amount of data makes the parameters very sensitive to the data. This sensitivity is reduced by using the regression problem with $N \gg p$.

Initial Conditions:

The problem is stated using initial conditions. If they are known, usually this means they are zero, include them as shown above. One of the nice things about the time domain estimation problem is that the initial conditions do not have to be known, in which case simply do not include the first n rows of e, y, and F.

The Input:

The columns of F that contain the input should have full column rank. If the initial conditions are used then this will be true (as long as $u_i \neq 0$ for some $i \leq N-m$), thanks to the triangle of zeros in the upper right hand corner of F. If the initial conditions are not known and the first n columns of F are not included, then the input columns might lose full rank. The consequence of losing rank is that the numerator coefficients will not be accurately determined, and the system order cannot be determined. This will happen, for example, if a step input is used. For this reason the input should be "sufficiently exciting," which is another way of saying the input columns should have full column rank. Sine waves, by the way, are not enough, though a sum of sines with at least $m/2$ different frequencies will work.

Delay:

The system may have a delay, and even if not a pure delay, high order dynamics in the system can be approximated as an "effective delay." The z-transform has a delay of $n-m$ steps, which means the most recent output that y_k depends on is $u_{k-(n-m)}$. Setting the difference $n-m$ is therefore one way to force a delay. If the size of the delay is known to be d steps, then it is better to set $m=n$ and multiply the denominator by the "pure delay" z^{-d}, and then just estimate the remaining a and b coefficients. In other words, the estimation routine should be helped out by not requiring that the denominator parameters that are known to be zero be part of the set of parameters to be estimated. This is simply done by shifting done the input columns by d rows.

Simulation Error:

A common test is to run the measured input through the estimated system and then compare the output of the simulation with the measured output. The resulting error, called the simulation error, is not the error that is minimized by the LLS estimation problem. The problem as stated minimizes the one-step prediction error. The problem can be reformulated to do minimize the simulation error, but then this becomes a nonlinear problem. It is solved after further consideration of the linear problem.

System Changes:

If the system changes the parameter estimates should also. The problem as stated, however, starts at time zero and uses all of the available data, in which case changes to the system will somehow be averaged. A better way to respond to changes is just to use the previous N data points; another way is to use a weight with an exponential "forgetting factor."

Stability:

There is no guarantee that the estimated system is stable. Indeed, the ability to estimate an unstable system can be considered an advantage of the method. The problem, however, is that systems known to be stable can be estimated to be unstable. This can occur due to noise, or due to setting the system order too high. In either case, it is quite typical for the estimated system to have right half plane (RHP) poles and zeros that almost cancel.

Why does this happen? Qualitatively, the one-step prediction is not "hurt" much by instability. The prediction is only one step, and a one step prediction error of an unstable system is not much different and in fact may be smaller than for a stable system. The simulation error, however, will be dramatically different, and even a "just barely" unstable estimated system can result in numerical overflow.

There is unfortunately no good solution to the "false stability" problem. Several steps are nevertheless recommended to make a bad situation not quite as bad.

- If the system is known to be stable then the solution can be restricted to be stable. First solve the unrestricted estimation problem. If the estimated system is unstable then map the unstable poles to their stable mirror images (opposite sign real part for Laplace domain poles, same phase inverse magnitude for z domain poles). This mapping is equivalent to solving the estimation problem using the autocorrelation. As a final step fix the pole locations and then resolve the estimation problem for just the numerator coefficients.
- If the estimated system is stable then a reduced order model can be found via the balanced realization that is guaranteed to be stable. Stability is therefore not compromised when transforming the high order model.
- Start the one-step prediction estimation problem with different system orders. Choose the one with the smallest simulation error.

3.1.4 Generalization

The one-step prediction problem is a starting point for the more general problem of parameter estimation. An overview of different types of generalization is presented below.

Error Weight:

The problem as stated is actually quite general – the generality being the choice W of the error weight, where:

$$e = W(F\theta - y)$$

The weight must be $W \geq 0$, so that the cost $J = e^T We \geq 0$. The weight does not have to be invertible. Possible choices:

- Restrict the set of measured outputs by setting parts of W equal to zero. This is what is done, for example, in "boxcar" averaging recursive estimation.

- Set the weight to $W = \text{diag}\{\alpha^{N-k}\}$. This is what is done, for example, in the "exponential forgetting" version of the recursive estimation problem.
- Filter the signals by setting rows of W to shifted versions of the filter impulse response. Filtering can be used to restrict the error to a given frequency range, or to filter out noise known to exist in a certain frequency range.
- Replace sets of input and output signals with their average. This increases correlation between input and output by averaging additive noise. Do this, for example, by setting:

$$W = \begin{bmatrix} (1 & \cdots & 1)/n_{avg} & 0 & & \cdots \\ 0 & & & (1 & \cdots & 1)/n_{avg} & \ddots \\ \vdots & & & & \ddots & & \ddots \end{bmatrix}$$

- Switch to the frequency domain by setting the weight to the digital Fourier transformation, where the *ki* element is:

$$W_{ki} = e^{-ijk2\pi/N}$$

- The shifted signals in the data matrix are the same as multiplying the un-shifted signal by z^{-k} (except for a small end effect). Select a subset of the resulting Fourier coefficients to restrict attention to a given set of frequencies. Increase coherence by averaging adjacent frequency bins. Select a subset of averaged coefficients with the highest coherence.
- Switch to the wavelet domain by setting the weight to a wavelet transformation. Use either continuous or discrete wavelets. Select a subset of coefficients to restrict attention to a given set of scales. Increase coherence by averaging scale coefficients (for discrete wavelets this is the same as using lower scales).

In each one of these examples it is numerically faster to compute the transformations directly using the input and output time series and then rebuild the data matrix. Filter the signals, for example, and then rebuild the data matrix with the filtered signals. Use the FFT, for example, rather than the DFT, and then rebuild the data matrix from the FFT coefficients.

Parameter Transformation:

Insert a transformation matrix U and its pseudo-inverse as shown below:

$$e = W\left(FUU^{+}\theta - y\right)$$

Use the pseudo-inverse because in general this transformation does not have to be invertible. The LLS solution is:

$$U^{+}\theta = (WFU)^{+}(Wy)$$

If U is invertible then the original coefficients can be recovered by $\theta = U(U^{-1}\theta)$. Reasons to do this transformation include:

- Select a reduced order set of parameters by defining U to select a subset of the columns of F and the corresponding subset of coefficients. In this case U is not invertible and the full order solution cannot be recovered.

- Some of the parameters may be known, and a transformation can be used to restrict attention to the known parameters. For example, an integrator can be inserted by fixing one of the poles at $(z-1)$, or zeros can be fixed at $(z+1)$, considered the digital version of infinity because they transform to infinity using the inverse bilinear transformation.
- Change from coefficients in polynomials of z to coefficients of orthogonal polynomials. This will improve the numerical condition of the estimation problem, especially for high order problems. In this case the transformation U is invertible.
- Change from polynomials in z to polynomials in $z-1$ or $\delta=(z-1)/h$ or $\delta=(z-1)/(\omega_{max}h)$ or $s=(2/T)(z-1)/(z+1)$, and so on. This will improve the numerical condition by shifting the poles away from a cluster around $z=1$, and/or will allow direct estimation of a Laplace transform. In this case the transformation U is invertible.

Nonlinear Minimization:

The nonlinear minimization problem can be stated in general using the error:

$$e=f(\theta,u,y)$$

which is a function of the parameters, input and output. The version that we will be using is the simulation error:

$$e=W(\hat{y}-y)$$

where the simulated output is defined by:

$$\hat{y}(z)=G(z)u(z)$$

or equivalently by a difference equation. In this case the initial conditions must be known or be estimated along with the parameters. Similar weights and parameter transformations as defined above can be used for the nonlinear problem.

3.2 Determining the System Order

3.2.1 Using Exhaustive Search

The standard form of the LLS estimation problem assumes the following orders are known:

m = numerator order

n = denominator order

Sometimes these quantities are not known, and one way to determine them is to perform an exhaustive search (i.e., check each possible combination). Comments:

- Fill a two-dimensional table with costs J, and then pick the lowest n followed by the lowest m where the cost is less than an acceptable threshold.
- Reduce the search from two to one dimension by setting $m=n$ or $m=n-1$.
- Better would be an estimation method that allows the system order to be selected after preliminary calculations that do not depend on the system order. Fortunately, this can be done.

3.2.2 Using Rank of Data Matrix

Setup the data matrix F using n larger than the expected system order. Set $m=n$ and then the total number of columns is $2n+1$. Assume that the input columns have full rank, and that the actual order of the system is n_1. The columns of F multiplied by parameters a_i for $i \geq n_1$ are linearly dependent on other columns, because each element of these columns satisfies the difference equation:

$$y_k = -a_{n_1-1}y_{k-1} - \cdots - a_0 y_{k-n_1} + b_m u_{k-n_1+m} + \cdots + b_0 u_{k-n_1}$$

The rank of the matrix will reduce by one for each of these dependent columns, and the actual system order will reduce from n by the same amount. Say r is the rank of the matrix, and then the system order is:

$$n_1 = r-(n+1)$$

3.2.3 Using Markov Parameters

For a digital system the Markov parameters are the impulse response of the system. The Markov parameters can be computed from the a and b coefficients of the z-transform using a series expansion:

$$G(z) = \frac{b_n + \cdots + b_0 z^{-n}}{1 + a_{n-1}z^{-1} \cdots + a_0 z^{-n}} = h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots$$

The h coefficients are the Markov parameters. Multiply through by the denominator:

$$\left(b_n + \cdots + b_0 z^{-n}\right) = \left(1 + a_{n-1}z^{-1} \cdots + a_0 z^{-n}\right)\left(h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots\right)$$

and then equate like powers of z:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots \\ a_{n-1} & 1 & \ddots & 0 & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ a_0 & \ddots & \ddots & 1 & 0 & \ddots \\ 0 & a_0 & \ddots & a_{n-1} & 1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \\ h_{n+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_0 \\ 0 \\ \vdots \end{bmatrix}$$

Invert the matrix with the a coefficients to determine the Markov parameters. Faster are to recursively compute the Markov parameters. Once found, sort the Markov parameters into a Hankel matrix:

$$H_p = \begin{bmatrix} h_1 & h_2 & \cdots & h_p \\ h_2 & h_3 & \iddots & h_{p+1} \\ \vdots & \iddots & \iddots & \vdots \\ h_p & h_{p+1} & \cdots & h_{2p-1} \end{bmatrix}$$

It follows that the rank of H_p is n for any $p \geq n$, and therefore a sufficiently large Hankel matrix can be used to determine the system order. The singular values are the best way to determine the rank, and furthermore, the Eigensystem Realization Algorithm (ERA), which uses all of the parts of the SVD can be used to determine a state space realization of the system.

3.2.4 Why the Minimum Norm Solution is not the Best Solution

The solution to the LLS estimation problem using the pseudo-inverse works for a data matrix of any rank, always giving a parameter vector that is minimum norm. Therefore, one approach is to select n larger than

expected, compute the minimum norm parameter vector, and then use subsequent calculations to reduce the system order. This is actually a good approach, but the point that needs to be made is that extra calculations are in fact needed because the minimum norm parameter vector is by itself not the best solution. A simple example will suffice to make this point.

Consider the case where the system is known to be of the form $G(z)=b/(z+a)$ but in the regression problem orders $n=2$ and $m=1$ are selected. A transfer function of the form $G(z)\times(z+\alpha)/(z+\alpha)$ will be estimated. The parameter vector is:

$$\theta^T=[a+\alpha \quad a\alpha \quad b \quad b\alpha]$$

The α that minimizes the norm is found by setting $\partial(\theta^T\theta)/\partial\alpha=0$ to be:

$$\alpha=\frac{-a}{1+a^2+b_2}$$

Ideally this α would be zero, because then the minimal order $G(z)$ can trivially be obtained, but this is not the case, because $\alpha=0$ only when $a=0$. It follows that $|\alpha|<1$, so at least the extra mode that is selected by the regress problem is stable.

3.3 Initial Conditions

Parameter estimation using the one-step prediction problem works using the total solution to the difference equation. This is important. We do not have to assume zero initial conditions. The system does not need to start at rest. The solution works with any initial condition and with any input, with one caveat – the input columns in the data matrix need to be full rank. A consequence of this generality is that the parameter estimation problem can be used "midstream" in a maneuver.

Here this independence on the initial conditions is established, and then we argue that even though the parameter estimates do not depend on the initial conditions, the initial conditions are still important.

3.3.1 Why they do not Matter for Prediction Error

The parameter estimation solution for the one-step prediction error problem is $\theta=(FW)^+Wy$, where the matrix F and vector y built from the time series samples u_k and y_k. The time series by definition of the problem satisfies the difference equation, and is therefore a total solution of the equation. The total solution takes into account the initial conditions, whatever they may be, zero or not. The matrix solution works for any solution of the difference equation, and therefore the parameters do not depend on the initial conditions.

The above statement, while true, does constitute a mathematical proof. We will not attempt a rigorous proof, but will do so for a special case, which is suggestive of a general proof. Take the case where $n=1$ and $m=1$, and then:

$$\begin{bmatrix} y_{k-2} & u_{k-2} \\ y_{k-1} & u_{k-1} \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}=\begin{bmatrix} y_{k-1} \\ y_k \end{bmatrix}$$

Solve for the parameters:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\Delta}\begin{bmatrix} u_{k-1} & -u_{k-2} \\ -y_{k-1} & y_{k-2} \end{bmatrix}\begin{bmatrix} y_{k-1} \\ y_k \end{bmatrix}$$
$$= \frac{1}{\Delta}\begin{bmatrix} u_{k-1} & -u_{k-2} \\ -y_{k-1} & y_{k-2} \end{bmatrix}\begin{bmatrix} a & b \\ a & b \end{bmatrix}\begin{bmatrix} y_{k-2} \\ y_{k-1} \end{bmatrix}$$
$$= \frac{1}{\Delta}\begin{bmatrix} a\Delta \\ b\Delta \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

where $\Delta = y_{k-2}u_{k-1} - y_{k-1}u_{k-2}$. What this shows is that any two sets of inputs and outputs can be used to estimate the parameters. There is no dependence on the initial conditions.

3.3.2 Why they do Matter for Simulation Error

The initial conditions are needed in order to simulate the identified system using the measured input, in order to try to duplicate the measured output. In other words, we need to know the initial conditions to use simulation to check validity. The resulting error between the measured and simulated output is called the simulation error. This is another measure of "goodness," and arguably a better measure than the one-step prediction error. The simulation error indicates how well the estimated system will respond to different inputs, not just one step in the future, but for all future time.

There is a subtle point that needs to be made about where to start the estimation problem. If the initial conditions are known then start the problem at $k = 0$:

$$e = y - F\theta = \begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \\ y_n \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} -y_{-1} & \cdots & -y_{-n} & u_{m-n} & \cdots & u_{-n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{n-1} & \cdots & -y_{-1} & u_{m-1} & \cdots & u_{-1} \\ -y_{n-1} & \cdots & -y_0 & u_m & \cdots & u_0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n} & u_{N+m-n} & \cdots & u_{N-n} \end{bmatrix}\begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

If the initial conditions are not known, however, then the estimation problem should start with $k = n$:

$$e = y - F\theta = \begin{bmatrix} y_n \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} -y_{n-1} & \cdots & -y_0 & u_m & \cdots & u_0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n-1} & u_{N+m-n} & \cdots & u_{N-n} \end{bmatrix}\begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

Sometimes these first n equations can make the difference between having full and reduced rank input columns. This will be the case, for example, if an ideal step input is used.

3.3.3 Estimating the Initial Conditions

If the transfer function parameters are known, then these parameters and the data can be used to estimate the initial conditions. Several ways to do so are listed.

Use the First n Inputs and Outputs To Estimate the Initial Conditions:

Use the data u_k and y_k for $k \geq n$ to estimate the transfer function parameters. Then use the parameters and the data for $0 \leq k < n$ to estimate the initial conditions. Consider the case $n=3$ and $m=2$, from which the general solution can be inferred. The difference equations for the first three outputs are:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -y_{-1} & -y_{-2} & -y_{-3} & 0 & 0 & 0 \\ -y_0 & -y_{-1} & -y_{-2} & u_0 & 0 & 0 \\ -y_1 & -y_0 & -y_{-1} & u_1 & u_0 & 0 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix}$$

Rearrange these equations:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ 0 & a_0 & a_1 \\ 0 & 0 & a_0 \end{bmatrix} \begin{bmatrix} y_{-3} \\ y_{-2} \\ y_{-1} \end{bmatrix} = -\begin{bmatrix} 1 & 0 & 0 \\ a_2 & 1 & 0 \\ a_1 & a_2 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ b_2 & 0 & 0 \\ b_1 & b_2 & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}$$

Invert the matrix on the left to solve for the initial conditions. The inverse exists if $a_0 \neq 0$. The matrix equation for the general *n*th order system is:

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ 0 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & a_0 \end{bmatrix} \begin{bmatrix} y_{-n} \\ \vdots \\ y_{-2} \\ y_{-1} \end{bmatrix} = -\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_1 & \cdots & a_2 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} + \begin{bmatrix} b_n & 0 & \cdots & 0 \\ b_{n-1} & b_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ b_1 & \cdots & b_{n-1} & b_n \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}$$

The initial conditions are used to compute the initial condition response for $k \geq 0$, which subtracted from the total response gives the forced response.

Use the First n Inputs and Outputs As A Surrogate for the Initial Conditions:

Again use the input and output data for $k \geq n$ to estimate the transfer function parameters. Compute the total response for $k \geq n$ using just the input and output data from $0 \leq k < n$. In other words, use the existing input for $0 \leq k < n$, and then set $u_k = 0$ for $k \geq n$. The total response for $k \geq n$ is equivalent to the initial condition response as computed above. The initial condition and forced responses can then be separated for $k \geq n$. This method has two advantages, extra calculations are not needed, and the result is not sensitive to the determination of system order.

Use All of the Data to Estimate the Initial Conditions:

Using just the first *n* data points makes the estimated initial condition sensitive to these data points. Better is to use all of the data to setup a regression problem, the solution of which is the estimated initial conditions. Use the data for $k \geq n$ to estimate the parameters of the system. Simulate the system $n+1$ times for $k \geq 0$ and label the outputs as follows:

$$\hat{y} = \text{zero initial conditoin response to input } u$$
$$\hat{y}_1 = \text{response to initial condition } y_{-1} = 1$$
$$\vdots$$
$$\hat{y}_n = \text{response to initial condition } y_{-n} = 1$$

The measured response, which is the total response, equals:

$$y = \hat{y} + \underbrace{\begin{bmatrix} \hat{y}_1 & \cdots & \hat{y}_n \end{bmatrix}}_{\hat{F}} \underbrace{\begin{bmatrix} y_{-1} \\ \vdots \\ y_{-n} \end{bmatrix}}_{x_0}$$

This completes the setup of the regression problem. Use the same weight used to estimate the system parameters, and then:

$$x_0 = \left(W\hat{F}\right)^{+} \left[W(y - \hat{y})\right]$$

Use State Space Methods to Estimate the Initial Conditions:

The output of the digital state space system is:

$$y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^{k-1-i} B u_i$$

Define $\hat{y}$ as the zero initial response and setup the regression problem:

$$y = \hat{y} + \underbrace{\begin{bmatrix} C \\ \vdots \\ CA^{N-2} \end{bmatrix}}_{\hat{F}} x_0$$

From which:

$$x_0 = \left(W\hat{F}\right)^{+} \left[W(y - \hat{y})\right]$$

3.4 Implementation of LLS Estimation Using Recursive Least Squares

The parameter estimate at step k can be efficiently computed in terms of the estimate at the previous step. The resulting savings in computation time is a major benefit and has been one of the key reasons why the one-step parameter estimation method is popular. The problem statement in Section 3.A defined data for time steps $0 \le i \le N-1$. The recursive version of the problem defines the data for $k-N \le i \le k-1$, with the objective being to estimate the next output y_k. In the following define:

$$A_k = \begin{bmatrix} \phi_{k-N}^T \\ \vdots \\ \phi_{k-1}^T \end{bmatrix} \quad , \quad b_k = \begin{bmatrix} y_{k-N+1} \\ \vdots \\ y_k \end{bmatrix}$$

The parameter estimate at time k using the square un-weighted solution is:

$$\hat{\theta}_k = \left(A_k^T A_k\right)^{-1} \left(A_k^T b_k\right)$$

Always using the previous N data points is called "boxcar" estimation, because the rectangular time window looks something like a boxcar rolling down the time line. The boxcar version of the problem adapts to changes in the system. Using all of the available data means that N grows with each step, and this is called "stationary" estimation, because this version of the estimation problem is appropriate for a

stationary system. Another way to achieve an adaptive solution is to use a weight that exponential decays old data. This is the subject of the following section.

We will present the recursive solution to the estimation problem but consider first a straightforward but naïve implementation of the square solution. The matrix A_k is of order $N \times p$, where $p = n + m + 1$ is the number of parameters. The required computations are:

$X = A_k^T A_k$ requires Np^2 multiplications

$Y = A_k^T b_k$ requires Np multiplications

$P_k = X^{-1}$ requires p^3 multiplications

$\hat{\theta}_k = P_k Y$ requires p^2 multiplications

Total multiplications are $p^3 + (N+1)p^2 + Np$.

This is not terrible, because usually p is small, but with little extra work we can do better. Use the outer product to compute the intermediate terms X and Y:

$$X = A_k^T A_k = \phi_{k-N}\phi_{k-N}^T + \cdots + \phi_{k-1}\phi_{k-1}^T$$

$$Y = A_k^T b_k = \phi_{k-N} y_{k-N} + \cdots + \phi_{k-1} y_k$$

And hence:

$X = X_{prev} - \phi_{k-N-1}\phi_{k-N-1}^T + \phi_{k-1}\phi_{k-1}^T$ requires $2p^2$ multiplications

$Y = Y_{prev} - \phi_{k-N-1} y_{k-N-1} + \phi_{k-1} y_k$ requires $2p$ multiplications

$P_k = X^{-1}$ requires p^3 multiplications

$\hat{\theta}_k = P_k Y$ requires p^2 multiplications

Total multiplications are $p^3 + 3p^2 + 2p$.

The boxcar version of the problem (shown above) subtracts off the terms at step $k - N$. The stationary version of the problem does not subtract off these terms. The matrix inversion dominates either version of the problem, but as long as the total number of multiplications is less than about 10,000 (p less that about 25) then with circa 2003 PC computers this does not pose a real-time problem. More likely p will be less than 10.

The computational burden can be reduced still further, resulting in what is known as the "Recursive Least Square" algorithm. The key step is to use the Matrix Inversion Lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Set:

$$A = P_{k-1}^{-1}$$

$$B = \phi_{k-1}$$

$$C = 1$$

$$D = \phi_{k-1}^T$$

Then:

$$P_k = P_{k-1} - \frac{P_{k-1}\phi_{k-1}\phi_{k-1}^T P_{k-1}}{1+\phi_{k-1}^T P_{k-1}\phi_{k-1}}$$

Thus the matrix inversion can be updated with just a scalar inversion. The estimated parameter vector is:

$$\hat{\theta}_k = P_k A_k^T b_k$$

Substitute for P_k and $A_k^T b_k$ and it follows non-easily that:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \frac{P_{k-1}\phi_{k-1}}{1+\phi_{k-1}^T P_k \phi_{k-1}}\left(y_k - \phi_{k-1}^T \hat{\theta}_{k-1}\right)$$

This is the parameter update. There are some common terms that can be grouped together, and at each step the following calculations are made:

$\xi_{k-1} = P_{k-1}\phi_{k-1}$ requires p^2 multiplications

$v_{k-1} = \xi_{k-1}\left(1+\phi_{k-1}^T \xi_{k-1}\right)^{-1}$ requires $2p$ multiplications

$\hat{\theta}_k = \hat{\theta}_{k-1} + v_{k-1}\left(y_k - \phi_{k-1}^T \hat{\theta}_{k-1}\right)$ requires $2p$ multiplications

$P_k = P_{k-1} - v_{k-1}\xi_{k-1}^T$ requires p^2 multiplications

$\hat{y}_k = \phi_k^T \hat{\theta}_k$ requires p multiplications

Total multiplications are $2p^2 + 5p$

This recursive algorithm has reduced the problem from order three to order two, increasing the number of estimated parameters (within the budget of 10,000 multiplications per step) from about 25 to 70, which is about a three-fold increase. This improvement is not quite the Holy Grail of linear algebra, but still quite a good thing. The Holy Grail is a reduction down to order 1.

3.5 Exponential Weighting

An alternative to boxcar weighting is exponential weighting. The advantage of exponential weighting is less data storage – you don't have to keep the old data around in order to subtract it off N steps later. Perhaps not a big advantage these days, but let us continue. The exponential weight is introduced into the cost function:

$$J_k = \sum_{i=k-N+1}^{k} \alpha^{k-i}\left(\hat{y}_k - y_k\right)^2$$

What is the time constant? This is the number of steps until the weight equals e^{-1}, in other words, m such that $\alpha^m = e^{-1}$, which is:

$$m = -1/\ln\alpha \text{ [steps]}$$

Set the step size to h seconds and then the time constant in seconds is:

$$\tau = mh = -h/\ln\alpha \text{ [secs]}$$

Turning the problem around, the exponential weight in terms of the time constant is:

$$\alpha = e^{-h/\tau}$$

In three time constants the weight is down to about 5%. Data before three time constants has little effect, unless there are anomalies like giant spikes, and three time constants is a good way to approximate the effective length of the input.

Now solve the estimation problem. Rewrite the cost function as:

$$J_k = e_k^T W_k e_k$$

where W_k is a diagonal matrix with the weights:

$$W_k = \text{diag}\begin{pmatrix}\alpha^{N-1} & \cdots & \alpha^2 & \alpha & 1\end{pmatrix}$$

The weighted linear least square solution is:

$$\hat{\theta}_k = P_k A_k^T W_k b_k$$

where:

$$P_k = \left(A_k^T W_k A_k\right)^{-1}$$

The update equations at each step change to:'

$$A_k^T W_k A_k = \alpha A_{k-1}^T W_{k-1} A_{k-1} + \phi_{k-1}\phi_{k-1}^T$$
$$A_k^T W_k b_k = \alpha A_{k-1}^T W_{k-1} b_{k-1} + \phi_{k-1} y_k$$

Apply the Matrix Inversion Lemma as before to get:

$$P_k = \frac{1}{\alpha}\left[P_{k-1} - \frac{P_{k-1}\phi_{k-1}\phi_{k-1}^T P_{k-1}}{\alpha + \phi_{k-1}^T P_{k-1}\phi_{k-1}}\right]$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \frac{P_{k-1}\phi_{k-1}}{\alpha + \phi_{k-1}^T P_k \phi_{k-1}}\left(y_k - \phi_{k-1}^T \hat{\theta}_{k-1}\right)$$

These are modest changes to the unweighted problem, increasing the number of multiplications by p^2, to a total of $3p^2 + 5p$.

3.6 Minimizing the Simulation Error (Nonlinear Minimization)

3.6.1 Problem Statement

A linear shift invariant digital system is characterized by the z-transform:

$$\frac{y(z)}{u(z)} = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1}z^{n-1} \cdots + a_0}$$

and the difference equation:

$$y_k = -a_{n-1}y_{k-1} - \cdots - a_0 y_{k-n} + b_m u_{k-n+m} + \cdots + b_0 u_{k-n}$$

Given the time series u_i and y_i defined for $i = 0,\ldots,N-1$, the objective is to estimate the parameters of the difference equation and the initial conditions y_{-n} to y_{-1} that minimize:

$$J = \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the simulation error:

$$\hat{y}_i = -a_{n-1}\hat{y}_{i-1} - \cdots - a_0\hat{y}_{i-n} + b_m u_{i-n+m} + \cdots + b_0 u_{i-n}$$

The simulation error problem differs from the one-step prediction error problem in the following way:

- The previous outputs used to compute $\hat{y}_i$ are estimated outputs – not measured outputs. This problem tries to match the measured output by playing back just the input through the estimated system.
- The initial conditions are included in the parameters to be identified. If the initial conditions are known, then they can be removed from the list of parameters to be identified.

3.6.2 Solution

This is a nonlinear minimization problem. Use the general solution, with partial derivatives specific to this version of the problem. The cost function is rewritten:

$$J(\theta) = \sum_{i=0}^{N-1} w_i e_i^2$$

where:

$$e_i = y_i - \hat{y}_i$$

$$\hat{y}_i = f(\theta, \psi_i)$$

$$\theta^T = (y_{-n} \quad \cdots \quad y_{-1} \quad a_{n-1} \quad \cdots \quad a_0 \quad b_m \quad \cdots \quad b_0)$$

$$\psi_i^T = (u_i \quad \cdots \quad u_0)$$

We need to compute the partial derivative of $f(\theta, \psi_i)$ with respect to the parameters. Rather than use the function $f(\theta, \psi_i)$ directly it is better to use the recursive formula for the output. The partial derivatives with respect to the z-transform parameters, for $i = 0,\ldots,N-1$:

$$\frac{\partial \hat{y}_i}{\partial a_{n-j}} = -\hat{y}_{i-j} - a_{n-1}\frac{\partial \hat{y}_{i-1}}{\partial a_{n-j}} - \cdots - a_0\frac{\partial \hat{y}_{i-n}}{\partial a_{n-j}} \quad \text{for } j = 1,\ldots,n$$

$$\frac{\partial \hat{y}_i}{\partial b_{m-j}} = u_{i-j} - a_{n-1}\frac{\partial \hat{y}_{i-1}}{\partial b_{m-j}} - \cdots - a_0\frac{\partial \hat{y}_{i-n}}{\partial b_{m-j}} \quad \text{for } j = 0,\ldots,m$$

These partial derivatives are respectively the signals $-\hat{y}_{i-j}$ and u_{i-j} passed through a system with just the denominator of the identified z-transform:

$$\frac{\text{Partial}(z)}{\hat{y}(z)} = \frac{z^{n-j}}{z^n + a_{n-1}z^{n-1} \cdots + a_0}$$

The initial conditions of this denominator-system are zero. The partial derivatives with respect to the initial conditions, for $i = 0,\ldots,N-1$:

$$\frac{\partial \hat{y}_i}{\partial y_{-j}} = -a - a_{n-1}\frac{\partial \hat{y}_{i-1}}{\partial y_{-j}} - \cdots - a_0\frac{\partial \hat{y}_{i-n}}{\partial y_{-j}} \quad \text{for } j = -n,\ldots,-1$$

where:

$$a = \begin{cases} -a_{n-j-i} & \text{for } 0 \le i \le n-j \\ 0 & \text{otherwise} \end{cases}$$

This is the initial condition response of the denominator-system.

3.6.3 Partial Derivatives for First Order Problem

The simulation output and partial derivatives are listed below for $n = 1$ and $m = 1$:

i	$\hat{y}_i$	$\partial\hat{y}_i / \partial a$	$\partial\hat{y}_i / \partial b$	$\partial\hat{y}_i / \partial y_{-1}$
0	ay_{-1}	y_{-1}	0	a
1	$a^2y_{-1} + bu_0$	$2ay_{-1}$	u_0	a^2
2	$a^3y_{-1} + abu_0 + bu_1$	$3a^2y_{-1} + bu_0$	$au_0 + u_1$	a^3
3	$a^4y_{-1} + a^2bu_0 + abu_1 + bu_2$	$4a^3y_{-1} + 2abu_0 + bu_1$	$a^2u_0 + au_1 + u_2$	a^4

The derivatives can be computed directly from the expansions of $\hat{y}_i$, or from the difference equations defined earlier.

3.7 Time Domain Estimation Examples

3.7.1 Ideal, Low Order Problem

The unweighted, LLS, one-step prediction, time domain parameter estimation problem is demonstrated using a known system with data created by simulation. The system is the F-15 Active fighter aircraft, longitudinal axis, short period approximation, linearized at Mach 0.9, h=30000, U0=895 ft/sec, alpha0=1.84 deg:

$$\frac{d}{dt}\begin{bmatrix}\alpha \\ q\end{bmatrix} = \begin{bmatrix}-.9543 & 1 \\ 7.1125 & -.7770\end{bmatrix}\begin{bmatrix}\alpha \\ q\end{bmatrix} + \begin{bmatrix}-.1199 & -.0297 & -.0290 \\ -12.44 & 6.346 & -3.104\end{bmatrix}\begin{bmatrix}\delta_{stab} \\ \delta_{can} \\ \delta_{noz}\end{bmatrix}$$

where:

α [deg] = angle of attack

q [deg/sec] = pitch rate

θ [deg] = pitch attitude

δ_{stab} [rad] = stabilator

δ_{can} [rad] = canard

δ_{noz} [rad] = nozzle

$\delta_e = \delta_{stab} - \delta_{can} + \delta_{noz}$ [rad] = effective elevator

Ideal actuators and sensors are assumed and a pitch SCAS is implemented as shown in Figure 128:

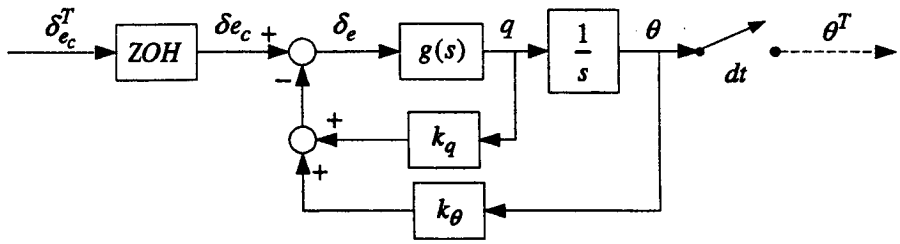


Figure 128. F-15 Active Pitch SCAS (Low Order Approximation)

With gains of $[k_q \quad k_\theta] = -.435 \times [1 \quad 3.5]$ the open and closed loop responses are:

$$\text{Open loop: } \frac{q}{\delta_e} = \frac{-21.9(0.993)}{(-1.8)(3.53)}$$

$$\text{Closed loop: } \frac{\theta}{\delta_{e_c}} = \frac{-21.9(0.993)}{(1.55)(3.39)(6.32)}$$

It is seen that the pitch SCAS stabilizes the unstable short period mode. The analog closed loop system is discretized using the ZOH equivalence and then simulated as shown in Figure 129. (The input is bandpass noise from 0.1 to 10 rad/sec, 3 seconds of zero on either end, no taper, sample period = 20 msec.)

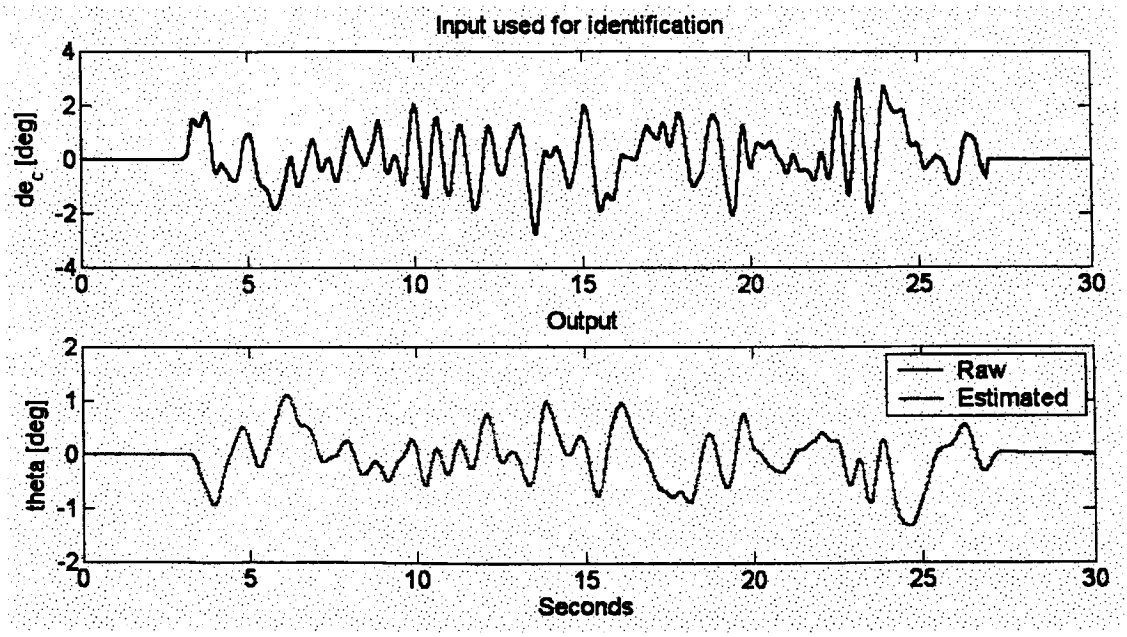


Figure 129. Time Domain Estimation (F-15 Active)

A z-transform is identified with 3 poles and 2 zeros, with parameter errors down in the level machine precision, and with similar small errors in the comparison of the raw and estimated outputs.

The one-step prediction problem does not depend on the initial conditions. This is demonstrated by estimating the parameters using a mid-stream, two-second segment of the input and output time responses. The results are shown in Figure 130. The performance is equally good. Pretty much anything works for low order problems using simulated data.

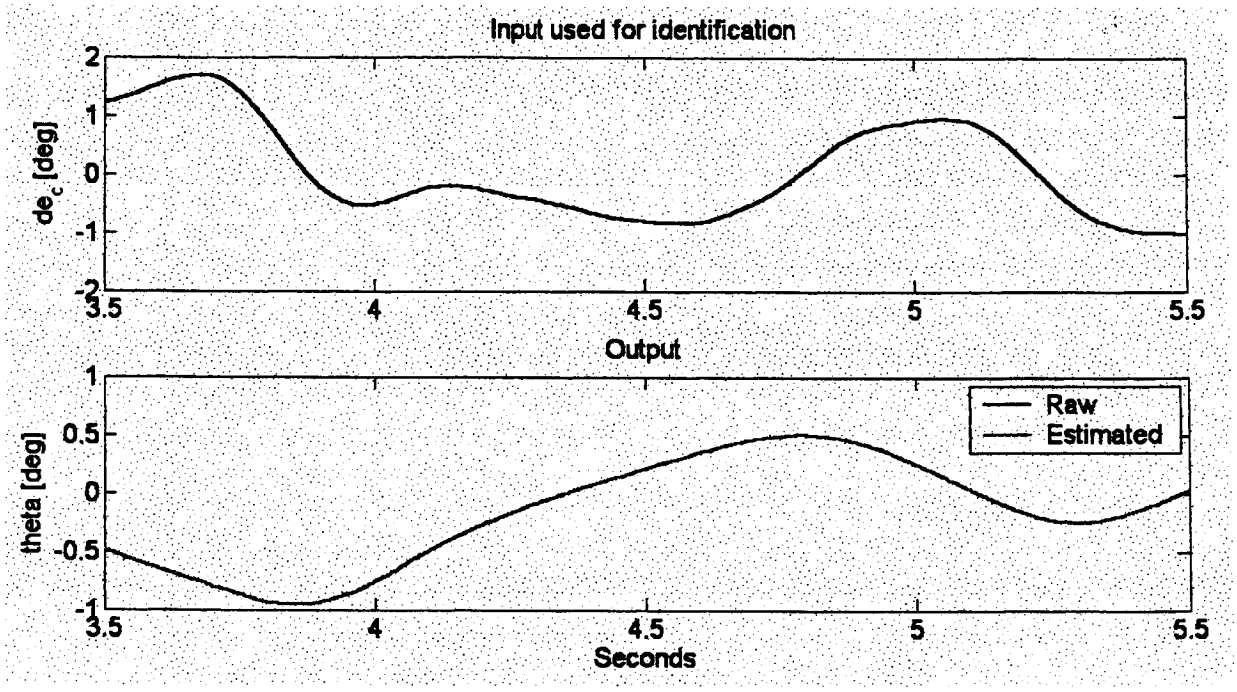


Figure 130. Time Domain Estimation with Non-Zero Initial Conditions

3.7.2 Ideal, High Order Problem

Expand the F15 example by using fifth order aircraft model and by including actuator and sensor models, but keep the estimation problem ideal by using simulated data. The pitch SCAS is shown in Figure 131:

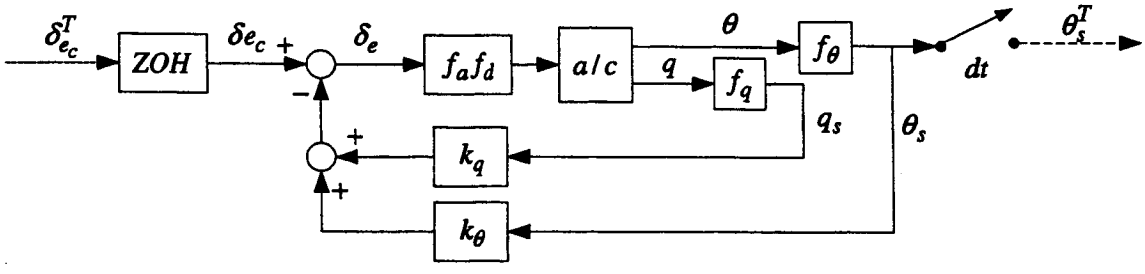


Figure 131. F-15 Active Pitch SCAS (High Order)

The state space model for the aircraft has 5 states:

$$\frac{d}{dt}\begin{bmatrix} v \\ \alpha \\ q \\ \theta \\ h \end{bmatrix} = \begin{bmatrix} -0.017 & -0.0709 & 0 & -0.5595 & 0 \\ -0.0068 & -0.9543 & 1 & 1e\text{-}4 & 1e\text{-}4 \\ 0.0689 & 7.1125 & -0.777 & 0 & 2e\text{-}4 \\ 0 & 0 & 1 & 0 & 0 \\ -0.0017 & -15.63 & 0 & 15.63 & 0 \end{bmatrix}\begin{bmatrix} v \\ \alpha \\ q \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} -0.2046 & -0.1084 & -0.0173 \\ -0.1199 & -0.0297 & -0.0290 \\ -12.44 & 6.346 & -3.104 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \delta_{stab} \\ \delta_{can} \\ \delta_{noz} \end{bmatrix}$$

The additional variables not in the low order problem are:

v [ft/sec] = forward velocity

h [ft] = altitude

q_s [deg/sec] = sensed pitch rate

θ_s [deg] = sensed pitch attitude

$f_a(s) = \omega_a^2 / [\zeta_a, \omega_a]$ = actuator model with $\zeta_a = 0.707$, $\omega_a = 30$ rad/sec

$f_d(s) = (\tau_d^2 s^2 - 6\tau_d s + 12)/(\tau_d^2 s^2 + 6\tau_d s + 12)$ = 2nd order Padé of delay with $\tau_d = 0.05$ sec

$f_q(s) = \omega_q^2 / [\zeta_q, \omega_q]$ = pitch rate sensor model with $\zeta_q = 0.707$, $\omega_q = 300$ rad/sec

$f_\theta(s) = \omega_\theta^2 / [\zeta_\theta, \omega_\theta]$ = pitch attitude sensor model with $\zeta_\theta = 0.707$, $\omega_\theta = 100$ rad/sec

Of course the model can get even more complicated with higher order and separate actuator models, structure dynamics, and structure filters, and so on. Selected transfer functions are:

$$\text{Open loop: } \frac{q}{\delta_e} = \frac{-21.9(0)(0.00165)(0.0165)(0.992)}{(0.0243)[-0.217, 0.0321](-1.8)(3.54)}$$

$$\text{Open loop: } \frac{\theta}{\delta_e} = \frac{-21.9(0.00165)(0.0165)(0.992)}{(0.0243)[-0.217, 0.0321](-1.8)(3.54)}$$

$$\text{Closed loop: } \frac{\theta}{\delta_{e_c}} = \frac{-2.19e5(0.00165)(0.0165)(0.992)[0.707, 30][0.866, 69.3][0.707, 300]}{(0.00134)(0.0166)(1.4)(3.73)[0.203, 10.9][0.759, 59.8](64.1)[0.706, 100][0.707, 300]}$$

Apply the same input as in Figure 129 and then estimate the transfer function. A plot of the raw and estimated outputs has a pixel or two difference between them, with both outputs looking just about the same as in Figure 129, and so a new figure is not needed. The estimation routine reported a one-step prediction error of 5.6e-8 and a simulation error is 0.991. The estimated transfer function converted from the z-domain to the Laplace domain is:

$$\frac{\theta}{\delta_{e_c}} = \frac{1.91e\text{-}6(0.995)[0.704, 30](77.7)(-400)(457)(-506)[0.793, 2.57e3](-1.28e4)}{(1.32)(3.76)[0.205, 11][0.844, 59.6][0.751, 101](2.78e3)(2.48e4)}$$

The mid frequency modes between about 0.1 and 100 rad/sec are well represented, but not exactly, differing in the second or third significant digit. The lower and higher frequency modes are either missed or not correctly located. Of course these low and high frequency modes make precious little contribution to the output response, but the point is that the exact model is no longer estimated. The setup of the estimation problem used 10 poles, a relative difference of one, and used scaling and orthogonal polynomials. The high order estimation using a two second segment of data did almost as well.

When the estimation routine was directed to identify a fourth order system the result had a higher one-step prediction error of 2.2e-6, actually a lower simulation error of 0.673, and an estimated transfer function of:

$$\frac{\theta}{\delta_{e_c}} = \frac{-0.000203(1.76)(30.1)(-152)(-400)}{[0.923,2.86][0.201,10.9]}$$

It is seen that matching the time response is easier then matching the pole and zero locations, which is a common theme in the world of estimation.

3.7.3 LOCATS Pitch Response From Fixed Base Simulator Data

An objective of this project is to test estimation methods using data from flight simulators and flight tests. Here we use data from a hardware-in-the-loop manned flight simulator for a large military transport, Run 68 from the LOCATS data base described in Volume II of this study. The simulator is represented in the Figure 132 block diagram.

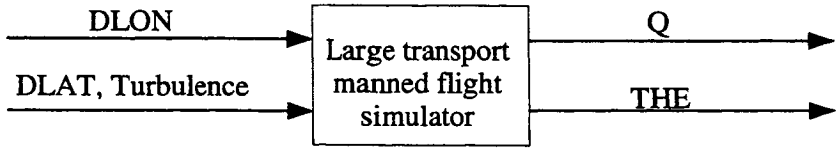


Figure 132. Manned Flight Simulator

The pilot inputs are DLON and DLAT, the longitudinal and lateral stick positions in inches. There is a turbulence input, turned on for this run, and there is also an auto-throttle loop turned on for this run. The auto-throttle loop is an internal feedback inside the main block. Here the interest is the closed loop response from DLON to Q and THE, respectively pitch rate and pitch attitude. The simulator is 6 DOF nonlinear, with a full aero model and nonlinear actuator models, the actual flight computers, and the actual stick. The stick has a deadband, friction, and preload, worse in the lateral axis, but still present in the longitudinal axis. We have not tried to extract a linear model from the simulator documents, and so we do not have a "truth" model for this example. The flight condition for this run is lightweight cruise. The input is from a pilot performing a pitch-tracking task.

The time responses compared with the estimated output are shown in Figure 133a. The fifth order identified transfer function for pitch attitude is:

$$\frac{\text{THE}}{\text{DLON}} = \frac{-0.00537(-9.91)(41.9)(-100)(-133)(1.53e3)}{(0.0795)(3.21)[0.507,131](595)}$$

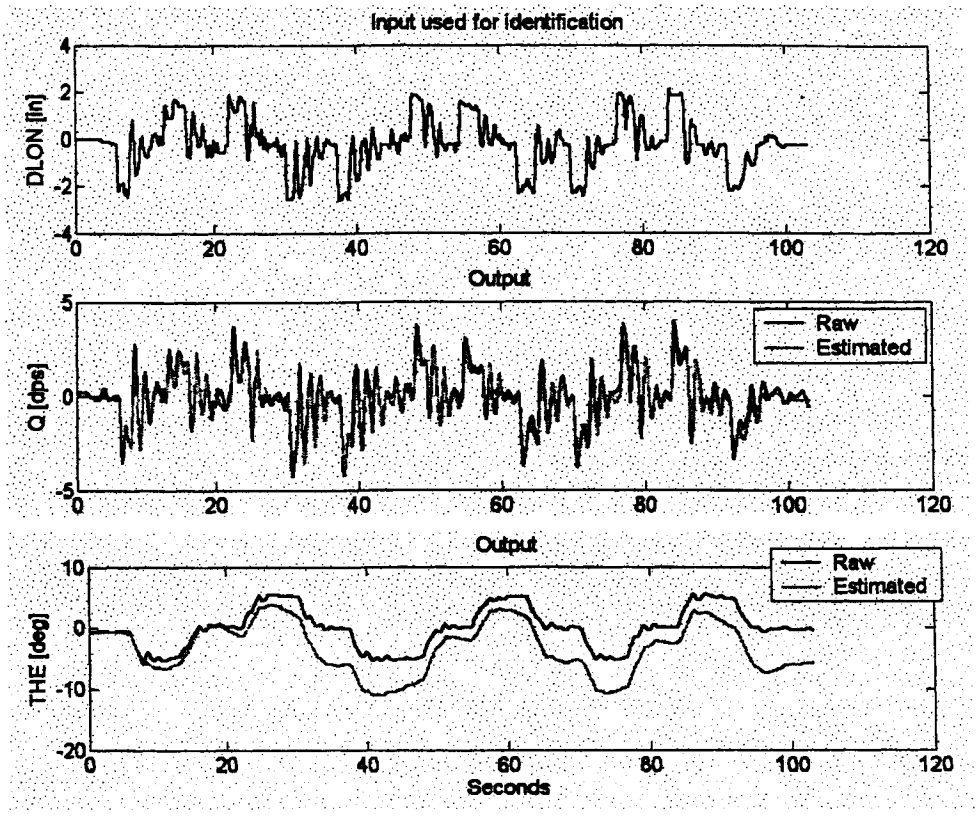
The fit is better when a higher order transfer function is used, but not significantly. The most notable feature in the estimated response is a significant low frequency error.

Switch to the pitch rate, and the estimated output has a better fit. The estimated transfer function is:

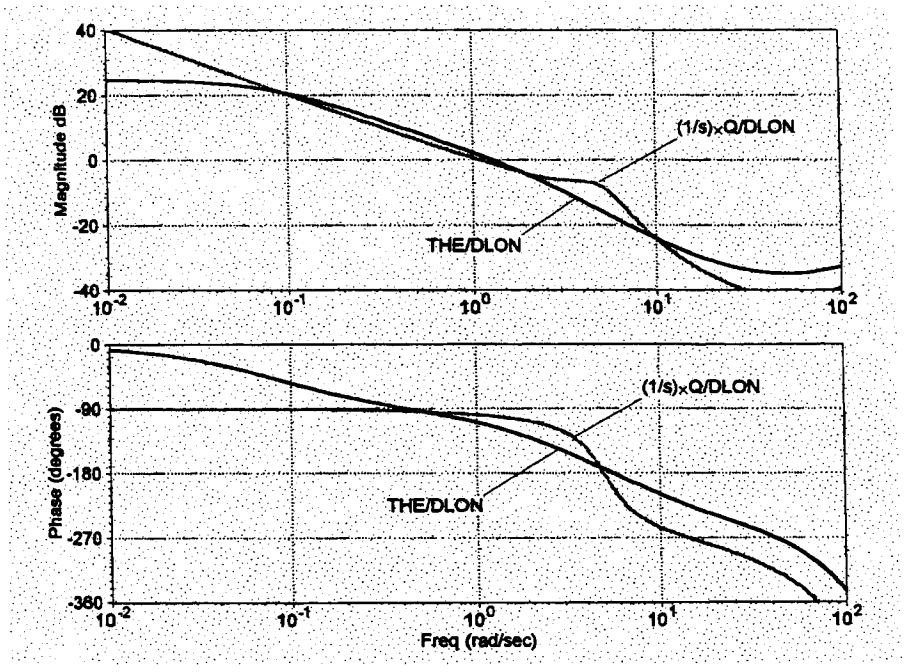
$$\frac{\text{Q}}{\text{DLON}} = \frac{5.43(10.1)(-11.6)(-100)[-0.862,197]}{[0.317,5.06][0.172,105](8.31e3)}$$

The dominant poles should be the same, or at least close, but this did not happen. The two output signals have synchronization errors, with many samples being help for two cycles, resulting in an increased signal-to-noise ratio. Pitch rate does better because it has more energy in the mid-frequency range. The frequency response of THE/DLON and the integrated Q/DLON are compared in Figure 133b. A significant difference is seen in the 1 to 10 rad/sec frequency range.

Better estimation is needed, which is a good reason to switch to the frequency domain.



a) Time Responses



b) Frequency Responses (of estimated transfer functions)

Figure 133. Time Domain Estimation of Q/DLON and THE/DLON

4. GENERALIZING THE LLS PROBLEM USING WEIGHTS

The parameters to be estimated are coefficients of the z-transform:

$$G(z)=\frac{b_m z^m+\cdots+b_0}{z^n+a_{n-1}z^{n-1}\cdots+a_0}$$

or equivalently the coefficients of the difference equation:

$$y_k=-(a_{n-1}y_{k-1}+\cdots+a_0 y_{k-n})+(b_m u_{k+m-n}+\cdots+b_0 u_{k-n})$$

The linear estimation problem minimizes the weighted cost $J=e^H We$ with the error:

$$e=y-F\theta$$

The vector y and the matrix F are built up from measurements of y_k and u_k for $0\le k<N$. The parameter vector is:

$$\theta^T=(a_{n-1}\cdots a_0\ b_m\cdots b_0)$$

In this Section the error vector is generalized to:

$$e=[y\ \ -FR]UU^{-1}\begin{bmatrix}1\\ \theta\end{bmatrix}$$

which reduces to the former when R and U are both the identity matrix. The problem variations are determined by the choices:

R = restrictions

W = error weight

U = parameter transformations

It is rare that setting each of these weights to identity gives the best solution – usually just for low order academic problems. Choices for each of the weights are presented.

4.1 Error Weights

The error weight has been defined as part of the original problem. Now it is time to show that this weight is quite powerful.

4.1.1 Time Variations

The error weight can be defined to implement boxcar estimation. Define W to be a diagonal matrix with ones and zeros on the diagonal that selects the previous N inputs and outputs. The parameter estimates will then change if the system changes. The point to be made is that this variation of the estimation problem amounts to an appropriate selection of the error weight. Another point to make is that the solution in this case and for most cases that we will consider can be computed more efficiently than by just blindly using the general equations. Here the famous recursive solution to the estimation problem is a much more efficient implementation of the general solution.

Similarly, the error weight can be defined to implement estimation with an exponential "forgetting factor." Define $W=\mathrm{diag}\{\alpha^{N-i}\}$, so that previous inputs and outputs are multiplied by successively higher powers of α, where $0<\alpha\le 1$. This solution can also be implemented recursively.

4.1.2 Averaging

Adjacent inputs and outputs can be averaged and the average values can be used to estimate the parameters. Use the following error weight to compute a running average of n_{avg} adjacent values:

$$W^{1/2} = \frac{1}{n_{avg}} \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 1 & 1 & 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 1 & \ddots & 1 & 1 & 0 & \ddots \\ 0 & 1 & \ddots & 1 & 1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

The total weight is:

$$W = W^{T/2} W^{1/2}$$

and where $W^{T/2}$ is the transpose of $W^{1/2}$. The system that is estimated will be the same due to superposition and time invariance. In other words, the averaged output due to the averaged input will be the same as computing the averaged output from shifted versions of the input computed separately. This averaging will reduce the signal to noise ratio if a frequency separation exists between the signal and noise.

4.1.3 Filtering

The running average is an example of the more general operation of filtering. Use the same filter on both the input and output:

$$\tilde{u}_k = \sum_{i=0}^{k} u_i f_{k-i} \quad , \quad \tilde{y}_k = \sum_{i=0}^{k} y_i f_{k-i}$$

This is the same as using the error weight:

$$W^{1/2} = \begin{bmatrix} f_0 & 0 & 0 & 0 \\ f_1 & f_0 & 0 & 0 \\ f_2 & f_1 & f_0 & 0 \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

and where the total weight is $W = W^{T/2} W^{1/2}$. It is more efficient to compute the filter response once and then build up the estimation problem using the output of the filters. This filtering will reduce the signal to noise ratio if a frequency separation exists between the signal and noise.

4.2 Frequency Domain Estimation

4.2.1 Transforming to the Frequency Domain

Linear least squares estimation in the time domain minimizes the one-step prediction error. The same problem can be posed in the frequency domain. Here we define a weight that transforms the time domain problem into the frequency domain. It will turn out that the weight is a factored version of the identity matrix, and so the transformed problem will give exactly the same answer.

Define the digital Fourier transform (DFT) and its inverse:

$$X_k = \sum_{i=0}^{N-1} x_i e^{-ijk2\pi/N} \quad , \quad x_i = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{ijk2\pi/N}$$

The DFT can equivalently be written as a vector inner product:

$$X_k = \begin{pmatrix} 1 & \cdots & z_k^{-(N-1)} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{N-1} \end{pmatrix}$$

where $z_k = e^{jk2\pi/N}$. Build up the weight using the row vectors of the inner product. Define the weight as $W = W^{H/2} W^{1/2} / N$ where:

$$W^{1/2} = \begin{bmatrix} w_0^T \\ \vdots \\ w_{N-1}^T \end{bmatrix} \quad , \quad w_k^T = \begin{pmatrix} 1 & \cdots & z^{-k(N-1)} \end{pmatrix}$$

Multiplying by the weight $W^{1/2}$ is equivalent but slower to using the Fast Fourier Transform (FFT), so of course in the implementation we will use the FFT, and the weight is a theoretical device that will not actually be implemented. The division by N is needed because $w_k^H w_k = N$. It follows that the total weight $W = I$, so by using this weight the solution is not changed. What this weight allows us to do is transform the error to the frequency domain:

$$E = W^{1/2} e = W^{1/2}(F\theta - y) = W^{1/2} F\theta - Y$$

The solution in terms of the transformed error is:

$$\theta = \begin{bmatrix} \mathrm{Re}\left(W^{1/2}F\right) \\ \mathrm{Im}\left(W^{1/2}F\right) \end{bmatrix}^{+} \begin{bmatrix} \mathrm{Re}(Y) \\ \mathrm{Im}(Y) \end{bmatrix}$$

The optimal cost is equivalently stated in the time and frequency domain:

$$J = e^T e = E^H E / N$$

This equivalence is called Parseval's theorem.

To summarize, the DFT has been computed of the output vector y and each of the columns of the data matrix F, which transforms the problem to minimizing the frequency domain error between the output and one-step predicted output. With these transformations the time and frequency domain solutions are exactly the same, but there is as yet there is no practical benefit to doing this transformation. There soon will be, however, when a subset of the frequency response is used to filter the measurements, and averaging in the frequency domain is used to reduce the effects of noise. In addition, under certain conditions that we now explore the DFT of the input and output only needs to be computed once.

4.2.2 The Transformed Data Matrix

The data matrix from the time domain problem is:

$$F = \begin{bmatrix} \phi_0^T \\ \vdots \\ \phi_{N-1}^T \end{bmatrix}, \quad \phi_i^T = \begin{pmatrix} -y_{i-1} & \cdots & -y_{i-n} & u_{i+m-n} & \cdots & u_{i-n} \end{pmatrix}$$

A key observation is that the columns of F are delayed versions of the input and output, and it will turn out under certain conditions that:

$$F = \left[-y\begin{pmatrix} z^{-1} & \cdots & z^{-n} \end{pmatrix} \quad u\begin{pmatrix} z^{m-n} & \cdots & z^{-n} \end{pmatrix} \right]$$

where z^{-1} is the shift operator. If these as yet unnamed conditions are true, it follows that:

$$W^{1/2}F = \begin{bmatrix} \psi_0^T \\ \vdots \\ \psi_{N-1}^T \end{bmatrix}, \quad \psi_k^T = \left[\begin{pmatrix} z_k^{-1} & \cdots & z_k^{-n} \end{pmatrix} Y_k \quad \begin{pmatrix} z_k^{m-n} & \cdots & z_k^{-n} \end{pmatrix} U_k \right]$$

The FFT of the input and output only have to be computed one time and then the data matrix is built up from the FFT coefficients and the powers of the digital frequencies. In addition, we can divide through by U_k, call $G_k = Y_k / U_k$ the estimated frequency response of the system, and transform the problem into one of minimizing the frequency response of the system.

Now let us examine the conditions for which the above equation is true. Expand one of the elements of $W^{1/2}F$, choose say the $k2$ element, and from this develop a general pattern. The DFT of the $k2$ element is:

$$\begin{pmatrix} 1 & \cdots & z_k^{-(N-1)} \end{pmatrix} \begin{pmatrix} y_{-2} \\ y_{-1} \\ y_0 \\ \vdots \\ y_{N-3} \end{pmatrix} = \sum_{i=0}^{N-1} y_{i-2} z_k^{-i}$$

$$= \underbrace{\left(y_{-2} + z_k^{-1} y_{-1} \right) - z_k^{-N} \left(y_{N-2} + y_{N-1} z_k^{-1} \right)}_{\text{extra terms assumed to be zero}} + z_k^{-2} \underbrace{\sum_{i=0}^{N-1} y_i z_k^{-i}}_{Y_k}$$

Note that $z_k^{-N} = 1$, and then it follows for $m = n$ that:

$$W^{1/2}F = \begin{bmatrix} \psi_0^T \\ \vdots \\ \psi_{N-1}^T \end{bmatrix}, \quad \psi_k^T = \left[\begin{pmatrix} z_k^{-1} & \cdots & z_k^{-n} \end{pmatrix} Y_k + \alpha_k \quad \begin{pmatrix} 1 & \cdots & z_k^{-n} \end{pmatrix} U_k + \begin{pmatrix} 0 & \beta_k \end{pmatrix} \right]$$

where:

$$\alpha_k = \begin{pmatrix} 1 & \cdots & z_k^{1-n} \end{pmatrix} \left[\begin{pmatrix} y_{-1} & y_{-2} & \cdots & y_{-n} \\ 0 & y_{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & y_{-2} \\ 0 & \cdots & 0 & y_{-1} \end{pmatrix} - \begin{pmatrix} y_{N-1} & y_{N-2} & \cdots & y_{N-n} \\ 0 & y_{N-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & y_{N-2} \\ 0 & \cdots & 0 & y_{N-1} \end{pmatrix} \right]$$

$$b_k = \begin{pmatrix} 1 & \cdots & z_k^{1-n} \end{pmatrix} \left[\begin{pmatrix} u_{-1} & u_{-2} & \cdots & u_{-n} \\ 0 & u_{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{-2} \\ 0 & \cdots & 0 & u_{-1} \end{pmatrix} - \begin{pmatrix} u_{N-1} & u_{N-2} & \cdots & u_{N-n} \\ 0 & u_{N-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{N-2} \\ 0 & \cdots & 0 & u_{N-1} \end{pmatrix} \right]$$

Therefore, the equation is true if $\alpha_k = 0$ and $\beta_k = 0$. Basically, the first and last n inputs and outputs need to be zero. If so, the solution of the frequency domain estimation problem using this equation will be exactly the same as the time domain, one-step prediction error minimization problem.

4.3 Restrictions

Restrictions are used when parts of the transfer function have a fixed form, such as a known relative order, a delay, an integrator, and/or zeros at infinity. In each of these cases the practical affect is to reduce the number of parameters for the estimation problem. The error vector is:

$$e = y - FR\theta$$

and the solution of the LLS estimation problem is:

$$\theta = (WFR)^{+}(Wy)$$

4.3.1 Relative Order

The relative order of the z-transform $G(z)$ is the difference $n-m$. This difference corresponds to a digital delay of $n-m$ steps, in the sense that the input u_k does not affect the output until that many steps later. One way to interpret the numerator is as an nth order polynomial with the high order coefficients restricted to be zero. The estimation problem can be setup using the nth order polynomial and then "asked" to estimate these high order coefficients; alternatively a restricted version of the problem can be defined using:

$$R\theta = \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ 0 \\ \vdots \\ 0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

You could form this R matrix and carry out the matrix multiplication FR, but it is easier to just select the last m columns of the input. This is what we have been doing all along.

4.3.2 Delay

Assume that the input is delayed by d steps, which changes the z-transform to:

$$G(z) = \frac{b_n z^n + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0} z^{-d}$$

The order of the system increases to $n+d$, but there is no need to have the estimation problem find the denominator coefficients that are known to be zero. Restrict the parameters using the following R matrix:

$$R\theta = \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ 0 \\ \vdots \\ 0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

This restriction looks the same as used above to define a relative order, but this time the extra zeros "belong" to the denominator. The error vector is:

$$e = y - FR\theta = \begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \\ y_n \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} -y_{-1} & \cdots & -y_{-n} & u_{m-n-d} & \cdots & u_{-n-d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{n-2} & \cdots & -y_{-1} & u_{m-1-d} & \cdots & u_{-1-d} \\ -y_{n-1} & \cdots & -y_0 & u_{m-d} & \cdots & u_{-d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n} & u_{N+m-n-d} & \cdots & u_{N-n-d} \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

The best way to form the matrix FR is just to shift the input down by d steps. There is no need to actually multiply the original matrix F by a bunch of ones and zeros.

Converting From Analog to Digital Delay:

Consider the case where an analog system with a delay of τ seconds is converting to a digital system using the ZOH equivalent transformation. The block diagram is shown in Figure 134.

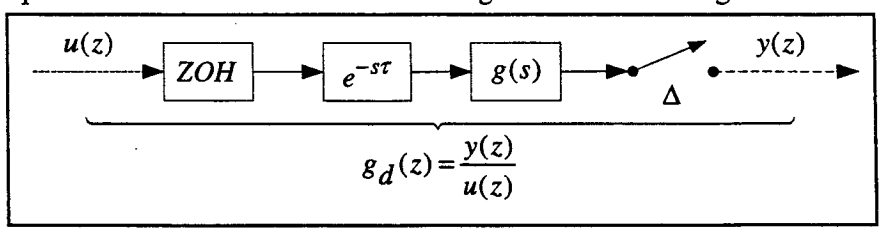


Figure 134. Conversion of an Analog System using a Zero Order Hold

The z-transform has the form:

$$\frac{y(z)}{u(z)} = \frac{b_n z^n + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0} z^{-d}$$

The order n is the same as the order of the analog system. The analog delay is:

$$\tau = d\Delta - \mu$$

where Δ is the sample period and $0 \le \mu < \Delta$. The analog delay is μ less than d full sample periods. Note that both the numerator and denominator polynomials have the same order n.

Estimating the Delay:

Step through different values of d and choose the delay that gives the smallest error. Decrease the number of times the linear problem is solved by using a trisection search for the minimizing delay.

4.3.3 Zeros at Infinity

Define the z-transform to be of the form:

$$G(z)=\frac{(z+1)^{n-m}\left(b_m z^m+\cdots+b_0\right)}{z^n+a_{n-1}z^{n-1}+\cdots+b_0}$$

The zeros at $z=-1$ are called "zeros at infinity." The numerator is nth order but only m parameters need to be estimated. Define the restricted set of parameters:

$$R\theta=\begin{bmatrix} I & 0 \\ 0 & R_u \end{bmatrix}\begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

where for $m=2$ and $n-m=1$:

$$R_u\begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}$$

and for $m=2$ and $n-m=2$

$$R_u\begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}$$

A pattern emerges; the columns of R_u contain coefficients of $(z+1)^{n-m}$. The easiest way to form the product FR is to successively add columns of u.

Interpretation of Zeros At Infinity Using Laplace Transforms:

Define the Laplace transform:

$$G_{analog}(s)=\frac{d_m s^m+\cdots+d_0}{s^n+c_{n-1}s^{n-1}\cdots+c_0}$$

The relative order $n-m$ equals the number of zeros at infinity. The term "zero at infinity" means that for large s the magnitude of the transfer function approaches zero. Defining an "order" for this zero has the interpretation that for large s the transfer function is approximated by d_m/s^{n-m}. The relative order

determines the high frequency characteristics of the Laplace transform. The magnitude has a high frequency roll-off of $20(n-m)$ dB/dec and the phase asymptotically approaches $-90(n-m)$ degrees. Now convert the Laplace transform to the z-domain using the bilinear transformation:

$$G(z) = G_{analog}(s)\Big|_{s=\frac{(2/h)(z-1)}{(z+1)}}$$

where h is the sample period, and to go the other way:

$$G_{analog}(s) = G(z)\Big|_{z=\frac{s+2/h}{-s+2/h}}$$

The z-transform will have a set of zeros at $z=-1$:

$$G(z) = \frac{(z+1)^{n-m}\left(b_m z^m + \cdots + b_0\right)}{z^n + a_{n-1}z^{n-1} + \cdots + b_0}$$

These zeros are considered to be "digital zeros at infinity." The z-transform can be defined to have $n-m$ digital zeros at infinity, which will in turn specify the high frequency behavior of the corresponding Laplace transform.

Interpretation Using the Difference Equation:

The difference equation for the digital system with $n-m$ zeros at infinity is:

$$y_k = -\left(a_{n-1}y_{k-1} + \cdots + a_0 y_{k-n}\right) + \left(b_m v_k + \cdots + b_0 v_{k-m}\right)$$

where the altered input v_k is defined by the binomial expansion:

$$v(z) = \left(1 + z^{-1}\right)^{n-m} u(z)$$

where:

$$v_k = u_k + u_{k-1} \text{ for } n-m=1$$

$$v_k = u_k + 2u_{k-1} + u_{k-2} \text{ for } n-m=2$$

and in general:

$$v_k = \sum_{j=0}^{n-m}\binom{n-m}{j}u_{k-j} \quad , \text{ where } \binom{p}{q} = \frac{p!}{(p-q)!q!}$$

Start with the input u_k, from this compute the altered input v_k, and then from the altered input form the columns of the data matrix F used in the estimation problem.

4.3.4 Integrators

Define the z-transform to be of the form:

$$G(z) = \frac{\left(b_m z^m + \cdots + b_0\right)}{(z-1)^r\left(z^n + a_{n-1}z^{n-1} + \cdots + b_0\right)}$$

The poles at $z=+1$ are integrators. The numerator is $n+r^{th}$ order but only n parameters need to be estimated. Define the restricted set of parameters:

$$R\theta = \begin{bmatrix} R_y & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix}$$

where for $n=3$ and $r=1$:

$$R_y \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

and for $n=2$ and $r=2$

$$R_y \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

Again a pattern emerges; this time the columns of R_y contain coefficients of $(z-1)^r$. The easiest way to form the product FR is to successively subtract the columns of y. The corresponding Laplace transform is:

$$G_{analog}(s) = \frac{d_m s^m + \cdots + d_0}{s^r \left(s^n + c_{n-1} s^{n-1} \cdots + c_0 \right)}$$

The altered difference equation is:

$$w_k = -\left(a_{n-1} w_{k-1} + \cdots + a_0 w_{k-n} \right) + \left(b_m u_{k+m-n-r} + \cdots + b_0 u_{k-n-r} \right)$$

where:

$$w(z) = \left(1 - z^{-1} \right)^r y(z)$$

4.4 Parameter Transformations

4.4.1 LLS Solution with Parameter Transformation

The original error is:

$$e = y - F\theta$$

A transformation can be placed between the data matrix F and the parameter vector θ:

$$e = y - FU_{smaller}U_{smaller}^{-1}\theta$$

It turns out, however, to be more useful to transform a combination of the data matrix F and the output vector y:

$$e = \begin{bmatrix} y & -F \end{bmatrix} UU^{-1} \begin{bmatrix} 1 \\ \theta \end{bmatrix}$$

This is the parameter transformation that we will use. To find the parameter vector that minimizes the error first define:

$$\begin{bmatrix} y & -F \end{bmatrix} U = \begin{bmatrix} y_1 & -F_1 \end{bmatrix} \quad , \quad U^{-1} \begin{bmatrix} 1 \\ \theta \end{bmatrix} = \begin{bmatrix} \mu \\ \theta_1 \end{bmatrix}$$

so that:

$$e = \mu y_1 - F_1\theta_1$$

and therefore the transformed parameter vector is:

$$\theta_1 = \mu(WF_1)^{+}(Wy_1)$$

The coefficient μ is not known apriori, but can be computed from the first row of:

$$U \begin{bmatrix} \mu \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \theta \end{bmatrix}$$

Resulting in:

$$\mu = \left[1 - U(1,2:p)\theta_1\right] / U(1,1)$$

and where p is the dimension of the square matrix U. In the case where the transformation matrix U is diagonal or lower triangular, which will happen sometimes, then μ can be computed apriori to be $\mu = 1/U(1,1)$. Finally, the untransformed parameters are:

$$\theta = U(2:p,:) \begin{bmatrix} \mu \\ \theta_1 \end{bmatrix}$$

4.4.2 Scaling

Scaling can reduce the dynamic range of polynomial coefficients. Start with:

$$G(z) = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_0}$$

Scale by replacing z by αz:

$$G(\alpha z) = \frac{b_m \alpha^{m-n} z^m + \cdots + b_0 \alpha^{-n}}{z^n + a_{n-1}\alpha^{-1} z^{n-1} + \cdots + a_0 \alpha^{-n}}$$

The scaling is implemented using the diagonal transformation matrix:

$$U = \text{diag}\left\{1 \cdots \alpha^n \ \alpha^{m-n} \cdots \alpha^{-n}\right\}$$

Poles and zeros at z_i are transformed to z_i/α. Scaling is more important for Laplace transforms then z-transforms. A good value to use for scaling is $\alpha = \omega_{max}$, where ω_{max} is the highest frequency of interest.

4.4.3 Orthogonal Polynomials

Better numerical properties for estimation can be achieved by transforming regular polynomials to orthogonal polynomials. The transfer function is equivalent to:

$$G(z) = \frac{d_m p_m(z) + \cdots + d_0}{c_m p_n(z) + \cdots + c_0}$$

where the $p_i(z)'s$ are orthogonal polynomials. Chebyshev polynomials are recommended, and the transformation matrices are defined in the Mathematical Tools Section. The Chebyshev transformation matrix for the case $n = m = 3$ is:

$$U = \text{diag}\{U_a\, U_b\} \quad , \text{ where } U_a = U_b = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

The transformation matrix is lower triangular with $\mu = 1/U(1,1) = 2^{1-n}$. When estimating Laplace transforms scaling should first be done using $\alpha = \omega_{max}$.

4.4.4 Delta Transformation (the Forward Rectangular Transformation)

The poles and zeros of a z-transform cluster around $z = +1$, causing loss of precision for high order systems. The delta transformation is:

$$\delta = \frac{z-1}{\alpha T}$$

where T [seconds] is the sample period. This transformation shifts the poles and zeros so they are centered around $z = 0$ and then scales them. The scaling coefficient $\alpha = 1$ makes the delta transformation the same as the forward rectangle approximation of the derivative, and hence the alternative name.

The delta transformation can be implemented using the matrix $U = \text{diag}\{U_a\, U_b\}$, which separately transforms the numerator and denominator. To derive the transformation matrix we will first examine first and second order polynomials, from which a general pattern is deduced. Reduce the algebraic clutter a bit by setting $\beta = \alpha T$, so that the delta transformation becomes $\delta = (z-1)/\beta$. The transformed polynomials are:

$$(z\ 1)\begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = (z\ 1)\underbrace{\frac{1}{\beta}\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}}_{U_b}\underbrace{\begin{pmatrix} \beta & 0 \\ 1 & 1 \end{pmatrix}}_{U_b^{-1}}\begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = (\delta\ 1)U_b^{-1}\begin{pmatrix} b_1 \\ b_0 \end{pmatrix}$$

$$\left(z^2\ z\ 1\right)\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix} = \left(z^2\ z\ 1\right)\underbrace{\frac{1}{\beta^2}\begin{pmatrix} 1 & 0 & 0 \\ -2 & \beta & 0 \\ 1 & -\beta & \beta^2 \end{pmatrix}}_{U_b}\underbrace{\begin{pmatrix} \beta^2 & 0 & 0 \\ 2\beta & \beta & 0 \\ 1 & 1 & 1 \end{pmatrix}}_{U_b^{-1}}\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix} = \left(\delta^2\ \delta\ 1\right)U_b^{-1}\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix}$$

Columns $i=1,\ldots,m+1$ of U_b contain coefficients of $[(z-1)/\beta]^{m+1-i}$. The transformation matrix U_b is computed using the Matlab commands:

```
m=5;
beta=2;
Ub=zeros(m+1,m+1);
Ub(m+1,m+1)=1;
for i=m:-1:1
  Ub(i:m+1,i)=([Ub(i+1:m+1,i+1);0]-Ub(i:m+1,i+1))/beta;
end
```

The transformation matrix U_a is computed similarly with n in place of m. The total transformation matrix is lower triangular with $\mu=1/U(1,1)=\alpha^{-n}$.

Scaling:

Suggested scaling:

$\alpha=1$ is no scaling. The digital zero at infinity, $z=-1$, is transformed to $|\delta|=2/T$. This choice makes the delta transformation the same as the forward rectangle approximation of a derivative. A problem with this choice is the transformed coefficients may have a large dynamic range, which is why the freedom of setting α to something other than unity is included.

$\alpha=2/T$ shifts $z=-1$ to $|\delta|=1$. The dynamic range is reduced, and the transformed frequencies are less than unity. Keeping $|\delta|\leq 1$ makes this a good first step for the Chebyshev transformation. The problem is that over the frequency range of interest it may be the case that $|\delta|\ll 1$, particularly when T is very small, and therefore ...

$\alpha=\omega_{max}$ shifts $z=-1$ to $|\delta|=2/(\omega_{max}T)$, where $\omega_{max}<2/T$ rad/sec is the largest frequency of interest. The transformed frequencies of interest cover more of the range $|\delta|\leq 1$ over which the Chebyshev polynomials are orthogonal.

$\alpha=|e^{j\omega_{max}T}-1|/T$ shifts $z=e^{j\omega_{max}T}$ to $|\delta|=1$, and makes even better use of the transformed frequency range $|\delta|\leq 1$. Note that in the limit as $T\rightarrow 0$ it follows that $\alpha\rightarrow\omega_{max}$.

The delta transformation is a scaled Laplace transform, where $s=\alpha\delta$. This suggests that the transformed LLS problem can be used to estimate the coefficients of a Laplace transform. The delta transformation just defined is one of several ways to get to the Laplace transform, and not the best way, with the bilinear transformation being preferred, and so we generalize the delta transformation and show how it is used to convert between analog and digital systems.

4.5 Digital to Analog Transformations

The transformation matrix U can be used to convert the estimation problem from z-transforms to Laplace transforms. You can estimate the z-transform coefficients and then transform the result, or as shown here you can directly compute the coefficients of the Laplace transform. The transformations of interest here are:

Forward Rectangle	$\delta=\dfrac{z-1}{\alpha T}$	$z=1+\alpha\delta T$
Backward Rectangle	$\delta=\dfrac{z-1}{z\alpha T}$	$z=\dfrac{1}{1-\alpha\delta T}$

$$\text{Bilinear} \qquad \delta = \frac{(2/(\alpha T))(z-1)}{z+1} \qquad z = \frac{\delta + 2/(\alpha T)}{-\delta + 2/(\alpha T)}$$

Scaling by α is included in the transformations, where $\delta = \alpha s$. The LLS problem is used to estimate the coefficients of the delta transformation, which has the allegedly good numerical properties, and then the Laplace transform is obtained by the final transformation $G(s) = G(\delta/\alpha)$. The forward rectangle transformation has already been covered, and so we start with the next on the list.

4.5.1 Backward Rectangle

Derive the transformation $U = \text{diag}\{U_a\, U_b\}$ as done before by expanding first and second order polynomials and then generalizing the result after establishing a pattern. Set $\delta = (z-1)/(\beta z)$ where $\beta = \alpha T$ and then:

$$\frac{1}{z}\begin{pmatrix} z & 1 \end{pmatrix}\begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = \frac{1}{z}\begin{pmatrix} z & 1 \end{pmatrix}\underbrace{\frac{1}{\beta}\begin{pmatrix} 1 & \beta \\ -1 & 0 \end{pmatrix}}_{U_b}\underbrace{\begin{pmatrix} 0 & -\beta \\ 1 & 1 \end{pmatrix}}_{U_b^{-1}}\begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} \delta & 1 \end{pmatrix}U_b^{-1}\begin{pmatrix} b_1 \\ b_0 \end{pmatrix}$$

$$\frac{1}{z^2}\begin{pmatrix} z^2 & z & 1 \end{pmatrix}\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix} = \frac{1}{z^2}\begin{pmatrix} z^2 & z & 1 \end{pmatrix}\underbrace{\frac{1}{\beta^2}\begin{pmatrix} 1 & \beta & \beta^2 \\ -2 & -\beta & 0 \\ 1 & 0 & 0 \end{pmatrix}}_{U_b}\underbrace{\begin{pmatrix} 0 & 0 & \beta^2 \\ 0 & -\beta & -2\beta \\ 1 & 1 & 1 \end{pmatrix}}_{U_b^{-1}}\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} \delta^2 & \delta & 1 \end{pmatrix}U_b^{-1}\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix}$$

The transformation is similar to the forward rectangle in that the columns of U_b contain coefficients of powers of $(z-1)/\beta$, but in different parts of the column. Build the matrix with the following snippet of Matlab code:

```
m=5;
beta=2;
Ub=zeros(m+1,m+1);
Ub(1,m+1)=1;
for i=m:-1:1
  Ub(1:m+2-i,i)=([0;Ub(1:m+1-i,i+1)]-Ub(1:m+2-i,i+1))/beta;
end
```

The transformation matrix U is not lower triangular and so the μ coefficient that is part of the LLS solution is not known apriori. There is another difference with the forward rectangular transformation in that the polynomials need to be multiplied by a power of $1/z$ to complete the transformation. The resulting delta transform is:

$$G(\delta) = G(z)\big|_{z=1/(1-\alpha\delta)} = \frac{(1-\alpha\delta)^{n-m}\left(d_m\delta^m + \cdots + d_0\right)}{\delta^n + c_{n-1}\delta^{n-1} + \cdots + c_0}$$

and so if there is a relative difference $n-m>0$ in the z-transform this results in a set of RHP zeros at $\delta = 1/\alpha$, which acts like a delay. To achieve a relative difference of $n-m$ in the delta transform (and hence the Laplace transform) the original z-transform should be restricted to the form:

$$G(z) = \frac{z^{n-m}\left(b_m z^m + \cdots + b_0\right)}{z^n + a_{n-1}z^{n-1} + \cdots + a_0}$$

The choices for the scaling coefficient α are the same as listed for the forward rectangular transformation.

4.5.2 Bilinear Transformation

The bilinear transformation has the desirable property that the stable regions in the analog and digital domains transform into each other. This is not true of the forward and backward rectangular transformations, and this is a good reason to use the more complicated bilinear transformation. Set $s=\beta(z-1)/(z+1)$ where $\beta=2/(\alpha T)$ and then:

$$\frac{1}{z+1}\begin{pmatrix}z & 1\end{pmatrix}\begin{pmatrix}b_1\\ b_0\end{pmatrix}=\frac{1}{z+1}\begin{pmatrix}z & 1\end{pmatrix}\underbrace{\begin{pmatrix}\beta & 1\\ -\beta & 1\end{pmatrix}}_{U_b}\underbrace{\frac{1}{2\beta}\begin{pmatrix}1 & -1\\ \beta & \beta\end{pmatrix}}_{U_b^{-1}}\begin{pmatrix}b_1\\ b_0\end{pmatrix}=\begin{pmatrix}\delta & 1\end{pmatrix}U_b^{-1}\begin{pmatrix}b_1\\ b_0\end{pmatrix}$$

$$\frac{\begin{pmatrix}z^2 & z & 1\end{pmatrix}}{(z+1)^2}\begin{pmatrix}b_2\\ b_1\\ b_0\end{pmatrix}=\frac{\begin{pmatrix}z^2 & z & 1\end{pmatrix}}{(z+1)^2}\underbrace{\begin{pmatrix}\beta^2 & \beta & 1\\ -2\beta^2 & 0 & 2\\ \beta^2 & -\beta & 1\end{pmatrix}}_{U_b}\underbrace{\frac{1}{(2\beta)^2}\begin{pmatrix}1 & -1 & 1\\ 2\beta & 0 & -2\beta\\ \beta^2 & \beta^2 & \beta^2\end{pmatrix}}_{U_b^{-1}}\begin{pmatrix}b_2\\ b_1\\ b_0\end{pmatrix}=\begin{pmatrix}s^2 & s & 1\end{pmatrix}U_b^{-1}\begin{pmatrix}b_2\\ b_1\\ b_0\end{pmatrix}$$

The pattern in the transformation matrix is not so obvious, but columns $i=1,\ldots,m+1$ of U_b contain coefficients respectively of $\beta^{m+1-i}(z-1)^{m+1-i}(z+1)^{i-1}$, built up via:

```
m=5;
beta=1;
Ub=zeros(m+1,m+1);
Ub(m+1,m+1)=1;
for i=m:-1:1
  Ub(i:m+1,i)=([Ub(i+1:m+1,i+1);0]-Ub(i:m+1,i+1))*beta;
  for i1=i+1:m+1
    Ub(i:m+1,i1)=([Ub(i+1:m+1,i1);0]+Ub(i:m+1,i1))*beta;
  end
end
```

The resulting delta transform is:

$$G(\delta)=G(z)\big|_{z=(\delta+\beta)/(-s+\beta)}=\frac{(-s+\beta)^{n-m}\left(d_m s^m+\cdots+d_0\right)}{s^n+c_{n-1}s^{n-1}+\cdots+d_0}$$

If there is a relative difference $n-m>0$ in the z-transform this results in a set of RHP zeros at $\delta=\beta=2/(\alpha T)$, which acts like a delay. To achieve a relative difference of $n-m$ in the delta transform (and hence the Laplace transform) the original z-transform should be restricted to the form:

$$G(z)=\frac{(z+1)^{n-m}\left(b_m z^m+\cdots+b_0\right)}{z^n+a_{n-1}z^{n-1}+\cdots+a_0}$$

The scaling choices listed earlier will work okay, but better is available. The digital zero at infinity, $z=-1$, maps to $\delta=\infty$, and for this reason $\alpha=2/T$ is not a good choice. The scaling $\alpha=\omega_{max}$ is still reasonable, but better is:

$$\alpha=\frac{(2/T)(e^{j\omega_{max}T}-1)}{e^{j\omega_{max}T}+1}=\frac{2}{T}\tan\frac{\omega_{max}T}{2}\rightarrow\omega_{\max}\ \text{ as }\ T\rightarrow 0$$

5. FREQUENCY DOMAIN IDENTIFICATION

5.1 LLS Estimation of the Laplace Transform

5.1.1 Problem Statement

The identification routines up to now have used time series to estimate the transfer function. The time series can first be used to compute the frequency response, and then the frequency response used to estimate the coefficients of the transfer function. The frequency data is available at a set of N frequencies:

$$\omega_i = \text{frequencies, for } i = 1,\ldots,N$$

$$\mathcal{G}_i = \text{frequency response, for } i = 1,\ldots,N$$

Define the transfer function in a general way as:

$$G(\theta,\omega) = \text{transfer function}$$

$$\theta = \text{parameter vector}$$

The objective is to find the parameter vector that minimizes the cost function:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i - G(\theta,\omega_i) \right|^2$$

The problem is in general a nonlinear minimization problem. We will start with linear version of the problem.

5.1.2 LLS Estimation

The transfer function to be identified is the Laplace transform:

$$G(s) = \frac{c(s)}{d(s)} = \frac{b_m s^m + \cdots + b_0}{s^n + a_{n-1}s^{n-1} \cdots + a_0}$$

The Laplace variable is $s = j\omega$. The parameter vector is:

$$\theta^T = \begin{bmatrix} a_{n-1} & \cdots & a_0 & b_m & \cdots & b_0 \end{bmatrix}$$

and the cost function to be minimized is:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i d(s) - c(s) \right|^2$$

The cost function is equivalent to:

$$J = e^H W e$$

where:

$$e = A\theta - b \quad , \quad W = \text{diag}(w_i)$$

$$b = \begin{bmatrix} s_1^n \mathcal{G}_1 \\ \vdots \\ s_N^n \mathcal{G}_N \end{bmatrix} \quad , \quad A = \begin{bmatrix} \phi_1^H \\ \vdots \\ \phi_N^H \end{bmatrix} \quad , \quad \phi_i^H = \begin{bmatrix} -\mathcal{G}_i \left(s_i^{n-1} & \cdots & 1 \right) & \left(s_i^m & \cdots & 1 \right) \end{bmatrix}$$

and where e^H is the complex conjugate transpose (aka the Hermitian) of the error vector. This is a linear least square problem with a complex error and a real parameter vector. A general version of this problem was solved in Section 4, and the solution is:

$$\theta = \left[\mathrm{Re}\left(A^H W A \right) \right]^{-1} \mathrm{Re}\left(A^H W b \right)$$

5.1.3 Scaling

The Laplace transform coefficients can have a large dynamic range, as can the frequencies raised to high powers. Scaling the Laplace variable reduces the dynamic range. Define:

$$G(s) = H(\xi(s)) = \frac{d_m \xi^m + \cdots + d_0}{\xi^n + c_{n-1}\xi^{n-1} \cdots + c_0}$$

where:

$$\xi(s) = \frac{s}{\omega_{max}}$$

Estimate the coefficients of $H(\xi)$ using the scaled frequencies $\xi(j\omega_i) = j\omega_i / \omega_{max}$ and the same frequency response $\mathcal{G}_i$. The partial derivatives are:

$$\phi_i^T = \left[-\mathcal{G}_i \left(\xi_i^{n-1} \quad \cdots \quad 1 \right) \quad \left(\xi_i^m \quad \cdots \quad 1 \right) \right]$$

The coefficients of $G(s)$ and $H(\xi)$ are related by:

$$b_i = \omega_{max}^{n-i} c_i \quad \text{and} \quad a_i = \omega_{max}^{n-i} b_i$$

5.1.4 Orthogonal Polynomials

Scaling the polynomial is an improvement, but even with scaling the high order terms become almost linearly dependent, and hence high order problem become numerically not well conditioned. It is better therefore to estimate a transfer function defined using a linear combination of orthogonal polynomials:

$$G(s) = H(\xi(s)) = K(p(\xi(s)) = \frac{f_m p_m + \cdots + f_0}{p_n + e_{n-1} p_{n-1} \cdots + e_0}$$

Estimate the coefficients of $K(p)$ using the orthogonal polynomials $p(\xi)$ evaluated at the scaled frequencies $\xi = s / \omega_{max}$. The frequency response $\mathcal{G}_i$ does not change, in other words, the objective is still to match this frequency response. The partial derivatives are:

$$\phi_i^T = \left[-\mathcal{G}_i \left[p_{n-1}(\xi_i) \quad \cdots \quad p_0(\xi_i) \right] \quad \left[p_m(\xi_i) \quad \cdots \quad p_0(\xi_i) \right] \right]$$

Any set of orthogonal polynomials will do, but the set usually recommended is the Chebyshev polynomials. This set of polynomials does a good (optimal?) job of approximating the frequency response. The Chebyshev polynomials are defined by the iteration:

$$p_0(x) = 1$$

$$p_1(x) = x$$

$$p_k(x) = 2x p_{k-1}(x) - p_{k-2}(x)$$

The series continues:

$$p_2(x) = 2x^2 - 1$$

$$p_3(x) = 4x^3 - 3x$$

$$p_4(x) = 8x^4 - 8x^2 + 1$$

$$p_5(x) = 16x^5 - 20x^3 + 5x$$

$$p_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

Use matrix operations to convert between the polynomial coefficients:

$$d_m \xi^m + \cdots + d_0 = f_m p_m + \cdots + f_0$$

where:

$$\begin{bmatrix} d_m \\ \vdots \\ d_0 \end{bmatrix} = T_m \begin{bmatrix} f_m \\ \vdots \\ f_0 \end{bmatrix}$$

For $m = 3$:

$$\begin{bmatrix} d_3 \\ d_2 \\ d_1 \\ d_0 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_3 \\ f_2 \\ f_1 \\ f_0 \end{bmatrix}$$

The transformation matrix is computed from the defining equations using the Matlab commands:

```
m=10;
T=zeros(m,m);
T(m-1:m,m-1:m)=eye(2);
for i=m-2:-1:1
  T(i:m,i)=[2*T(i+1:m,i+1);0]-T(i:m,i+2);
end
```

5.2 Iterative Solution (SK Iteration)

The un-weighted cost function used for the linear problem minimizes the frequency response of just the numerator. As a result the solution tends to match the frequency response data better at frequencies where the numerator is large, which will be at high frequencies. This is not usually where the best match is desired, but for many problems, usually low order problems with small errors, the linear solution will work just fine. Better, however, is to minimize either the absolute or relative error of the frequency response. Either of these objectives can be achieved using a weighted cost function:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i d(s) - c(s) \right|^2$$

where either:

$w_i = \left| d(\omega_i) \right|^{-2}$ to match $\mathcal{G}_i$ at frequencies where $\mathcal{G}_i$ is large (absolute error)

$w_i = \left| c(\omega_i) \right|^{-2}$ to match $\mathcal{G}_i$ equally across the frequency range (relative error)

The weights depend on the estimated transfer function, so they can only be used in iteration. Set the weight based on the previous estimate, and then solve the linear problem again to update the solution. Do this a fixed number of times or until a stopping condition is reached. This iteration is not strictly speaking an optimal solution, but again is often is good enough, and if better is needed this solution is a good starting point for the true nonlinear optimal solution. The same iteration scheme works using the numerator and denominator of $H(\xi)$ and $K(p)$.

5.3 Nonlinear Frequency Domain Estimation

The transfer function is:

$$G(s)=\frac{G_{num}(s)}{G_{den}(s)}=\frac{b_m s^m+\cdots+b_0}{s^n+a_{n-1}s^{n-1}\cdots+a_0}$$

which has the parameter vector:

$$\theta^T=\begin{bmatrix}a_{n-1} & \cdots & a_0 & b_m & \cdots & b_0\end{bmatrix}$$

The objective is to find the parameter vector that minimizes the cost function:

$$J=\sum_{i=1}^{N} w_i\left|\mathcal{G}_i-G(s)\right|^2$$

The error is no longer a linear function of the parameters, and hence this is a nonlinear minimization problem. The partial derivatives are:

$$\frac{\partial G}{\partial a_i}=G_{num}\frac{\partial G_{den}^{-1}}{\partial a_i}=\frac{-1}{G_{den}}Gs^i \quad \text{for} \quad i=0,\ldots,n-1$$

$$\frac{\partial G}{\partial b_i}=\frac{1}{G_{den}}\frac{\partial G_{num}}{\partial b_i}=\frac{1}{G_{den}}s^i \quad \text{for} \quad i=0,\ldots,m$$

Fill up a matrix with these partials:

$$F=\begin{bmatrix}\phi_1^H\\ \vdots \\ \phi_N^H\end{bmatrix}, \quad \phi_i^H=\frac{1}{G_{den}(s_i)}\left[-G(s_i)\begin{pmatrix}s_i^{n-1} & \cdots & 1\end{pmatrix}\ \begin{pmatrix}s_i^m & \cdots & 1\end{pmatrix}\right]$$

The Newton-Raphson update is:

$$\delta\theta=\left(F_R^T W F_R+F_I^T W F_I\right)^{-1}\left(F_R^T W e_R+F_I^T W e_I\right)$$

Setting the weights w_i to unity will minimize the absolute error. Setting the weights to $w_i=\left|\mathcal{G}_i\right|^{-2}$ will minimize the relative error. Start the iteration with the linear solution, or alternatively after several iterations of the weighted linear solution.

The nonlinear solution works similarly with the scaled and orthogonal polynomial versions of the transfer function, and just to be bloody obvious, the scaled transfer function is:

$$H(\xi)=\frac{H_{num}(\xi)}{H_{den}(\xi)}=\frac{d_m\xi^m+\cdots+d_0}{\xi^n+c_{n-1}\xi^{n-1}\cdots+c_0}$$

which has the parameter vector:

$$\theta^T = [c_{n-1} \quad \cdots \quad c_0 \quad d_m \quad \cdots \quad d_0]$$

The cost function is:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i - H(\xi) \right|^2$$

And the vector of partials is:

$$\phi_i^H = \frac{1}{H_{den}(\xi_i)} \left[-H(\xi_i) \left(\xi_i^{n-1} \quad \cdots \quad 1 \right) \quad \left(\xi_i^m \quad \cdots \quad 1 \right) \right]$$

The transfer function with orthogonal polynomials is:

$$K(p) = \frac{K_{num}(p)}{K_{den}(p)} = \frac{f_m p_m + \cdots + f_0}{p_n + e_{n-1} p_{n-1} \cdots + e_0}$$

which has the parameter vector:

$$\theta^T = [e_{n-1} \quad \cdots \quad e_0 \quad f_m \quad \cdots \quad f_0]$$

The cost function is:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i - K(p) \right|^2$$

And the vector of partials is:

$$\phi_i^H = \frac{1}{K_{den}} \left[-K \left(p_{n-1}(\xi_i) \quad \cdots \quad 1 \right) \quad \left(p_m(\xi_i) \quad \cdots \quad 1 \right) \right]$$

5.4 Estimating Delay

The delay can be included in the vector of parameters to be estimated. Change the transfer function to include a delay:

$$G(s) = \frac{G_{num}(s)}{G_{den}(s)} e^{-\tau s} = \frac{b_m s^m + \cdots + b_0}{s^n + a_{n-1} s^{n-1} \cdots + a_0} e^{-\tau s}$$

Include the delay parameter in the vector of parameters:

$$\theta^T = [c_{n-1} \quad \cdots \quad c_0 \quad d_m \quad \cdots \quad d_0 \quad \tau]$$

The cost function does not change:

$$J = \sum_{i=1}^{N} w_i \left| \mathcal{G}_i - G(s) \right|^2$$

The new partial derivatives are:

$$\frac{\partial G}{\partial a_i} = G_{num} e^{-\tau s} \frac{\partial G_{den}^{-1}}{\partial a_i} = \frac{-1}{G_{den}} G s^i \quad \text{for} \quad i = 0, \ldots, n-1$$

$$\frac{\partial G}{\partial b_i} = \frac{e^{-\tau s}}{G_{den}} \frac{\partial G_{num}}{\partial b_i} = \frac{e^{-\tau s}}{G_{den}} s^i \quad \text{for} \quad i = 0,\ldots,m$$

$$\frac{\partial G}{\partial \tau} = \frac{G_{num}}{G_{den}} \frac{\partial e^{-\tau s}}{\partial b_i} = -sG$$

Place these partial derivatives in a vector:

$$\phi_i^H = \frac{1}{G_{den}} \left[-G\left(s_i^{n-1} \quad \cdots \quad 1 \right) \quad e^{-\tau s_i} \left(s_i^m \quad \cdots \quad 1 \quad -s_i G_{num} \right) \right]$$

The rest of the solution is the same. The scaling affects the delay, and so we need to show how the scaled delay enters into the scaled transfer function:

$$H(\xi) = \frac{H_{num}(\xi)}{H_{den}(\xi)} e^{-\tau_h \xi} = \frac{d_m \xi^m + \cdots + d_0}{\xi^n + c_{n-1}\xi^{n-1} \cdots + c_0} e^{-\tau_h \xi}$$

where $\xi = s/\omega_{max}$ and $\tau = \tau_h / \omega_{max}$. The parameter vector is:

$$\theta^T = \left[c_{n-1} \quad \cdots \quad c_0 \quad d_m \quad \cdots \quad d_0 \quad \tau_h \right]$$

and the vector of partials:

$$\phi_i^H = \frac{1}{H_{den}} \left[-H\left(\xi_i^{n-1} \quad \cdots \quad 1 \right) \quad e^{-\tau_h \xi_i} \left(\xi_i^m \quad \cdots \quad 1 \quad -\xi_i H_{num} \right) \right]$$

Use the same scaled delay when the rest of the transfer function is defined using orthogonal polynomials:

$$K(p) = \frac{K_{num}(p)}{K_{den}(p)} e^{-\tau_h \xi} = \frac{f_m p_m + \cdots + f_0}{p_n + e_{n-1} p_{n-1} \cdots + e_0} e^{-\tau_h \xi}$$

The parameter vector is:

$$\theta^T = \left[e_{n-1} \quad \cdots \quad e_0 \quad f_m \quad \cdots \quad f_0 \quad \tau_h \right]$$

And the vector of partials is:

$$\phi_i^H = \frac{1}{K_{den}} \left[-K\left(p_{n-1} \quad \cdots \quad 1 \right) \quad e^{-\tau_h \xi_i} \left(p_m \quad \cdots \quad 1 \quad -\xi_i K_{num} \right) \right]$$

5.5 Estimating the z-Transform

Frequency domain estimation works just as well for z-transforms. Start with the same frequency response estimates $\mathcal{G}_i$, available at frequencies ω_i, say for M points, and define the frequency response index to be $i = 1,\ldots,M$. The transfer function to be identified is the z-transform:

$$G(z) = \frac{c(z)}{d(z)} = \frac{b_m z^m + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0}$$

The z-transform variable is $z = e^{j\omega\Delta}$, where Δ is the sample period. The parameter vector is:

$$\theta^T = \left[a_{n-1} \quad \cdots \quad a_0 \quad b_m \quad \cdots \quad b_0 \right]$$

For the linear version of the estimation problem use the cost function:

$$J = \sum_{i=1}^{M} w_i \left| \mathcal{G}_i d(z_i) - c(z_i) \right|^2$$

The solution is:

$$\theta = \left[\mathrm{Re}\left(A^H W A \right) \right]^{-1} \mathrm{Re}\left(A^H W b \right)$$

where:

$$b = \begin{bmatrix} z_1^n \mathcal{G}_1 \\ \vdots \\ z_M^n \mathcal{G}_M \end{bmatrix}, \quad A = \begin{bmatrix} \phi_1^H \\ \vdots \\ \phi_M^H \end{bmatrix}, \quad \phi_i^H = \left[-\mathcal{G}_i \left(z_i^{n-1} \quad \cdots \quad 1 \right) \quad \left(z_i^m \quad \cdots \quad 1 \right) \right]$$

5.6 Frequency Domain Estimation Examples

5.6.1 Ideal, Low Order Problem

The F-15 Active example in Section 3.7.1 has the closed loop response:

$$\frac{\theta}{\delta_{e_c}} = \frac{-21.9(0.993)}{(1.55)(3.39)(6.32)}$$

The objective is the estimate this Laplace transform using a simulated time response. The following calculations are made:

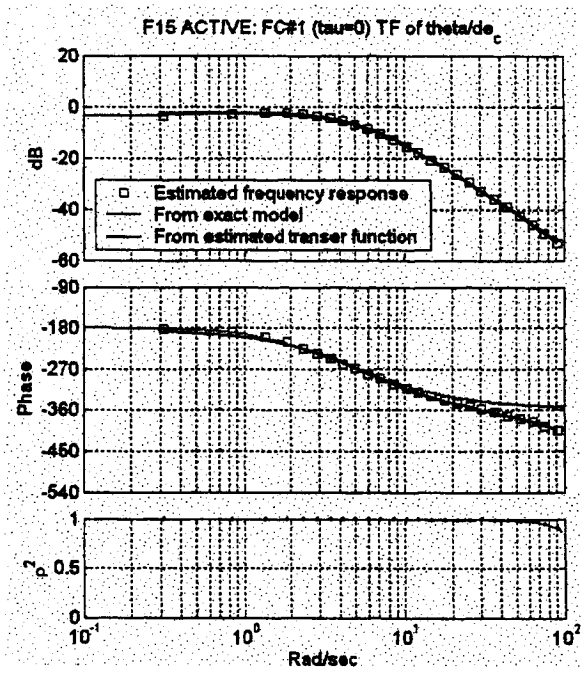
> The closed loop system is simulated using bandpass noise (0.1 to 10 rad/sec, 60 secs with 3 secs zero at start and end, no taper, sample period = 50 msec).
>
> The time series is used to estimate the frequency response (5% taper both sides, no taper, log binning with bin ratio = 1.2, min number bins= 3, keep points with rho2 > 0.5).
>
> The frequency points are used to estimate a Laplace transform (max order = 3, relative difference = 2, use points 0 to 50 rad/sec, do SK iterations, do nonlinear minimization).

The estimated transfer function is:

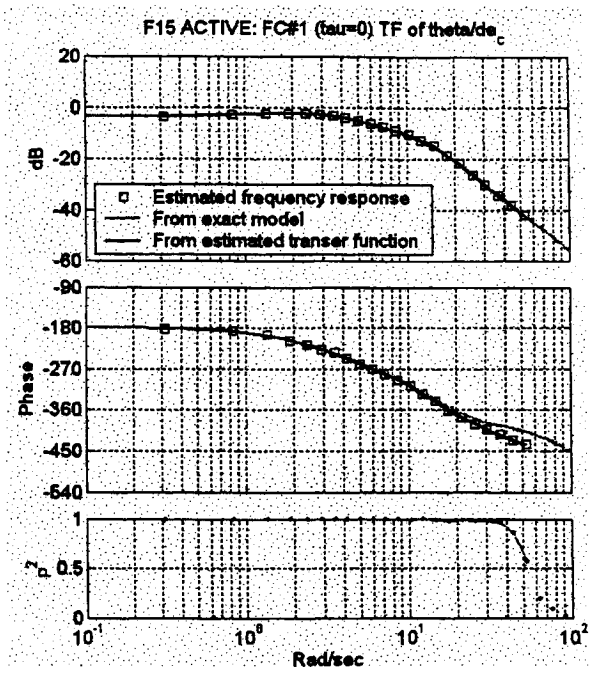
$$\frac{\theta}{\delta_{e_c}} = \frac{7.41(-397)}{[0.896,5.5](131)}$$

Close, but not right on. The process, complicated as it may be, has systematic errors built into it. To exactly match the original model: correct for the FFT end effects when estimating the frequency response, so that the FFTs of the delayed input and output are exact, estimate the coefficients of the z-transform, and then convert to the Laplace domain using the inverse ZOH transformation.

Here and throughout this project the systematic errors are not corrected, but they are small, as shown in Figure 135a. Three frequency responses are compared: The dark solid lines are the exact frequency response using the model, the squares are frequency response estimated from the time series, and the light, dashed lines are the frequency response from the estimated transfer function. The phase error above 10 rad/sec is due to the sampled-data delay, which is in the data, but not the original Laplace transform model.



a) Ideal, low order model



b) Ideal, high order model

Figure 135. Estimated Frequency Responses (F-15 Active Example)

5.6.2 Ideal, High Order Problem

Continue the example from Section 3.7.2, using the F-15 Active model extended to include 5th order longitudinal dynamics, 2nd order actuator and sensor models, and some extra delay in the pitch loop. The exact closed loop transfer function is:

$$\frac{\theta}{\delta_{e_c}} = \frac{-2.19e5(0.00165)(0.0165)(0.992)[0.707,30][0.866,69.3][0.707,300]}{(0.00134)(0.0166)(1.4)(3.73)[0.203,10.9][0.759,59.8](64.1)[0.706,100][0.707,300]}$$

Use the same procedure as in the previous example to estimate this transfer function, the only difference being the simulation sample period is decreased to 5 msec. The estimated 8th order transfer function is:

$$\frac{\theta}{\delta_{e_c}} = \frac{9.31(0.916)[0.505,9.32][0.044,14.7](-119)}{(1.27)[0.934,6.25][0.466,10.8][0.0427,14.7](23.5)}$$

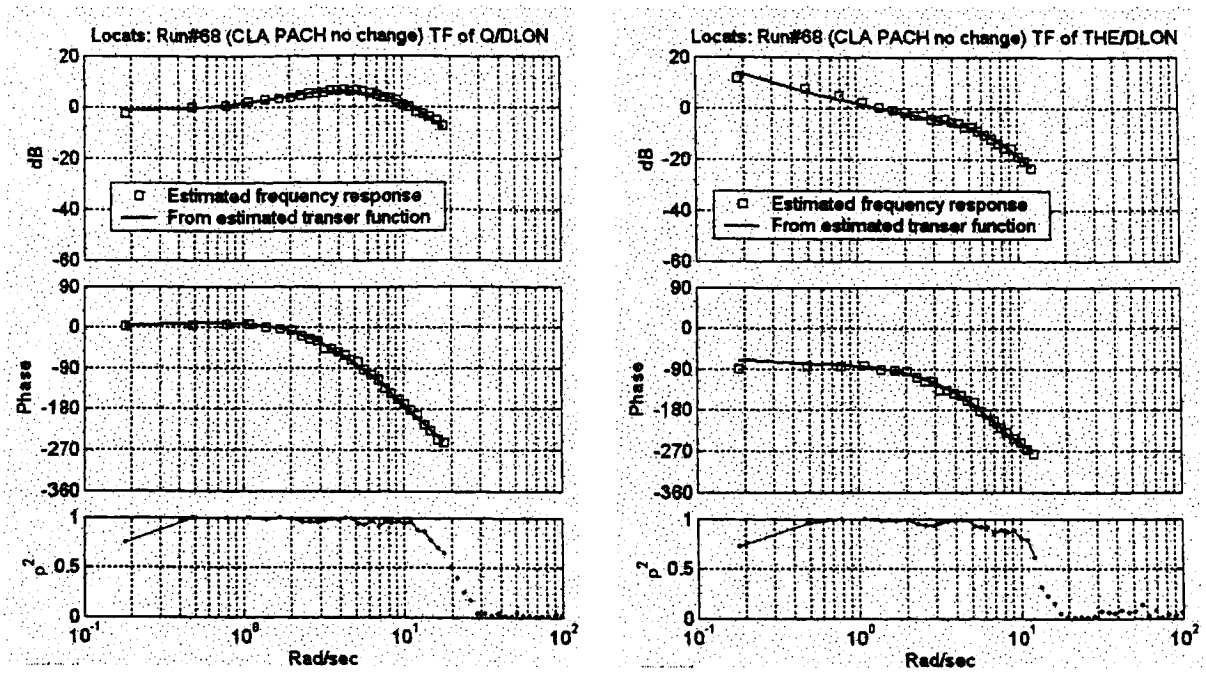
An estimated 3rd order transfer function is:

$$\frac{\theta}{\delta_{e_c}} = \frac{8.48(-126)}{(6.67)[0.922,14.6]}$$

Bode plots of the frequency responses (using the 8th order estimated transfer function) are compared in Figure 135b. Comparing this example to the previous: higher order, same story: good match on the frequency responses, and systematic errors prevent exact match of the poles and zeros.

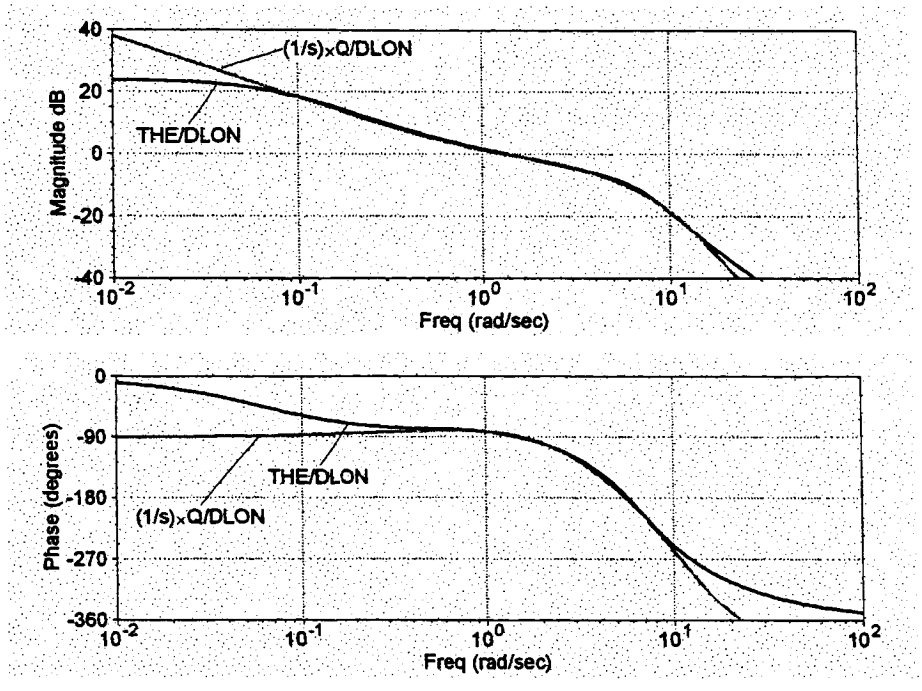
5.6.3 LOCATS Pitch Response From Fixed Base Simulator Data

Use the same time series from Section 3.7.3, using the full 105 seconds to estimate the coefficients of the selected closed loop transfer functions. There is no truth model to use for comparison. The pitch rate and pitch attitude responses from the longitudinal stick are estimated to be:



a) Pitch rate

b) Pitch attitude



c) Comparison

Figure 136. Estimated Frequency Response (LOCATS Run 68, Q/DLON and THE/DLON)

$$\frac{Q}{DLON} = \frac{-95.3(0.245)(1.03)(-16.4)}{(0.267)[0.949,3.99][0.625,10.7]}$$

$$\frac{THE}{DLON} = \frac{-6.99(0.296)(0.846)(-12)}{(0.0618)(0.272)(1.75)[0.548,6.74]}$$

The frequency responses of these transfer functions are compared in Figure 136 with the FFT-derived frequency response estimates (5% taper both sides, no taper, log binning with bin ratio = 1.1, min number bins= 3, keep points with rho2 > 0.5). Comments:

- The transfer function frequency responses in parts a) and b) match the data very well.
- The THE/DLON and integrated Q/DLON frequency responses in part c) are a good match from 0.3 to 10 rad/sec, significantly better than the comparison in Section 5.6.3 based on time domain estimation.
- The poles of THE/DLON and Q/DLON do not match.

6. ESTIMATION USING MARKOV PARAMETERS

System estimation starts with a time series and ends with system parameters. An intermediate step is to compute Markov parameters, and it turns out there are good reasons to do so. The Markov parameters can be used to determine the system order, and then used to compute either transfer function parameters or a state space realization.

6.1 What Are They

The Markov parameters of a digital system are the impulse response.

Transfer Functions:

Define the z-transform:

$$G(z)=\frac{b_n z^n+\cdots+b_0}{z^n+a_{n-1}z^{n-1}\cdots+a_0}$$

By long division the z-transform can also be written:

$$G(z)=h_0+h_1z^{-1}+h_2z^{-2}+\cdots$$

The h_i's are the Markov parameters. The Markov parameters are used for convolution:

$$y_k=\sum_{i=0}^{k}h_{k-i}u_i$$

State Space:

The state space realization of a digital system is:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

The output is:

$$y_k=CA^kx_0+\sum_{i=0}^{k-1}CA^{k-1-i}Bu_i+Du_k$$

The Markov parameters are:

$$h_k=\begin{cases} D & \text{if } k=0 \\ CA^{k-1}B & \text{if } k>0 \end{cases}$$

For multivariable systems the Markov parameters are matrices.

6.2 Determining System Order

The Markov parameters can be used to determine the system order. The best way to show this is using the state space realization. First define some terms. The observability matrix, controllability matrix, and Hankel matrix are respectively:

$$\mathcal{O}=\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\ ,\ \mathcal{P}=\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}\ ,\ M(i,j)=\begin{bmatrix} h_i & h_{i+1} & \cdots & h_{i+j} \\ h_{i+1} & h_{i+2} & \cdots & h_{i+j+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{i+j} & h_{i+j+1} & \cdots & h_{i+2j} \end{bmatrix}$$

The product $\mathcal{OP}$ is the Hankel matrix:

$$M(1,n-1)=\mathcal{OP}=\begin{bmatrix} CB & CAB & \cdots & CA^{n-1}B \\ CAB & CA^2B & \ddots & CA^nB \\ \vdots & \ddots & \ddots & \vdots \\ CA^{n-1}B & CA^nB & \cdots & CA^{2n-2}B \end{bmatrix}=\begin{bmatrix} h_1 & h_2 & \cdots & h_n \\ h_2 & h_3 & \ddots & h_{n+1} \\ \vdots & \ddots & \ddots & \vdots \\ h_n & h_{n+1} & \cdots & h_{2n-1} \end{bmatrix}$$

The rank of M cannot be more than the rank of the parts $\mathcal{O}$ and $\mathcal{P}$, and in fact equals the minimum rank of parts $\mathcal{O}$ and $\mathcal{P}$, which is the order of the system. Furthermore, if more Markov parameters are added, this will not increase the rank, which follows from the Cayley-Hamilton theorem, which states that the matrix product A^k for $k \geq n$ is a linear combination of the matrix products less than n. In other words, adding higher powers of A to parts $\mathcal{O}$ and $\mathcal{P}$ will not increase the ranks of $\mathcal{O}$ and $\mathcal{P}$, and hence will not increase the rank of M. It therefore follows that:

$$\text{System order } n = \text{Rank}[M(i,j)] \text{ if } j \geq n-1$$

The condition $j \geq n-1$ is needed because the Hankel matrix has to be at least size n in order to have rank n. In practice make the Hankel matrix much larger than the expected system order, and then compute the rank to find the actual system order. The best numerical way to determine the rank is by the singular value decomposition (SVD). The rank is the number of nonzero singular values. Due to noise in the measurements and finite precision in the calculations the small singular values will not be exactly zero, and a threshold needs to be established below which the singular values are considered to be zero.

6.3 From Markov Parameters to Transfer Functions

Estimating the Markov parameters from a time series is difficult; we will treat that problem shortly, and first show the easy, or at least easier, step of converting the Markov parameters to the parameters of a z-transform. The Markov parameters can be computed in terms of the z-transform parameters by equating like powers of the polynomial equation:

$$\left(1+a_{n-1}z^{-1}+\cdots+a_0z^{-n}\right)\left(h_0+h_1z^{-1}+h_2z^{-2}+\cdots\right)=\left(b_n+b_{n-1}z^{-1}+\cdots b_0z^{-n}\right)$$

It follows that:

$$\left[\begin{array}{cccc|c} & & & h_0 & \\ & 0 & h_0 & h_1 & \\ & \iddots & \iddots & \vdots & \\ & h_0 & \iddots & h_{n-2} & h_{n-1} \\ h_0 & h_1 & \cdots & h_{n-1} & h_n \\ \hline h_1 & h_2 & \cdots & h_n & h_{n+1} \\ h_2 & h_3 & \iddots & h_{n+1} & h_{n+2} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ h_n & h_{n+1} & \iddots & h_{2n-1} & h_{2n} \end{array}\right]\left[\begin{array}{c} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ 1 \end{array}\right]=\left[\begin{array}{c} b_n \\ b_{n-2} \\ \vdots \\ b_1 \\ b_0 \\ 0 \\ \vdots \\ 0 \end{array}\right]$$

Use the bottom part of the large matrix to compute the a coefficients:

$$\left[\begin{array}{ccc} h_1 & \cdots & h_n \\ \vdots & \iddots & \vdots \\ h_n & \cdots & h_{2n-1} \end{array}\right]\left[\begin{array}{c} a_0 \\ \vdots \\ a_{n-1} \end{array}\right]=-\left[\begin{array}{c} h_{n+1} \\ \vdots \\ h_{2n} \end{array}\right]$$

And use the top part of the larger matrix to compute the b coefficients:

$$\left[\begin{array}{c} b_n \\ \vdots \\ b_0 \end{array}\right]=\left[\begin{array}{ccc} 0 & \cdots & h_0 \\ \vdots & \iddots & \vdots \\ h_0 & \cdots & h_n \end{array}\right]\left[\begin{array}{c} a_0 \\ \vdots \\ 1 \end{array}\right]$$

The first several Markov parameters may be zero, which is okay. Say the first nonzero Markov parameter is h_i, and then the first nonzero b coefficient is b_{n-i}.

When computing the a coefficients the Markov parameter need not start at h_1, and the expression generalizes to:

$$M(i,n-1)\left[\begin{array}{c} a_0 \\ \vdots \\ a_{n-1} \end{array}\right]=\left[\begin{array}{c} h_{n+i} \\ \vdots \\ h_{2n+i-1} \end{array}\right]$$

In fact we can go further and use a larger Hankel matrix to say:

$$M(i,p-1)\left[\begin{array}{c} a_0 \\ \vdots \\ a_{n-1} \\ 1 \\ 0 \\ \vdots \end{array}\right]=0$$

where $p \geq n+1$. The a coefficients can be computed using a basis for the null space of $p \times p$ matrix $M(i,p-1)$. We will now show how to do this, the advantage being that the same SVD as used to determine the order of the system can be used to create the basis for the null space. Define the SVD as:

$$M(i,p-1)=\begin{bmatrix}U_1 & U_2\end{bmatrix}\begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\ V_2^T\end{bmatrix}$$

where Σ_1 is an $n\times n$ diagonal matrix containing the nonzero singular values, and the block matrices have compatible dimensions. (The blocks $U_1=V_1$ and $U_2=V_2$ are equal in the SISO case because in that case the Hankel matrix is symmetric. This symmetry does not hold for MIMO systems, and so we keep the block names different). The $n\times p$ matrix V_2 is an orthonormal basis for the nullspace, and hence there exists a $p\times 1$ vector α such that:

$$V_2\alpha=\begin{bmatrix}V_{21}\\ V_{22}\end{bmatrix}\alpha=\begin{bmatrix}a_0\\ \vdots\\ a_{n-1}\\ 1\\ 0\\ \vdots\end{bmatrix}\begin{matrix}\}n\times 1\\ \\ \\ \}p\times 1\end{matrix}$$

where V_{21} is $n\times p$ and V_{22} is $p\times p$. Use the bottom part of the vector to solve for α, and hence:

$$\begin{bmatrix}a_0\\ \vdots\\ a_{n-1}\end{bmatrix}=V_{21}V_{22}^{-1}\begin{bmatrix}1\\ 0\\ \vdots\end{bmatrix}$$

6.4 The Eigensystem Realization Algorithm

Hankel matrices can be used to create a state space realization using the Eigensystem Realization Algorithm (ERA). Compute the SVD of the Hankel matrix:

$$M(1,p-1)=\begin{bmatrix}U_1 & U_2\end{bmatrix}\begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\ V_2^T\end{bmatrix}=\left(U_1\Sigma_1^{1/2}\right)\left(\Sigma_1^{1/2}V_1^T\right)$$

where Σ_1 is $n\times n$ and $p\geq n$. The terms with $\Sigma_1^{1/2}$ are respectively observability and controllability matrices, and for this reason it follows that the next Hankel matrix satisfies:

$$M(2,p-1)=\left(U_1\Sigma_1^{1/2}\right)A\left(\Sigma_1^{1/2}V_1^T\right)$$

This equation can be used to compute the A matrix of the state space realization:

$$A=\left(\Sigma_1^{-1/2}U_1^T\right)M(2,p-1)\left(V_1\Sigma_1^{-1/2}\right)$$

Again because they are "ibility" matrices the C matrix is the first row of the matrix $U_1\Sigma_1^{1/2}$, and the B matrix is the first column of $\Sigma_1^{1/2}V_1^T$ (or the first set of rows and columns in the MIMO case). Define $E_1=\begin{pmatrix}1 & 0 & \cdots\end{pmatrix}$ and then:

$$B=\Sigma_1^{1/2}V_1^TE_1^T\quad,\quad C=E_1U_1\Sigma_1^{1/2}$$

And finally, the D matrix is equal to the first Markov parameter:

$$D = h_0$$

6.5 Estimating the Markov Parameters

Given the time series: u_i and y_i, each defined for $i = 0,\ldots,N-1$ and connected by the convolution:

$$y_k = \sum_{i=0}^{k} h_{k-i} u_i$$

The Markov parameters can be determined from the matrix equation:

$$\begin{bmatrix} u_0 & 0 & 0 & 0 \\ u_1 & u_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ u_{N-1} & \cdots & u_1 & u_0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix}$$

Comments:

- The matrix is invertible and therefore a solution exists if u_0 is nonzero. The time series should be shifted until u_0 is nonzero. The matrix does not actually have to inverted, and the h_i's can be computed recursively.
- The Markov parameters are very sensitive to the data. The parameter h_0 depends only on u_0 and y_0, and so on, where in general h_i depends only on data up to that time.
- A transfer function or state space system of order n only needs Markov parameters up to $2n$, and hence only data up to that time is used. Say the order is not known, and Markov parameters are used up to $2p$, where $p > n$, which means more data is used, but still not beyond $2p$. Whatever information is in the data beyond that time is not used. Presumably this data could be used to help reduce sensitivity to noise.

If the data is from a simulation then the simulation must start with zero initial conditions.

This straightforward method to compute the Markov parameters is theoretically valid but has significant practical problems. It is a good starting point for improvement, which we will do shortly, after first showing conditions where two different estimation methods give the same transfer function parameters.

6.6 Relationship Between LLS Parameter Estimation and Markov Parameter Estimation

Given the time series: u_i and y_i, each defined for $i = 0,\ldots,N-1$, the objective is to estimate the coefficients of the transfer function:

$$G(z) = \frac{b_n z^n + \cdots + b_0}{z^n + a_{n-1} z^{n-1} \cdots + a_0}$$

Here we show that two different estimation methods give the same answer.

Method 1:

The LLS estimation problem using the time series can be used estimate the transfer function parameters:

$$\begin{bmatrix} 0 & \cdots & 0 & u_0 & 0 & \cdots & 0 \\ -y_0 & \ddots & \vdots & u_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ -y_{n-1} & \ddots & -y_0 & u_n & \ddots & \ddots & u_0 \\ -y_n & \ddots & -y_1 & u_{n+1} & \ddots & \ddots & u_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ -y_{2n-1} & \cdots & -y_n & u_{2n} & \cdots & u_{n+1} & u_n \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_n \\ \vdots \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{2n} \end{bmatrix} \qquad (1)$$

The minimum length time series with $N = 2n+1$ is used to estimate the $2n+1$ parameters.

Method 2:

An alternative method is to first compute the Markov parameters:

$$\begin{bmatrix} u_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ u_n & \ddots & u_0 & \ddots & \ddots & \vdots \\ u_{n+1} & \ddots & u_1 & u_0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ u_{2n} & \cdots & u_n & \cdots & u_1 & u_0 \end{bmatrix} \begin{bmatrix} h_0 \\ \vdots \\ h_n \\ h_{n+1} \\ \vdots \\ h_{2n} \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{2n} \end{bmatrix} \qquad (2)$$

where again we have restricted attention to the first $N = 2n+1$ parameters, and then from these Markov parameters the transfer function coefficients are computed:

$$\begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ -h_0 & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ -h_{n-1} & \ddots & -h_0 & 0 & \ddots & \ddots & 1 \\ -h_n & \ddots & -h_1 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ -h_{2n-1} & \cdots & -h_n & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ \vdots \\ a_0 \\ b_n \\ \vdots \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} h_0 \\ \vdots \\ \vdots \\ h_n \\ h_{n+1} \\ \vdots \\ h_{2n} \end{bmatrix} \qquad (3)$$

Comparison:

These two methods should get to the same answer. Equation (2) can be extended to:

$$\begin{bmatrix} u_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ u_n & \ddots & u_0 & \ddots & \ddots & \vdots \\ u_{n+1} & \ddots & u_1 & u_0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ u_{2n} & \cdots & u_n & \cdots & u_1 & u_0 \end{bmatrix} \begin{bmatrix} h_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ h_n & \ddots & h_0 \\ h_{n+1} & \ddots & \vdots \\ \vdots & \ddots & h_n \\ h_{2n} & \cdots & h_{n+1} \end{bmatrix} = \begin{bmatrix} y_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ y_n & \ddots & y_0 \\ y_{n+1} & \ddots & \vdots \\ \vdots & \ddots & y_n \\ y_{2n} & \cdots & y_{n+1} \end{bmatrix} \qquad (4)$$

which follows due to the triangular nature of the input matrix. Multiply (3) on the left by the input matrix, apply the identities established in (4), and the result is (1). This establishes that both methods give the same transfer function parameters.

APPENDIX C – WAVELET-BASED ANALYISIS OF ROLL RATCHET USING A FLIGHT TEST DATABASE

1. OVERVIEW

Degraded lateral axis handling qualities have at times resulted in sustained high frequency, small amplitude pilot-vehicle system oscillations or roll ratcheting. The degraded handling qualities can result from a combination of high roll damping, high levels of roll control sensitivity, and poor feel system dynamics. In some cases the roll ratchet phenomenon has been considered more of a nuisance that primarily affects ride quality, while in the more severe cases it significantly impacts aircraft handling qualities. Past ground based simulation research has indicated a peaking of the human pilot neuromuscular mode around the ratchet frequency. This link between the dynamics of the human pilot and the roll ratchet phenomenon was further explored using the flight research database from a lateral handling qualities study conducted with the USAF variable stability NT-33A. In this analysis Fast Fourier Transform techniques (FFT) were used to compute pilot and pilot-vehicle system describing functions. Strong evidence was generated to support the theory that the pilot neuromuscular mode couples with the feel system dynamics to produce an effective limb-manipulator mode. However, the smoothing of the frequency domain data over a given time window that results from the FFT analysis may have obscured the pilot-vehicle dynamic coupling that was changing over time. In this appendix the WERA system identification technique is used to reevaluate a subset of the same flight test database and to identify key pilot parameters that vary in time. These new results appear to indicate that: a lightly-damped mode in the open-loop system is introduced by the human pilot at frequencies higher than the task bandwidth; a direct correspondence exists between the peak amplitude of the lightly-damped mode and the occurrence of roll ratchet; and the wavelet-based identification technique is a powerful tool for assessing pilot-vehicle system performance over time intervals that were previously not feasible with FFT-based analysis.

2. INTRODUCTION

2.1 Feel Systems and Handling Qualities

Aircraft cockpit controls are either force or position sensing. As shown in Figure 137a the feel system is in series for position sensing. Position sensing does not include the feel system dynamics, while measurements made in terms of force do include these dynamics. For the force sensing feel system shown in Figure 137b the feel system dynamics are in parallel. Thus neither force or position measurements include the feel system dynamics, since they are "downstream" of this point. Although new aircraft continue to be designed with both types of controls, the force sensing feel system generates less lag during closed-loop piloted control (Ref. 48). Examples of force sensing cockpit feel systems include the side stick controllers found on the F-16 and F/A-22 fighters, while position sensing examples include the center sticks found on the F/A-18 Hornet and Super Hornet fighters and the C-17 military transport as well as the center control yolk found on the Boeing 777 commercial transport.

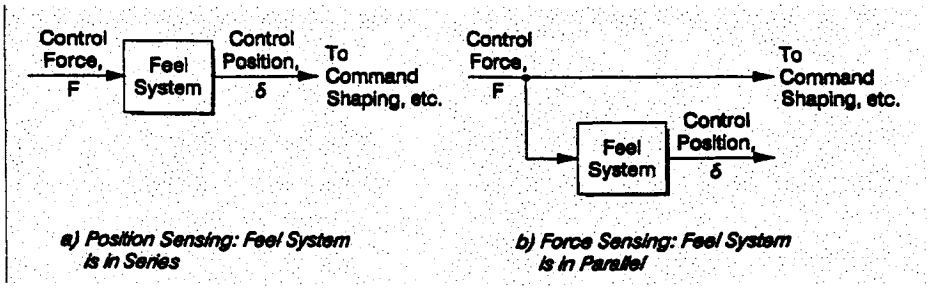


Figure 137. Mechanization of Feel Systems

Within the handling qualities community a debate continues as to the importance of cockpit feel systems in a pilot's assessments of aircraft handling qualities. As detailed in Ref. 49 much of the debate centers on how to best account for the effect of the feel system dynamics, if any, when applying handling qualities criteria. Although it has generally become accepted practice to include the lags associated with control surface actuators when modeling aircraft dynamics for handling qualities assessments, no similar agreement has been found regarding the feel system. By reviewing the existing flight test databases involving the evaluation of cockpit feel systems, it is concluded in Ref. 49 that the feel system dynamics do indeed impact handling qualities. Unfortunately no comprehensive flight test database yet exists to effectively set requirements for the many types of feel systems.

2.2 Limb-Manipulator Dynamics

In Ref. 48 human neuromuscular actuation is described, albeit in what the reference calls a greatly oversimplified manner, as a two loop system. The inner loop involves short pathways from the Golgi and muscle spindle receptors directly to the spinal level and back again to the musculature. The time lags associated with information flow are low because of these short neural pathways. Thus the effective bandwidth of this loop that is primarily sensitive to forces can be quite high. The outer loop features joint and other receptors (e.g., peripheral vision) as the major feedback elements. Here the pathways and associated delays are longer and thus result in a lower outer loop bandwidth. With a force sensing stick there is little or no joint movement, so the higher bandwidth inner loop should dominate. With a position sensing stick, on the other hand, the joint receptors are key elements, so the lower bandwidth outer loop should dominate. The higher bandwidth closure achieved with a force sensing feel system thus produces less lag.

There are several schools of thought regarding whether or not the feel system lag is altered by pilot loop closures around the stick itself. As outlined in Ref. 49 and detailed in Ref. 50, strong evidence now exists through analysis of a flight test database that the pilot couples with the feel system dynamics to form a combined limb-manipulator second order mode. The analysis techniques employed in Ref. 50 included extensive use of Fast Fourier Transform techniques (FFT) to compute pilot and pilot-vehicle system describing functions. The describing functions for runs in which roll ratchet was noted by the pilots featured significantly greater peaking at the limb-manipulator mode frequency, when compared with runs where no ratchet was noted or observed. The describing function assumes time-stationary pilot dynamics, and is a measurement of the mean or average pilot behavior for the analyzed run time. In Ref. 51 roll ratchet is described as a time-varying, non-stationary phenomenon that would be expected to be related to pilot gain and compensation variations throughout the run. It is thus the objective of this appendix to reevaluate a subset of the flight data analyzed in Ref. 50 using wavelet-based techniques from which the time-varying nature of key pilot-vehicle system parameters, including the limb-manipulator mode, can be examined.

3. PILOT-VEHICLE SYSTEM IDENTIFICATION USING WAVELETS

3.1 Discrete Wavelet Transforms

A technique for extracting the impulse response characteristics of a linear, time-varying system is described in Section 5 of this report. It is shown that the discrete wavelet transform (DWT) can be used in conjunction with state-space methods to estimate the impulse response function of a system over short intervals of time. The order of the system is assumed, an estimate of the free response is subtracted from the observed total response, and the residual forced response is used in the convolution integral to compute the impulse response using wavelet analysis. The Fourier transform of the impulse response is then computed to produce a bode plot of the observed system, and a least-squares fit of the model to the observed bode plot (by iterating on the model poles and zeros) is performed using a non-linear regression method. The assumed model for describing the pilot in this paper is,

$$\frac{\phi_e(s)}{\delta(s)} = K_p \frac{(s+z)}{(s^2+2\zeta\omega_n s+\omega_n{}^2)} e^{-\tau_e s}$$

where ϕ_e is the roll error signal, δ is the pilot stick position, K_p is the root-locus gain, z is the lead (zero), τ_e is the effective pure time delay of the pilot, and ω_n and ζ are the natural frequency and damping of the pilot neuromuscular mode.

For identification precision it is necessary to shift one of the two signals (ϕ_e or δ) in time by the amount of pure time delay introduced by the pilot, τ_e. Figure 138 depicts the method used for computing τ_e, where both signals are filtered, and the filtered error signal is fed through the assumed pilot model (without τ_e). The resulting output is then scaled to be of similar magnitude with the filtered stick output of the pilot. Using a given time window (e.g., 3.7 seconds for the analysis presented in this paper) the sum of the squared differences between the two signals is computed. The stick signal is then shifted in time (the error signal remains fixed) and the computation repeated. The time delay estimate τ_e is the shift corresponding to the minimum squared difference summation. Figure 139 compares the estimated τ_e to a simulated time delay that varies sinusoidally in time.

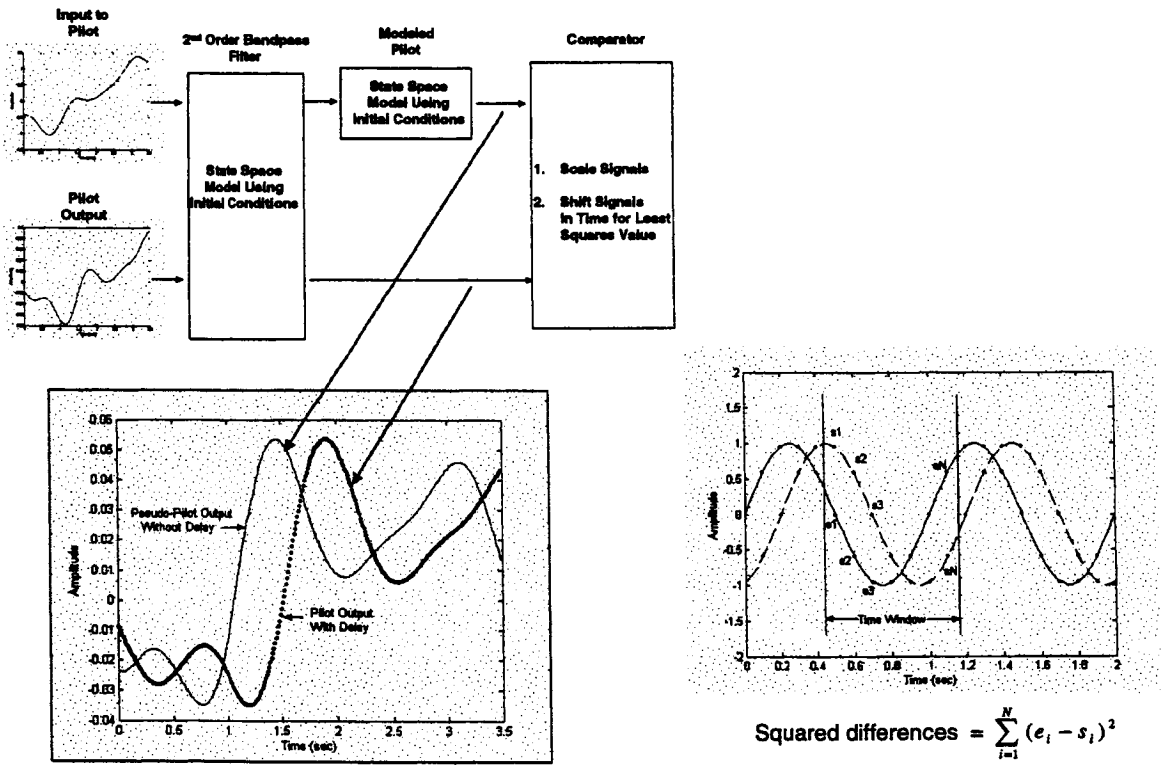


Figure 138. Time Delay Computation Method

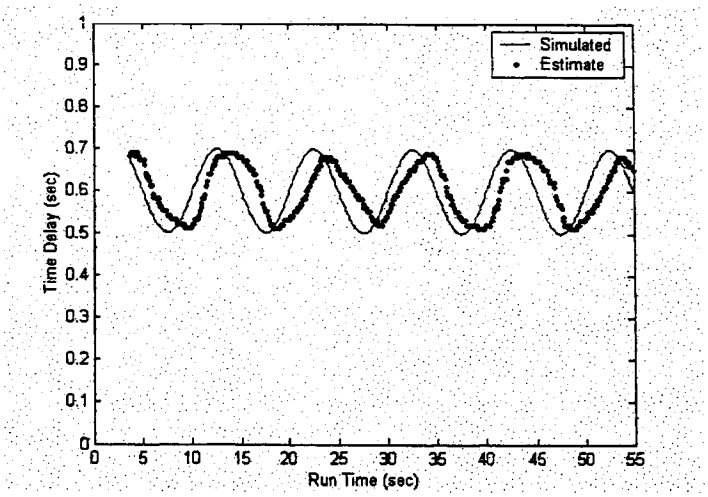


Figure 139. Least Squares Estimation of Simulated Sinusoidal Time Delay

3.2 Time Domain Wavelet-Based System Identification

As a demonstration of the identification techniques described, a Matlab simulation was exercised using a sum-of-sines commanded roll signal and aircraft lateral dynamics closely corresponding to those flown in the Ref. 51 flight research program. A step decrease in pilot damping from 0.25 to 0.05 occurred at 5 seconds, followed by a step increase back to 0.25 at 15 seconds. Figure 140a, b, and c show that all the pilot parameter estimates closely tracked the actual values in time, so that the crossover frequency (ω_c), phase margin, and gain margin were generally well-estimated. In Figure 140d the simulated stick output is compared with the peak magnitude of the limb-manipulator mode, and it is seen that while the estimated limb-manipulator peak lags the actual value, it is essentially synchronous in rise and fall with the onset and departure of roll ratchet. Numerous simulations confirmed that roll ratchet is present if the limb-manipulator peak approaches or exceeds 0 dB.

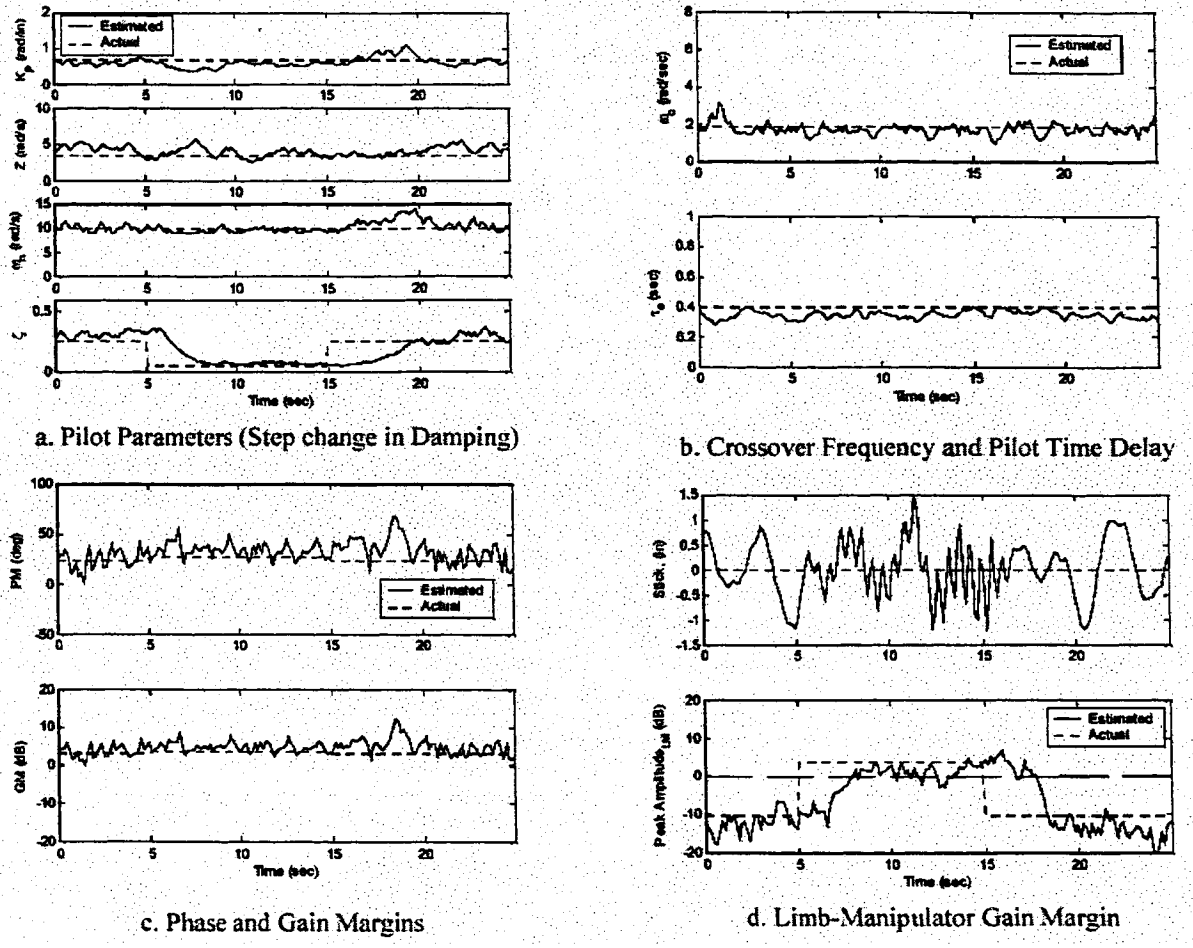
In Figure 141 the bode plots corresponding to the pilot-vehicle system open-loop transfer function estimates are superimposed, showing the spread of magnitude and phase over the full run-time. The limb-manipulator peaking that occurs in the vicinity of 10 rad/sec is depicted over the course of time in the right panel of Figure 141, where the region of 10 rad/s is magnified. Note in Figure 140d that from 7 to 17 seconds the limb-manipulator peak rises above the 0 dB plane, corresponding to the occurrence of ratchet

4. FLIGHT TEST DATA OVERVIEW

4.1 Flight Research Program Description

Under contract to the U.S. Air Force and supported by NASA Dryden Flight Research Center, an in-flight lateral handling qualities research program was conducted by Calspan Corporation (now Veridian) in 1987. As documented in Ref. 51 the program used the variable stability NT-33A aircraft to investigate lateral handling qualities issues including stick feel system dynamics, gradient, and sensing type (i.e., position or force) in combination with roll mode time constant variations. The evaluation tasks were sum-of-sines (SOS) and discrete head-up display (HUD) tracking, air-to-air tracking, and power approach and landing. Following the initial work of Ref. 52, one specific objective of the program was to further investigate the effects of time delays and feel system delays on handling qualities. In addition to the published results of Ref. 51, an independent assessment of the resulting flight test database was carried out by Systems Technology, Inc. (STI) and reported in Ref. 50. This work featured the describing

function analysis of the pilot-vehicle system including the limb-manipulator mode as discussed in the Section 2 of this appendix.



a. Pilot Parameters (Step change in Damping)

b. Crossover Frequency and Pilot Time Delay

c. Phase and Gain Margins

d. Limb-Manipulator Gain Margin

Figure 140. Pilot-Vehicle Parameter Identification using the DWT (simulated)

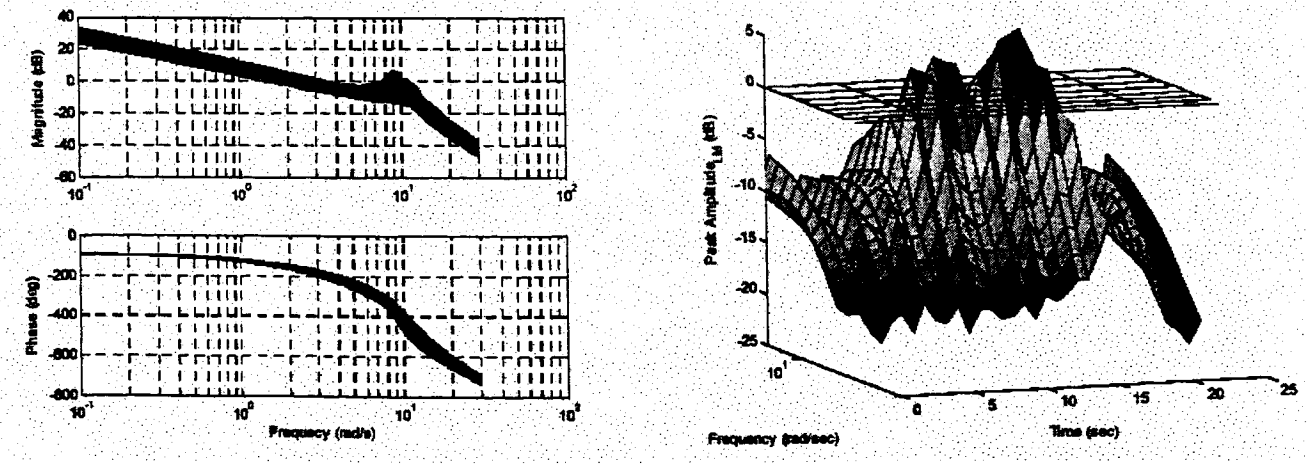


Figure 141. Pilot-Vehicle System Open-Loop Bode Plot and Corresponding Limb-Manipulator Peak Magnitude

4.2 Sum-of-Sines Tracking Task

The describing function analysis undertaken in the Ref. 50 independent review of the Ref. 51 flight test data focused on the SOS tracking task. A block diagram description of this closed-loop, compensatory tracking task is shown in Figure 142. In the figure the pilot is represented by the Y_p block, while the aircraft or controlled element is represented by the Y_c block. The objective of the task is for the pilot to minimize the error ($\phi_e = \phi_c - \phi$) displayed on the HUD within defined tolerances. Data from this type of tracking task where the input is known can be used to identify the dynamics of the pilot (Y_p), controlled element (Y_c), open-loop system (Y_pY_c), and closed-loop system [$Y_pY_c/(1+Y_pY_c)$].

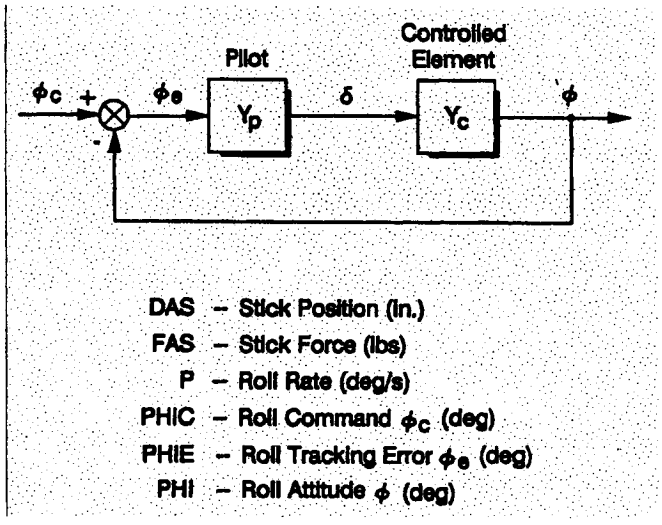


Figure 142. HUD Sum-of-Sines Tracking Task

An example SOS bank angle command (ϕ_c) signal is shown in Figure 143, and the corresponding power spectra density plot is given in Figure 144. The SOS signal is made up of 13 frequencies that are essentially evenly spaced on the log-frequency axis of Figure 144. In general, far fewer frequencies are required to ensure that the task appears random to the pilot. The range of frequencies found in the signal (i.e., approximately 0.4 to 22 rad/sec) adequately extends from the region of piloted control beyond limb-manipulator mode interaction. Note that the four lowest frequencies all have essentially the same input power. The power then drops off at an approximately -20 dB per decade slope, which prevents the higher frequencies from dominating the task.

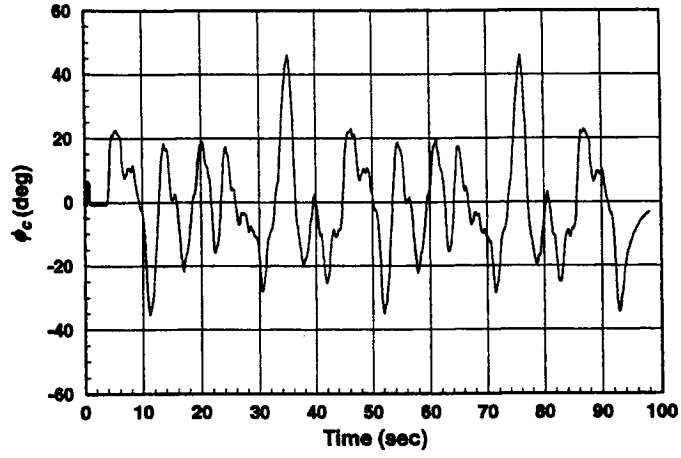


Figure 143. Example Sum-of-Sines Bank Angle Command (ϕ_c)

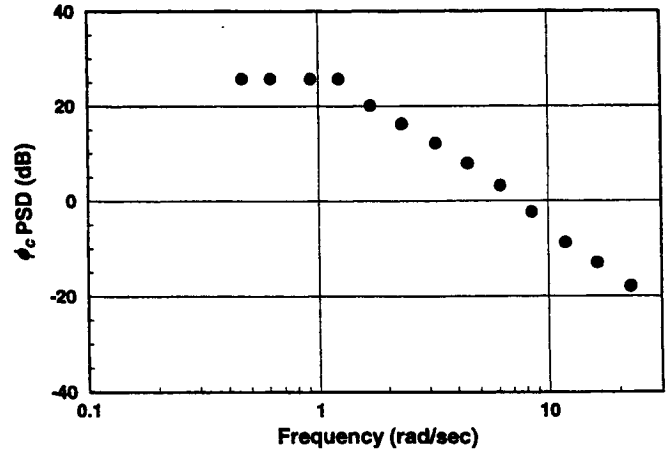


Figure 144. Power Spectra Density for Example Sum-of-Sines Bank Angle Command (ϕ_c)

4.3 Selected Runs

Configurations flown with the variable stability NT-33A featured both force and position sensing feel systems with fast, slow, and degraded dynamics as defined by the feel system natural frequency (26, 13, and 8 rad/sec, respectively). The aircraft lateral dynamics featured roll mode time constants of 0.15 sec (highest roll damping), 0.25 sec, and 0.40 sec (lowest roll damping). Some configurations also included added command path time delays of either 55 or 110 msec.

Four SOS tracking task runs from the Ref. 51 flight research database were selected for the WERA-based analysis described in this appendix. All four runs were flown by the same pilot with the same roll control sensitivity (18 deg/sec/lbs). As identified in Table 37 the first selected case (Run 4069.1) featured a slow feel system (13 rad/sec) and the highest roll mode time constant. This configuration received a Level 1 handling qualities rating for the tracking task evaluation with no roll ratchet or PIO tendencies. The remaining three configurations listed in the table all had significant handling qualities deficiencies. The Run 4069.4 configuration had the degraded feel system combined with the highest roll damping, and resulted in continuous roll ratchet and a Level 3 handling qualities assessment. Next, the Run 4161.3 configuration had the fast feel system, a 55 msec command path time delay, and medium roll damping. This run featured intermittent roll ratchet and a Level 3 handling qualities rating. Finally, Run 4176.1

featured the same configuration as 4161.3 except with no command path time delay. This run resulted in Level 3 handling qualities with continuous lower frequency ratchet.

Table 37. System Characteristics and Flight Results for Selected Runs

	Command Path Dynamics		Roll Mode	Flight Results	
Run No.	ω_{FS} (rad/sec)	τ (msec)	T_r (sec)	Description	HQR
4069.1	13	None	0.40	No sustained ratchet	2
4069.4	8	None	0.15	Continuous ratchet	8
4161.3	26	55	0.25	Intermittent ratchet	7
4176.1	26	None	0.25	Lower frequency ratchet	7

Stick force time histories for the four runs identified in Table 37 are shown in Figure 145. In general, ratchet can essentially be identified by simple inspection of the time histories. For example, no sustained ratchet is evident in the Level 1 configuration stick response of Figure 145a, however, the other configurations clearly display the sustained high frequency oscillations. Note also that the intermittent nature of the ratchet in Figure 145c and the lower frequency ratchet in Figure 145d. Time histories showing the set of signals identified in Figure 142 is presented in Figure 146 for the intermittent ratchet example (Run 4161.3).

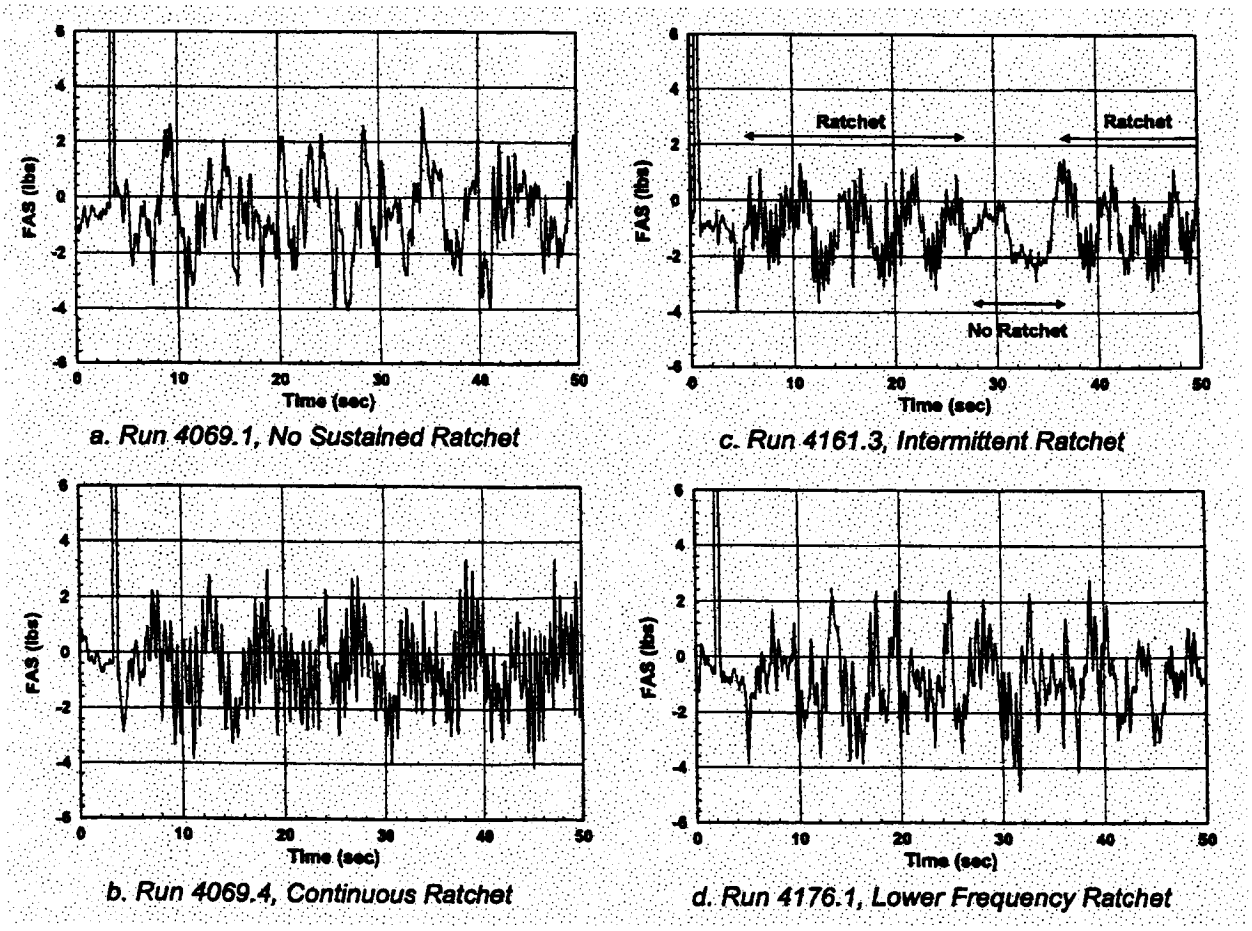


Figure 145. Stick Force Time Histories for Selected Analysis Runs

5. WERA ANALYSIS OF TRACKING TASK FLIGHT TEST DATA

The wavelet identification method was initially applied to flight data from the intermittent ratchet case (4161.3). Pilot parameters were estimated using roll attitude error (ϕ_e) and pilot stick output (δ). Figure 147a shows the reconstructed pilot stick output that is generated using the pilot parameter estimates and the roll attitude error – both time series share very similar shapes and ratcheting. Figure 147b shows the limb-manipulator frequency roughly corresponding to the ratchet frequency of 12.5 rad/s. In Figure 147b pilot damping is seen to inversely affect ratchet – although this is not as good an indicator of ratchet as the limb-manipulator peak amplitude, shown later in Figure 148a.

Time variations in gain margin, phase margin, pilot time delay, and crossover frequency (Figure 147c and d) are much larger than those observed in the simulated case, indicating that time variations in these parameters are actually occurring. In Figure 148 there is excellent agreement between ratchet and limb-manipulator peak as it increases toward 0 dB. From the data, a threshold appears to exist at approximately -2dB, above which ratchet occurs. Even for the case where the pilot did not report ratchet (Figure 148d), the pilot stick time history shows short intervals of ratchet-like behavior during which the estimated limb-manipulator peak approached 0 dB.

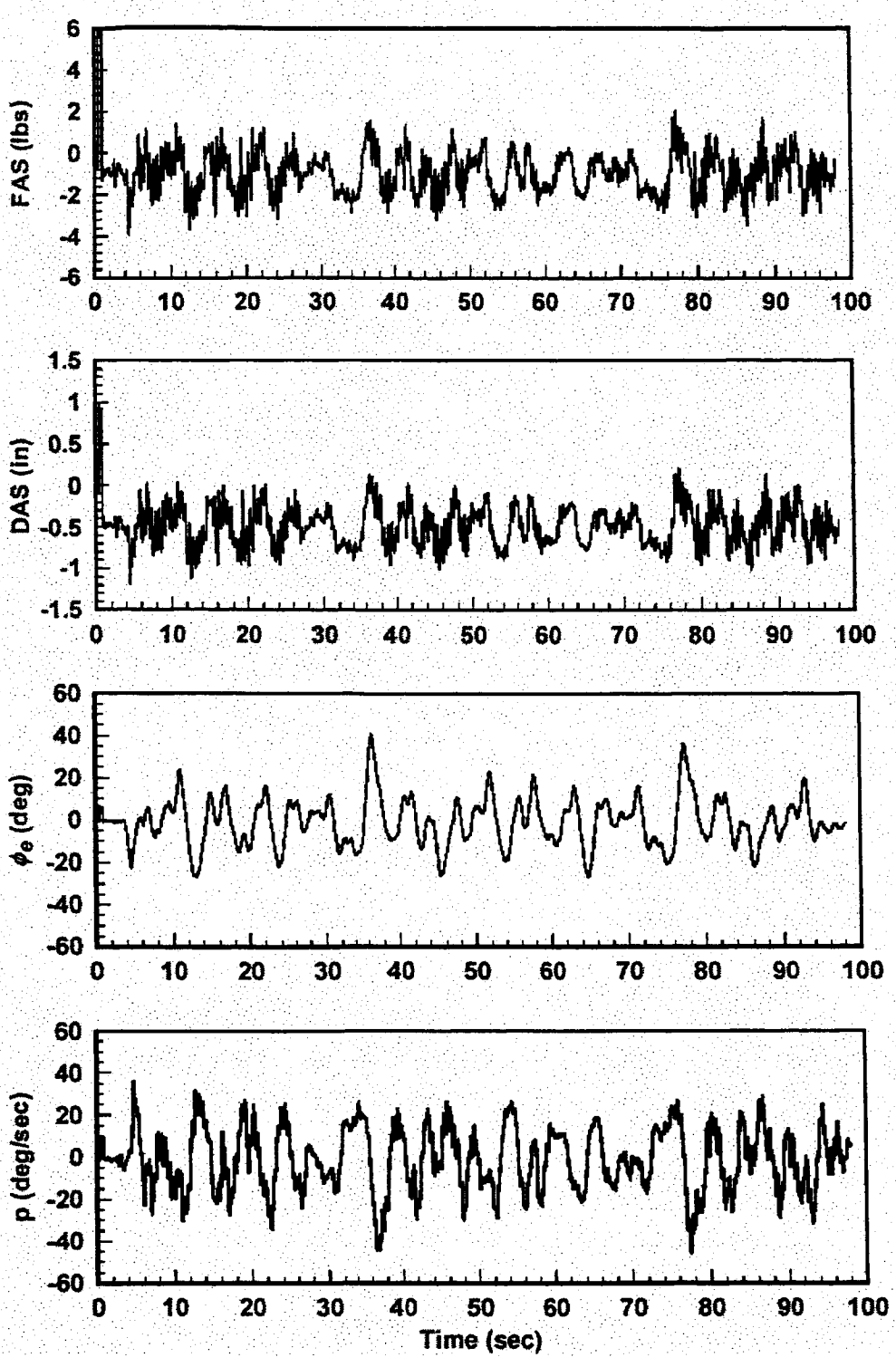


Figure 146. Example Time Histories for Intermittent Ratchet Run (4161.3)

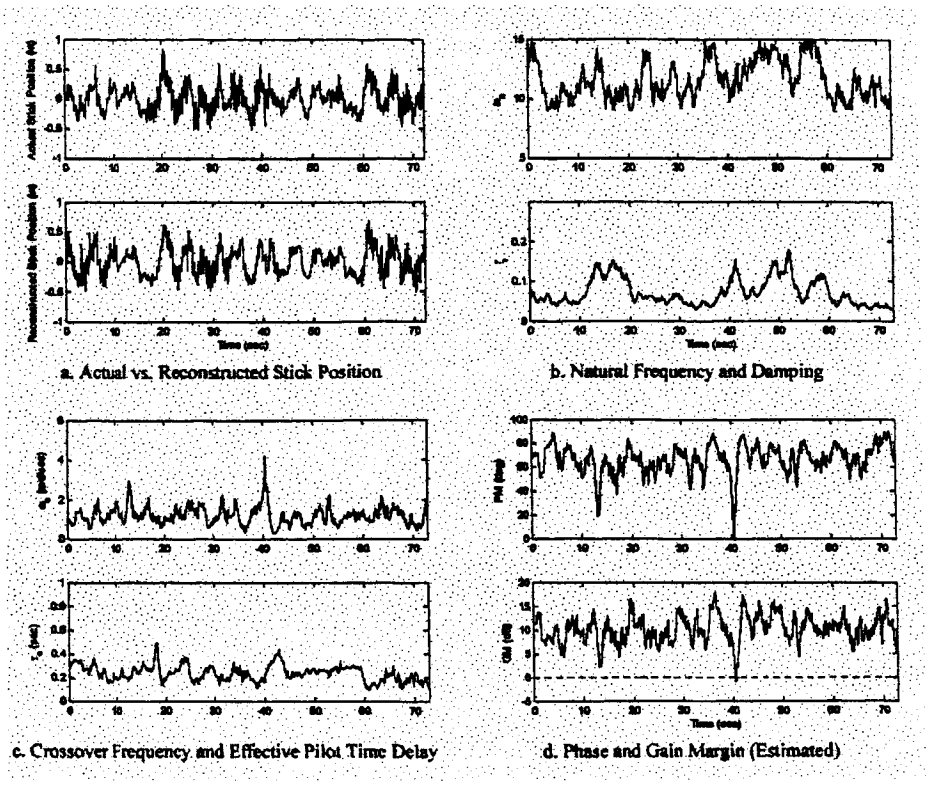


Figure 147. Pilot-Vehicle System Parameter Identification for the Intermittent Ratchet Example (Run 4161.3)

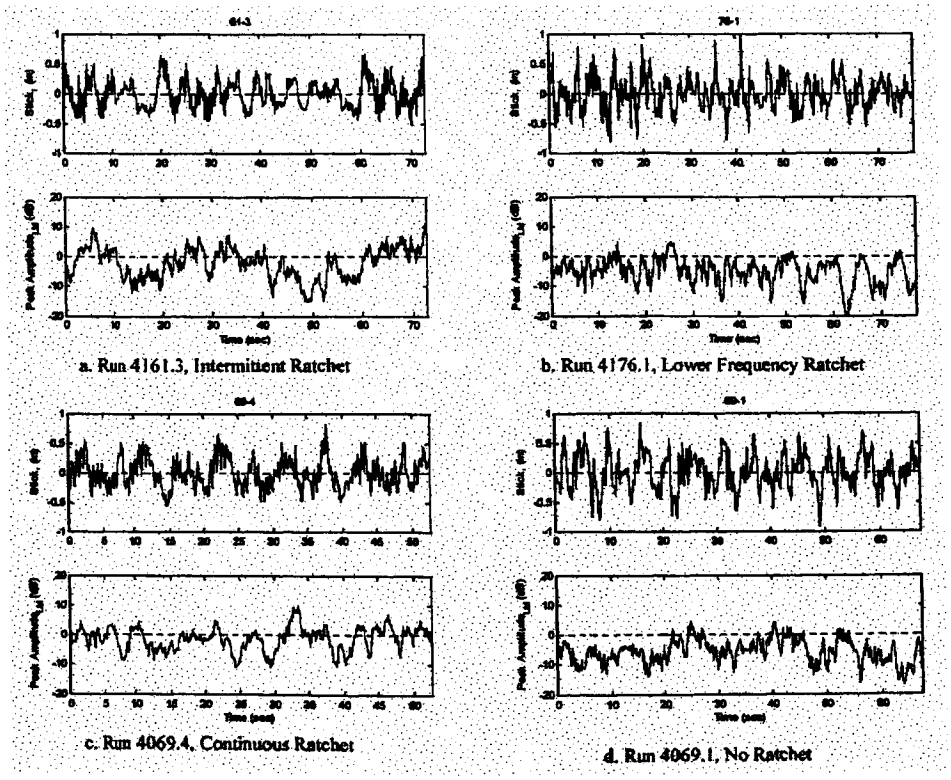


Figure 148. Stick Position (Flight Data) and Estimated Limb-Manipulator Gain Margin Time Histories

Because the tracking task generally required actions at frequencies (i.e., crossover) well below the limb-manipulator peaking, the secondary rise in open-loop magnitude at the limb-manipulator frequency can in most cases be viewed as a "nuisance" mode where higher-frequency/low amplitude ringing rides on top of the primary tracking stick signal. This can, however, significantly impact perceived handling qualities as reflected in the Level 3 pilot ratings listed in Table 37.

6. CONCLUSIONS

In previous analysis of a lateral handling qualities flight research database strong evidence was generated to support the theory that the pilot neuromuscular mode couples with the feel system dynamics to produce an effective limb-manipulator mode, the frequency of which corresponded to observed roll ratchet frequencies. A subset of this same database was analyzed using wavelet-based techniques from which the time-varying nature of key pilot-vehicle system parameters, including the limb-manipulator mode, could be examined. The results of this analysis appear to indicate that a lightly-damped mode in the open-loop system is introduced by the human pilot at frequencies higher than the task bandwidth and this corresponds to the previously identified limb-manipulator mode. A direct relationship was found to exist between the peak amplitude of the limb-manipulator mode and the occurrence of roll ratchet. From the data, a threshold appears to be at approximately -2dB, above which ratchet occurs. Even for the case where the pilot did not report ratchet, the pilot stick time history shows short intervals of ratchet-like behavior during which the estimated limb-manipulator peak approached 0 dB. Finally, the WERA identification technique was found to be a powerful tool for assessing pilot-vehicle system performance over time intervals that were previously not feasible with FFT-based analysis.

REFERENCES

1. Angner, J., C. Jensen, and M. Seidl, "JAS 39 Grippen EFCS: How to Deal With Rate Limiting" *Society of Experimental Test Pilots Fortieth Symposium Proceedings*, Beverly Hills, CA, Sept. 1996, pp. 220-233.
2. Anon., "Boeing 737 Rudder Blamed for Crashes," AeroWorldNet, 29 March 1999.
3. Dornheim, Michael A., "Report Pinpoints Factors Leading to YF-22 Crash," *Aviation Week & Space Technology*, 9 November 1992, pp. 53-54.
4. Graham, D., and D. McRuer, "Retrospective Essay on Nonlinearities in Aircraft Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 6, Nov. – Dec. 1191, pp. 1089-1099.
5. *Aviation Safety and Pilot Control – Understanding and Preventing Unfavorable Pilot-Vehicle Interactions*, Committee on the Effects of Aircraft-Pilot Coupling on Flight Safety, National Academy Press, Washington D.C., 1997.
6. Klyde, D. H., D. T. McRuer, T. T. Myers, "Pilot-Induced Oscillation Analysis and Prediction with Actuator Rate Limiting," *Journal of Guidance, Control, and Dynamics*, Volume 20, Number 1, Jan. - Feb. 1997, pp. 81 to 89.
7. Klyde, D. H., D. T. McRuer, and T. T. Myers, Unified Pilot-Induced Oscillation Theory Volume I: PIO Analysis with Linear and Nonlinear Vehicle Characteristics, Including Rate Limiting, WL-TR-96-3028, Dec. 1995.
8. McRuer, D. T., and D. E. Johnston, Flight Control Systems Properties and Problems, NASA CR-2500, Feb. 1975.
9. Magdaleno, R. E., Jex, H.R., and Sahasrabudhe, V., *Outrider AV102 Flights 19 and 20: SAS-Off Frequency Responses and Fitted Aero Model Derivatives*, Systems Technology, Inc., STI WP No. 2580-38-I, April 1998.
10. Trame, L. W., L. E. Williams, and R. N. Yurkovich, "Active Aeroelastic Oscillation Control of the F/A-18 Aircraft," AIAA Paper No. 85-1858 presented at the *AIAA Guidance, Navigation and Control Conference*, Snowmass, CO, 19 to 21 Aug. 1985.
11. McRuer, D. T., and E. S. Krendel, Numerical Models of Human Pilot Behavior, AGARDograph No. 188, Nov. 1973.
12. *Final Report on the Accident of the Falcon 900B registered SX-ECH – 14 September 1999 –* , Romania Ministry of Transport Civil Aviation Inspectorate, Nr. 711, 01.08.2000.
13. Thompson, P. M., D. H. Klyde, and V. Sahasrabudhe, *On-Line Detection of Loss of Control Using Wavelets*, STI TR-1333-1, Systems Technology, Inc., 7 June 2000.
14. Thompson, P. M., D. H. Klyde, and M. Brenner, "Wavelet-Based Time-Varying Human Operator Models," AIAA Paper 2001-4009, presented at *Atmospheric Flight Mechanics Conference*, Montreal, Canada, 6-9 Aug. 2001.
15. Mitchell, D. G., and R. H. Hoh, *Development of Methods and Devices to Predict and Prevent Pilot-Induced Oscillations*, AFRL-VA-WP-TR-2000-3046, Dec. 2000.
16. McRuer, D. T., and E. S. Krendel, Mathematical Models of Human Pilot Behavior, AGARDograph No. 188, Jan. 1974.
17. Hoh, R. H., "Advances in Flying Qualities: Concepts and Criteria for a Mission Oriented Flying Qualities Specification," *Advances in Flying Qualities*, AGARD-LS-157, May 1988, pp. 5-1 to 5-28.

18. Klyde, D. H., and D. G. Mitchell, "Investigating the Role of Rate Limiting in Pilot-Induced Oscillations," AIAA Paper No. 2003-5463 presented at the *Atmospheric Flight Mechanics Conference & Exhibit*, Austin, TX, 11-14 Aug. 2003.

19. Daubechies, I., Ten Lectures on Wavelets, SIAM, Philadelphia, 1992.

20. Mallat, S., A Wavelet Tour of Signal Processing, Academic Press, San Diego, 1998.

21. Teolis, A., Computational Signal Processing with Wavelets, Birkhauser, Boston, 1998.

22. Strang, G. and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, Wellesley, MA, 1997.

23. Shensa, M. J., "The Discrete Wavelet Transform: Wedding the A Trous and Mallat Algorithms," *IEEE Transaction on Signal Processing*, Vol. 40, No. 10, Oct. 1992, pp. 2464-2482.

24. Juang, J. N., Applied System Identification, Prentice-Hall PTR, Upper Saddle River, New Jersey, 1994.

25. Bayard, D. S., "An Algorithm for State-Space Frequency Domain Identification Without Windowing Distortions," *IEEE Trans. Automatic Control*, Vol. 39, No. 9, September 1994, pp. 1880-1885.

26. Robertson, A. N., K. C. Park, and K. F. Alvin, "Extraction of Impulse Response Data via Wavelet Transforms for Structural System Identification," *ASME Journal of Vibration and Acoustics*, Vol. 120, 1993, pp. 252-260.

27. Bendat, J. S., and A. G. Piersol, Random Data Analysis and Measurement Procedures, John Wiley and Sons, New York, N.Y., 2000.

28. Schwartz, R. J., Linear Systems, McGraw-Hill, New York, N.Y., 1985.

29. Newland, D. E., "Wavelet Analysis of Vibration, Part 1: Theory," *ASME Journal of Vibration and Acoustics*, Vol. 116, October 1994, pp. 409-416.

30. Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets", *Commun. Pure Appied Math*, Vol. 41, 1988, pp. 909-996.

31. Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Transactions in Information Theory*, Vol. 36, 1990, pp. 961-1005.

32. Robertson, A. N., K. C. Park, and K. F. Alvin, "Extraction of Impulse Response Data via Wavelet Transforms for Structural System Identification," *ASME Journal of Vibration and Acoustics*, Vol. 120, 1993, pp. 252-260.

33. Newland, D. E., Random Vibrations, Spectral & Wavelets Analysis, 3rd edition, Longman Scientific and Technical, Essex, England, 1988.

34. Aplevich, D., The Essentials of Linear State Space Systems, John Wiley and Sons, New York, N.Y., 2000.

35. Niewoehner, R., LCDR USN, and Minnich, S., "F-14 Dual Hydraulic Failure Flying Qualities Evaluation," *Thirty-Fifth Symposium Proceedings*, Society of Experimental Test Pilots, Lancaster, CA, Sep. 1991, pp. 4-15.

36. Klyde, D. H., and D. G. Mitchell, "Extraction of Pilot-Vehicle Characteristics In the Presence of Rate Limiting," AIAA 99-1069 presented at *37th Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 11 – 14, 1999.

37. Schroeder, J. A., W. W. Y. Chung, and D. T. Tran, "Pilot-Induced Oscillation Prediction with Three Levels of Simulation Motion Displacement," AIAA Paper No. 98-4333 presented at the *Atmospheric Flight Mechanics Conference*, Boston, MA, 10-12 August, 1998.

38. Nguyen, B. T., T. J. Cord, D. B. Leggett, *Comparisons of Flight to Ground-Based Pilot-Induced Oscillation Evaluation Methods*, AFRL-VA-WP-TR-1998-3032, May 1998.

39. Strang, G., Linear Algebra and Its Applications, Academic Press, New York, 1976.

40. Press, W., W. Vettering, S. Teukolsky, and B. Flannery, Numerical Recipes in C, The Art of Scientific Computing, 2nd. Ed., Cambridge Univ. Press, 1992.

41. Beckmann, P., Orthogonal Polynomials for Engineers and Physicists, Golem Press, Boulder Colorado, 1973.

42. Goodwin, G. C. and K. S. Sin, Adaptive Filtering Prediction and Control, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

43. Bayard, D. S., "Multivariable Frequency Domain Identification via 2-Norm Minimization," *Proc. American Control Conference*, Chicago, Illinois, 1992, pp. 1253-1257.

44. Bayard, D. S., "Multivariable State-Space Identification in the Delta and Shift Operators: Algorithms and Experimental Results," *Proc. American Control Conference*, San Francisco, California, 1992, pp. 3038-3041.

45. Kailath, T., Linear Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

46. Juang, J. and R. S. Papps, "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *J. of Guidance, Control, and Dynamics*, Vol. 8, No. 5, Sept.-Oct. 1985, pp. 620-627.

47. Robertson, A. N. and K. C. Park, "An Investigation of Time Efficiency in Wavelet-Based Markov Parameter Extraction Methods," AIAA Paper No. 98-1889, 1998.

48. Johnston, D. E., and D. T. McRuer, *Investigation of Interactions Between Limb-Manipulator Dynamics and Effective Vehicle Roll Control Characteristics*, NASA CR-3983, May 1986.

49. Mitchell, D. G., B. L. Aponso, and D. H. Klyde, "Feel Systems and Flying Qualities," AIAA Paper No. 95-3425 presented at the *Atmospheric Flight Mechanics Conference*, Baltimore, MD, Aug. 7-10, 1995.

50. Mitchell, D. G., B. L. Aponso, and D. H. Klyde, *Effects of Cockpit Lateral Stick Characteristics on Handling Qualities and Pilot Dynamics*, NASA CR-4443, June 1992.

51. Bailey, R. E., and L. H. Knotts, *Interactions of Feel System and Flight Control System Dynamics on Lateral Flying Qualities*, NASA CR-179445, 1990.

52. Smith, R. E., and S. K. Sarrafian, "Effect of Time Delay on Flying Qualities: an Update," *J. of Guidance, Control, and Dynamics*, Vol. 9, No. 5, Sept.-Oct. 1986, pp. 578-584.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 30, 2003	Final Technical Report 01/31/01 – 09/30/03

4. TITLE AND SUBTITLE
On-Line Loss of Control Detection Using Wavelets
Volume I: Final Technical Report

5. FUNDING NUMBERS
NAS4-010004

6. AUTHORS
Peter M. Thompson, Ph.D., David H. Klyde, Edward N. Bachelder, Ph.D., Theodore J. Rosenthal

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Technology, Inc.
13766 S. Hawthorne Blvd.
Hawthorne, CA 90250

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-1341-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASA
Dryden Flight Research Center
P.O. Box 273
Edwards, CA 93523-0273

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Wavelet transforms are used for on-line detection of aircraft loss of control. Wavelet transforms are compared with Fourier transform methods and shown to more rapidly detect changes in the vehicle dynamics. This faster response is due to a time window that decreases in length as the frequency increases. New wavelets are defined that further decrease the detection time by skewing the shape of the envelope. The wavelets are used for power spectrum and transfer function estimation. Smoothing is used to tradeoff the variance of the estimate with detection time. Wavelets are also used as front-end to the eigensystem reconstruction algorithm. Stability metrics are estimated from the frequency response and models, and it is these metrics that are used for loss of control detection. A Matlab toolbox was developed for post-processing simulation and flight data using the wavelet analysis methods. A subset of these methods was implemented in real time and named the Loss of Control Analysis Tool Set or LOCATS. A manual control experiment was conducted using a hardware-in-the-loop simulator for a large transport aircraft, in which the real time performance of LOCATS was demonstrated. The next step is to use these wavelet analysis tools for flight test support.

14. SUBJECT TERMS
Loss of control, wavelets, time varying transfer functions, Fourier transforms, eigensystem reconstruction algorithm

15. NUMBER OF PAGES
253

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

SYSTEMS TECHNOLOGY, INC

13766 S. HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250-7083 • PHONE (310) 679-2281
FAX (310) 644-3887

Technical Report No. 1341-2

ON-LINE LOSS OF CONTROL DETECTION USING WAVELETS VOLUME II: LOCATS EVALUATION

September 30, 2003

David H. Klyde
Peter M. Thompson, Ph.D.
Edward N. Bachelder, Ph.D.
Theodore J. Rosenthal

Prepared for
NASA Dryden Flight Research Center
Edwards, CA
Contract No. NAS4-01004

FOREWORD

This report documents the results of a Phase II Small Business Innovation Research (SBIR) contract to develop an on-line wavelet-based loss of control detection tool. The work was conducted by Systems Technology, Inc. (STI) in Hawthorne, CA for NASA Dryden Flight Research Center under Contract No. NAS4-01004. The period of performance for this contract was 31 January 2001 to 30 September 2003. Martin Brenner served as the Contracting Officer's Technical Representative for NASA. Peter M. Thompson was the Principal Investigator and David H. Klyde was the program manager for STI. Simulator facilities and related support were provided by BAE SYSTEMS Platform Solutions Sector in Johnson City, NY. The program manager for BAE SYSTEMS subcontract was Matthew Trouve, while the engineering effort was led by Richard Parkinson. The authors would also like to acknowledge the significant efforts of Jeffrey Gernhart, the simulator facility manager. In addition, the authors recognize the initial BAE SYSTEMS work of Andrew Corea, Timothy Horan, and Feng Liang. Finally, the authors would like to thank Douglas Thrall of BAE SYSTEMS for his support of this program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This three volume report documents the results of a Phase II Small Business Innovation Research (SBIR) contract to develop an on-line means to predict aircraft loss of control using wavelets. Systems Technology, Inc. (STI) conducted the work for NASA Dryden Flight Research Center at Edwards, CA. The overall objective of this proposed program was to apply wavelet technology in the form of a Loss of Control Analysis Tool Set or LOCATS as a means of predicting loss of control for the broad range of refractory automatic and manual control system problems. These control problems are those that escape detection by typical design criteria and methodologies, surface under unusual or rare circumstances, and threaten flight safety. Specific Phase II technical objectives were as follows:

- Extend the capabilities of the LOCATS to include improved wavelet formulations for real-time applications with validity measures;
- Identify time varying pilot-vehicle system characteristics that can be an indicator of impending loss of control;
- Identify, develop and validate wavelet-based time varying metrics to predict loss of control of automatic and manual flight control systems and to monitor overall flight control system health;
- Develop software and/or hardware solutions for the implementation of the real-time LOCATS;
- Provide a graphical user interface, real-time graphics displays, and networking capabilities for the prototype real-time LOCATS; and
- Evaluate the prototype real-time LOCATS in a piloted simulation that features flight control system software and hardware representative of current state-of-the-art aircraft.

The evaluation of the prototype LOCATS was conducted using the flight control system hardware-in-the-loop simulator facility of BAE SYSTEMS Platform Solutions Sector (here after referred to as BAE SYSTEMS) of Johnson City, NY.

Volume I of this report begins with an overview of issues involving aircraft loss of control. This is followed by a description of the LOCATS software, detailed development of the wavelet-based loss of control detection techniques, and a discussion of the results from the LOCATS piloted simulation evaluation. Supporting material is provided in the appendices. Volume II of this report provides the LOCATS piloted simulation test plan and detailed descriptions of the simulation checkout and formal evaluation. Finally, Volume III of this report contains the LOCATS User's Guide.

2. LOCATS SIMULATION TEST PLAN

2.1 Background

2.1.1 Simulation Objectives

The overall objective of this piloted simulation is to evaluate the capabilities of the STI wavelet-based LOCATS to detect impending loss of control. The specific objectives are as follows:

- Install the LOCATS software on a personal computer provided by BAE SYSTEMS that is connected via a network to the simulator facility computers;
- Receive selected simulator data in real time over the network while the pilot performs various closed-loop tasks;
- Perform assessments of the pilot-vehicle system and flight control system and present results in real time;
- Determine which wavelet-based algorithms or combination thereof provide the best results for a given loss of control scenario;
- Assess how quickly degrading conditions or impending loss of control can be detected; and
- Evaluate the utility of the selected metrics and corresponding real-time displays.

2.1.2 BAE SYSTEMS Simulator Facility

BAE SYSTEMS maintains and operates a real-time simulation facility of a fly-by-wire military transport aircraft that features a fixed-based engineering development cockpit, Figure 1, equipped with a visual display system and a control feel system.

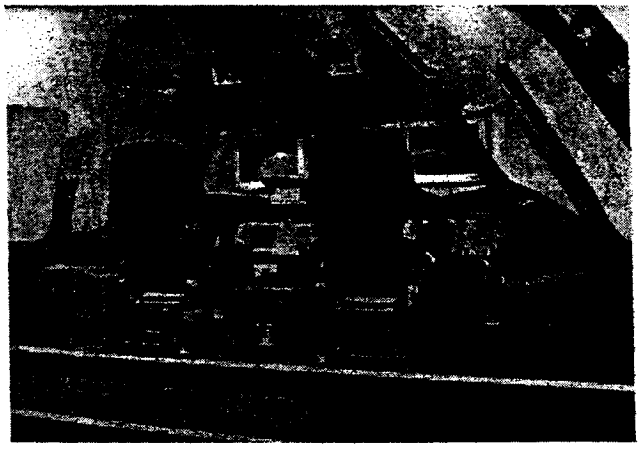


Figure 1. Fixed Based Cockpit

The cockpit interfaces with a real-time simulation host computer that is capable of running real-time models of the aircraft, environment, engine, actuator and sensor models as well as performing high-speed data acquisition. Both the cockpit and the host computer are integrated with the actual Flight Control Computer hardware located in a Hot Bench. Figure 2 shows a representative arrangement for the simulation facility. The basic development cockpit is fitted with functional cockpit panels and stick force/position transducers for operation of the flight control system. Other essential pilot vehicle interfaces such as the throttle quadrant, head-up display (HUD), and engine instrumentation are also provided to allow the pilot to perform testing effectively in a suitable and representative environment.

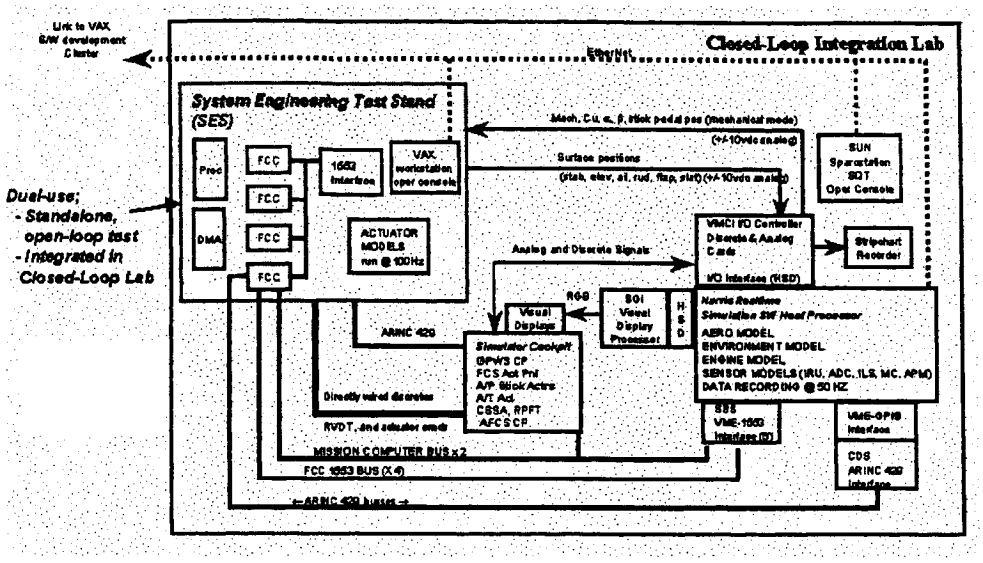


Figure 2. Simulation Facility Hardware Architecture

As shown in Figure 3 the real-time simulation facility utilizes a Concurrent computer as its host computer. The simulator cockpit interfaces with the real-time simulation host processor via an I/O subsystem. The I/O subsystem provides the data conditioning and conversion of the necessary analog and discrete signals required to allow the cockpit controls, displays, and indicators to interface properly with the simulation models executed in the simulation host computer.

Figure 3. Simulation Host Computer

The cockpit contains a visual display driven by a separate display processor that interfaces with the host computer. The visual display provides the pilot/engineer with an out-the-window visual reference that is essential in providing attitude, distance, and rate information required to allow accurate pilot loop closure. The visual display provides information to the pilot by means of a 2-dimensional drawing of the HUD generated by the visual computer and directly superimposed on the out-the-window view. Figure 4 shows the Silicon Graphics Octane computer used for visual display processing for the cockpit and for a separate monitoring station.

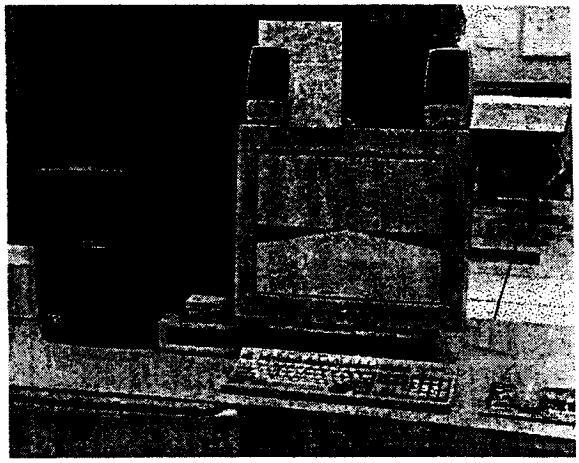


Figure 4. Visual Display Computer and Monitoring Station

The real-time simulation can be configured to run in two basic modes. The first, Software-in-the-Loop (SIL), runs with a simulated flight control model implemented in FORTRAN that runs on the host computer. The second, Hardware-in-the-Loop (HIL), utilizes the Hot Bench mentioned earlier to integrate actual Quad Redundant Flight Control Computers into the simulation environment. The Hot Bench, called the Subsystem Evaluation Station (SES), provides an excellent method for testing of the entire system, with a pilot-in-the-loop, with actual and simulated sensor and actuator signals feeding actual flight control computers. Figure 5depicts the SES station utilized during HIL simulations.

Figure 5. System Evaluation Station

2.2 Flight conditions and aircraft configurations

2.2.1 Flight Conditions

The piloted evaluations will be conducted in both up-and-away and terminal flight conditions. Specific flight condition variations are identified in Table 1. Each evaluation will begin with the aircraft trimmed.

Table 1. Flight Conditions for the Piloted Simulation

No.	ID	Description	Altitude (1000 ft)	Airspeed (KCAS)	Turbulence	Crosswinds (knots)
1	C	Cruise	20	270	None	n/a
2	CM	Cruise with Turbulence	20	270	Moderate	n/a
3	CH	High Altitude Cruise	40	200	None	n/a
4	A	Approach	0.2	130	n/a	None
5	AX	Approach with Crosswind	0.2	130	n/a	20, R to L
6	SA	Steep Approach	0.3	130	n/a	None
7	SAX	Steep Approach with Crosswind	0.3	130	n/a	20, R to L

2.2.2 Configuration Descriptions

As mentioned above the piloted evaluations will be conducted in both up-and-away and terminal flight conditions. Specific aircraft configurations variations are identified in Table 2. Once again, each evaluation will begin with the aircraft trimmed.

Table 2. Aircraft Configurations for the Piloted Simulation

No.	ID	Description	Gross Weight (1000 lb)	CG (% chord)	Flaps (deg)	Gear
1	CHF	Cruise/Heavy/FwdCG	580	0.3	Retracted	Up
2	CHA	Cruise/Heavy/AftCG	580	0.4	Retracted	Up
3	CLF	Cruise/Light/FwdCG	300	0.27	Retracted	Up
4	CLA	Cruise/Light/AftCG	300	0.42	Retracted	Up
5	ALF	Approach/Light/FwdCG	300	0.27	27.4	Down
6	ALA	Approach/Light/AftCG	300	0.42	27.4	Down
7	SALF	Steep Approach/Light/FwdCG	300	0.27	37	Down
8	SALA	Steep Approach/Light/AftCG	300	0.42	37	Down
9	SAHF	Steep Approach/Heavy/FwdCG	400	0.3	17.1	Down
10	SAHA	Steep Approach/Heavy/AftCG	400	0.4	17.1	Down

2.3 Piloted Evaluation Tasks

2.3.1 Overview

The piloted evaluation tasks included herein were defined as part of an Air Force program conducted by STI to develop a catalog of fixed-wing aircraft handling qualities demonstration maneuvers (Ref. 1). Thus the maneuver descriptions feature the format established in the aforementioned program. This format includes a list of objectives, the task description, and the desired and adequate performance requirements. The objectives identify specific handling qualities characteristics that the maneuver is designed to expose. The task descriptions provide step by step guidance to the pilot for carrying out the maneuver. These descriptions are written as if they are to be used by the pilot as a flight card. The performance requirements are written to correspond with direct use of the Cooper-Harper handling qualities rating scale (Figure 6) and if applicable the pilot-induced oscillation tendency scale (Figure 7).

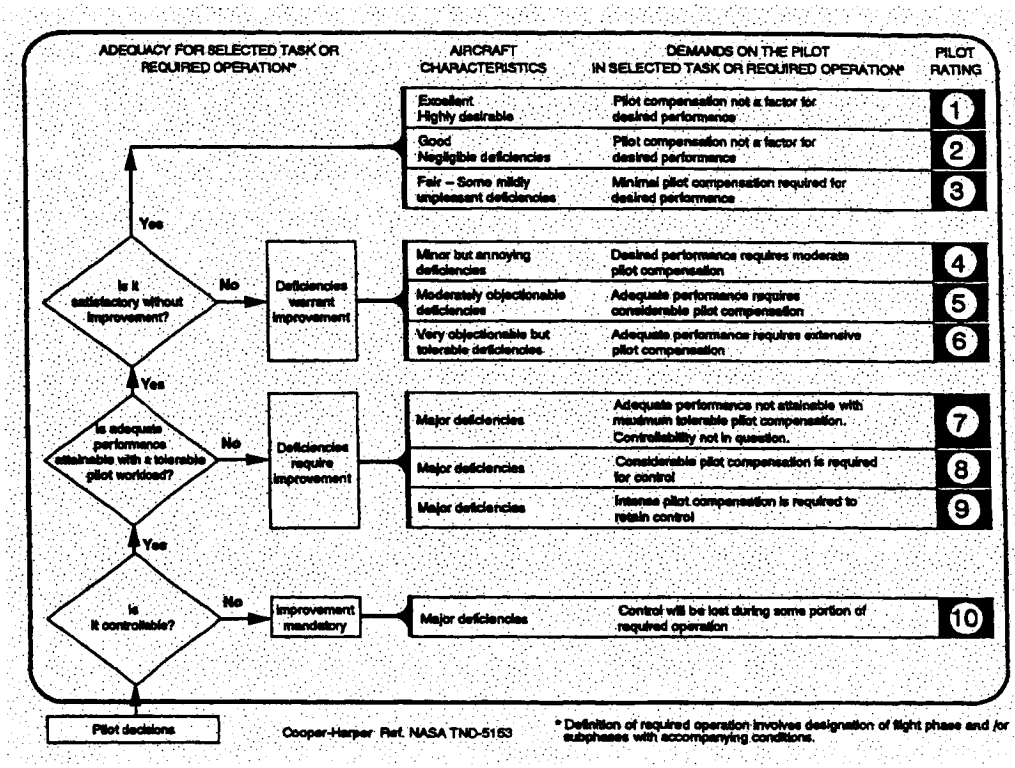


Figure 6. Cooper-Harper Handling Qualities Rating Scale (Ref. 2)

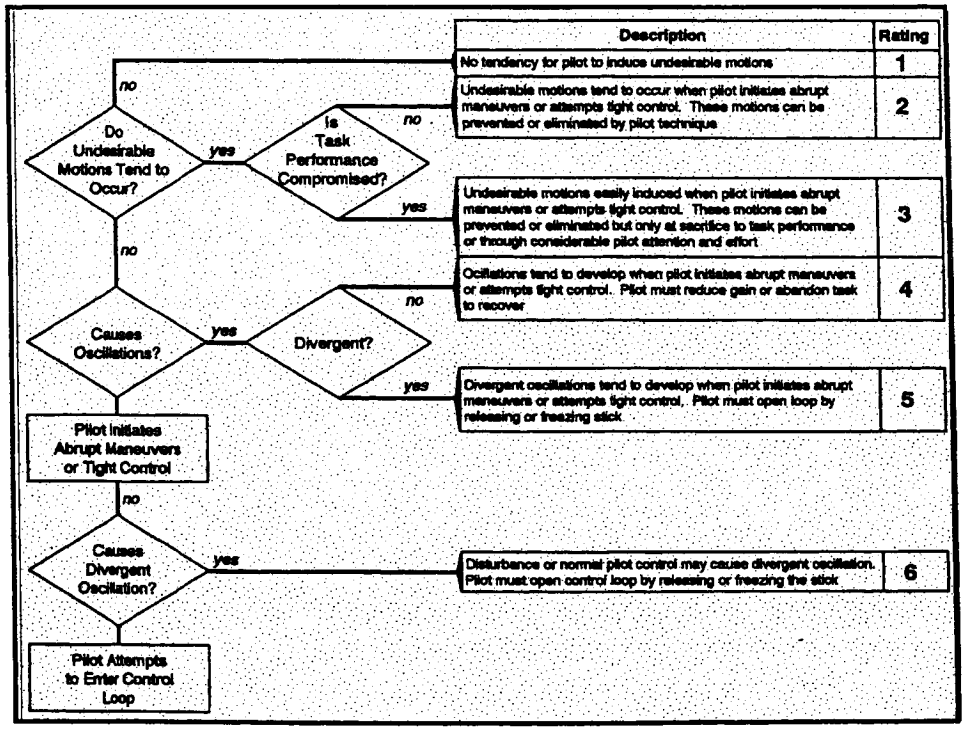


Figure 7. Pilot-Induced Oscillation Rating Scale (Refs. 3 and 4)

The demonstration maneuvers selected for this program were as follows:

- Pitch and Roll Sum-of-Sines Tracking (SOS);

- Pitch and Roll Discrete Tracking (DT);
- Pitch Attitude Capture and Hold (PACH);
- Bank Angle Capture and Hold (BACH);
- Flightpath Capture and Hold (FPCH);
- Altitude Rate Capture and Hold (ARCH);
- Precision Offset Landing (POL);
- STOL Precision Offset Landing (STOL/POL); and
- Precision ILS Capture and Track (ILS).

Note that both up-and-away and terminal flight operations are featured. Background material and task descriptions from Ref. 1 are provided below for each maneuver. Note that because this program is not a handling qualities evaluation per se, some adjustments to the maneuver descriptions and performance requirements were later made to better meet the objectives of this piloted simulation.

2.3.2 Pitch and Roll Sum-of-Sines Tracking (SOS)

2.3.2.1 Background

Tracking a computer-generated command signal produces a highly repeatable, precise task from which measures of pilot performance can be obtained. While such measures should never be interpreted as a direct indicator of handling qualities, they are one sure way to monitor pilot workload and confirm that workload is as desired. Use of similar tracking tasks has for decades provided a source of information on pilot/vehicle closed-loop dynamics. The maneuver proposed here is based on this experience, but with some of the nuances of past applications removed for simplicity.

Two of the most common types of tracking commands are the Veridian (formerly Calspsan Corporation) step-and-ramp (discrete tracking) and a sum-of-sinewaves. The discrete task was found in two USAF TPS projects (Refs. 5 and 6) to be better at producing repeatable, consistent pilot ratings. The sum-of-sines (SOS) task, however, is considered a more robust form of command for extracting pilot/vehicle information as long as the effective bandwidth of the forcing function is sufficiently high (Ref. 7). Also, the pilots must understand that this maneuver is a surrogate for an air-to-air tracking task and that it should not be viewed as a video game.

Ideally, the display for this maneuver will be a head-up display (HUD) with an appropriate command bar and fixed pippers for desired and adequate performance. Realistically, however, such a display may be restricted only to experimental and research aircraft, at least for the near future, so the maneuver should not be limited only to HUDs. The display device may be the command bars on a flight director, the glideslope and localizer bars on a horizontal situation indicator, etc. If a HUD is not used, the definitions of command scaling (units of actual error per displayed error) and performance limits become critical. In addition, for any display, the dynamics of the display drivers become important. Total throughput delay should at least be known, even if nothing can be done about it.

It is important to recognize that the signal displayed to the pilot is tracking error, not the command. For example, in pitch $\theta_{display} = \theta_c - \theta$, where θ_c is the command from the sum of sinewaves and θ is airplane attitude. This makes the task compensatory in nature: the pilot compensates for displayed changes in attitude rather than pursuing an attitude command. Ideally, the pilot is not given any information on actual attitude, including out-the-window, if the cockpit orientation allows. If a HUD is used, blackout material behind the HUD and an instrument training hood will usually suffice.

The form of the SOS command is as follows:

$$X_c = \sum_{i=1}^{n} A_i \sin(\omega_i t + \phi_i)$$

A recommended forcing function is defined below for pitch control and for roll control. These are not the only acceptable forcing functions, and the user may wish to devise other sets of commands to emphasize different frequencies of control. Based on considerable experience with SOS tracking, certain guidelines can be specified in devising the forcing function:

- Several discrete frequencies should be used. At least six sinewave signals should provide a broad range of commands. More are better but are not necessary.
- For pitch and roll, the frequencies of the sinewaves (ω_i) should vary from around 0.2 to 0.4 rad/sec up to 8 to 10 rad/sec. Magnitude of the sinewaves (A_i) should be reduced above about 1.5 to 2 rad/sec, as needed to provide a bandwidth of 1.5 rad/sec or so.
- For other axes, such as heading, flightpath, load factor, or airspeed, the pilot's bandwidth of control is typically lower than for pitch or roll. The upper frequency should be reduced accordingly, as should the effective bandwidth of the SOS signal. (If the pilot is given a direct control of one of these axes, or if there is a requirement for high-frequency control, the guidance for pitch and roll should be followed.)
- Concentrate the sinewaves below the bandwidth frequency of the total signal. If only six discrete frequencies are used, one or two should be dedicated to higher frequencies (but with reduced magnitudes as noted above). This provides a high-frequency "gust-like" component that helps make the command look more random to the pilot. If more than six sinewaves are used more high-frequency components can be included.
- Overall root mean square (RMS) magnitudes should be around 2.5° attitude in pitch and 25° in roll (Ref. 8) for simulated up-and-away operations. These numbers may need to be reduced somewhat if the tracking is to be performed in landing configuration, or at high speed if the resulting load factor is excessive.
- Start some of the sinewaves with negative amplitudes or with nonzero initial phase angles), so the initial displayed error is not always large and in the positive direction. If randomness is important (which is the case if one pilot is to fly the task many times, the signal can be made to look random by adding an arbitrary phase shift (ϕ_i) to some of the sinewaves and varying this phase shift from run to run.
- The entire run should be around one minute to 90 seconds in length. Anything longer may be fatiguing to the pilot; a shorter run may not provide sufficient exposure to the airplane, even if the task is immediately repeated.
- If pilot/vehicle data are to be extracted, care should be taken to assure that the entire run of scored data (data to be analyzed) includes at least one full cycle of each sinewave and that there is an integer number of all sinewaves. In addition, a warm-up period at the start should be added. This period is several seconds of non-scoring time during which the command is ramped in, allowing the pilot to get into the control loop before formal data taking begins.

More information about the design of the sum-of-sines forcing function, including example numbers, can be obtained from Refs. 7 and 8. Many more references on this topic are available.

Recommended forcing functions are given in Table 3. The pitch numbers are from Ref. 6, while the roll values are from Ref. 7. Figure 8 shows a time history of the pitch signal for reference. The task time for both command sets is 74.25 seconds, consisting of the following: 10 seconds of warm-up (non-scored time); 63 seconds of tracking for scoring; 1.25 seconds of cool-down (non-scored time). All sinewaves complete an integer number of cycles in 63 seconds, as listed in the table. Sinewave frequency, ω_i (in rad/sec), is computed from the scoring time, t_s, and number of cycles of each sinewave, N_i, as follows:

$$\omega_i = \frac{2\pi N_i}{t_s}$$

Table 3. Recommended SOS Command Signals for Pitch and Roll

Sine-wave No.	Pitch Attitude			Roll Attitude		
	A_i (deg)	No. cycles	ω_i (r/s)	A_i (deg)	No. cycles	ω_i (r/s)
1	-1	2	0.19947	-8	3	0.2992
2	1	5	0.49867	-8	4	0.39893
3	1	9	0.8976	8	7	0.69813
4	0.5	14	1.39626	4	18	1.79519
5	-0.2	24	2.39359	-1.6	30	2.99199
6	0.2	42	4.18879	-1.6	40	3.98932
7	-0.08	90	8.97597	0.64	70	6.98131

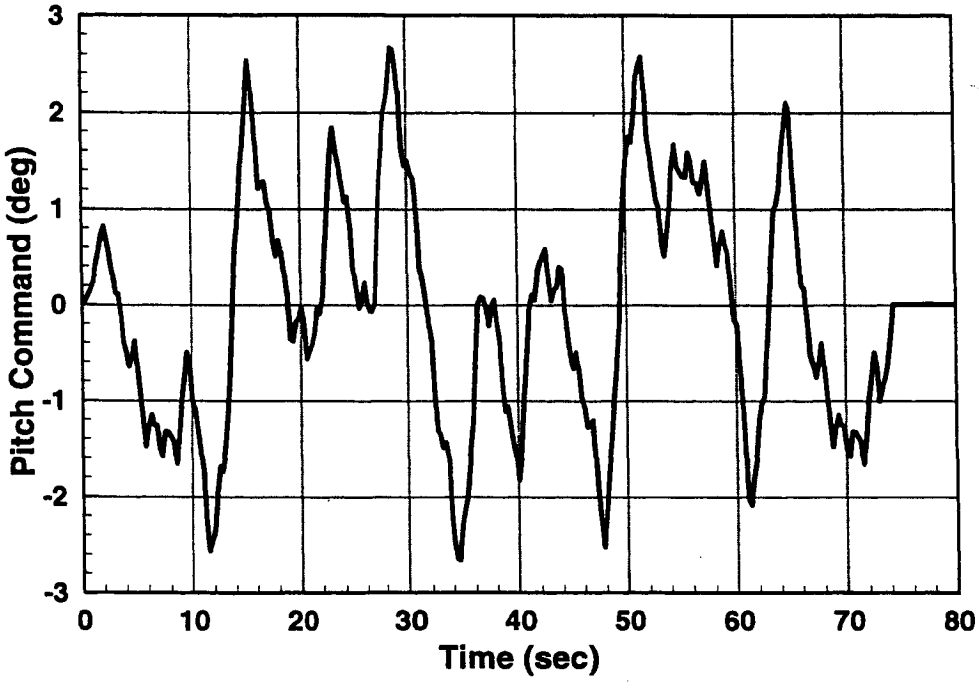


Figure 8. Example Pitch Axis Sum-of-Sines Command Signal

In the flight research of Ref. 6, the pitch SOS command resulted in load factors between about 0.5g and 1.6g. Higher load factors are possible but will begin to require extremely large pilot inputs; lower load factors may not reveal much about the handling qualities of the airplane.

Desired and adequate performance limits are written in terms of mils of tracking error and are clearly geared toward a HUD. The format of these requirements is taken from Ref. 8, but the numbers are twice those used in Ref. 8. It was found in initial shakedown for the tests of Ref. 6 that the original limits were too small for the pilots to observe much less attempt to meet.

Single-axis tracking may be obtained by simply turning off the command signal in one axis. This is perfectly acceptable and may be appropriate when the focus is on only one axis. Likewise, more than two axes may be tested simultaneously by cascading commands for more controls. For example, in the ground-based simulations of Ref. 7 tracking was performed in pitch, roll, and airspeed singly and in every combination.

2.3.2.2 Task Description

Objectives

- Evaluate handling qualities in a tight, closed-loop tracking task.
- Evaluate feel system and control sensitivity characteristics.
- Identify bobble or PIO tendencies.

Description

Aggressively track the displayed signal and attempt to keep errors within the specified tolerances.

Desired Performance

- ±10 mils in pitch 50% of the time.
- ±10 mils in roll (measured at the end of the command bar) 50% of the time.

Adequate Performance

- ±20 mils in pitch 50% of the time.
- ±20 mils in roll (measured at the end of the command bar) 50% of the time.

2.3.3 Pitch and Roll Discrete Tracking (DT)

2.3.3.1 Background

Tracking a computer-generated command signal produces a highly repeatable, precise task. Two of the most common types of tracking commands are the Veridian step-and-ramp (discrete tracking) and a sum-of-sinewaves. As mentioned above, the discrete task was found in two USAF TPS projects (Refs. 5 and 6) to be better at producing repeatable, consistent pilot ratings.

The discrete tracking task has been developed and refined by Veridian over the years, from the "Neal-Smith" task (Ref. 4) in pitch, to "LATHOS" (Ref. 9) in roll, and a pitch-and-roll combination (Ref. 10). The two-axis version presented here is the most recent, flown in both Refs. 5 and 6. Many of the subtleties of the task were provided by Veridian to the authors of 6 and are repeated here.

Ideally, the display for this maneuver will be a head-up display (HUD) with an appropriate command bar and fixed pippers for desired and adequate performance. Realistically, however, such a display may be restricted only to experimental and research aircraft, at least for the near future, so the maneuver should not be limited only to HUDs. The display device may be the command bars on a flight director, the glideslope and localizer bars on a horizontal situation indicator, etc. If a HUD is not used, the definitions of command scaling (units of actual error per displayed error) and performance limits become critical. In addition, for any display, the dynamics of the display drivers become important. Total throughput delay should at least be known, even if nothing can be done about it.

Table 4 lists the breakpoints for the pitch and roll command signals as a function of time. Plots of the command signals are shown in Figure 9 and Figure 10. These signals may be scaled as needed to suit the load factor and roll rate ranges under study. In the flight research of Ref. 8 scale factors were applied in both pitch and roll, as described below. For Ref. 8, the pitch command resulted in load factors between about 0.3g and 2g. Higher load factors are possible but will begin to require extremely large pilot inputs; lower load factors may not reveal much about the handling qualities of the airplane. For this simulator evaluation the roll command was reduced by 25% to better meet the performance characteristics of the large transport aircraft simulation model. This reduction is reflected in Figure 10, while the original values remain in Table 4 for comparison.

The signal displayed to the pilot is tracking error, not the command. When used by Veridian, there is a scale factor on the signal before it is displayed as well. In addition, Euler pitch attitude is adjusted for bank angle. In pitch, $\theta_{display} = K_{e\theta}(\theta_c - \theta' + \theta_{bias})$, where $K_{e\theta} = 0.86$, θ_c is the command (from the table), θ_{bias} is airplane attitude at the start of the task, and $\theta' = \theta/cos(\phi)$. In roll, $\phi_{display} = K_{e\phi}(\phi_c - \phi + \phi_{bias})$, where $K_{e\phi} = 0.88$, ϕ_c is the command (from the table), ϕ is bank angle, and ϕ_{bias} is airplane bank angle at the start of the task. The display of error, rather than pure command, makes the task compensatory in nature: the pilot compensates for displayed changes in attitude rather than pursuing an attitude command.

Desired and adequate performance limits are written in terms of mils of tracking error and are clearly geared toward a HUD. The format of these requirements is taken from Ref. 8, but the numbers are twice those used in Ref. 8. It was found in initial shakedown for the tests of Ref. 6 that the original limits were too small for the pilots to observe much less attempt to meet. Single-axis tracking may be performed by zeroing the command signal in one axis. Veridian recommends, however, that two-axis tracking always be performed to assure good pilot-in-the-loop operations.

Table 4. Breakpoints for Discrete Tracking Command Signals

Time (sec)	Pitch Command (deg)	Roll Command (deg)	Time (sec)	Pitch Command (deg)	Roll Command (deg)	Time (sec)	Pitch Command (deg)	Roll Command (deg)
0	0	0	45	3	-30	92.6	0	0
0.1	2	0.6	45.1	3	0	95	0	0
5	2	30	52.5	3	70	95.1	-1	0
5.5	3	30.6	55	3	70	97.5	-1	0
10	3	60	60	1.5	70	97.6	-1	-70
10.1	2	30	65	0	-23.3	100	-1	-70
12.5	2	30	67.5	0	-70	102.5	-1.67	-70
15	0.33	30	70	0	-70	107.5	-3	-23.3
17.5	-1.33	-45	70.1	0	-70	110	-3	0
20	-3	-30	72.5	0	-70	112.5	-3	0
22.5	2	-15	72.6	-2	-70	112.6	-1	60
25	2	0	75	-2	-70	115	-1	60
25.1	2	15	75.1	0	-70	115.1	3	60
27.5	2	30	77.5	0	-70	117.5	3	60
30	0	45	80	0	30	117.6	3	-30
35	-3	45	80.1	0	30	120	3	-30
35.1	-2	43.5	82.5	0	30	120.1	2	-30
37.5	-2	7.5	85	0	30	122.5	2	-30
37.6	-1	6	87.5	0	0	126.6	1	-15
40	-1	-30	90	0	0	127.5	1	-15
42.5	-1	-30	90.1	0	30	127.6	0	0
42.6	3	-30	92.5	0	30	150	0	0

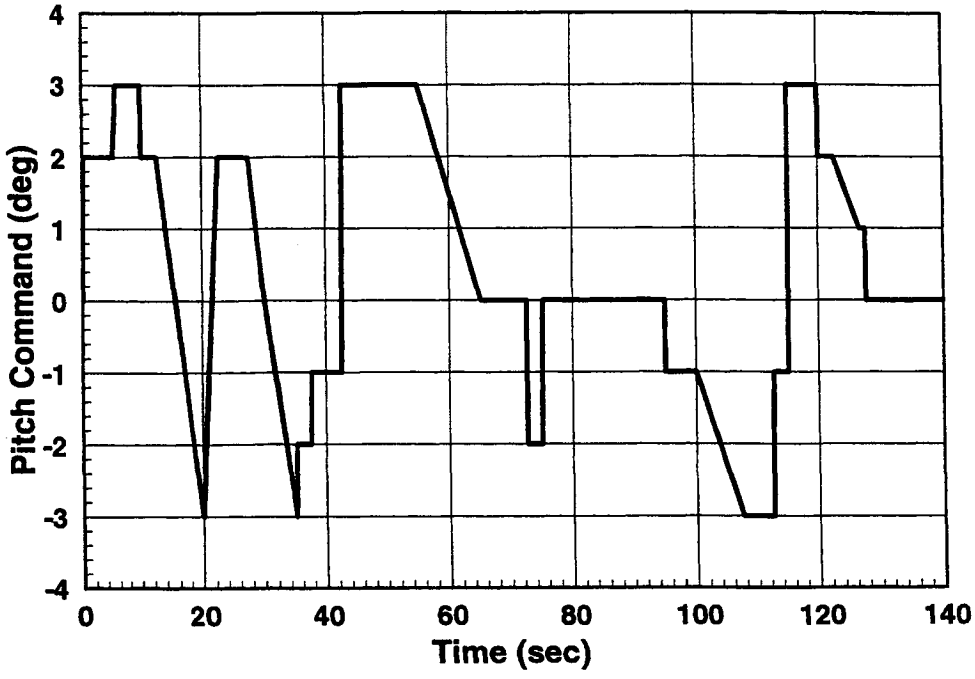


Figure 9. Pitch Axis Discrete Command Signal

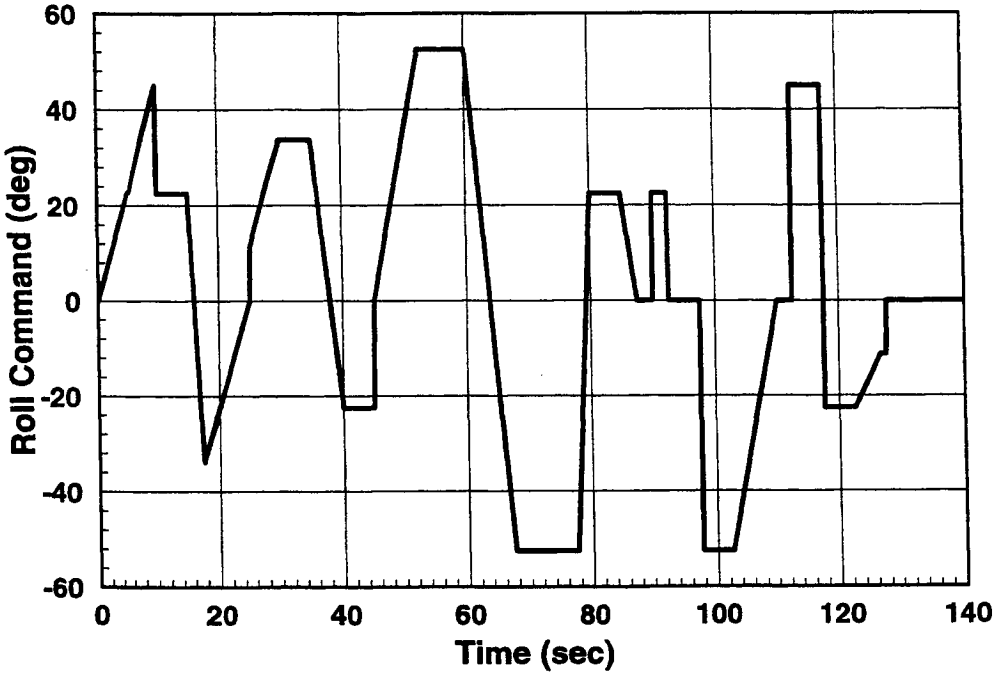


Figure 10. Roll Axis Discrete Command Signal

2.3.3.2 Task Description

Objectives

- Evaluate handling qualities in a tight, closed-loop tracking task.

- Evaluate feel system and control sensitivity characteristics.
- Identify bobble or PIO tendencies.

Description

Aggressively track the displayed signal and attempt to keep errors within the specified tolerances.

Desired Performance

- ±10 mils in pitch 50% of the time.
- ±10 mils in roll (measured at the end of the command bar) 50% of the time.

Adequate Performance

- ±20 mils in pitch 50% of the time.
- ±20 mils in roll (measured at the end of the command bar) 50% of the time.

2.3.4 Pitch Attitude Capture and Hold (PACH)

2.3.4.1 Background

Flight and simulation evaluations of pitch capture tasks have shown that the pilot's assessment of performance is greatly affected by the quality and type of visual display. For example, if a HUD pitch ladder with 5° increments is used, a 5° capture from a nonzero trim attitude means the pilot must judge performance with no obvious reference. If, on the other hand, the target attitude is a 5° line on the HUD, the pilot will have a more compelling cue. For this reason the *Description* refers to an attitude change of "at least 5°" and a target attitude that is clearly marked.

2.3.4.2 Task Description

Objectives

- Evaluate ability to pitch and capture an attitude.
- Identify maneuverability limitations and PIO tendencies.

Description

From steady level flight rapidly capture a pitch attitude of at least 5° above trim and maintain this attitude within the specified tolerances for 5 seconds or until stable. The target attitude should be defined by a clearly displayed reference (e.g., a HUD pitch ladder). Then perform subsequent captures of the initial trim attitude, -5° below trim, and trim attitude to complete the maneuver. As with the initial capture, the actual target attitudes should again be defined by a clearly defined reference. Before proceeding to the next capture, maintain each attitude within the specified tolerances for 5 seconds or until stable. Maintain wings level flight within the specified tolerances throughout the maneuver.

Desired Performance

- ±1° pitch attitude.
- ±2° deviation in bank angle.
- No more than one pitch attitude overshoot for each capture. Magnitude of overshoot remains within the desired region.

Adequate Performance

- ±2° pitch attitude.

- ±5° deviation in bank angle.
- No more than one pitch attitude overshoot for each capture. Magnitude of overshoot remains within the adequate region.

2.3.5 Bank Angle Capture and Hold (BACH)

2.3.5.1 Background

This maneuver is representative of a standard attitude capture task.

2.3.5.2 Task Description

Objectives

- Evaluate ability to roll and capture a bank angle.
- Identify maneuverability limitations and PIO tendencies.

Description

From steady, wings level flight roll and capture a bank angle of 45° and maintain this bank angle within the specified tolerance for 5 seconds or until stable. Capture the 45° bank angle before achieving a maximum heading change of 10°. Then capture and hold a bank angle of -45° and maintain this bank angle within the specified tolerance for 5 seconds or until stable. Finally, return to steady, wings level flight and maintain within the specified tolerance for 5 seconds or until stable.

Desired Performance

- ±2° bank angle.
- No more than one bank angle overshoot for each capture. Magnitude of overshoot remains within the desired region.

Adequate Performance

- ±5° bank angle.
- No more than one bank angle overshoot for each capture. Magnitude of overshoot remains within the adequate region.

2.3.6 Flightpath Capture and Hold (FPCH)

2.3.6.1 Background

Precise control of flightpath, while maintaining airspeed, is critical for many operational elements. This maneuver is aimed specifically at evaluating path/speed coupling that might degrade the ability to perform aerial recovery or rendezvous tasks.

2.3.6.2 Task Description

Objectives

- Evaluate ability to maintain airspeed during initiation of climbs and descents.
- Evaluate coupling between airspeed and flightpath.

Description

From steady level flight, rapidly pitch over and reduce power to attain a steady dive angle of -5° within the specified tolerances. After an altitude loss of 500 ft, rapidly pitch up and increase power to attain a

steady climb angle of 5° within the specified tolerances. Smoothly level off at the initial altitude. Repeat the maneuver by first climbing and then diving.

Desired Performance

- ±1° deviation in flightpath angle.
- Attain target descent and climb angles with no more than one overshoot. Magnitude of overshoot remains within the desired region.
- ±10 kt deviation in airspeed.

Adequate Performance

- ±2° deviation in flightpath angle.
- Attain target descent and climb angles with no more than one overshoot. Magnitude of overshoot remains within the adequate region.
- ±20 kt deviation in airspeed.

2.3.7 Altitude Rate Capture and Hold (ARCH)

2.3.7.1 Background

This maneuver is representative of a standard altitude rate capture task. Without a direct readout of rate, however, the evaluation pilot will need to determine the appropriate pitch attitudes that correspond to the desired climb and descent rates.

2.3.7.2 Task Description

Objectives

- Evaluate precision climb rate characteristics in isolation.
- Evaluate phugoid stability.

Description

From steady level flight rapidly capture and maintain a climb rate of 1,000 fpm for an altitude change of 500 ft. Then immediately capture and hold a sink rate of 1,000 fpm and return to the initial altitude. Then immediately return to level flight. Power should be held constant throughout the maneuver. To evaluate potential low frequency problems maintain steady conditions for at least 30 seconds to start and end the maneuver and at the acquired climb and sink rates.

Desired Performance

- ±100 fpm deviation in climb or sink rate.
- No more than one climb or sink rate overshoot. Magnitude of overshoot remains within the desired region.

Adequate Performance

- ±300 fpm deviation in climb or sink rate.
- No more than one climb or sink rate overshoot. Magnitude of overshoot remains within the adequate region.

2.3.8 Precision ILS Capture and Track (ILS)

2.3.8.1 Background

Ability to capture and precisely track an ILS (Instrument Landing System) signal depends upon good airspeed stability and control as well as favorable path/speed coupling. This maneuver is written specifically in terms of an ILS command signal, but it is not meant to be used only when an ILS is to be used. It may be applied for other ground-based, or even aircraft-based (such as GPS), landing signals intended for operations in IMC. The requirements for precision, written here in terms of conventional "dots," will need to be converted to comparable requirements for other types of displays.

2.3.8.2 Task Description

Objectives

- Evaluate ability to acquire and track an approach signal in IMC.
- Evaluate speed stability (control force per knot).
- Identify path/speed coupling in IMC.

Description

With the aircraft configured for final approach and from an approach course that is offset at least 1½ dots laterally and vertically, rapidly maneuver to reacquire centerline of the ILS beam. Start correction when the aircraft is 5 nm from the runway threshold, or as it crosses the outer marker. The most critical combination of offsets (high, low, left, right) should be emphasized. Track the ILS beam until the missed-approach point or an altitude of 400 ft AGL, whichever is higher.

Desired Performance

- Attain trimmed flight on the ILS (within desired performance limits) at a range from the runway of 2 nm or greater.
- Maintain trimmed approach speed within ±5 kts.
- Maintain glideslope and localizer within ±½ dot.

Adequate Performance

- Attain trimmed flight on the ILS (within adequate performance limits) before crossing the middle marker (approximately 0.5 nm from the runway).
- Maintain trimmed approach speed within ±10 kts.
- Maintain glideslope and localizer within ±1 dot.

2.3.9 Precision Offset Landing (POL)

2.3.9.1 Background

The offset landing has become a common task for evaluation of handling qualities in landing formation. As recently as the 1960s and early '70s, it was not unusual for "landing" handling-qualities evaluations tasks to end with a low approach. But it has since become clear that the final few feet before main-gear touchdown are critically important in exposing any dangerous deficiencies. The landing task has evolved to include lateral offsets from the runway centerline, both to investigate lateral handling qualities and to force the pilot to maneuver aggressively just prior to touchdown. When flown on variable-stability aircraft at Veridian, one or two vertical gusts (with a (1-cosine) shape) are injected on short final to further perturb the airplane from a stable approach.

The task as described here is taken from the 1995 Notice of Change to MIL-STD-1797A (Ref. 11). It has been modified and clarified based upon the review of the offset landing task documented in Ref. 12.

Based on a careful review of the precision offset landing, both the offset correction altitude and the length of the touchdown box have been made variables that are functions of groundspeed. Experience has shown that aircraft approach speed and ambient winds can combine to introduce great variability to the landing task. An airplane that has a very low approach airspeed, or an airplane with a high approach airspeed in a strong headwind, will have a much easier task than a high-speed airplane in no winds or in a tailwind. The landing box lengths are nominally 400 and 1,000 ft for desired and adequate performance, respectively, for an airplane with an approach speed of 132 kts and no wind. This is roughly the approach speed of the USAF TIFS and NT-33A (operated by Veridian) from which the bulk of experience with this maneuver comes. We have assumed that KIAS is approximately equal to groundspeed for flight experiments with these aircraft.

Obviously, including task descriptions and performance parameters that are based on a variable such as wind can make the definition of the task difficult at best. It is not intended that the offset altitude and landing box size change from run to run, but rather that they be adjusted whenever there is a sufficiently large wind to justify such a change. A good rule of thumb is that any difference in approach speed from 132 kts should be accounted for when it will result in a change in offset altitude of 20 ft or more or a change in box length of 100 ft or more.

Although it is not specified in the maneuver, a clearly defined aimpoint for the landing can greatly improve performance. Such an aimpoint may be a large white box (5 ft by 5 ft or larger) or a line, typically placed ¼ of the way down the desired landing zone. Lack of such a target sometimes results in apparent landing scatter simply because different pilots will use different target points on the runway

2.3.9.2 Task Description

Objectives

- Evaluate ability to precisely control horizontal and vertical flightpath and airspeed.
- Evaluate ability to precisely control sink rate and attitude in the flare.
- Evaluate tendency for nose bobble or PIO.
- Evaluate control sensitivity and harmony in landing.

Description

Initiate the maneuver approximately one mile out on final approach with a lateral offset of 200 ft from the runway centerline and on glideslope. The approach angle (glideslope) shall be as appropriate for the aircraft being tested (roughly 3° for conventional approach). At 200 ft AGL aggressively correct the lateral offset to land on the runway centerline with wings level. This is for an approach groundspeed of 132 kts. If approach groundspeed differs from this speed by more than 10 kts, adjust the offset correction altitude by $1.515 \cdot (V_{G/S} - 132)$, where $V_{G/S}$ is the approximate groundspeed in the approach (in kts). Touch down within the prescribed touchdown zone with the aircraft centerline within the width of the zone and the main gear within the length of the zone. The task may be performed in any wind or turbulence conditions as allowed by the operational limits of the aircraft. Testing in extreme wind conditions (crosswinds and shears) is recommended. Additional constraints that are necessary for safety may be implemented as appropriate. For example, restrictions may be placed on bank angles attained by large aircraft at low altitude.

Desired Performance

- Touch down within ±5 kts of landing airspeed.

- Touch down in a box that is 20 ft wide by 400 ft long. This is for an approach groundspeed of 132 kts. If approach groundspeed differs from this speed by more than 10 kts, adjust the box length by $3.03 \cdot (V_{G/S} - 132)$, where $V_{G/S}$ is the approximate groundspeed in the approach (in kts).
- Touchdown sink rate less than 4 fps (or no bounce and no hard landing if touchdown sink rate is difficult to measure).

Adequate Performance

- Touch down within ±10 kts of landing airspeed.
- Touch down in a box that is 40 ft wide by 1,000 ft long. This is for an approach groundspeed of 132 kts. If approach groundspeed differs from this speed by more than 10 kts, adjust the box length by $7.576 \cdot (V_{G/S} - 132)$, where $V_{G/S}$ is the approximate groundspeed in the approach (in kts).
- Touchdown sink rate less than 6 fps or 80% of the gear limit, whichever is greater (or no more than one bounce and no hard landing if touchdown sink rate is difficult to measure).

2.4 Simulated Flight Control System Failures

A list of flight control system changes that can be induced in real-time using the simulator change-a-gain feature is provided in Table 5. The demonstration maneuvers that will be used with a given scenario are also identified in the table. It is anticipated that this table will be revised pending the results of the simulation checkout.

Table 5. Simulated Flight Control System Failures

No.	ID	Description	Demonstration Maneuvers
1	PitchTC	Alter pitch stick filter time constant	All pitch axis and dual axis tasks
2	PitchGS	Alter pitch stick gain shaping	All pitch axis and dual axis tasks
3	PitchTD	Inject pitch stick time delay	DT, SOS, PACH, POL, STOL/POL
4	RollTD	Inject roll stick time delay	DT, SOS, BACH, POL, STOL/POL
5	QGain	Alter pitch rate feedback	All pitch axis and dual axis tasks
6	PGain	Alter roll rate feedback	All roll axis and dual axis tasks
7	QLIM	Reduce pitch rate command path limiter	DT, SOS, PACH, POL, STOL/POL
8	PLIM	Reduce roll rate command path limiter	DT, SOS, BACH, POL, STOL/POL

2.5 Requirements and Procedures

2.5.1 Simulator Facility Requirements

2.5.1.1 LOCATS Host PC

To ease logistical and potential security constraints, BAE SYSTEMS acquired the host PC that was integrated into the simulator facility. The host PC was equipped as follows:

- Windows 2000 OS;
- Pentium IV 2.4 GHz or higher processor with 1 GB ram; and
- ATI Radeon video card.

In addition the PC included a dual-channel SBS 1553 Bus Monitor card to access simulator data.

2.5.1.2 Simulator HUD

When performing the up-and-away pitch and roll tracking tasks, an error bar and desired and adequate performance target pippers of 10 and 20 milliradian radii, respectively, will be superimposed on the HUD as shown in Figure 11. The error bars will indicate the difference between the commanded attitudes and the actual attitudes as shown in Figure 12. It will be the goal of the pilot to minimize this error. The desired and adequate performance requirements are set to the values defined in the above SOS and DT task descriptions.

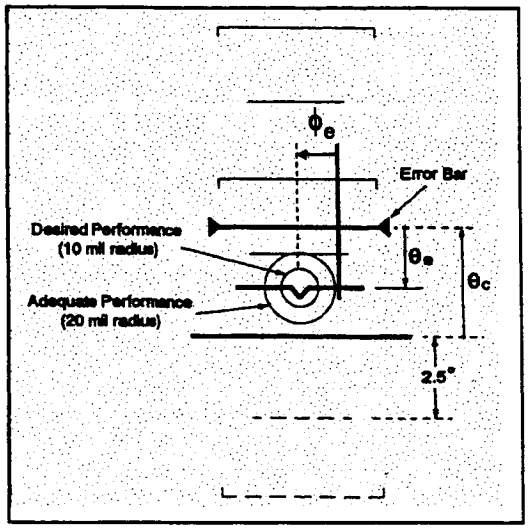


Figure 11. Head-Up Display with Pitch and Roll Tracking Task Symbology

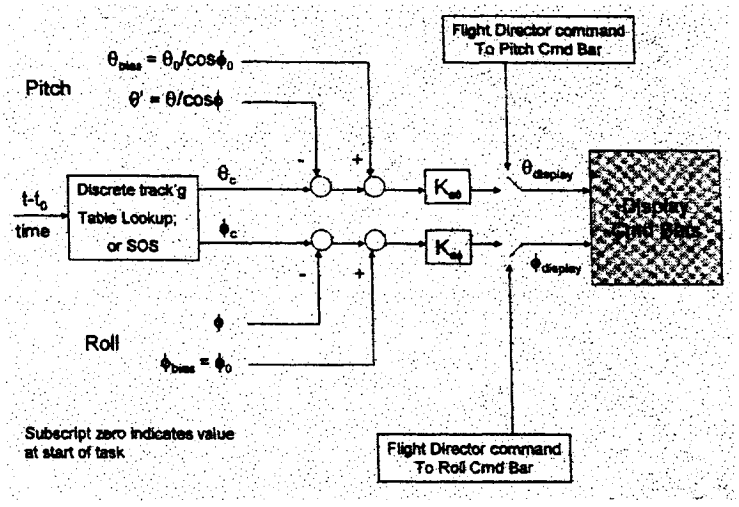


Figure 12. Tracking Task Command Bar Implementation in the BAE SYSTEMS Simulator

2.5.1.3 Data Requirements

Table 6 contains a list of variables that will acquired off the Flight Control Computer (FCC) bus by the LOCATS software. All data runs will be saved to the LOCATS host PC hard disk and later archived to CD.

Table 6. Minimum Set of Parameters for LOCATS Software

Parameter	Parameter Description	Units	Comment
Flight Condition and Aircraft Configuration Variables			
VCAS	Calibrated Airspeed	knots	
VT	True Airspeed	knots	
MACH	Mach Number		
H	Altitude	ft	Pressure altitude
CG	Center of Gravity	%MAC	
LBD	Localizer Beam Deviation	DDM	
GBD	Glideslope Beam Deviation	DDM	
Pilot Command Inputs			
FLON	Pitch Stick Force	lbs	
DLON	Pitch Stick Position	inches	
FLAT	Roll Stick Force	lbs	
DLAT	Roll Stick Position	Inches	
FP	Rudder Pedal Force	lbs	
DP	Rudder Pedal Position	deg	
DT	Throttle Lever Angle	deg	
Aircraft Control Surfaces			
DEC	Elevator Command	deg	
DE_LIB	Left Inboard Elevator Position	inches	Local actuator position
DE_LOB	Left Outboard Elevator Position	Inches	Local actuator position
DE_RIB	Right Inboard Elevator Position	Inches	Local actuator position
DE_ROB	Right Outboard Elevator Position	inches	Local actuator position
DAC_L	Left Aileron Command	deg	
DAC_R	Right Aileron Command	deg	
DA_L	Left Aileron Position	inches	Local actuator position
DA_R	Right Aileron Position	inches	Local actuator position
DRC	Rudder Command	deg	Lower
DR_L	Lower Rudder Position	inches	Local actuator position
DR_U	Upper Rudder Position	inches	Local actuator position
DFLAP	Flap Position	deg	Average
DSP	Spoiler Position	deg	Average
Aircraft Response Variables			
Q	Pitch Rate	deg/sec	
THE	Pitch Attitude	deg	

Parameter	Parameter Description	Units	Comment
P	Roll Rate	deg/sec	
PHI	Roll Attitude	rad	Units specified by airframer
R	Yaw Rate	deg/sec	
TPSI	True Heading	deg	ISM selected
ALPHA	Angle of Attack	Deg	
BETA	Angle of Sideslip	deg	Estimated
GAMMA	Flightpath Angle	deg	
AZ	Vertical Acceleration	ft/sec^2	
NZ	Load Factor	g	Estimated
NX_CG	Longitudinal Acceleration at CG	g	
NY_CG	Lateral Acceleration at CG	g	
NZ_CG	Normal Acceleration at CG	g	
HD	Altitude Rate	ft/min	Sink Rate
Flight Control System Variables			
GKP4	Pitch Stick Filter Time Constant		PRC mode
GKP5	Pitch Stick Filter Time Constant		PAC mode
GKP6	Pitch Stick Filter Time Constant		TO mode
GPFPT1	Pitch Stick Gain Shaping		
GPFPT2	Pitch Stick Gain Shaping		
GPFPT3	Pitch Stick Gain Shaping		
GKQ	Pitch Rate Feedback Gain		
Flight Control System Variables (Continued)			
GKP	Roll Rate Feedback Gain		
QCRL	Pitch Rate Command Limiter		
PCRL	Roll Rate Command Limiter		
P_Delay	Pitch Stick Delay	msec	
R_Delay	Roll Stick Delay	msec	

2.5.2 Procedures

In general, the piloted simulation will be conducted as a standard handling qualities evaluation. The sessions will begin with a pilot briefing that will include a discussion of the evaluation tasks and qualitative data requirements (i.e., comments and ratings). If necessary, the pilot will then be given a simulator demonstration with an emphasis on the features relevant to the evaluation tasks. Before the formal evaluation process begins, the pilot will be given an opportunity to practice the tasks.

A given evaluation will consist of four distinct variables: configuration, flight condition (speed, crosswinds, etc.), task, and loss of control scenario. A sample run log is presented in Table 7 that contains the run number, configuration identifier, flight condition identifier, flight control systen failure scenario, pilot ratings, and comments. When appropriate to the LOCATS evaluation, the pilot will be required to perform a given task at least twice before ratings and comments will be taken. In practice, the pilot will be able to repeat the tasks as many times as is necessary to provide ratings with confidence. Detailed commentary should be provided for each task, supplemented by an additional indication of the pilot's confidence in the rating(s) assigned. Identification of those situations demanding unusually high workload, high degree of singular focus on the task, unexpected aircraft responses, etc. are particularly important. Although there will be a recommended evaluation order for a given session, the experimenter will be free to make changes as needed.

For any given run a simulated flight control system failure (see Table 5) may be introduced at the discretion of the test conductor. These "random" events are designed to test the capabilities of the LOCATS.

2.6 LOCATS Evaluation Success Criteria

There are two fundamental factors that will determine the ultimate success of the LOCATS concept. First, the LOCATS software must be able to detect impending loss of control events while minimizing "false alarms" and/or missed events. Second, the time to detection must be minimized, so that the pilot or automatic flight control system can react to the situation. Thus for a given flight control system failure scenario, the following questions will be addressed:

- Can the wavelet-based computations run in a real-time environment?
- Was loss of control or impending loss of control observed?
- If yes, what was the triggering event?
- If yes, was the event detected by the LOCATS software?
- If no, was a non-event detected by the LOCATS software?
- What metric(s) provided the detection mechanism?
- What was the detection time?
- Can the detection time be improved?

Later versions of the LOCATS software may include recommended courses of action that could be provided to the pilot in the form of a flight director or as direct commands to flight control system.

Table 7. Example Run Log

Date: Pilot:

Run#	FC	Configuration	Task	FCS	Comments

3. LOCATS SIMULATION CHECKOUT

3.1 Summary

The LOCATS simulation checkout was conducted at the BAE SYSTEMS facility in Johnson City, NY from 23 to 27 June 2003. The objectives for the checkout were as follows:

- Install required software and otherwise configure the LOCATS host PC;
- Configure the SBS Technologies 1553 data bus card and integrate the card with LOCATS software;
- Assess and refine aircraft configurations and flight conditions;
- Collect diagnostic data for each aircraft configuration at its nominal flight condition (i.e., no crosswinds or turbulence)
- Evaluate and refine pilot tasks; and
- Assess loss of control scenarios.

Although there were several key issues to resolve, by the end of the five day checkout all objectives were met. Further details on aircraft configuration verification and loss of control scenario assessment are provided below.

3.2 LOCATS DATA ACQUISITION

A significant effort during the simulation checkout week was spent in creating the dynamic link library (DLL) that was required for the LOCATS software to access simulator data off of the 1553 data bus. Most of this effort was expended on proper configuration of the SBS Technologies 1553 data bus card. Although the card had a dual bus capability, it could only be configured to read data from one bus at a time despite the best efforts of STI and BAE SYSTEMS personnel. The good news was that the majority of the signals that STI requested be made available were located on a single data bus. In the end only six signals could not be recorded. The bad news was that these signals were those necessary to identify the pilot describing function from the SOS and discrete tracking tasks. When fully integrated with LOCATS at the end of the checkout session, the software could read, display, and record simulator data at 50 Hz.

3.3 Configuration Review

3.3.1 Run Log

As part of the simulation checkout diagnostic maneuvers were conducted on the ten aircraft configurations. A complete run log is provided in Table 8.

Table 8. Simulation Checkout Run Log

Run	Configuration (BAE SYSTEMS ID)	Description	Task
1	CHF (sti11)	Cruise/Heavy/FwdCG	Pitch Axis Frequency Sweep
2	CHF (sti11)	Cruise/Heavy/FwdCG	Pitch Axis Pulse
3	CHF (sti11)	Cruise/Heavy/FwdCG	Pitch Axis Doublet
4	CHF (sti11)	Cruise/Heavy/FwdCG	Roll Axis Frequency Sweep
5	CHF (sti11)	Cruise/Heavy/FwdCG	Roll Axis Pulse
6	CHF (sti11)	Cruise/Heavy/FwdCG	Roll Axis Doublet
7	CHA (sti12)	Cruise/Heavy/AftCG	Pitch Axis Frequency Sweep

Run	Configuration (BAE SYSTEMS ID)	Description	Task
8	CHA (sti12)	Cruise/Heavy/AftCG	Pitch Axis Pulse
9	CHA (sti12)	Cruise/Heavy/AftCG	Pitch Axis Doublet
10	CHA (sti12)	Cruise/Heavy/AftCG	Roll Axis Frequency Sweep
11	CHA (sti12)	Cruise/Heavy/AftCG	Roll Axis Pulse
12	CHA (sti12)	Cruise/Heavy/AftCG	Roll Axis Doublet
13	CLF (sti13)	Cruise/Light/FwdCG	Pitch Axis Frequency Sweep
14	CLF (sti13)	Cruise/Light/FwdCG	Pitch Axis Pulse
15	CLF (sti13)	Cruise/Light/FwdCG	Pitch Axis Doublet
16	CLF (sti13)	Cruise/Light/FwdCG	Roll Axis Frequency Sweep
17	CLF (sti13)	Cruise/Light/FwdCG	Roll Axis Pulse
18	CLF (sti13)	Cruise/Light/FwdCG	Roll Axis Doublet
19	CLA (sti14)	Cruise/Light/AftCG	Pitch Axis Frequency Sweep
20	CLA (sti14)	Cruise/Light/AftCG	Pitch Axis Pulse
21	CLA (sti14)	Cruise/Light/AftCG	Pitch Axis Doublet
22	CLA (sti14)	Cruise/Light/AftCG	Roll Axis Frequency Sweep
23	CLA (sti14)	Cruise/Light/AftCG	Roll Axis Pulse
24	CLA (sti14)	Cruise/Light/AftCG	Roll Axis Doublet
25	AHF (sti35)	Approach/Heavy/FwdCG	Pitch Axis Frequency Sweep
26	AHF (sti35)	Approach/Heavy/FwdCG	Pitch Axis Pulse
27	AHF (sti35)	Approach/Heavy/FwdCG	Pitch Axis Doublet
28	AHF (sti35)	Approach/Heavy/FwdCG	Roll Axis Frequency Sweep
29	AHF (sti35)	Approach/Heavy/FwdCG	Roll Axis Pulse
30	AHF (sti35)	Approach/Heavy/FwdCG	Roll Axis Doublet
31	AHA (sti36)	Approach/Heavy/AftCG	Pitch Axis Frequency Sweep
32	AHA (sti36)	Approach/Heavy/AftCG	Pitch Axis Pulse
33	AHA (sti36)	Approach/Heavy/AftCG	Pitch Axis Doublet
34	AHA (sti36)	Approach/Heavy/AftCG	Roll Axis Frequency Sweep
35	AHA (sti36)	Approach/Heavy/AftCG	Roll Axis Pulse
36	AHA (sti36)	Approach/Heavy/AftCG	Roll Axis Doublet
37	ALF (sti39)	Approach/Light/FwdCG	Pitch Axis Frequency Sweep
38	ALF (sti39)	Approach/Light/FwdCG	Pitch Axis Pulse
39	ALF (sti39)	Approach/Light/FwdCG	Pitch Axis Doublet
40	ALF (sti39)	Approach/Light/FwdCG	Roll Axis Frequency Sweep
41	ALF (sti39)	Approach/Light/FwdCG	Roll Axis Pulse

Run	Configuration (BAE SYSTEMS ID)	Description	Task
42	ALF (sti39)	Approach/Light/FwdCG	Roll Axis Doublet
43	ALA (sti3A)	Approach/Light/AftCG	Pitch Axis Frequency Sweep
44	ALA (sti3A)	Approach/Light/AftCG	Pitch Axis Pulse
45	ALA (sti3A)	Approach/Light/AftCG	Pitch Axis Doublet
46	ALA (sti3A)	Approach/Light/AftCG	Roll Axis Frequency Sweep
47	ALA (sti3A)	Approach/Light/AftCG	Roll Axis Pulse
48	ALA (sti3A)	Approach/Light/AftCG	Roll Axis Doublet
49	SALF (sti57)	Steep Approach/Light/FwdCG	Pitch Axis Frequency Sweep
50	SALF (sti57)	Steep Approach/Light/FwdCG	Pitch Axis Pulse
51	SALF (sti57)	Steep Approach/Light/FwdCG	Pitch Axis Doublet
52	SALF (sti57)	Steep Approach/Light/FwdCG	Roll Axis Frequency Sweep
53	SALF (sti57)	Steep Approach/Light/FwdCG	Roll Axis Pulse
54	SALF (sti57)	Steep Approach/Light/FwdCG	Roll Axis Doublet
55	SALA (sti58)	Steep Approach/Light/AftCG	Pitch Axis Frequency Sweep
56	SALA (sti58)	Steep Approach/Light/AftCG	Pitch Axis Pulse
57	SALA (sti58)	Steep Approach/Light/AftCG	Pitch Axis Doublet
58	SALA (sti58)	Steep Approach/Light/AftCG	Roll Axis Frequency Sweep
59	SALA (sti58)	Steep Approach/Light/AftCG	Roll Axis Pulse
60	SALA (sti58)	Steep Approach/Light/AftCG	Roll Axis Doublet

3.3.2 Example Diagnostic Maneuver Time Histories

As described previously, the DLL needed to read, display, and record simulator data through LOCATS was not completed until the end of the checkout session. A subset of the required data set from Table 6 was made available for storage using the BAE SYSTEMS file saving capabilities. It was wrongly assumed by STI personnel that the data would be recorded at the 50 Hz update rate available over the FCS data bus. The data, however, were actually stored in a print file format at an update rate of approximately 8 Hz. To further complicate matters key output variables were saved in a separate file from the input variables that had a unique time stamp and update rate. Example pitch and roll axis frequency sweep time histories are shown in Figure 13 and Figure 14, respectively. Note that the coarseness of the data is clearly evident in the pitch (THE) and roll (PHI) attitude traces. To remedy this problem, diagnostic maneuvers were repeated during the formal evaluation simulator sessions (see Section 3) for all of the configurations used in the LOCATS assessment.

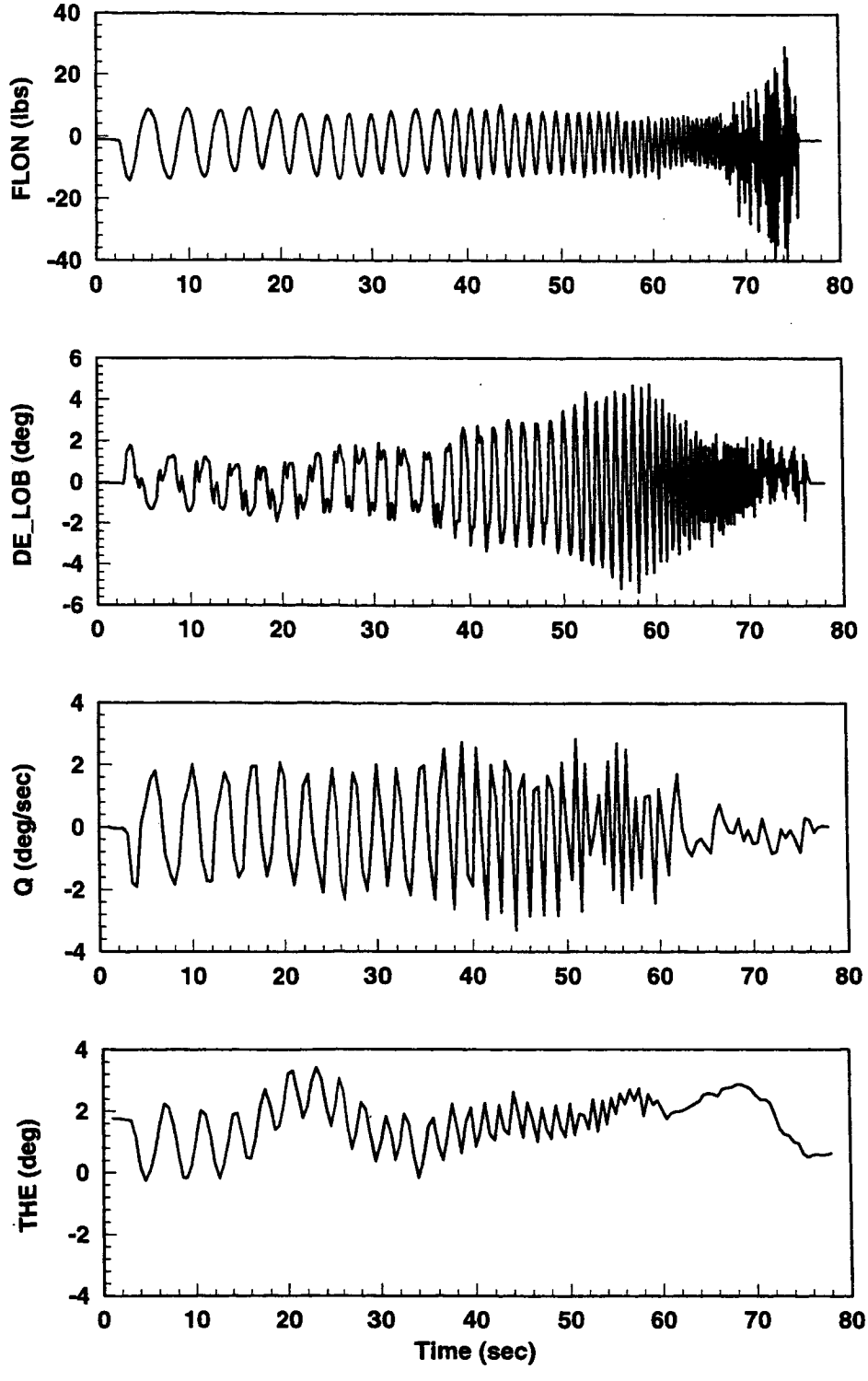


Figure 13. Example Pitch Axis Frequency Sweep Time Histories for Configuration CHF

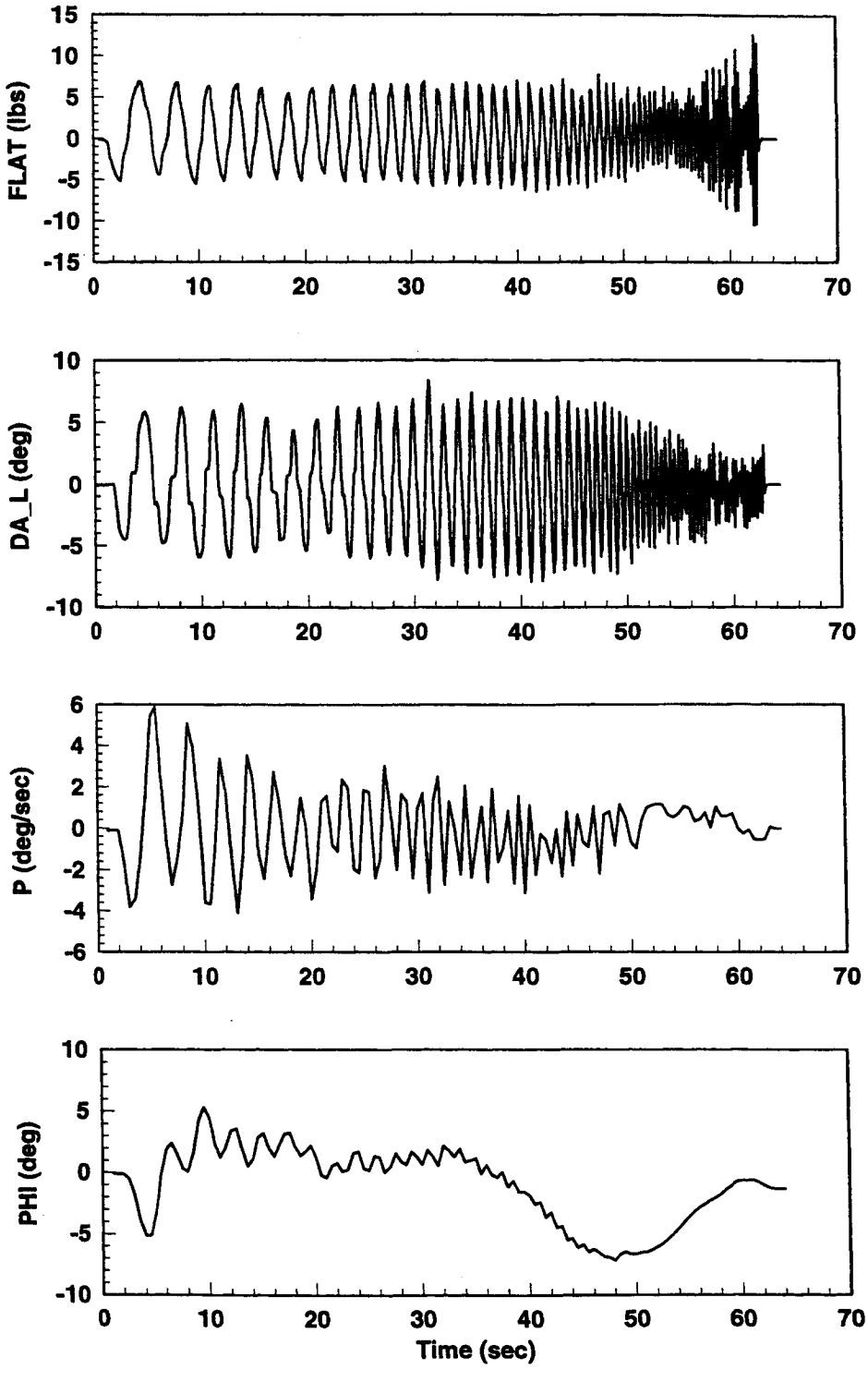


Figure 14. Example Roll Axis Frequency Sweep Time Histories for Configuration CHF

3.4 Assessment of Maneuvers and Simulated Flight Control Failures

3.4.1 Maneuvers

Going into the checkout session it was assumed that the SOS and discrete tracking tasks would provide the high gain environment needed to induce pilot-vehicle system loss of control in the form of pilot-induced oscillations. There was considerable concern among the BAE SYSTEMS personnel that the tasks were beyond the capabilities of the transport aircraft modeled in the simulator. It was to the good fortune of this program that the STI evaluation pilot had spent hours in the simulator flying similar HUD tracking tasks as part of his dissertation research. In the end the aircraft was capable of performing the SOS tasks with no adjustments to the task. For the discrete tracking tasks the pitch axis command magnitudes were left unchanged, while the roll axis commands were reduced by 25% to avoid some maneuvering restrictions. BAE SYSTEMS personnel included an option in the simulator that allowed the tracking tasks to be flown as either single or dual axis. Selection of the type of task was made via a toggle switch.

The pitch and bank attitude capture and hold tasks (PACH and BACH) also worked out quite well. What is especially effective about these single axis tasks is the ability to modulate the aggressiveness of the task. Once the pilot is comfortable with a configuration, it is possible to fly the task essentially open-loop with well timed and sized inputs. It was, however, possible to avoid this situation by having the pilot aggressively attempt to minimize the capture errors, thus ensuring a higher gain, closed-loop task. Reducing the capture attitudes also kept the pilot in the loop.

The flightpath capture and hold and altitude rate capture and hold tasks were not as interesting. Once the maneuver was initiated, there was significant dwell times, while the pilot waited for the large aircraft to climb and descend. The control activity of the pilot was clearly not at the same level as the HUD tracking or PACH and BACH maneuvers.

Significant effort was expended evaluating the offset landing tasks. One problem in particular arose from the limitations of the simulator visual scene. It was just not possible to create the cues necessary to fly the task as defined in Ref. 1. The BAE SYSTEMS personnel did an impressive job of adjusting the localizer and glideslope beam deviation flight director bars to mimic the offset landing task. The maneuver began with a fairly low gain beam capture task and ended with a lagged offset correction via the flight director that, in the final analysis, could not replicate the required correction to centerline.

3.4.2 Failures

Flight control failures were introduced into the simulator at the discretion of the operator using the simulator change-a-gain feature. This feature allowed flight control system parameters to be varied from a default value to some new value, while the simulator was operating. The change-a-gain feature provided a choice of eight parameter values including the default value. Because of the interest in inducing loss of control, only the most severe changes from the default value were used. As identified in Table 5 eight flight control system changes were selected for the checkout phase. The default and changed values for each flight control system change are identified in Table 9. The change-a-gain parameters listed in the table were included in the data sets recorded by LOCATS. Thus the actual changed value for a given run could be easily verified.

To exercise the change-a-gain option three typed entries had to be made at the operator station. This entry procedure did not pose a problem for single parameter changes, since the change did not engage until the final entry was made. For changes that involved multiple entries, however, it was difficult for the operator to make the required entries in a timely matter. As a result the PitchGS, which had three parameters or nine required entries, was not used.

After the evaluation pilot was comfortable with the evaluation tasks, the various flight control system failure scenarios were investigated. As expected the pitch (QGain) and roll (PGain) rate feedback gain

failures worked well with the intended tasks. It was observed that the pitch axis was more sensitive to this augmentation failure then was the roll axis.

Table 9. Change-a-Gain Values for Simulated Flight Control System Failures

No.	ID	Default Value	Changed Value
1	PitchTC	0 sec	0.25 sec
2	PitchGS	Not Used	Not Used
3	PitchTD	0 msec	240 msec
4	RollTD	0 msec	240 msec
5	QGain	0.4	0
6	PGain	1.0	0
7	QLIM	25 deg/sec	7.5 deg/sec
8	PLIM	120 deg/sec	36 deg/sec

The default pitch control system had no stick filter. Thus, the change associated with the PitchTC essentially added a command path lag to the system. The addition of this lag was noticed by the pilot although the effects were not dramatic.

The next set of changes that were assessed was the reduction of the command path limiters. Because it is difficult to generate sustained pitch rates with the simulated transport aircraft, the pitch command limiter had no real effect, even at its lowest value. Reduction of the roll command path limiter, on the other hand, was quite effective, particularly during the discrete tracking task.

The final flight control system changes that were evaluated were the pitch and roll stick added time delays. Because the baseline aircraft can appear sluggish in aggressive maneuvering, the added time delays did not pose a significant problem for the pilot. He seemed to compensate for the added delay with relative ease. Although this result was somewhat unexpected, it has been difficult in the past to induce unfavorable pilot-vehicle system coupling in a fixed base simulator with time delay alone. Including the added time delay in combination with other failures produced much more promising results.

3.5 Evaluation Pilot Background

Dr. Ed Bachelder of STI served as the evaluation pilot for both the simulation checkout and the formal LOCATS evaluation sessions. As a Naval Aviator Dr. Bachelder was a Helicopter Aircraft Commander for the SH-60B Seahawk. During the Navy flight training syllabus he flew the T-34C turboprop and the TH-57B/C. He logged a total of 926 hours while on active duty. In addition, he logged approximately 2000 hours in a helicopter simulator, largely performing hover tracking tasks, while conducting his doctoral research work at MIT.

4. LOCATS SIMULATION EVALUATION

4.1 Summary

4.1.1 Overview

The formal LOCATS simulation evaluation was conducted at the BAE SYSTEMS facility in Johnson City, NY from 28 July to 1 August 2003. The objectives for the evaluation were as follows:

- Install latest versions of the LOCATS software on the LOCATS host PC;
- Conduct diagnostic maneuvers on all aircraft configurations used in the formal evaluations;
- Verify that the wavelet-based algorithms can run in real time;
- Evaluate the ability of LOCATS to detect pilot-vehicle system changes; and
- Assess the ability of the LOCATS metrics to predict loss of control.

The results of the LOCATS evaluation are documented in Volume I of this report.

4.1.2 Pilot Evaluation Tasks

The pilot evaluation tasks used in the LOCATS evaluation were as follows:

- Pitch Attitude Capture and Hold (PACH);
- Bank Angle Capture and Hold (BACH);
- Pitch and Roll Sum-of-Sines Tracking (SOS); and
- Pitch and Roll Discrete Tracking (DT).

The offset landing tasks were not used in the formal evaluation because of limited effectiveness given the constraints of the implementation. The altitude rate, altitude, flightpath, and ILS capture and hold tasks were discarded because the long periods of essentially open-loop control were not conducive to loss of control.

4.1.3 Revised Simulated Flight Control System Failures

Table 10 provides a revised list of simulated flight control system failures based on the results of the simulation checkout. In addition, a new scenario was introduced involving the roll axis. As the formal evaluations progressed it became clear that loss of control in the roll axis (i.e., PIO or otherwise) could not be induced with the available flight control system changes. To help "loosen" up the roll axis, a new change-a-gain option was made available that eliminated the sideslip rate feedback in the roll axis via the GBDEST gain. Eliminating this gain alone did not produce a noticeable change to the pilot. When used in conjunction with the elimination of the roll rate feedback, however, it was possible to generate at a minimum mild PIO in the roll axis. To coordinate its use in the evaluation process, this gain change was only used in conjunction with a PGain change. As mentioned in the previous section, multiple key strokes are required to make a single change. Because it was not possible to add the GBDEST to the saved signal list, the BGain change was always made first, followed by the PGain.

Table 11 gives an updated list of the change-a-gain values used in the LOCATS formal evaluation. Included in this list are the values for the BGain changes.

Table 10. Revised Simulated Flight Control System Failures

No.	ID	Description	Demonstration Maneuvers
1	PitchTC	Alter pitch stick filter time constant via change-a-gain	All pitch axis and dual axis tasks
2	PitchTD	Inject pitch stick time delay	DT, SOS, PACH
3	RollTD	Inject roll stick time delay	DT, SOS, BACH
4	QGain	Alter pitch rate feedback via change-a-gain	All pitch axis and dual axis tasks
5	PGain	Alter roll rate feedback via change-a-gain	All roll axis and dual axis tasks
6	BGain	Alter GBDEST via change-a-gain (only used in conjunction with PGain)	All roll axis and dual axis tasks
7	QLIM	Reduce pitch rate command path limiter	DT, SOS, PACH
8	PLIM	Reduce roll rate command path limiter	DT, SOS, BACH

Table 11. Updated Change-a-Gain Values for Simulated Flight Control System Failures

No.	ID	Default Value	Changed Value
1	PitchTC	0 sec	0.25 sec
2	PitchTD	0 msec	240 msec
3	RollTD	0 msec	240 msec
4	QGain	0.4	0
5	PGain	1.0	0
6	BGain	1.2	0.2
7	QLIM	25 deg/sec	7.5 deg/sec
8	PLIM	120 deg/sec	36 deg/sec

4.2 Run Log

A run log for the five simulator sessions conducted as part of the LOCATS formal evaluation is provided in Table 12. During the formal evaluation of the LOCATS system 122 analysis runs were completed. Of these 41 were diagnostic maneuvers that are used to characterize the configurations used in the formal evaluation. The diagnostic maneuvers included pitch and roll axis doublets and pitch and roll axis frequency sweeps. The maneuvers were conducted with autothrottle on and off. Throttle frequency sweeps were conducted with the autopilot on and off. The remaining 81 maneuvers featured the flight control system failure scenarios defined in Table 10. Because of the limitations of the types of failures that could be introduced via the change-a-gain feature, the aircraft could not be made to depart. It was possible, however, to repeatedly produce significant pilot-induced oscillations as indicated in the "Comments" column of Table 12.

Table 12. LOCATS Simulation Evaluation Run Log

Run#	FC	Configuration	Task	FCS	Comments
28 July 2003 (Autothrottle on for all runs)					
R1	C	CLF	SOS	n/a	Saved data ok. DLL error identified and fixed.
R2	C	CLF	SOS	n/a	Pitch axis only. Update rate error identified and fixed.
R3	C	CLF	SOS	n/a	Roll axis only. No data.
R4	C	CLF	SOS	n/a	Pitch axis only. Pilot was more aggressive in 2nd half of run.
R5	C	CLF	BACH	n/a	Checkout.
R6	C	CLF	SOS	n/a	Roll axis only.
R7	C	CLF	SOS	QGain	Pitch axis only.
R8	C	CLF	SOS	PGain	Roll axis only.
29 July 2003 (Autothrottle on for all runs)					
R10	C	CHF	Diagnostic	n/a	Roll Doublet
R11	C	CHF	Diagnostic	n/a	Pitch Doublet
R12	C	CHF	BACH	PGain	
R13	C	CHF	BACH	RollTD	
R14	C	CHF	BACH	PLIM	
R15	C	CHF	DT	n/a	
R16	C	CHF	DT	PGain	
R17	C	CHF	DT	RollTD	
R18	C	CHF	DT	PLIM	
R19	C	CHF	SOS	PGain	Roll axis only, PGain=1.75
R20	C	CHF	SOS	PGain	Roll axis only, PGain=0
R21	C	CHF	SOS	PLIM	
R22	C	CHF	SOS	RollTD	
R23	C	CHF	PACH	QGain	
R24	C	CHF	PACH	PitchTC	
R25	C	CHF	PACH	PitchTD	
R26	C	CHF	DT	PitchTD	
R27	C	CHF	DT	PitchTD + RollTD	Dual axis delays
R28	C	CLA	BACH	BGain + PGain	

Run#	FC	Configuration	Task	FCS	Comments
R29	C	CLA	BACH	BGain + PGain + RollTD	
R30	C	CLA	DT	BGain + PGain	
R31	C	CLA	DT	BGain + PGain + RollTD	
R32	CM	CLA	BACH	BGain + PGain	
R33	CM	CLA	DT	BGain + PGain	
R34	CM	CLA	SOS	BGain + PGain	
R35	CM	CLA	SOS	RollTD	BGain=0.2, PGain=0 at start
R36	CM	CLA	PACH	QGain	Authothrottle disengaged
R37	CM	CLA	PACH	QGain	
R38	CM	CLA	PACH	QGain	PitchTD whole run, autothrottle disengage triggers sustained PIO
R39	C	CLA	Diagnostic	n/a	Pitch doublet, LOCATS timing test – strip charts only
R40	C	CLA	Diagnostic	n/a	Pitch doublet, LOCATS timing test – strip charts + FFT
R41	C	CLA	Diagnostic	n/a	Pitch doublet, LOCATS timing test – strip charts + FFT
R42	C	CLA	Diagnostic	n/a	Pitch doublet, LOCATS timing test – strip charts + FFT + TVTF
R43	C	CLA	Diagnostic	n/a	Pitch doublet, LOCATS timing test – strip charts + FFT + TVTF + 10 persistent lines
30 July 2003 (Autothrottle on for all runs)					
R50	C	CHF	Diagnostic	n/a	Pitch Sweep
R51	C	CHF	Diagnostic	n/a	Roll Sweep
R52	C	CLA	Diagnostic	n/a	Roll Doublet
R53	C	CLA	Diagnostic	n/a	Roll Sweep
R54	C	CLA	Diagnostic	n/a	Pitch Doublet
R55	C	CLA	Diagnostic	n/a	Pitch Sweep
R56	C	CLA	BACH	BGain + PGain	RollTD whole run
R57	C	CLA	BACH	BGain + PGain	RollTD whole run, extra aggressive pilot technique

Run#	FC	Configuration	Task	FCS	Comments
R58	C	CLA	DT	QGain	PitchTD + RollTD whole run, mild PIO after failure
R59	C	CLA	DT	QGain + PGain	PitchTD + RollTD whole run
R60	CM	CLA	Diagnostic	n/a	Wavelet test, 8192 pts, some computational slow down
R61	CM	CLA	Diagnostic	n/a	Wavelet test, 4096 pts, no computational slow down
R62	CM	CLA	Diagnostic	n/a	Wavelet test, 6144 pts, no computational slow down
R63	CM	CLA	Diagnostic	n/a	Wavelet + FFT test, no computational slow down
R64	CM	CLA	BACH	BGain + PGain	
R65	CM	CLA	BACH	BGain + PGain	20 degree captures, extra aggressive
R66	CM	CLA	BACH	BGain + PGain	RollTD whole run
R67	CM	CLA	BACH	BGain + PGain	RollTD whole run, extra aggressive, mild PIO
R68	CM	CLA	PACH	n/a	No change
R69	CM	CLA	PACH	QGain	PitchTD whole run, PIO
R70	CM	CLA	DT	QGain + BGain + PGain	Mild pitch PIO
R71	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, pitch PIO
R72	A	ALA	Diagnostic	n/a	Roll Doublet
R73	A	ALA	Diagnostic	n/a	Roll Doublet repeat
R74	A	ALA	Diagnostic	n/a	Roll Sweep
R75	A	ALA	Diagnostic	n/a	Pitch Doublet
R76	A	ALA	Diagnostic	n/a	Pitch Sweep
R77	A	ALA	BACH	n/a	
R78	A	ALA	BACH	BGain + PGain	
R79	A	ALA	BACH	BGain + PGain	RollTD whole run, Large overshoots
R80	AX	ALA	BACH	n/a	
R81	AX	ALA	BACH	BGain + PGain	

Run#	FC	Configuration	Task	FCS	Comments
R82	AX	ALA	BACH	BGain + PGain	RollTD whole run, mild PIO
31 July 2003					
R90	C	CLA	Diagnostic	n/a	Pitch Doublet, autothrottle off
R91	C	CLA	Diagnostic	n/a	Pitch Sweep, autothrottle off
R92	CM	CLA	PACH	QGain	Autothrottle on
R93	CM	CLA	PACH	QGain	Autothrottle off Runs 93-106
R94	CM	CLA	PACH	QGain	PitchTD whole run, significant PIO
R95	CM	CLA	PACH	QLIM	PIO after failure
R96	CM	CLA	PACH	QLIM	PitchTD whole run, some PIO
R97	CM	CLA	SOS	QGain	
R98	CM	CLA	SOS	QGain	PitchTD whole run, more oscillatory, no real PIO
R99	CM	CLA	DT	QGain	Mild PIO
R100	CM	CLA	DT	QGain	PitchTD whole run, large overshoot following failure, mild PIO
R101	C	CLA	Diagnostic	n/a	Roll Doublet
R102	C	CLA	Diagnostic	n/a	Roll Sweep
R103	CM	CLA	BACH	PGain	Not a problem for pilot
R104	CM	CLA	BACH	BGain + PGain	
R105	CM	CLA	BACH	BGain + PGain	RollTD whole run, large overshoots, mild PIO
R106	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, multi-axis failures
R107	CH	CLA	Diagnostic		Pitch Doublet, autothrottle on Runs 107-110
R108	CH	CLA	Diagnostic		Pitch Sweep
R109	CH	CLA	Diagnostic		Roll Doublet
R110	CH	CLA	Diagnostic		Roll Sweep
R111	CH	CLA	Diagnostic	n/a	Pitch Doublet, autothrottle off Runs 111-128
R112	CH	CLA	Diagnostic	n/a	Pitch Sweep
R113	CH	CLA	Diagnostic	n/a	Roll Doublet
R114	CH	CLA	Diagnostic	n/a	Roll Sweep
R115	CH	CLA	PACH	QGain	
R116	CH	CLA	PACH	QGain	PitchTD whole run, PIO

Run#	FC	Configuration	Task	FCS	Comments
R117	CH	CLA	PACH	QLIM	PitchTD whole run, PIO
R118	CH	CLA	SOS	n/a	Pitch axis only
R119	CH	CLA	SOS	QGain	Pitch axis only, PitchTD whole run, PIO
R120	CH	CLA	BACH	PGain	
R121	CH	CLA	BACH	BGain + PGain	
R122	CH	CLA	BACH	BGain + PGain	RollTD whole run, mild PIO in middle or run
R123	CH	CLA	SOS	n/a	Roll axis only
R124	CH	CLA	SOS	BGain + PGain	Roll axis only, RollTD whole run, sustained oscillations
R125	CH	CLA	DT	BGain + PGain	Some PIO
R126	CH	CLA	DT	QGain	Data lost
R127	CH	CLA	DT	QGain	PitchTD and RollTD whole run, pitch PIO
R128	CH	CLA	DT	BGain + PGain	PitchTD and RollTD whole run, oscillatory, roll PIO
1 August 2003					
R130	C	CLA	Diagnostic	n/a	Pitch doublet, autothrottle off Runs 130-134, QGain=0
R131	C	CLA	Diagnostic	n/a	Repeat of R130, extended time
R132	C	CLA	Diagnostic	n/a	Pitch Sweep, QGain=0
R133	C	CLA	Diagnostic	n/a	Roll Doublet, PGain=0
R134	C	CLA	Diagnostic	n/a	Roll Sweep, PGain=0
R135	C	CLA	Diagnostic	n/a	Throttle Sweep with autopilot
R136	C	CLA	Diagnostic	n/a	Throttle Sweep with autopilot off
R137	C	CLA	PACH	QGain	PitchTD from start

4.3 Diagnostic Maneuvers

4.3.1 Pilot Control Stick Characteristics

The CLA (autothrottle on) frequency sweep runs were used to identify the dynamic characteristics of the simulator's control stick. Coinciding with comments made by the evaluation pilot, it was found that the pitch stick characteristics were essentially linear, while the roll stick characteristics were modified by common control system nonlinearities. Details are provided below.

4.3.1.1 Pitch Stick

The pitch stick characteristics were identified from R55. Partial stick force and position time histories are shown in Figure 15 for the first 20 seconds of the frequency sweep. Here, the slowly increasing frequency sinusoidal stick force input is matched by a sinusoidal stick position output. No significant nonlinear characteristics are evident in the responses. The Figure 16 frequency response was generated using the fast Fourier transform (FFT) tools included in the STI Matlab-based wavelet toolbox. Note the excellent coherence from 0.3 rad/sec to nearly 30 rad/sec and the apparent second order response within this range. From this observed characteristic a second order transfer function fit was made to the frequency response as shown in Figure 17. Although the control stick dynamics are in most cases represented by a second order response, the result here indicates a lower than anticipated breakpoint frequency (i.e., 13.5 rad/sec) that will have a significant phase lag contribution to the overall aircraft to stick force response.

4.3.1.2 Roll Stick

The roll stick characteristics were identified from R33. Partial stick force and position time histories are shown in Figure 18 for the first 20 seconds of the frequency sweep. In this case, the slowly increasing frequency sinusoidal stick force input is not matched by a sinusoidal stick position output. By reviewing the characteristics of common flight control system nonlinearities as defined in Ref. 13, hysteresis is observed in the "flattened" response as the stick position output crosses zero, and some additional nonlinearity is producing a "clipped" response at the peak output amplitudes. Ignoring these observed nonlinearities at first, the Figure 19 frequency response was generated. The coherence was, generally speaking, good in the frequency range of interest, but this measure of the linear correlation between the input and outputs signals is clearly not as good as that seen in Figure 16 for the pitch stick. A predominately second order response is again observed and the resulting fit is shown in Figure 20. Here a second order fit matches the amplitude response well, but there is a nearly constant phase offset throughout the frequency range of interest.

To explain the phase mismatch of Figure 20, the nonlinear characteristics were investigated further, beginning with the cross plot of stick position to stick force as shown in Figure 21. This plot was generated using the nearly 90 seconds of essentially constant amplitude stick force and position responses of R53. Here, the common preload plus friction response is clearly revealed. An idealized response represented by the thick, gray line indicates that the characteristic is not quite symmetric. By estimating the 2b and F_s parameters from Figure 21, the negative inverse describing function for this nonlinearity (see Figure 22, revised from Ref. 14) is used to estimate the added phase lag resulting from the nonlinearities. In this case the resulting phase lag is in the neighborhood of 20 degrees (the "noisy" data of Figure 21 prevents a more accurate estimation). Finally, the transfer function fit is adjusted to account for this added phase lag, the result of which is shown in Figure 23. When accounting for the nonlinearity there is also a resulting gain attenuation. This adjustment in gain was accounted for in the linear fit, so no further adjustment was required. Because the effects of this nonlinearity are sensitive to changes in input amplitude, a unique fit is not possible. As with the pitch stick characteristics, the relatively low breakpoint frequency (i.e., 16 rad/sec) will also have a significant phase lag contribution to the overall aircraft to stick force response.

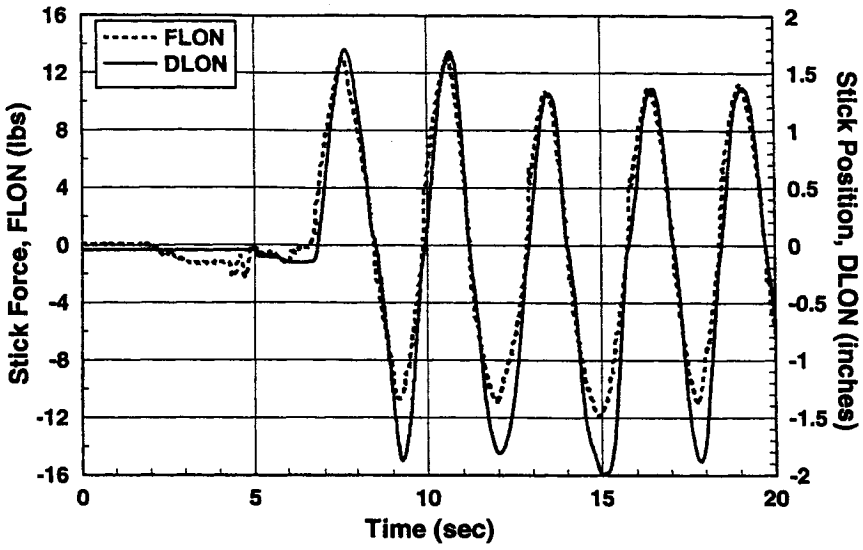


Figure 15. Partial Pitch Stick Force and Position Time Histories from R55 Frequency Sweep

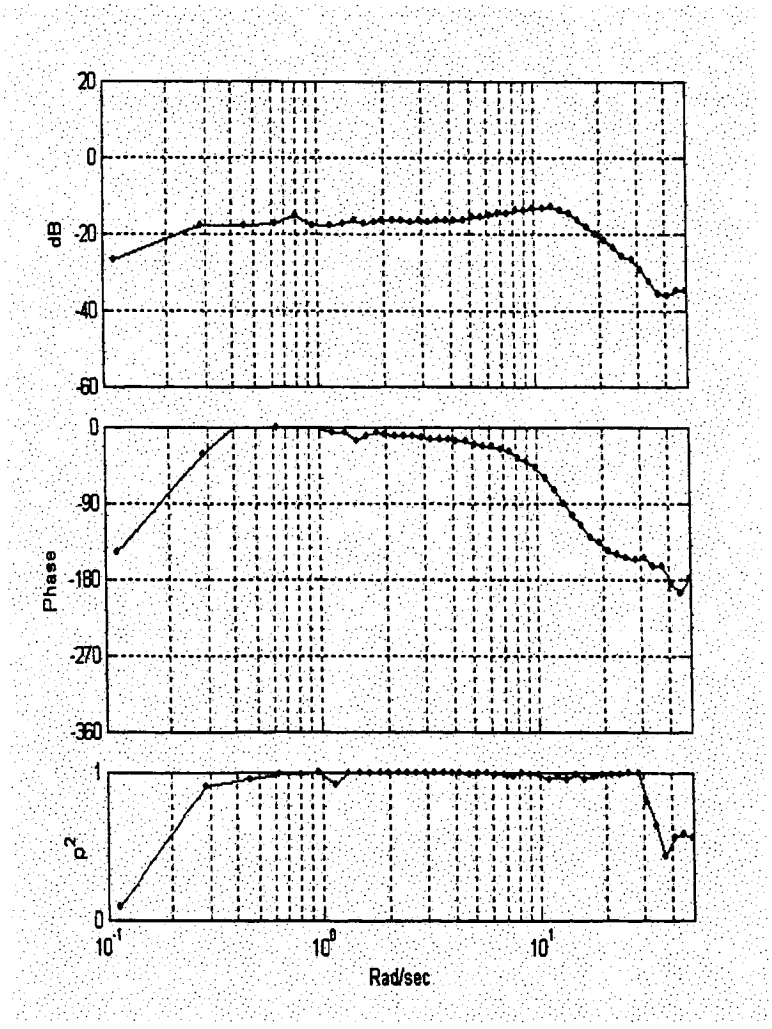
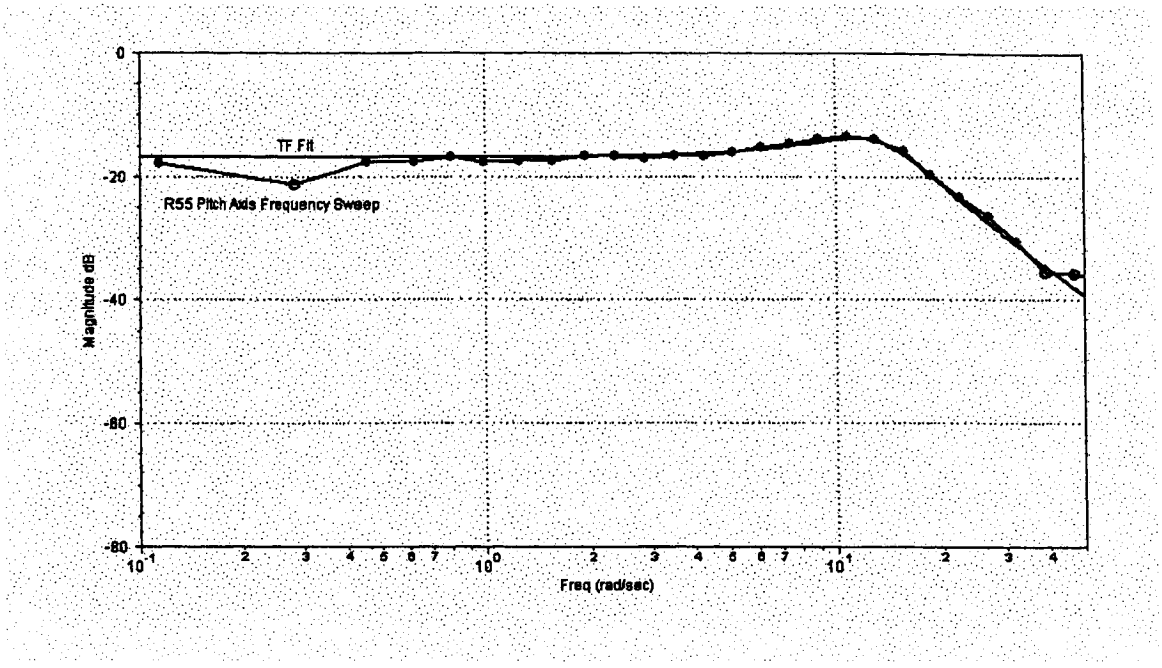


Figure 16. Run 55 CLA DLON/FLON Frequency Response



$$\frac{DLON}{FLON} = \frac{26}{[0.38, 13.5]}$$

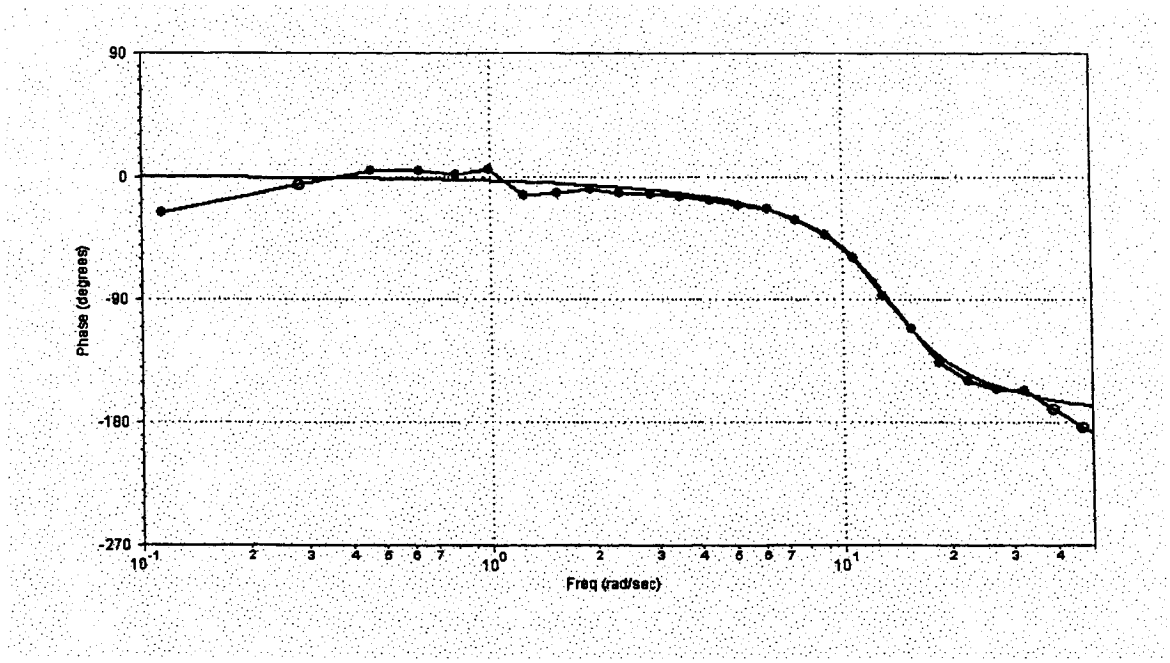


Figure 17. Transfer Function Fit to Pitch Stick Frequency Response

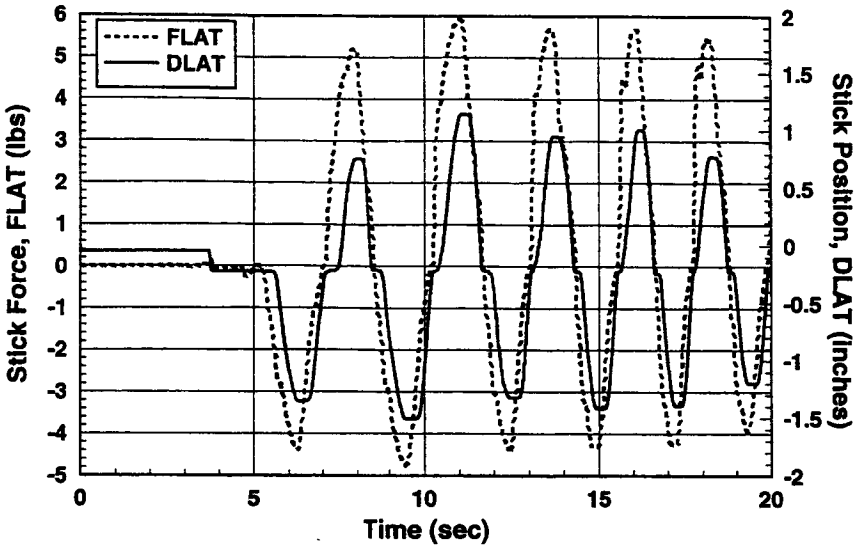


Figure 18. Partial Roll Stick Force and Position Time Histories from R53 Frequency Sweep

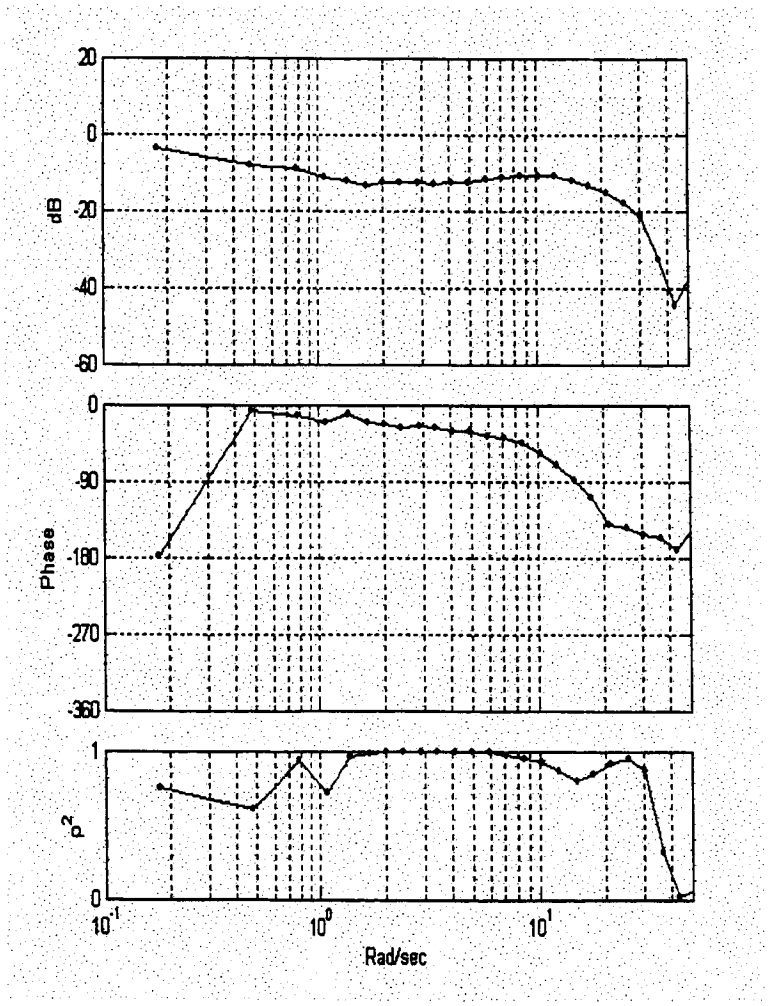
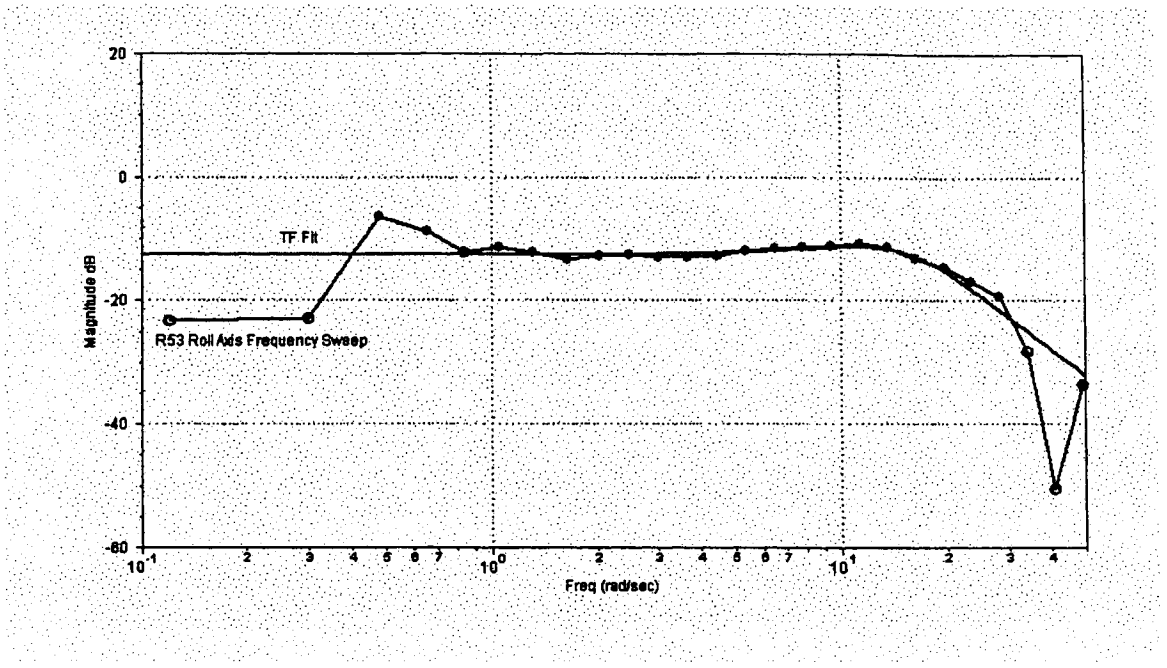


Figure 19. Run 53 CLA DLAT/FLAT Frequency Response



$$\frac{DLAT}{FLAT} = \frac{60}{[0.5,16]}$$

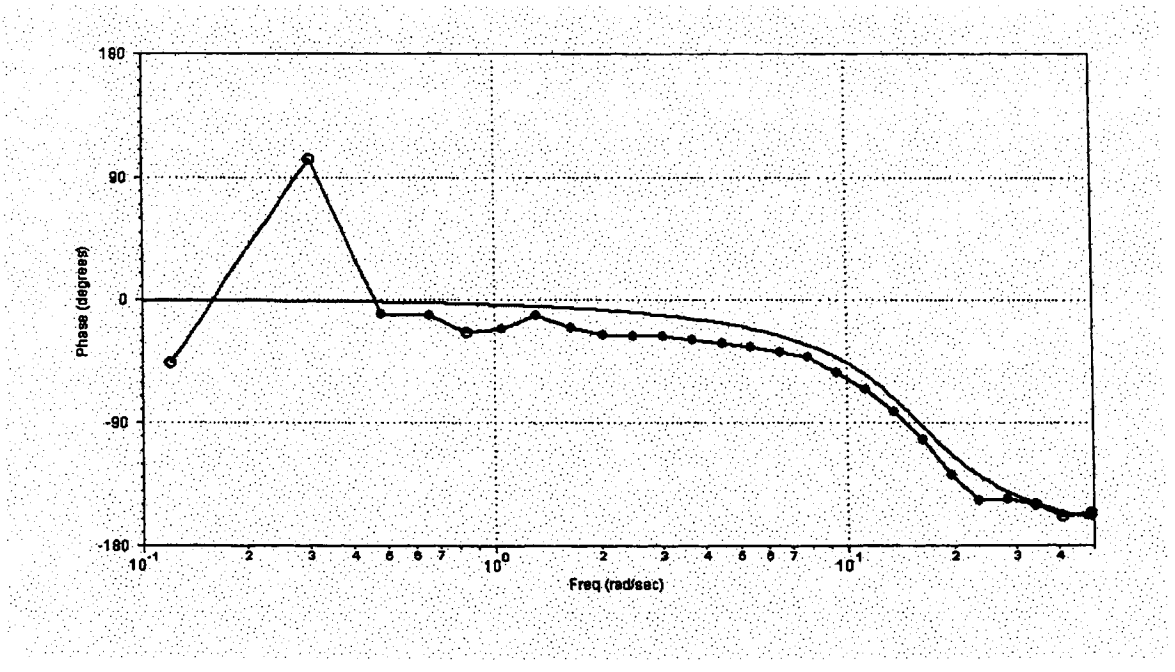


Figure 20. Transfer Function Fit to Roll Stick Frequency Response

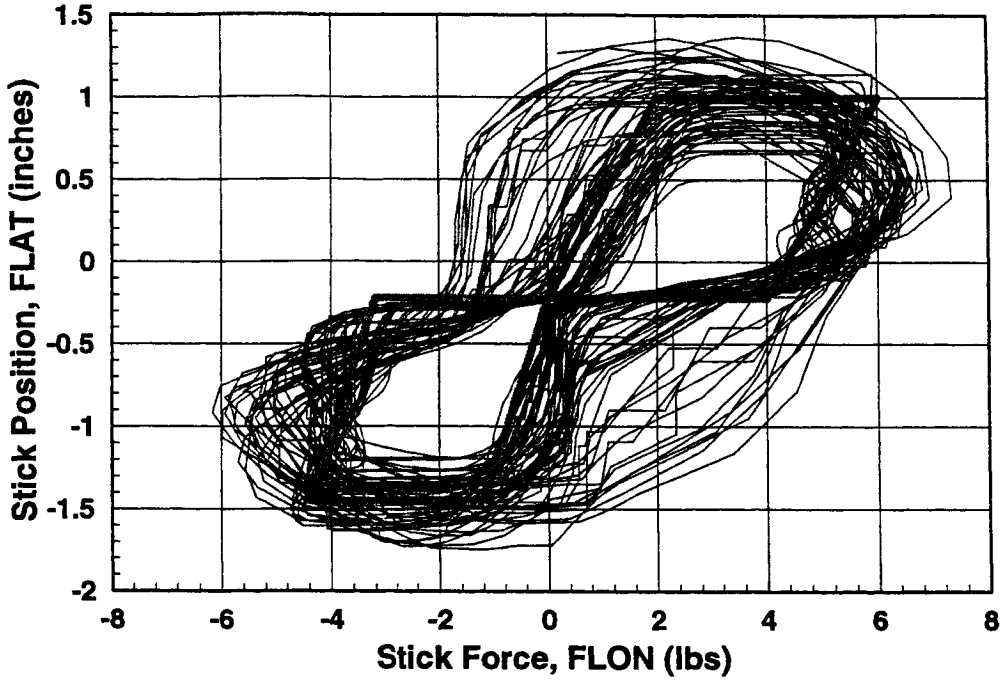


Figure 21. Observed Roll Stick Friction (Hysteresis) Plus Preload Characteristic

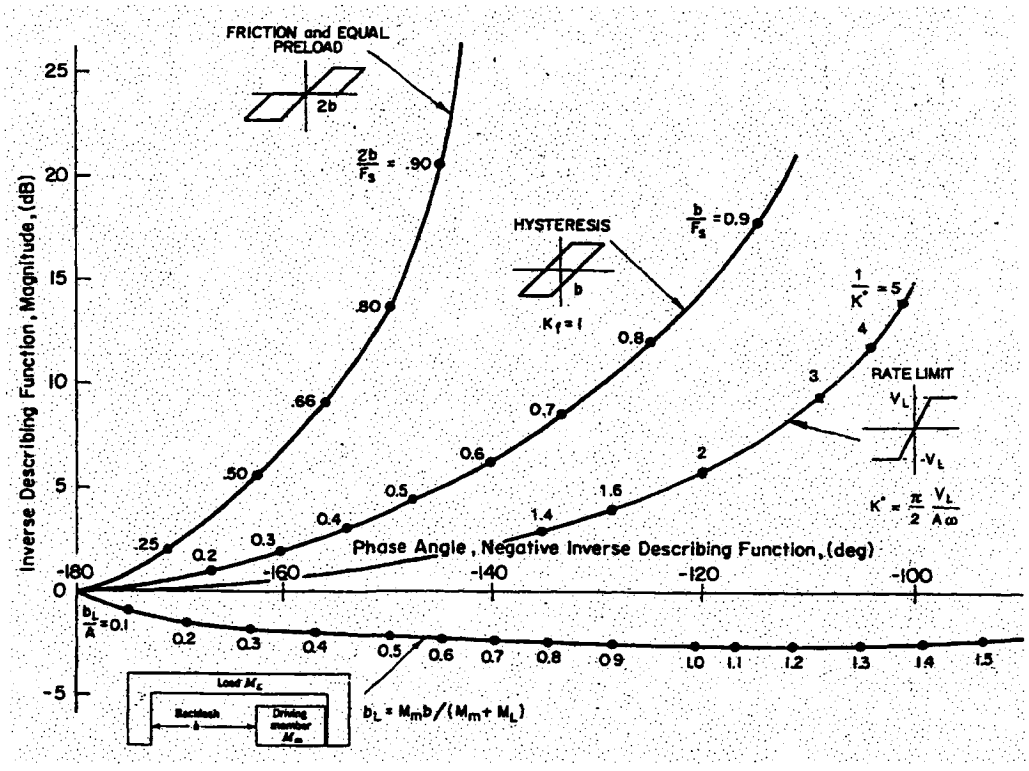


Figure 22. Inverse Describing Functions for Common Flight Control System Nonlinearities

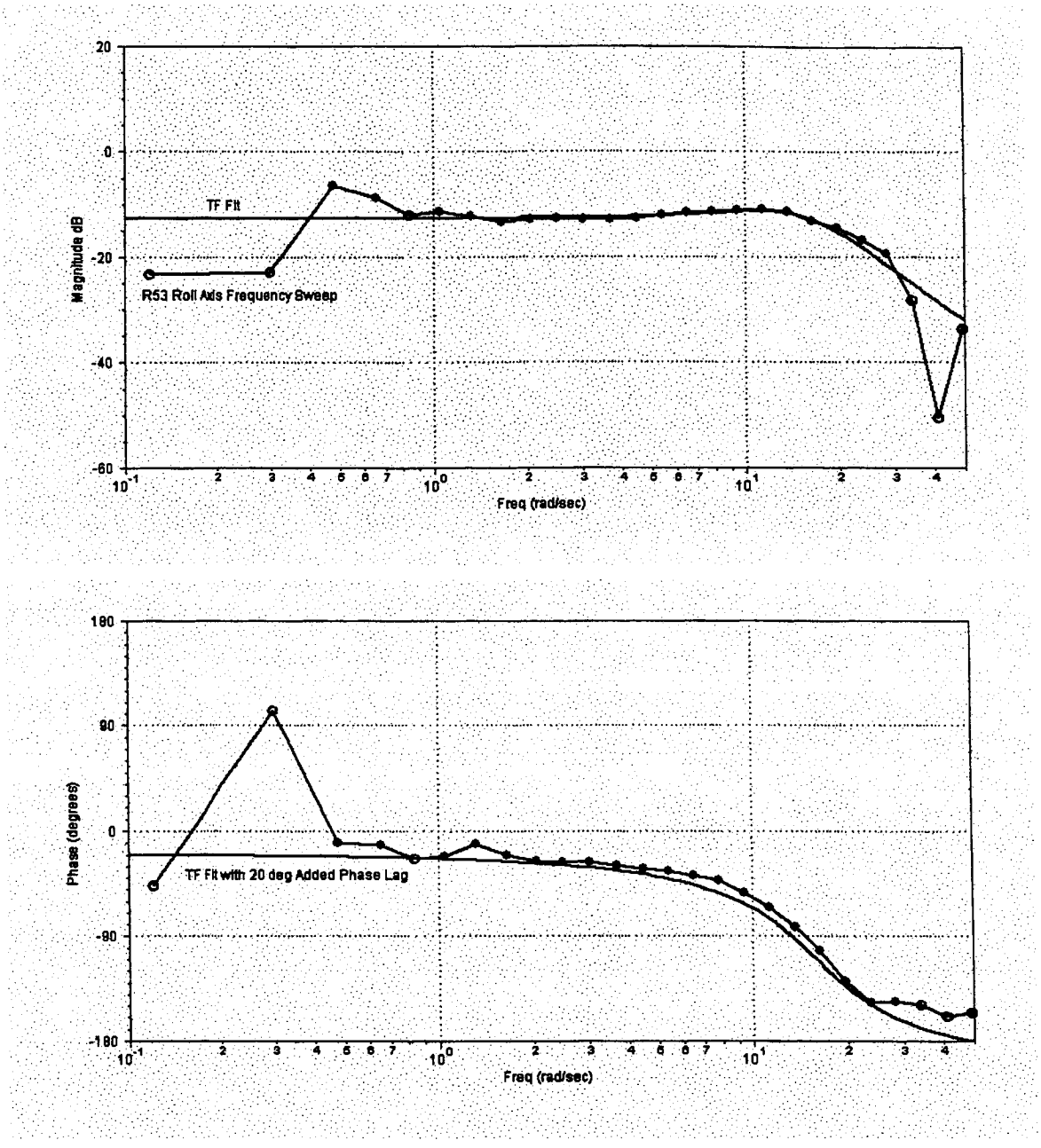


Figure 23. Revised Transfer Function Fit to Roll Stick Frequency Response (20 degrees of added phase lag due to friction plus preload nonlinearity)

4.3.2 Configuration Frequency Responses and Airplane Bandwidth Assessments

As identified in Table 12 diagnostic maneuvers consisting of pitch and roll axis frequency sweeps and pitch and roll axis doublets were completed for each aircraft configuration flown as part of the LOCATS formal assessment. Attitude to stick force frequency response Bode plots were generated from the frequency sweep runs using the fast Fourier transform techniques included in the Matlab version of the STI wavelet analysis toolbox. Included with these plots was the corresponding coherence, a measure of how well the input and output signals are linearly correlated. It is desirable to have coherence values above 0.66 in terms of confidence in the resulting transfer function correlation.

The resulting plots were then used to obtain airplane bandwidth parameters, as defined in Volume I of this report, for each configuration. In the analysis described herein, results were not accepted for runs where the coherence fell below this threshold in the critical frequency region where the bandwidth measures are made.

4.3.2.1 Longitudinal Configurations

As indicated in Table 13, the baseline longitudinal aircraft configurations from which flight control system failures were introduced were flown with and without the autothrottle. An additional frequency sweep run was also made for the failed pitch rate feedback case, the most commonly introduced longitudinal flight control system failure. Time histories and corresponding frequency response plots from the frequency sweep runs were generated for each configuration. Each set of time histories includes longitudinal stick force (FLON) and position (DLON), elevator command (DEC), averaged elevator position (DE), pitch rate (Q), and pitch attitude (THE). Note that for all cases the pilot was able to maintain a nearly constant stick force amplitude until the highest frequencies. The computed frequency responses include a Bode magnitude and phase plot and the corresponding coherence measure.

Airplane bandwidth parameters were computed for each pitch attitude to stick force (THE/FLON) frequency response. The results are tabulated in Table 13. The actual bandwidth frequency is the lowest of the phase and gain bandwidth values. Note that for all of the configurations, the bandwidth frequency results from the gain value. In terms of pitch tracking handling qualities, the configurations have a Level 2 bandwidth frequency. When the phase delay, a measure of the high frequency phase roll off, is taken into consideration, only the augmented autothrottle off configurations (R91 and R112) remain in the Level 2 region. The increased augmentation associated with the autothrottle on runs produce over 100 msec of additional phase delay.

Table 13. Pitch Axis (THE/FLON) Airplane Bandwidth Parameters

Run	Configuration	Time Series/ Freq. Response	Phase ω_{BW} (rad/sec)	Gain ω_{BW} (rad/sec)	$\omega_{180°}$ (rad/sec)	τ_P (sec)
R50	CHF (autothrottle on)	Figure 24 & Figure 25	3.340	1.409	5.127	0.332
R55	CLA (autothrottle on)	Figure 26 & Figure 27	3.186	2.036	4.925	0.314
R91	CLA (autothrottle off)	Figure 28 & Figure 29	2.925	2.005	4.515	0.192
R112	CHLA (autothrottle off)	Figure 30 & Figure 31	1.758	1.439	3.740	0.170
R132	CLA, QGain = 0 (autothrottle off)	Figure 32 & Figure 33	1.697	1.523	2.094	0.231

All of the configurations suffer from large phase delay values, regardless of the effect of the autothrottle. The control stick analysis detailed in the previous section identified one source of the significant phase delay. In pitch the control stick transfer function is characterized by a 2nd order mode at 13.5 rad/sec. To investigate the impact of this effective command path lag, a pitch attitude to stick position (THE/DLON) frequency response was generated for comparison (see Figure 34). In this case the computed bandwidth parameters yield a phase delay of 0.232 sec, 100 msec lower than the phase delay produced from the corresponding THE/FLON frequency response. A 60 msec reduction in phase delay was achieved with the R91 autothrottle off case. There were not enough data runs available to fully investigate why this value is not the same as the autothrottle on case, but nonetheless the added phase delay from the stick is significant.

4.3.2.2 Lateral Configurations

Frequency sweeps runs were generated for each of the baseline lateral aircraft configurations from which flight control system failures were introduced. An additional frequency sweep run was also made for the failed roll rate feedback case, the most commonly introduced lateral flight control system failure. Time histories and corresponding frequency response plots from frequency sweep runs were generated for each configuration. Because of the impact of the preload/hysteresis nonlinearity described in the previous section, poor coherence resulted from sweeps in which the pilot's input amplitude did not exceed the force threshold, thus rendering a number of diagnostic runs unusable. Each set of time histories includes lateral stick force (FLAT) and position (DLAT), aileron command (DAC), aileron position (DA), roll rate (P), and roll attitude (PHI). As with the longitudinal configurations the pilot was able to maintain a nearly constant stick force amplitude until the highest frequencies. Once again the computed frequency responses include a Bode magnitude and phase plot and the corresponding coherence measure.

Computed bandwidth parameters are provided in Table 14 for those runs that produced high coherence data. For these configurations the gain and phase bandwidths were close in value with the phase bandwidth being lower in two of the three cases. The phase delay values were also similar in value. As would be expected, no phase delay effect from the longitudinal axis autothrottle augmentation is observed in the lateral axis. In terms of roll tracking handling qualities, the configurations have a Level 1 bandwidth frequency, however the high phase delay values move the configurations into the Level 3 region. In addition to the significant nonlinearities, the analysis of the lateral stick dynamics yielded a 2nd order pole at 16 rad/sec. A frequency response of roll attitude to stick position was generated for the R51 frequency sweep (see Figure 41). The resulting phase delay was in this case 0.178 sec, 80 msec less than the value for PHI/FLAT shown in Table 14. Thus, a significant portion of the phase delay comes from the stick dynamics.

Table 14. Roll Axis (PHI/FLAT) Airplane Bandwidth Parameters

Run	Configuration	Time Series/ Freq. Response	Phase ω_{BW} (rad/sec)	Gain ω_{BW} (rad/sec)	$\omega_{180°}$ (rad/sec)	τ_p (sec)
R51	CHF (autothrottle on)	Figure 35 & Figure 36	1.04	1.14	1.74	0.259
R53	CLA (autothrottle on)	Figure 37 & Figure 38	1.416	1.356	2.039	0.247
R102	CLA (autothrottle off)	Figure 39 & Figure 40	low coherence data	1.019	1.708	0.274

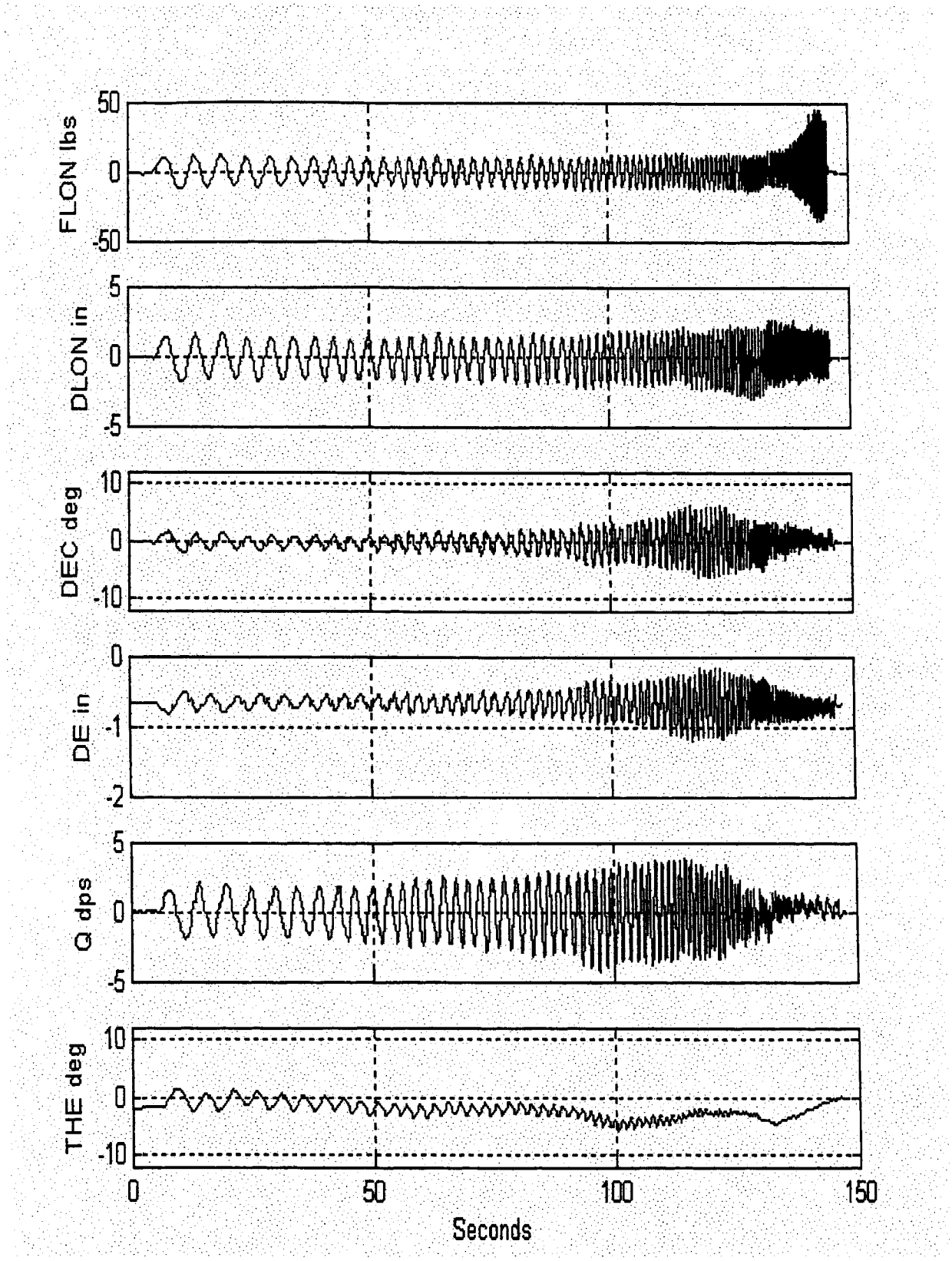


Figure 24. Run 50 CHF Pitch Sweep Time Histories (Autothrottle On)

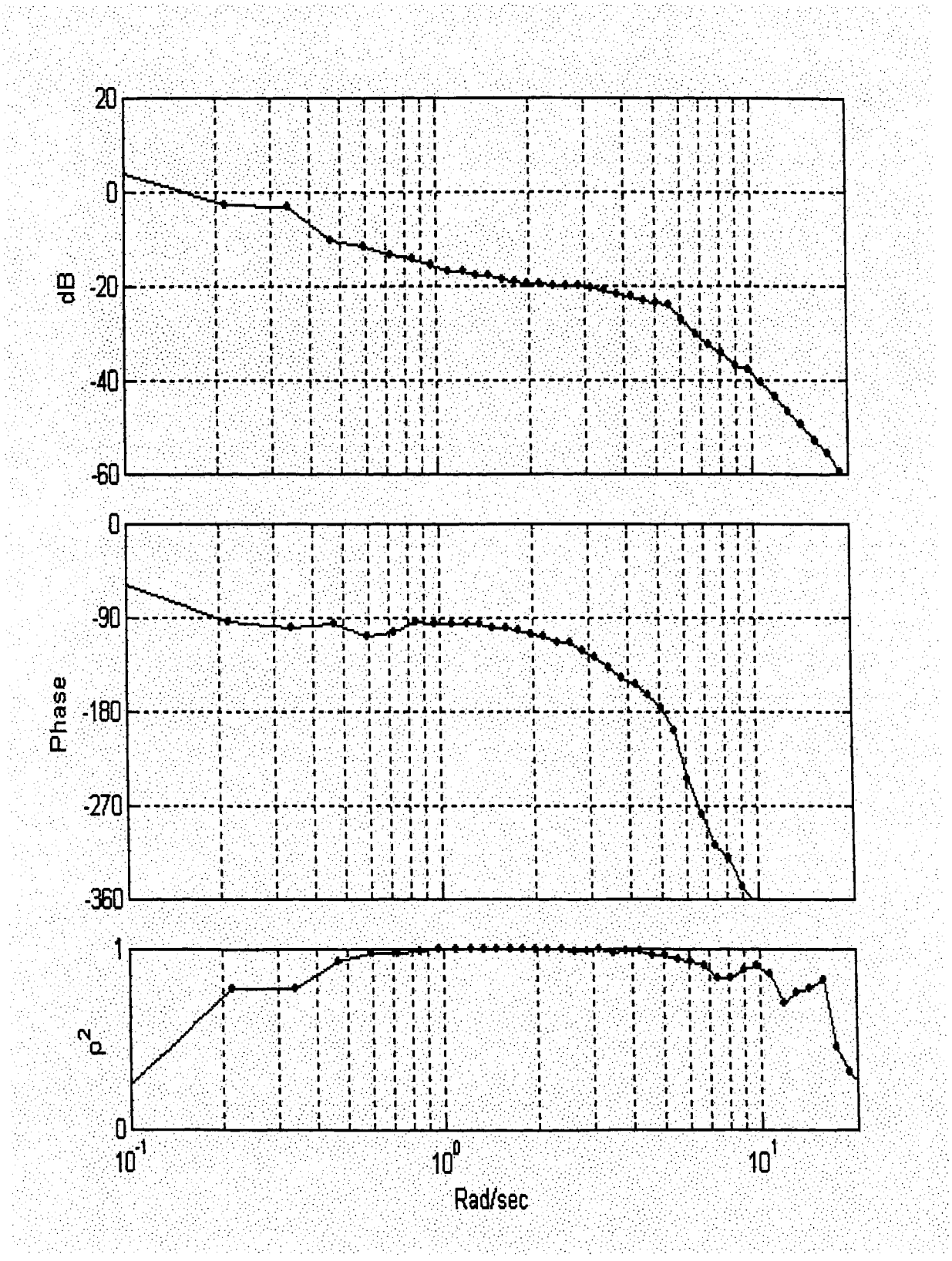


Figure 25. Run 50 CHF THE/FLON Frequency Response (Autothrottle On)

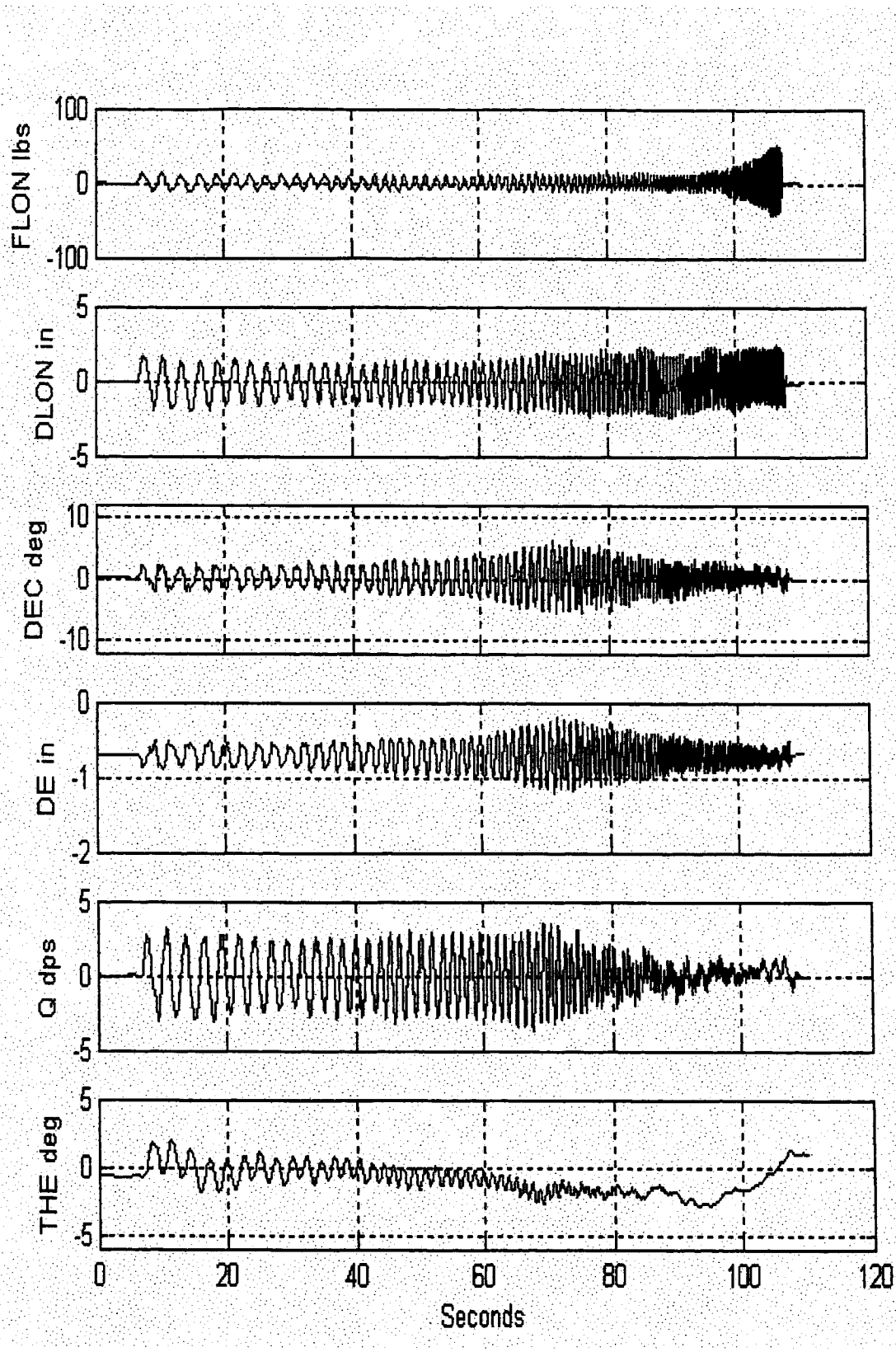


Figure 26. Run 55 CLA Pitch Sweep Time Histories (Autothrottle On)

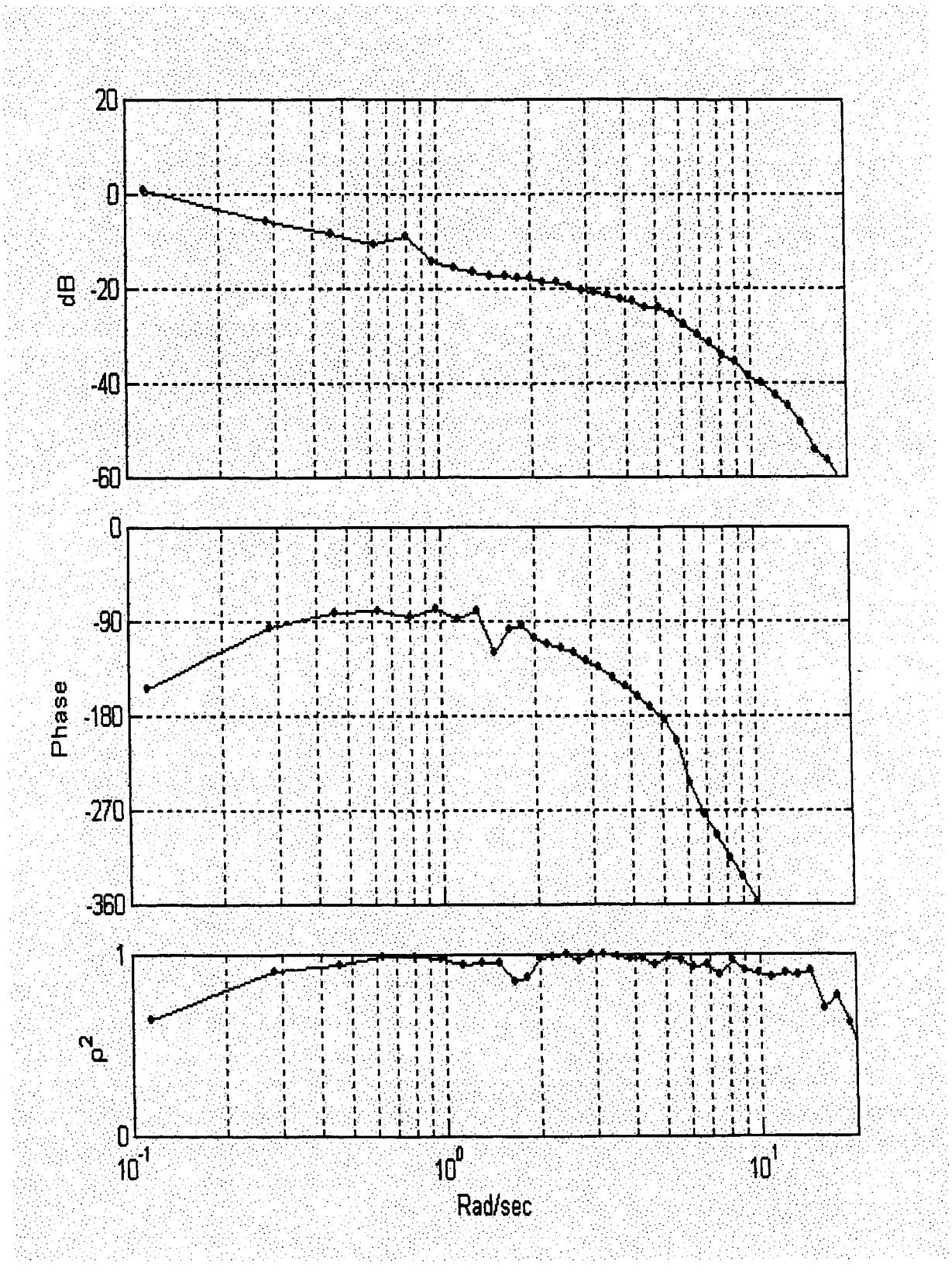


Figure 27. Run 55 CLA THE/FLON Frequency Response (Autothrottle On)

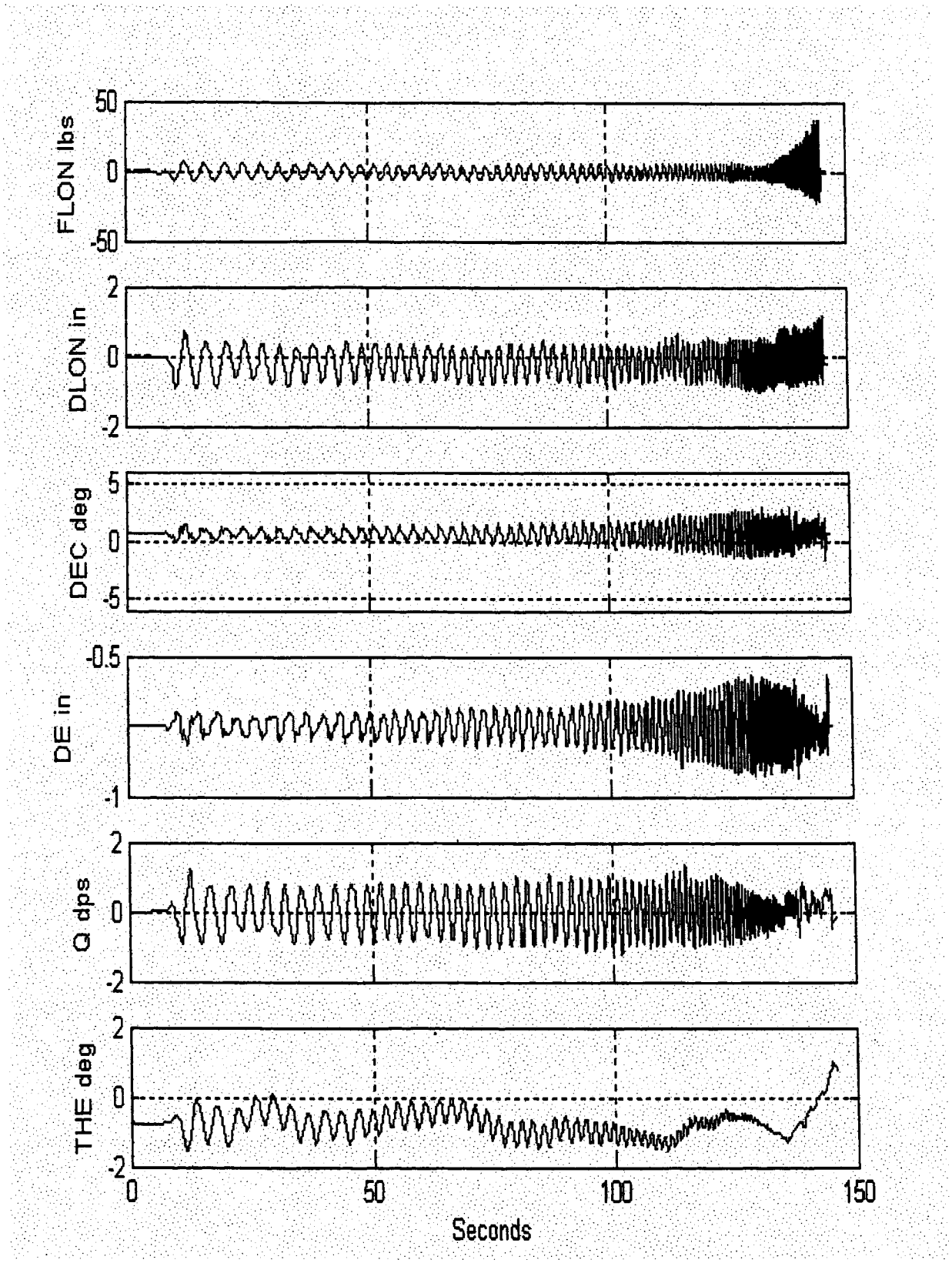


Figure 28. Run 91 CLA Pitch Sweep Time Histories (Autothrottle Off)

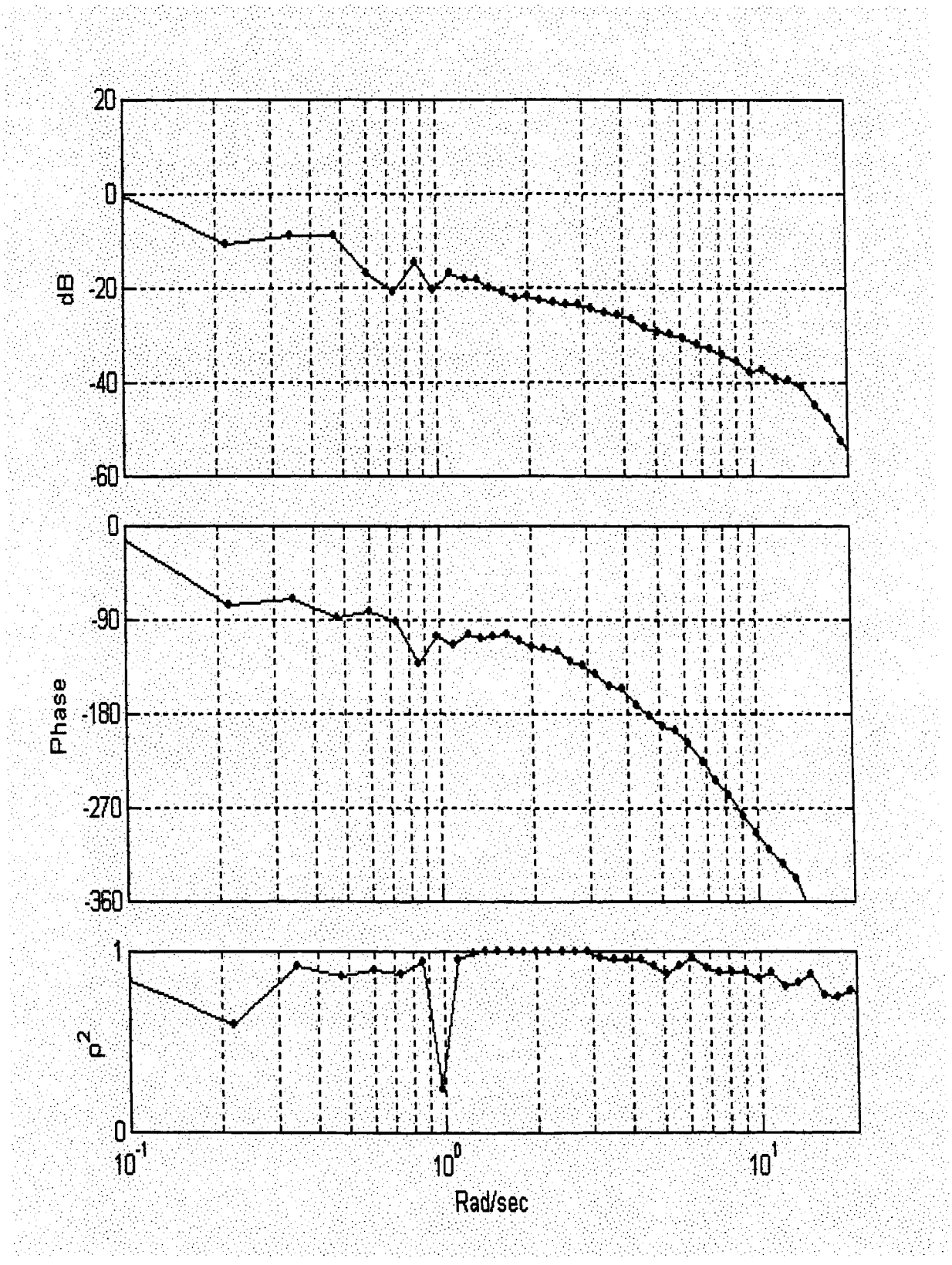


Figure 29. Run 91 CLA THE/FLON Frequency Response (Autothrottle Off)

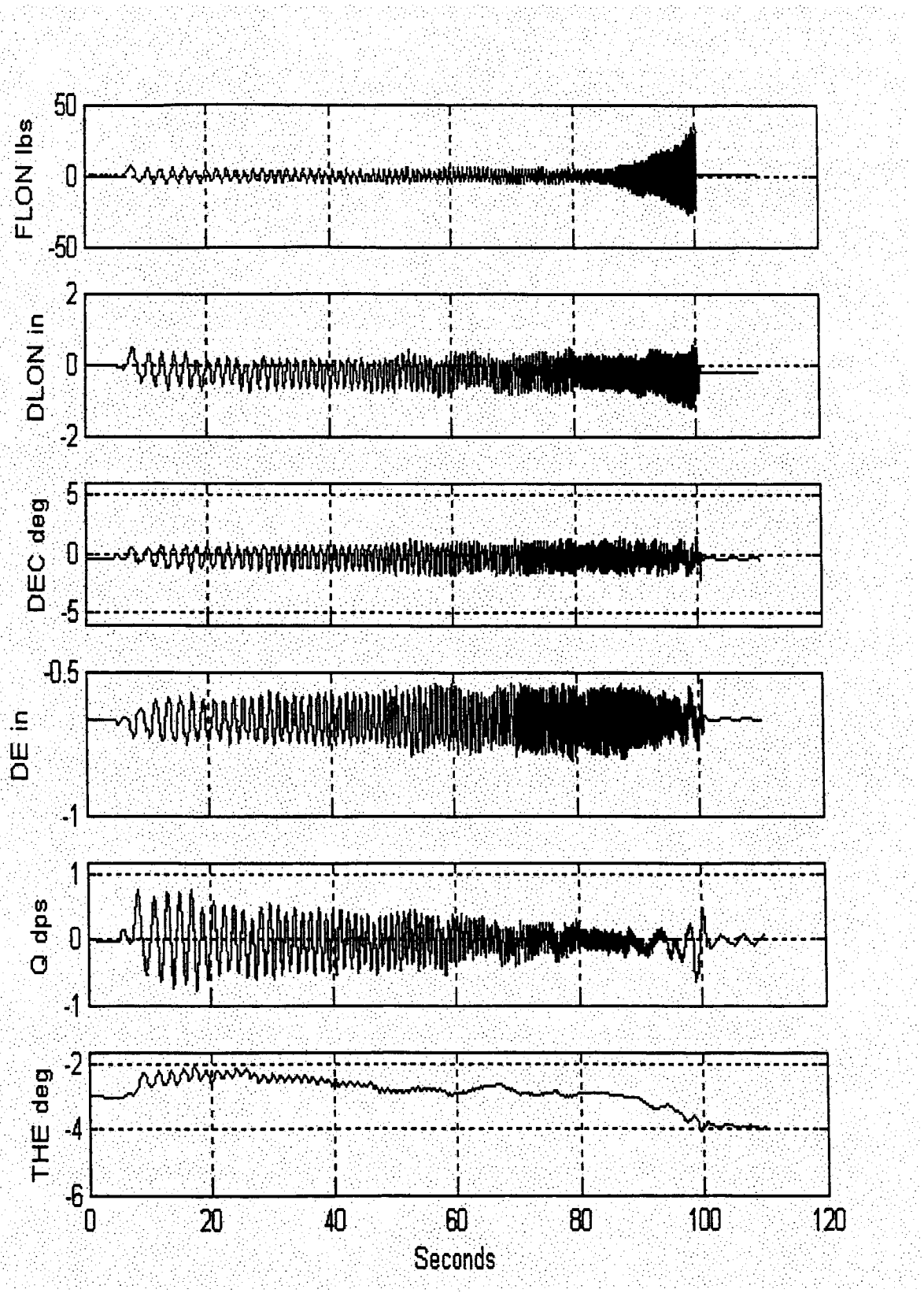


Figure 30. Run 112 CHLA Pitch Sweep Time Histories (Autothrottle Off)

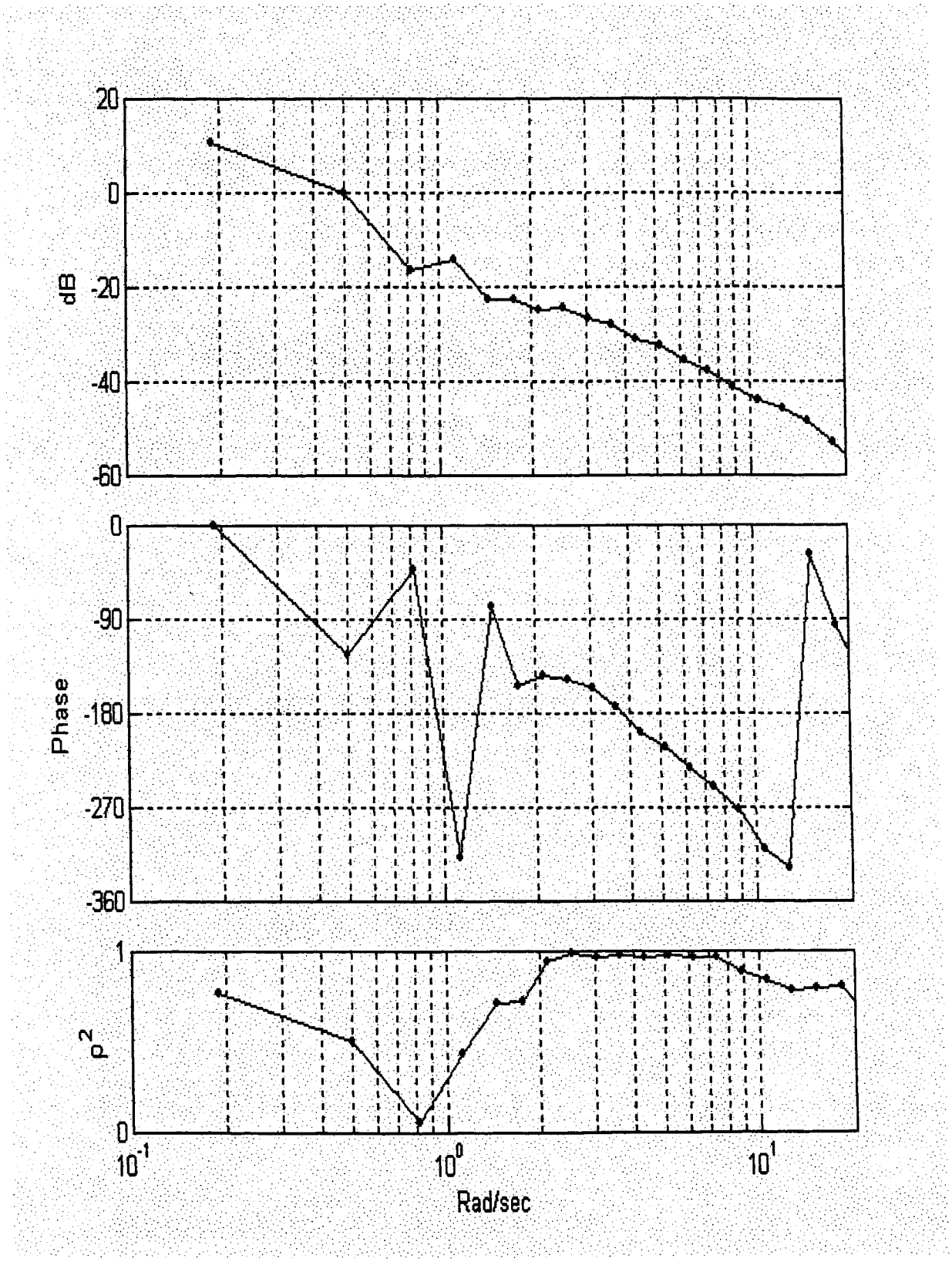


Figure 31. Run 112 CHLA THE/FLON Frequency Response (Autothrottle Off)

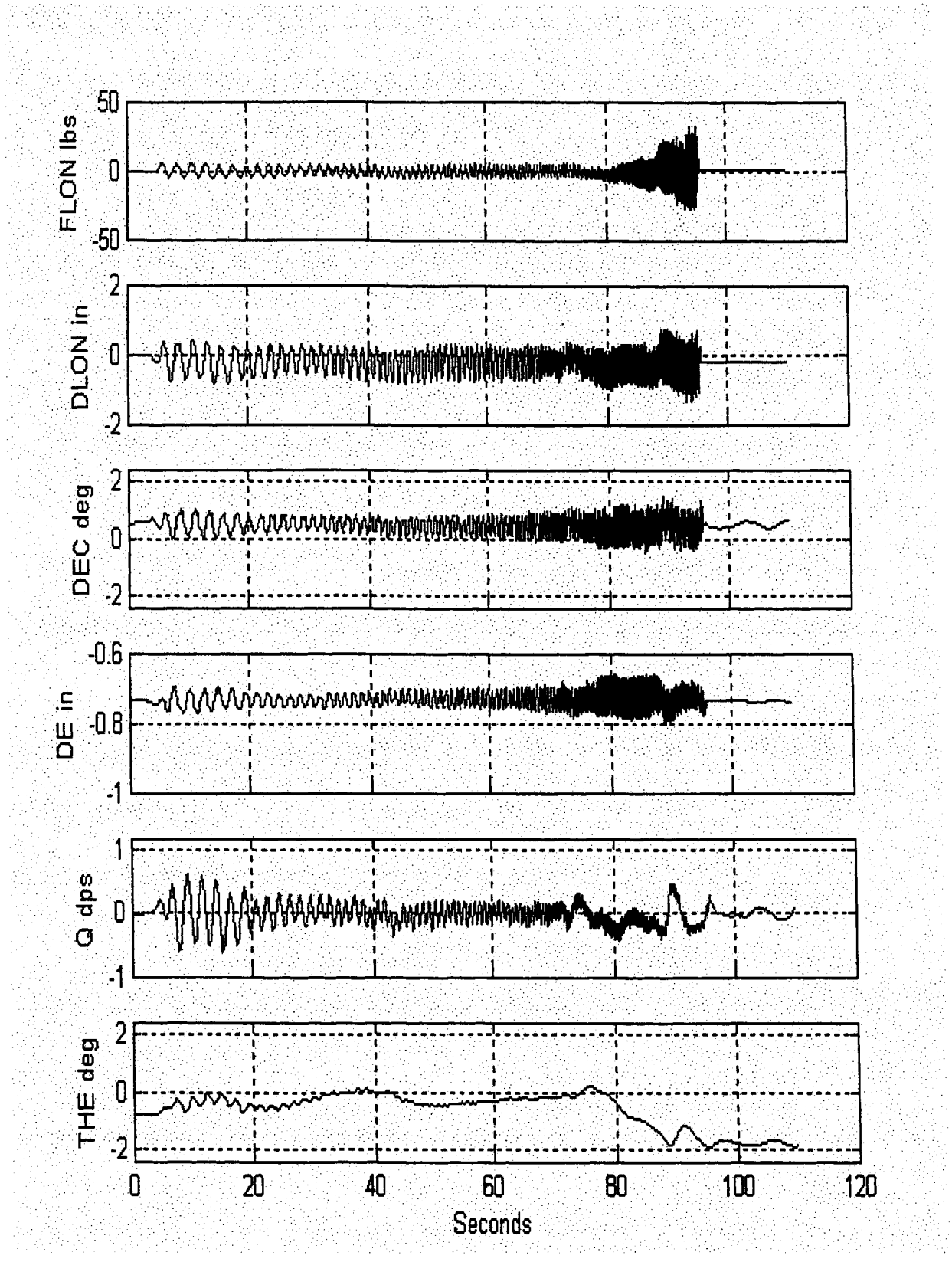


Figure 32. Run 132 CLA Pitch Sweep Time Histories (Autothrottle Off, QGain = 0)

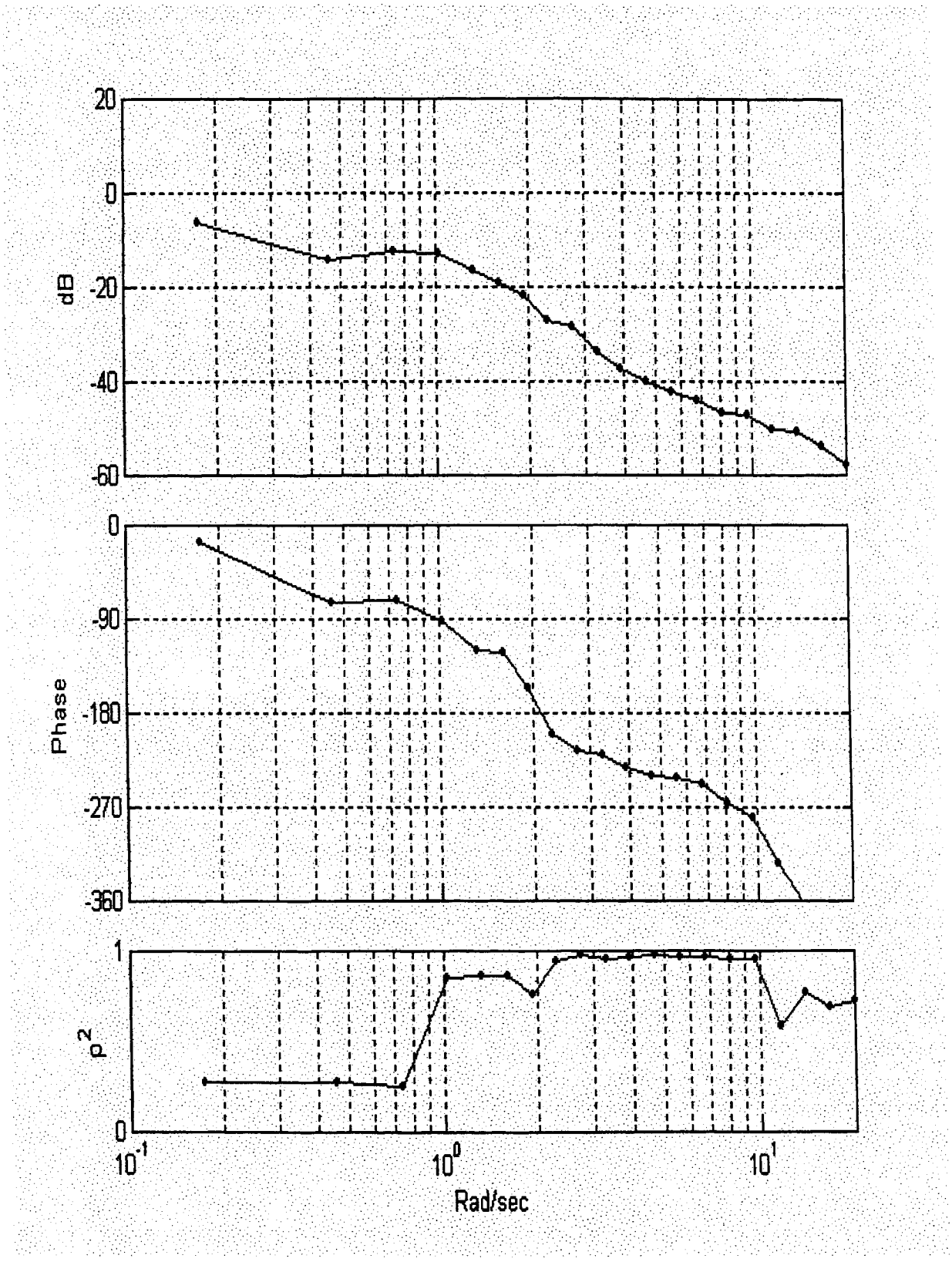


Figure 33. Run 132 CLA THE/FLON Frequency Response (Autothrottle Off, QGain = 0)

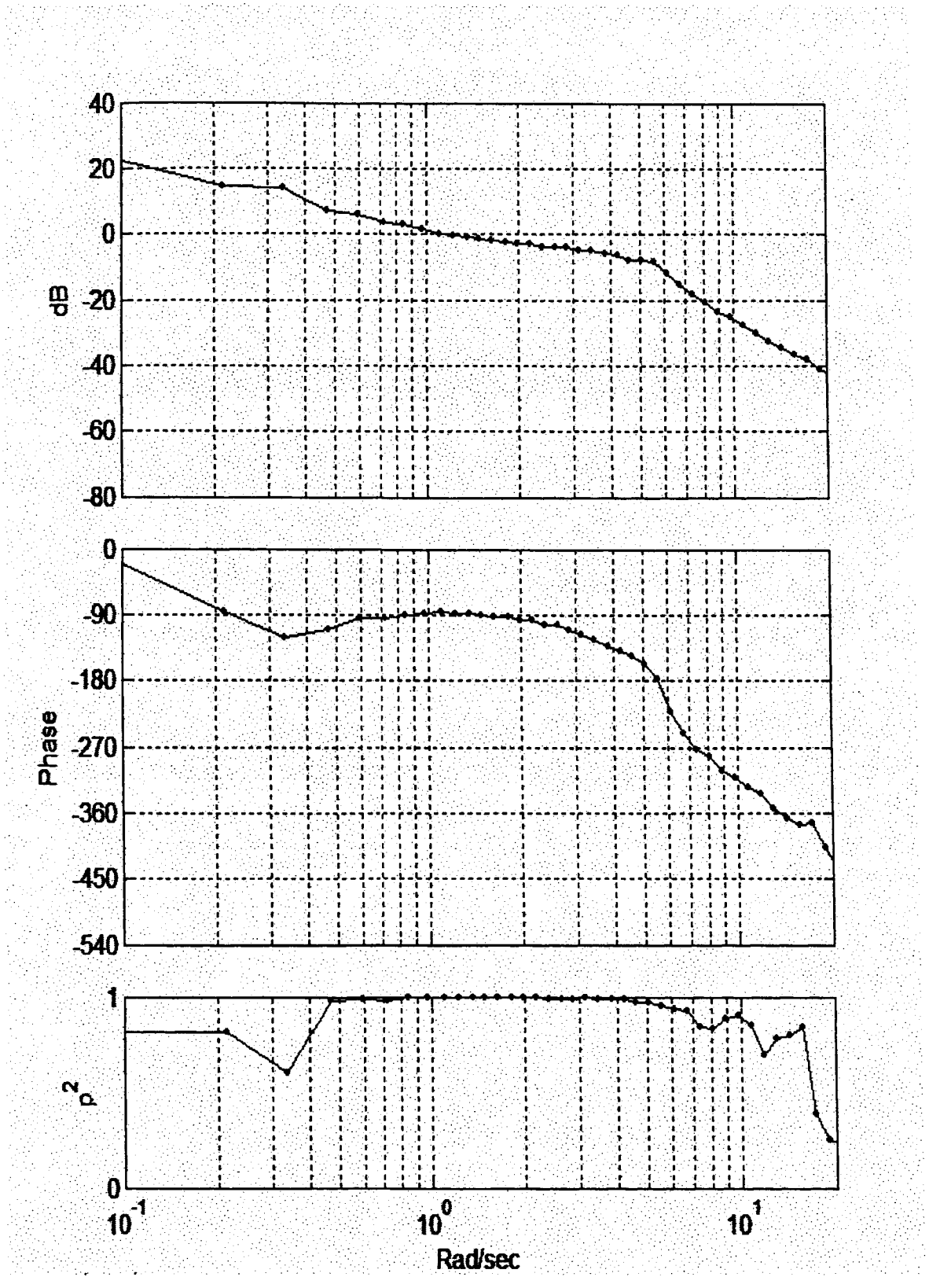


Figure 34. Run 50 CHF THE/DLON Frequency Response (Autothrottle On)

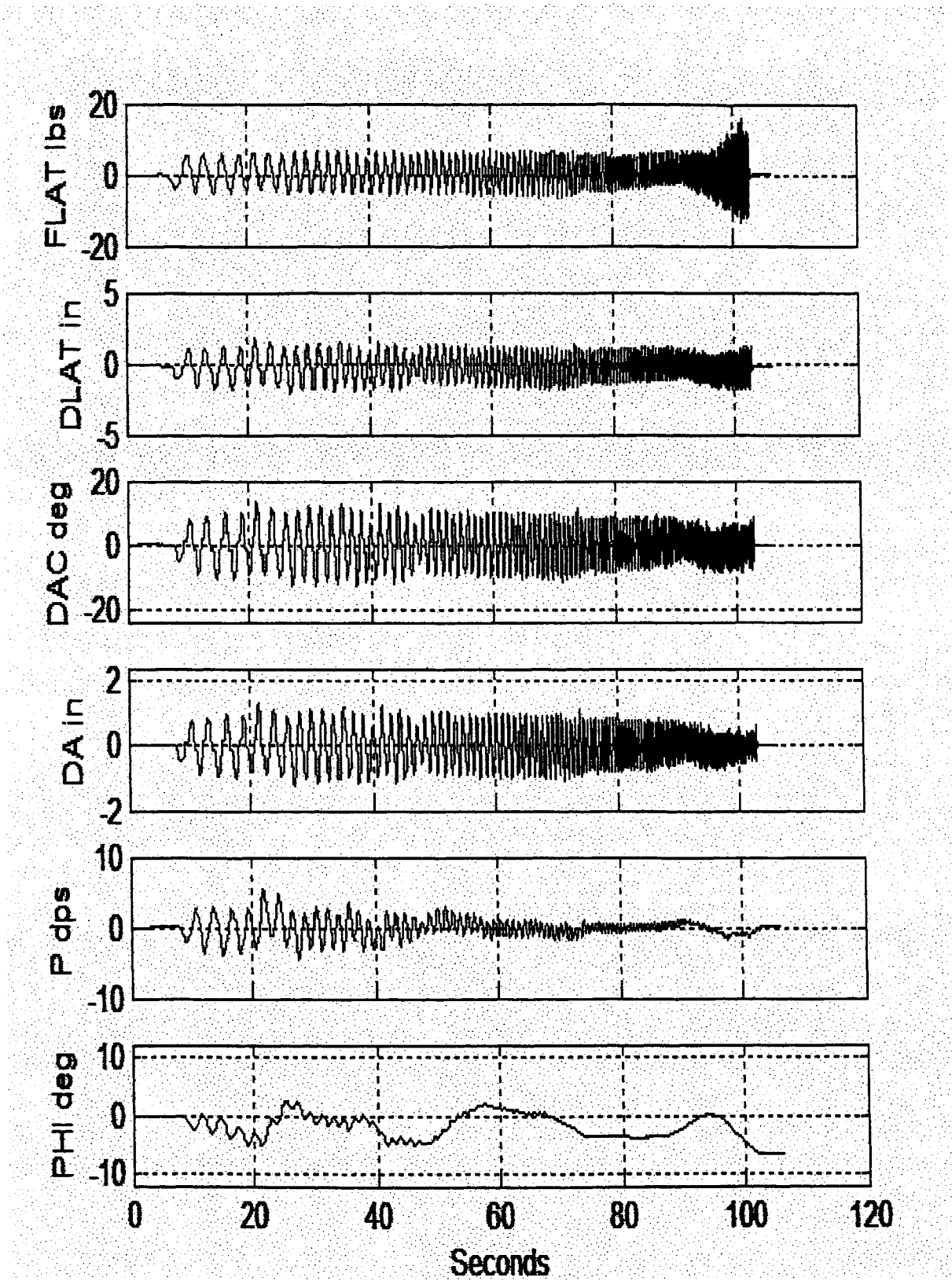


Figure 35. Run 51 CHF Roll Sweep Time Histories (Autothrottle On)

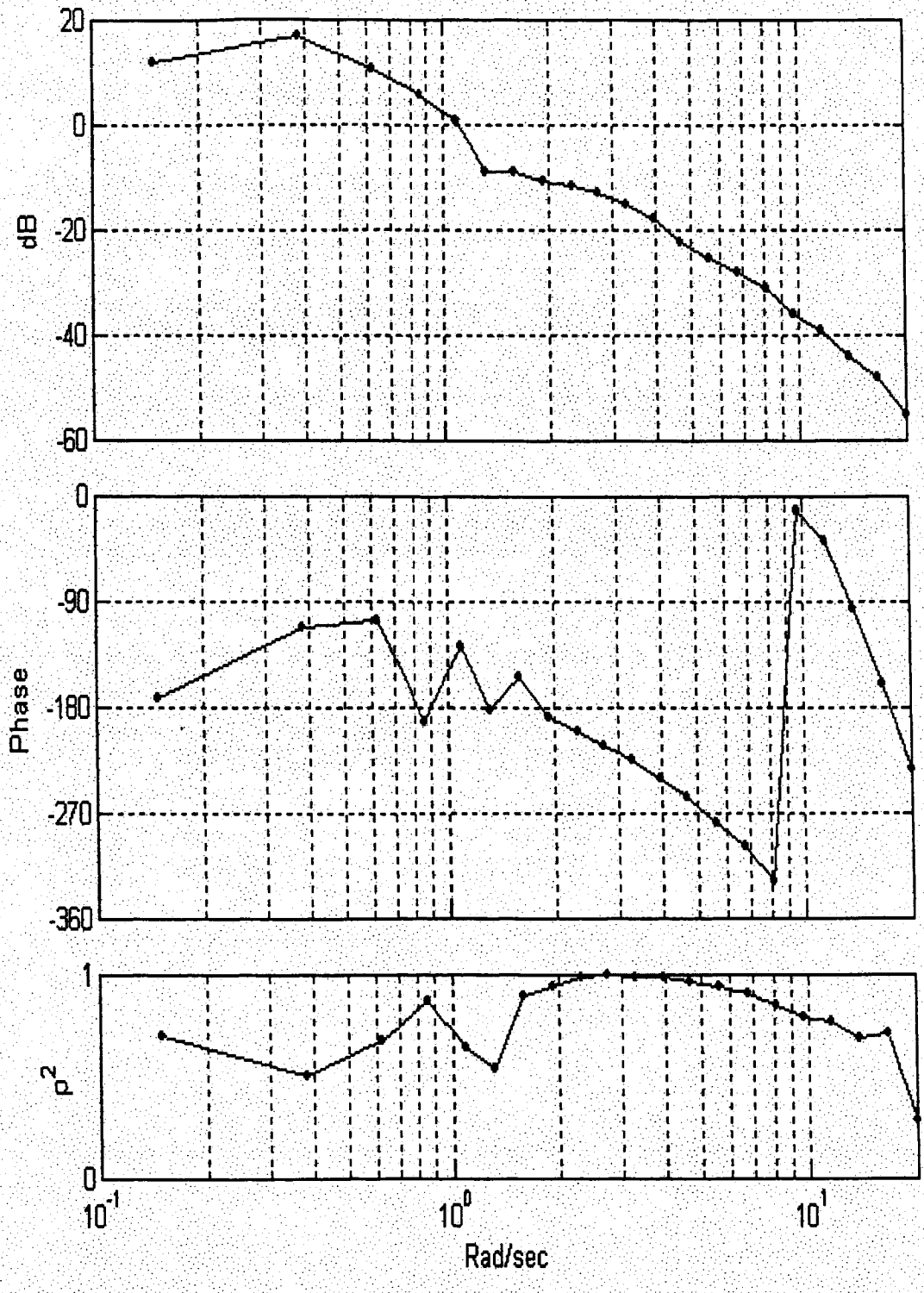


Figure 36. Run 51 CHF PHI/FLAT Frequency Response (Autothrottle On)

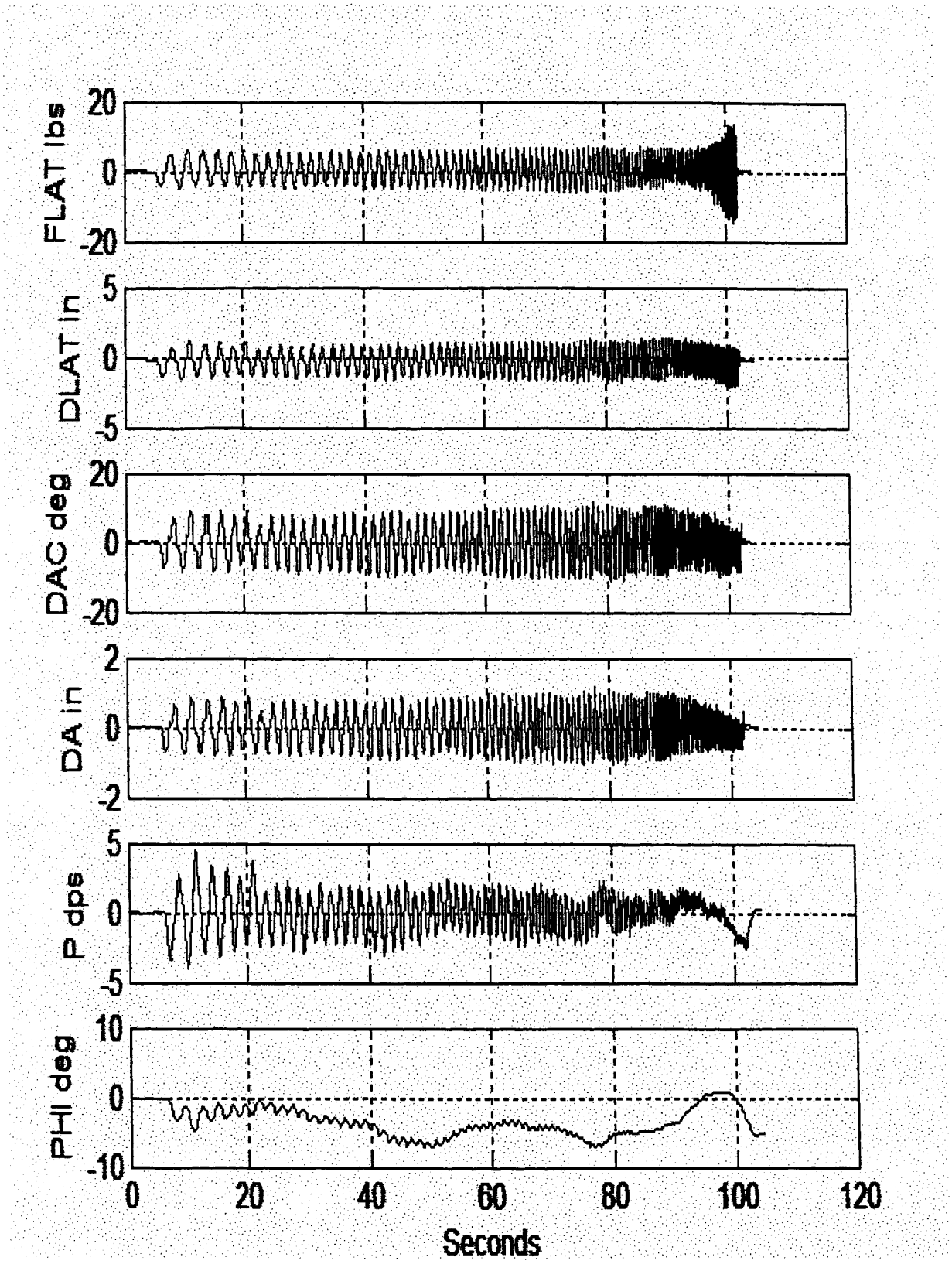


Figure 37. Run 53 CLA Roll Sweep Time Histories (Autothrottle On)

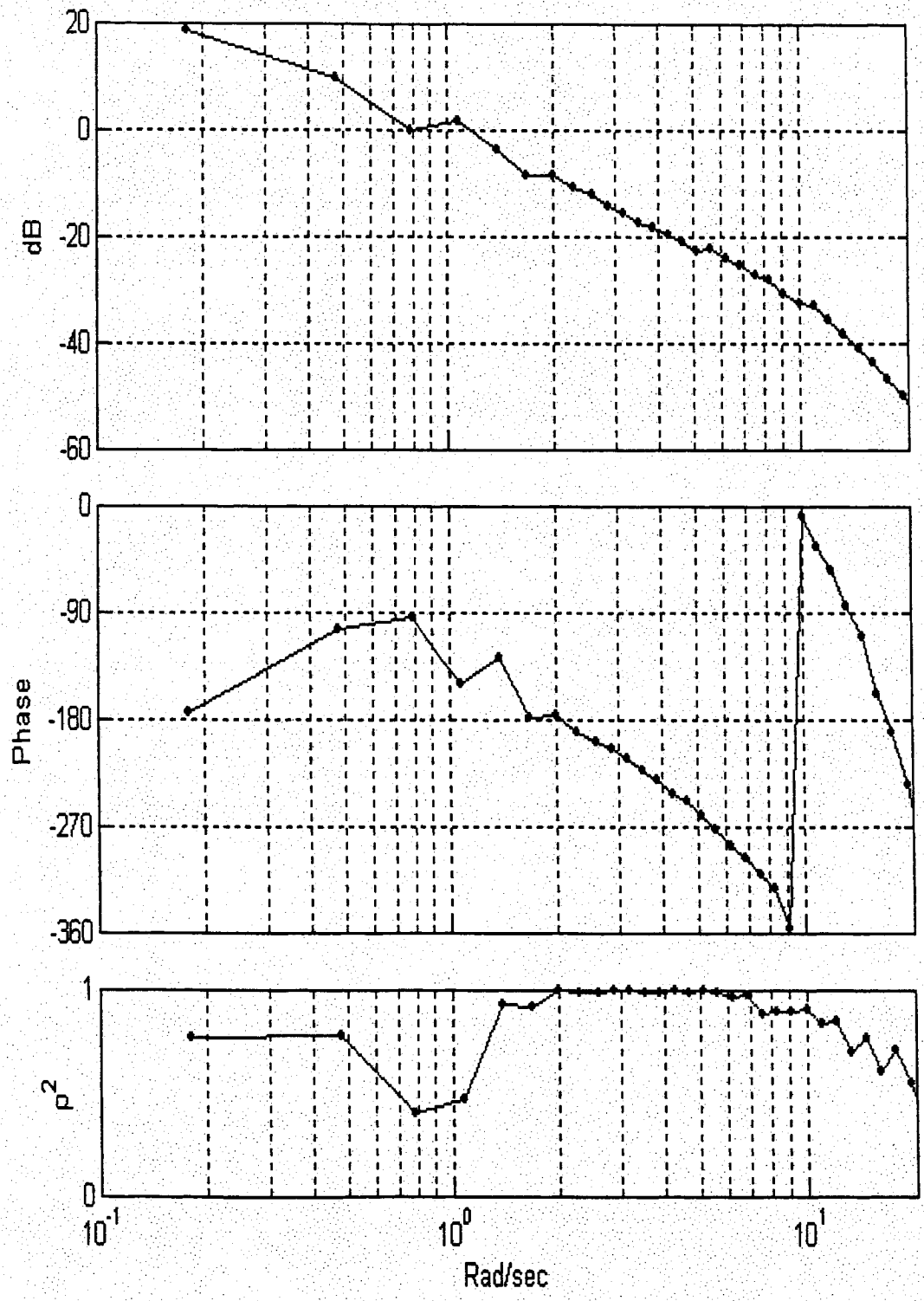


Figure 38. Run 53 CLA PHI/FLAT Frequency Response (Autothrottle On)

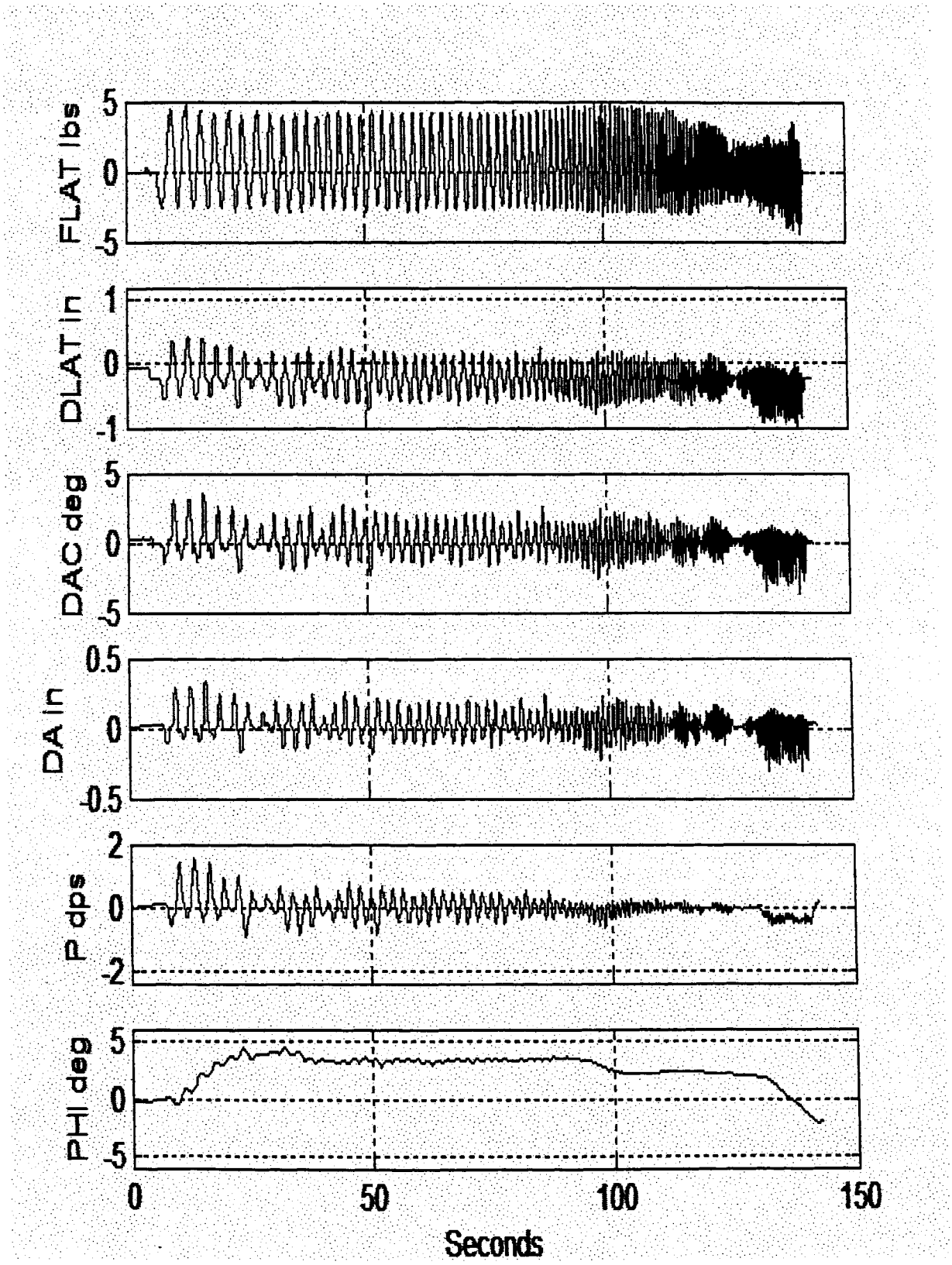


Figure 39. Run 102 CLA Roll Sweep Time Histories (Autothrottle Off)

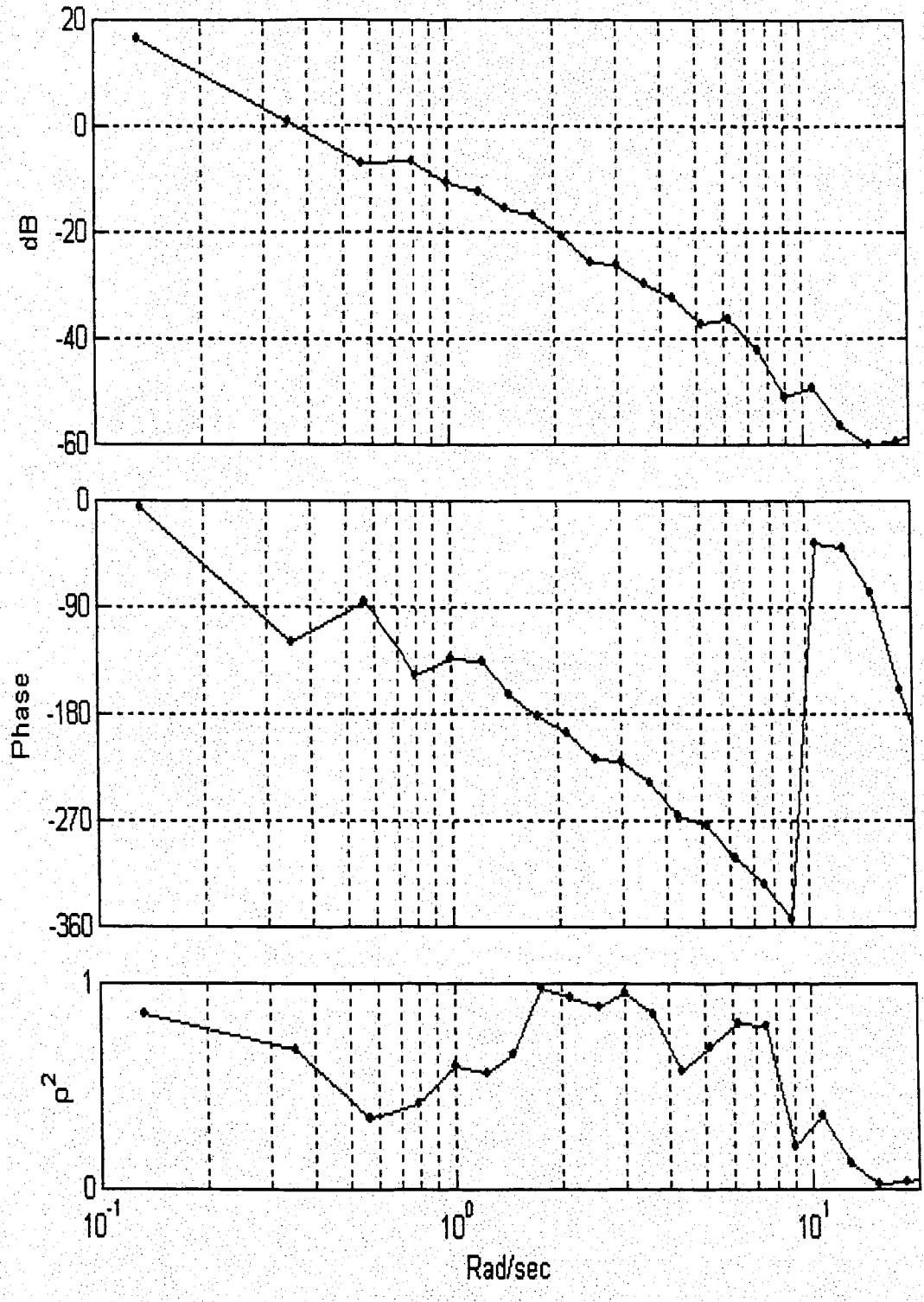


Figure 40. Run 102 CLA PHI/FLAT (Autothrottle Off)

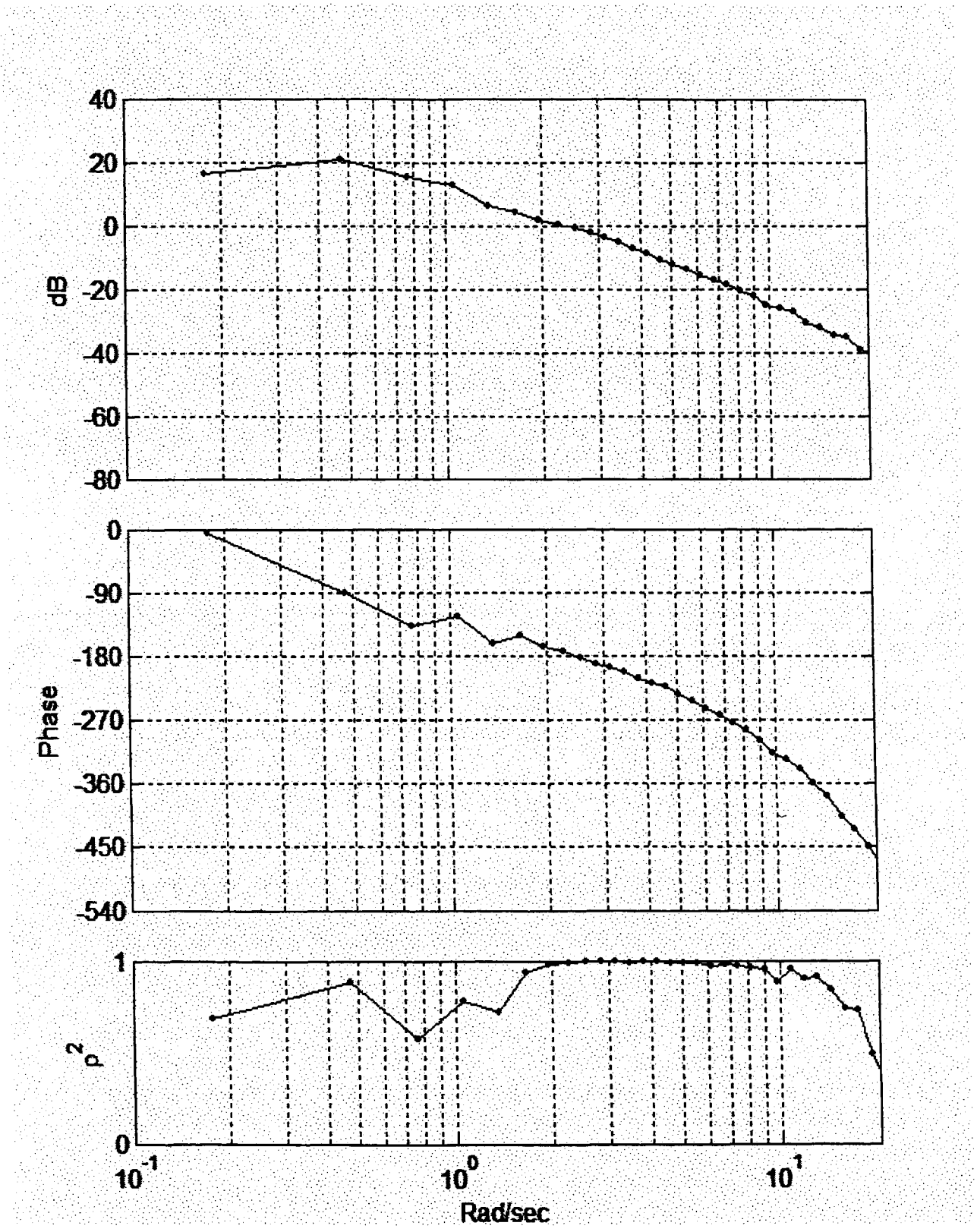


Figure 41. Run 51 CHA PHI/DLAT Frequency Response (Autothrottle On)

4.4 Bank Angle Capture and Hold Evaluations

A listing of the key bank angle capture and hold evaluation runs is provided in Table 15. Time histories and time varying transfer function (TVTF) and wavelet eigen value realization (WERA) analysis plots for the Table 15 runs are provided in Figure 42 through Figure 65.

Table 15. Selected BACH Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R77	A	ALA	BACH	n/a	
R80	AX	ALA	BACH	n/a	
R120	CH	CLA	BACH	PGain	
Mild PIO Runs					
R28	C	CLA	BACH	BGain + PGain	Mild, damped oscillations
R79	A	ALA	BACH	BGain + PGain	RollTD whole run, Large overshoots
R122	CH	CLA	BACH	BGain + PGain	RollTD whole run, mild PIO in middle or run
Sustained or Severe PIO Runs					
R29	C	CLA	BACH	BGain + PGain + RollTD	Large overshoots follow failure and lead to PIO, oscillations damp out
R82	AX	ALA	BACH	BGain + PGain	RollTD whole run, mild PIO transitions to severe, sustained oscillations

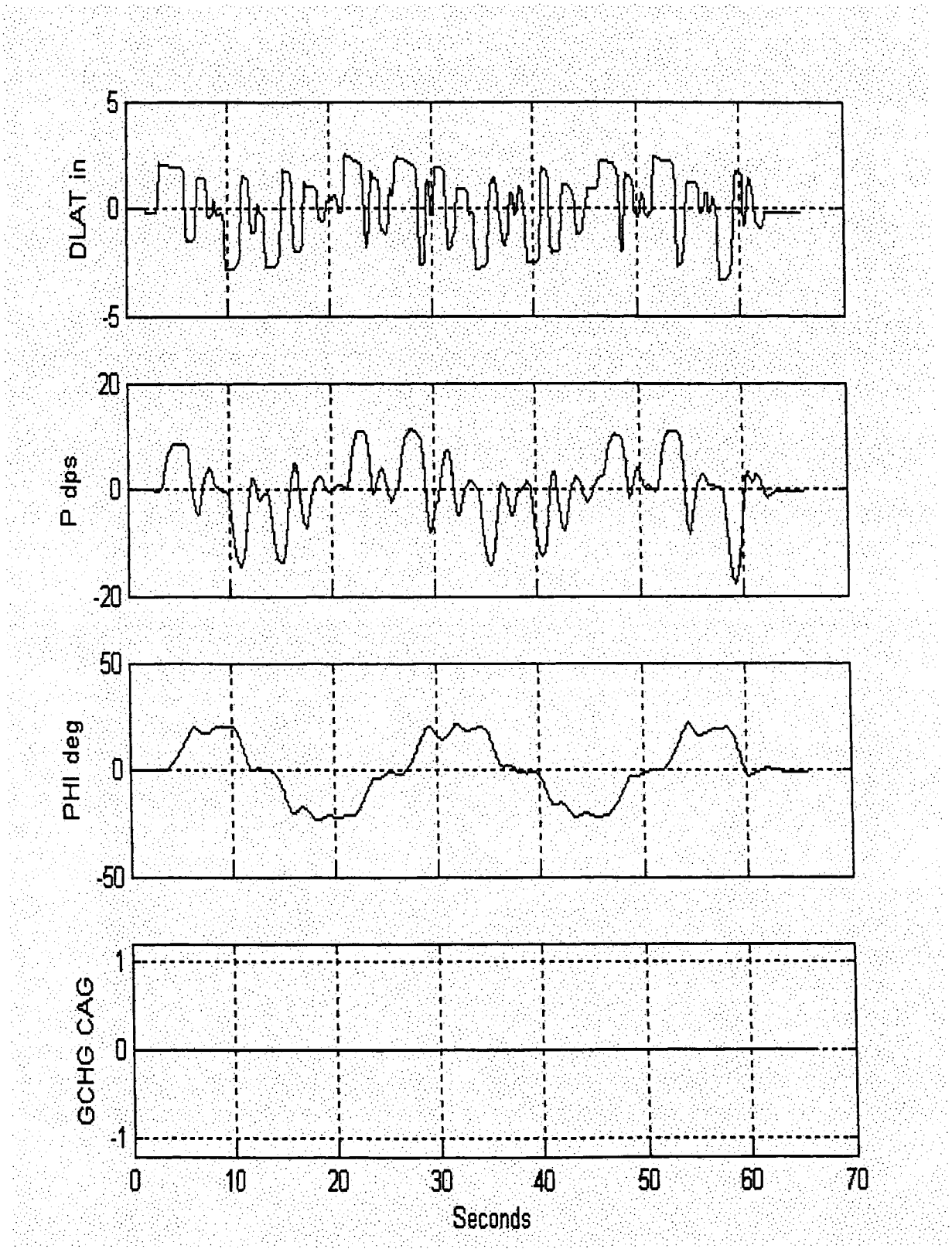
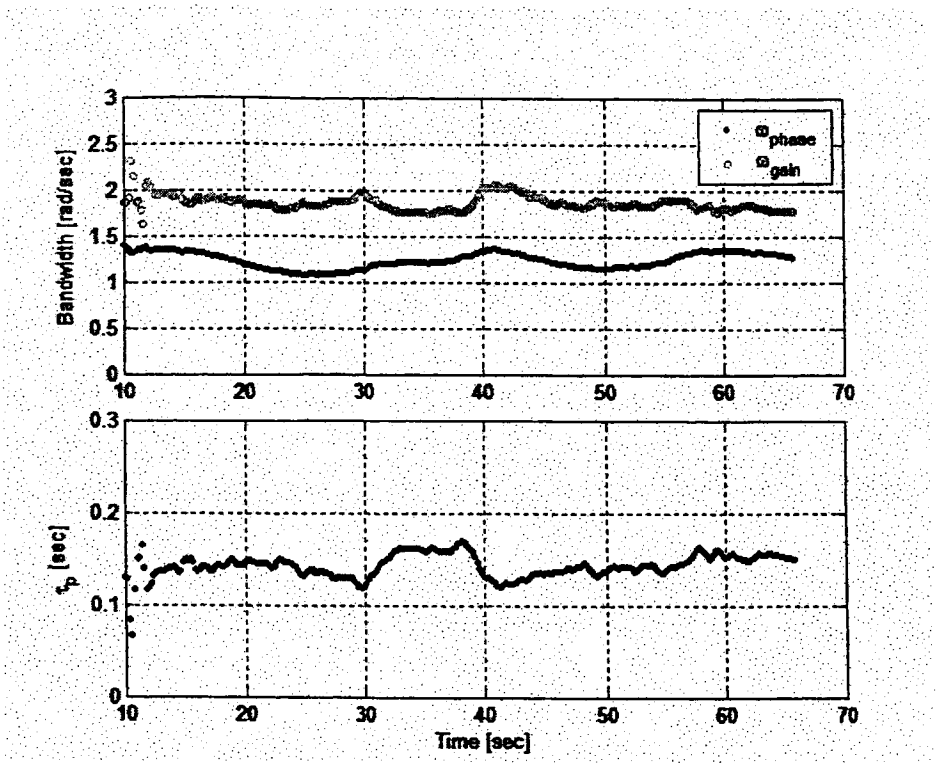
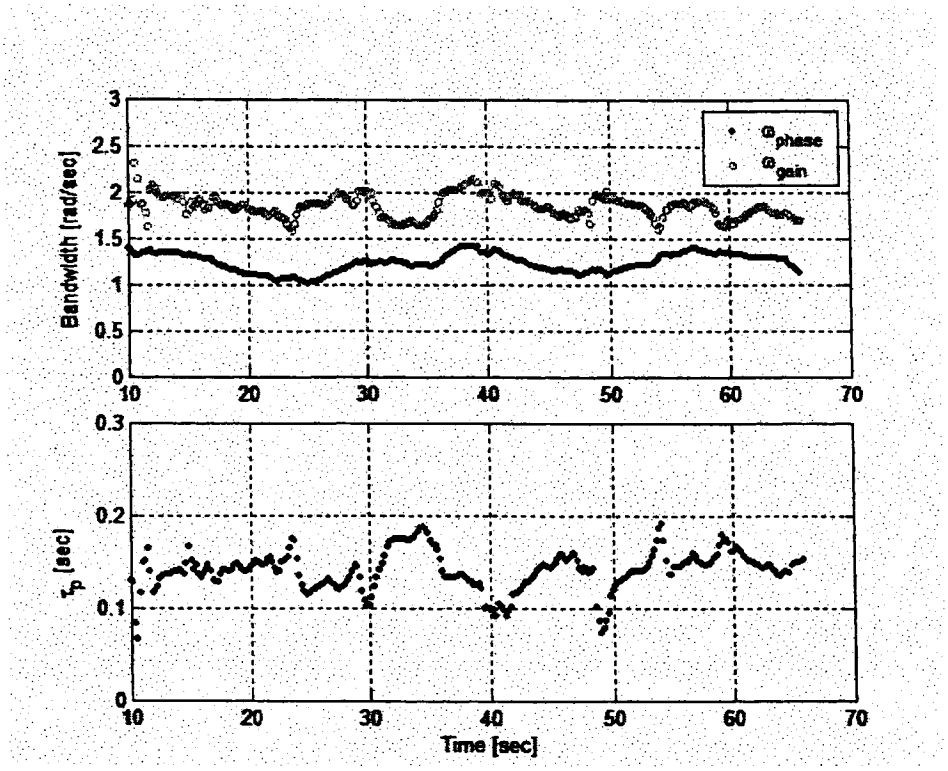


Figure 42. R77 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 43. R77 TVTF Analysis

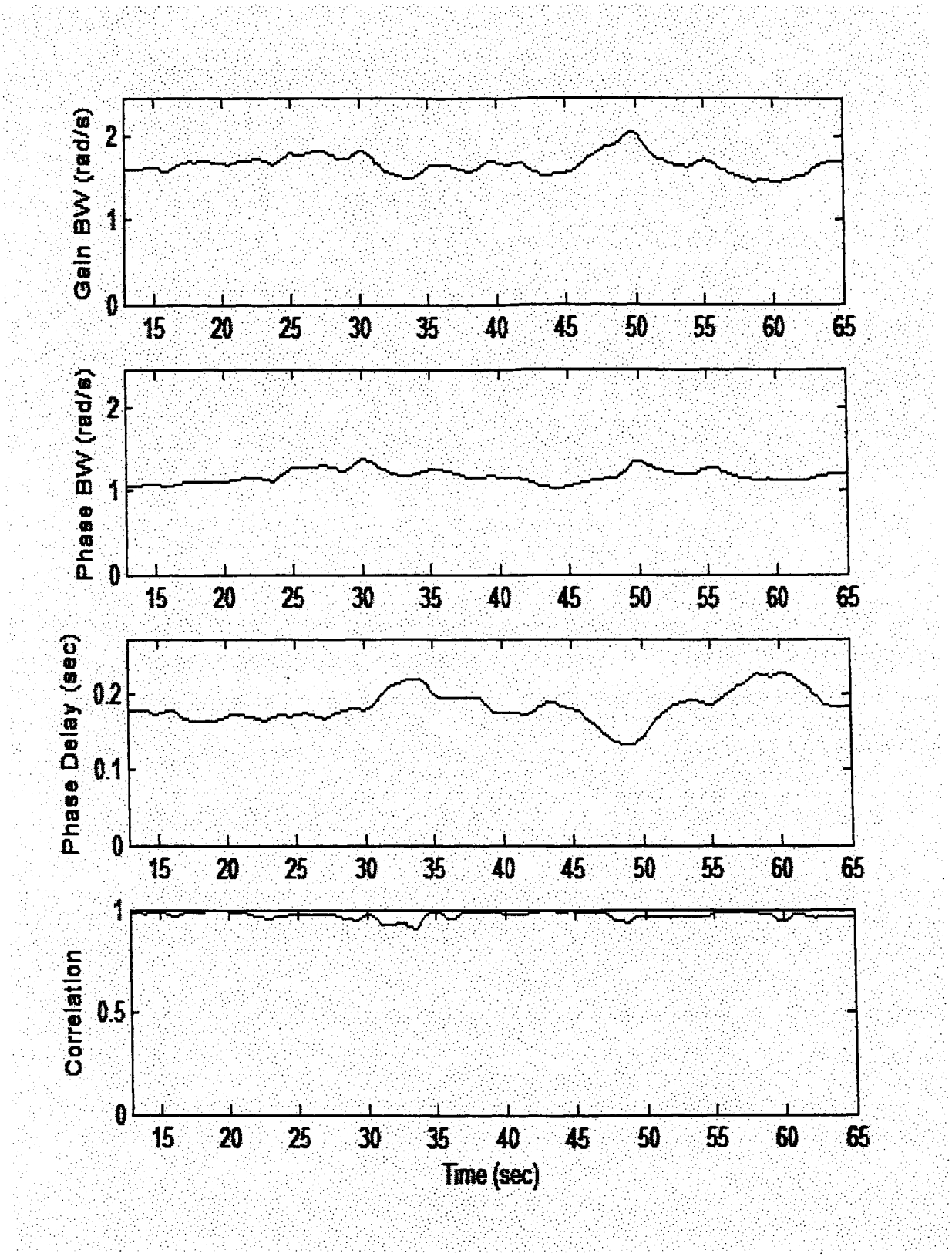


Figure 44. R77 WERA Analysis

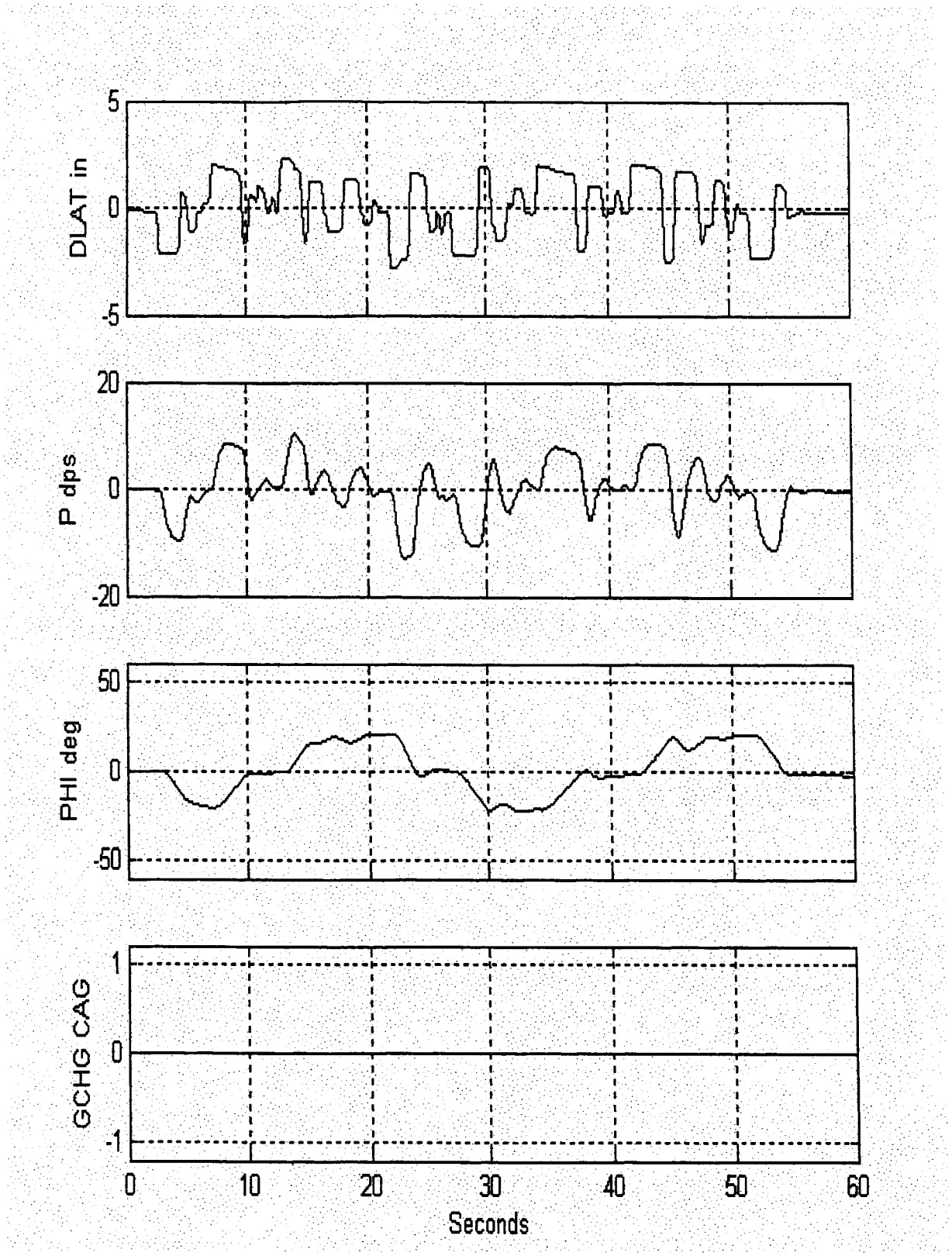
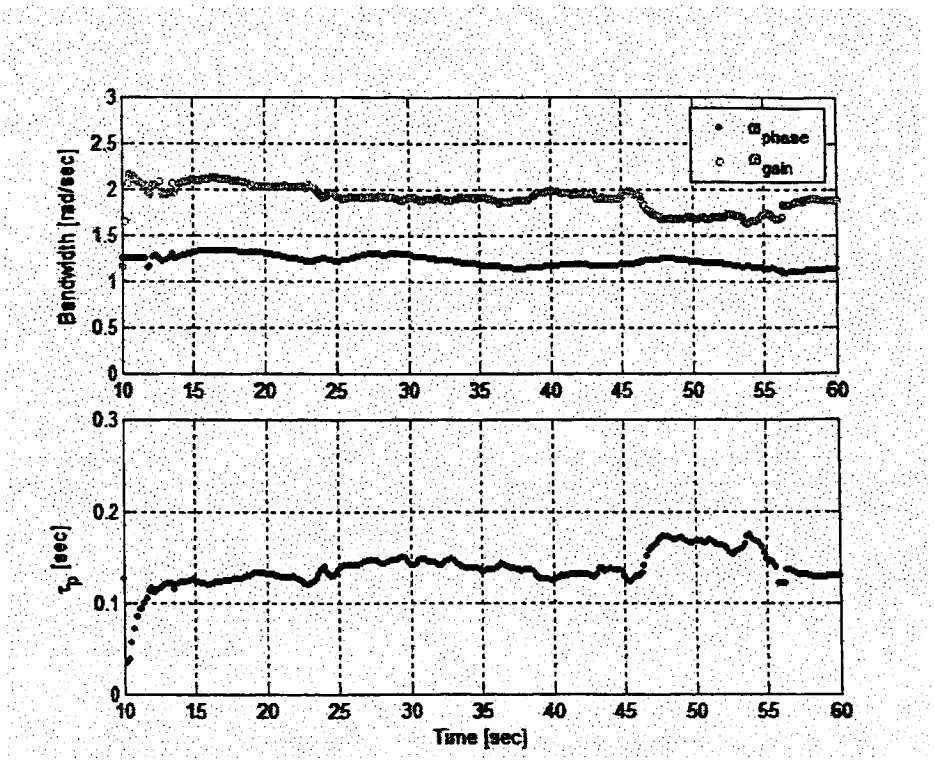
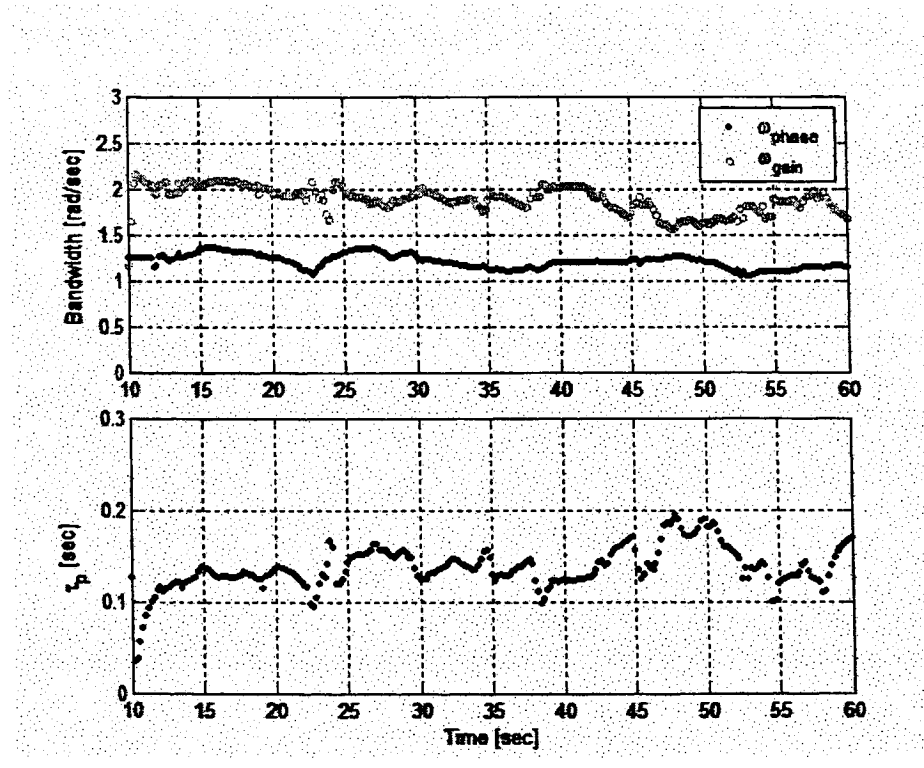


Figure 45. R80 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 46. R80 TVTF Analysis

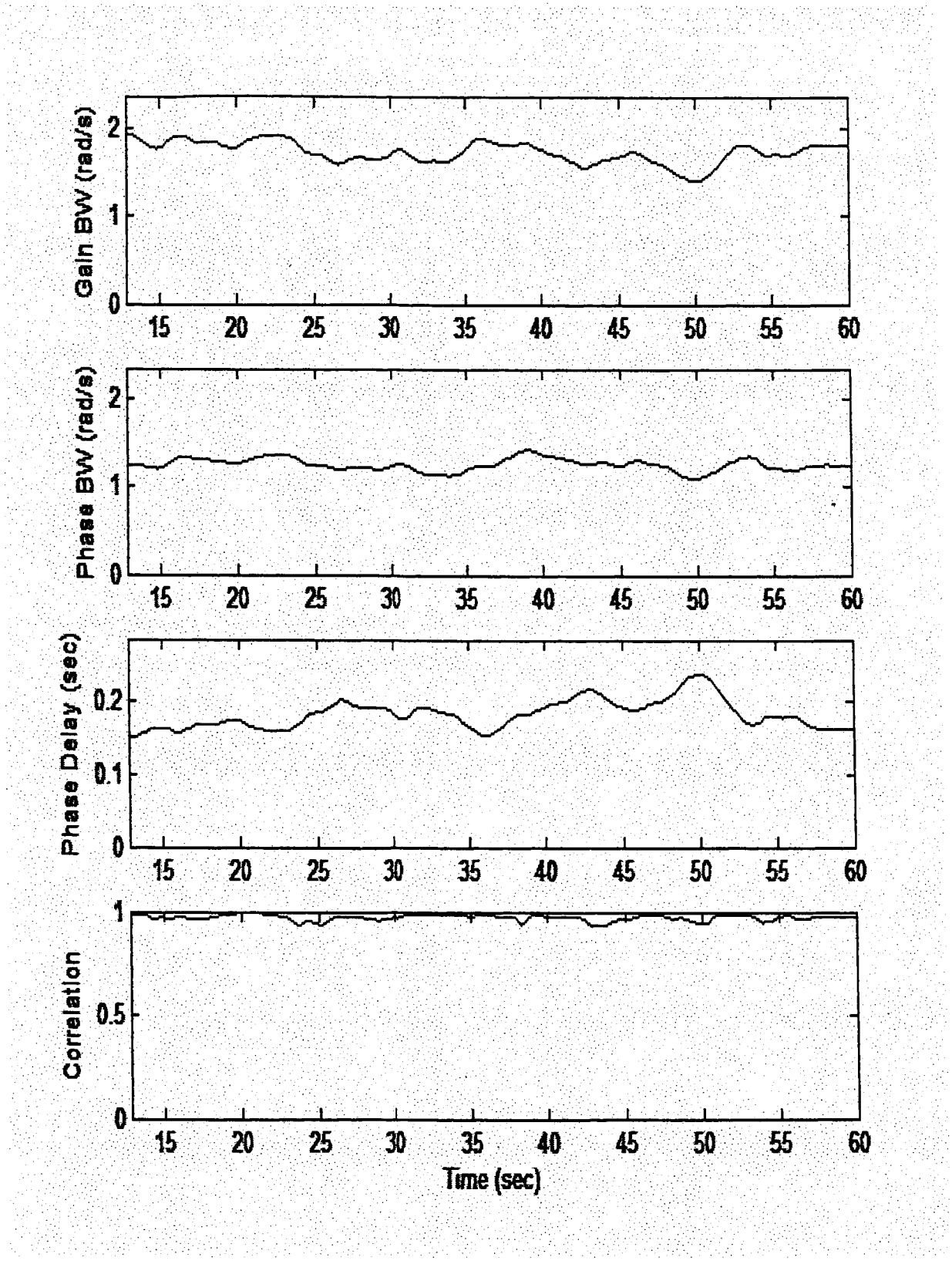


Figure 47. R80 WERA Analysis

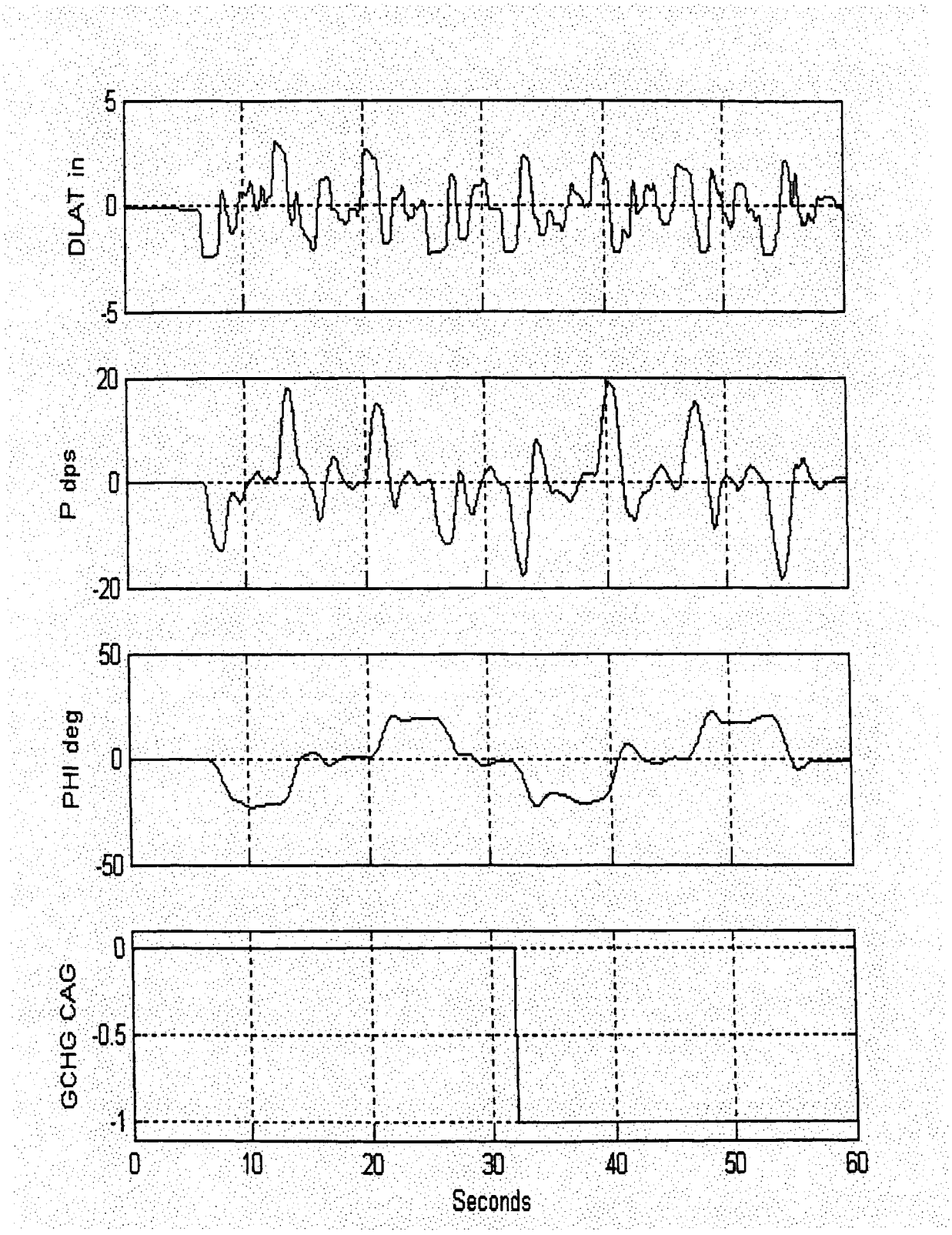
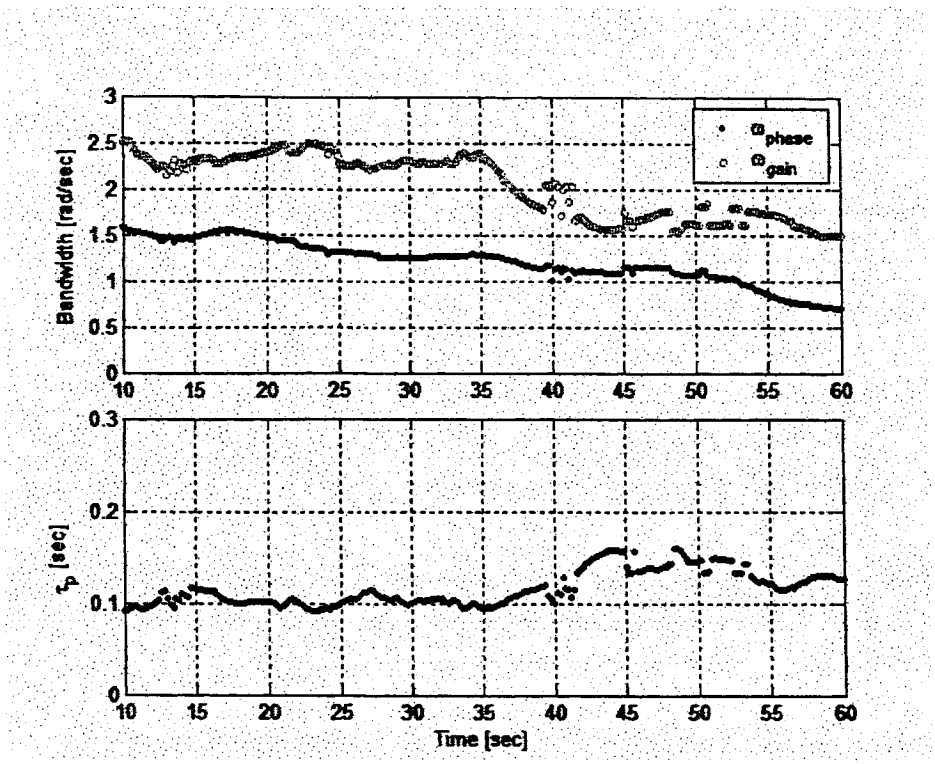
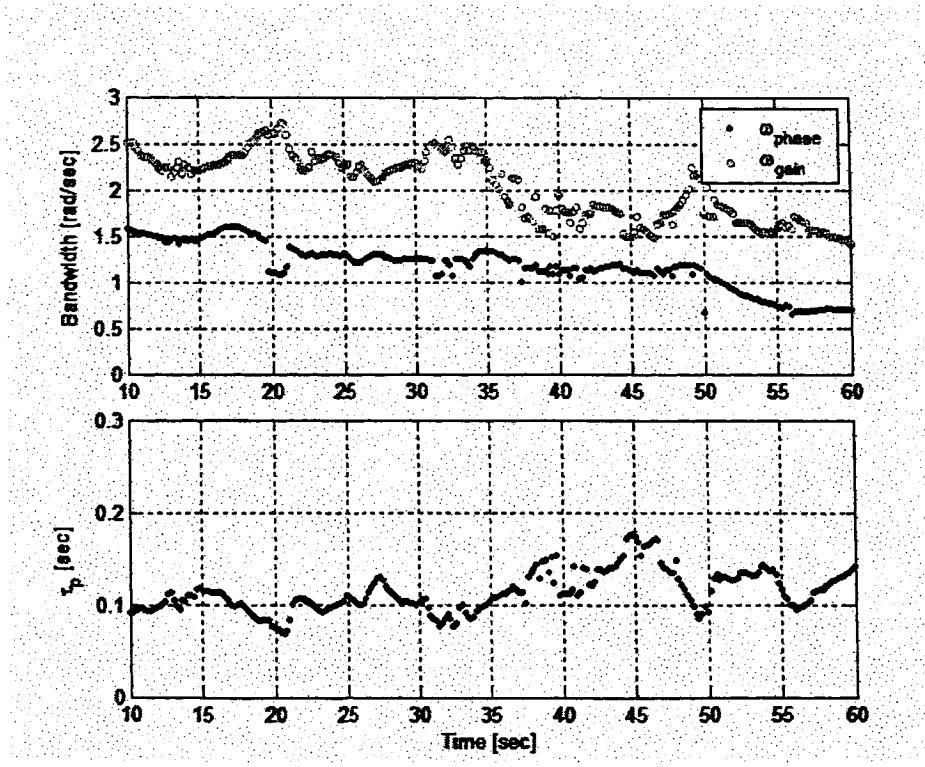


Figure 48. R120 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 49. R120 TVTF Analysis

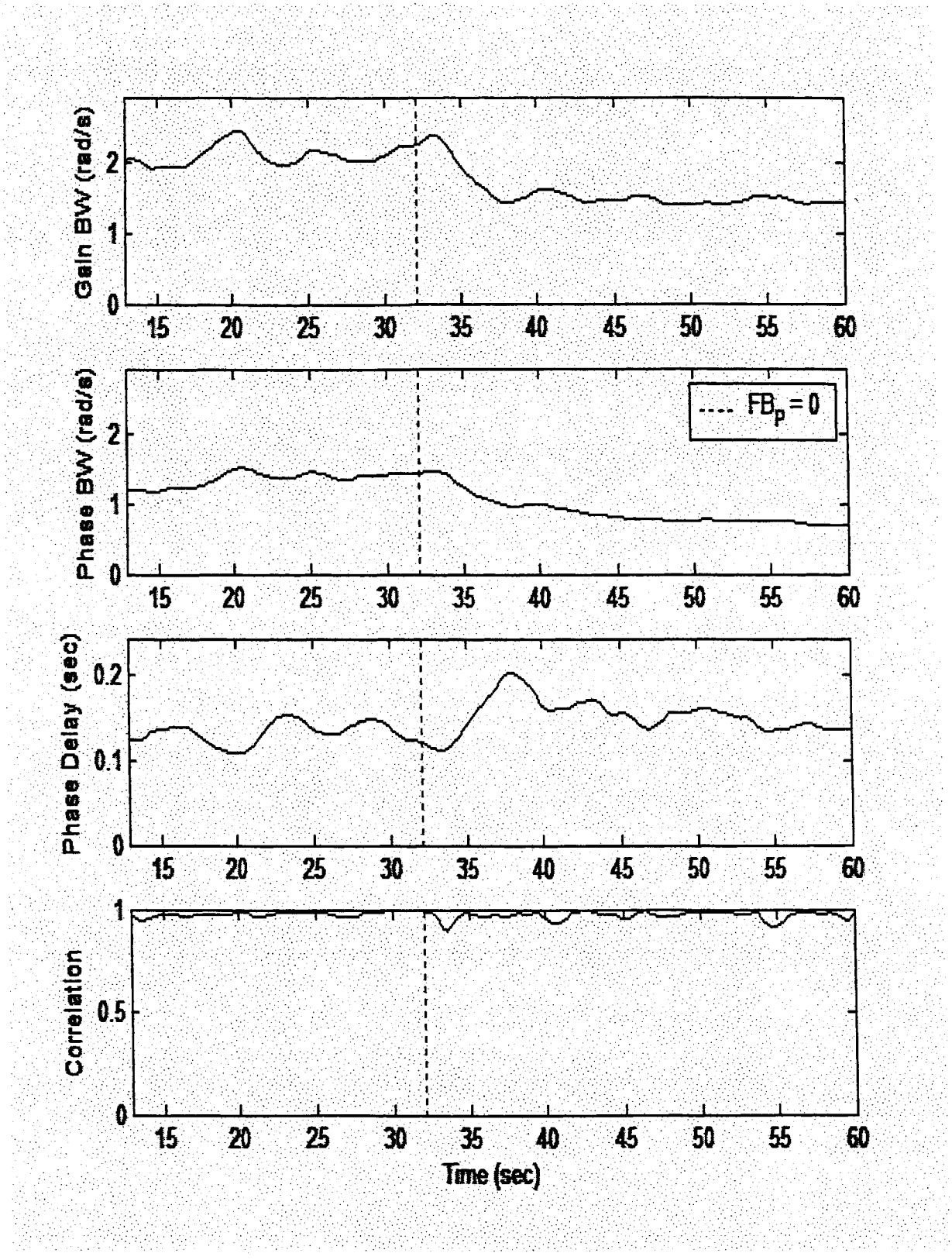


Figure 50. R120 WERA Analysis

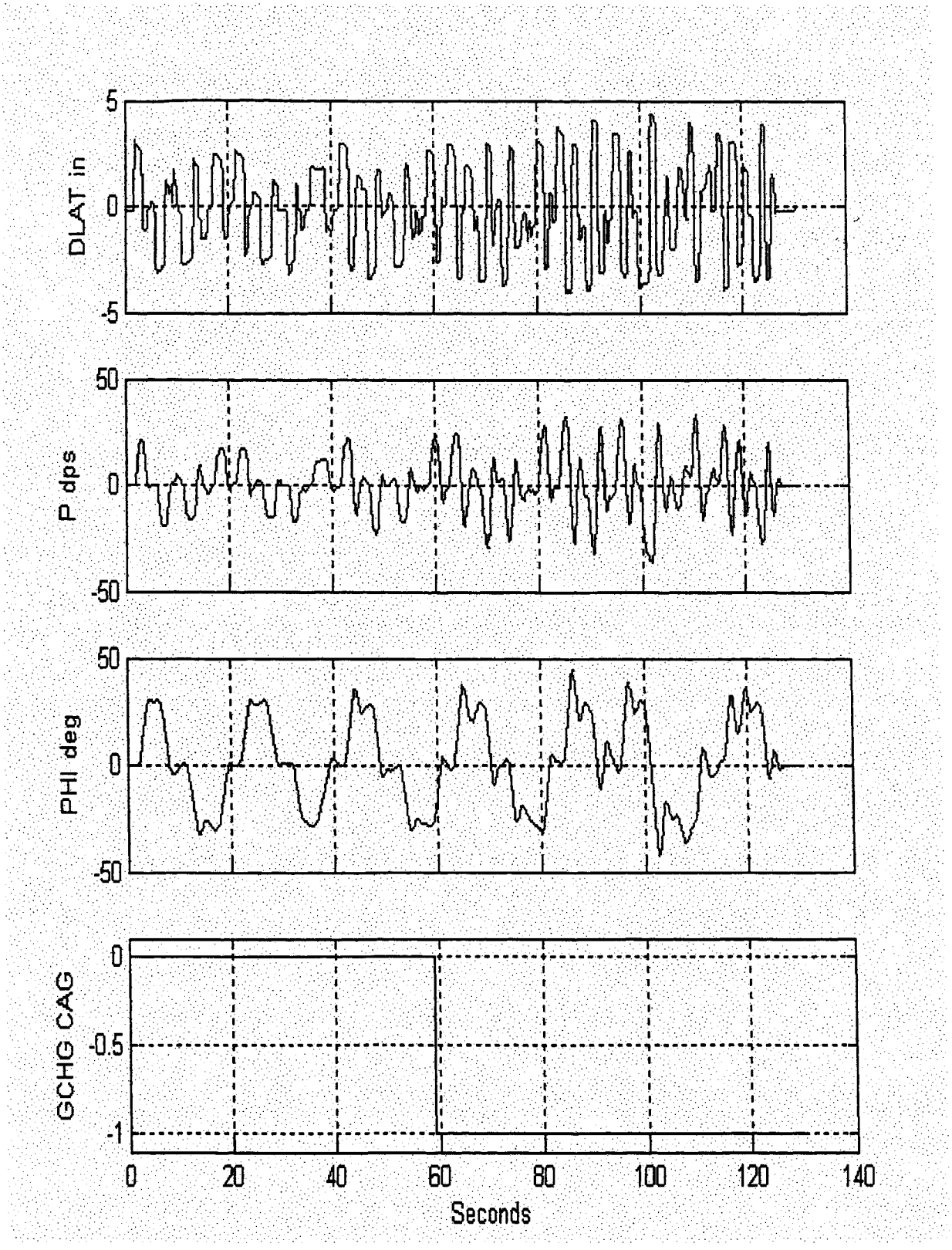
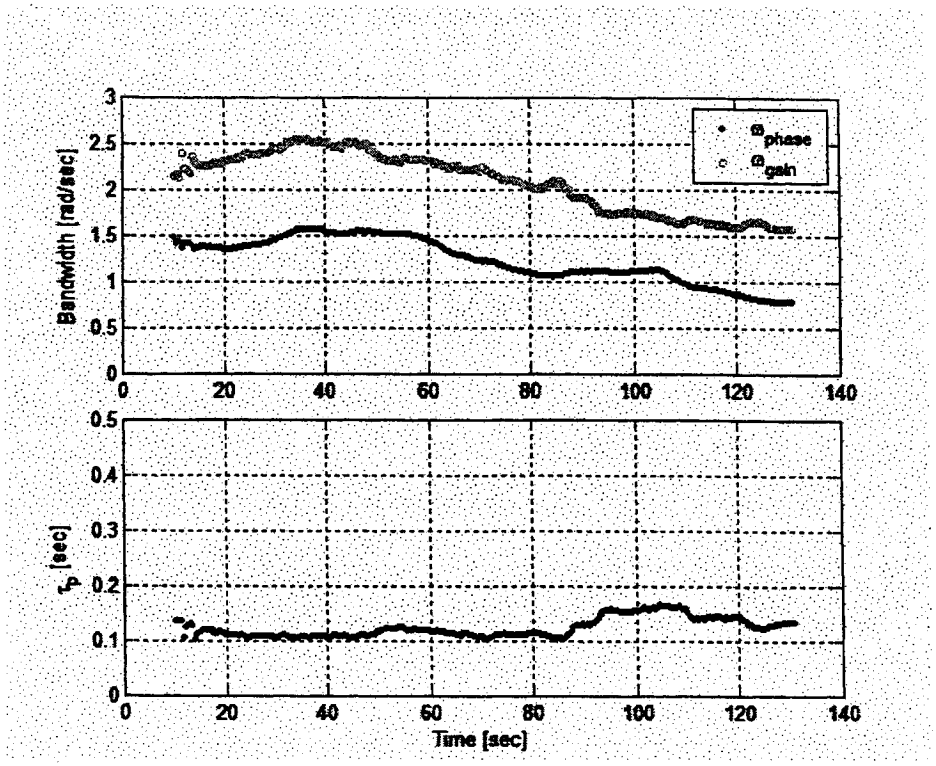
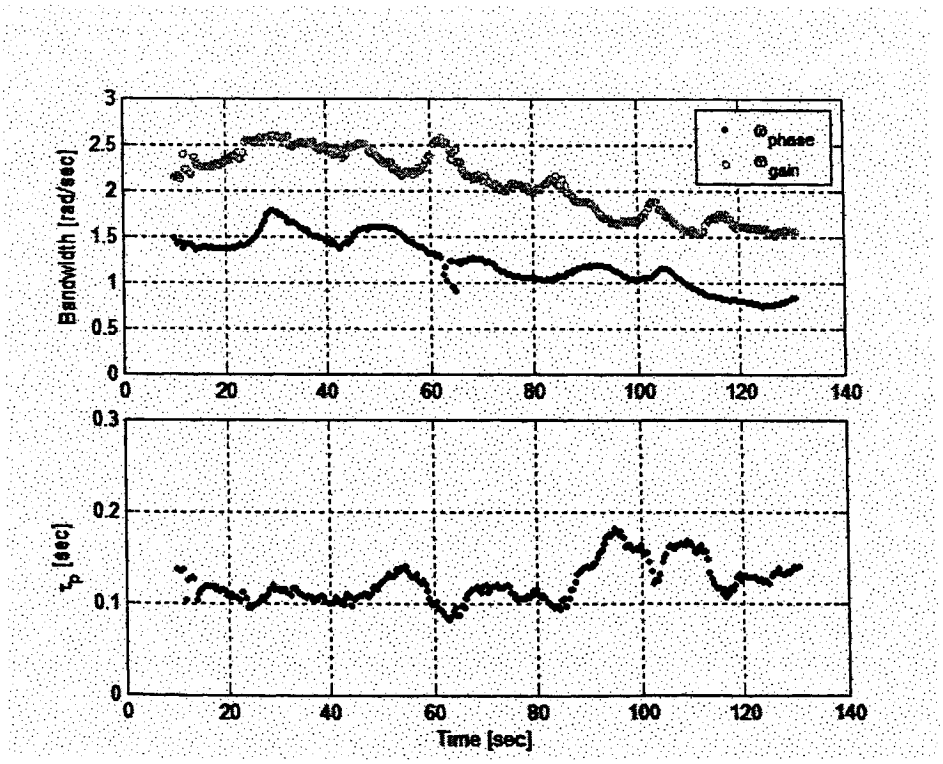


Figure 51. R28 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 52. R28 TVTF Analysis

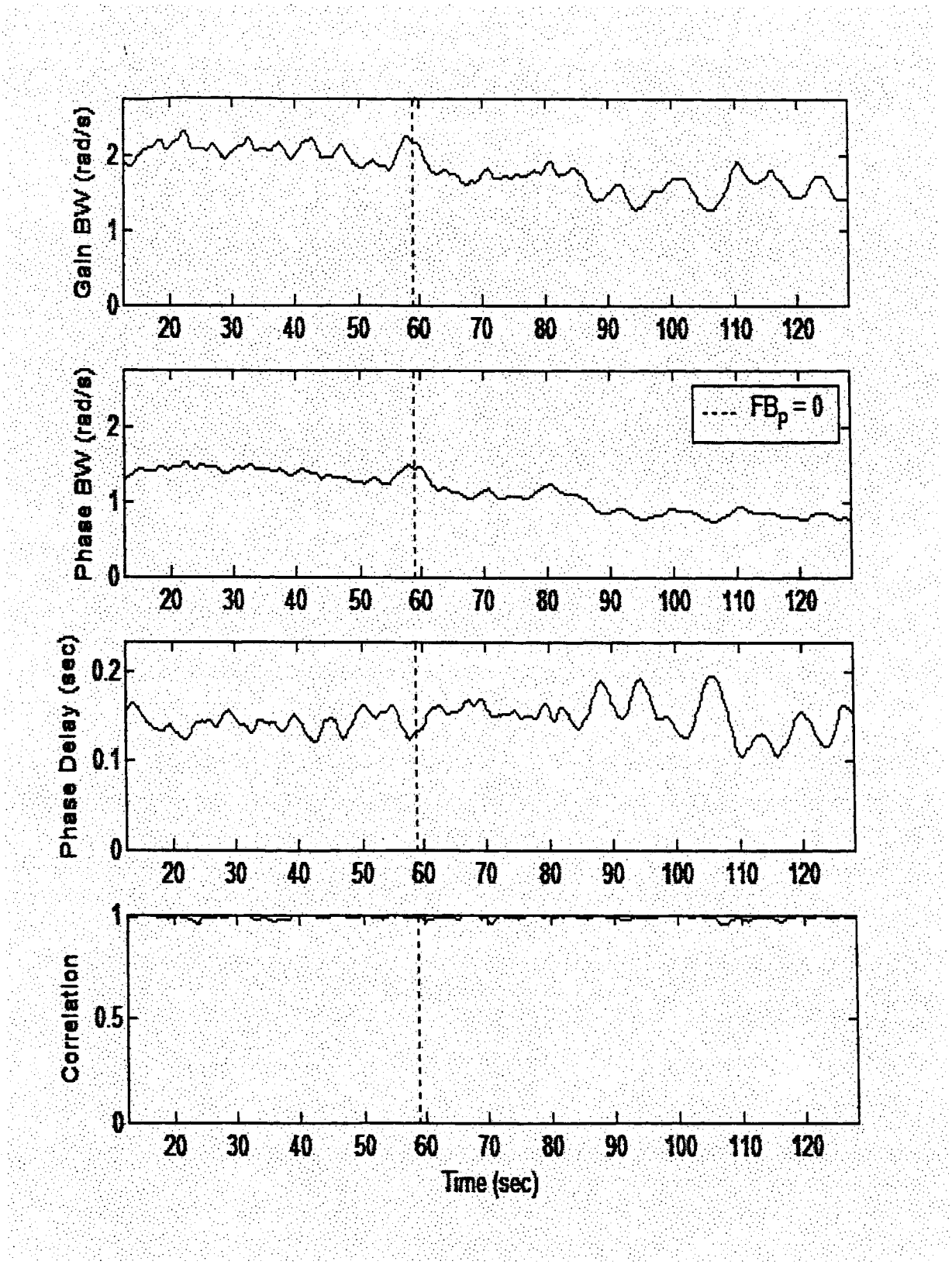


Figure 53. R28 WERA Analysis

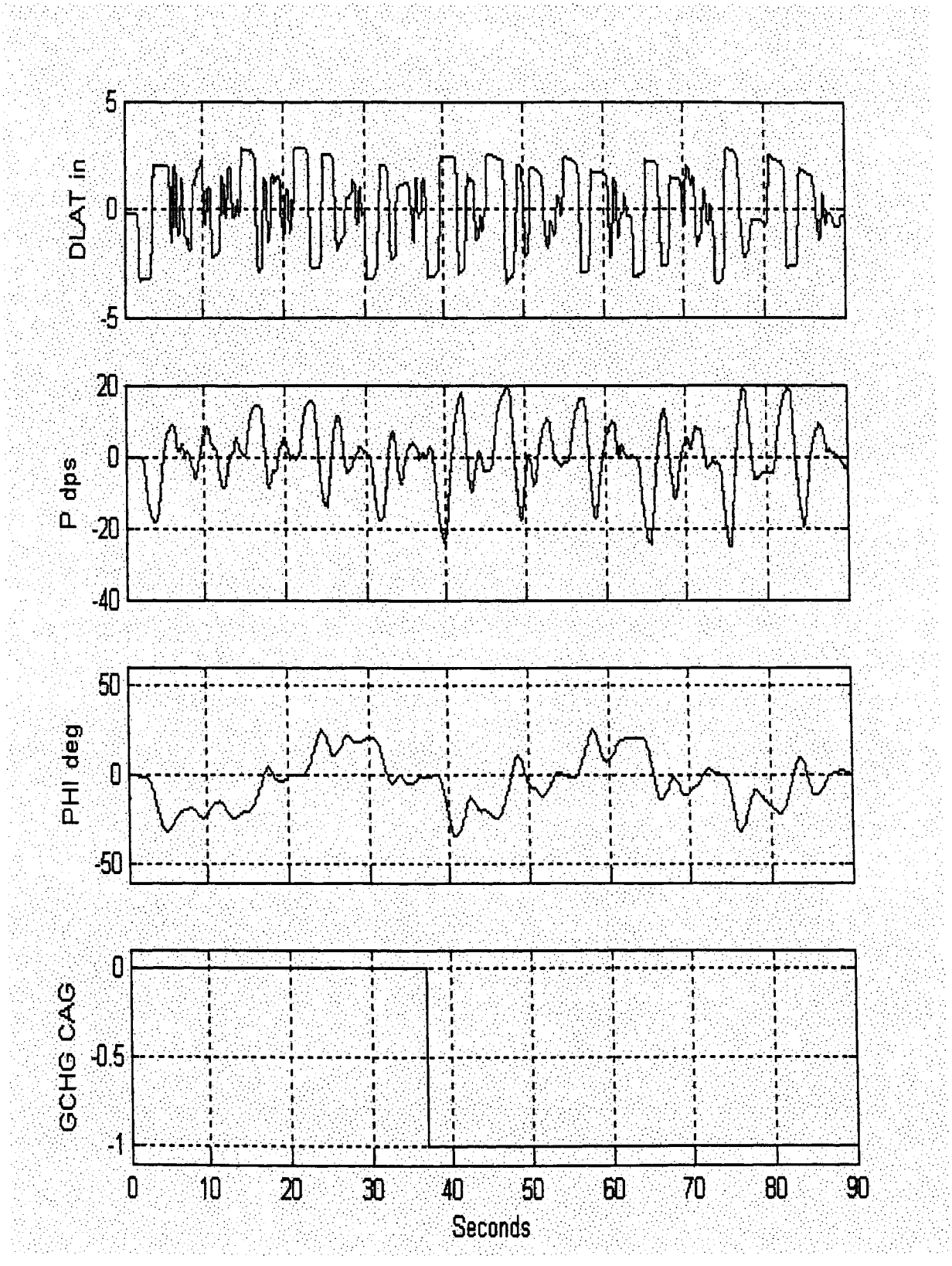
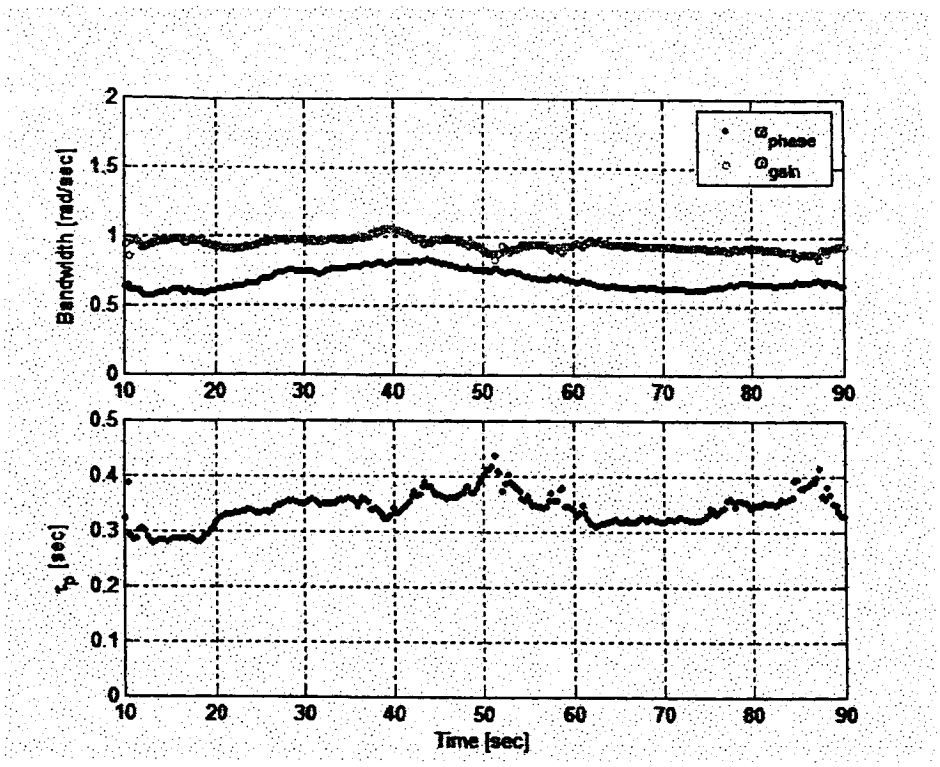
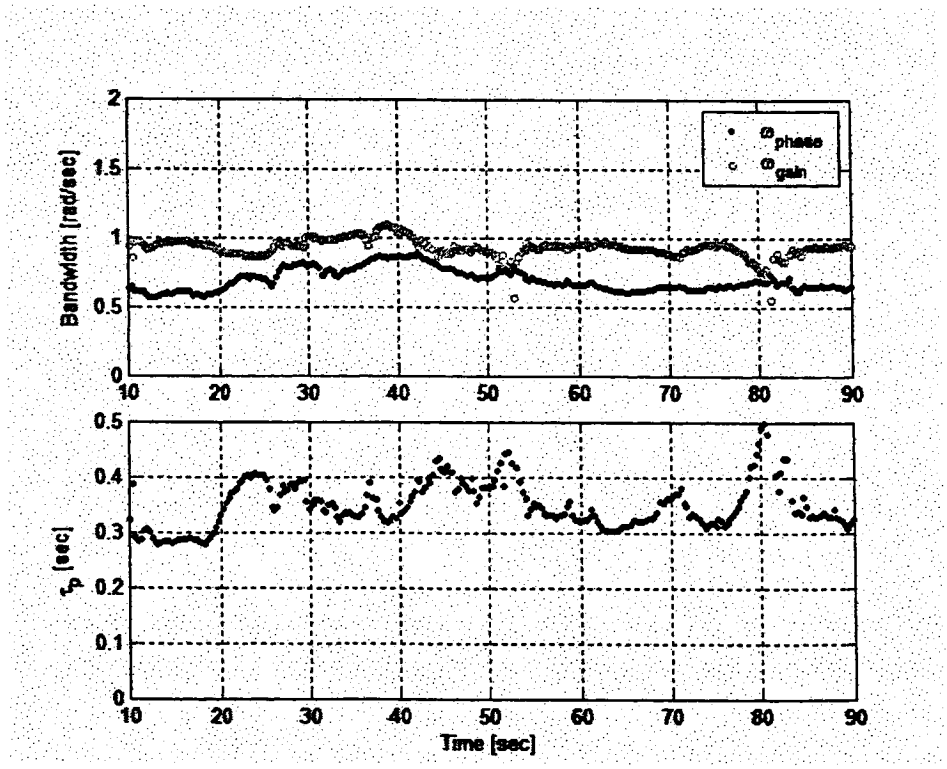


Figure 54. R79 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 55. R79 TVTF Analysis

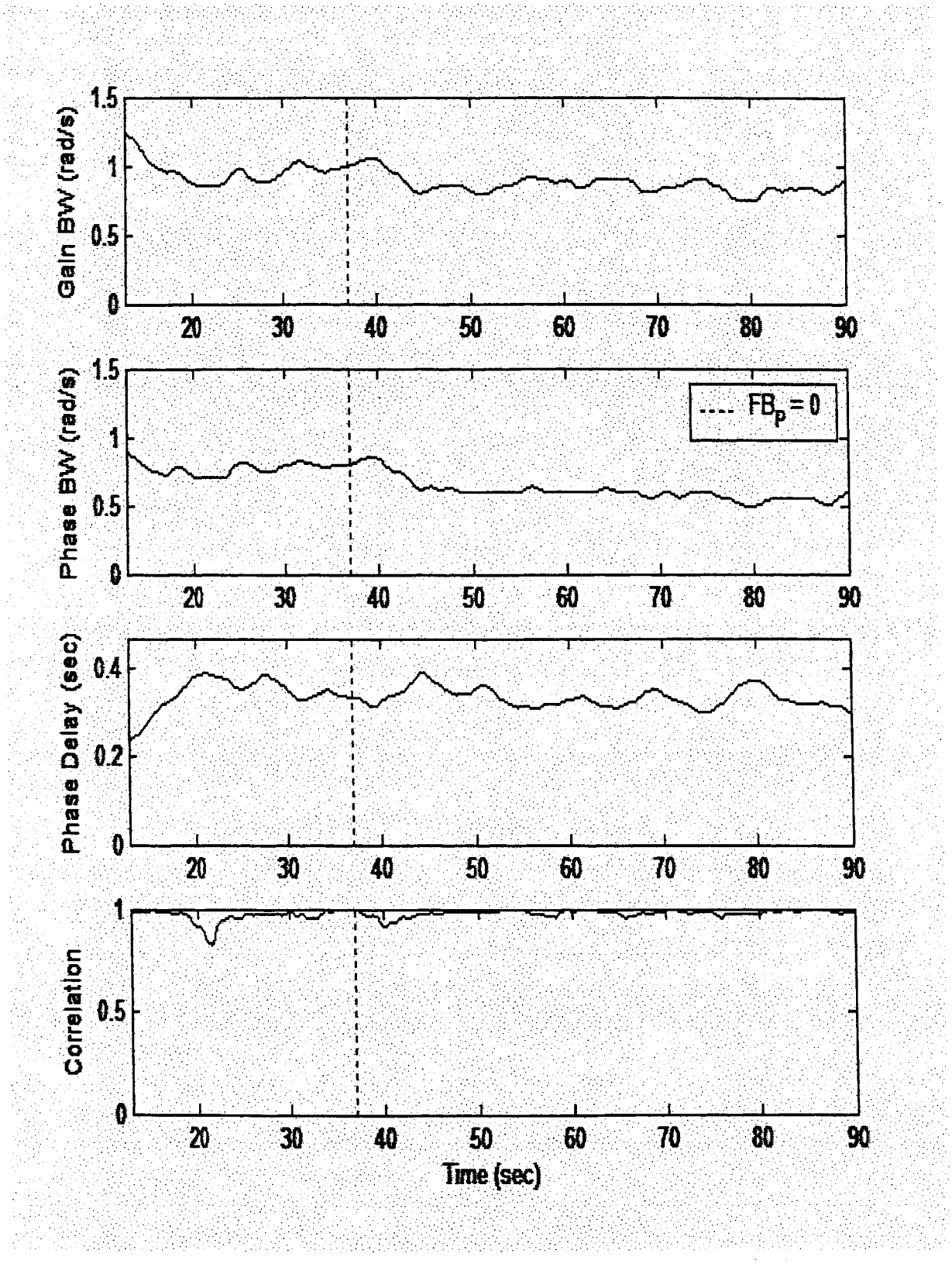


Figure 56. R79 WERA Analysis

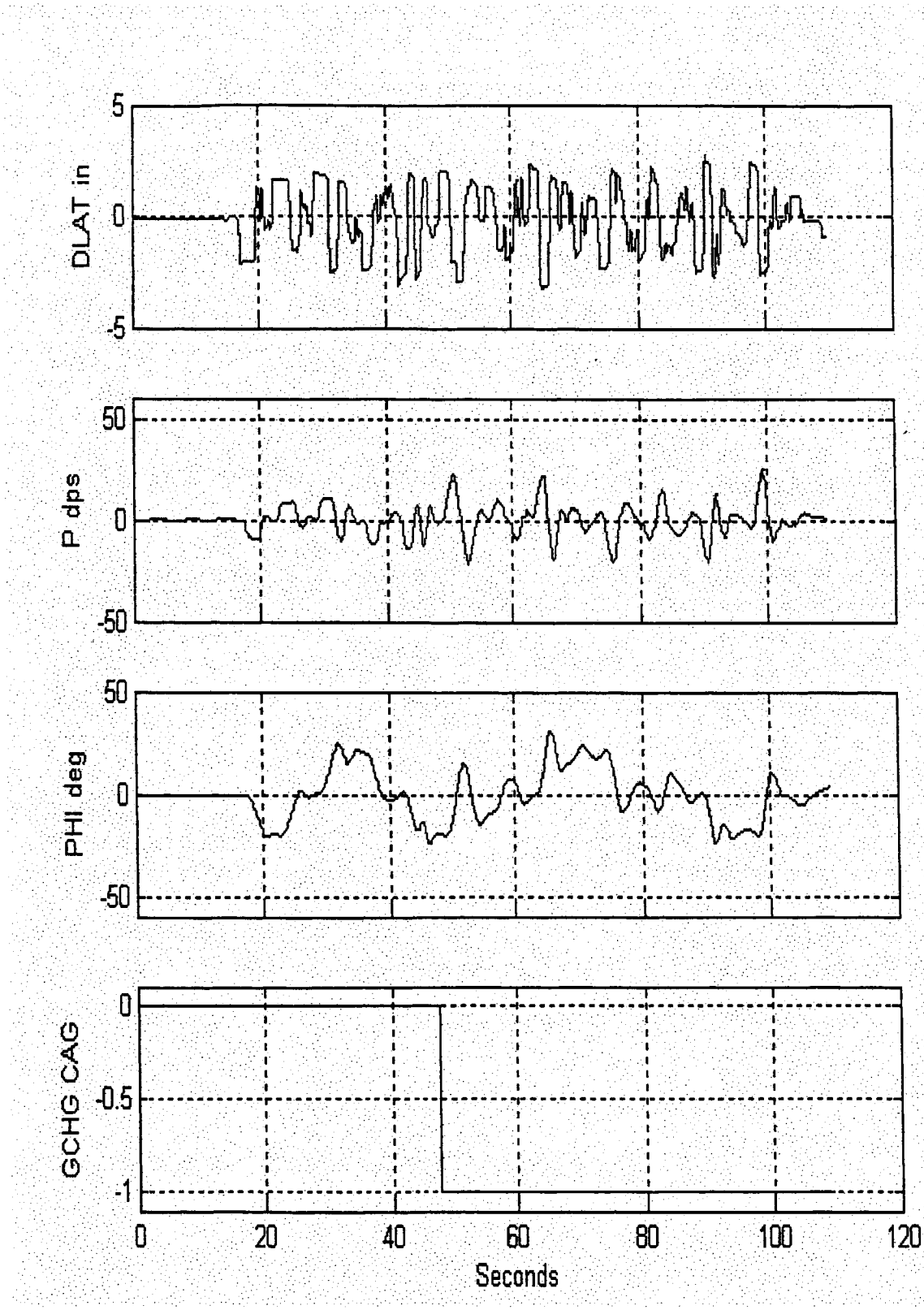
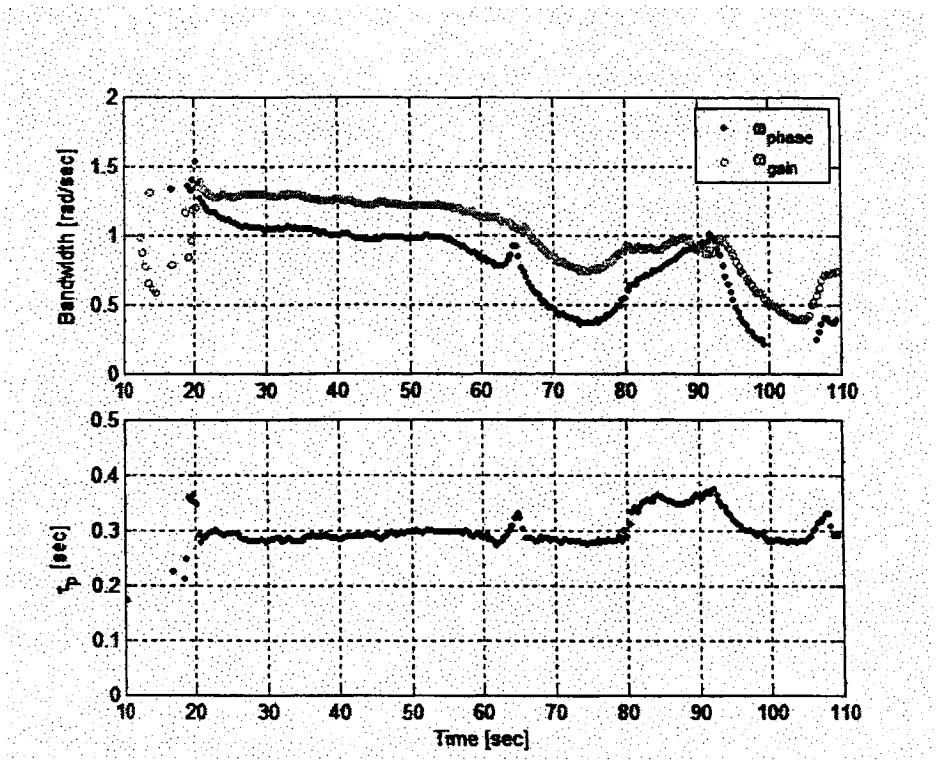
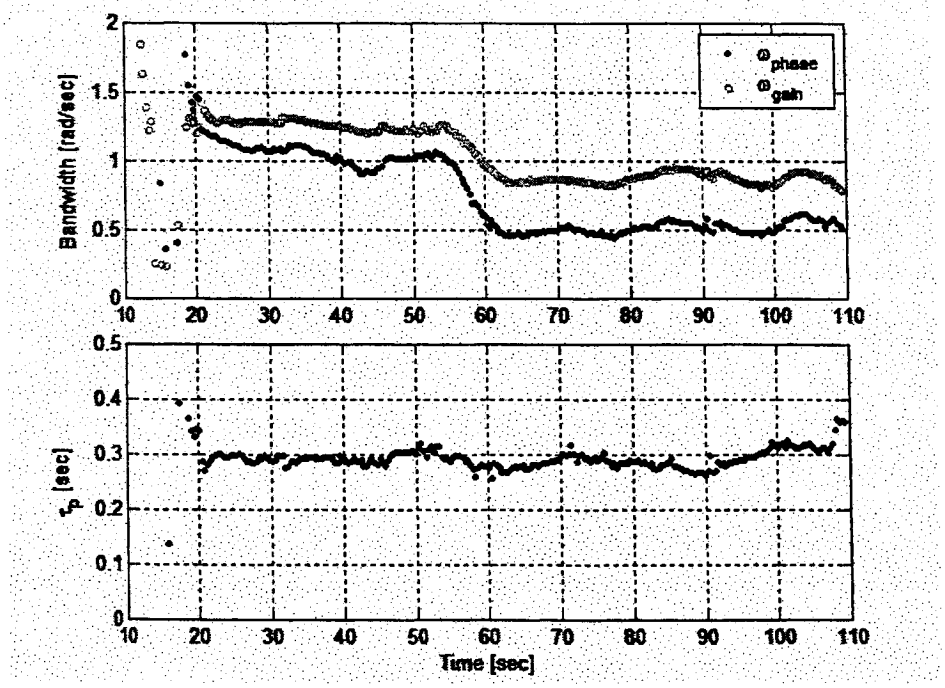


Figure 57. R122 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 58. R122 TVTF Analysis

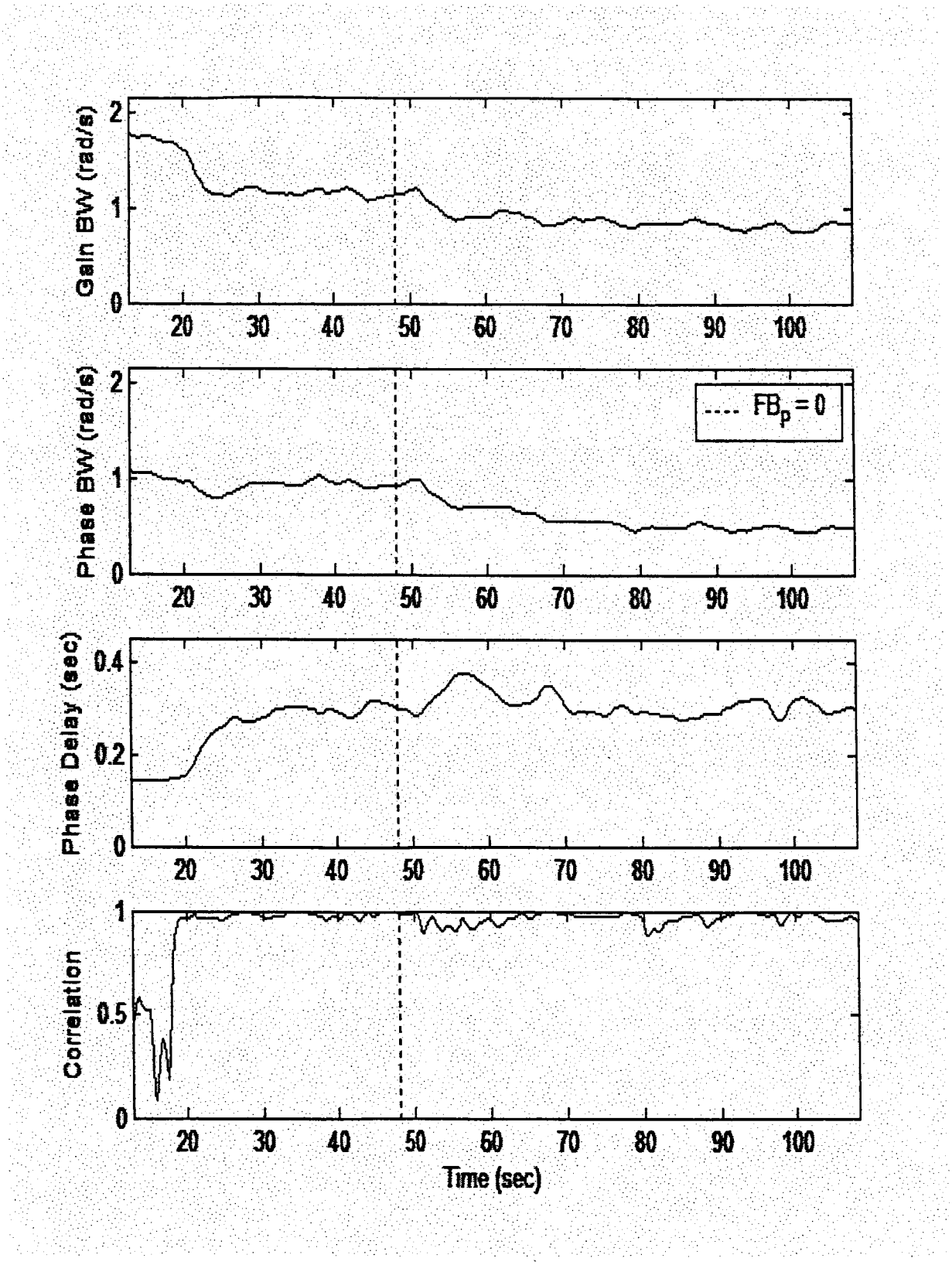


Figure 59. R122 WERA Analysis

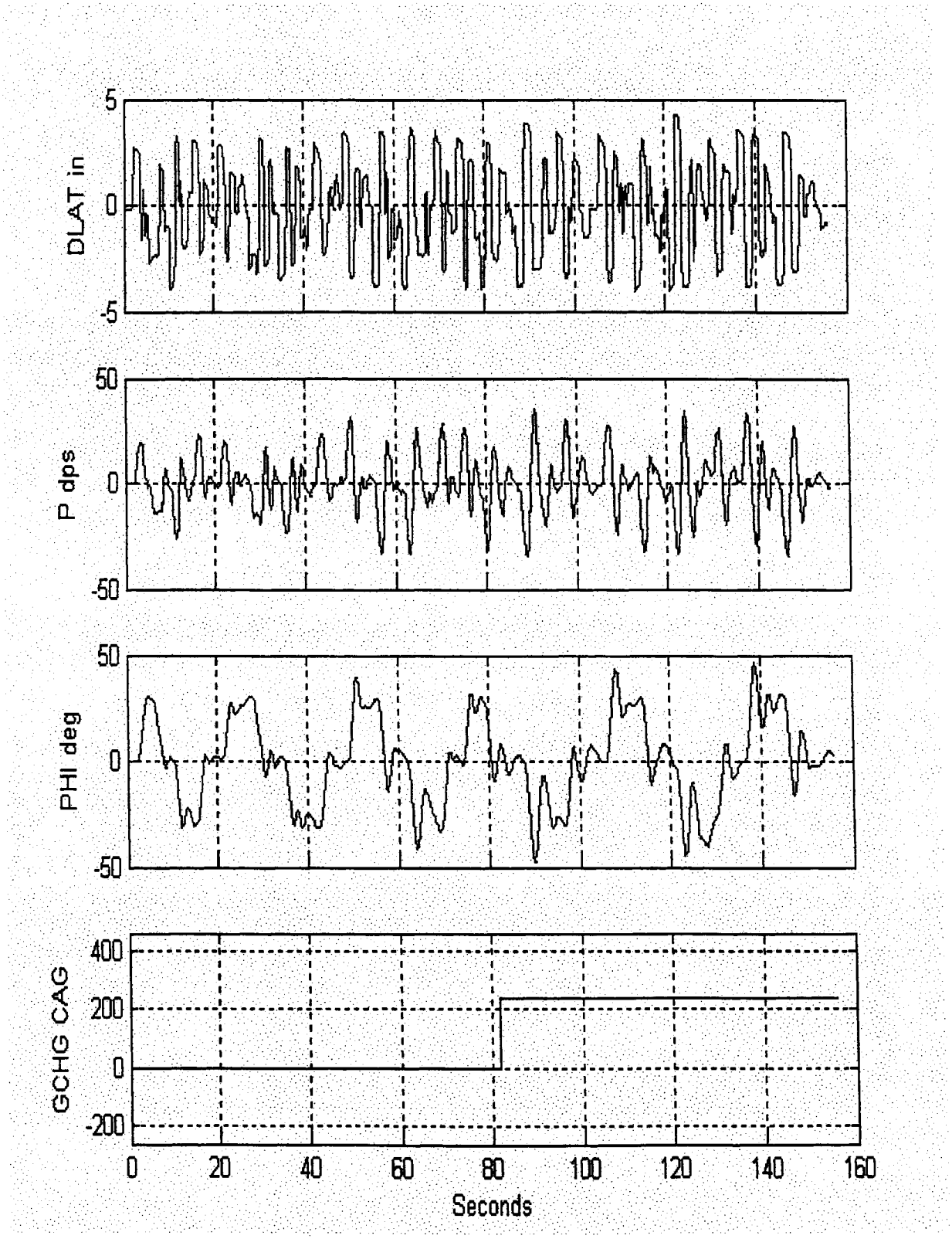
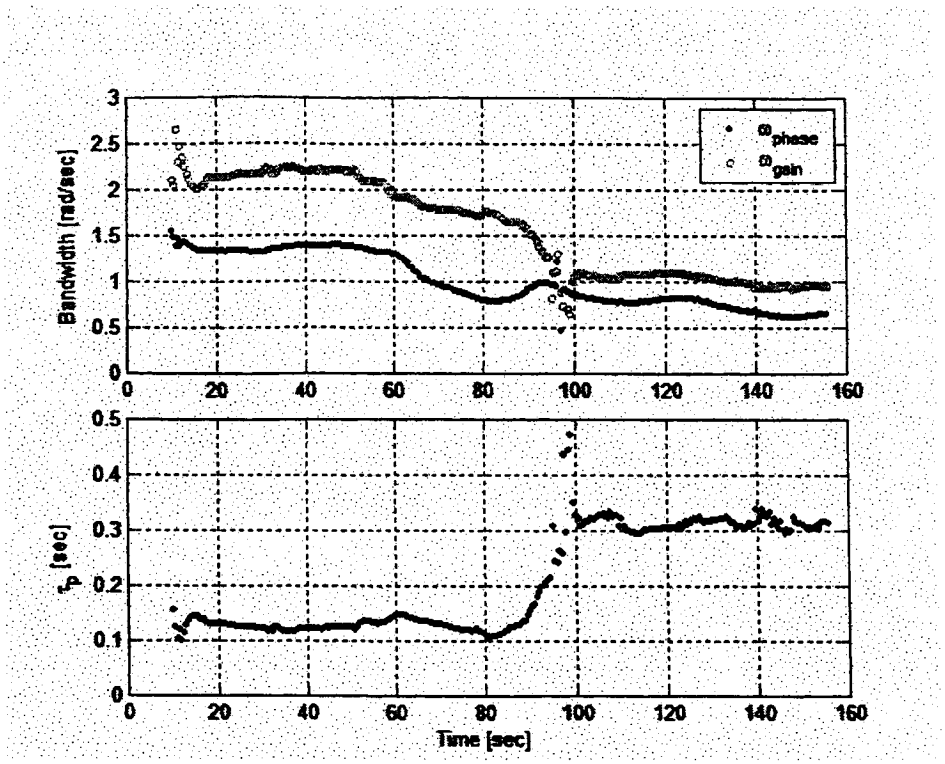
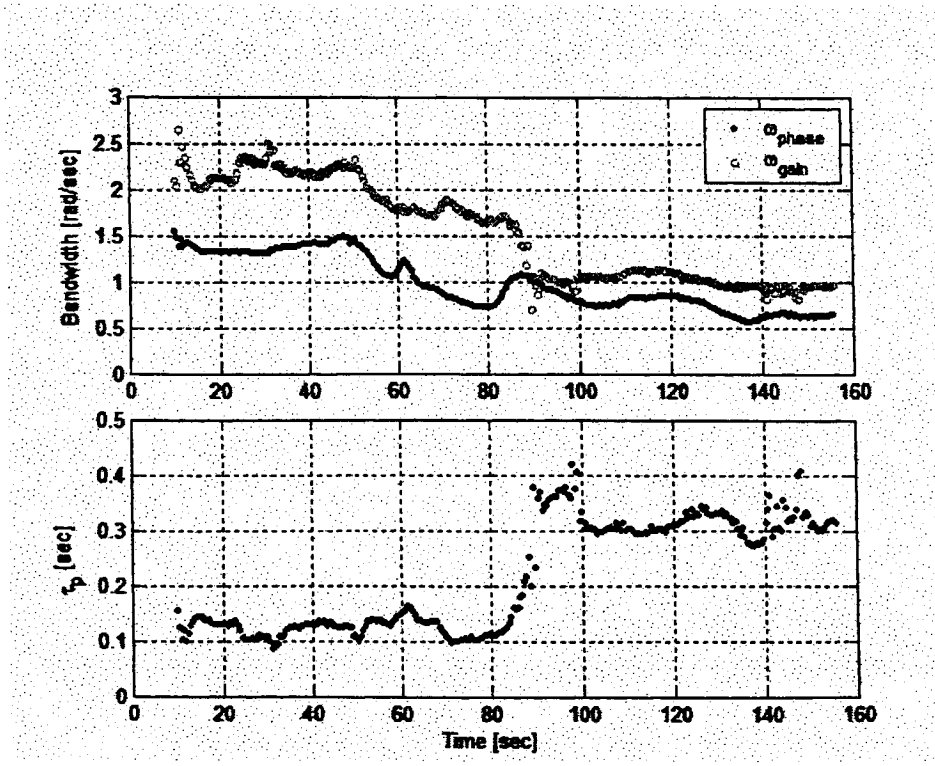


Figure 60. R29 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 61. R29 TVTF Analysis

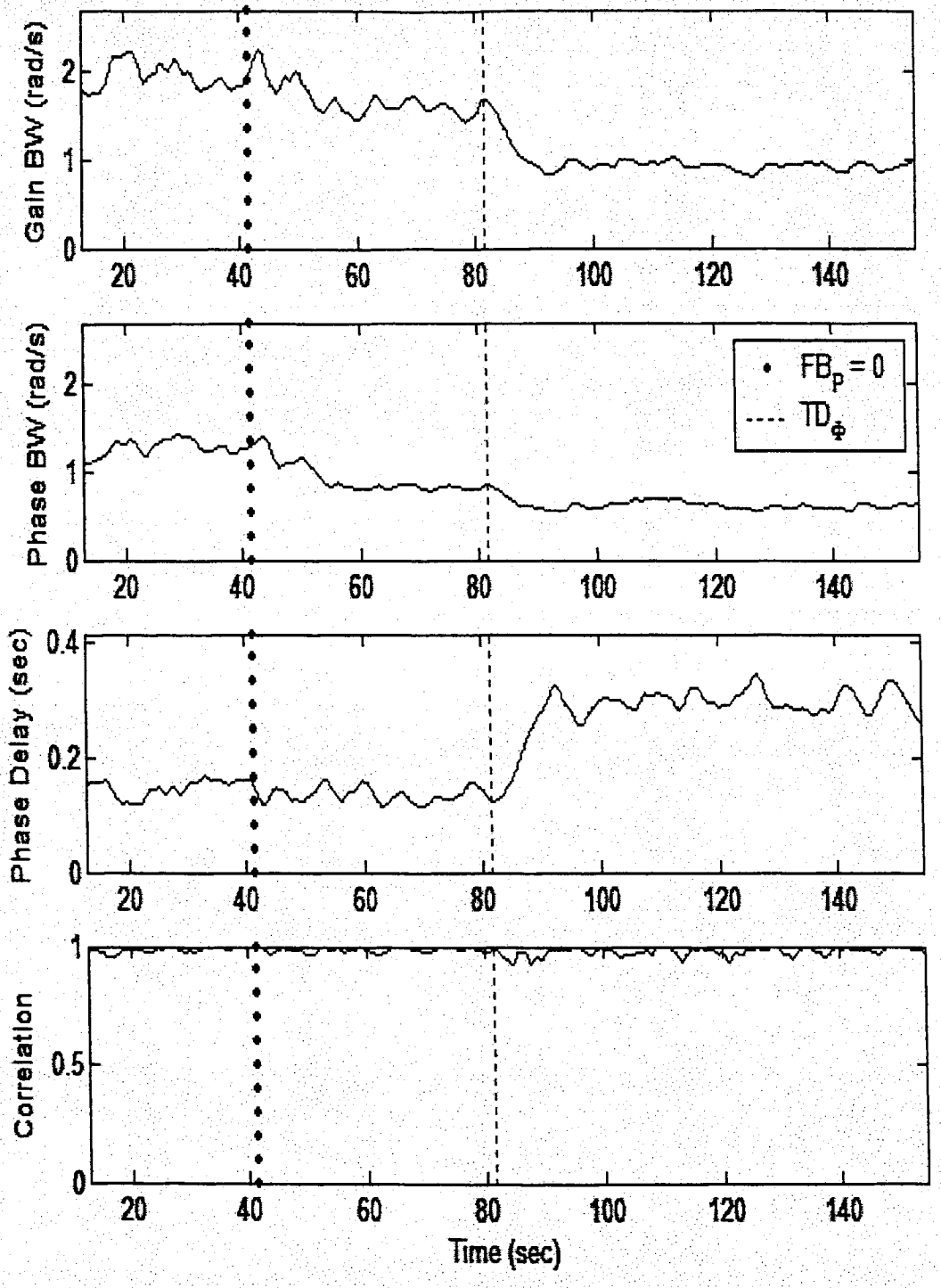


Figure 62. R29 WERA Analysis

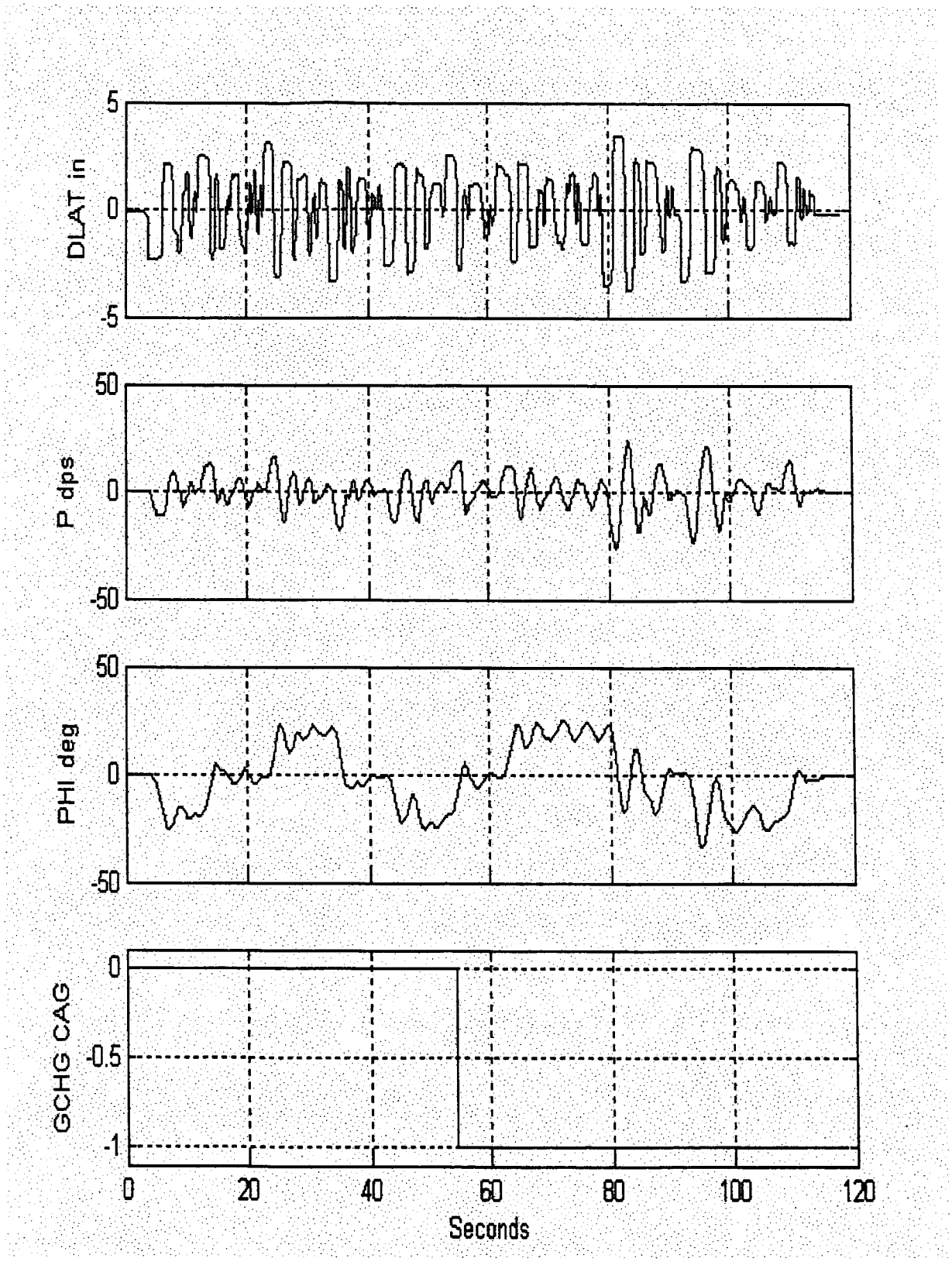
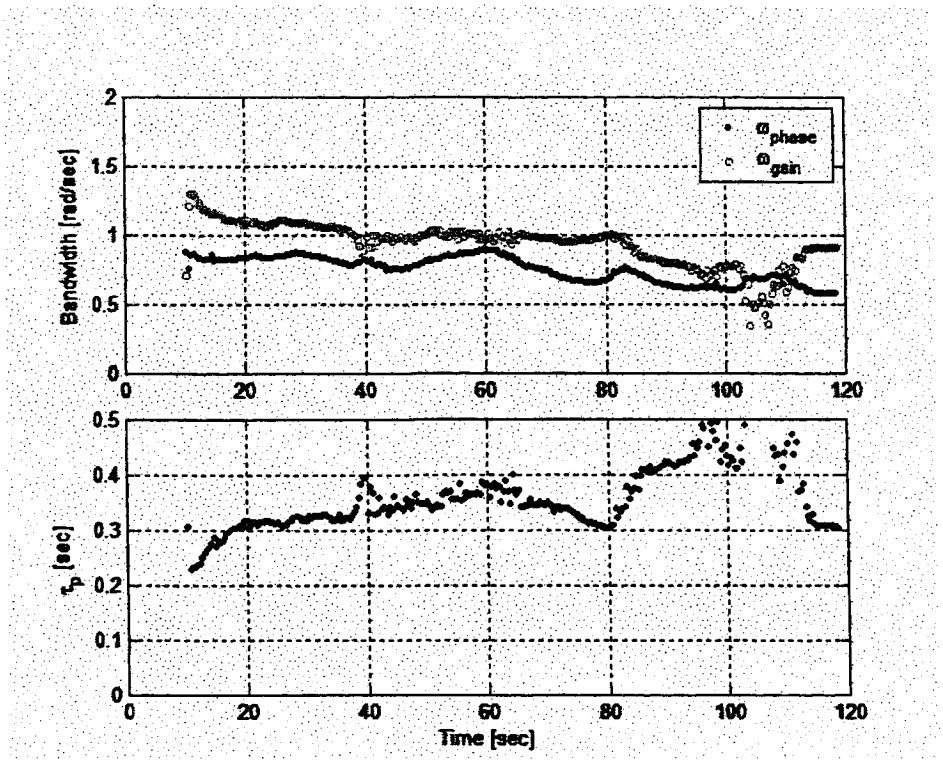
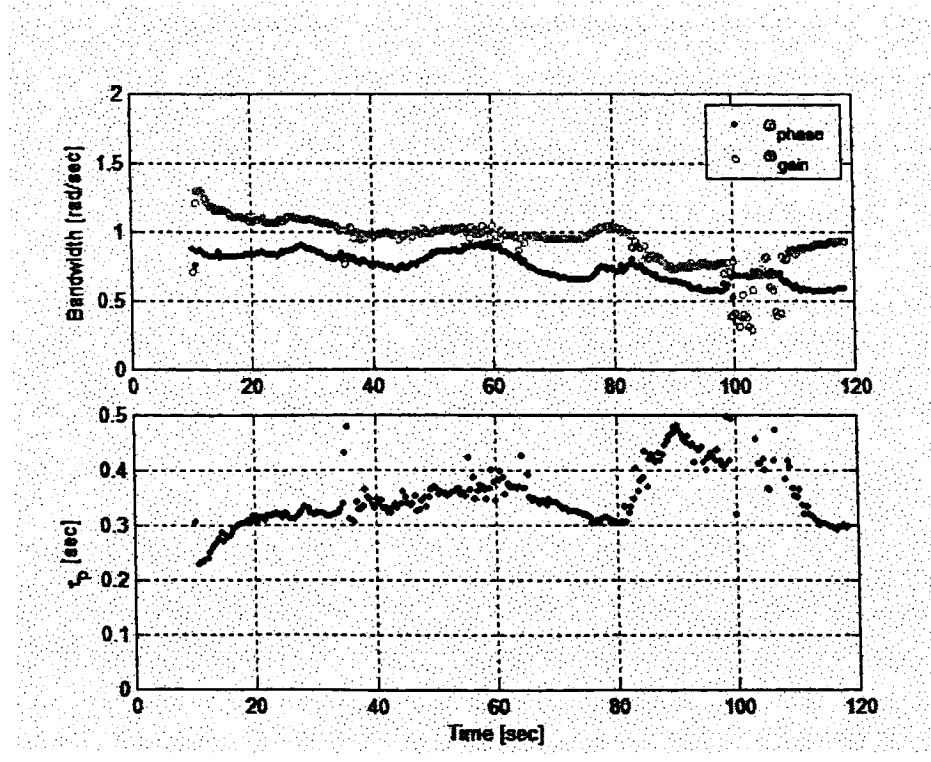


Figure 63. R82 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 31 Point Time Averaging

Figure 64. R82 TVTF Analysis

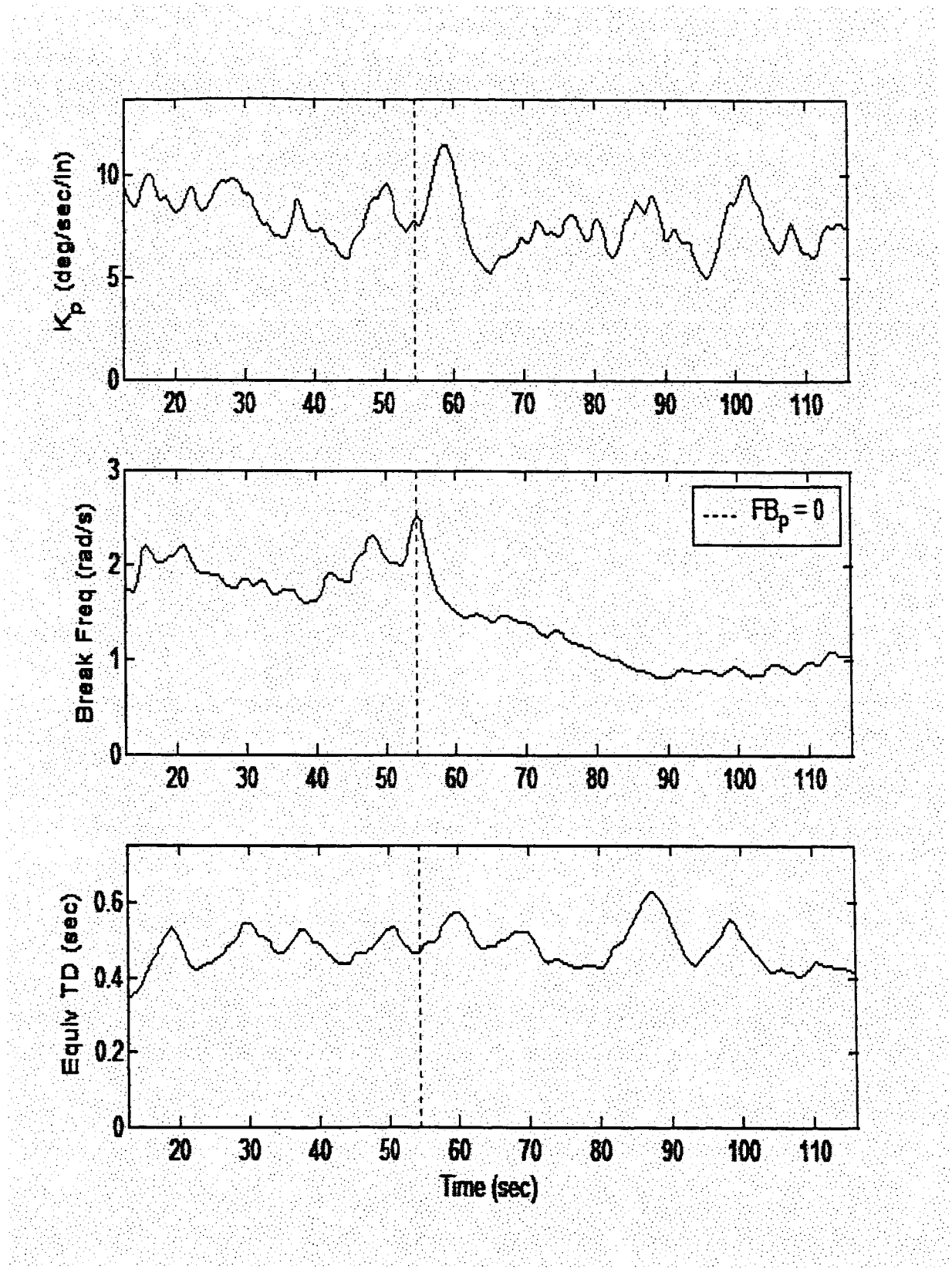


Figure 65. R82 WERA Analysis

4.5 Pitch Attitude Capture and Hold Evaluations

A listing of the key pitch attitude capture and hold evaluation runs is provided in Table 16. Time histories and time varying transfer function (TVTF) and wavelet eigen value realization (WERA) analysis plots for the Table 16 runs are provided in Figure 66 through Figure 92.

Table 16. Selected PACH Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R68	CM	CLA	PACH	n/a	No change
Mild PIO Runs					
R95	CM	CLA	PACH	QLIM	Oscillations after failure
R96	CM	CLA	PACH	QLIM	PitchTD whole run, some PIO
R115	CH	CLA	PACH	QGain	Three cycle PIO following failure
R117	CH	CLA	PACH	QLIM	PitchTD whole run, small amplitude but sustained PIO
Sustained or Severe PIO Runs					
R38	CM	CLA	PACH	QGain	PitchTD from start, autothrottle disengage triggers sustained PIO
R69	CM	CLA	PACH	QGain	PitchTD whole run, PIO
R94	CM	CLA	PACH	QGain	PitchTD whole run, significant PIO after failure
R116	CH	CLA	PACH	QGain	PitchTD whole run, PIO

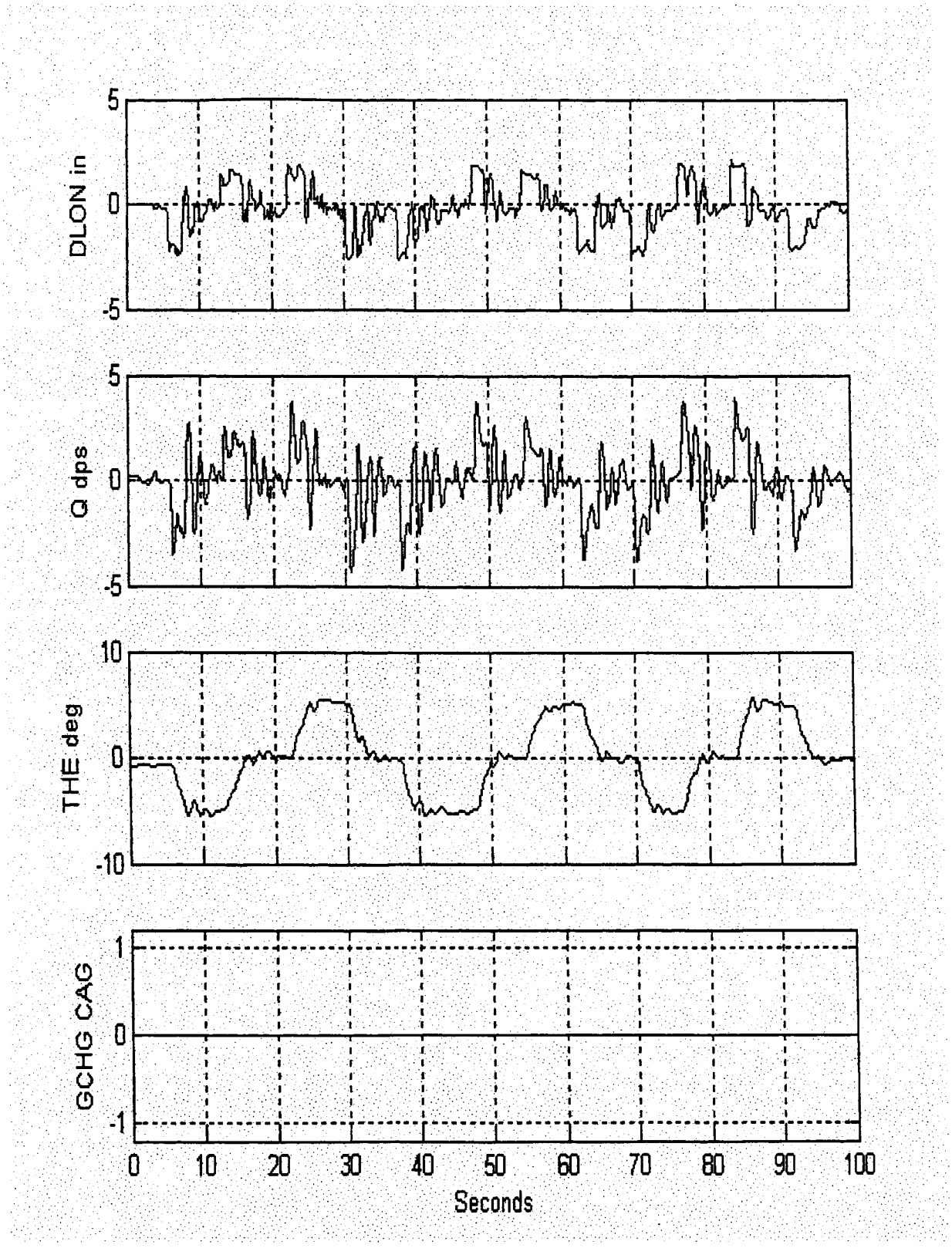
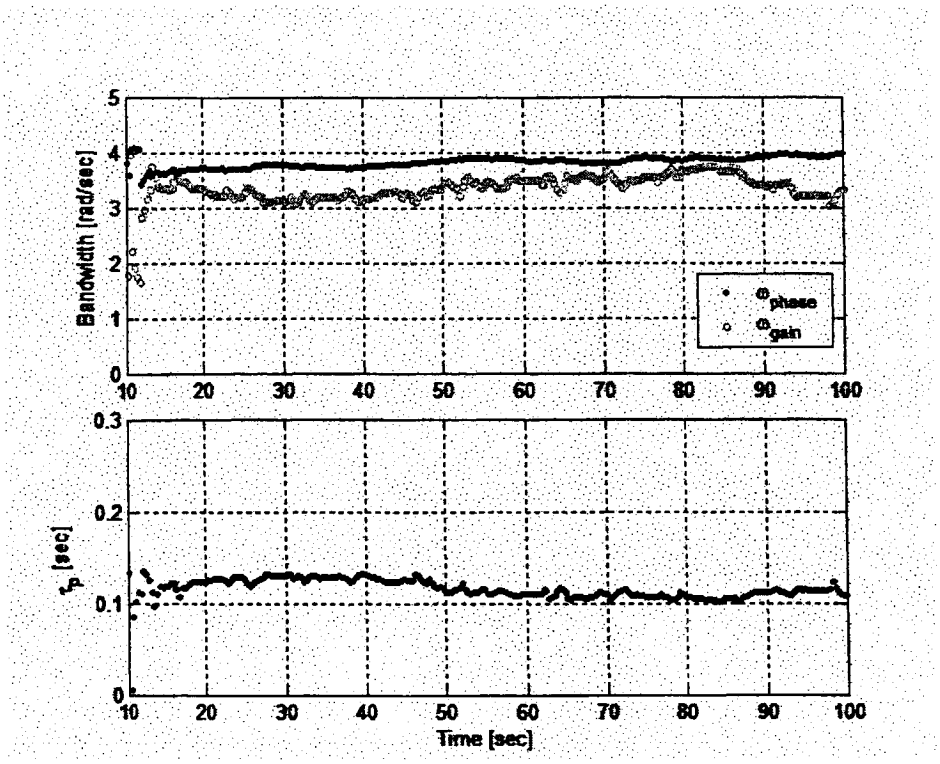
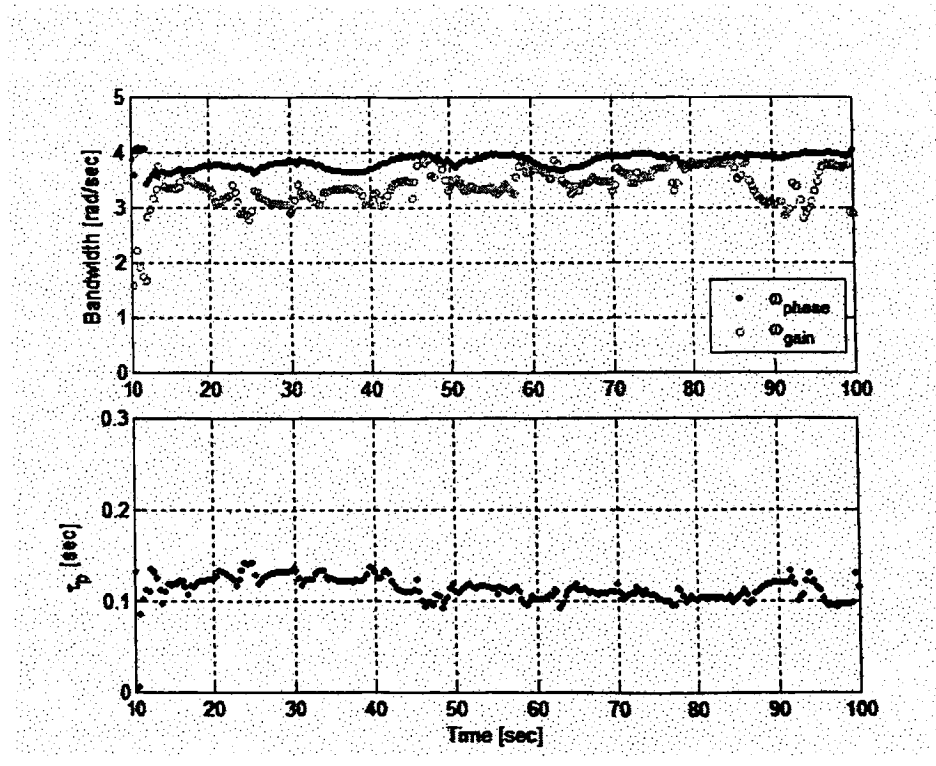


Figure 66. R68 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 67. R68 TVTF Analysis

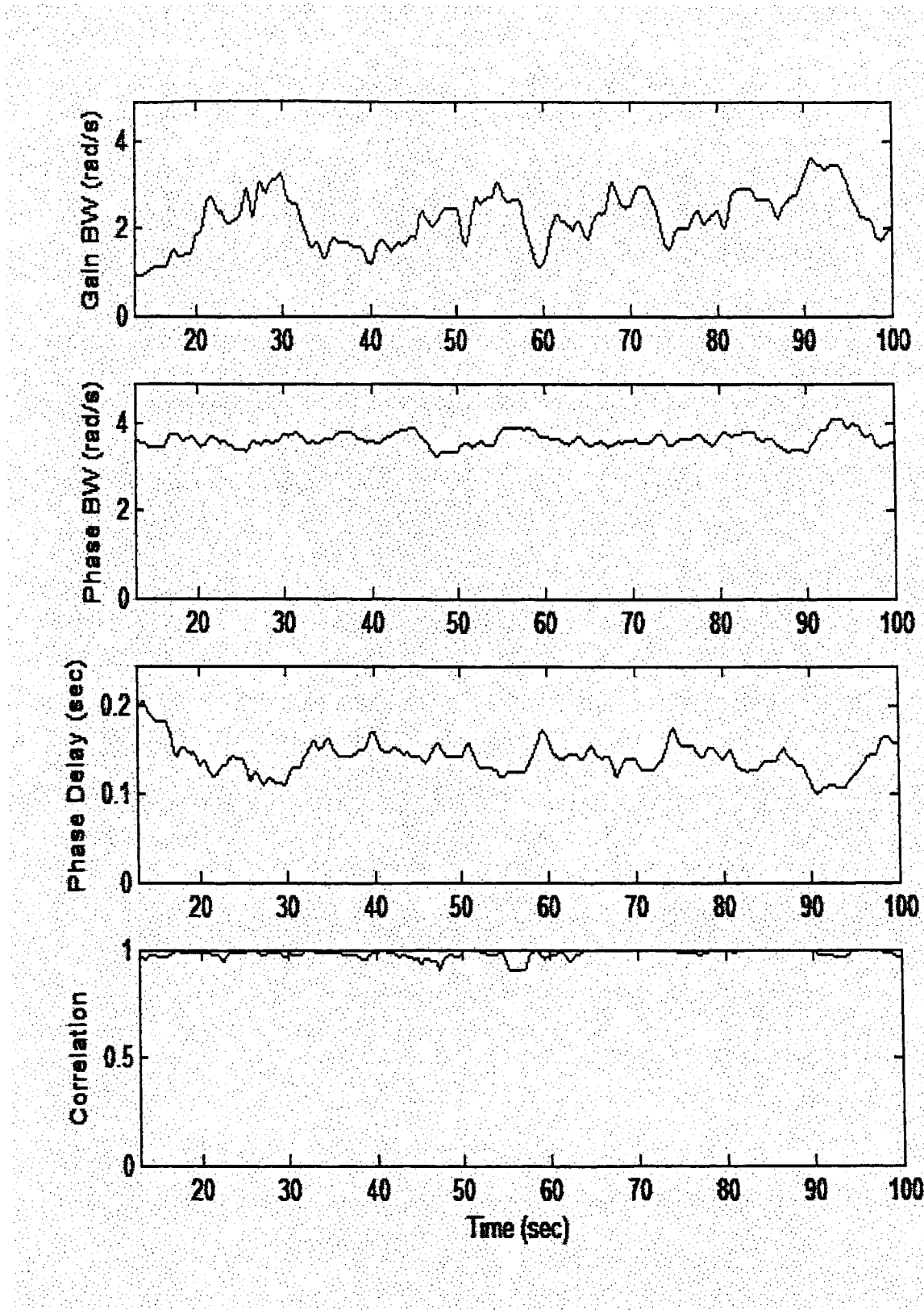


Figure 68. R68 WERA Analysis

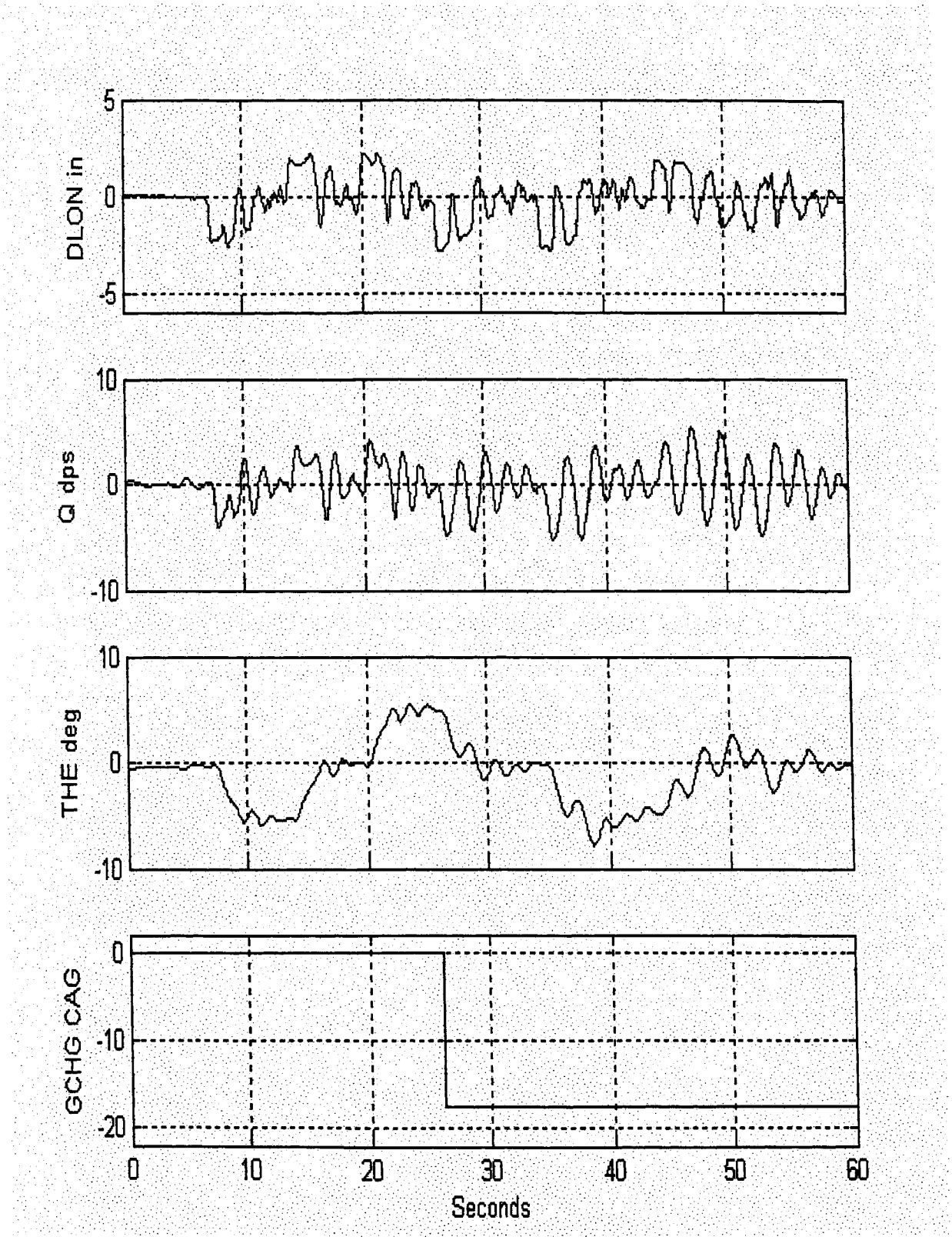
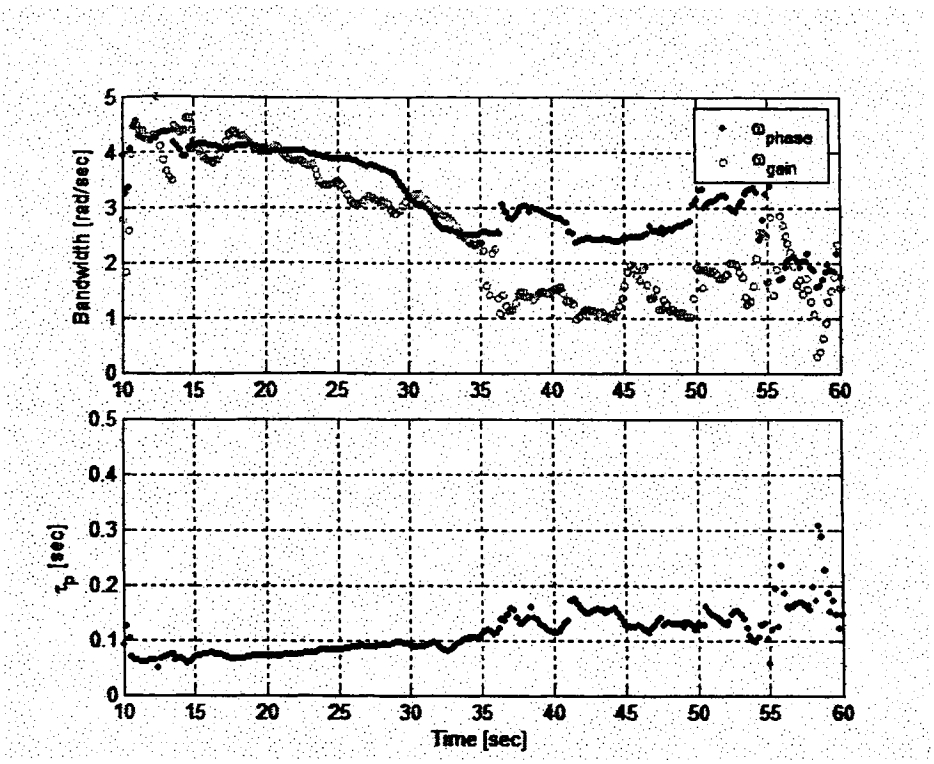
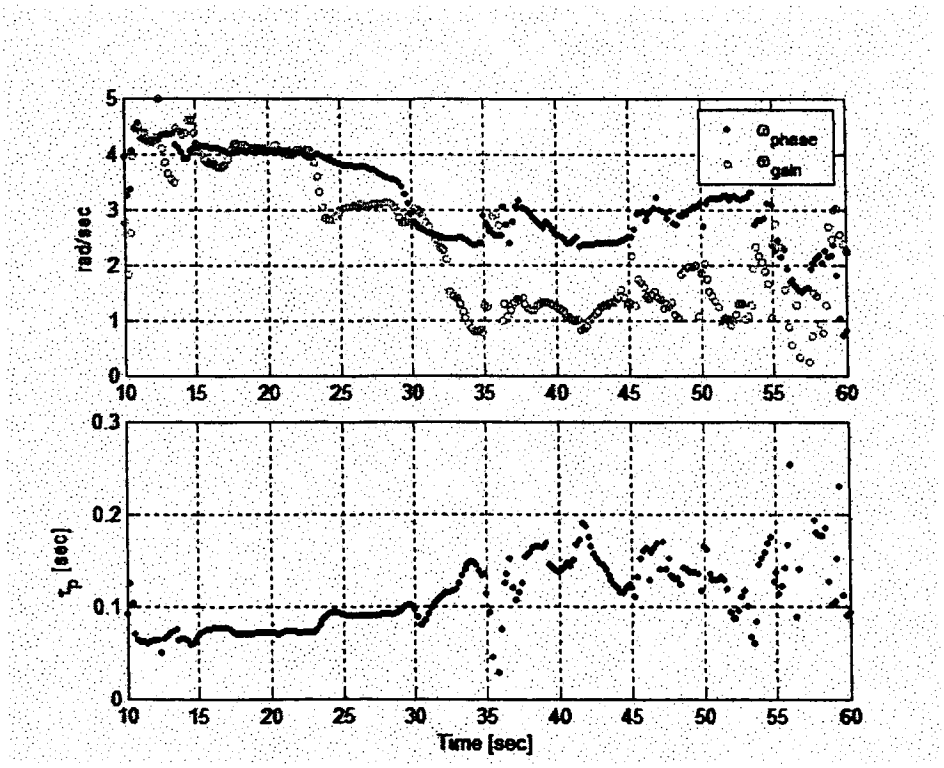


Figure 69. R95 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 31 Point Time Averaging

Figure 70. R95 TVTF Analysis

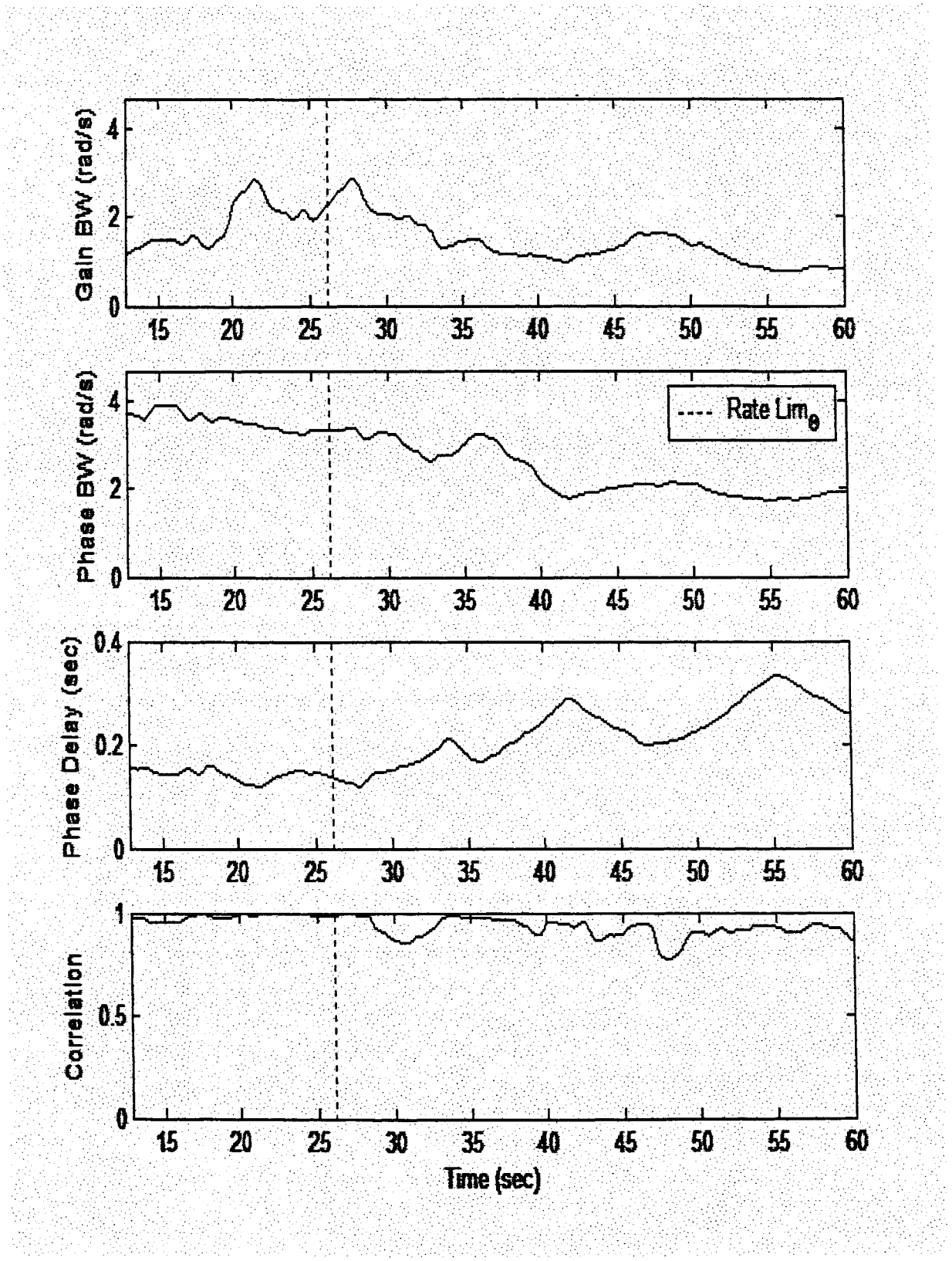


Figure 71. R95 WERA Analysis

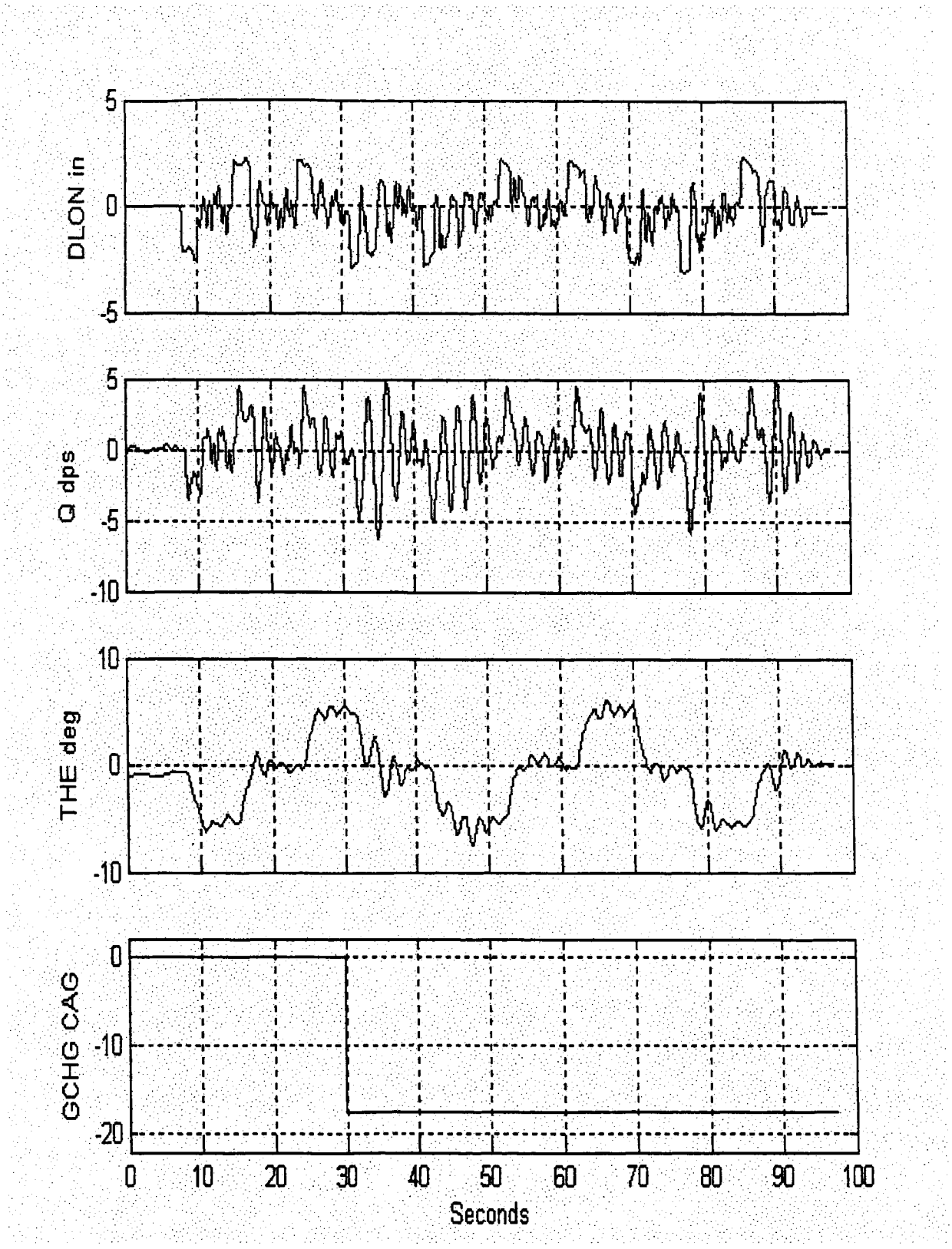
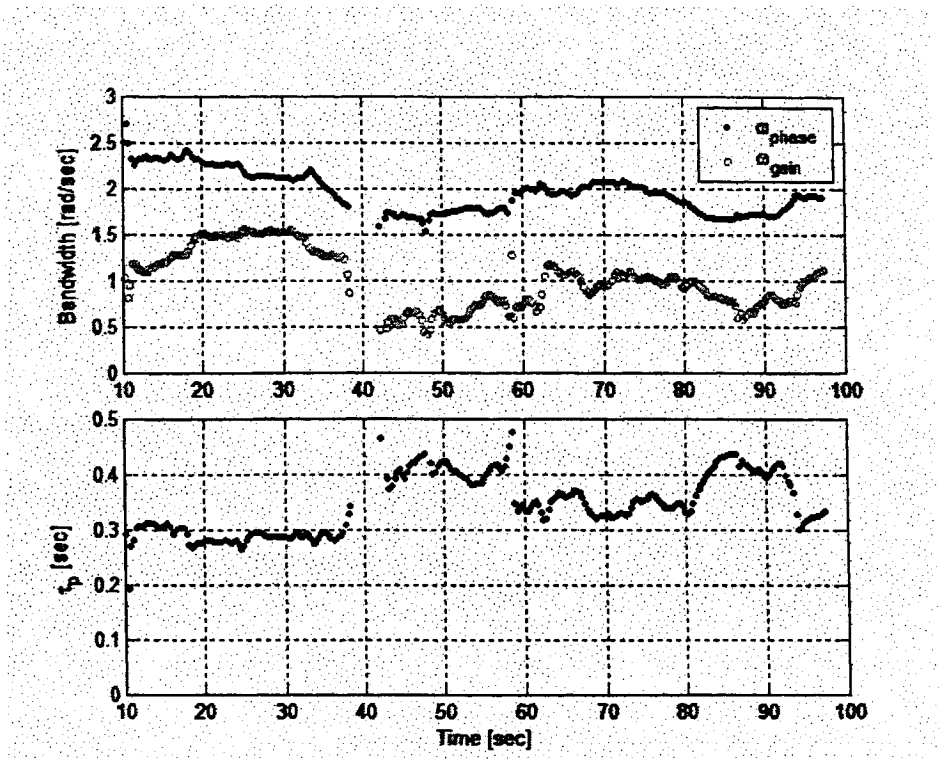
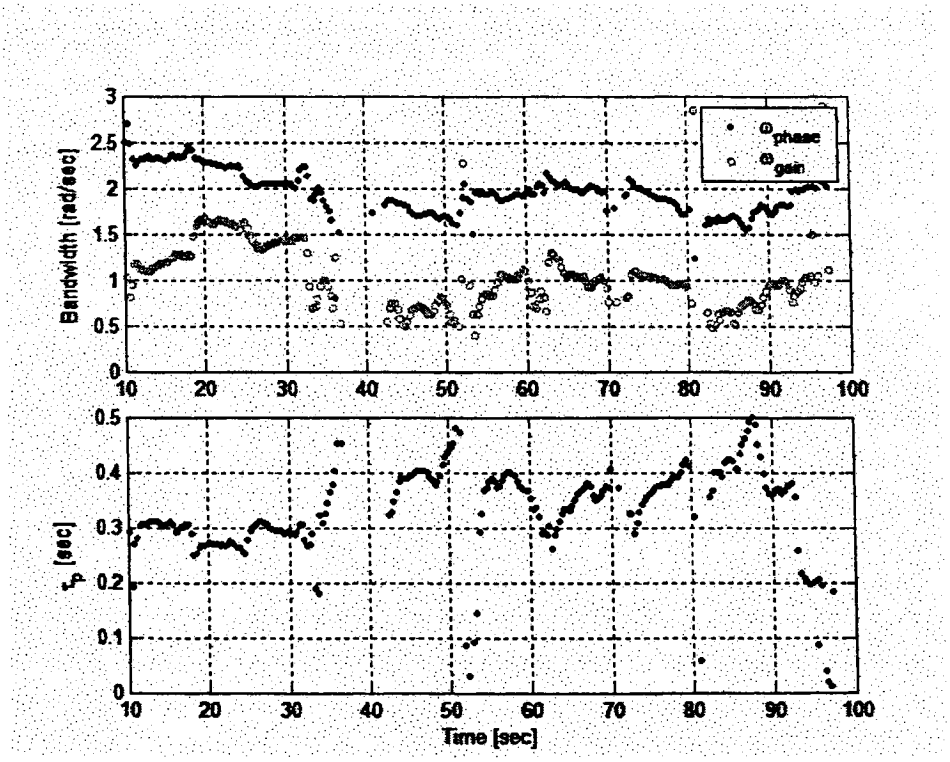


Figure 72. R96 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 73. R96 TVTF Analysis

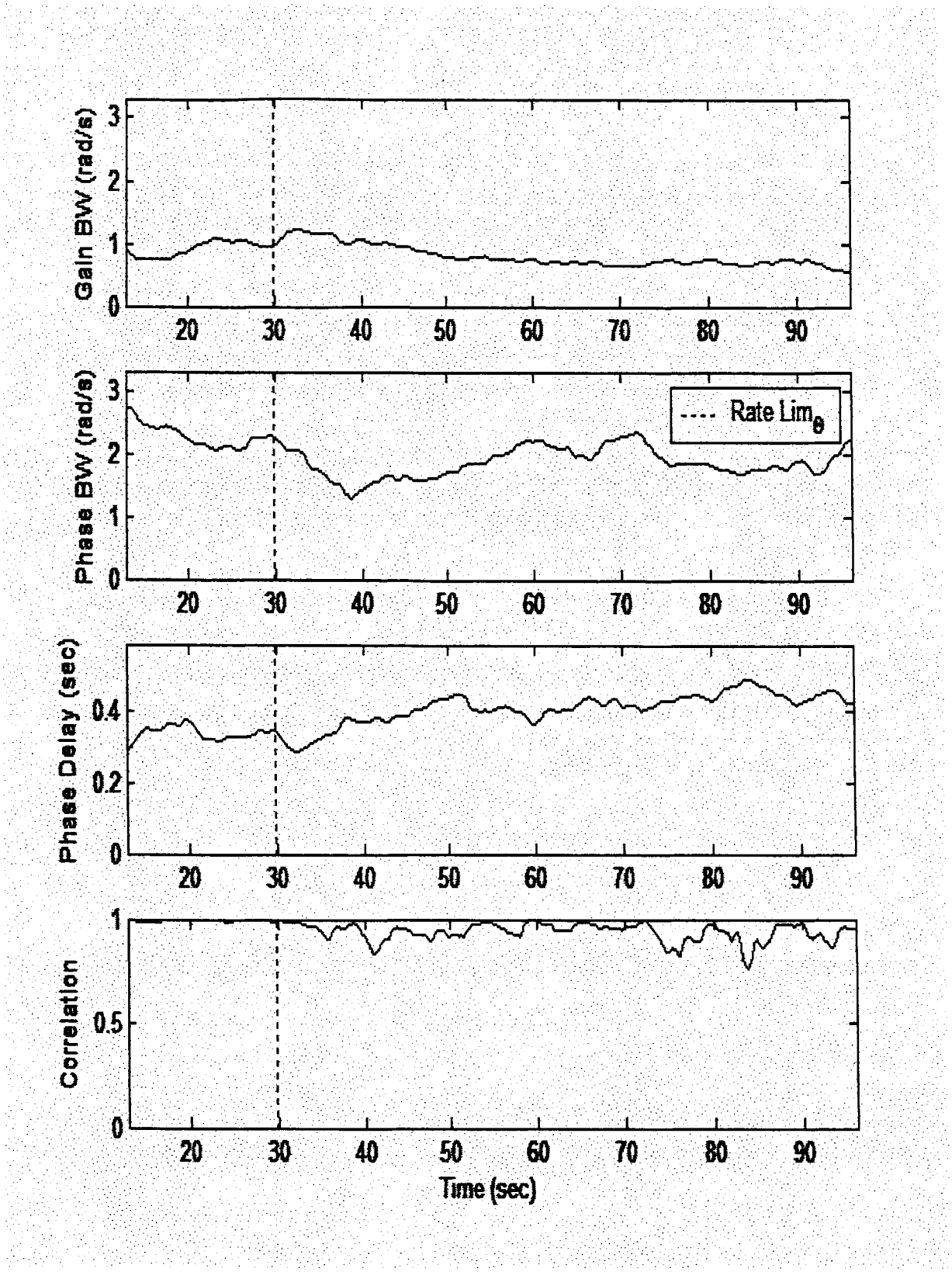


Figure 74. R96 WERA Analysis

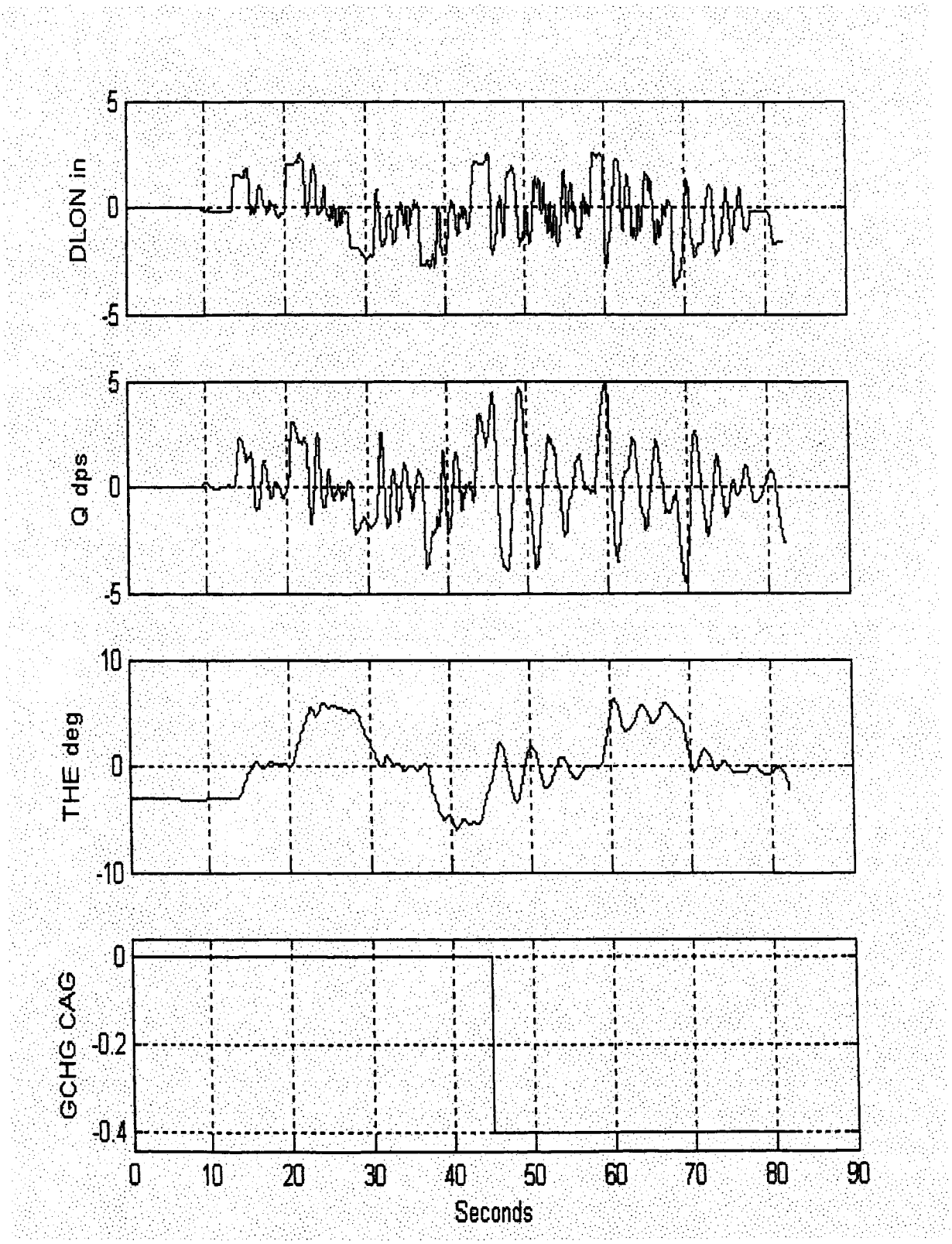
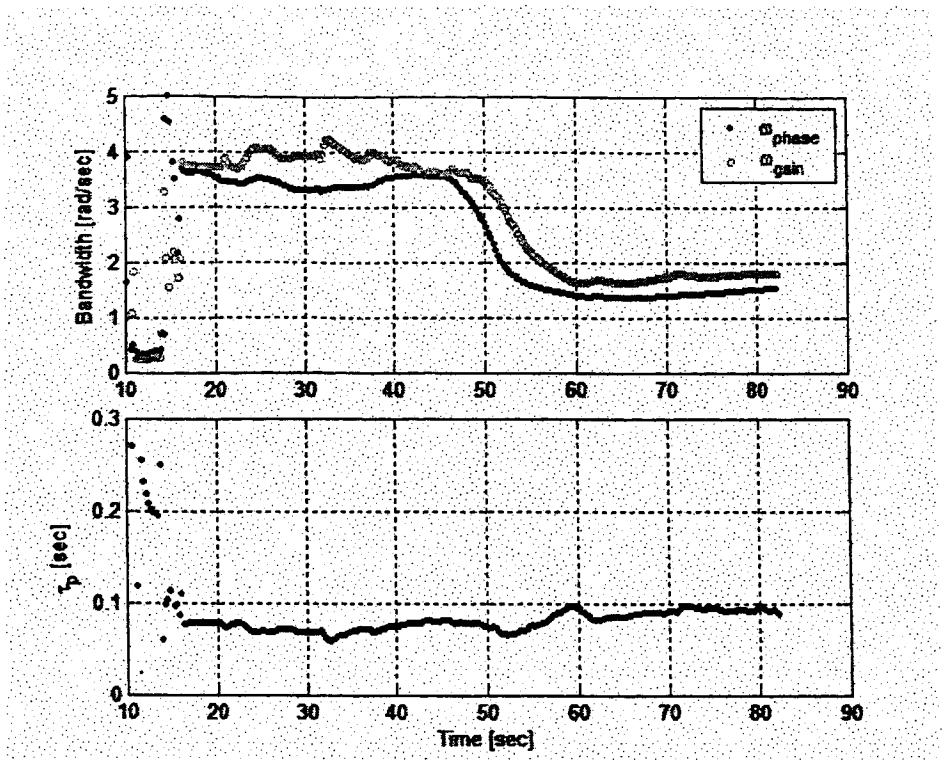
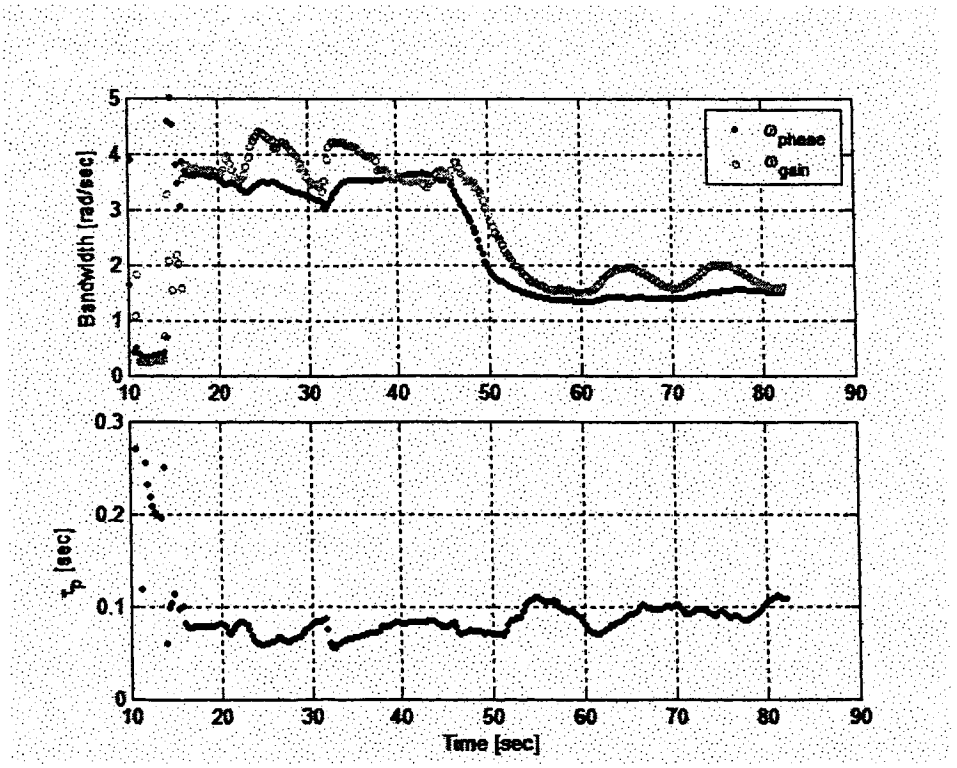


Figure 75. R115 Time Histories, Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 76. R115 TVTF Analysis

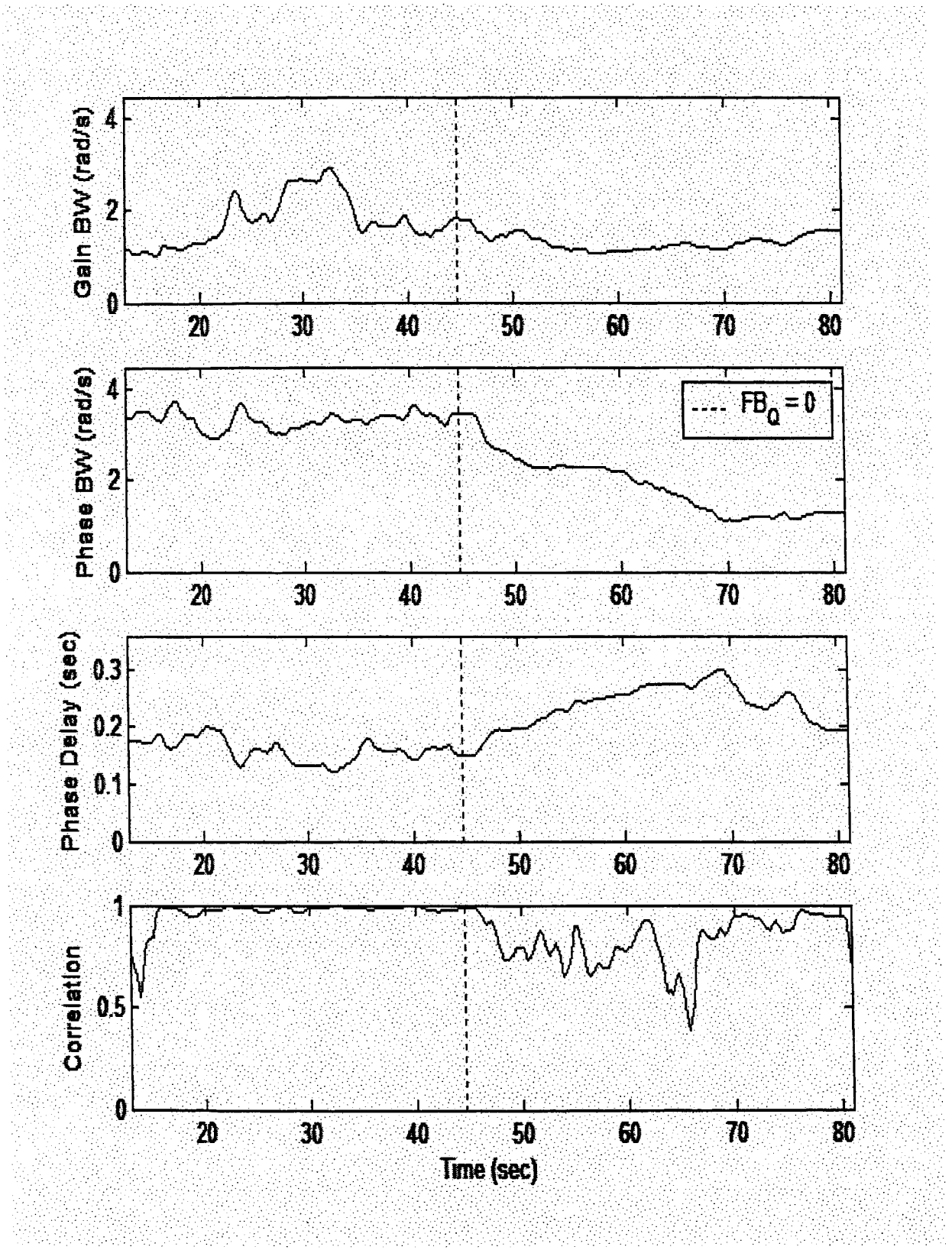


Figure 77. R115 WERA Analysis

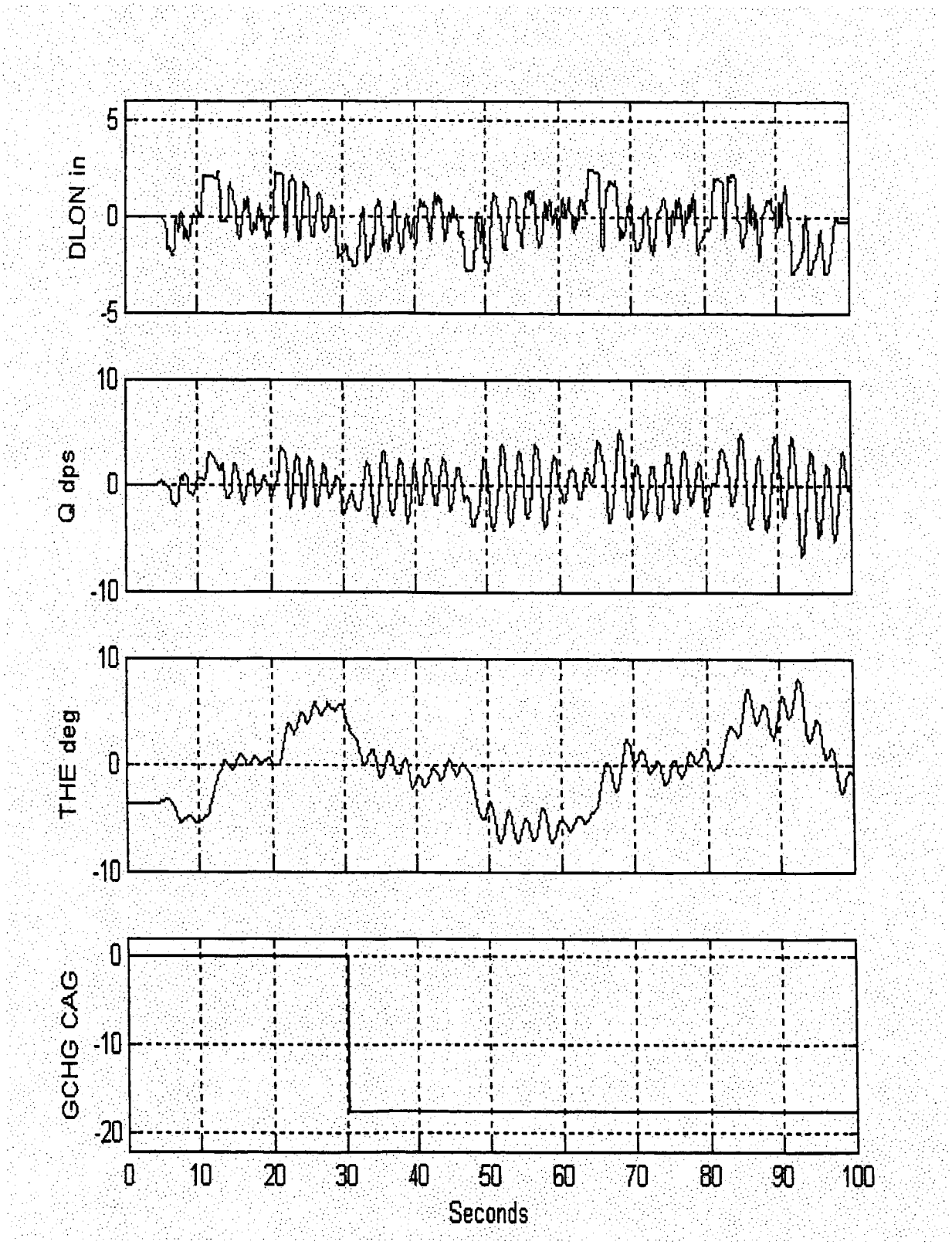
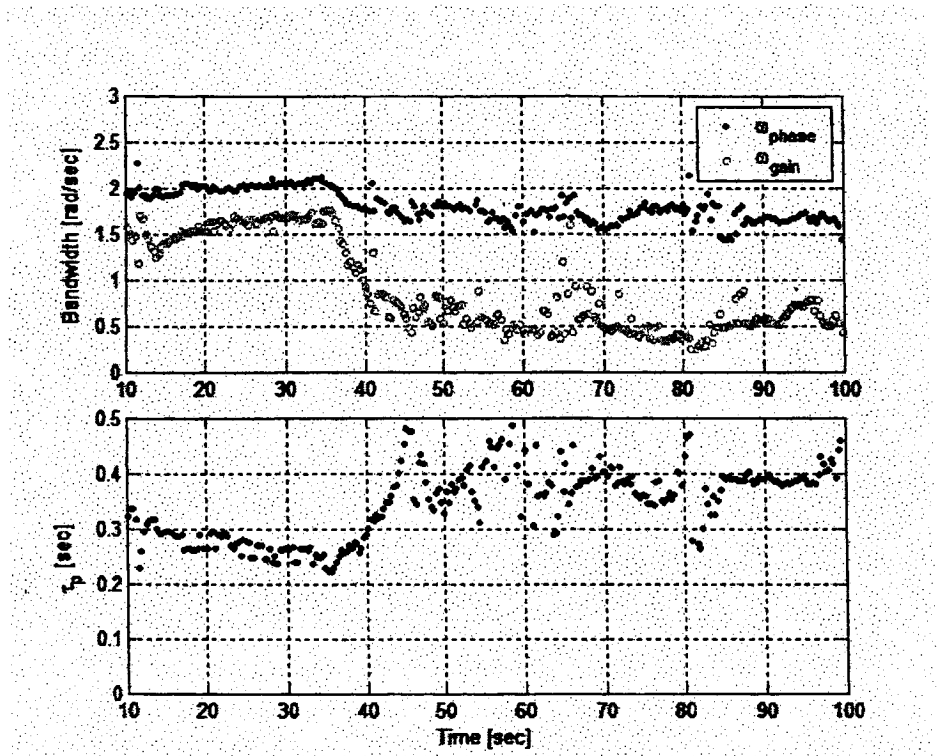
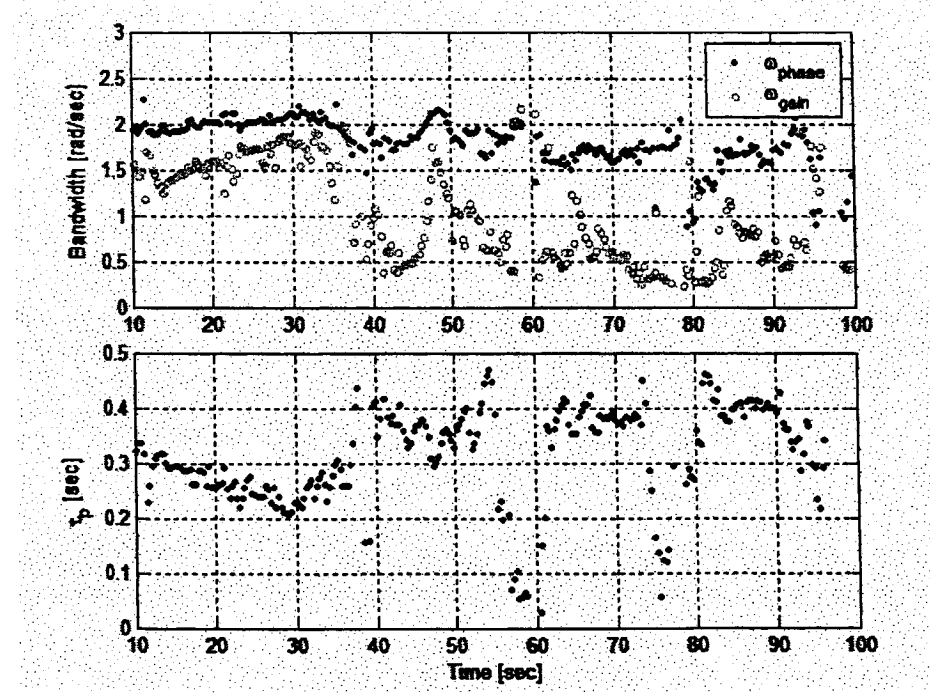


Figure 78. R117 Time Histories Mild PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 79. R117 TVTF Analysis

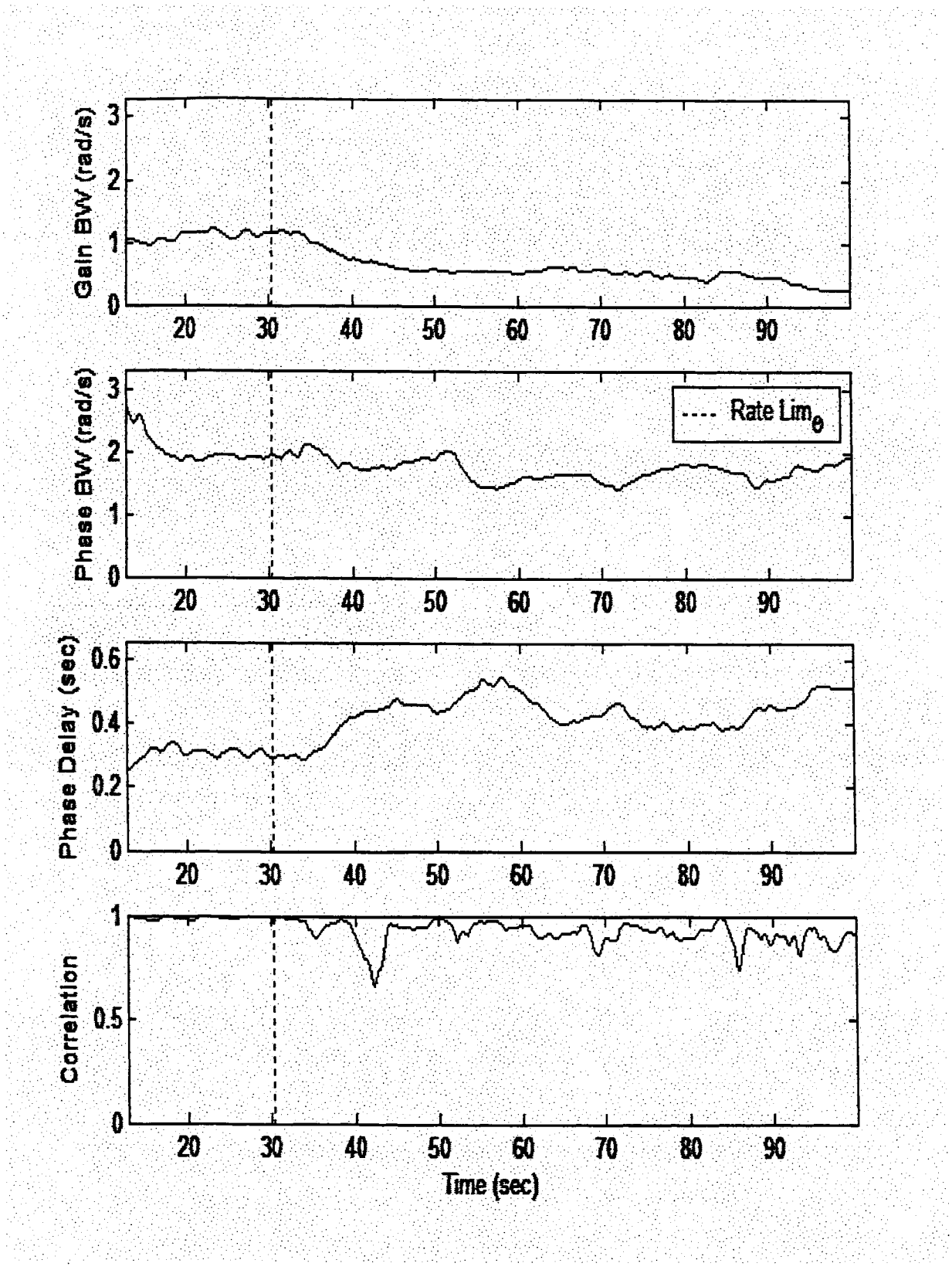


Figure 80. R117 WERA Analysis

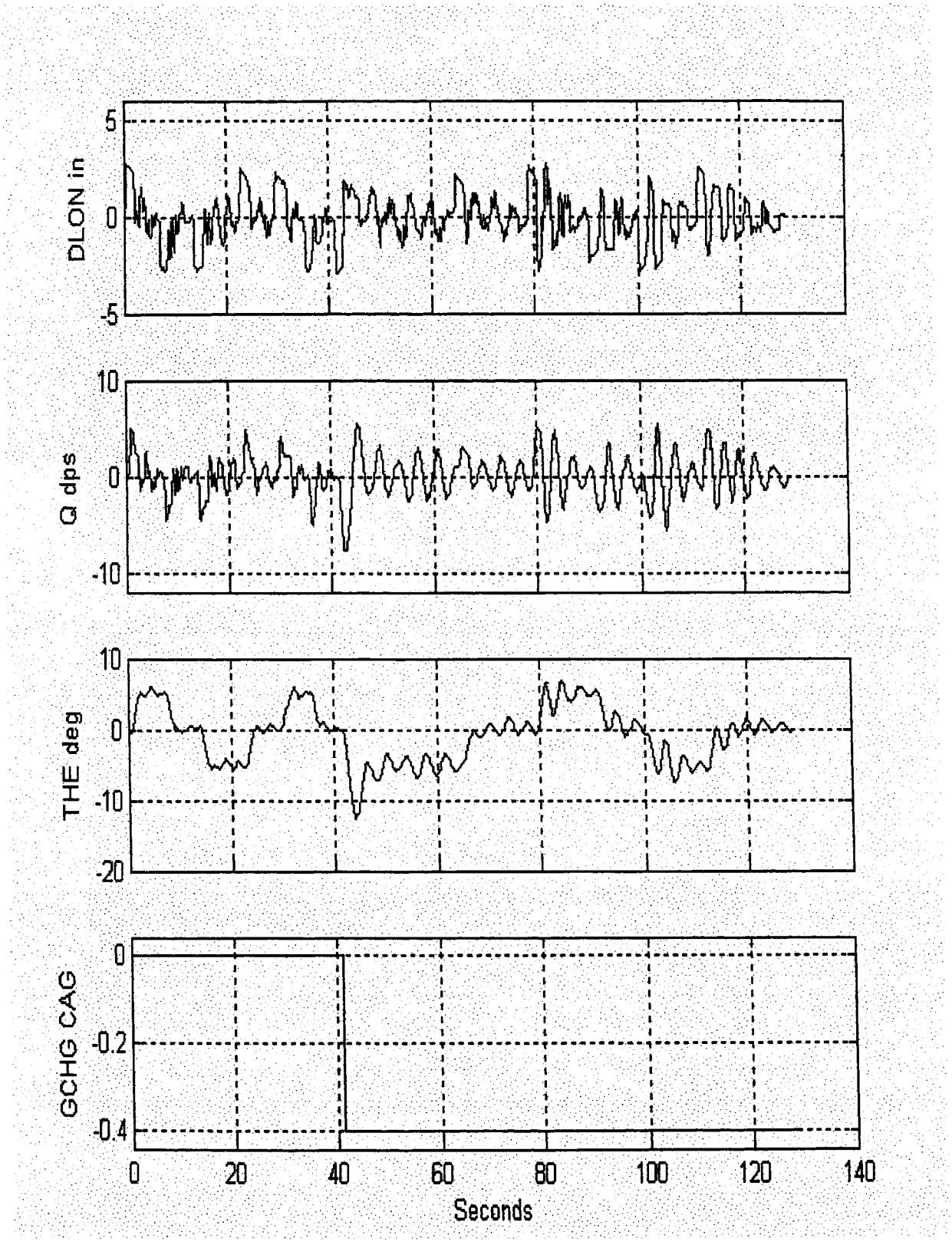
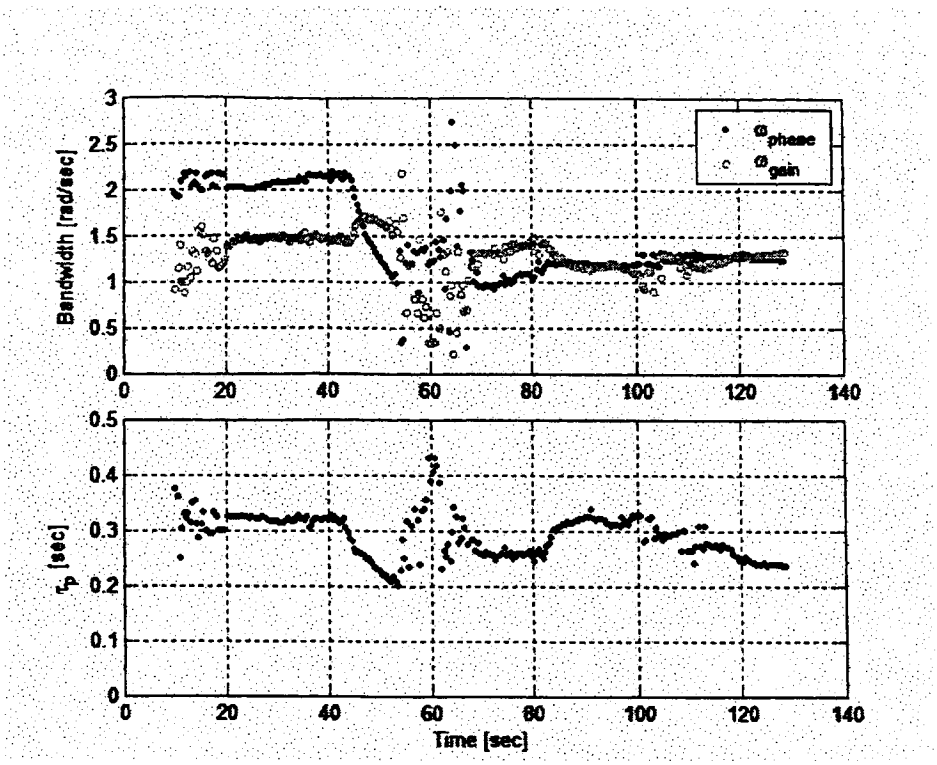
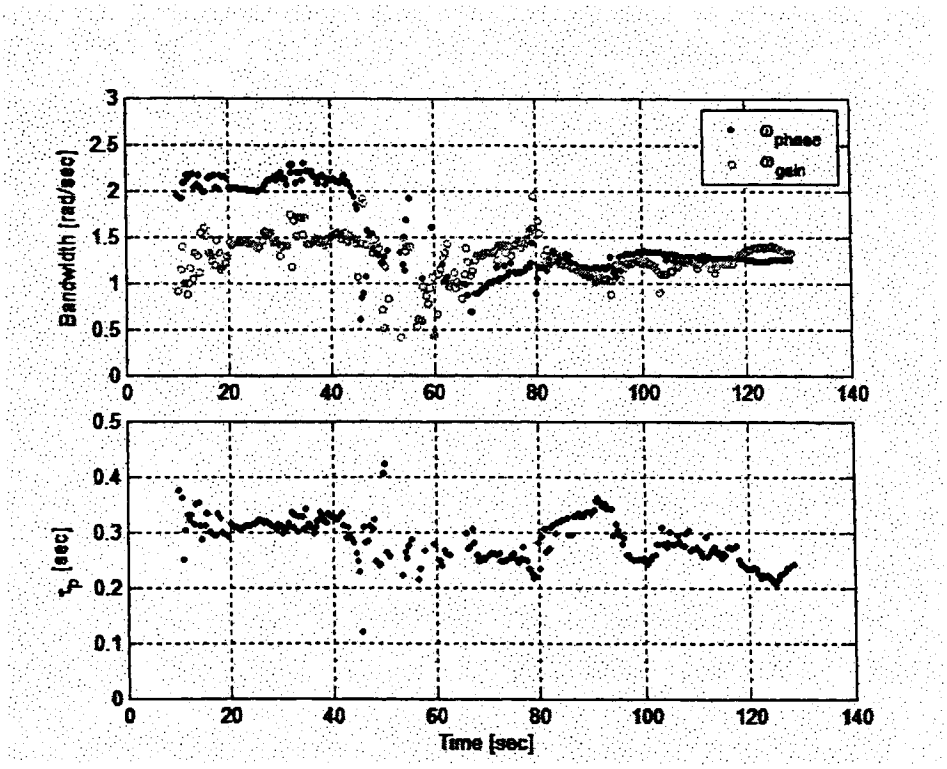


Figure 81. R38 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 82. R38 TVTF Analysis

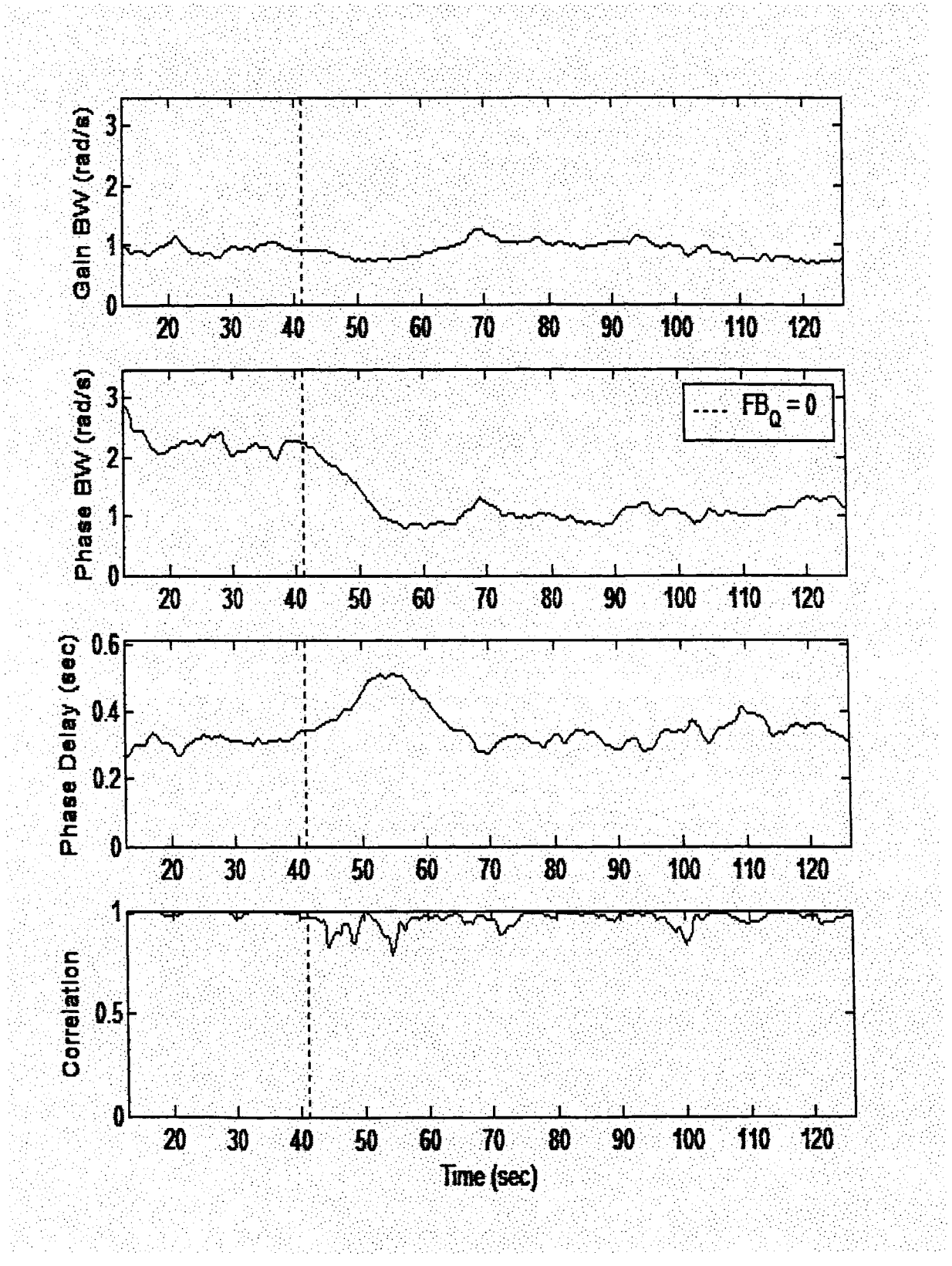


Figure 83. R38 WERA Analysis

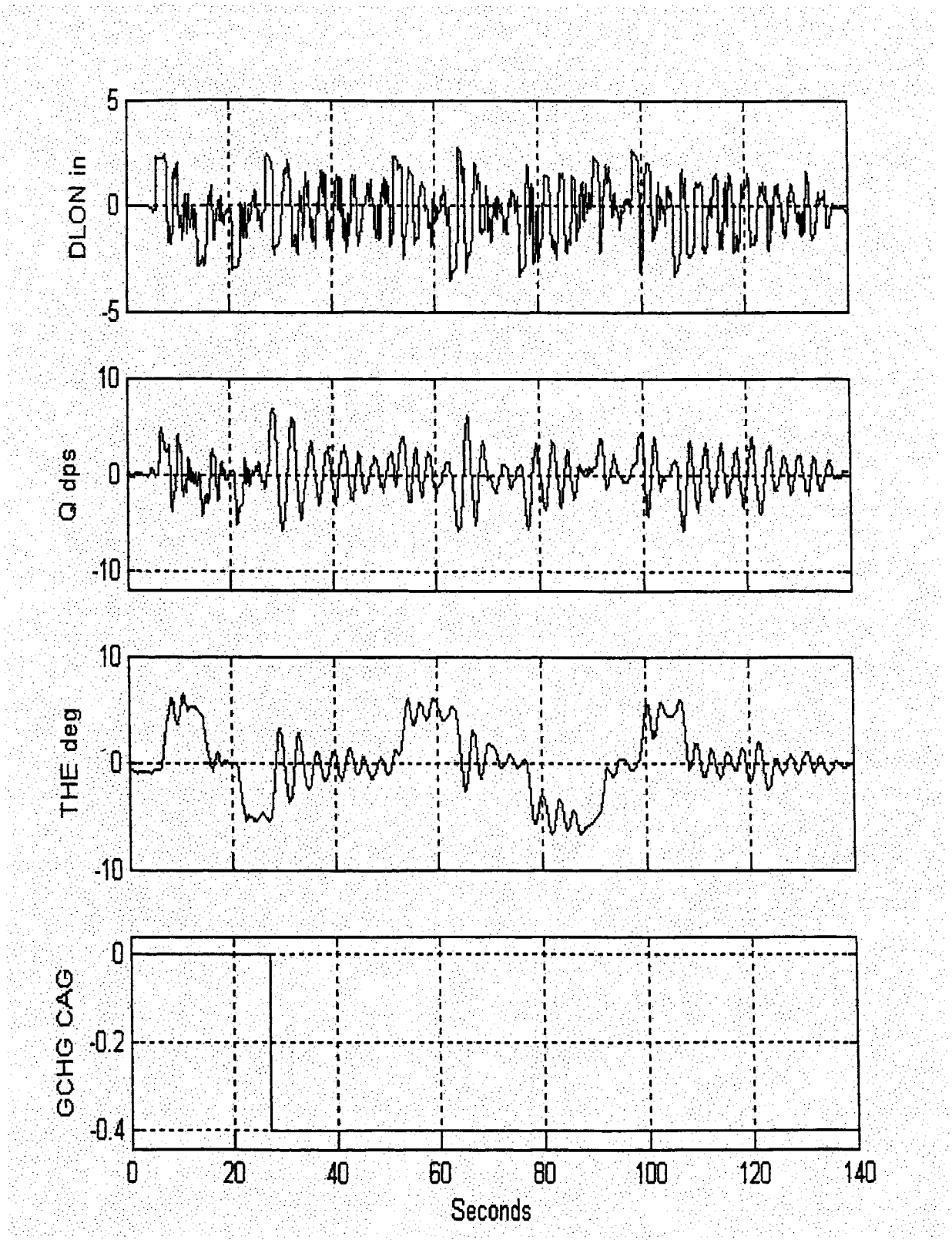
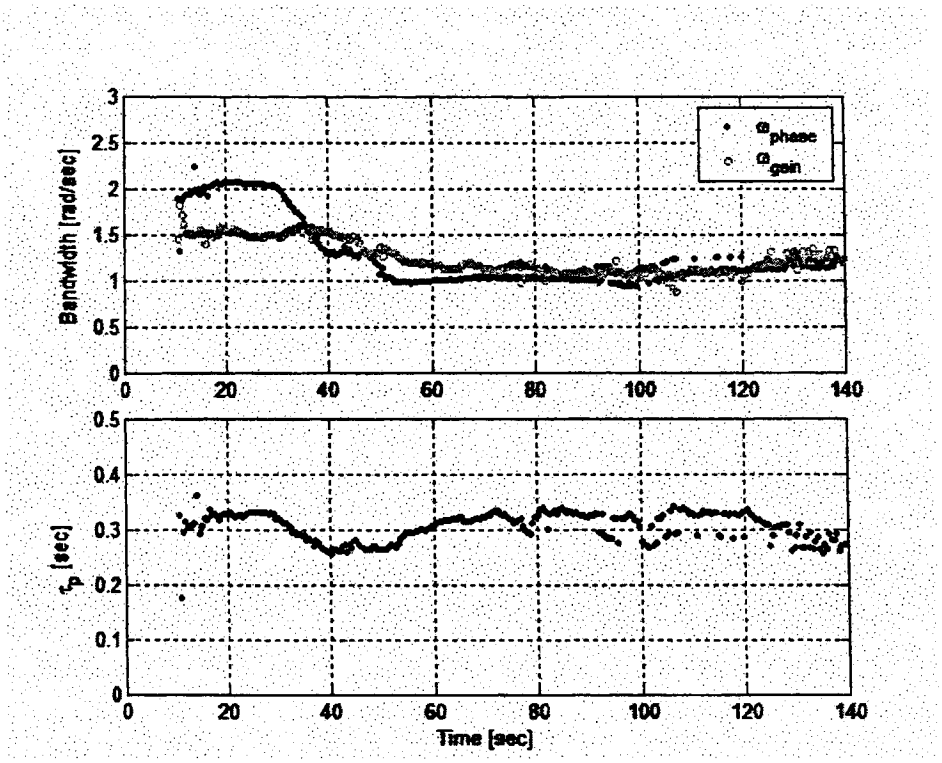
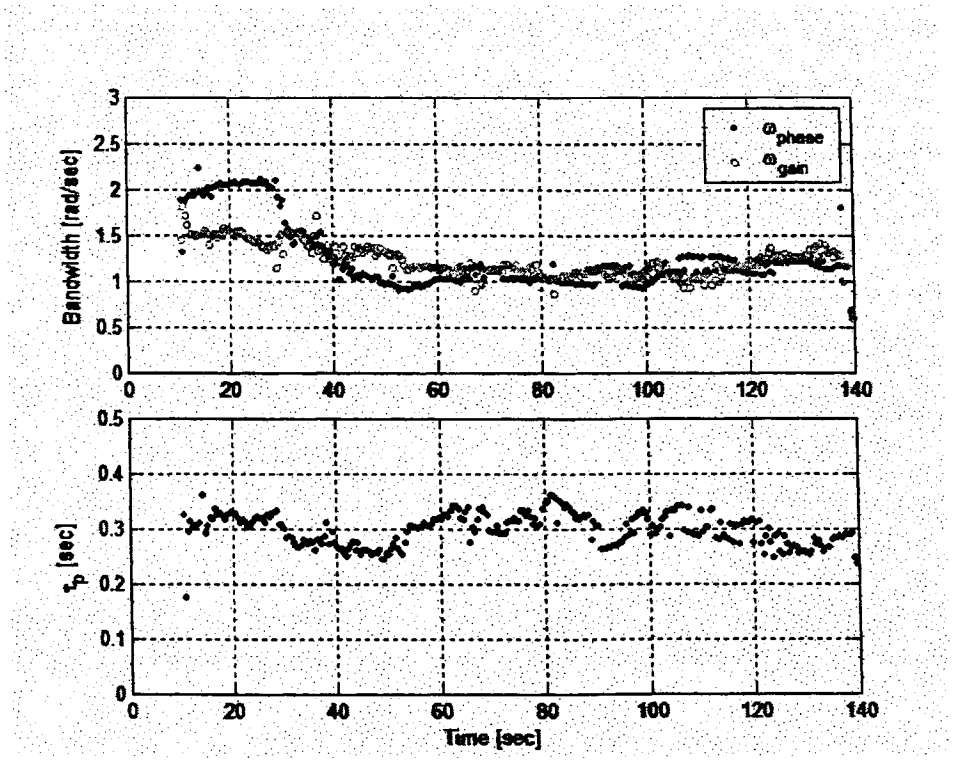


Figure 84. R69 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 85. R69 TVTF Analysis

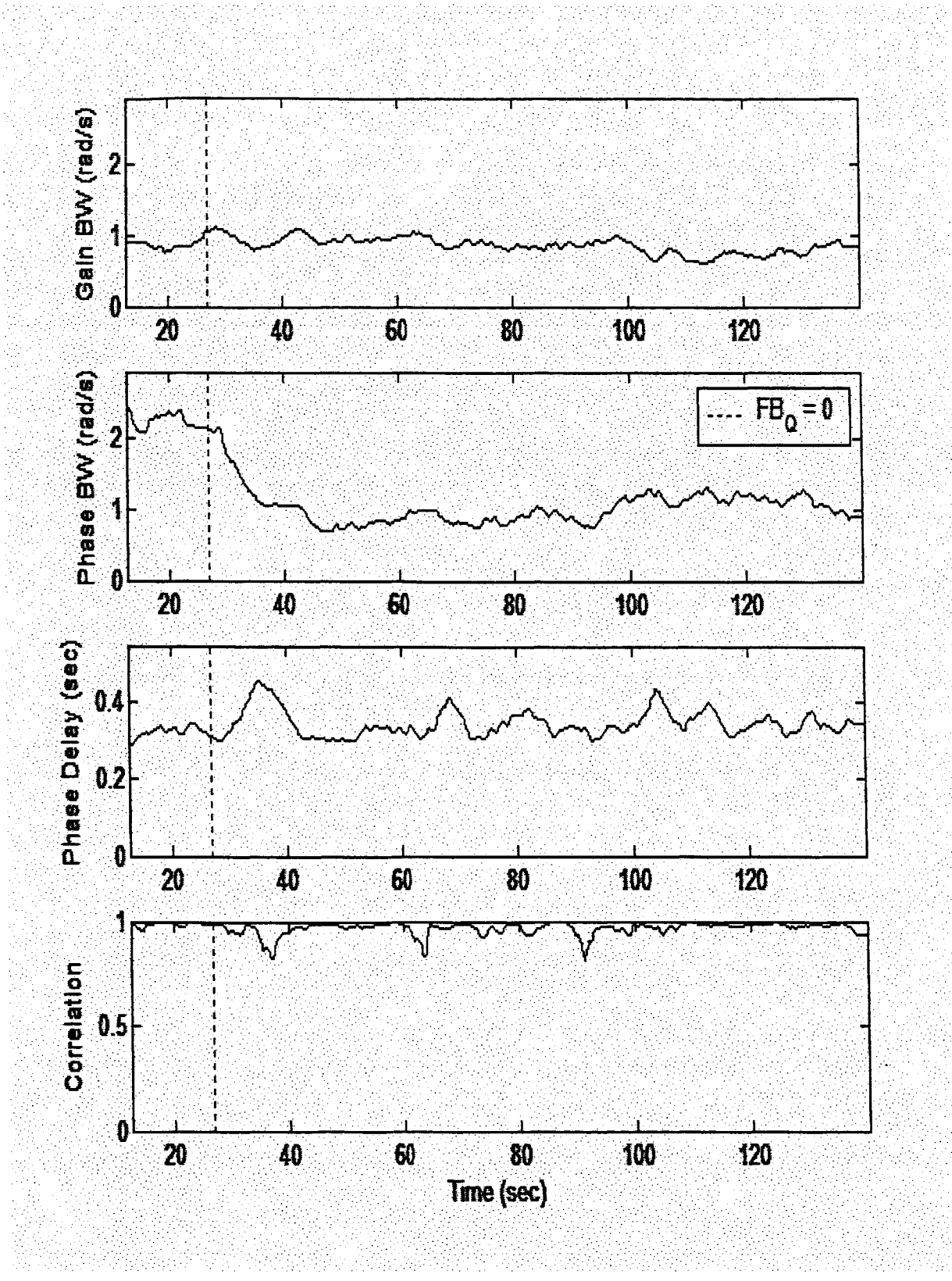


Figure 86. R69 WERA Analysis

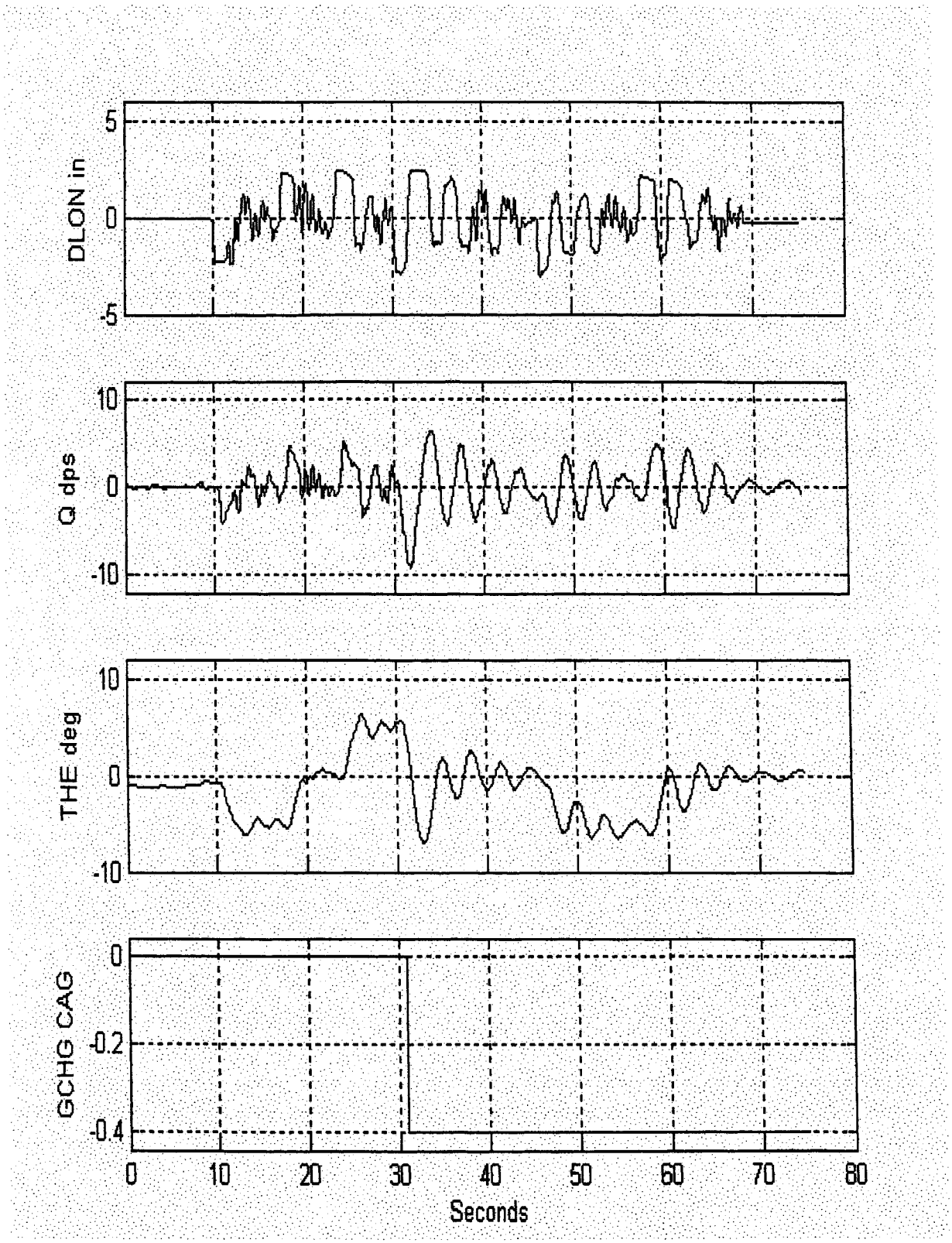
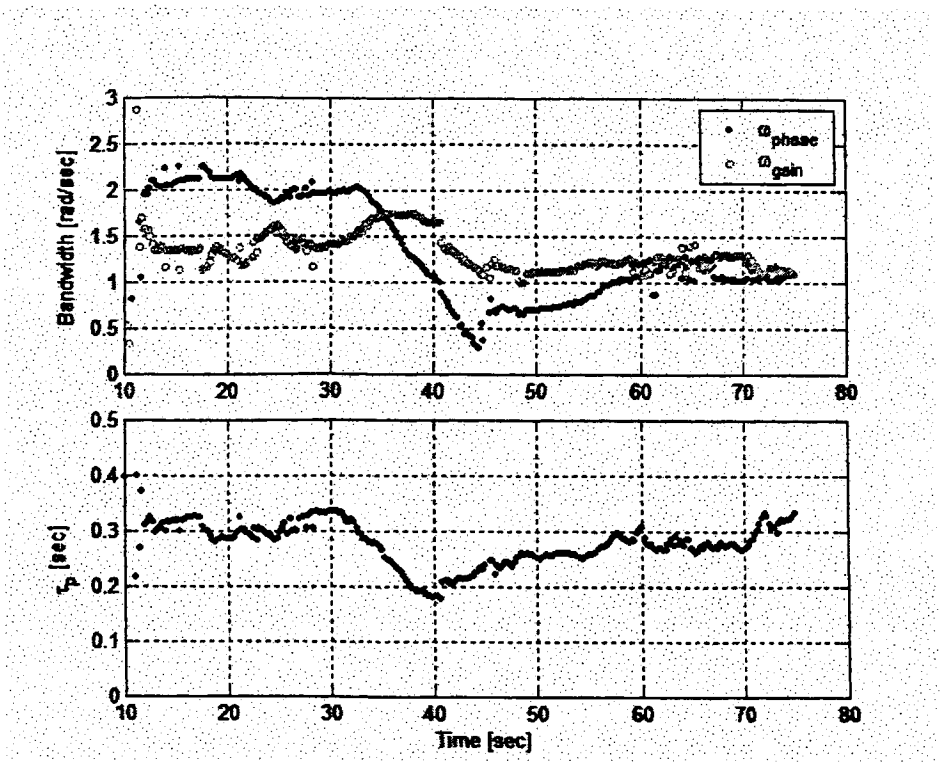
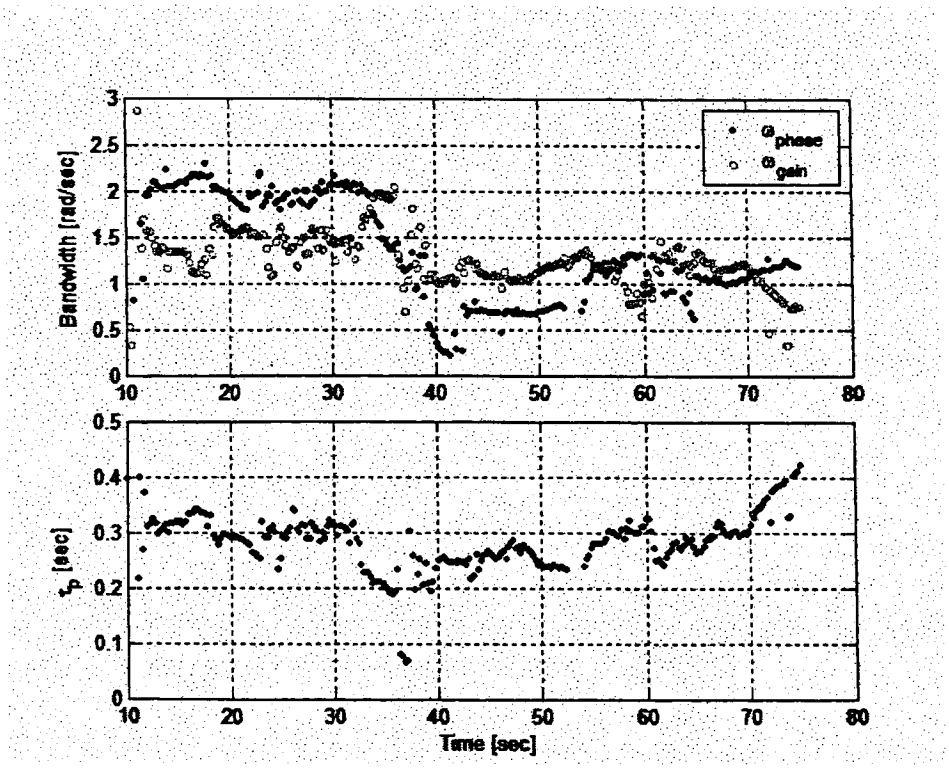


Figure 87. R94 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 88. R94 TVTF Analysis

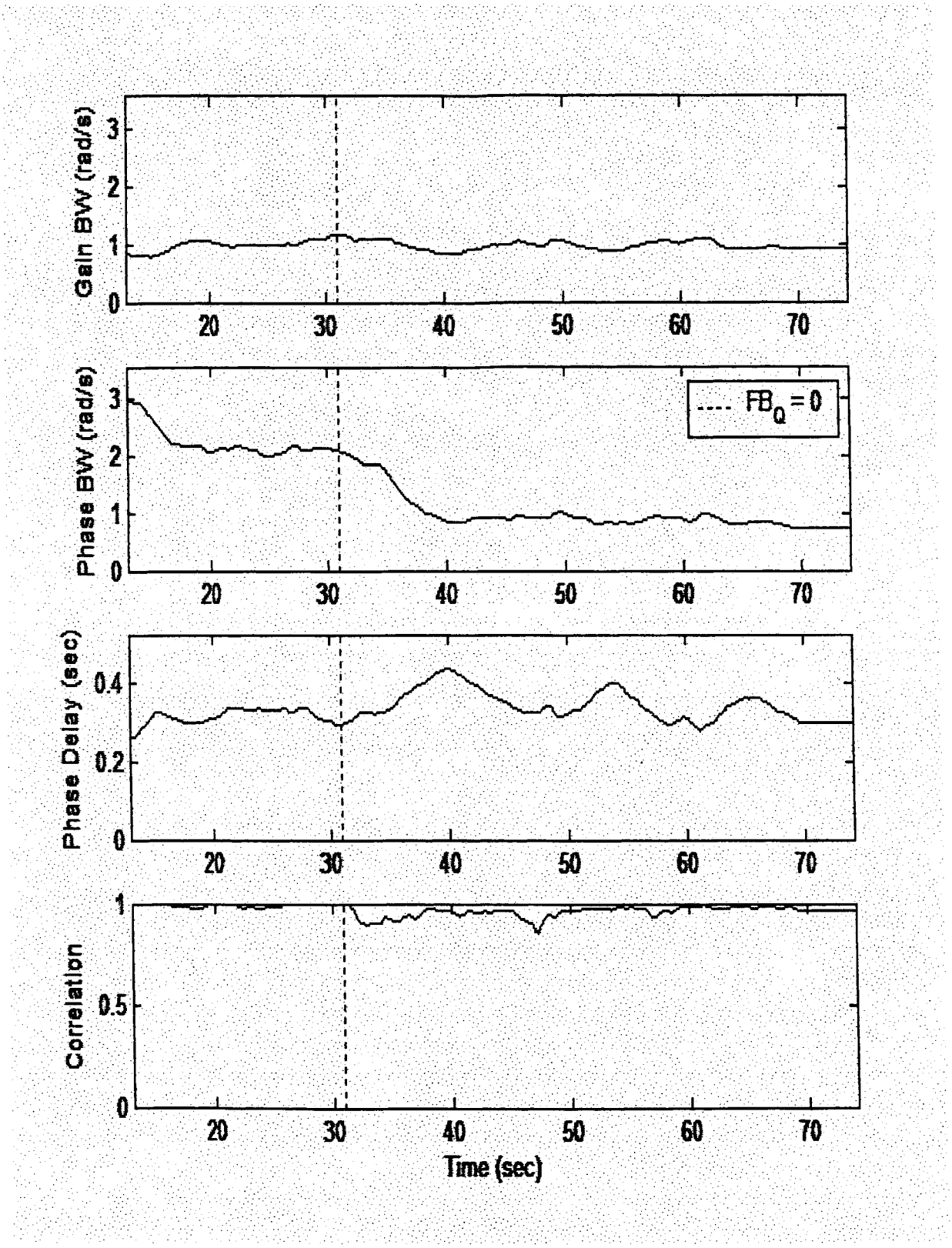


Figure 89. R94 WERA Analysis

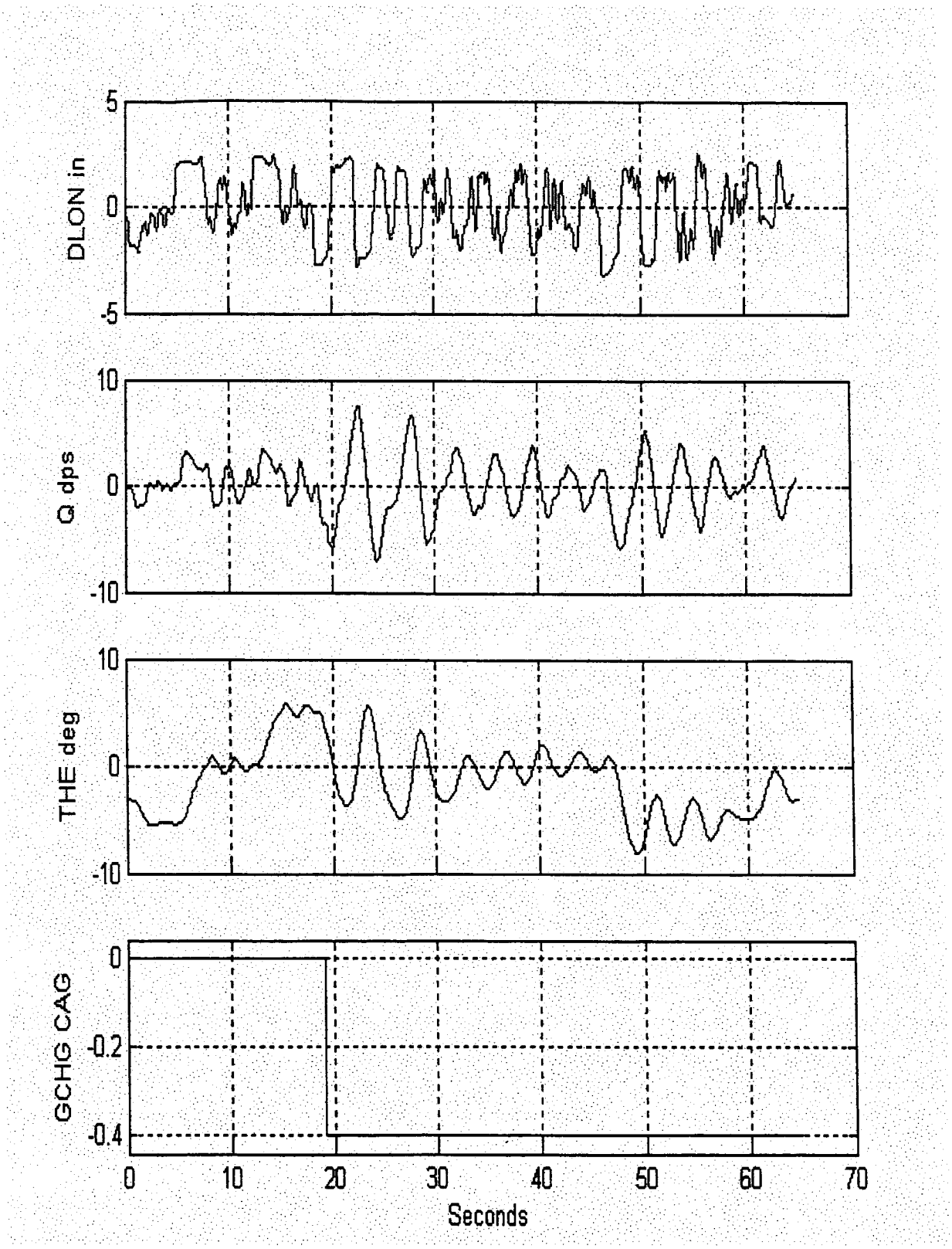
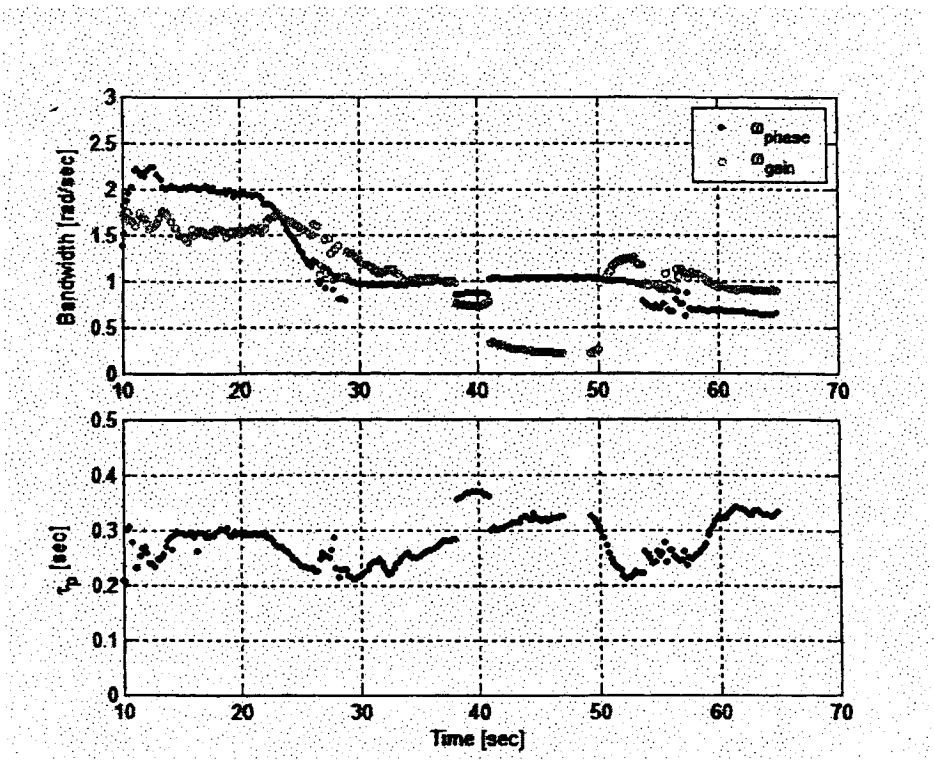
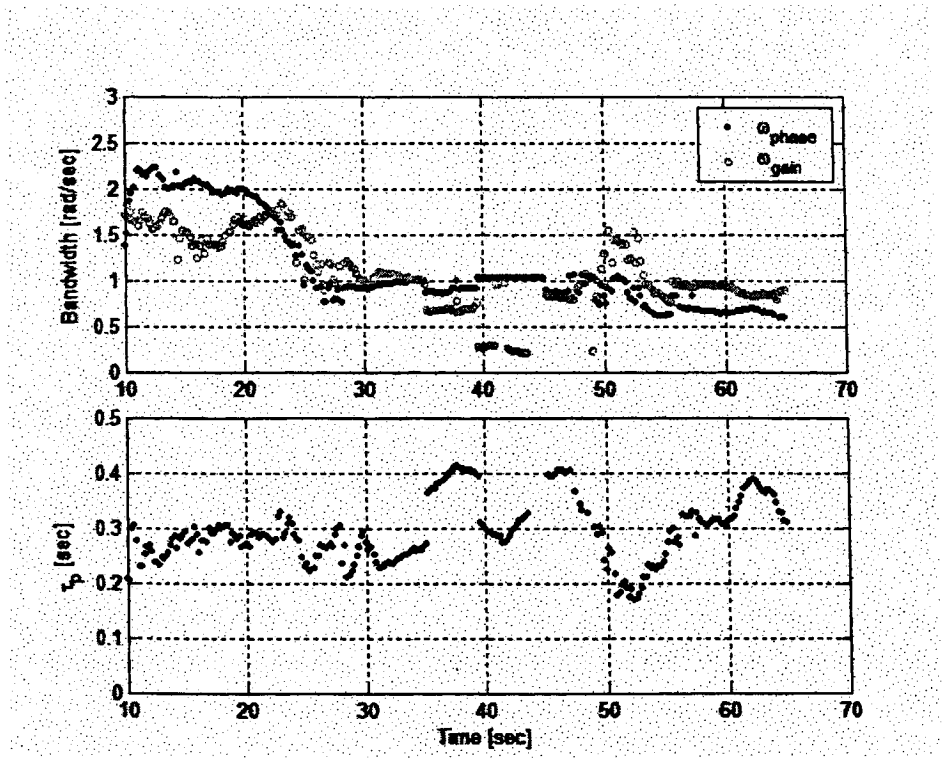


Figure 90. R116 Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 91. R116 TVTF Analysis

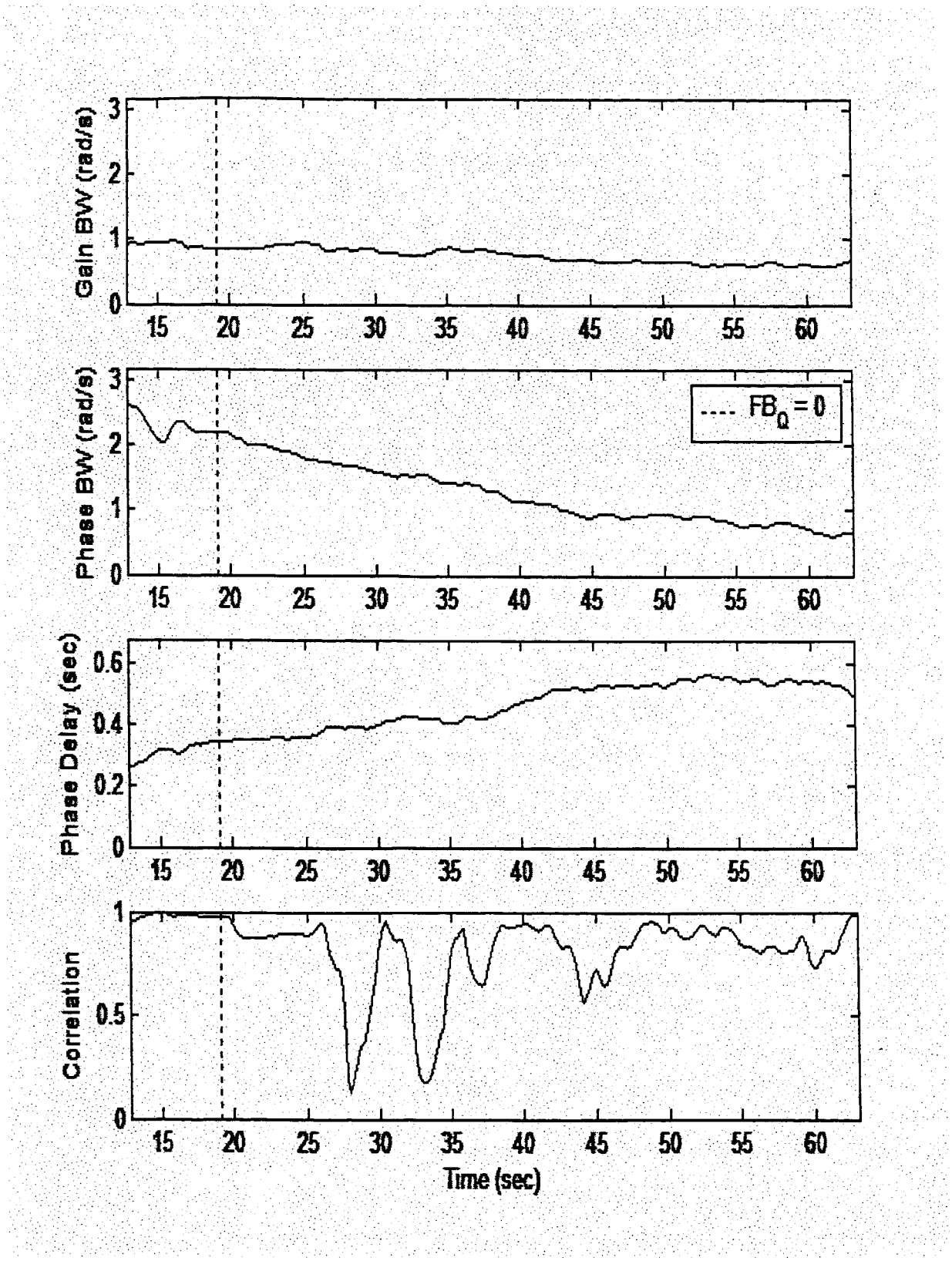


Figure 92. R116 WERA Analysis

4.6 Sum-of-Sines Tracking Evaluation Runs

A listing of the sum-of-sines tracking evaluation runs is provided in Table 17. Time histories and time varying transfer function (TVTF) and wavelet eigen value realization (WERA) analysis plots for the Table 17 runs are provided in Figure 93 through Figure 104.

Table 17. Selected SOS Evaluation Runs

Run#	FC	Configuration	Task	FCS	Comments
No PIO Runs					
R118	CH	CLA	SOS	n/a	Pitch axis only
R123	CH	CLA	SOS	n/a	Roll axis only
Sustained or Severe PIO Runs					
R119	CH	CLA	SOS	QGain	Pitch axis only, PitchTD whole run, sustained PIO follows failure
R124	CH	CLA	SOS	BGain + PGain	Roll axis only, RollTD whole run, sustained oscillations

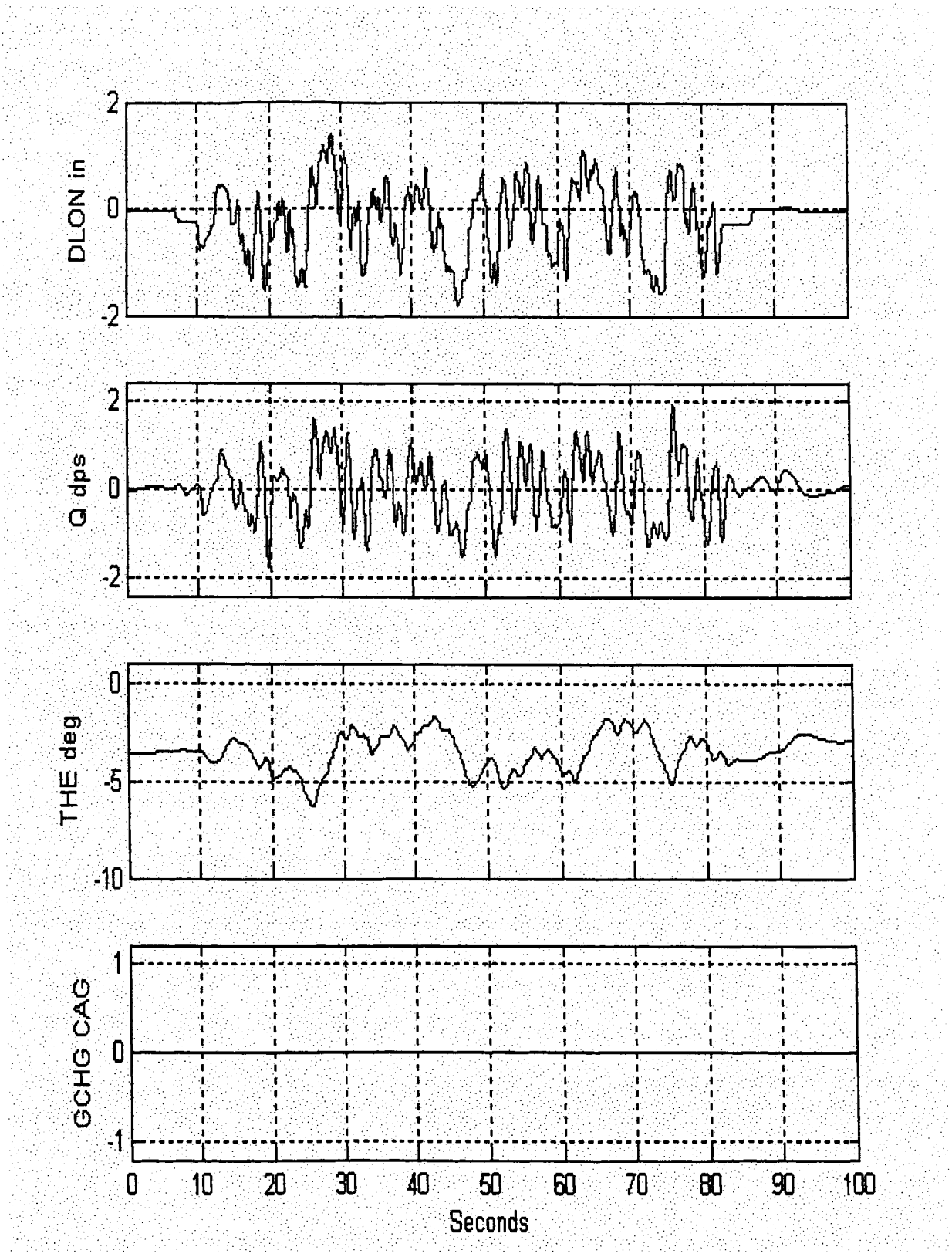
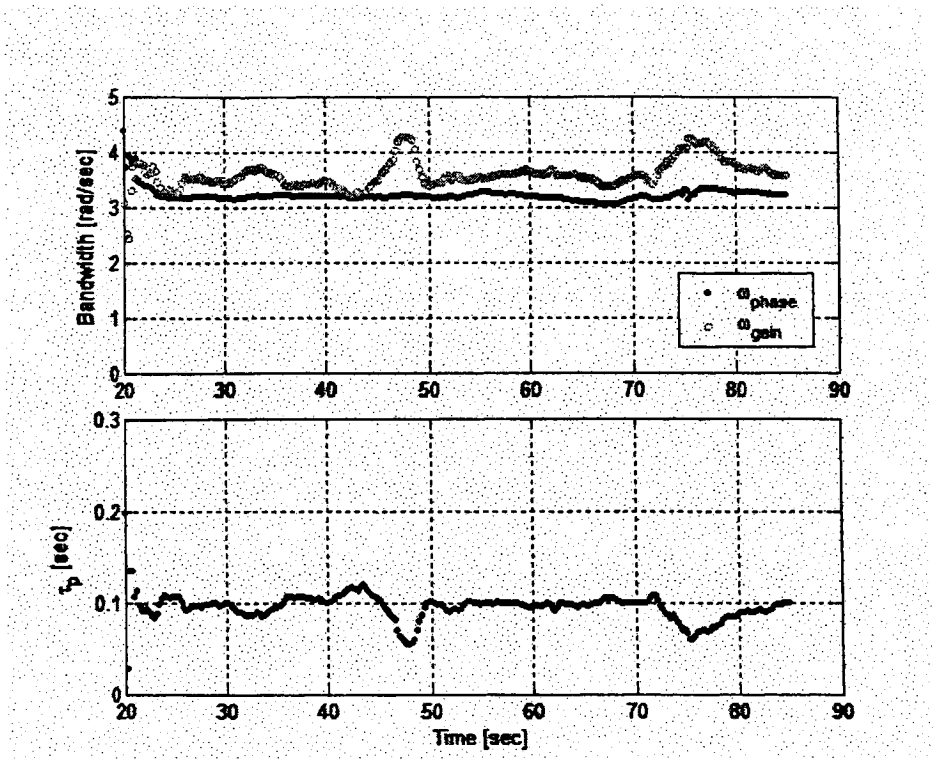
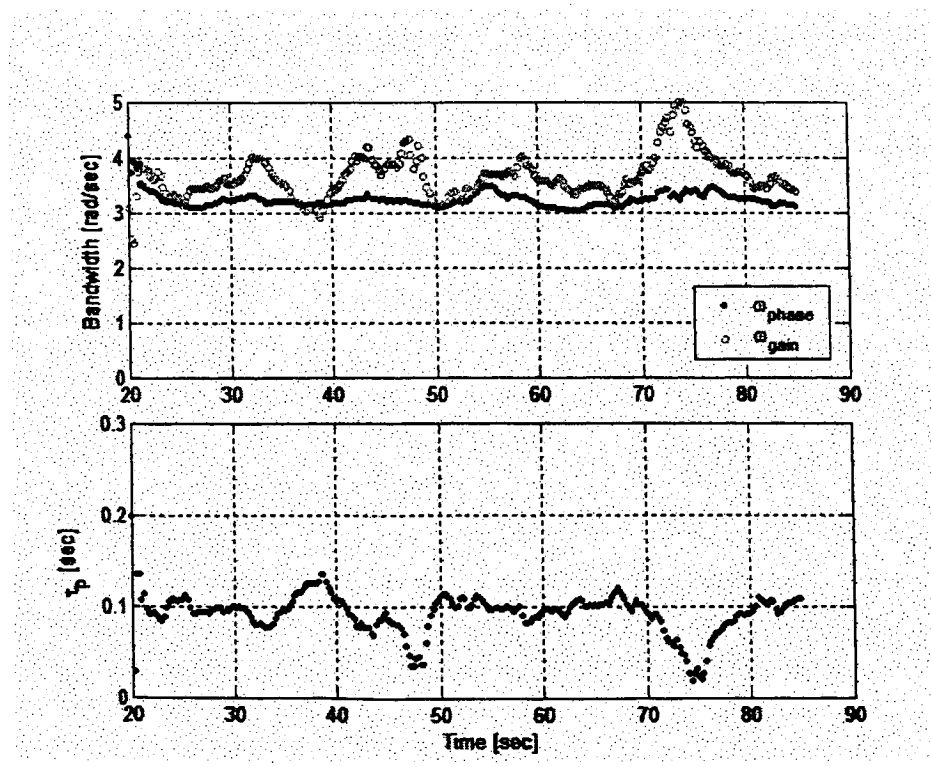


Figure 93. R118 Pitch Axis SOS Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 94. R118 TVTF Analysis

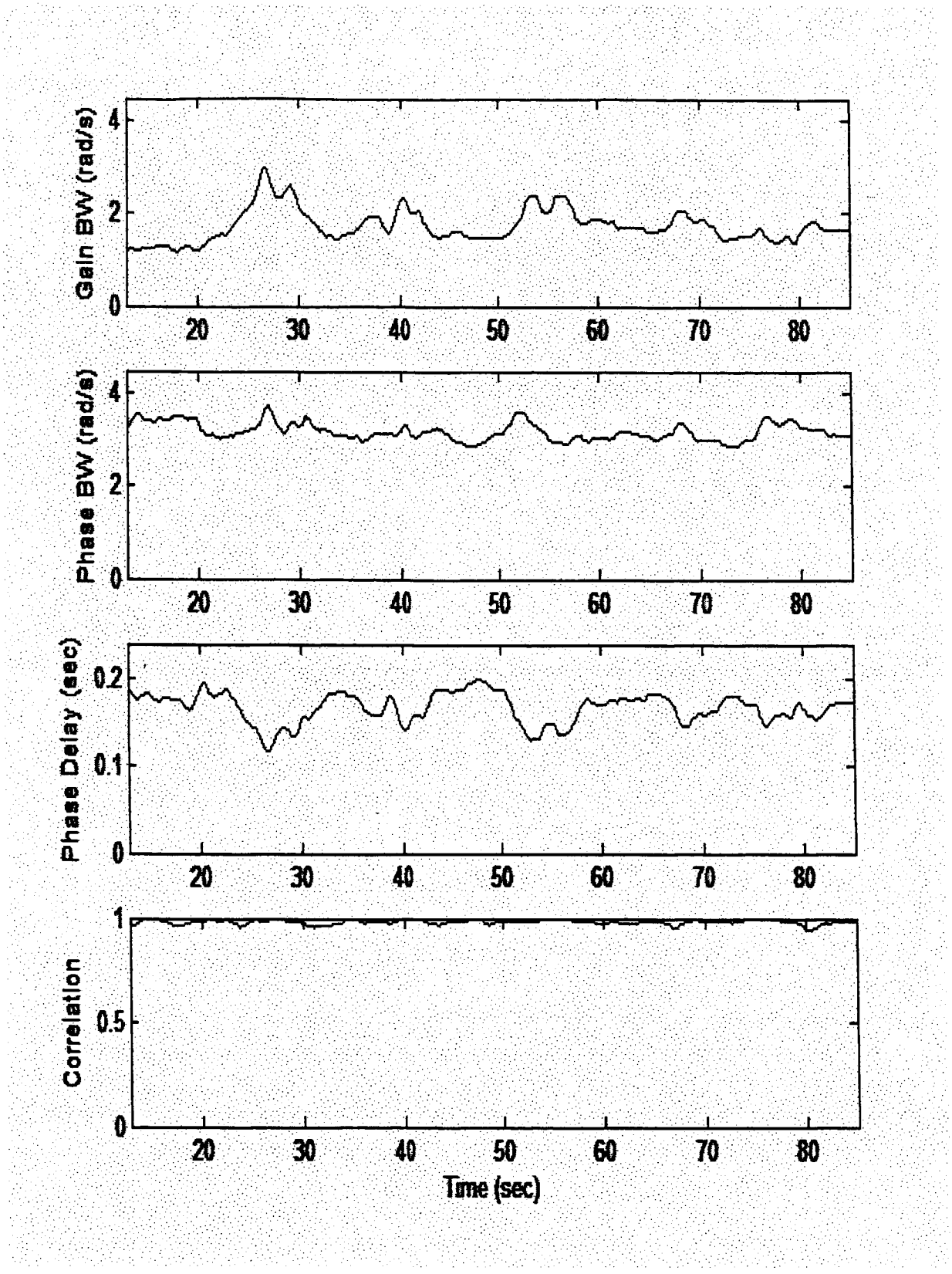


Figure 95. R118 WERA Analysis

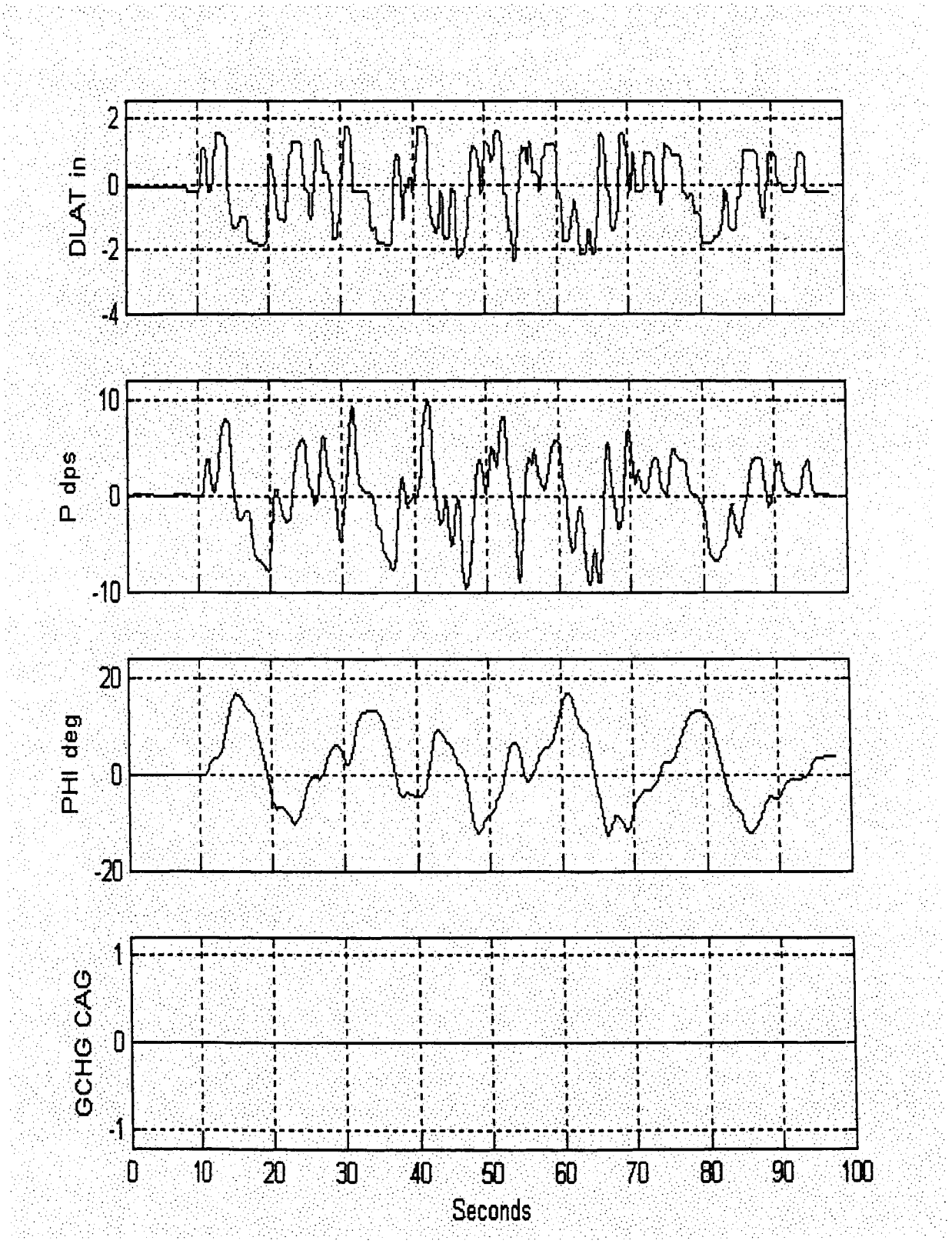
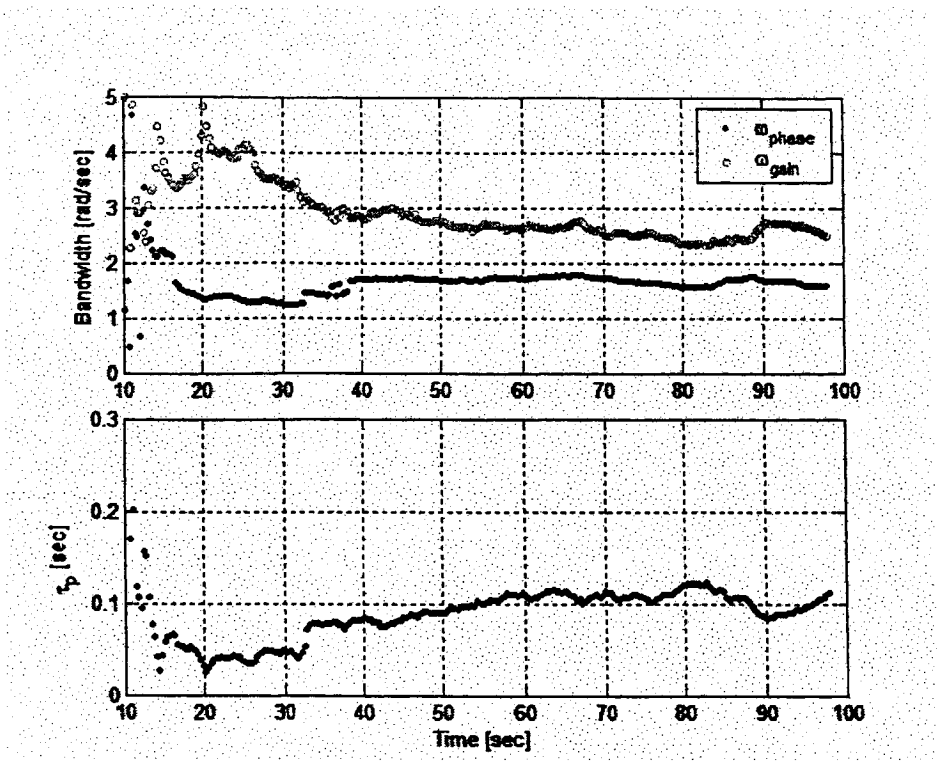
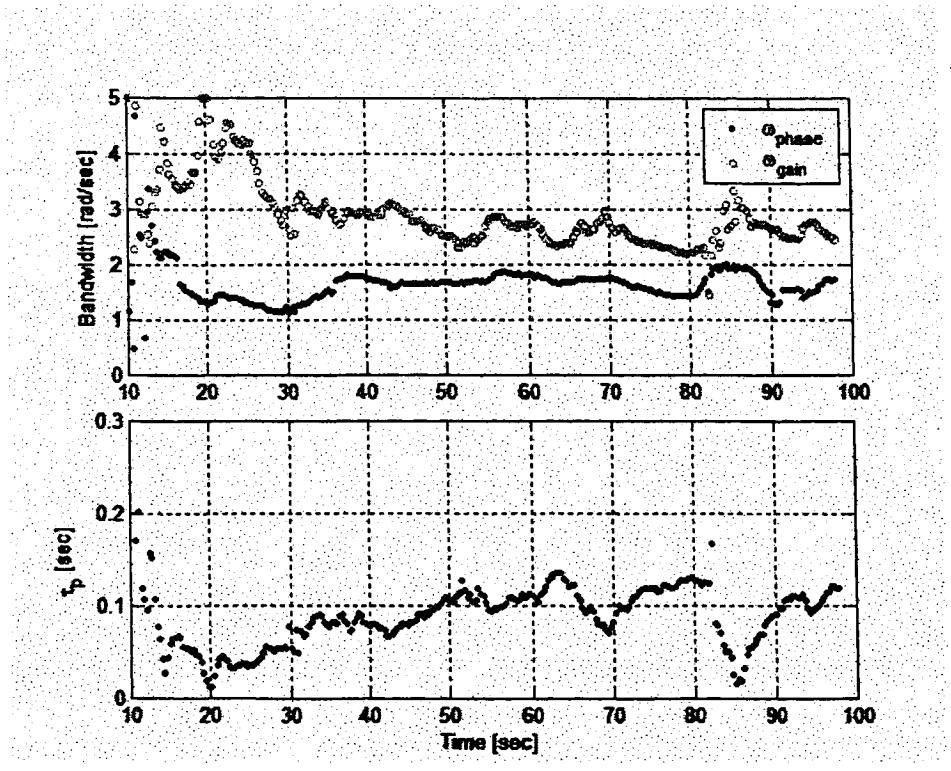


Figure 96. R123 Roll Axis SOS Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 97. R123 TVTF Analysis

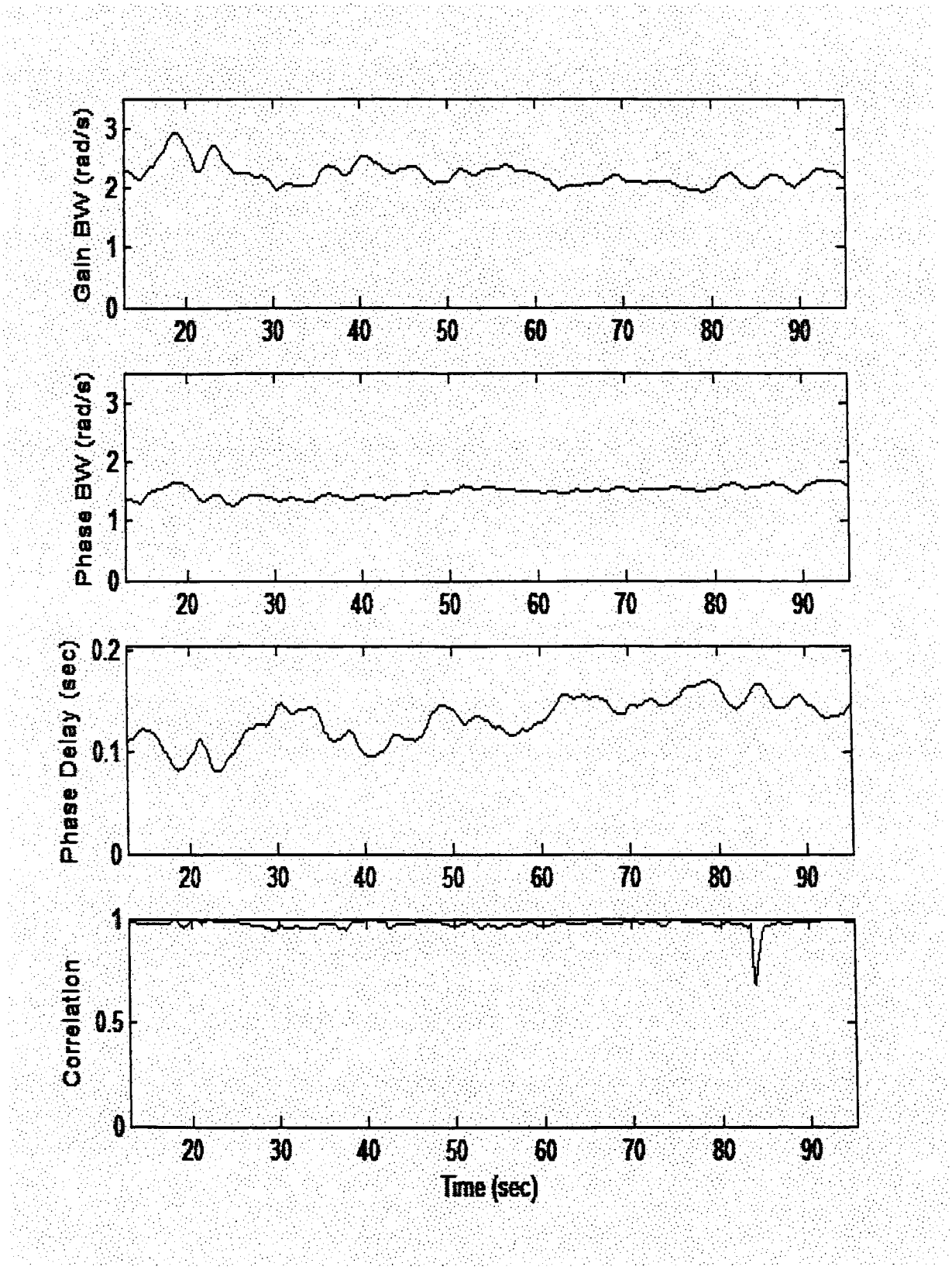


Figure 98. R123 WERA Analysis

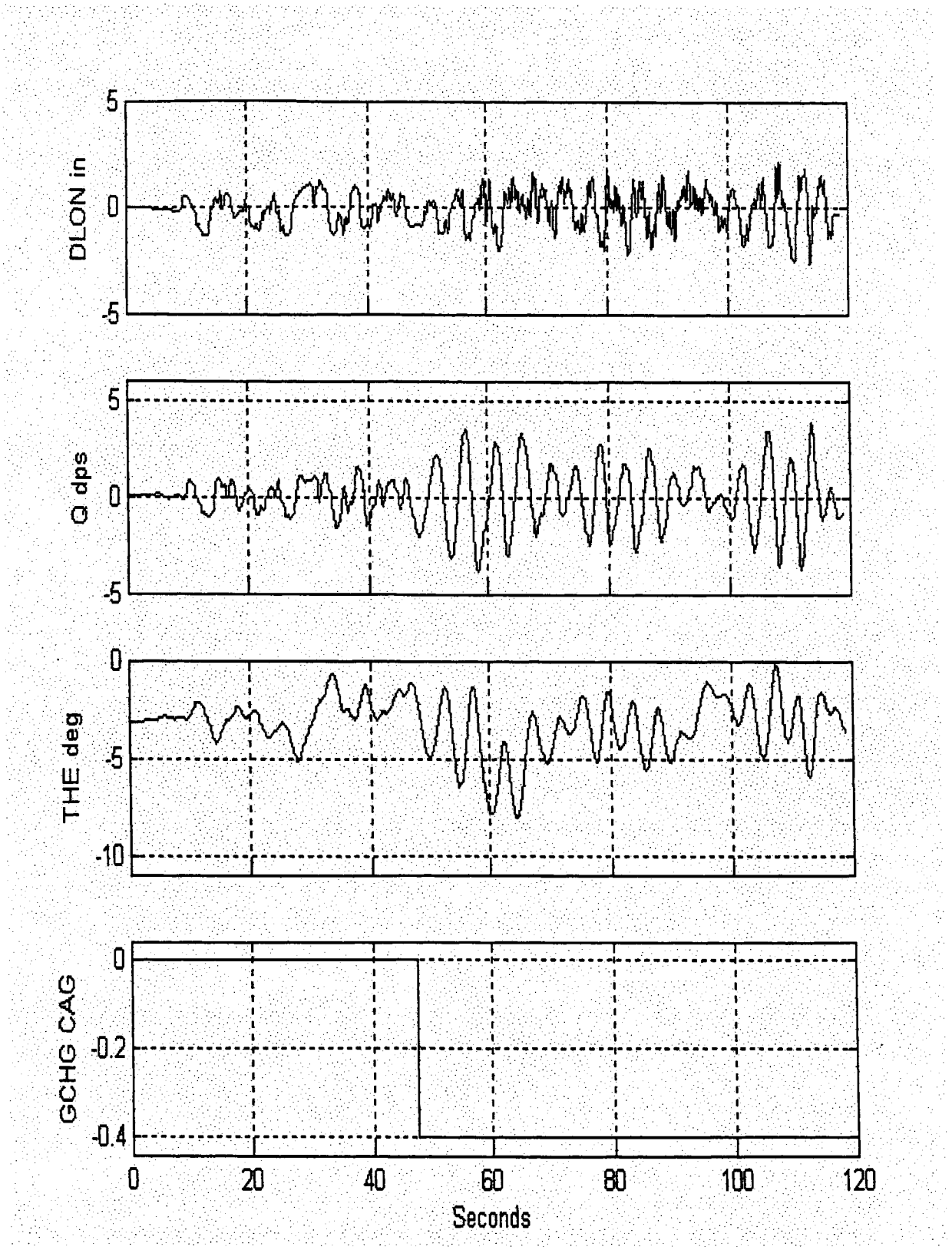
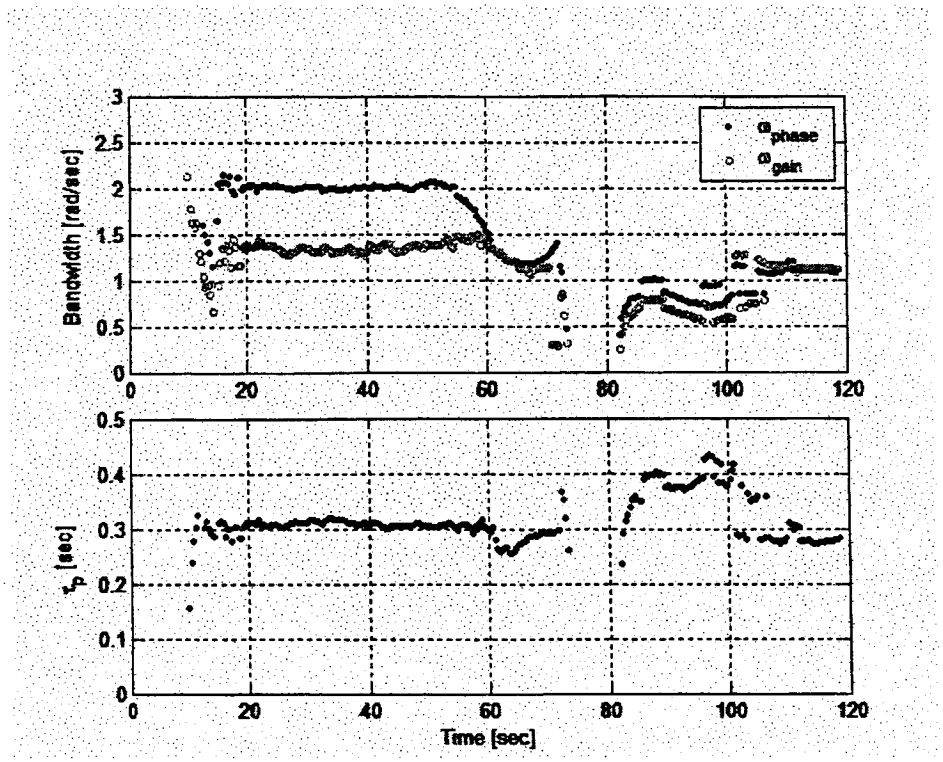
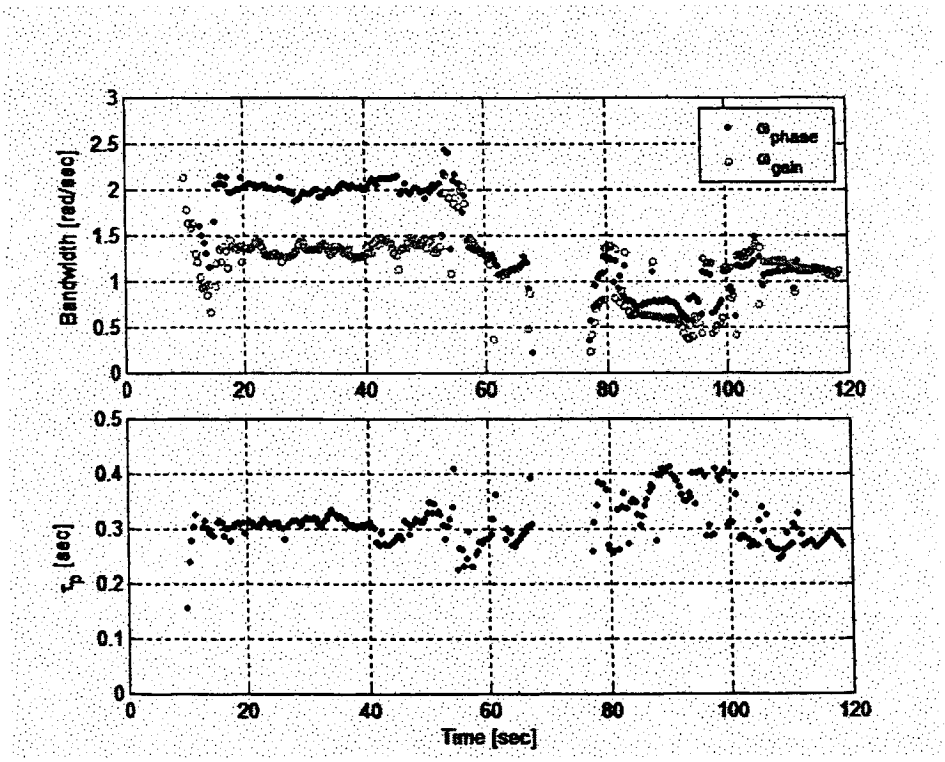


Figure 99. R119 Pitch Axis SOS Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 100. R119 TVTF Analysis

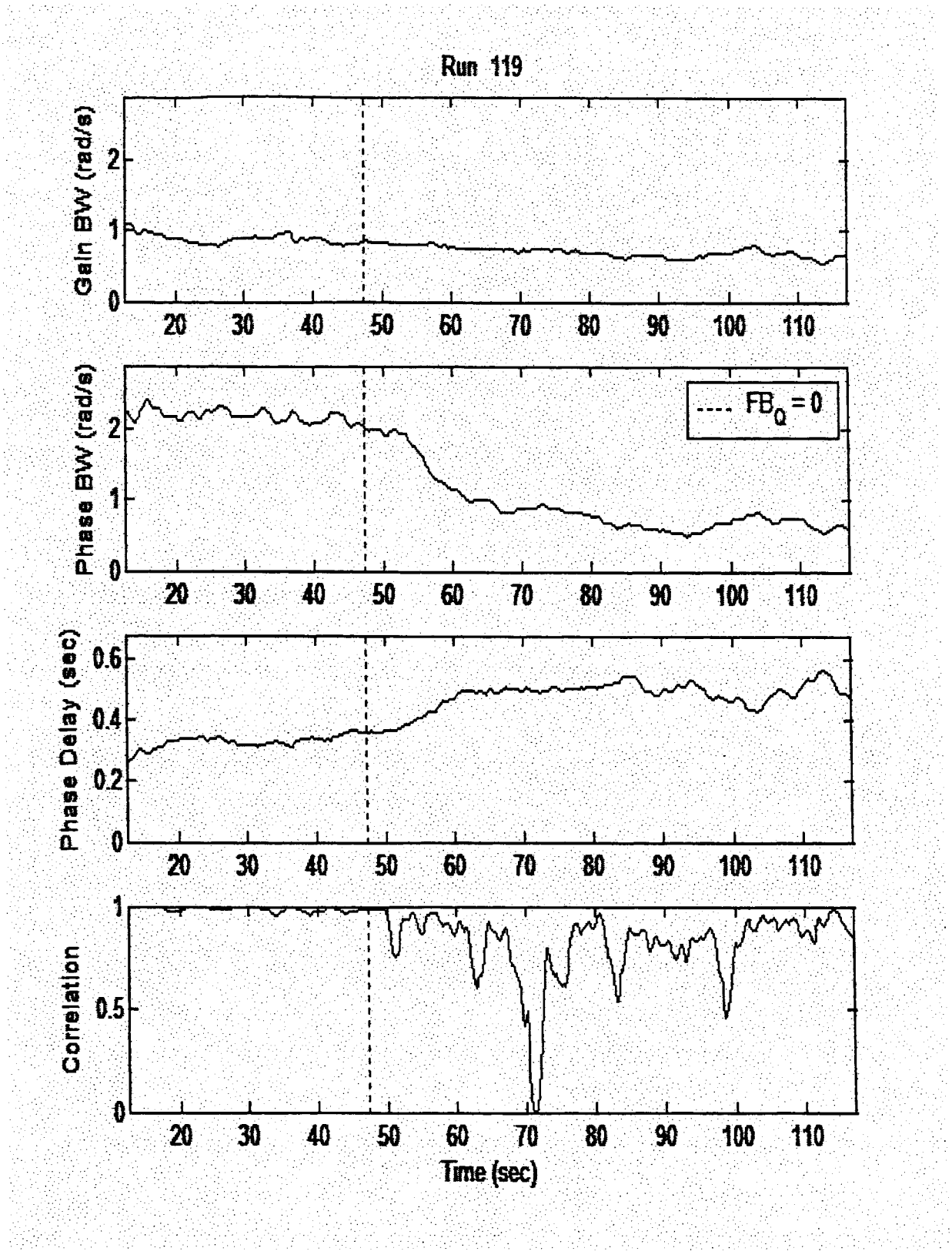


Figure 101. R119 WERA Analysis

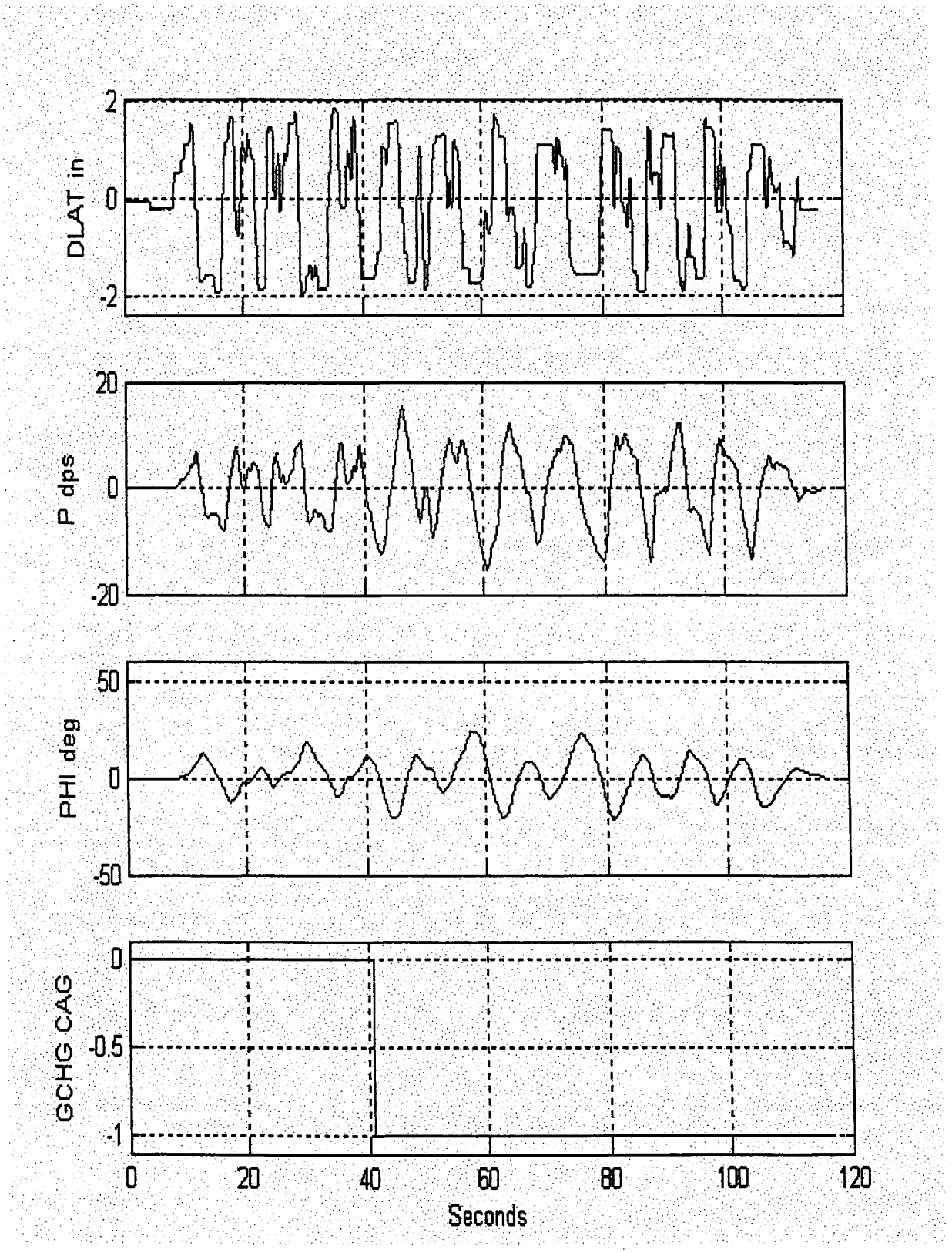
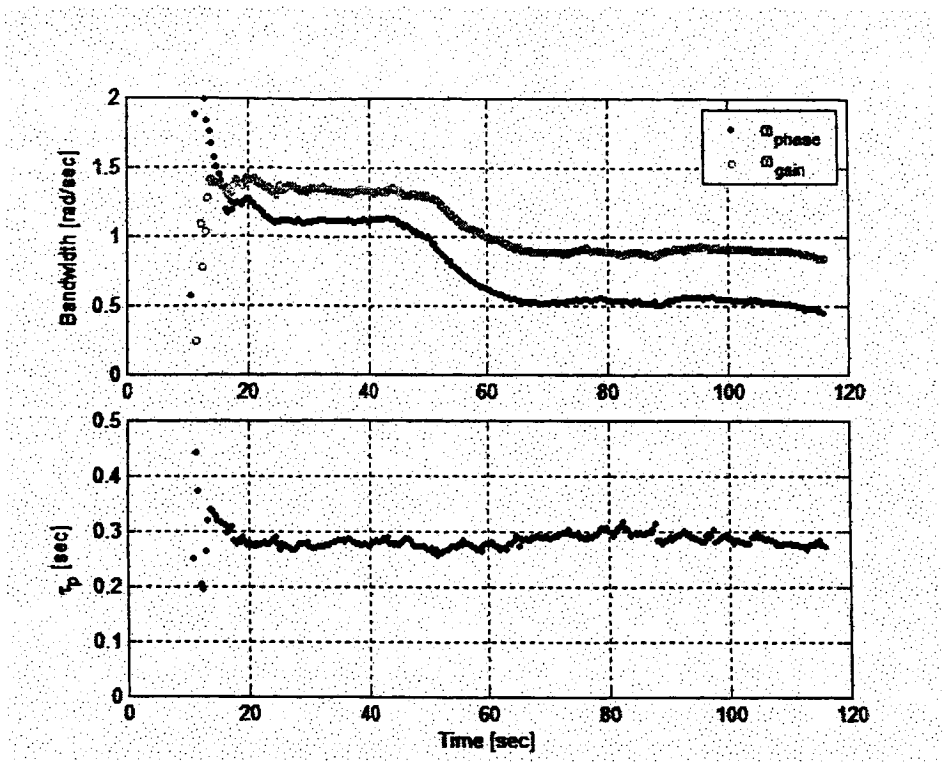
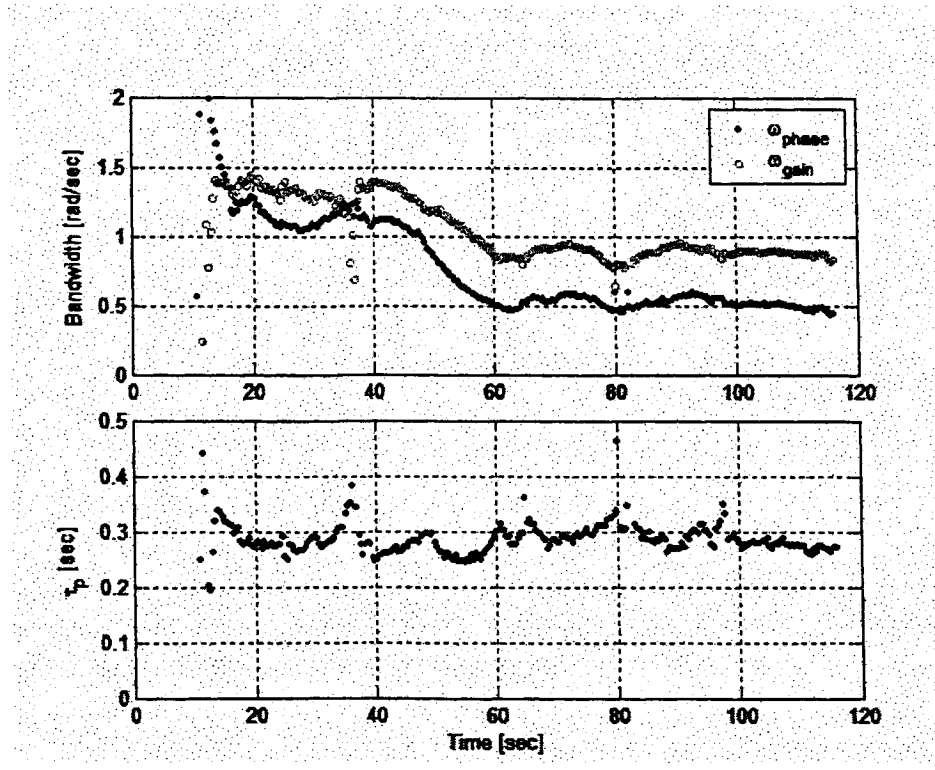


Figure 102. R124 Roll Axis SOS Time Histories, Sustained or Severe PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 103. R124 TVTF Analysis

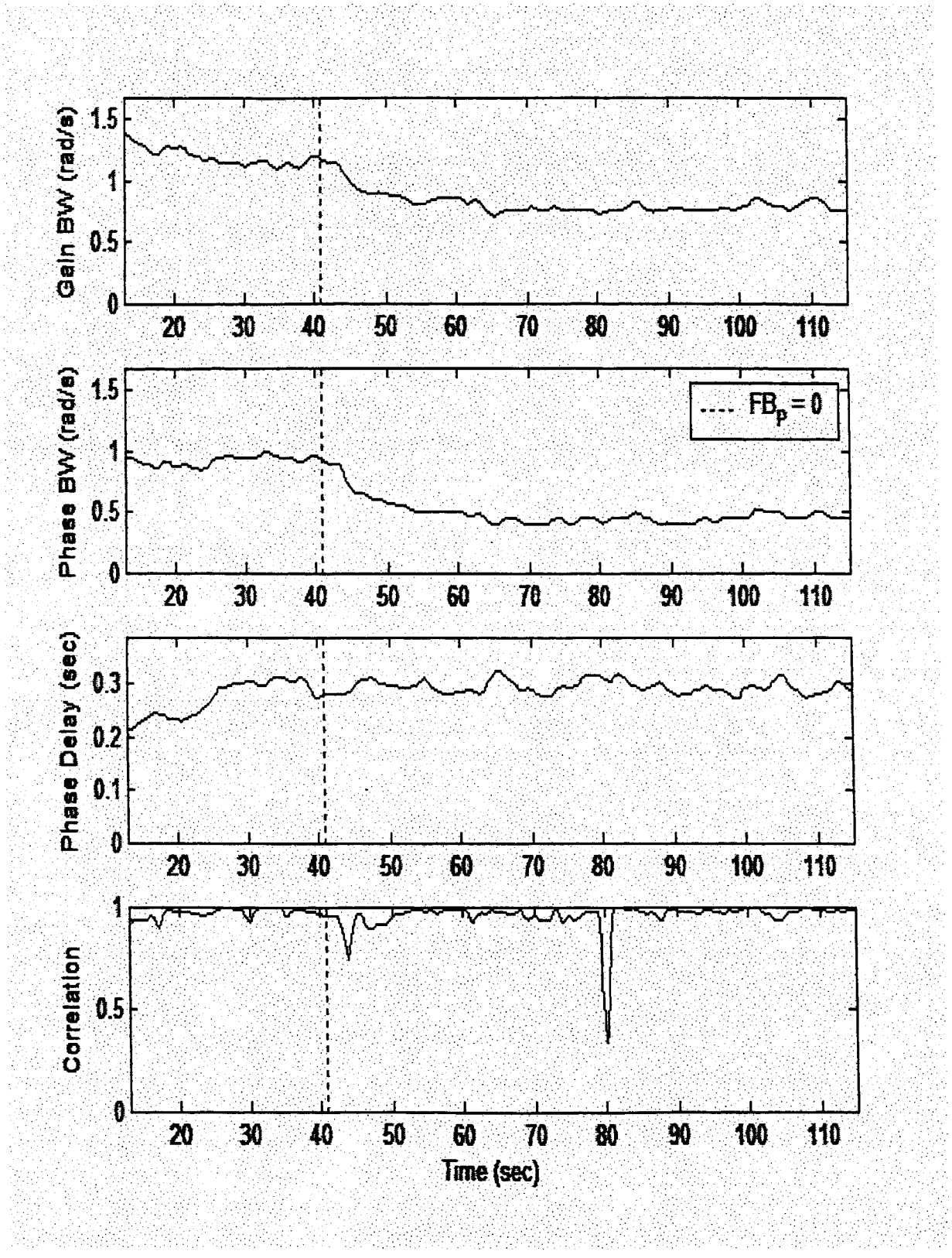


Figure 104. R124 WERA Analysis

4.7 Discrete Tracking Task Evaluations

A listing of the discrete tracking evaluation runs is provided in Table 18. Time histories and time varying transfer function (TVTF) and wavelet eigen value realization (WERA) analysis plots for the Table 18 runs are provided in Figure 105 through Figure 128.

Table 18. Selected DT Evaluation Runs

Run#	FC	Configuration	Task	LOC	Comments
No PIO Runs					
R15	C	CHF	DT	n/a	
R18	C	CHF	DT	PLIM	
Mild PIO Runs					
R70	CM	CLA	DT	QGain + BGain + PGain	Mild pitch PIO
R71	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, pitch PIO
R99	CM	CLA	DT	QGain	Mild pitch PIO
R100	CM	CLA	DT	QGain	PitchTD whole run, large overshoot following failure, mild PIO
R106	CM	CLA	DT	QGain + BGain + PGain	PitchTD and RollTD whole run, multi-axis failures
Sustained or Severe PIO Runs					
R128	CH	CLA	DT	BGain + PGain	PitchTD and RollTD whole run, sustained roll PIO

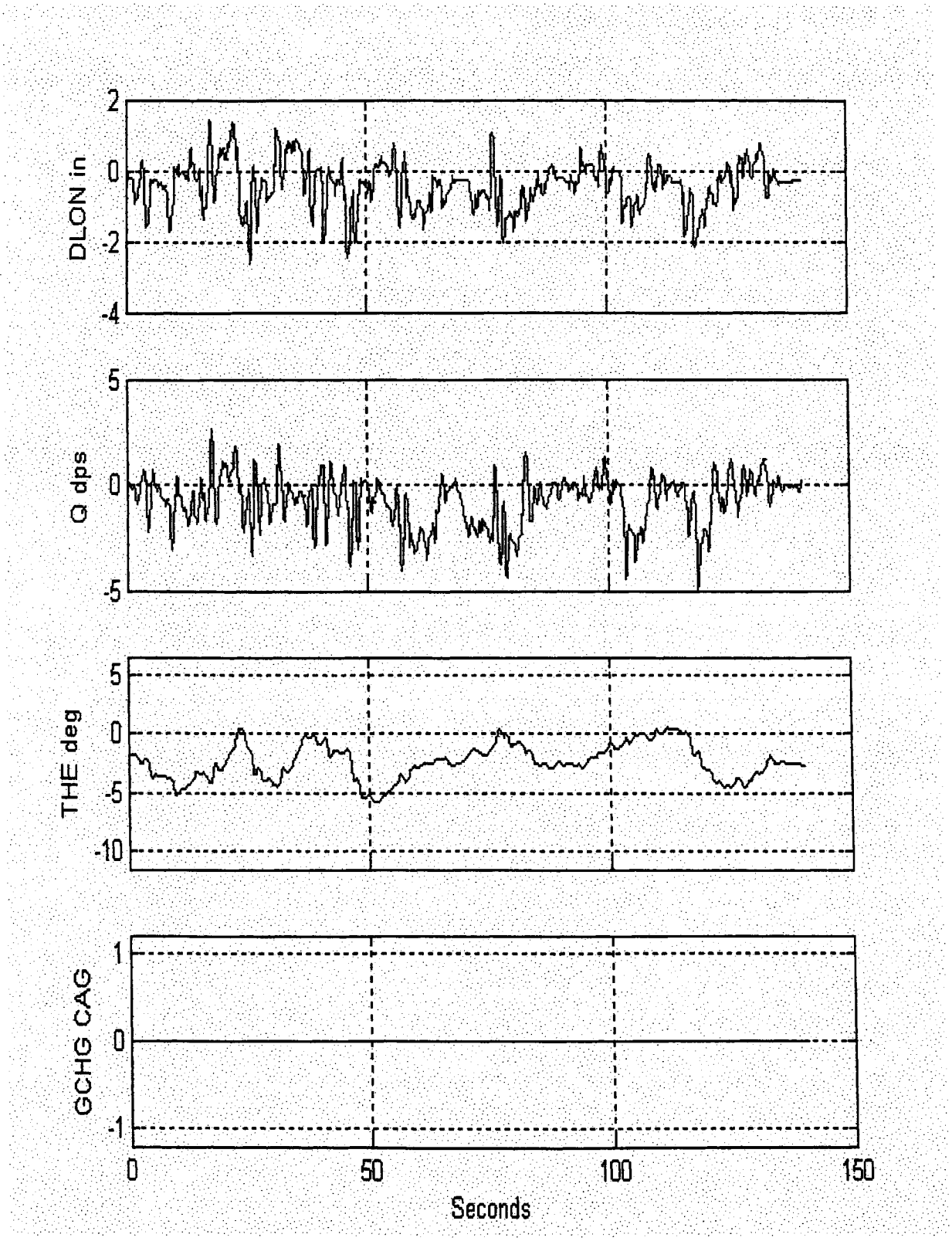
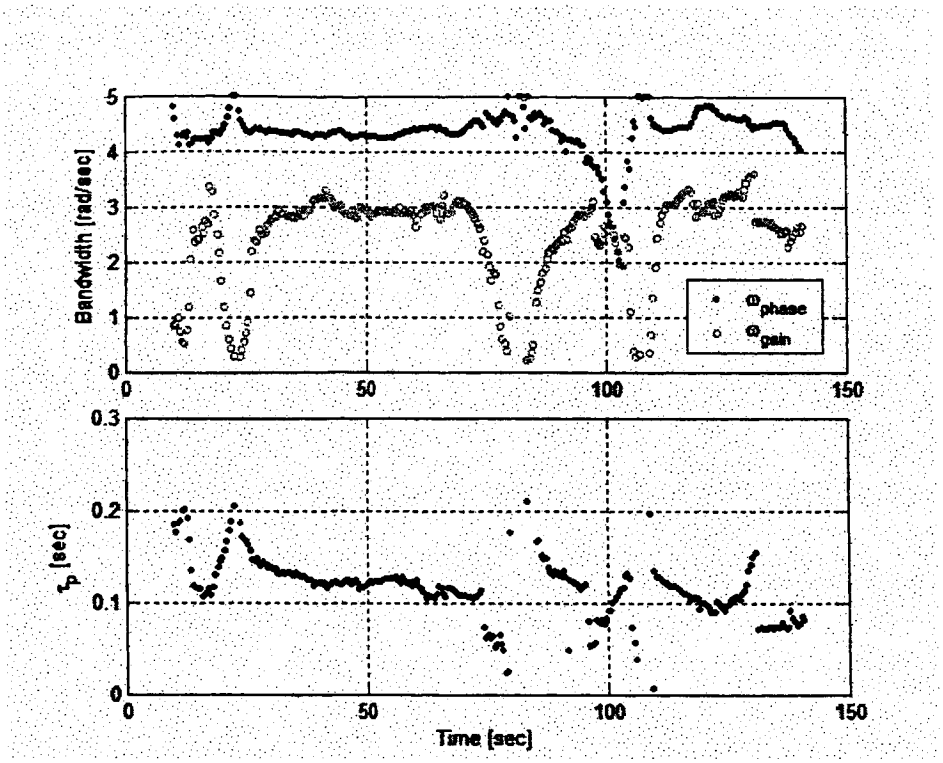
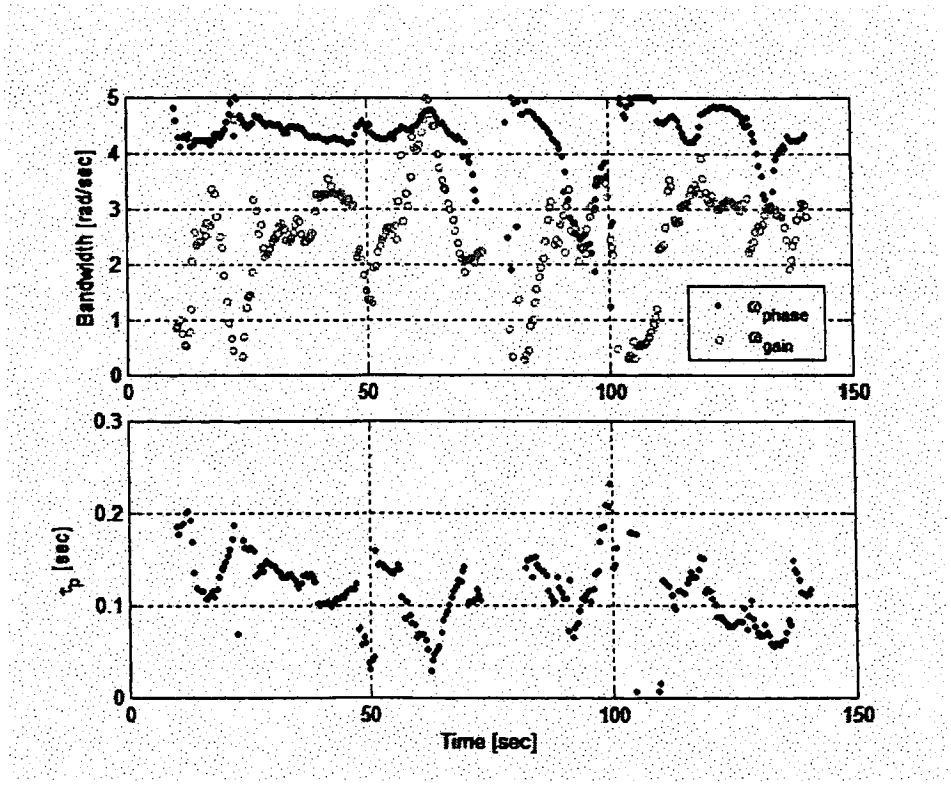


Figure 105. R15 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 106. R15 TVTF Analysis

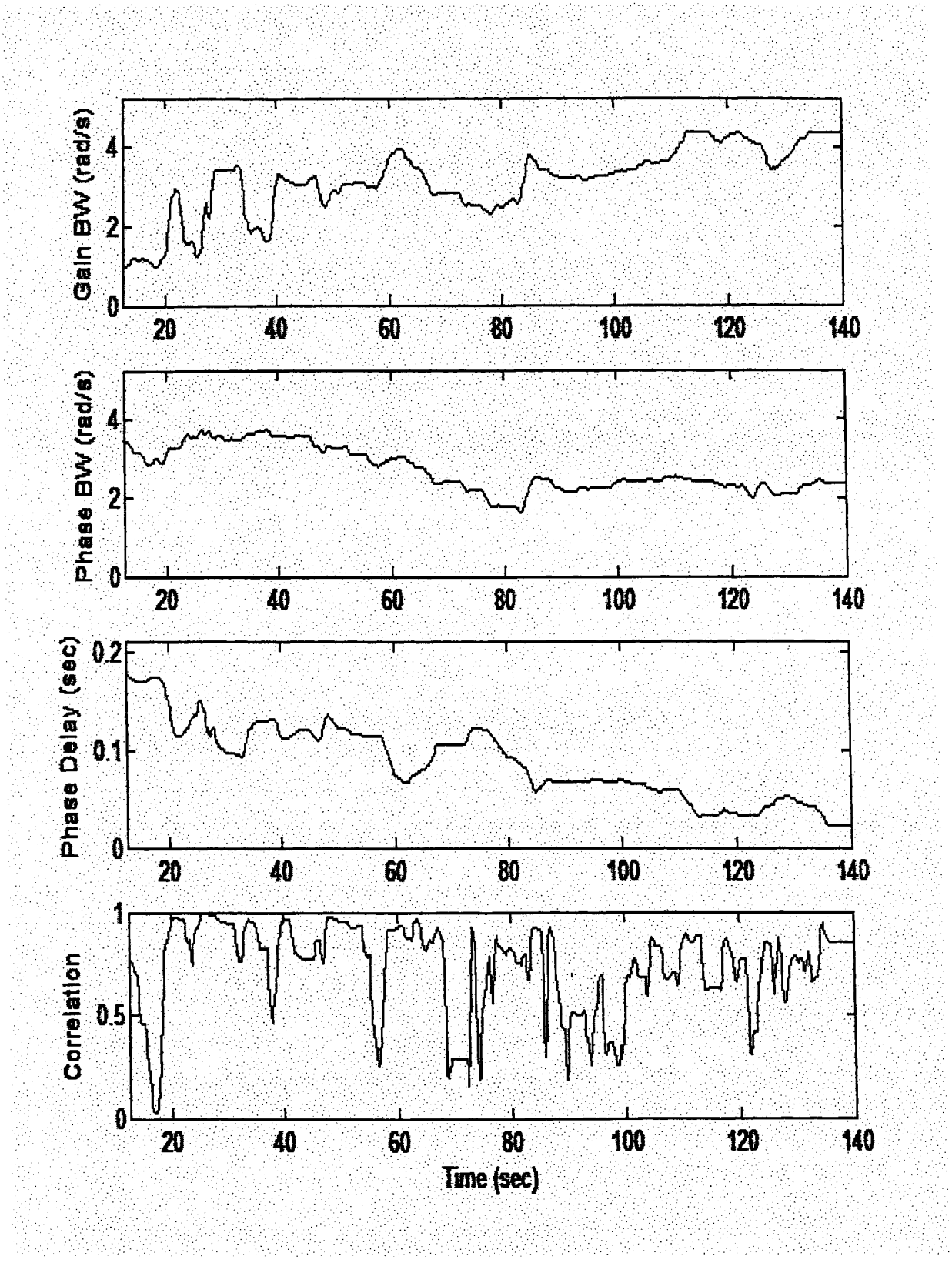


Figure 107. R15 WERA Analysis

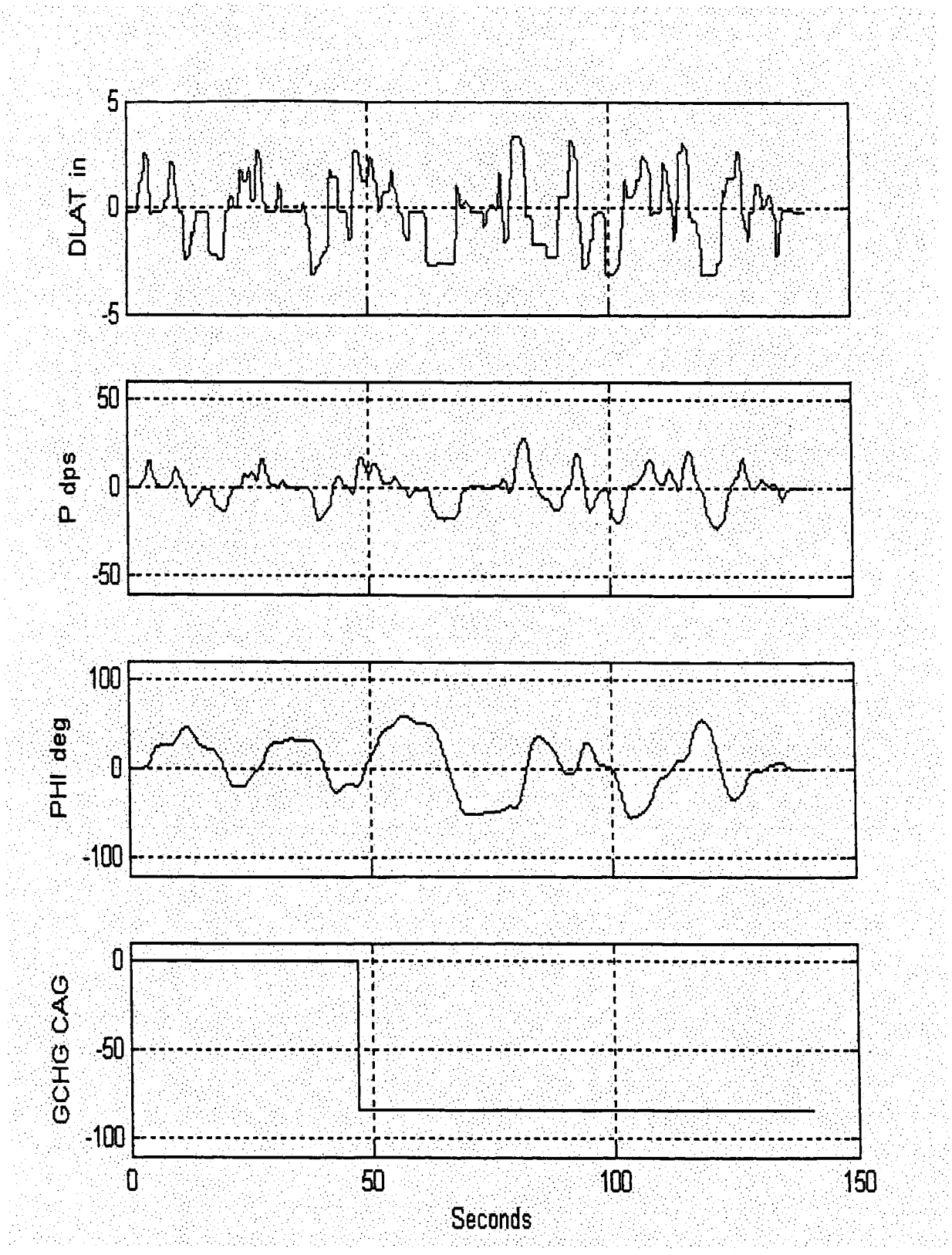
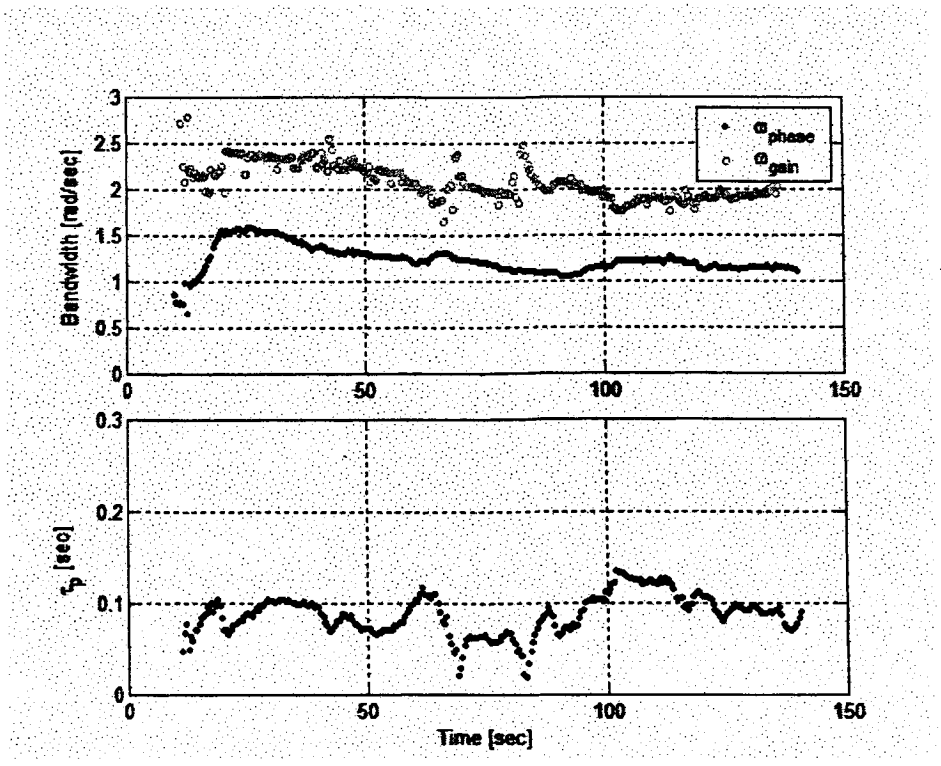
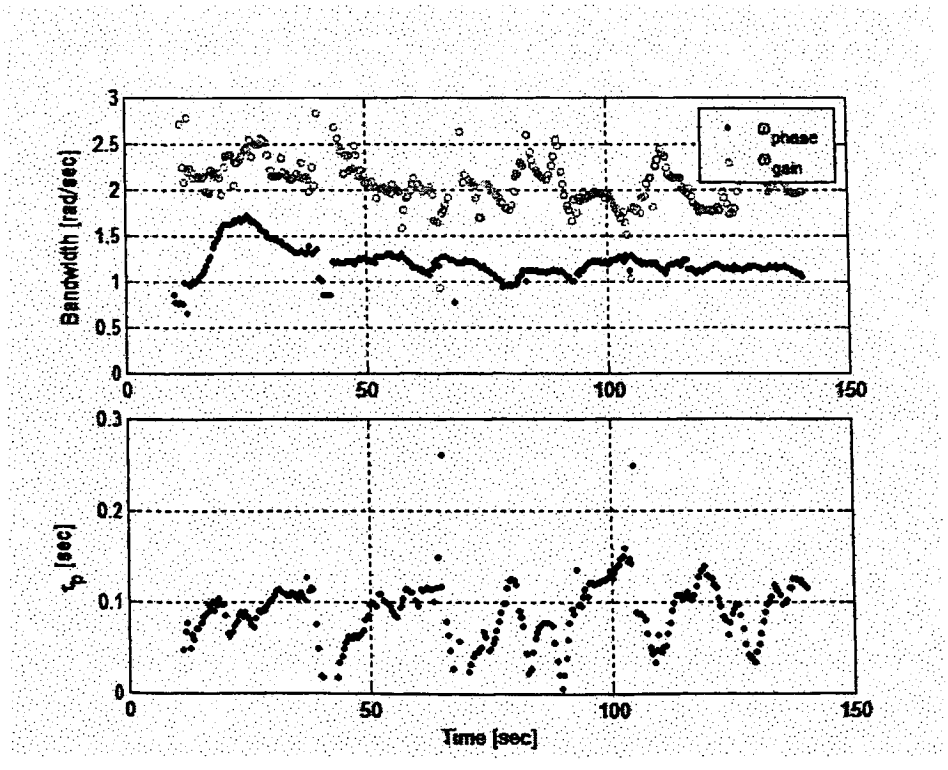


Figure 108. R18 Time Histories, No PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 109. R18 TVTF Analysis

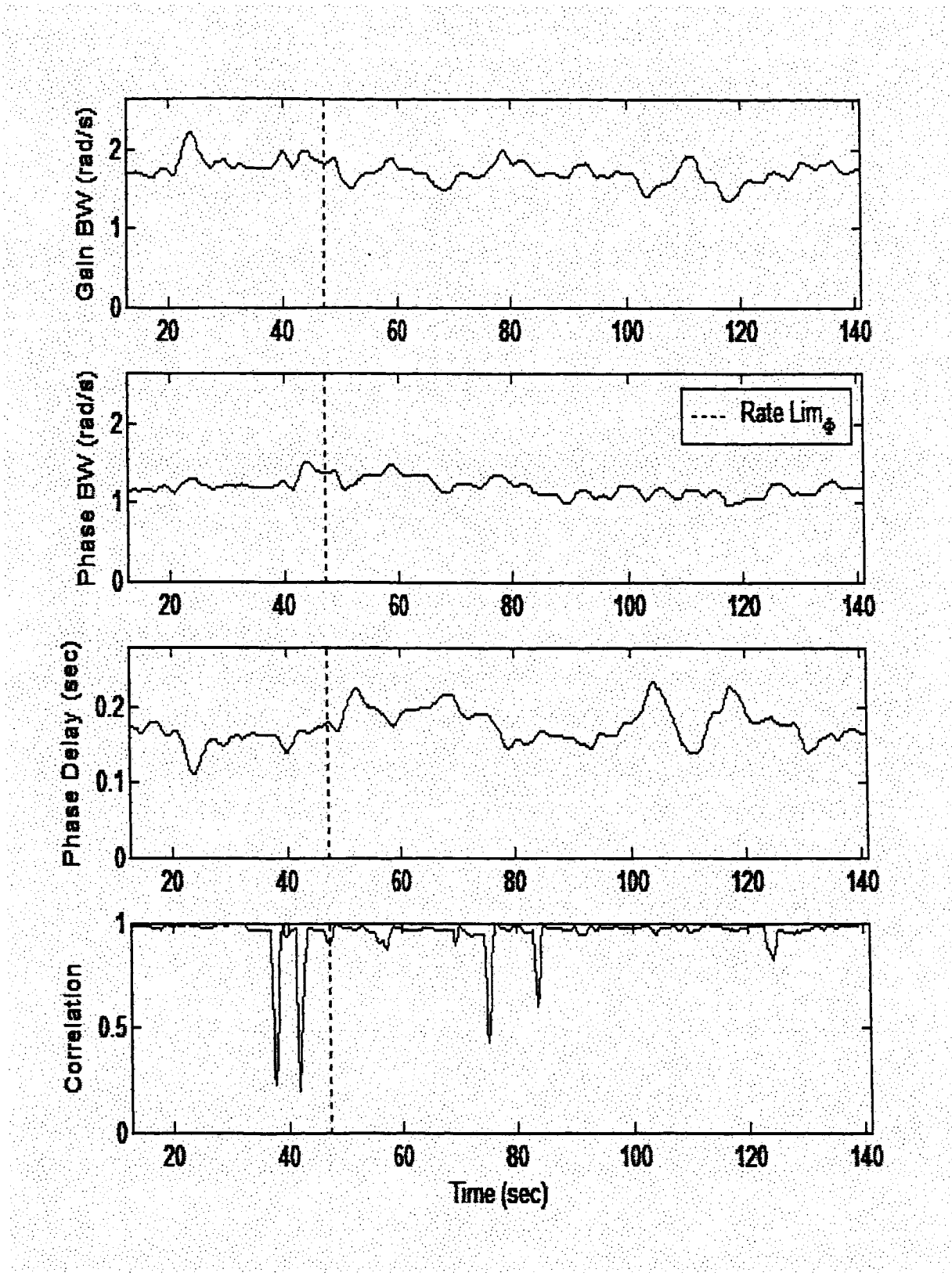


Figure 110. R18 WERA Analysis

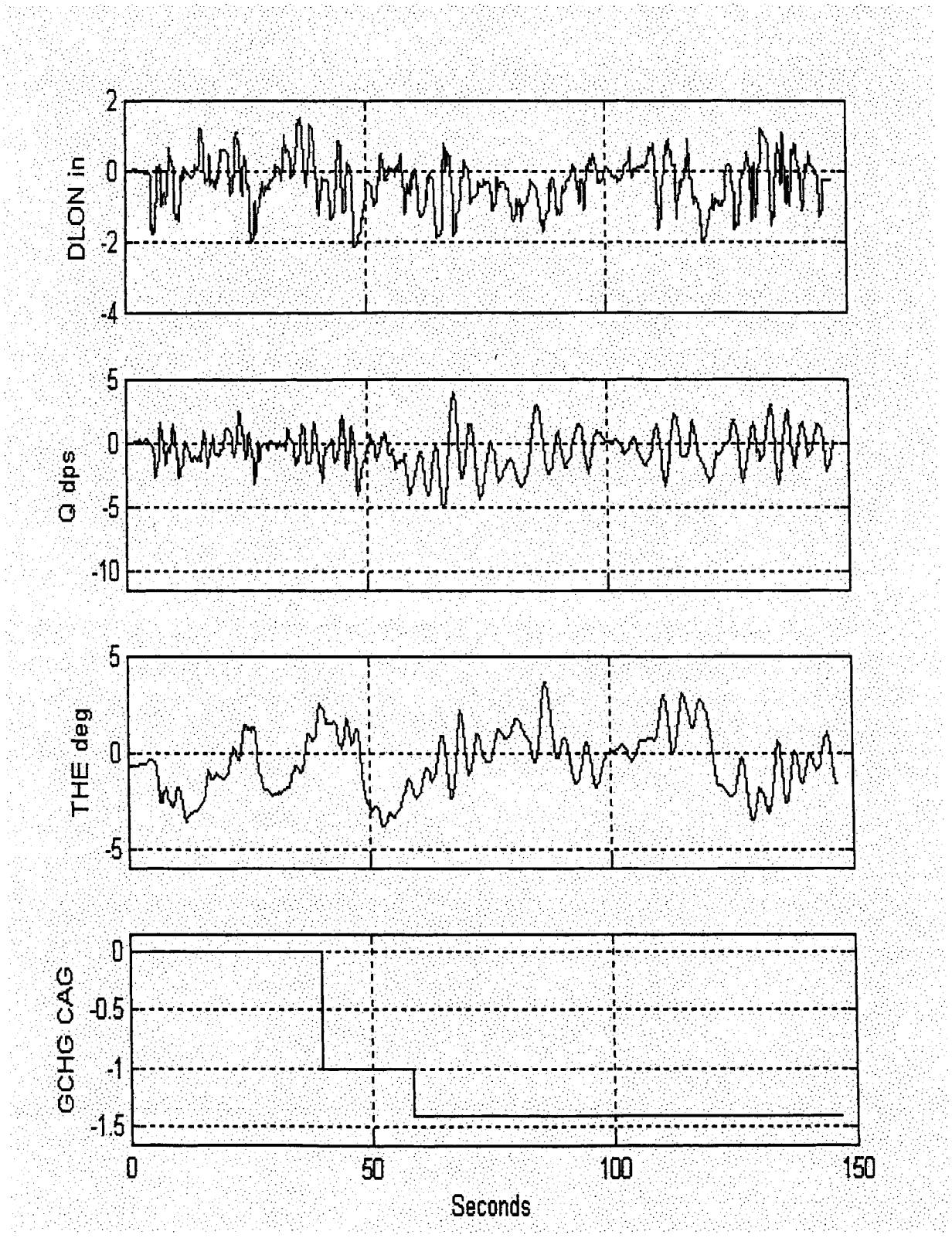
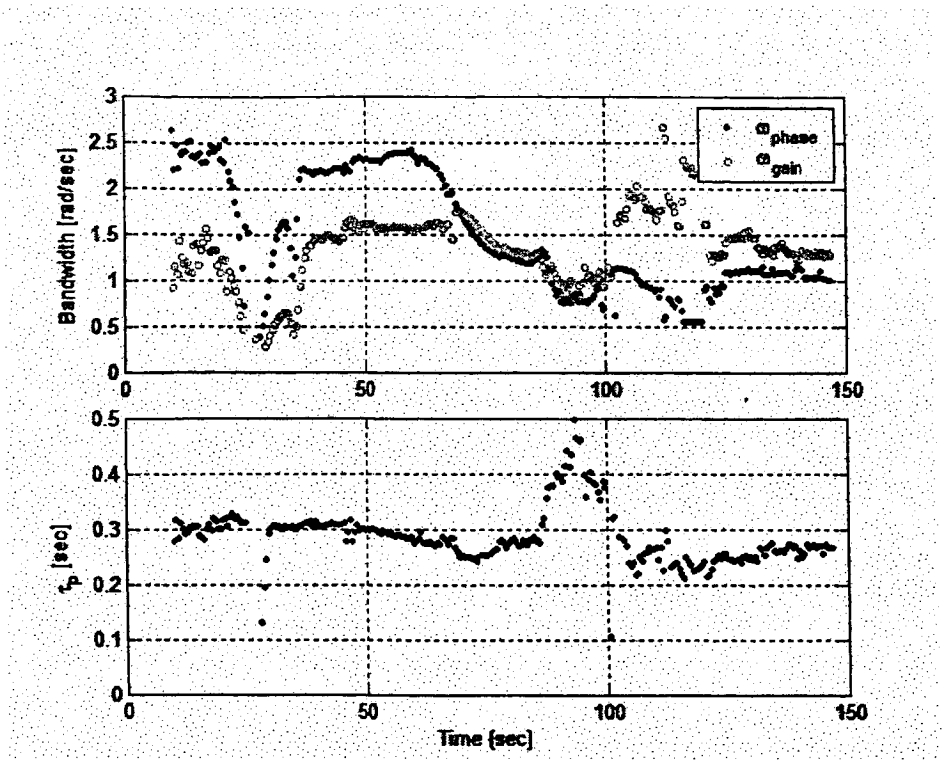
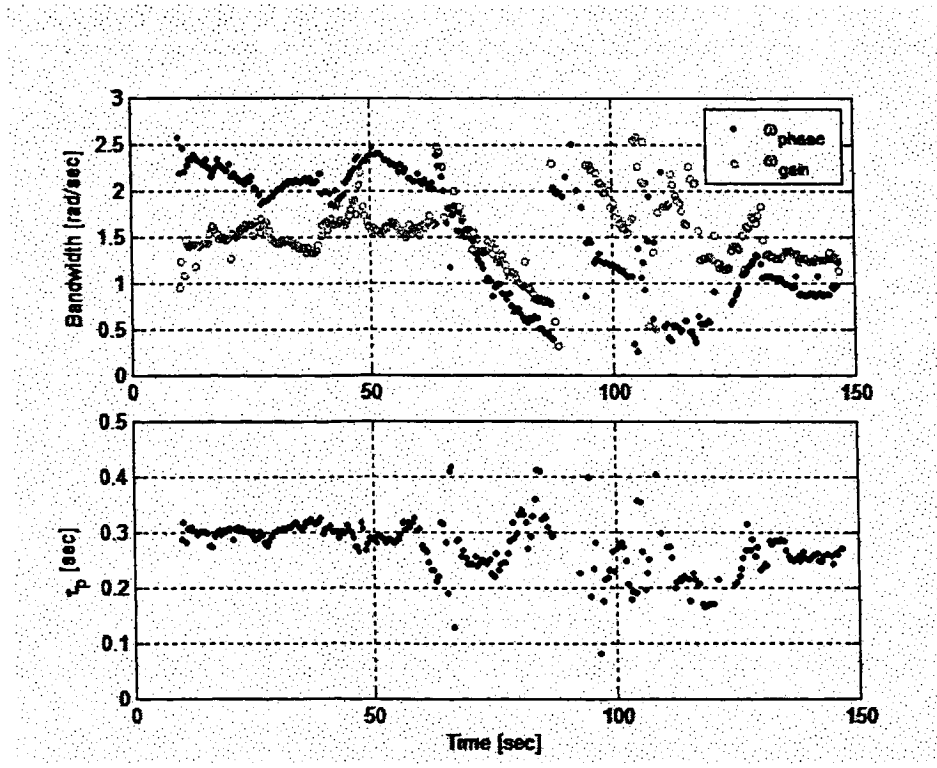


Figure 111. R70 Time Histories, Mild Pitch PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 112. R70 TVTF Analysis

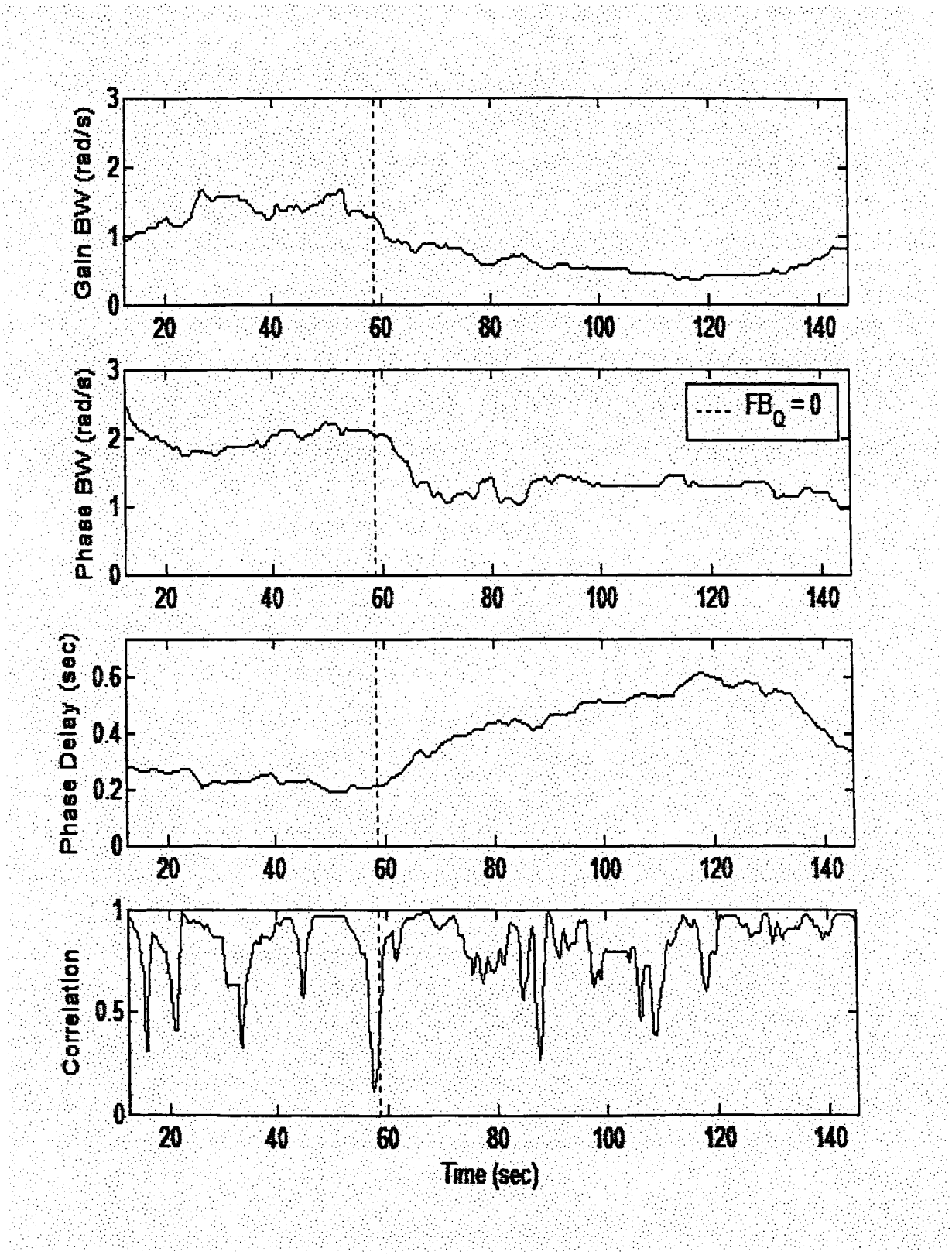


Figure 113. R70 WERA Analysis

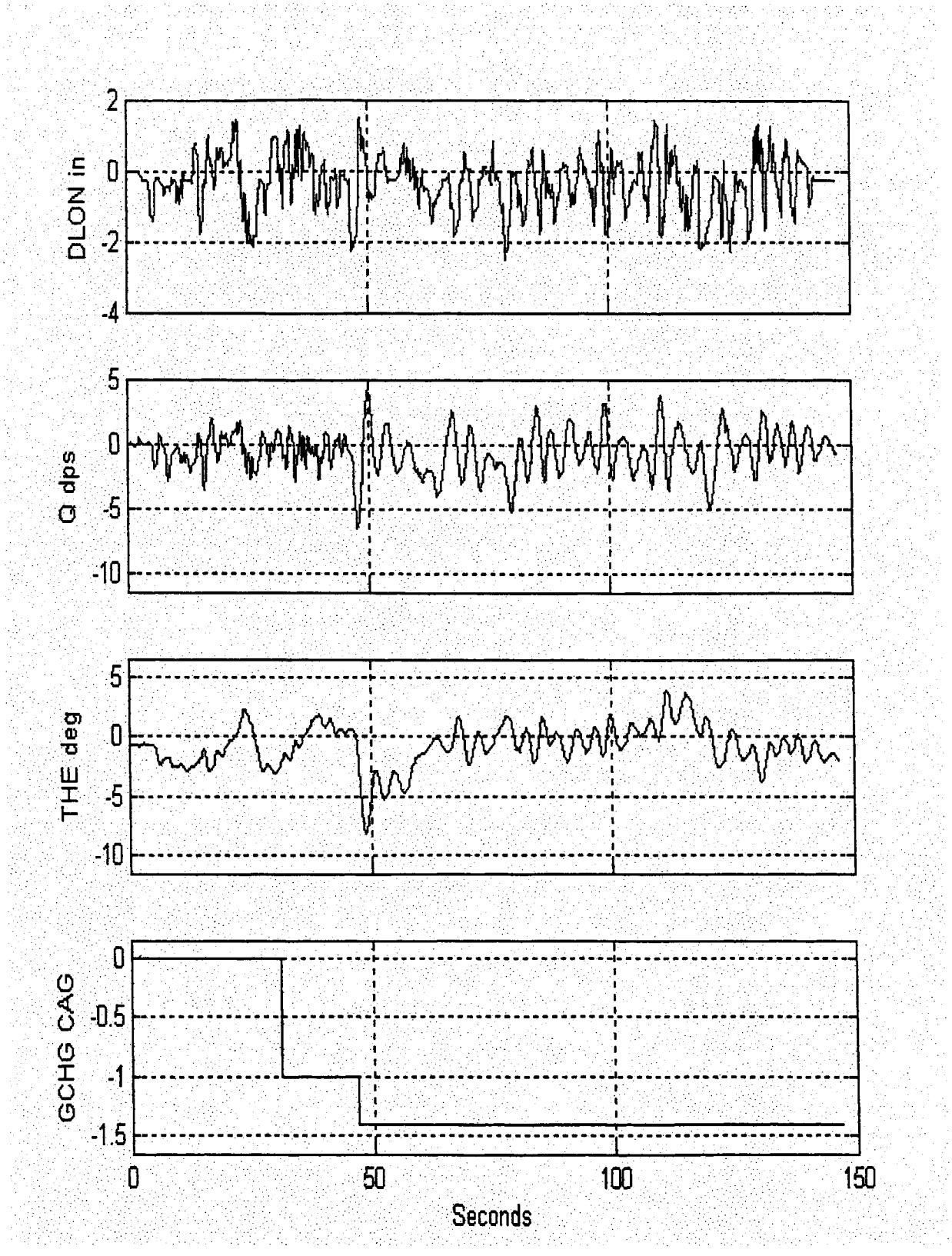
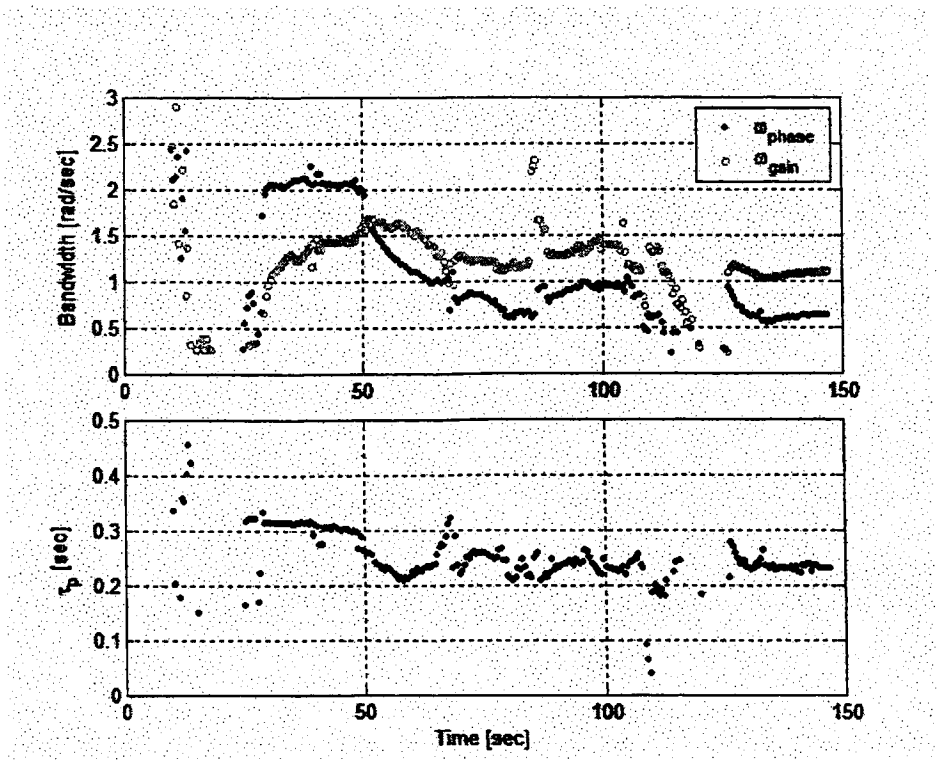
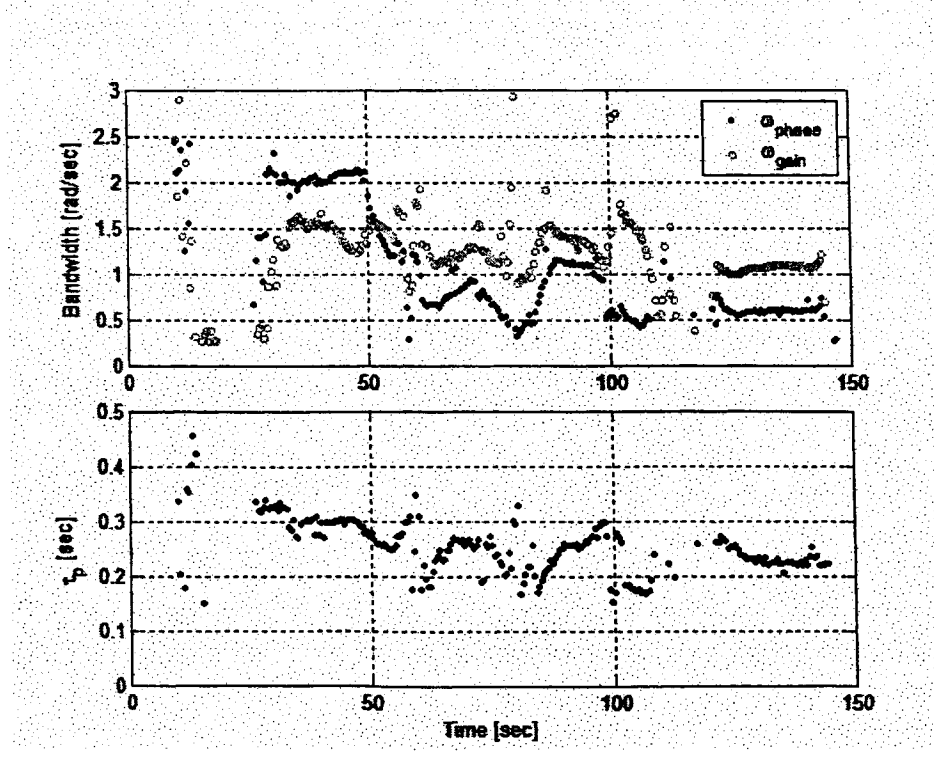


Figure 114. R71 Time Histories, Mild Pitch PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 115. R71 TVTF Analysis

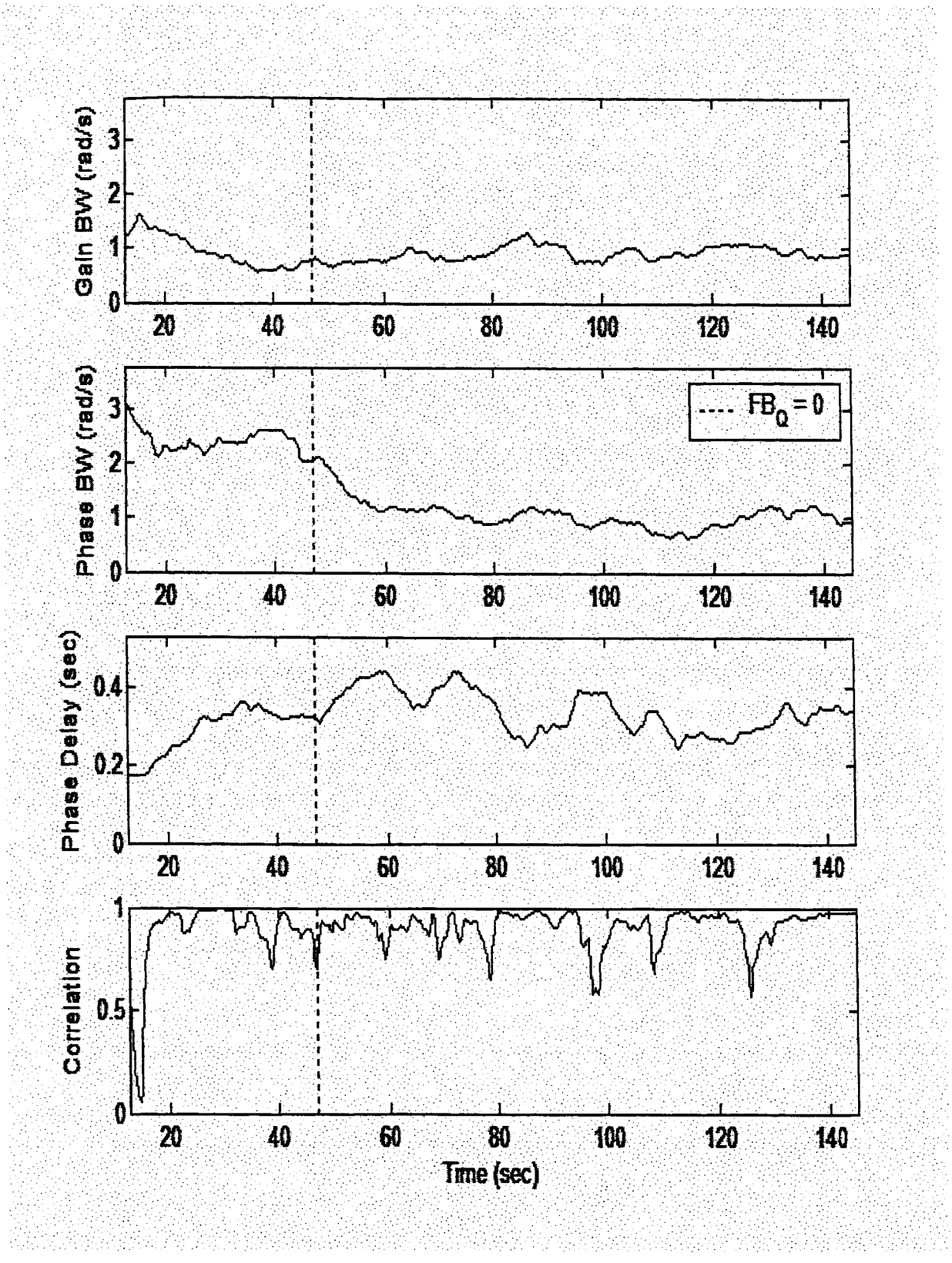


Figure 116. R71 WERA Analysis

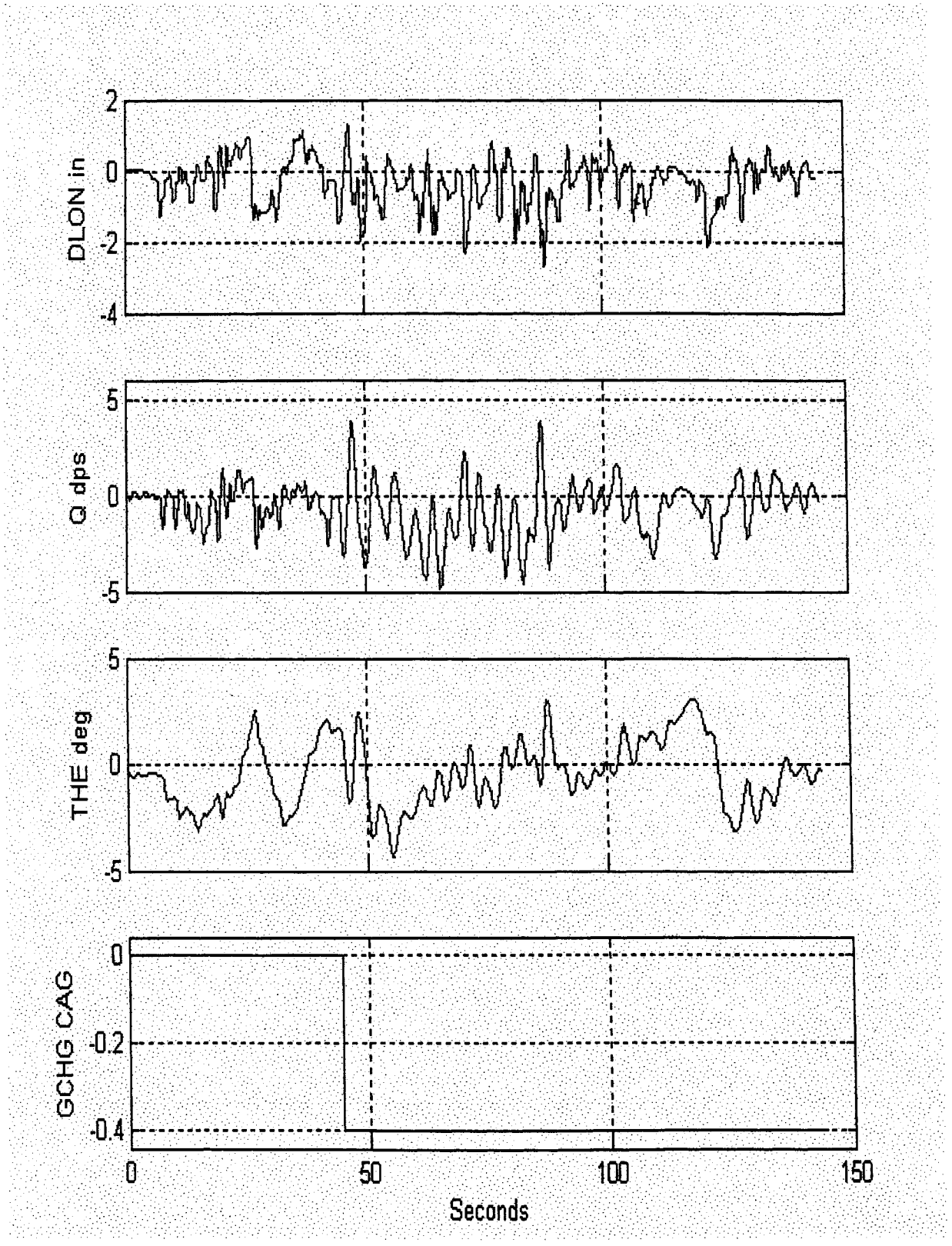
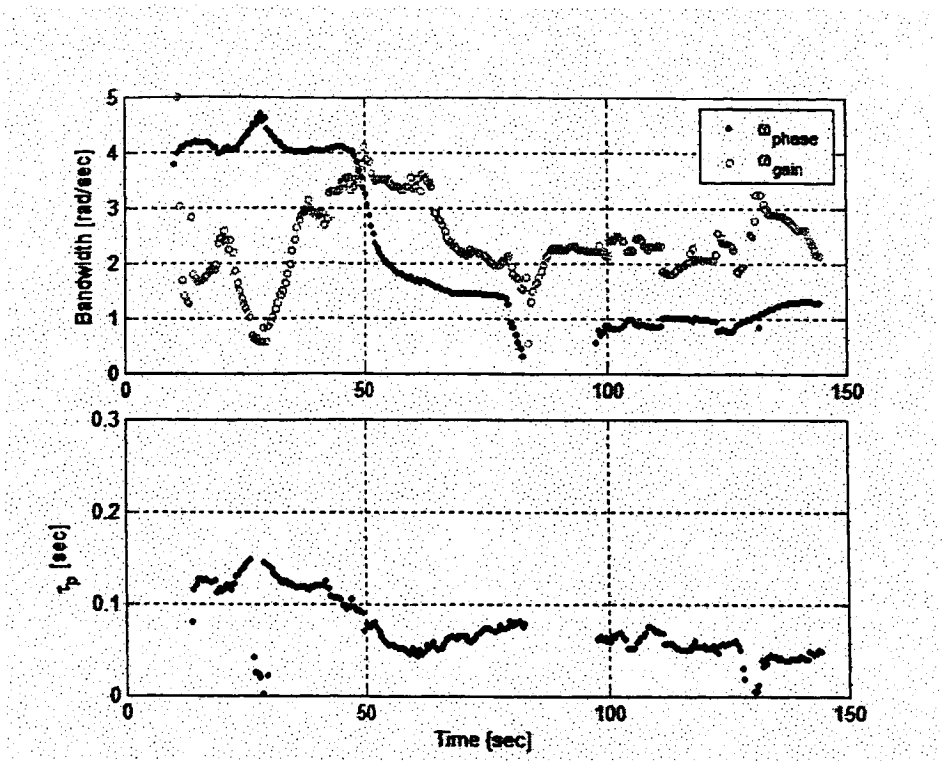
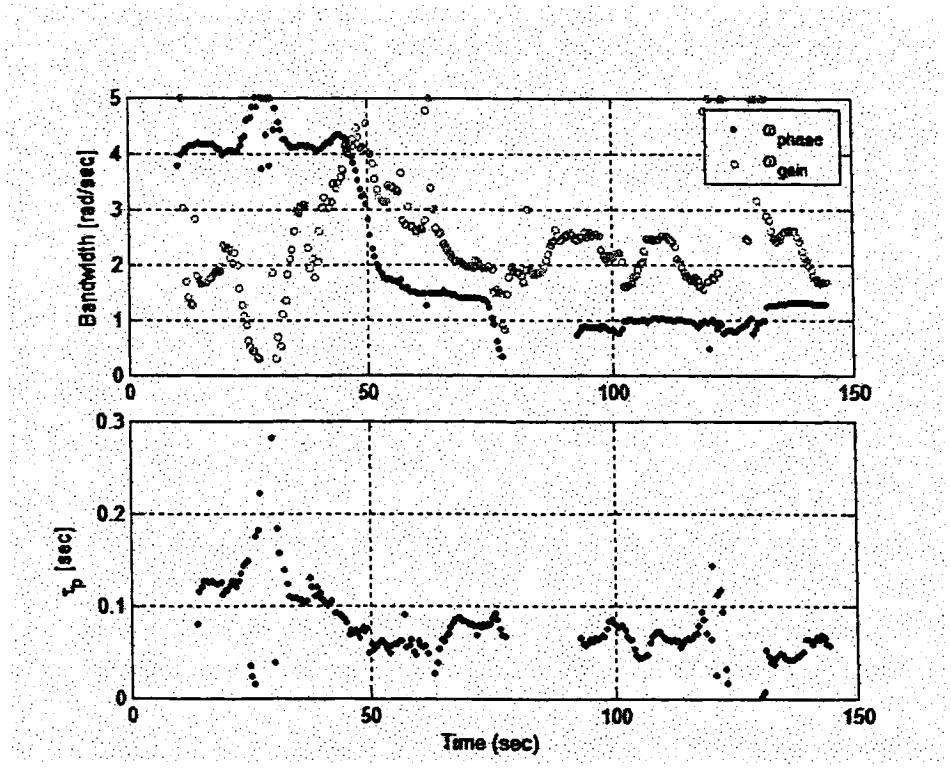


Figure 117. R99 Time Histories, Mild Pitch PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 118. R99 TVTF Analysis

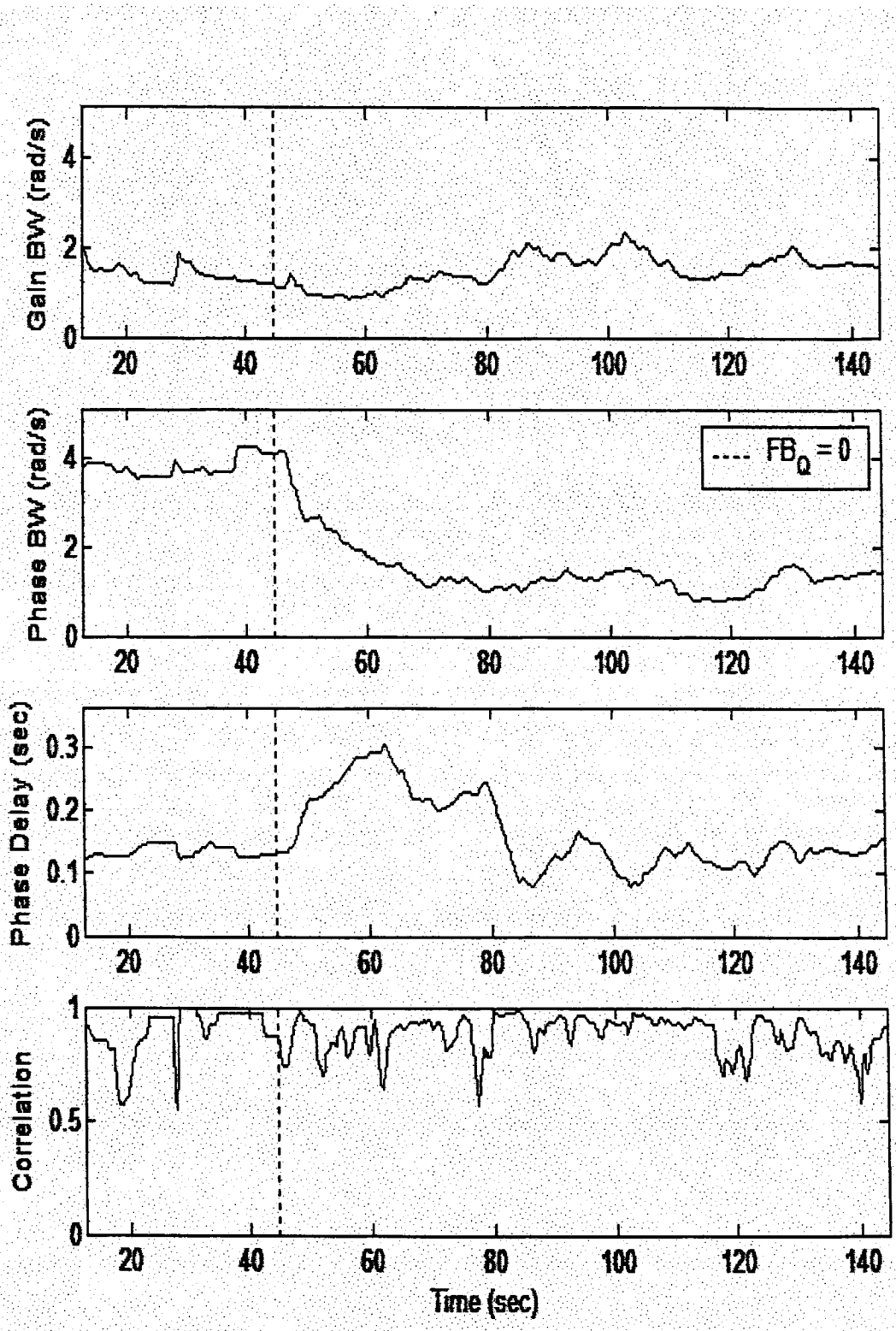


Figure 119. R99 WERA Analysis

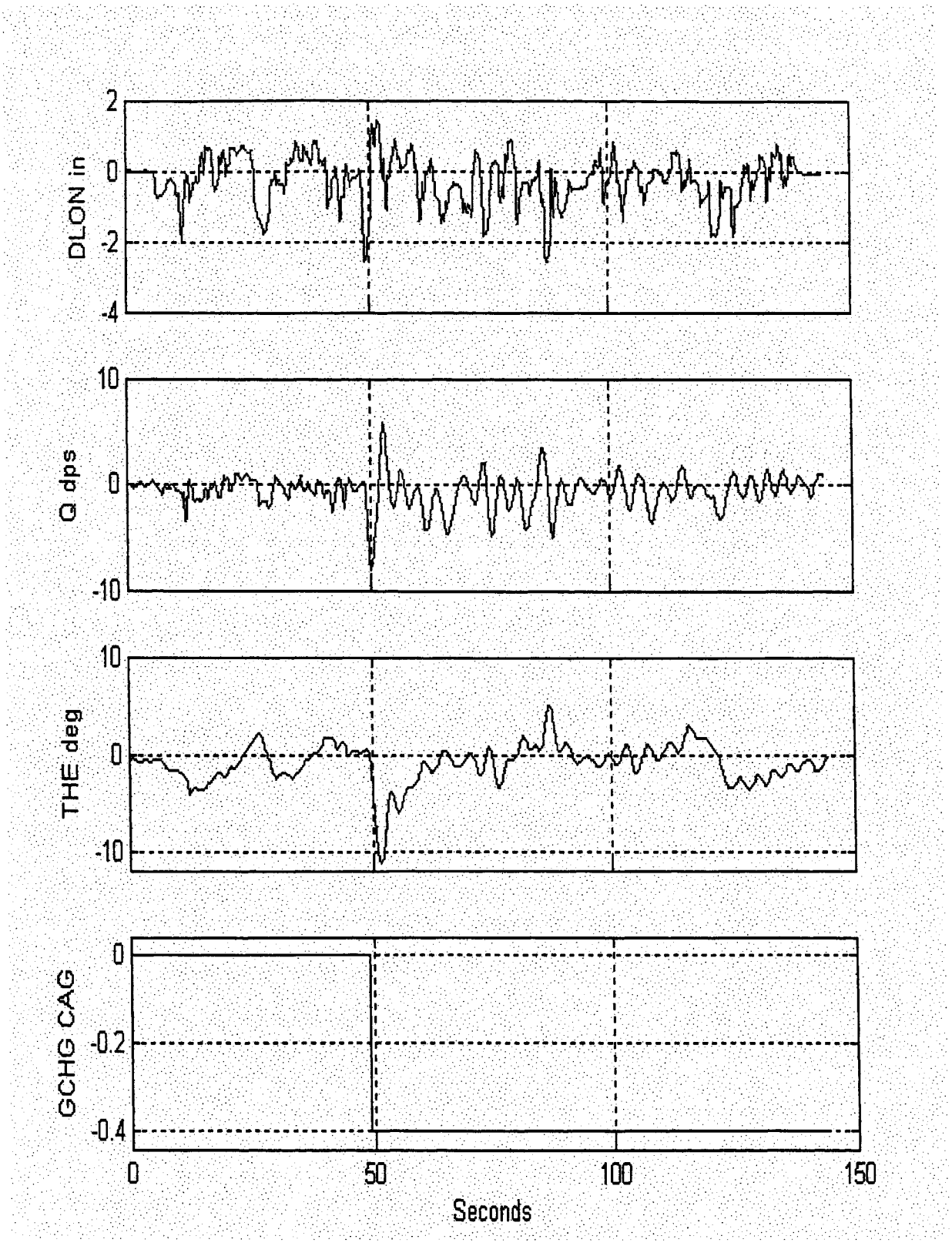
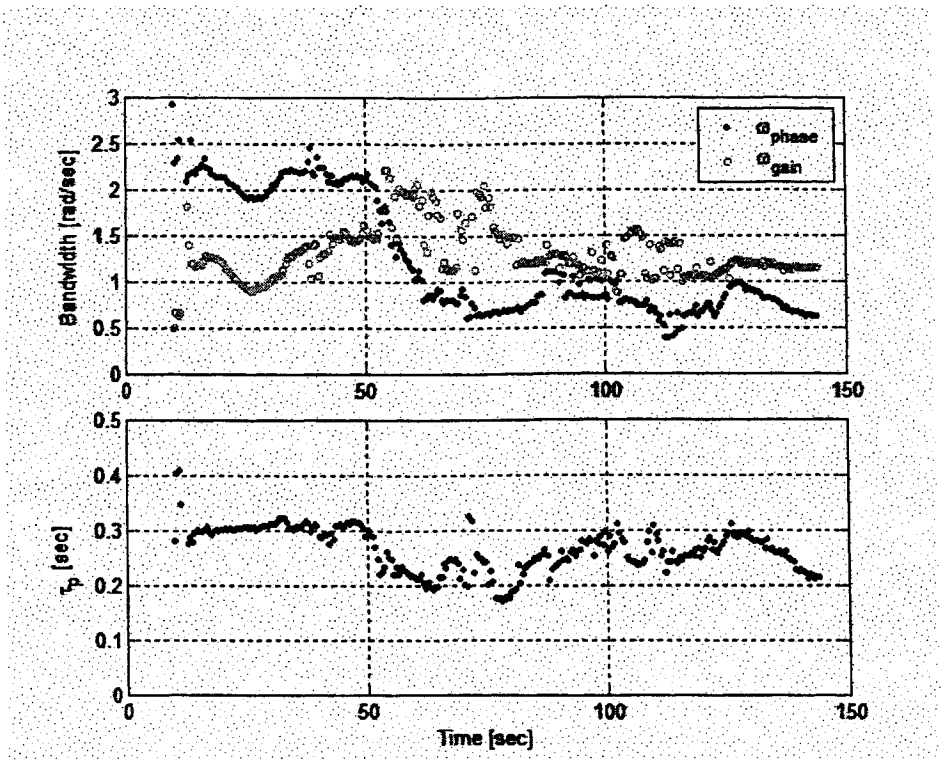
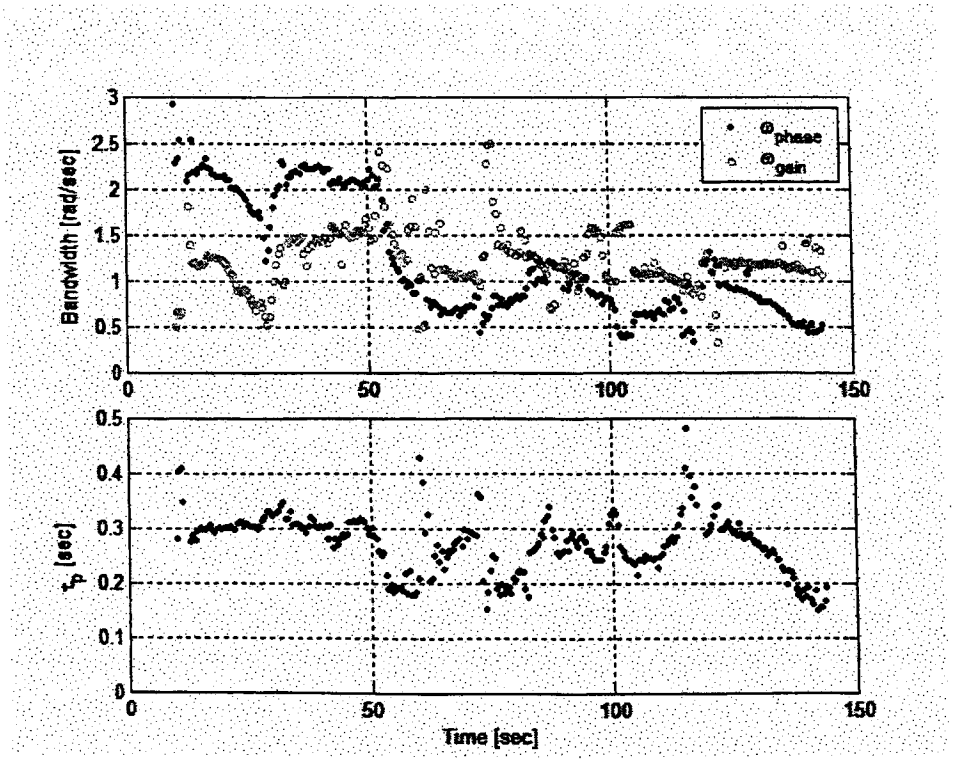


Figure 120. R100 Time Histories, Mild Pitch PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 121. R100 TVTF Analysis

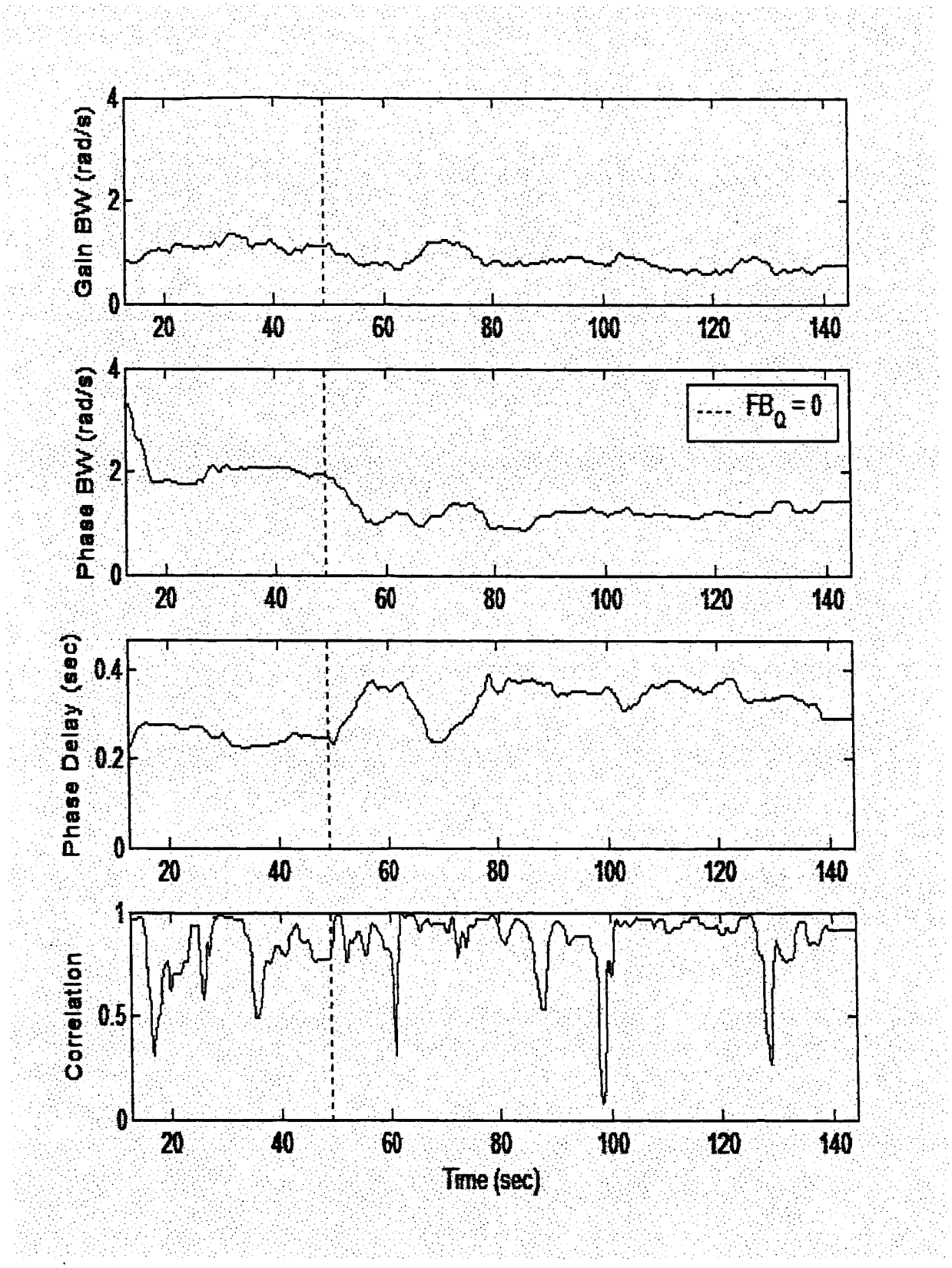


Figure 122. R100 WERA Analysis

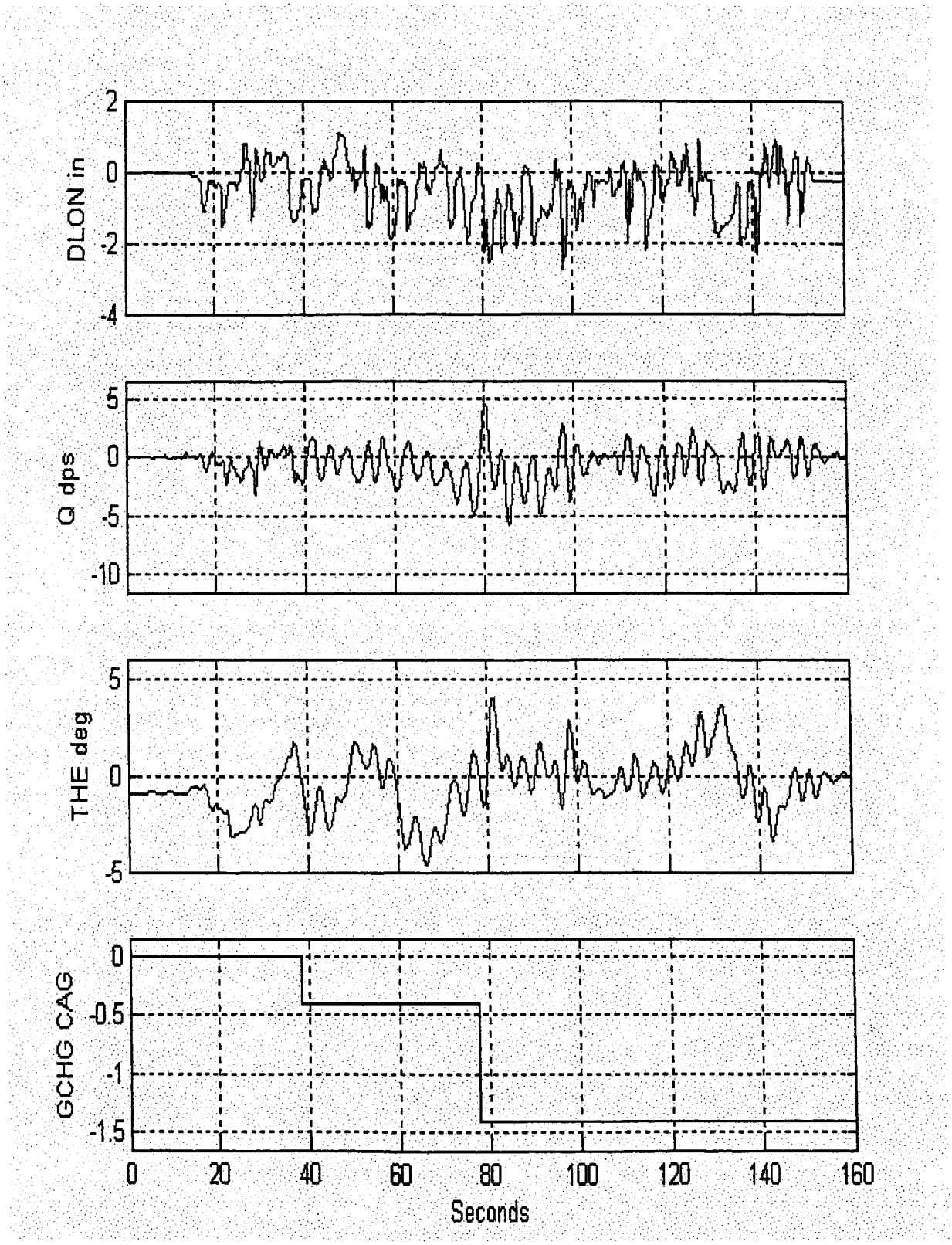
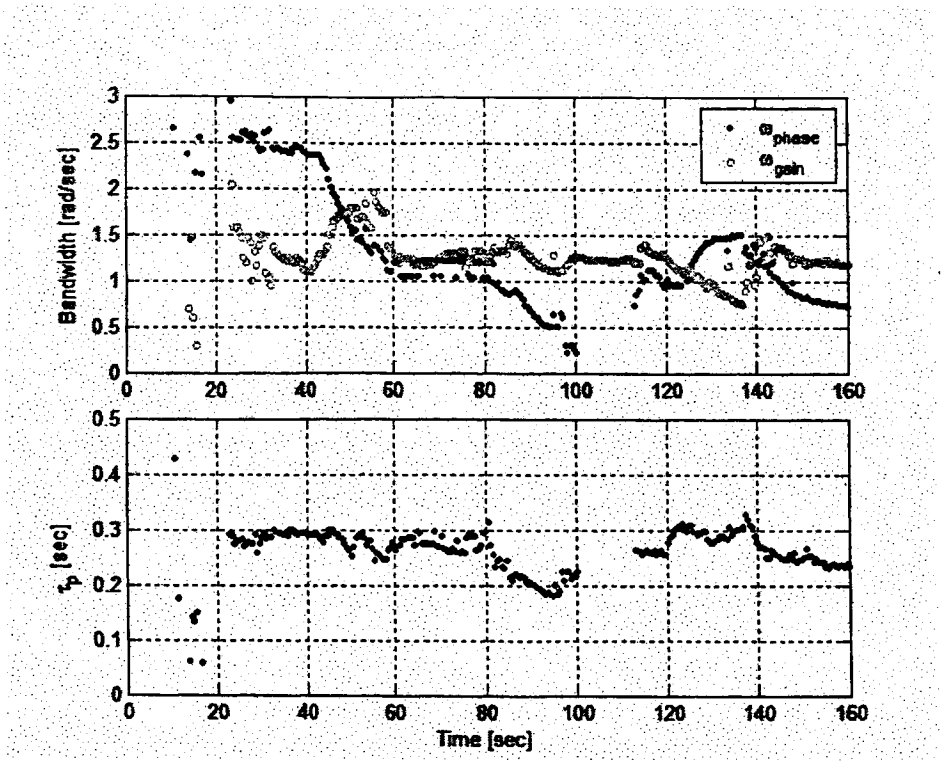
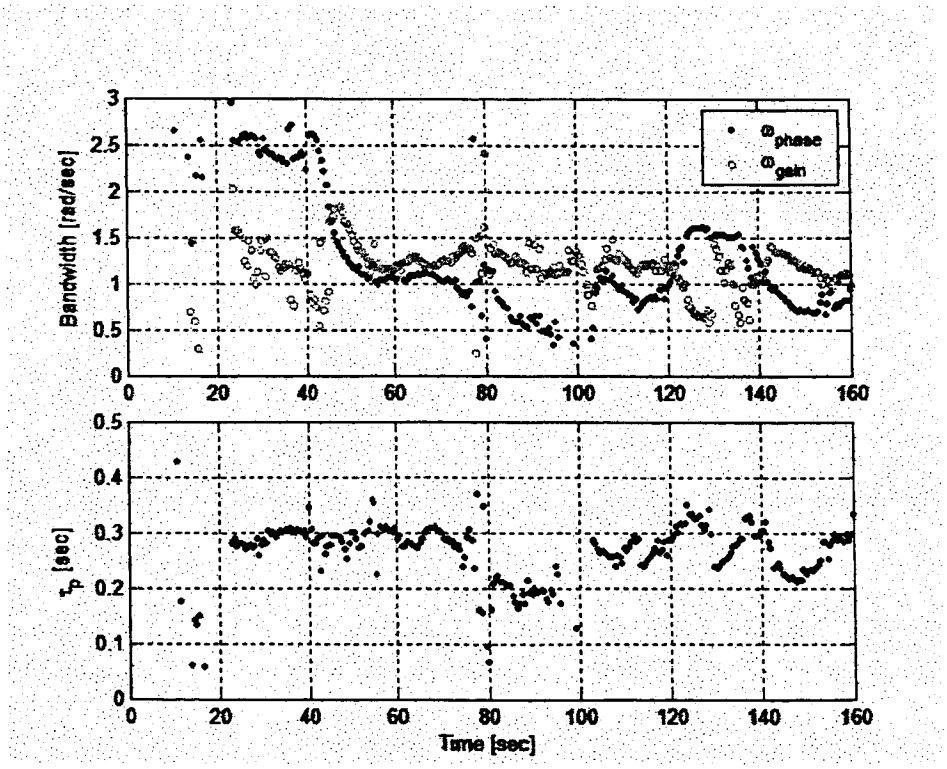


Figure 123. R106 Time Histories, Mild Pitch PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 124. R106 TVTF Analysis

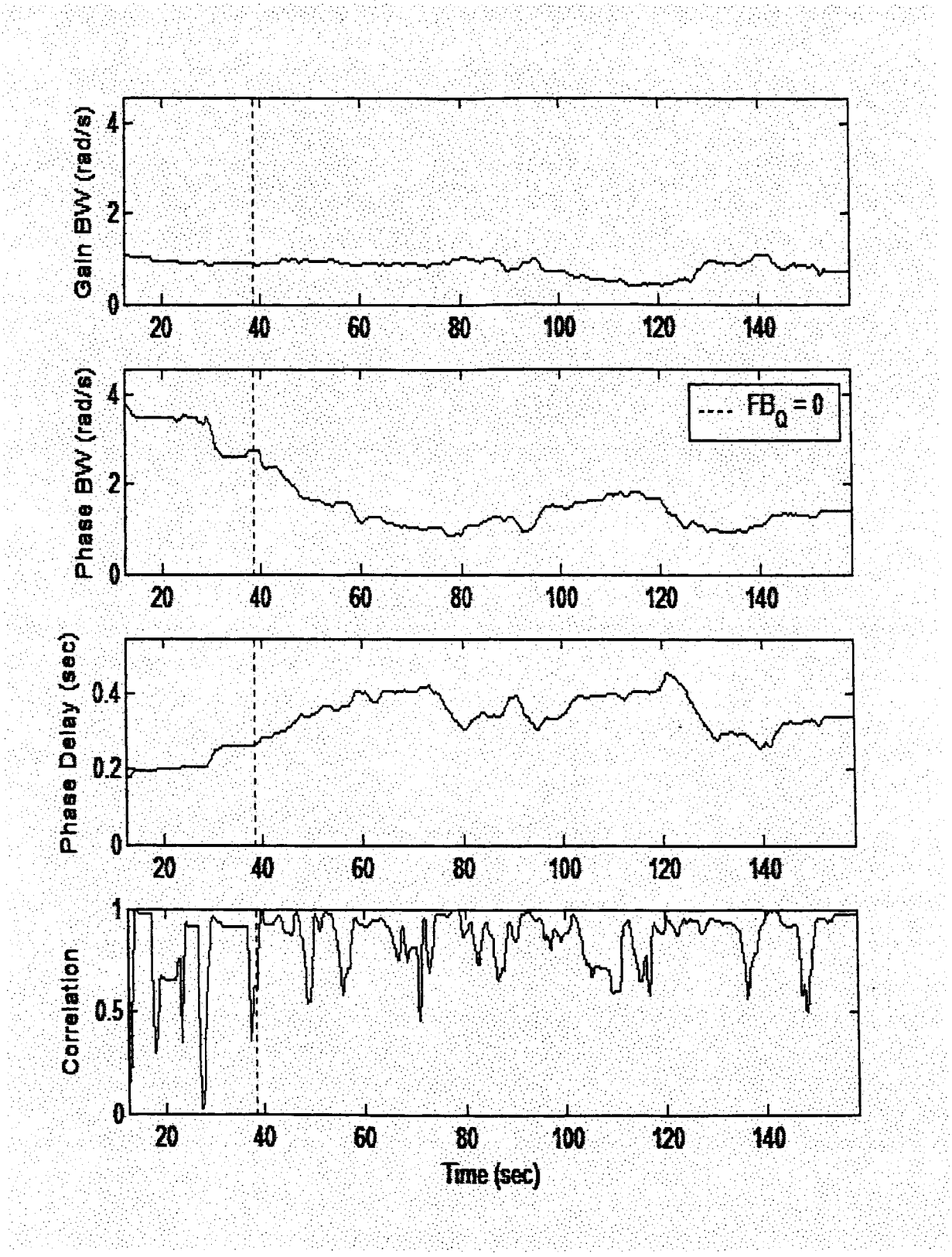


Figure 125. R106 WERA Analysis

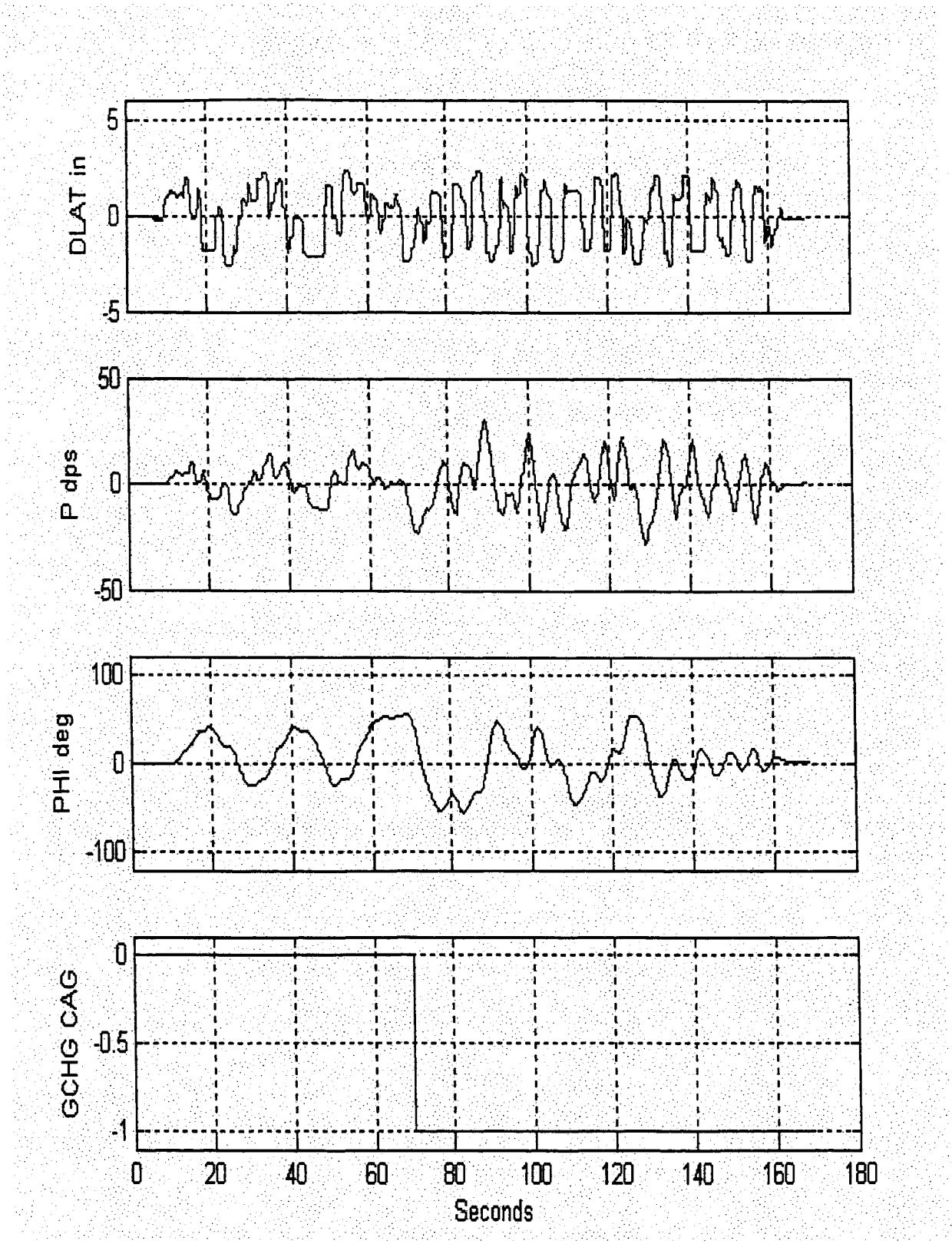
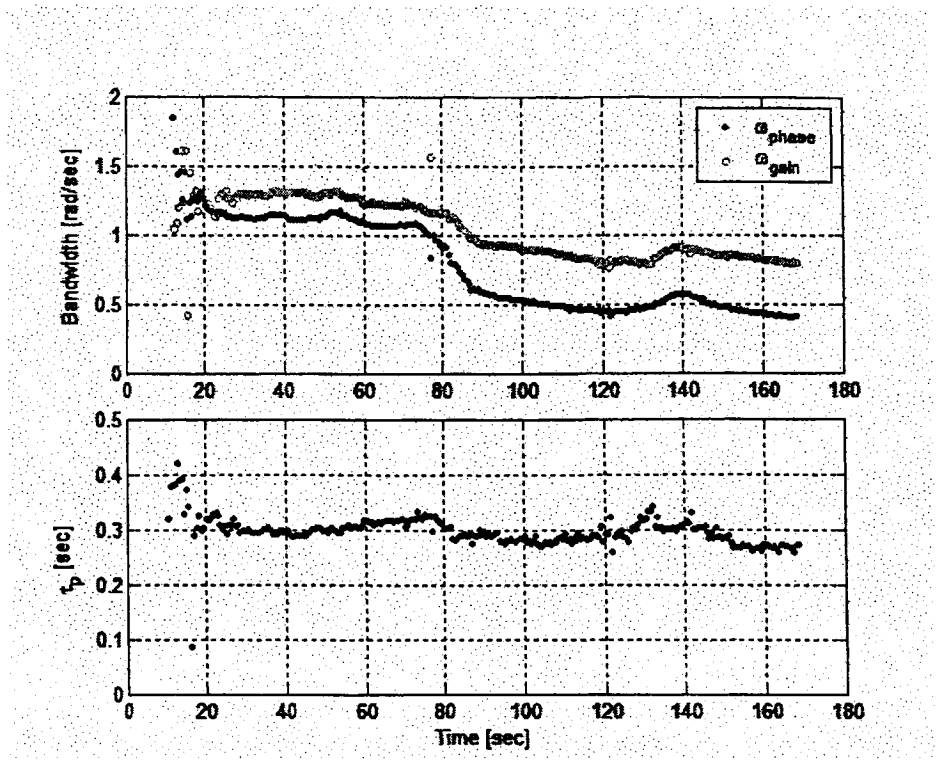
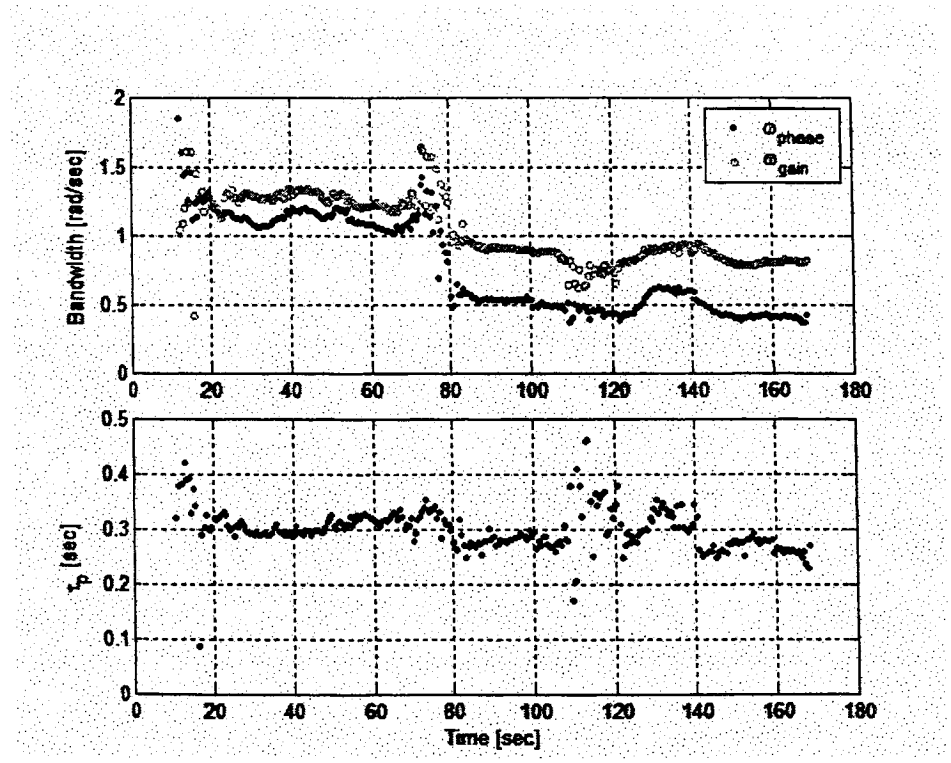


Figure 126. R128 Time Histories, Sustained or Severe Roll PIO



a) 41 Point Time Averaging



b) 21 Point Time Averaging

Figure 127. R128 TVTF Analysis

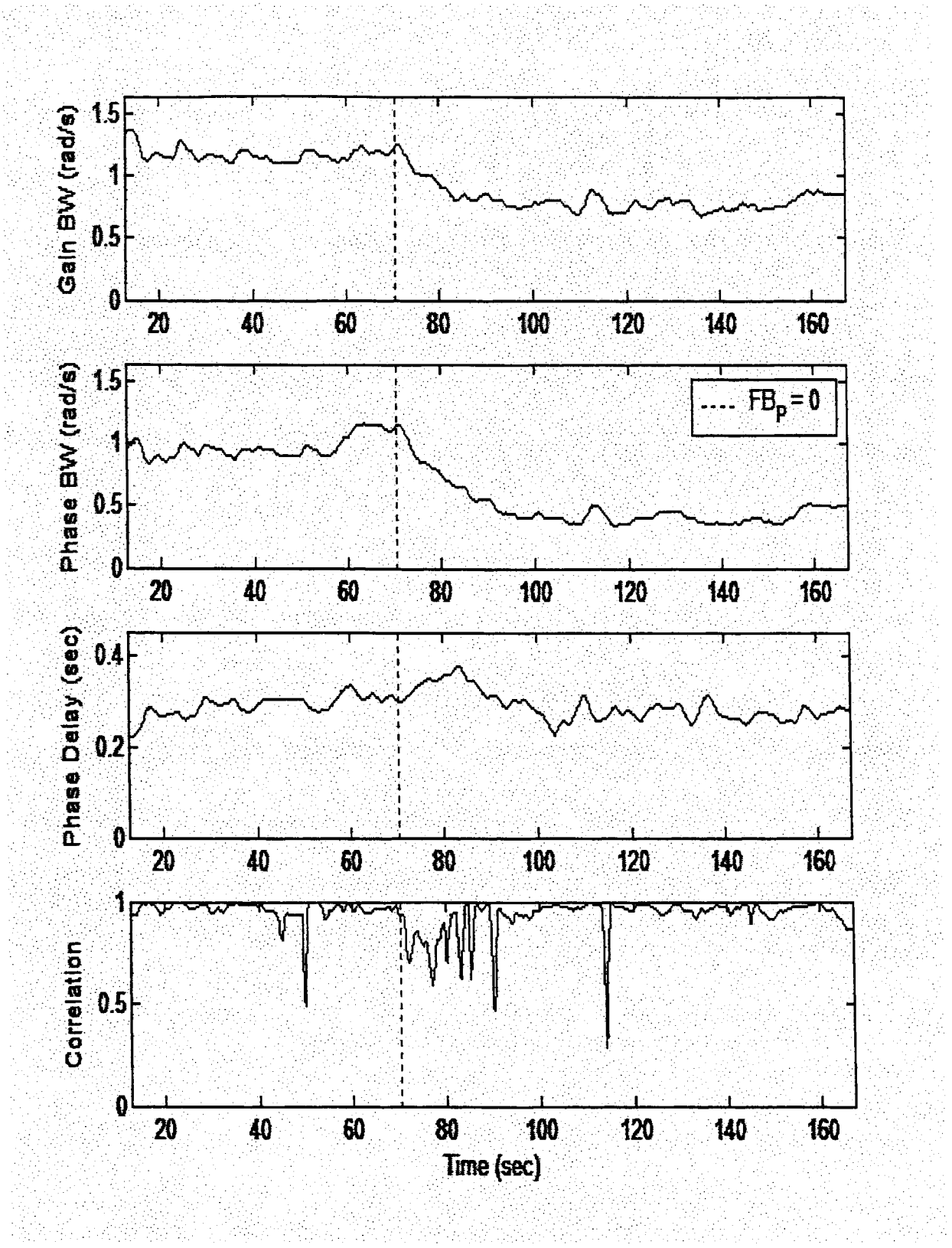


Figure 128. R128 WERA Analysis

5. REFERENCES

1. Klyde, D. H., and D. G. Mitchell, *Handling Qualities Demonstration Maneuvers for Fixed Wing Aircraft Vol. II: Maneuver Catalog*, WL-TR-97-3100, Oct. 1997.
2. Cooper, G. E., and R. P. Harper, Jr., *The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities*, NASA TN D-5153, April 1969.
3. Weingarten, N. C., and C. R. Chalk, *In-Flight Investigation of Large Airplane Flying Qualities for Approach and Landing*, AFWAL-TR-81-3118, Sept. 1981.
4. Neal, T. P., and R. E. Smith, *An In-Flight Investigation to Develop Control System Design Criteria for Fighter Airplanes*, AFFDL-TR-70-74, Volume I, Dec. 1970.
5. Randall, P. A., LT, USN, *et al.*, *A Limited Inflight Investigation of the Neal-Smith Criteria*, AFFTC-TLR-94-27.
6. Kish, B. A., et al, *A Limited Flight Test Investigation of Pilot-Induced Oscillation due to Elevator Rate Limiting*, AFFTC-TR-97-12, June 1997.
7. Mitchell, D. G., B. L. Aponso, and R. H. Hoh, *Minimum Flying Qualities, Volume I: Piloted Simulation Evaluation of Multiple Axis Flying Qualities*, WRDC-TR-89-3125, Vol. I, Jan. 1990.
8. Bailey, R. E., and L. H. Knotts, *Interaction of Feel System and Flight Control System Dynamics on Lateral Flying Qualities*, NASA CR-179445, Dec. 1990.
9. Monagan, S. J., R. E. Smith, and R. E. Bailey, *Lateral Flying Qualities of Highly Augmented Fighter Aircraft*, AFWAL-TR-81-3171, June 1982.
10. Bailey, R. E., and L. H. Knotts, *Interaction of Feel System and Flight Control System Dynamics on Lateral Flying Qualities*, NASA CR-179445, Dec. 1990.
11. *Notice of Change to MIL-STD-1797A, Notice 1*, Department of Defense Interface Standard, 28 June 1995.
12. Klyde, D. H., B. L. Aponso, and D. G. Mitchell, *Handling Qualities Demonstration Maneuvers for Fixed Wing Aircraft Vol. I: Maneuver Development Process*, WL-TR-97-3099, Oct. 1997.
13. Graham, D., and D. McRuer, Analysis of Nonlinear Control Systems, John Wiley & Sons, Inc., New York, NY, 1961.
14. McRuer, D. T., and D. E. Johnston, *Flight Control Systems Properties and Problems*, NASA CR-2500, Feb. 1975.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 30, 2003	Final Technical Report 01/31/01 – 09/30/03

4. TITLE AND SUBTITLE
On-Line Loss of Control Detection Using Wavelets
Volume II: LOCATS Evaluation

5. FUNDING NUMBERS
NAS4-010004

6. AUTHORS
David H. Klyde, Peter M. Thompson, Ph.D., Edward N. Bachelder, Ph.D., Theodore J. Rosenthal

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Technology, Inc.
13766 S. Hawthorne Blvd.
Hawthorne, CA 90250

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-1341-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASA
Dryden Flight Research Center
P.O. Box 273
Edwards, CA 93523-0273

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Wavelet transforms are used for on-line detection of aircraft loss of control. Wavelet transforms are compared with Fourier transform methods and shown to more rapidly detect changes in the vehicle dynamics. This faster response is due to a time window that decreases in length as the frequency increases. New wavelets are defined that further decrease the detection time by skewing the shape of the envelope. The wavelets are used for power spectrum and transfer function estimation. Smoothing is used to tradeoff the variance of the estimate with detection time. Wavelets are also used as front-end to the eigensystem reconstruction algorithm. Stability metrics are estimated from the frequency response and models, and it is these metrics that are used for loss of control detection. A Matlab toolbox was developed for post-processing simulation and flight data using the wavelet analysis methods. A subset of these methods was implemented in real time and named the Loss of Control Analysis Tool Set or LOCATS. A manual control experiment was conducted using a hardware-in-the-loop simulator for a large transport aircraft, in which the real time performance of LOCATS was demonstrated. The next step is to use these wavelet analysis tools for flight test support.

14. SUBJECT TERMS
Loss of control, wavelets, time varying transfer functions, Fourier transforms, eigensystem reconstruction algorithm

15. NUMBER OF PAGES
165

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

SYSTEMS TECHNOLOGY, INC

13766 S. HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250-7083 • PHONE (310) 679-2281
FAX (310) 644-3887

Technical Report No. 1341-3

ON-LINE LOSS OF CONTROL DETECTION USING WAVELETS VOLUME III: LOCATS USER'S MANUAL

September 30, 2003

Peter M. Thompson, Ph.D.
David H. Klyde
Theodore J. Rosenthal
Edward N. Bachelder, Ph.D.

Prepared for
NASA Dryden Flight Research Center
Edwards, CA
Contract No. NAS4-01004

FOREWORD

This report documents the results of a Phase II Small Business Innovation Research (SBIR) contract to develop an on-line wavelet-based loss of control detection tool. The work was conducted by Systems Technology, Inc. (STI) in Hawthorne, CA for NASA Dryden Flight Research Center under Contract No. NAS4-01004. The period of performance for this contract was 31 January 2001 to 30 September 2003. Martin Brenner served as the Contracting Officer's Technical Representative for NASA. Peter M. Thompson was the Principal Investigator and David H. Klyde was the program manager for STI. Simulator facilities and related support were provided by BAE SYSTEMS Platform Solutions Sector in Johnson City, NY. The program manager for BAE SYSTEMS subcontract was Matthew Trouve, while the engineering effort was led by Richard Parkinson. The authors would also like to acknowledge the significant efforts of Jeffrey Gernhart, the simulator facility manager. In addition, the authors recognize the initial BAE SYSTEMS work of Andrew Corea, Timothy Horan, and Feng Liang. Finally, the authors would like to thank Douglas Thrall of BAE SYSTEMS for his support of this program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This three volume report documents the results of a Phase II Small Business Innovation Research (SBIR) contract to develop an on-line means to predict aircraft loss of control using wavelets. Systems Technology, Inc. (STI) conducted the work for NASA Dryden Flight Research Center at Edwards, CA. The overall objective of this proposed program was to apply wavelet technology in the form of a Loss of Control Analysis Tool Set or LOCATS as a means of predicting loss of control for the broad range of refractory automatic and manual control system problems. These control problems are those that escape detection by typical design criteria and methodologies, surface under unusual or rare circumstances, and threaten flight safety. Specific Phase II technical objectives were as follows:

- Extend the capabilities of the LOCATS to include improved wavelet formulations for real-time applications with validity measures;
- Identify time varying pilot-vehicle system characteristics that can be an indicator of impending loss of control;
- Identify, develop and validate wavelet-based time varying metrics to predict loss of control of automatic and manual flight control systems and to monitor overall flight control system health;
- Develop software and/or hardware solutions for the implementation of the real-time LOCATS;
- Provide a graphical user interface, real-time graphics displays, and networking capabilities for the prototype real-time LOCATS; and
- Evaluate the prototype real-time LOCATS in a piloted simulation that features flight control system software and hardware representative of current state-of-the-art aircraft.

The evaluation of the prototype LOCATS was conducted using the flight control system hardware-in-the-loop simulator facility of BAE SYSTEMS Platform Solutions Sector (here after referred to as BAE SYSTEMS) of Johnson City, NY.

Volume I of this report begins with an overview of issues involving aircraft loss of control. This is followed by a description of the LOCATS software, detailed development of the wavelet-based loss of control detection techniques, and a discussion of the results from the LOCATS piloted simulation evaluation. Supporting material is provided in the appendices. Volume II of this report provides the LOCATS piloted simulation test plan and detailed descriptions of the simulation checkout and formal evaluation. Finally, Volume III of this report contains the LOCATS User's Guide.

2. GETTING STARTED

The LOCATS software has the following parts:

Data collection: Real time data transfer from the Mil 1553 data bus or pseudo real time playback from ASCII files.

Configuration management: Selection of data source and selection of plots.

Displays: Real time graphical displays of time series and transforms.

Wavelet and FFT Transforms: Calculations for power spectra, transfer functions, and stability parameters.

This user's guide explains the operation of the program for each of these parts.

The software runs in a Windows operating system (Win98, Win2000, or WinXP). The visual and data collection parts of the program are written in Visual Basic, and the transform calculations are written in C and contained in a dynamic link library or DLL. The program is available either as a compiled program, or it can be run inside of the Visual Basic development environment. The latter is more flexible for making changes, but requires some knowledge of the software. The program is started from an icon, located on the start menu, the desktop, or in the explorer. The icon in Figure 1 is the default for a Visual Basic Program.

Figure 1. Starting LOCATS

Install the program using a standard Windows install shield, which loads the executable file, various DLL files that contain support software, data files collected on the BAE SYSTEMS simulator, and sample configuration files. The real time data transfer is specific to the SBS Technologies Mil 1553 data bus card used to read signals from the BAE SYSTEMS simulator, and changes to the software and perhaps new hardware will be required to support other data interfaces. Start the program, the splash screen in Figure 2 will be displayed for a few seconds, and then you are ready for configuration management.

Figure 2. LOCATS Splash Screen

3. DATA COLLECTION

3.1 Real Time

The real time signals are read from the Mil 1553 bus. The signal names, scale factors, and bus locations are stored in an ASCII file with file type "sig." A sample is shown in Figure 3.

```
20000
Q,5.49316E-03,-180,180,12,21,13,1,Pitch Rate (deg/sec)
VCAS,3.1250E-02,-1024,1024,12,1,4,1,Calibrated Airspeed (kt)
AZ,3.1250E-02,-1024,1024,12,4,30,1,Vertical Acceleration (ft/sec^2)
THE,5.49E-03,-90,90,25,2,11,1,Pitch Attitude (deg)
PHI,1.2207E-04,-4,4,12,17,25,1,Roll Attitude (rad)
P,5.4932E-03,-180,180,12,22,11,1,Roll Rate (deg/sec)
```

Figure 3. Example .sig File

The .sig file is read by LOCATS. The file format is defined as follows:

First line is max # data points that will be collected.

Each additional line defines a signal. Using the first signal as an example:

Q = name of signal that appears in dialog boxes and on plots

5.49316E-03 = scale factor that multiplies data received from the bus.

-180 = minimum value, initial value used by dialog box

180 = maximum value, initial value used in dialog box

12 = Mil 1553 address for this signal (RT Address)

21 = Mil 1553 address for this signal (Sub Address)

13 = Mil 1553 address for this signal (Data word within sub address)

1 = BAE SYSTEMS simulator bus number

Pitch Rate (deg/sec) = detailed description of signal (not used in LOCATS)

For remaining details please see the documentation for the Mil 1553 bus. The LOCATS code that reads and interprets the Mil 1553 bus is unique to the experimental setup used at BAE SYSTEMS and will likely have to be modified for a different site.

3.2 Recorded Data

Data stored in ASCII files can be loaded and then played back in pseudo-real time (never faster, sometimes slower). Part of a data file is shown in Figure 4. The file structure is defined by:

Header on first 3 lines can contain any information and is ignored.

Fourth line has signal names, 9 characters per name with blank fill.

Data begins on fifth line, one number per signal separated by blanks.

The "File/Load Data" menu item and the "R2D2" toolbar icon loads the recorded data.

The "File/Save Data" menu item saves recorded real time data in this format.

```
LoCats Build 1.00.01
Real-Time data collected
July 30, 2003  2:26 PM
Time      Q          VCAS      AZ         THE        PHI        P          VT         R          TPSI       ALPHA
  0.0000000e+000  2.7465800e-001  2.6825000e+002  3.1250000e-001 -7.2468000e-001 -4.8828000e-003
  2.0000870e-002  2.7465800e-001  2.6825000e+002  3.1250000e-001 -7.1919000e-001 -4.8828000e-003
  4.0002290e-002  2.9663060e-001  2.6825000e+002  3.1250000e-001 -7.1370000e-001 -4.8828000e-003
  6.0005670e-002  2.9663060e-001  2.6825000e+002  1.8125000e+000 -7.0821000e-001 -4.8828000e-003
  8.0006820e-002  3.0761690e-001  2.6828130e+002  1.8125000e+000 -7.0272000e-001 -4.8828000e-003
  1.0000910e-001  3.0761690e-001  2.6828130e+002  1.8125000e+000 -6.9723000e-001 -5.3710800e-003
  1.2001190e-001  3.1311010e-001  2.6828130e+002  1.8125000e+000 -6.8076000e-001 -5.3710800e-003
  1.4001420e-001  3.1311010e-001  2.6828130e+002  2.5000000e+000 -6.8076000e-001 -5.3710800e-003
  1.6001560e-001  3.0761690e-001  2.6843750e+002  2.5000000e+000 -6.7527000e-001 -5.3710800e-003
  1.8001900e-001  3.0761690e-001  2.6843750e+002  2.5000000e+000 -6.6978000e-001 -5.3710800e-003
  2.0002070e-001  2.9663060e-001  2.6843750e+002  2.5000000e+000 -6.6429000e-001 -5.3710800e-003
  2.2002210e-001  2.9663060e-001  2.6843750e+002  1.4687500e+000 -6.5880000e-001 -5.3710800e-003
  2.4002440e-001  2.8015120e-001  2.6828130e+002  1.4687500e+000 -6.5331000e-001 -5.3710800e-003
  2.6002720e-001  2.8015120e-001  2.6828130e+002  1.4687500e+000 -6.4782000e-001 -5.3710800e-003
  2.8002940e-001  2.4719220e-001  2.6828130e+002  1.4687500e+000 -6.4233000e-001 -5.3710800e-003
  3.0003170e-001  2.4719220e-001  2.6828130e+002  7.8125000e-001 -6.3684000e-001 -5.3710800e-003
  3.2003340e-001  2.0324690e-001  2.6787500e+002  7.8125000e-001 -6.3135000e-001 -5.3710800e-003
  3.4003570e-001  2.0324690e-001  2.6787500e+002  7.8125000e-001 -6.2586000e-001 -5.3710800e-003
```

Figure 4. .dat Data File

4. CONFIGURATION MANAGEMENT

4.1 The Main Screen

Start the program and the main screen shown in Figure 5 will appear. The main screen has the follow parts:

Menu bar: File operations. There is currently no on-line help.

Tool bar: Most of the operations are started from the toolbar:

Icon1 (Save): Save collected data. Enabled for real time operations.
Icon2 (Tools): Configuration options. Not currently used.
Icon3 (R2D2): Load data from file.
Icon4 (6-Plot): Add new plots. Creates a new branch in the Active Plots panel.
Icon5 (Jet): Begin data collection. Pops up the active plots.
Icon6 (Pause): Pause playback.
Icon7 (Stop): Stop data collection.
Icon8 (Plot): Plot existing data. Brings back plots after stopping.

System Information: The panel on the left hand side is a record of operations, including error messages.

Active Plot Tree: The panel on the right hand side defines the real time plots. Change signals and options by clicking on a branch.

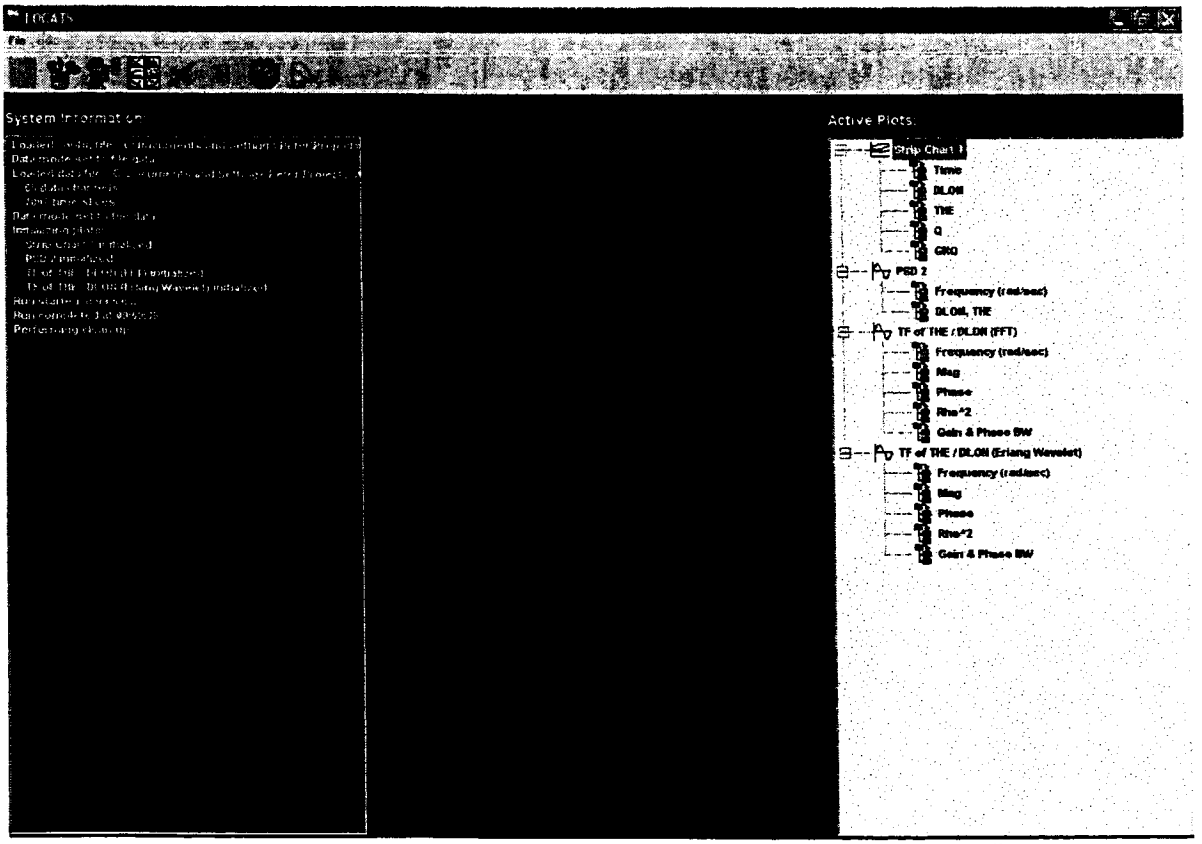


Figure 5. LOCATS Main Screen

4.2 Active Plots

The objective of this user's guide is for you to be able to create sets of active plots such as those shown in Figure 6. To play back stored data and duplicate these plots:

Start the program.
Use "File/Load Configuration" and select "pitch1.cfg" located in the "BAE Data" directory.
Click "R2D2" and open "Run69.dat" located in the same directory.
Click the Jet icon.
View the plots in real time.

Position

Drag the plots to the desired positions. The positions are remembered when the configuration is recalled. The plots always stay on top.

Timing

When running in real time the transform calculations and/or plotting routines can take longer than a cycle time, in which case data during the missed cycles is lost. See Timing in Section 6.10 for further information.

When playing back recorded data the program delays during each cycle if time remains so that the data is not played back faster than real time. If the transform and/or plotting routines take longer than a cycle the data is not skipped and the playback occurs slower than real time – hence in pseudo-real time.

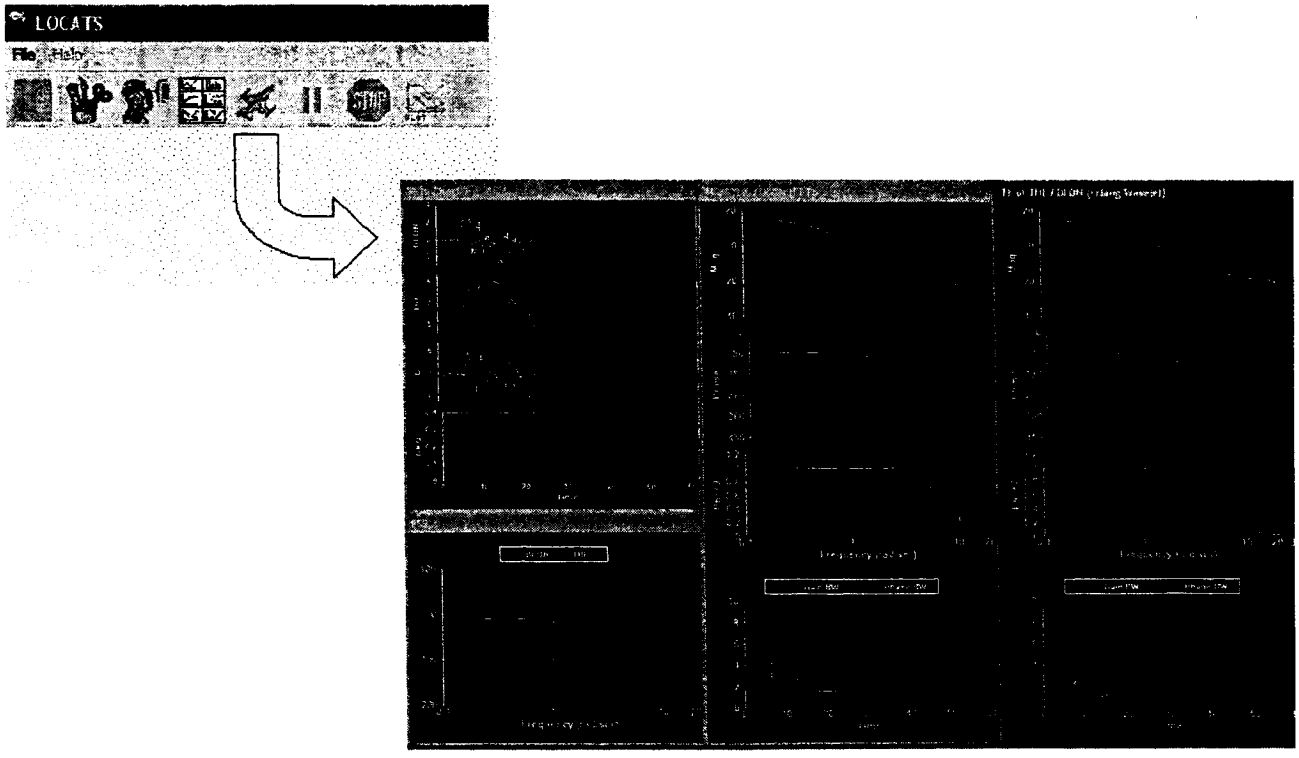


Figure 6. Active Plots

4.3 File Menu

Use the File menu to open the three types of files used by LOCATS:

.cfg = configuration files (information about the plots to be created)
.dat = data files (ASCII files with data for playback in pseudo-real time)
.sig = signal files (information about the real time signals)

The File menu and the open dialog boxes are shown in Figure 7.

Configuration Files

The configuration files can be saved using "Save configuration," or the current configuration cleared using "New configuration." The program starts by loading the previous configuration file (but not the previous data file or signal file). The user is prompted to save the current configuration when quitting if changes have been made. The program can "crash" on start-up if out-of-date versions of the configuration files are used, in which case the out-of-date versions should be deleted.

Data Files

Previously stored data is loaded using "Load data." The "R2D2" toolbar icon does the same thing. Real time data just collected is stored using "Save data."

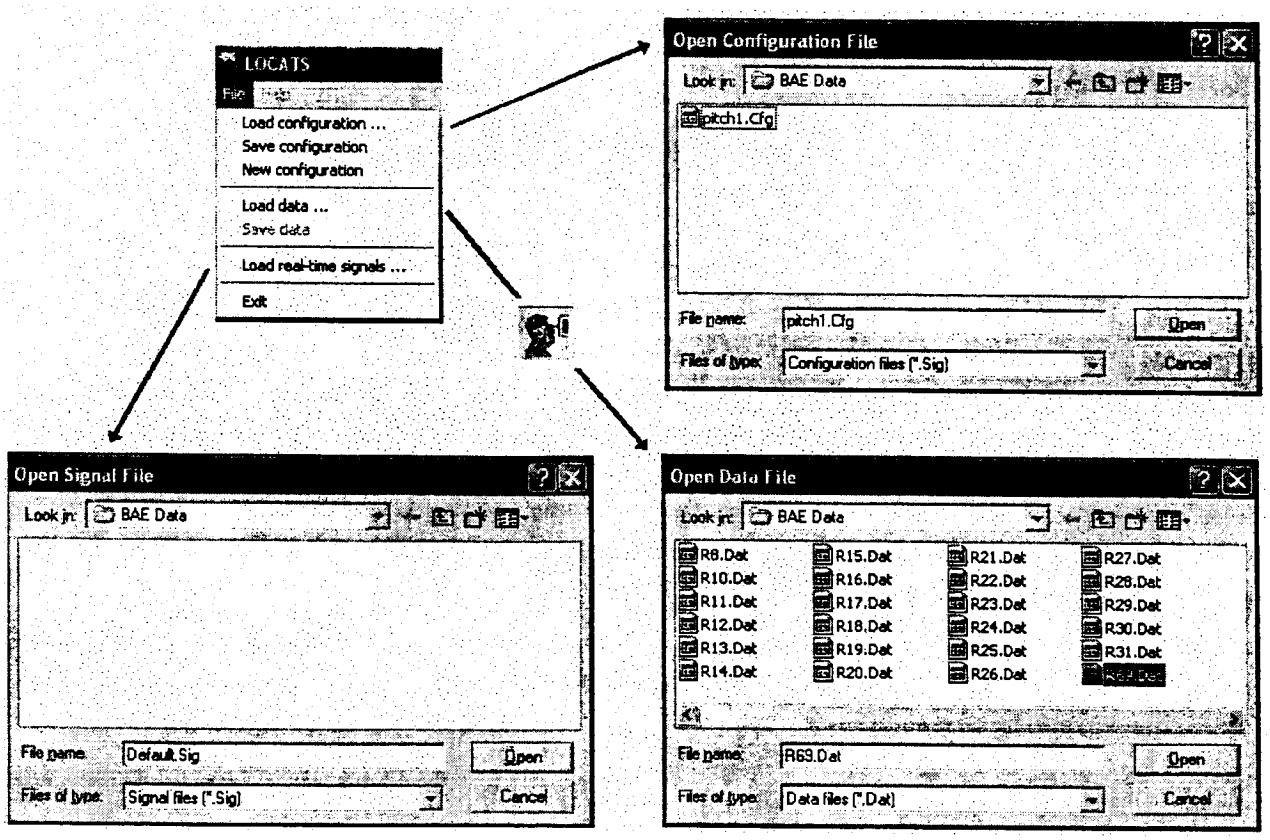


Figure 7. File Menu and Open Dialog Boxes

4.4 Add New Plots

Click the 6-Plot toolbar icon to add a new plot. The following plots are available:

Strip charts: Signals versus time. Up to 4 subplots (also known as channels). Each subplot can be multiple signals. Automatically scrolls.

XY Plot: Any two signals plotted versus each other. Does not scroll.

Scatter Plot: Like an XY plot, but with dots.

Estimated PSD: Computes and plots the power spectra of the selected signals. Up to 4 subplots, each with one or more signals.

Estimate TF: Computes and plots the transfer function of selected pairs of signals. Only one pair of signals, but up to 4 subplots, containing magnitude, phase, coherence, and so on.

The new plots appear in the active plot tree. Click on the branches to select signals and parameters.

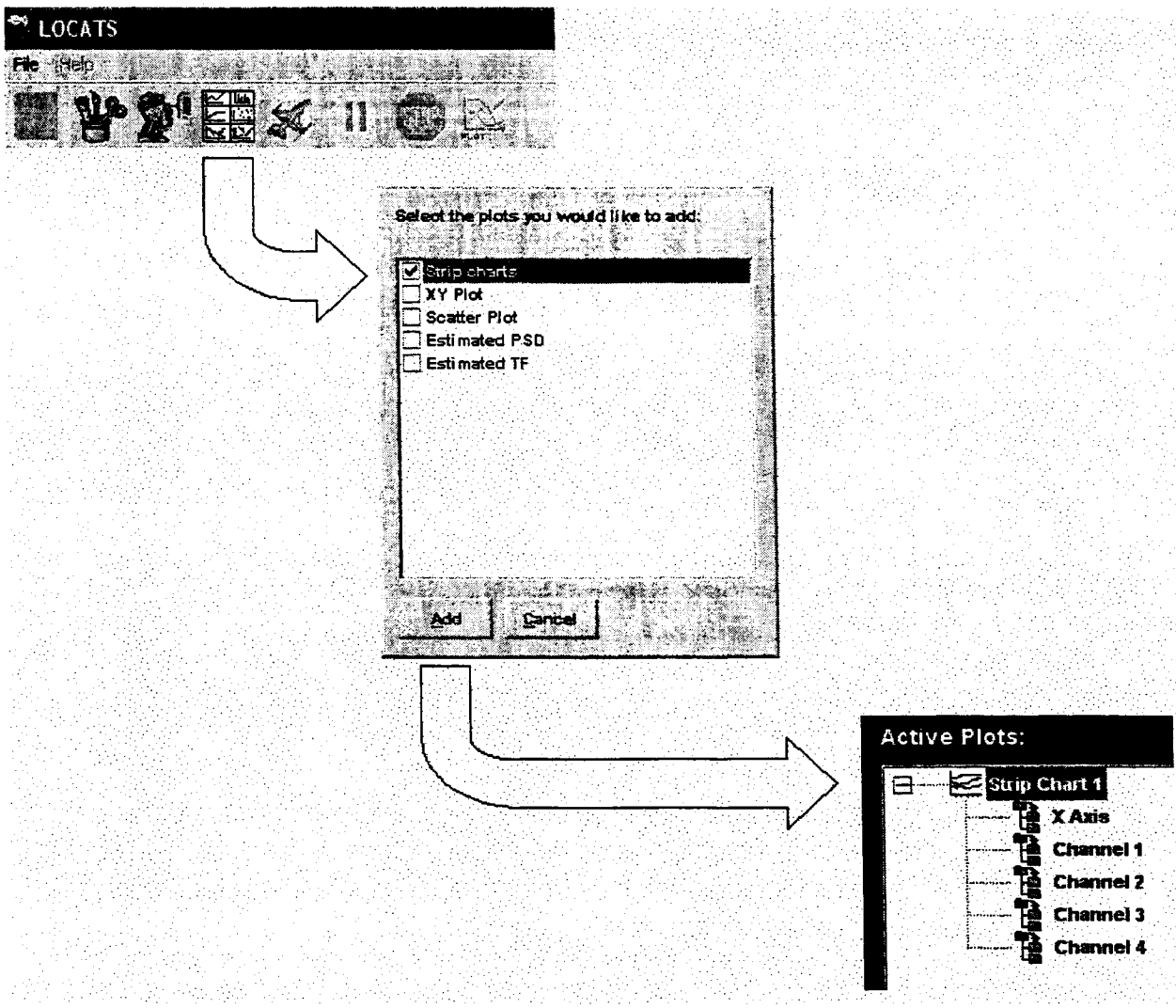


Figure 8. Add New Plots

4.5 Plot Information

Click on a plot title in the active plot tree and the Plot Information dialog box pops up.

Title: The plot title is automatically created when signals are selected. Change the title in the edit box and click Ok. (The new title may be lost if other options are selected, so immediately save the title).

Display: The plots on the active plot tree are not displayed unless the "Display Plot" checkbox is checked.

 # Channels: Each plot can have up to four channels. The active plot tree has a sub-branch for time or frequency and then a sub-branch for each channel.

Order: Use the up and down arrows to change the order of the plots on the active plot tree.

Delete: Delete the plot using the X button.

Transform Method: Selects parameters for the PSD and TF plots. See Section 6.

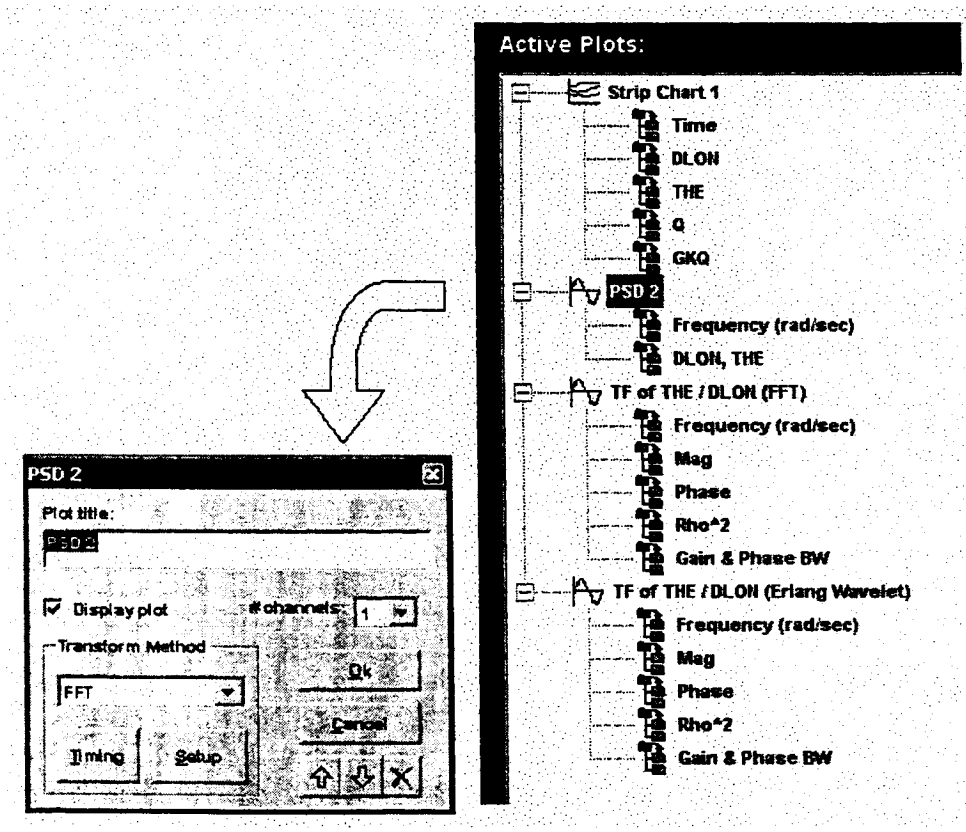


Figure 9. Plot Information

5. DISPLAYS

5.1 Strip Chart

A strip chart plots signals versus time. The plot scrolls when the max time is reached. There can be up to 4 channels on each strip chart. To plot more signals use multiple strip charts. The signal name appears on the left axis when there is one signal, or in a legend on top of the channel where there are multiple signals. To create a strip chart:

Click the 6-Plot toolbar icon to pop up the Add New Plot Dialog Box (see Figure 8). Check the Strip Chart Option and then click Okay.

Click on the title branch in the Active Plot Tree to pop up the Plot Information Dialog Box (see Figure 9). Change the title and/or number of channels.

Click the first sub-branch to pop up the Time Dialog Box. Select the time signal and axis limits (see Figure 11)

Click the remaining sub-branches to pop up the Channel Dialog Box (see Figure 12). Select the signal and axis limits.

Click the Jet toolbar icon to activate the plot. An example strip chart is shown in Figure 10.

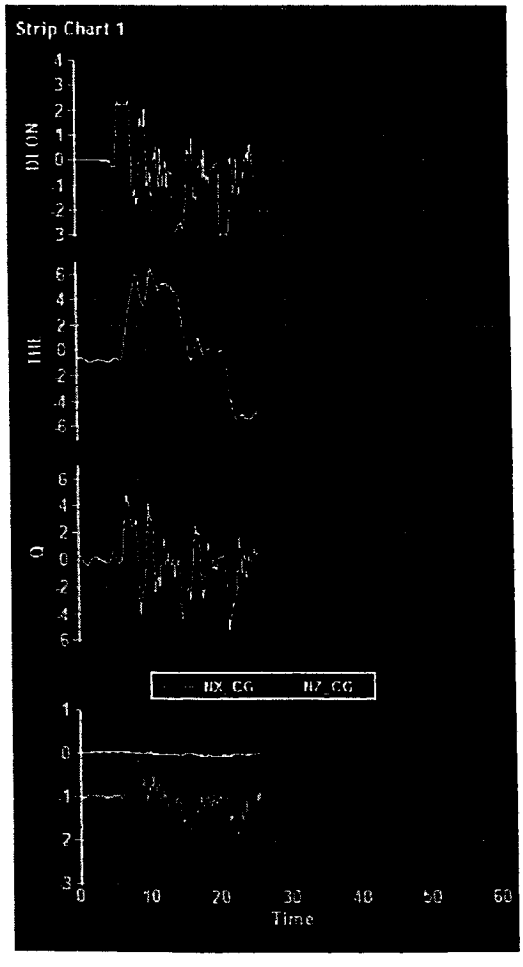


Figure 10. Strip Chart

5.1.1 Strip Chart Time

Click on the first sub-branch of the strip chart to pop up the time axis dialog box, as seen in Figure 11. The following controls are available:

Select signal: Select the time signal. Only one signal can be selected. Usually this is the first signal, but this is not assumed and you must make the choice.

Minimum Value, Maximum Value, Divisions: Set the max-min range of time displayed on the strip chart.

Percent Scroll: Percentage the time across scrolls when the max time is reached.

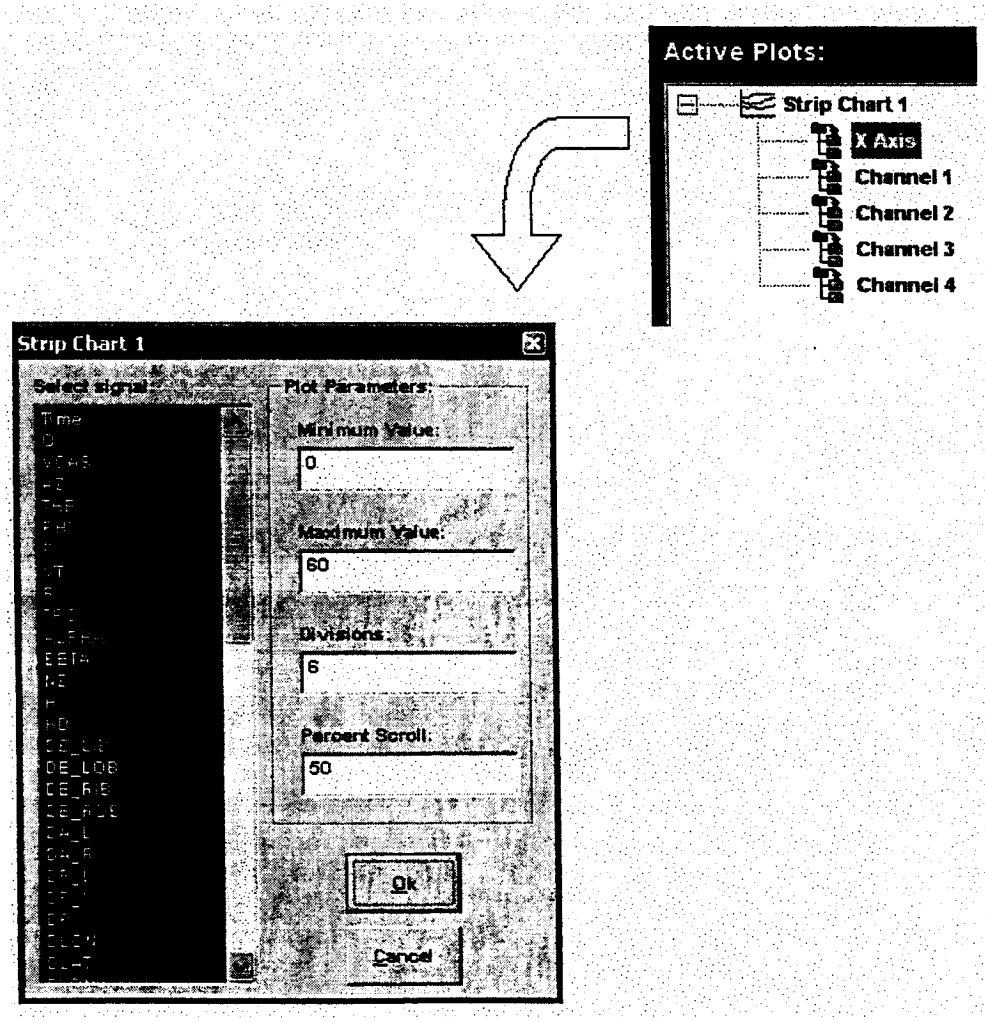


Figure 11. Strip Chart Time

5.1.2 Strip Chart Channel

The strip chart defaults to 4 channels. To change click on the title branch in active plot tree and change the number of channels in the plot information dialog box.

Click on the second and subsequent branches to select signals and vertical axis limits using the dialog box shown in Figure 12. The following controls are available:

Select Signals: Click on the signal in the list box to make a selection. Make multiple selections to compare signals on the same axes. Shift-click and/or Ctrl-click to make multiple selections.

AutoScale: Check for auto scaling. The vertical scale is increased as needed to fit the data, but does not automatically decrease.

Minimum Value, Maximum Value, Divisions: Defaults when signals are selected to fit the data.

Advanced Plot Editor: More options, not all implemented, so best not to use.

Thresholds: Sets thresholds on the plot.

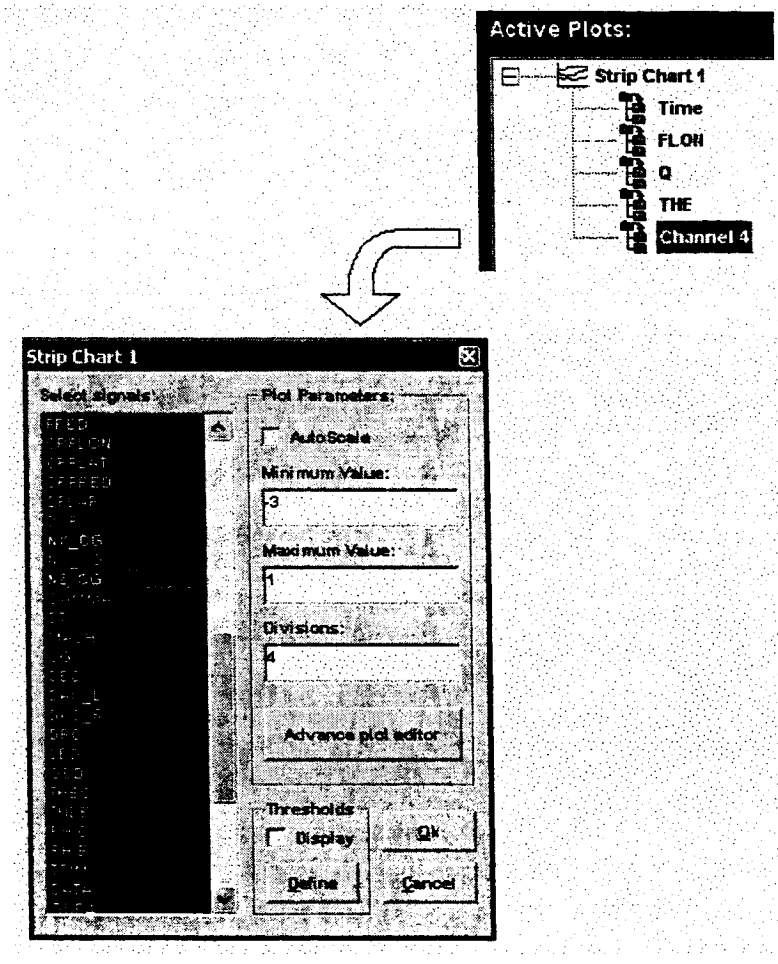


Figure 12. Strip Chart Channel

5.1.3 Thresholds

All of the plots can have thresholds, not just strip charts. The thresholds are plotted as solid red lines or line segments that mark important values on the subplots. Create the thresholds as shown in Figure 13: click on the channel, click on the "Define" button, and then enter the thresholds in the threshold dialog box. The thresholds are not displayed unless the "Display" checkbox is checked in the channel dialog box. An example of thresholds is shown in the figure marking acceleration limits. The following controls are on the threshold dialog box:

Threshold Index: There can be up to 5 thresholds on a subplot.

Horizontal Line: Enter a single value for thresholds that are horizontal lines. The lines will extend across the entire subplot.

Line Segments: Enter x,y points that define line segment thresholds.

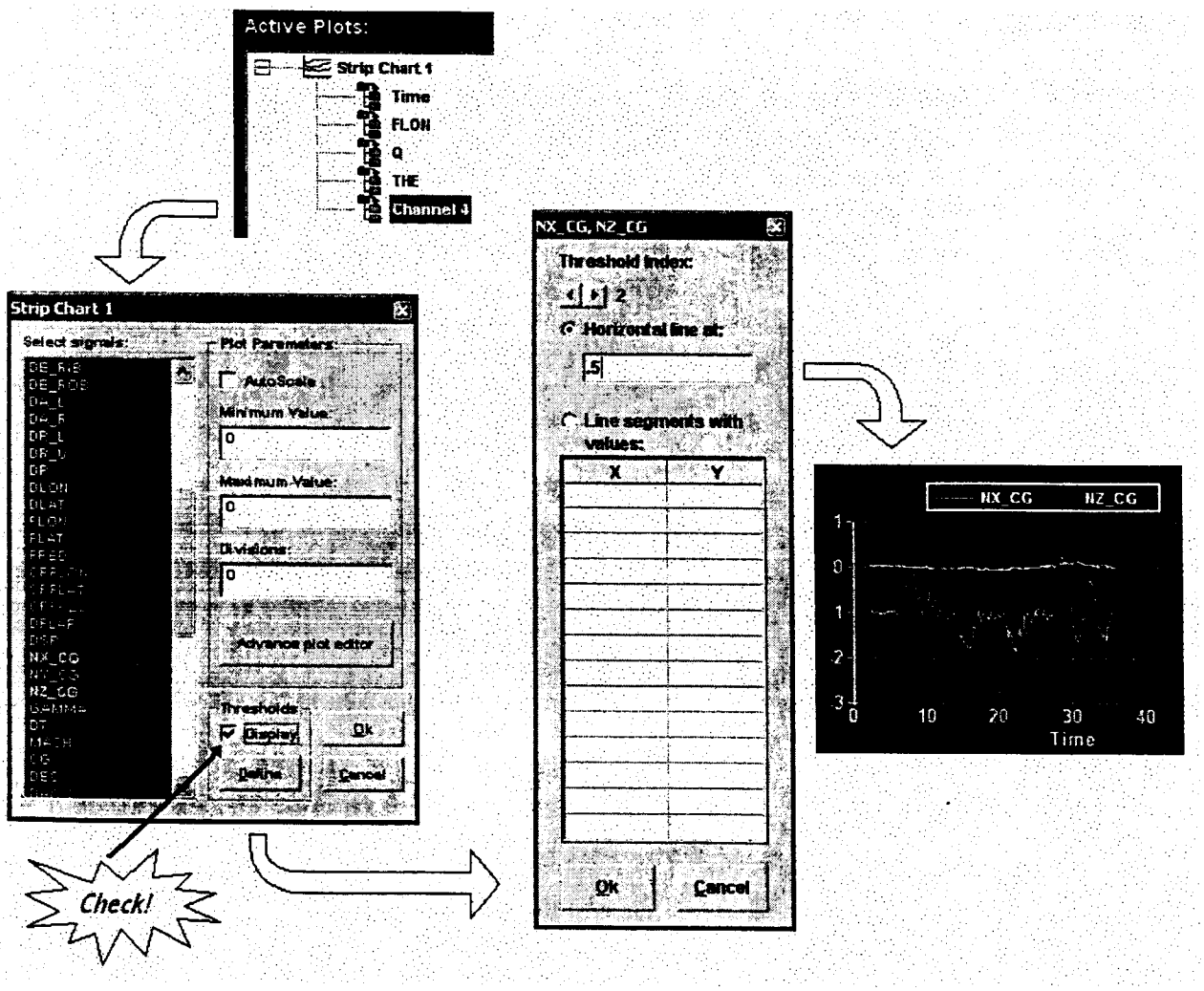


Figure 13. Thresholds

5.2 XY Plot

Any signal can be plotted versus any other signal. There can be up to 4 channels, with the same horizontal axis signal, but usually there is just 1 channel. XY Plots do not scroll. To create an XY Plot:

Click the 6-Plot toolbar icon to pop up the Add New Plot Dialog Box (see Figure 8). Check the XY Plot Option and then click Okay.

Click on the title branch in the Active Plot Tree to pop up the Plot Information Dialog Box (see Figure 9). Change the title and/or number of channels.

Click the sub-branches to pop up the Channel Dialog Box (see Figure 12). Select the signal and axis limits.

Click the Jet toolbar icon to activate the plot. An example XY-Plot is shown in Figure 14.

The plotting package (T-Chart) used in LOCATS has a "bug" and does not correctly plot lines that go right to left, so until an update to the plotting package is available the XY-Plot is not of much use. Try the Scatter Plot instead.

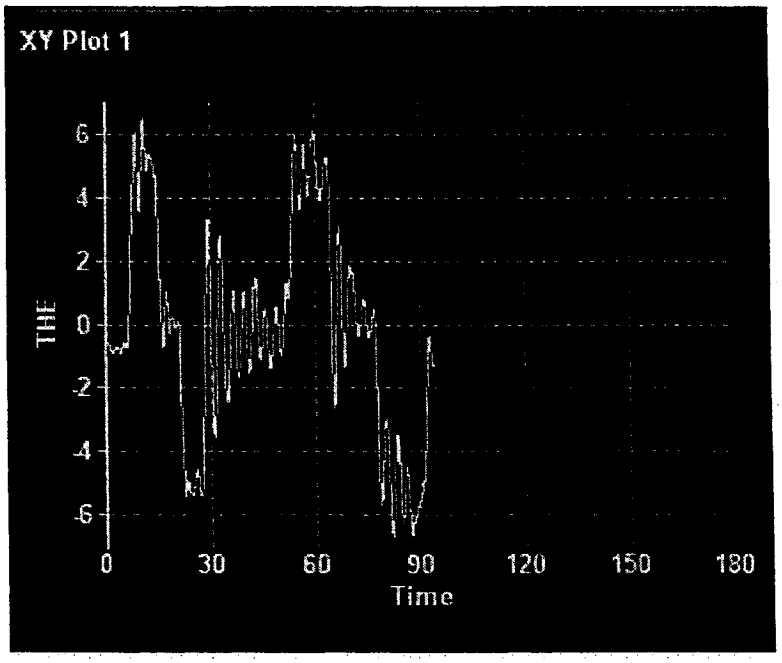


Figure 14. XY-Plot

5.3 Scatter Plot

Any signal can be plotted versus any other signal. There can be up to 4 channels, with the same horizontal axis signal, but usually there is just 1 channel. Scatter plots plot every 100th point as a dot. Scatter Plots do not scroll. To create a scatter plot:

Click the 6-Plot toolbar icon to pop up the Add New Plot Dialog Box (see Figure 8). Check the Scatter Plot Option and then click Okay.

Click on the title branch in the Active Plot Tree to pop up the Plot Information Dialog Box (see Figure 9). Change the title and/or number of channels.

Click the sub-branches to pop up the Channel Dialog Box (see Figure 12). Select the signal and axis limits.

Click the Jet toolbar icon to activate the plot. An example Scatter Plot is shown in Figure 15.

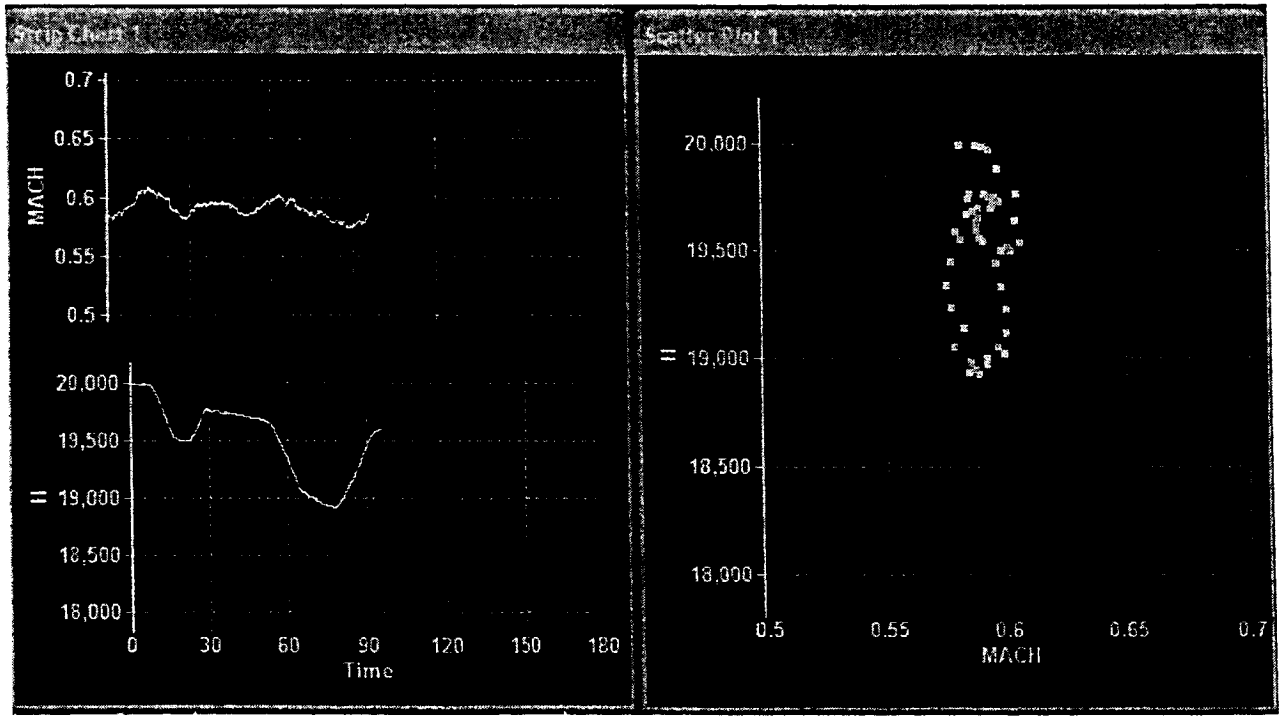


Figure 15. Scatter Plot

5.4 PSD Plot

The Power Spectral Density (PSD) plot is estimated from time series data using either Fourier or Wavelet transform methods. There can be up to 4 subplots, each containing one or more PSDs. Plotting more than one PSD on the same axis is a good way to compare magnitudes. The frequency axis is common to all of the subplots. To create a PSD plot:

Click the 6-Plot toolbar icon to pop up the Add New Plot Dialog Box (see Figure 8). Check the "Estimated PSD" Option and then click Okay.

Click on the title branch in the Active Plot Tree to pop up the Plot Information Dialog Box (see Figure 9). Change the title and/or number of channels.

Starting from the Plot Information Dialog box select the transform method and parameters (see Section 6).

Click the first sub-branch and select the frequency axis limits (see Figure 17)

Click the remaining sub-branches to pop up the Magnitude Dialog Box (see Figure 18). Select the signal and axis limits.

Click the Jet toolbar icon to activate the plot. An example PSD Plot is shown in Figure 16, shown next to the time series from which the PSDs were estimated.

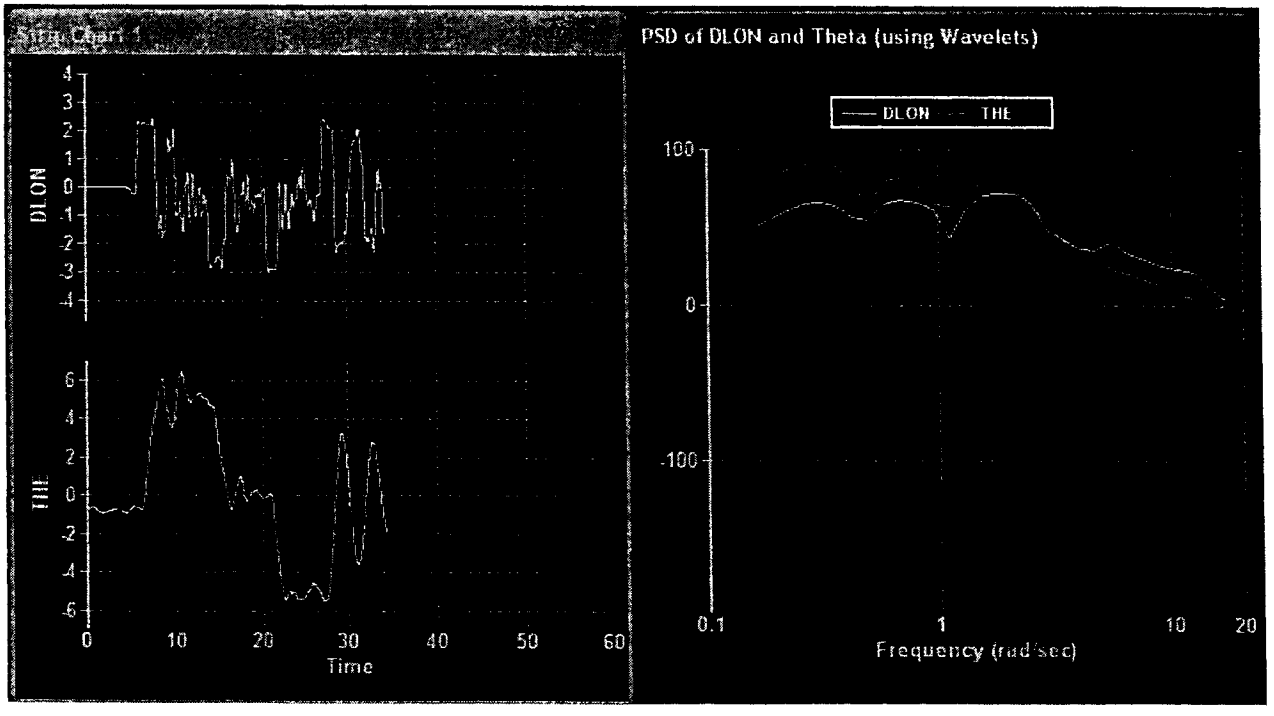


Figure 16. PSD Plot

5.4.1 PSD Frequency

Click on the first sub-branch of the PSD plot in the Active Plot Tree to pop up the Frequency Dialog Box as shown in Figure 17. The same frequency axis is used for each of the subplots. The controls on the dialog box are:

Autoscale: Doesn't work well for log axes and is disabled.

Minimum Value, Maximum Value, Divisions: Frequencies.

Nice Values: for min, max, and divisions.

Options: Frequencies are rad/sec or Hz, linear or log10. The min and max values are rad/sec or Hz depending on the selected option.

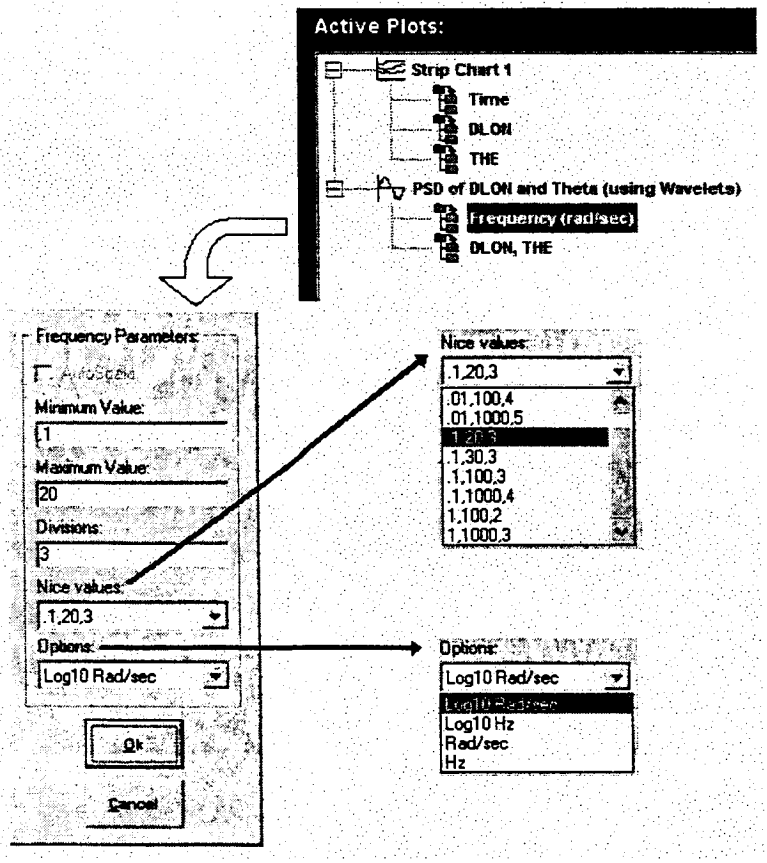


Figure 17. PSD Frequency

5.4.2 PSD Magnitude

Click on the second and higher sub-branches of the PSD plot to pop up the Magnitude Dialog Box as shown in Figure 18. The controls on the dialog box are:

Select Signals: Click on the signal in the list box to make a selection. Make multiple selections to compare PSDs on the same axes. Shift-click and/or Ctrl-click to make multiple selections.

AutoScale: Check for auto scaling. The vertical scale is increased as needed to fit the data, but does not automatically decrease.

Minimum Value, Maximum Value, Divisions: Magnitude, scaled according to the selected option.

Nice Values: for min, max, and divisions.

Options: Magnitude can be in dB, power dB, or absolute.

Thresholds: See Figure 13.

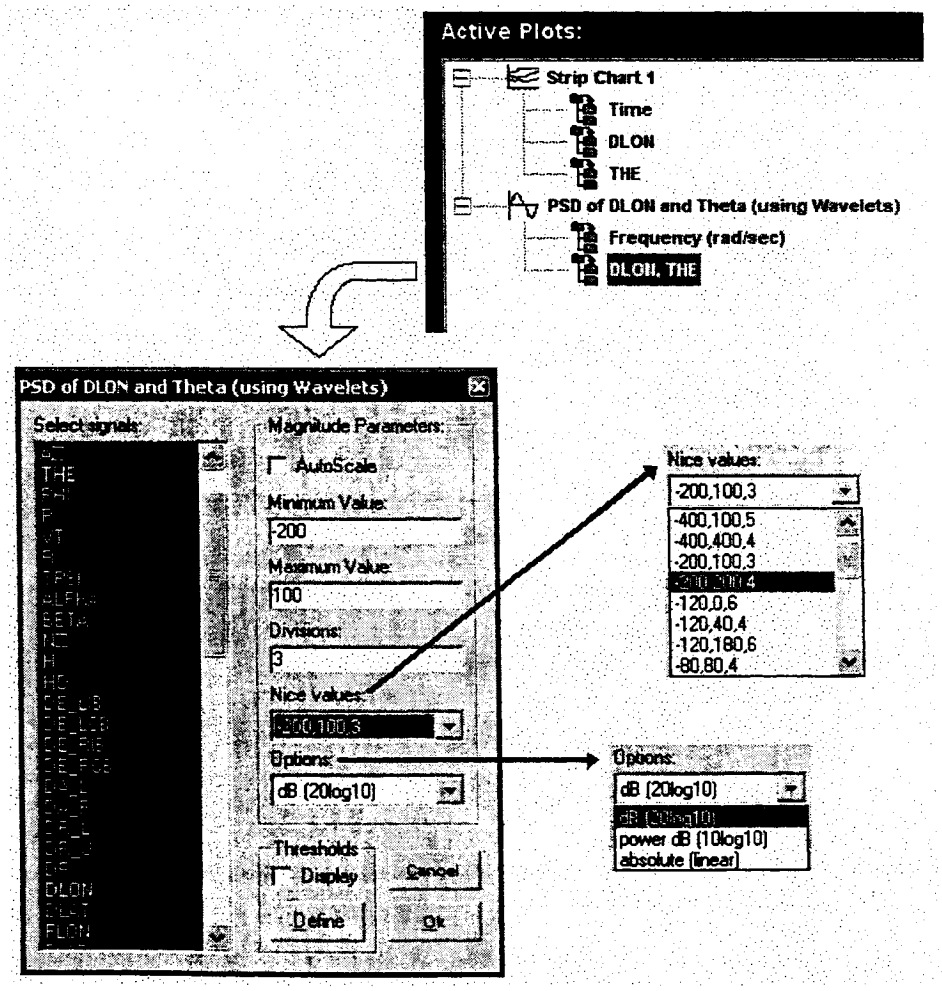


Figure 18. PSD Magnitude

5.5 TF Plot

The Transfer Function (TF) Plot is estimated from input and output time series data using either Fourier or Wavelet transform methods. There can be up to 4 subplots, each containing a different part of the transfer function or a parameter estimated from the frequency response. A transfer function can be linked to the previous TF Plot, so that more than 4 subplots can be created without duplicating the computations needed to estimate the transfer function. To create a TF plot:

Click the 6-Plot toolbar icon to pop up the Add New Plot Dialog Box (see Figure 8). Check the "Estimated TF" Option and then click Okay.

Click on the title branch in the Active Plot Tree to pop up the Plot Information Dialog Box (see Figure 9). Change the title and/or number of channels.

In the Plot Information Dialog and Transform Method Dialog Boxes select the transform method and parameters (see Section 6).

Click the first sub-branch and select the input and output signals and the frequency axis limits (see Figure 20)

Click the remaining sub-branches to pop up the Plot Type Dialog Box (see Figure 21 and Figure 22). Select the Plot Type (e.g., Magnitude, Phase, and so on) and then enter the vertical axis limits.

Click the Jet toolbar icon to activate the plot. An example TF Plot is shown in Figure 16, shown next to the time series used to estimate the transfer function.

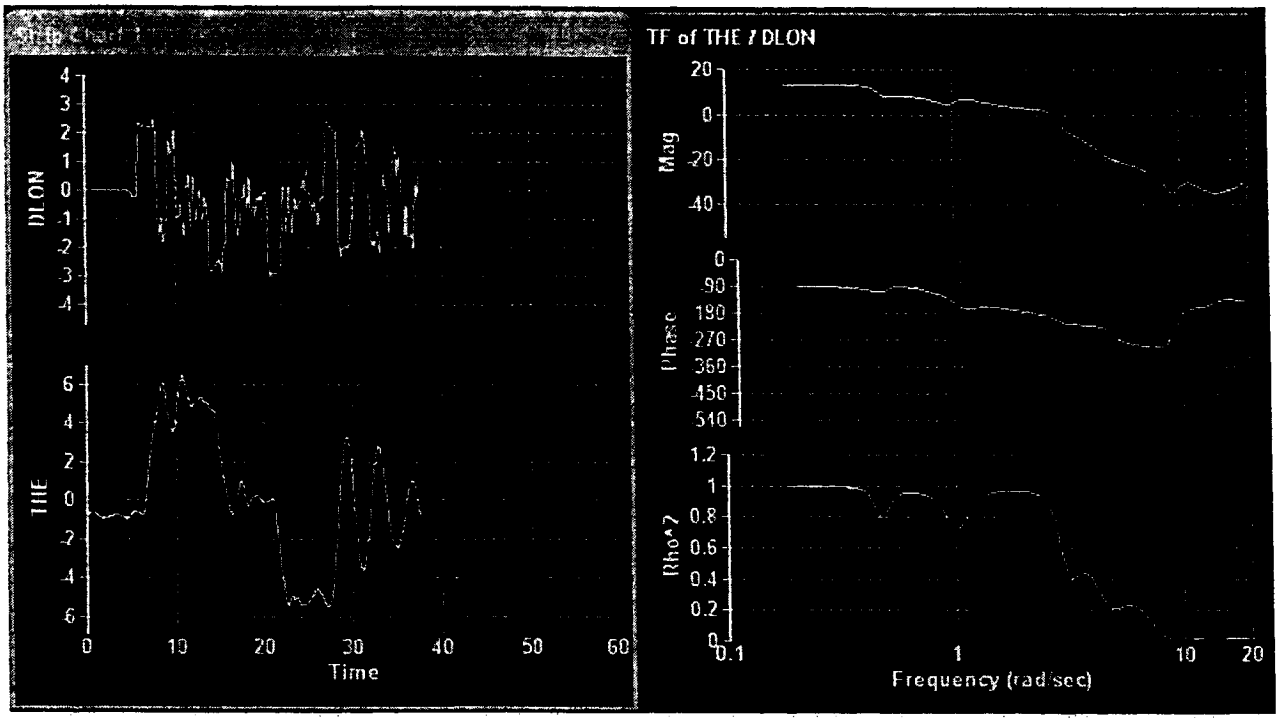


Figure 19. TF Plot

5.5.1 TF Signals and Frequency Axis

Click on the first sub-branch of the TF Plot in the Active Plot Tree and the TF Signals and Frequency Axis Dialog Box pops up as shown in Figure 20. The controls are:

Input Channel and Output Channel: Select one signal in each list box to define the transfer function.

AutoScale: Disabled.

Minimum Value, Maximum Value, Divisions: Frequencies.

Nice Values: for min, max, and divisions.

Options: Frequencies are rad/sec or Hz, linear or log10. The min and max values are rad/sec or Hz depending on the selected option.

Link to Previous TF: Enabled if the previous plot on the Active Plot Tree is a TF Plot, in which case up to four more subplots can use the same TF estimate.

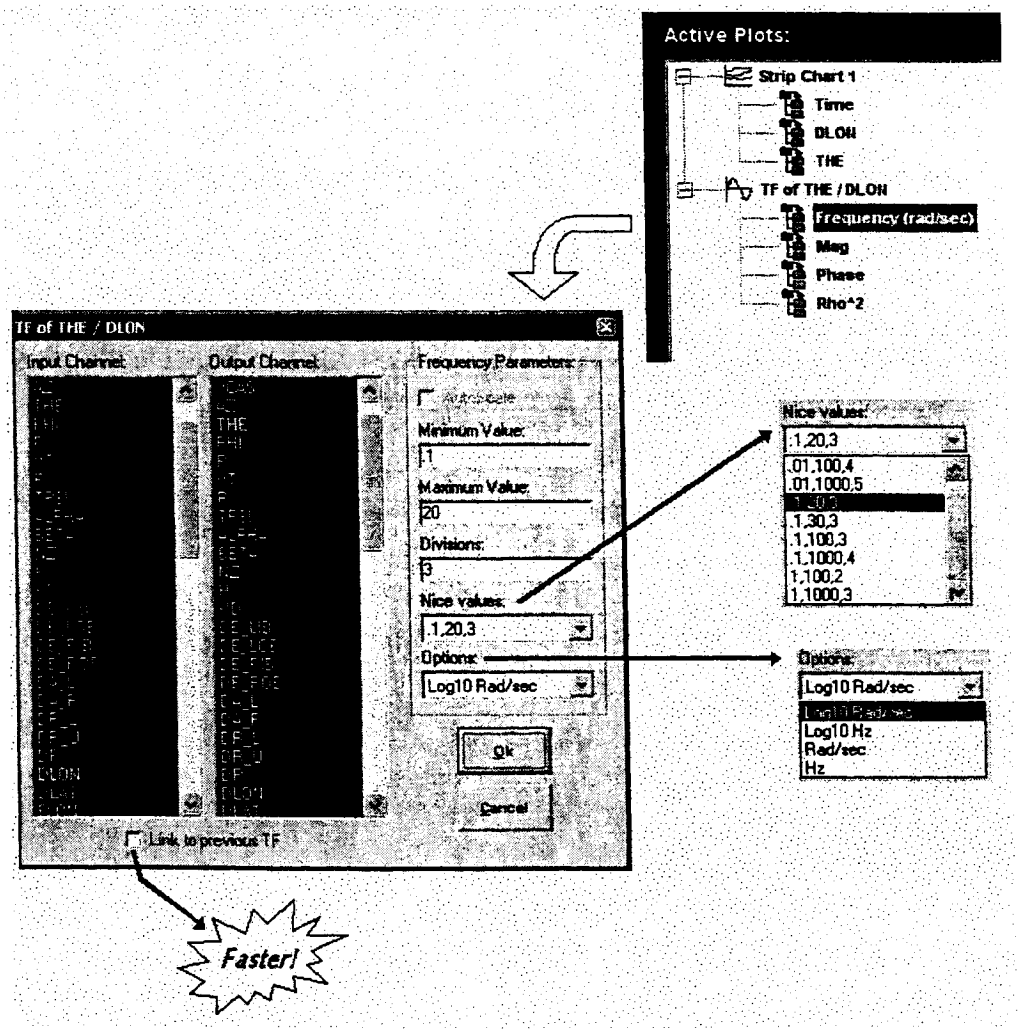


Figure 20. TF Signals and Frequency Axis

5.5.2 TF Magnitude, Phase, and Coherence

Click on the sub-branches below the frequency to pop up the Plot Type Dialog Box shown in Figure 21. The first five plot types are spectral and cross-spectra quantities:

Mag: $|G(j\omega)| = \hat{S}_{yx}(j\omega) / \hat{S}_{xx}(j\omega)$, where $\hat{S}_{yx}(j\omega)$ = smoothed cross spectra

Phase: $\text{Phase}[G(j\omega)] = \text{Phase}[\hat{S}_{yx}(j\omega)]$

Coherence: $\rho^2(j\omega) = |\hat{S}_{yx}(j\omega)|^2 / [\hat{S}_{xx}(j\omega)\hat{S}_{yy}(j\omega)]$

Input Power: $\hat{S}_{xx}(j\omega)$ = smoothed input spectra

Output Power: $\hat{S}_{yy}(j\omega)$ = smoothed output spectra

The example TF Plot in Figure 19 has subplots of magnitude, phase, and coherence. The remaining controls in the Plot Type Dialog Box are similar to those for the PSD Magnitude, except the Nice Values and Options change appropriately. Note that the phase can either be pinned to a selected 360 degree range or can be unwrapped. The plots can be either lines or dots, where dots are sometimes better when plotting transfer function metrics.

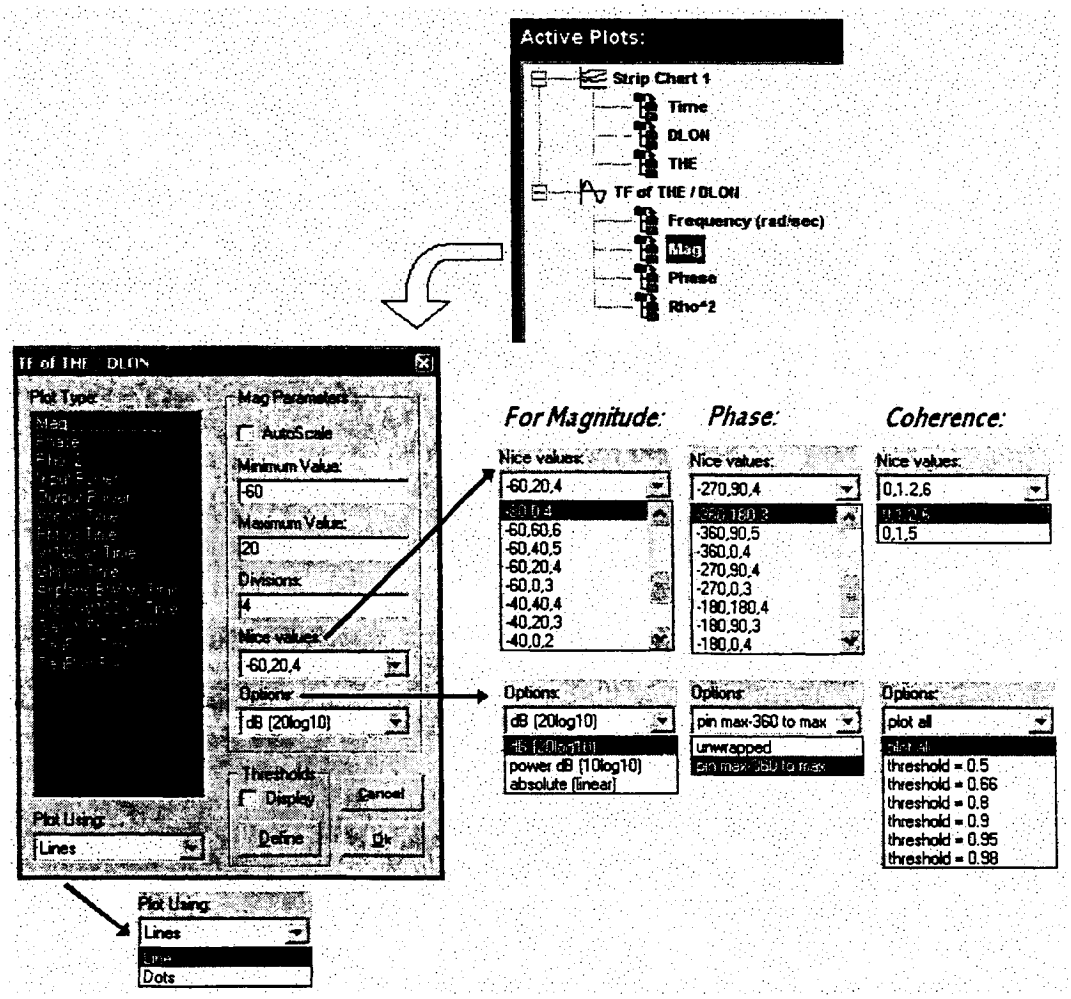


Figure 21. TF Magnitude, Phase, and Coherence

5.5.3 Transfer Function Metrics

The remaining Plot Types are parameters estimated from the frequency response. A value is obtained each time the transforms are computed, and then the metrics are plotted versus time. An example is shown in Figure 22 (Run 69 from Locats BAE Sim Data, the labels were inserted using Microsoft Paint).

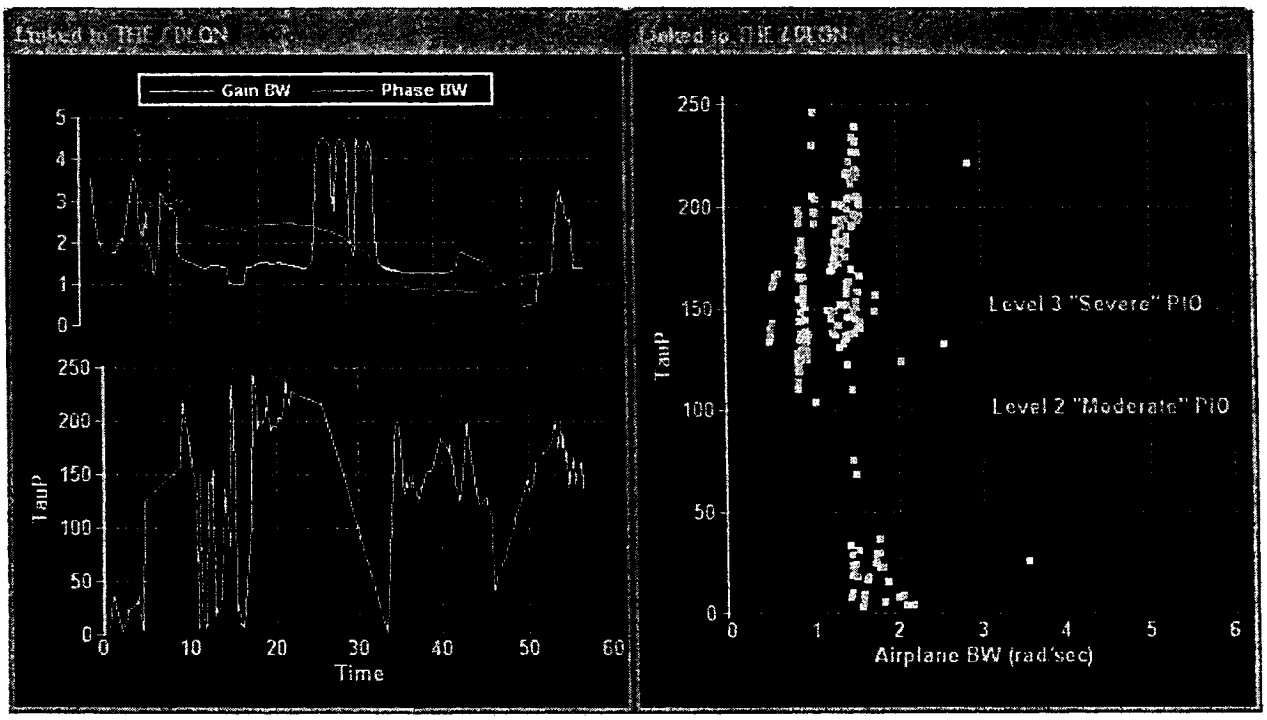


Figure 22. TF Metric Plots

The metrics that can be selected from the Plot Type Dialog Box are listed below:

Wc vs Time: Unit magnitude crossover frequency of $|G(j\omega)|$. Mainly for loop transfer functions (LTFs), in which case ω_c is the bandwidth and the frequency at which the phase margin is computed.

PM vs Time: The Phase Margin (PM) is 180 plus the phase of $G(j\omega)$ at $\omega = \omega_c$. Mainly for LTFs, in which case the phase margin is the extra phase at $\omega = \omega_c$ that will destabilize the system.

W180 vs Time: The frequency where the phase of $G(j\omega)$ crosses -180 degrees.

GM vs Time: The Gain Margin (GM) is minus the gain in dB of $G(j\omega)$ at $\omega = \omega_{180}$. Mainly for LTFs, in which case the gain margin is the gain increase that will destabilize the system.

Airplane BW vs Time: The airplane bandwidth is the minimum of the phase and gain bandwidths.

Wc and W180 vs Time: A combined plot of ω_c and ω_{180}.

Gain and Phase BW: The gain bandwidth is the lowest frequency where $|G(j\omega)|$ crosses 6 dB plus the magnitude at $\omega = \omega_{180}$. The phase bandwidth is the frequency where $G(j\omega)$ crosses -135 degrees. The gain and phase bandwidths are mainly of interest for manual control analysis when $G(j\omega)$ is an attitude controlled element. If the human operator acts like a constant gain, then the gain bandwidth is the $\omega = \omega_{180}$ frequency that results from a gain selection with a gain margin of 6 dB. The phase bandwidth is the $\omega = \omega_c$ frequency that results from a gain selection with a phase margin of 45 degrees.

TauP vs Time: Taup is a measure of the effective delay of the controlled element. It is defined as one half the phase slope of $G(j\omega)$ between ω_{180} and $2\omega_{180}$. Systems with large TauP (larger than about 150 msecs) require a higher workload on the human operator.

TauP vs BW: A cross plot of TauP and airplane bandwidth.

To set the parameters click on the appropriate branch in the Active Plot Tree. Select a metric Plot Type and then the dialog box changes as shown in Figure 23. The time signal must be selected, and the axis limits are needed for both axes. The "Nice Values" and "Options" change depending on the other selections.

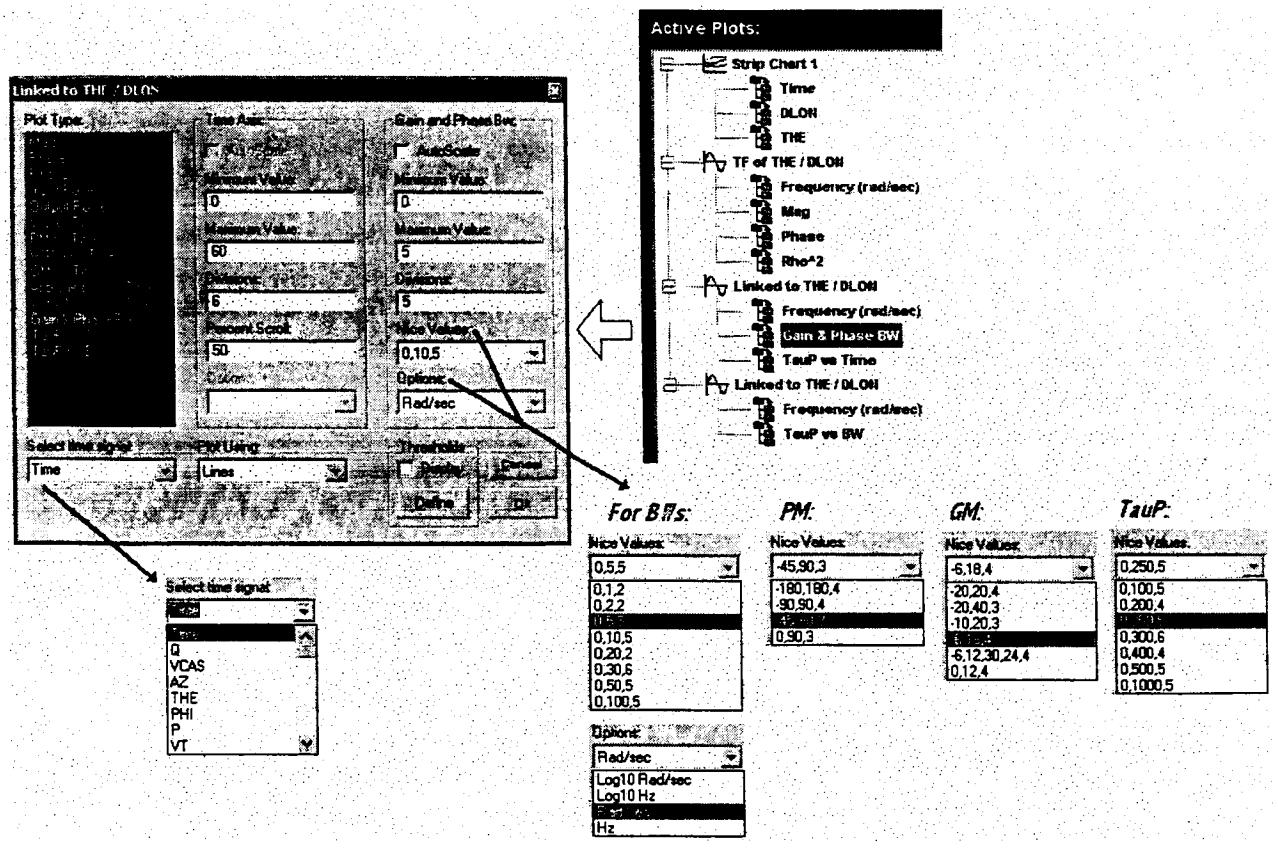


Figure 23. TF Metrics

6. TRANSFORM METHODS

6.1 Transform Method Dialog Box

Fourier and wavelet transforms of time series are used to create Power Spectral Density (PSD) and Transfer Function (TF) plots. LOCATS was developed to demonstrate the use of wavelet transforms for this purpose, and so selecting transform methods and their parameters is at the heart of this program. Use the following dialog boxes and controls to do so, as shown in Figure 24.

Click on the plot title in the Active Plot Tree to pop up the Plot Information Dialog Box.

Use the transform method list box to select the transform method.

Click on "Timing" push button to pop up the Transform Method Tabbed Dialog Box with the "Timing" tab selected.

Click on the "Setup" push button to pop up the same Tabbed Dialog Box with either the "FFT" or "CWT" tab selected.

Transform Method Tabbed Dialog Box has the following tabs:

Timing: Set parameters for the sliding window, persistence, and time domain smoothing.

Fmax: Set the max frequency at which the transform is computed.

FFT: Set parameters for Fourier transform methods.

CWT: Set parameters for Continuous Wavelet Transforms.

LLS: Set parameters for LLS parameter estimation.

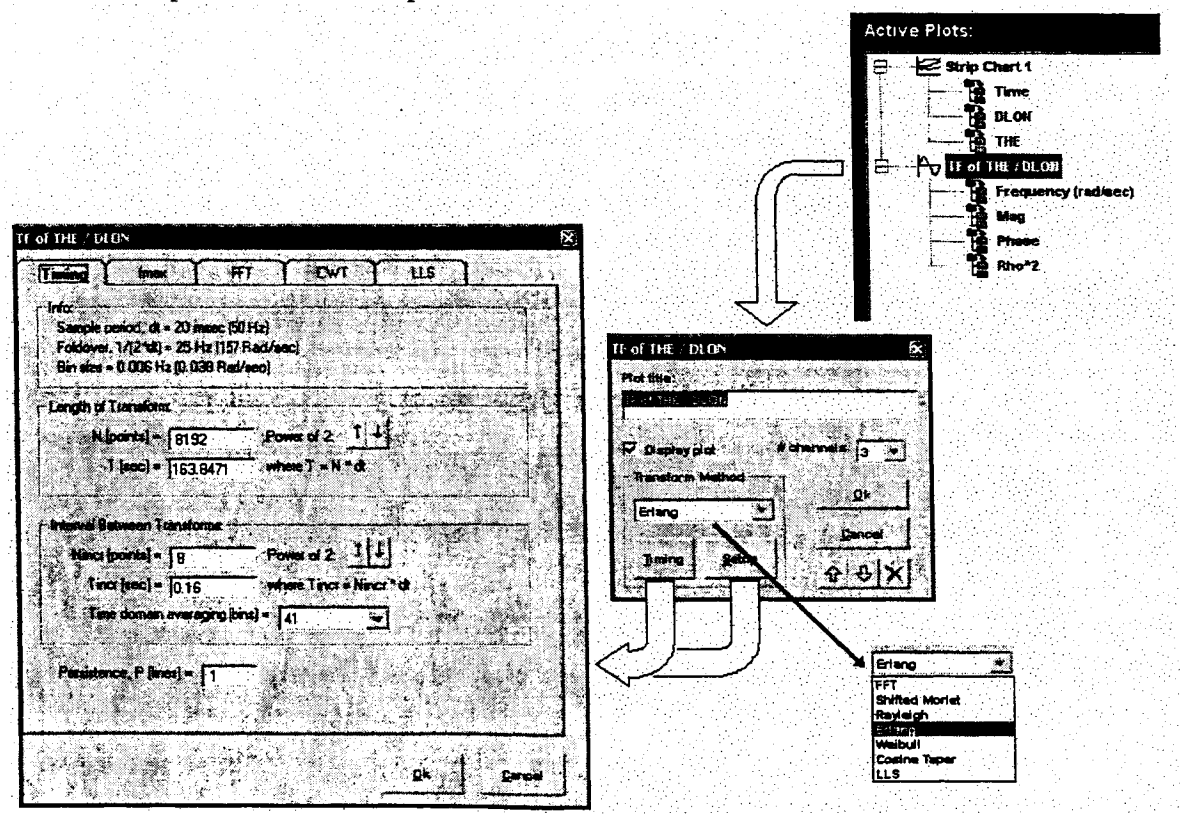


Figure 24. Transform Method Dialog Box

6.2 Timing Tab

The Timing Tab on the Transform Method Dialog Box is shown in Figure 24.

Information:

The sample period is determined by subtracting the first two times in a data file and then stored in the configuration file. If known, the following information is displayed:

dt [sec] = sample period
1/(2*dt) [Hz] = foldover frequency
1/T [Hz] = bin size

Length of Transform:

Use the edit boxes to enter:

T [sec] = length of transform
N [points] = number of points in transform, where T = N*dt

Enter one of these and the other is automatically updated. Use the up and down arrows to change N by powers of 2. The parameters T and N can also be changed in the FFT and CWT tabs, in which case T and N are changed in the other tabs to be consistent.

The number of points does not have to be a power of 2, the FFT algorithm breaks N down into prime factors to speed up the calculation, and the CWT computation time scales with N.

Interval Between Transforms:

Use the edit boxes to enter:

Tincr [sec] = time between transform calculations
Nincr [points] = incremental number of points in transform, where Tincr = Nincr*dt

Enter one of these and the other is automatically updated. Use the up and down arrows to change Nincr by powers of two. The incremental number of points does not have to be a power of 2. If many transforms are being computed, try setting Nincr to values like 5 and 7 so that the calculations rarely occur during the same time step.

At the start of the run there are less than full amount of N points available. The transforms are nevertheless computed every Nincr points, with zero fill until the full number of points is available. This is why the spectra appear to grow out of the bottom of the plot at the start of the run. The same does not happen for transfer functions because the input and output spectra are reduced by the same amount, though the variance of the estimate starts large and then reduces.

Sliding Window:

The length and increment define the sliding window shown in Figure 25. If Tincr = 0.1 [sec], for example, then the transform is computed and plotted 1/Tincr = 10 times a second.

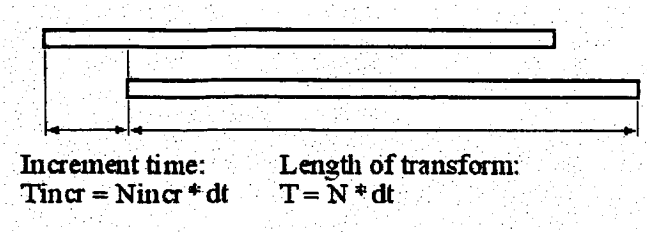


Figure 25. Sliding Window

Time Domain Averaging:

Sets the number of adjacent-in-time spectra that are averaged. The time domain averaging set here is used for both FFT and wavelet transforms. The number of adjacent-in-frequency spectra is determined from controls in the FFT and CWT tabs. See Section 6.5 for examples and further information about averaging.

Persistence:

P = persistence is the number of lines plotted for the PSD and TF plots. The line for the current time is always plotted, together with $P-1$ previous lines that fade with receding time. Persistence makes it visually more apparent when changes occur, and makes the variance of the estimate more apparent. An example with and without persistence is shown in Figure 26. The variance of the estimate is significantly larger above 2 rad/sec.

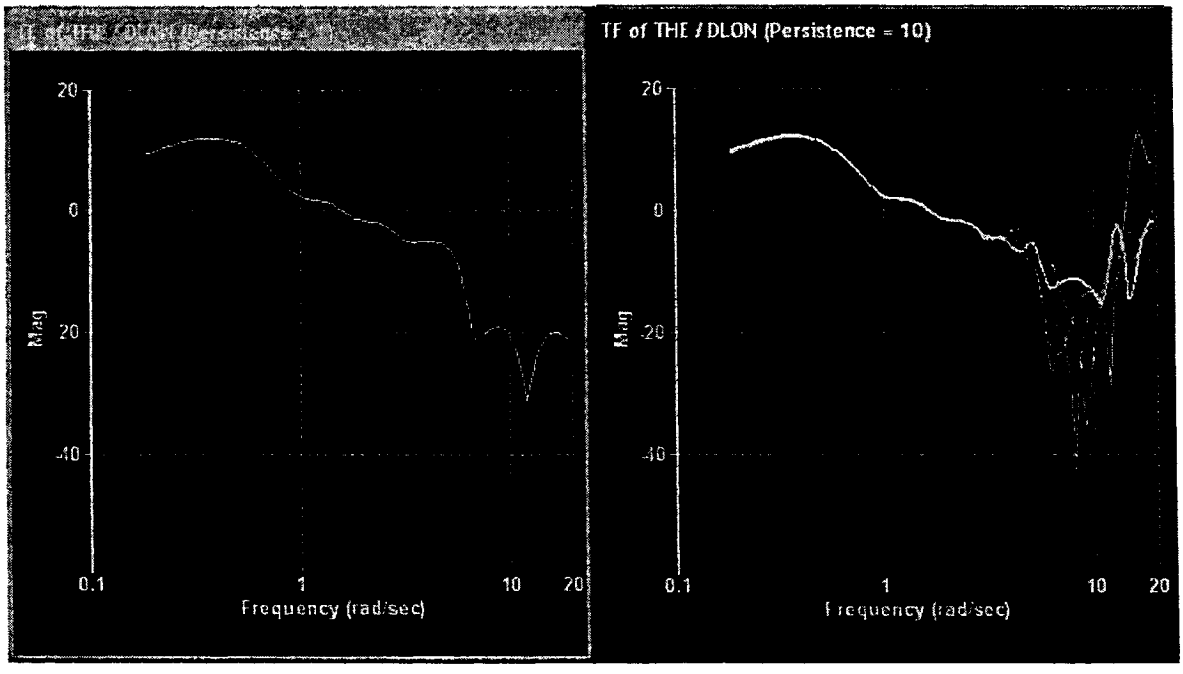


Figure 26. Persistence

6.3 Fmax Tab

The maximum frequency at which the spectra is estimated is set using the fmax tab shown in Figure 27.

The foldover frequency is shown in Hz and rad/sec.

The checkbox selects the foldover frequency as the maximum frequency.

Or, the max frequency can be entered either in Hz or rad/sec, with the other being automatically updated.

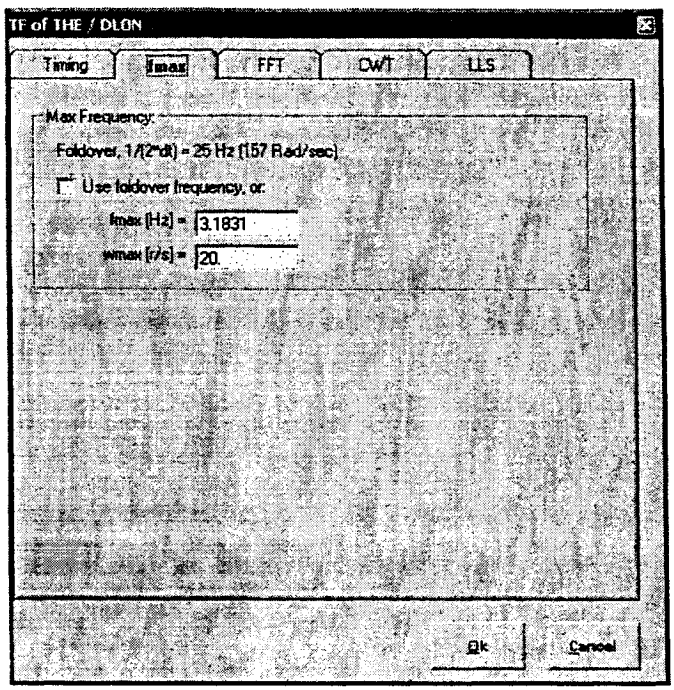


Figure 27. Fmax Tab

6.4 FFT Tab

Parameters that define the Fast Fourier Transfer (FFT) are set using the FFT tab shown in Figure 28.

Binning (Frequency Averaging):

Adjacent frequency bins are averaged if the "Use log binning" box is checked. All of the FFT bins from 1/T to fmax are otherwise used. The number of adjacent bins averaged increases with frequency (if Fratio is greater than one) so that the average frequencies are approximately equally spaced on a log scale. The frequency of the averaged set of bins is the average frequency. The log binning parameters are:

Nmin: The minimum number of bins averaged, primarily sets the number of low frequency bins that are averaged.

Fratio: The approximate ratio of adjacent averaged frequencies, so that the averaged frequencies are approximately log-spaced. The same number of bins is averaged across the frequency range if Fratio = 1.00. A Fratio larger than one is approximately one plus the damping ratio, so systems known to have lightly damped modes should use Fratio equal to 1.01 to 1.05. Systems known to be well damped should use Fratio equal to 1.10 or 1.20.

See Section 6.5 for an example of log binning and for further information about smoothing.

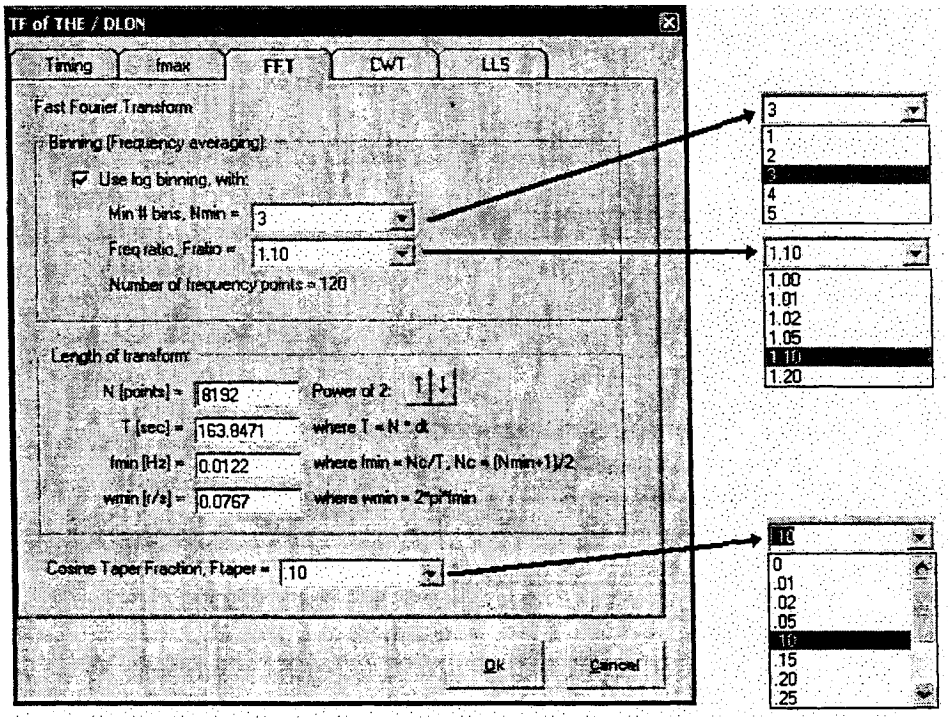


Figure 28. FFT Tab

Length of Transform:

The number of points, length in seconds, minimum frequency in Hz and rad/sec all depend on each other. Set one, whichever is the most relevant for you, and the others are automatically updated. The lengths in the "Timing" and "CWT" tabs are also changed. Use the arrows to change the number of points by powers of 2, though it is not necessary that N actually be a power of 2.

Cosine Taper Fraction:

The time series is tapered at each end by a half cycle of a cosine with the selection fraction. The maximum fraction of 0.50 is equivalent to the cosine window. The taper helps to reduce the end effects caused by discontinuities at the beginning and end of the time series. A value of 0.05 or 0.10 is recommended.

6.5 CWT Tab

Parameters that define the Continuous Wavelet Transform (CWT) are set using the CWT tab shown in Figure 29.

Continuous Wavelet Transform:

Select the type of transform, the length in cycles, and the cosine taper fraction. The wavelet transform expands or contracts in time so that the length in cycles remains the same. Click the "Show" button to see a plot of the mother wavelet and its Fourier transform. The different types of transforms are defined and compared in Section 6.6.

Length of transform (at fmin):

The length of the transform in seconds depends on the frequency. The maximum length in seconds or points is set, and then the minimum frequency in Hz or rad/sec depends of this length and the number of cycles. (For the FFT the minimum frequency is 1/T Hz, here it is Nc/T Hz). Set any one of the four parameters and the others are automatically updated. The max length in the "Timing" and "FFT" tabs is also updated.

Number of frequencies:

The wavelet transform is computed at Nw frequencies log-spaced from fmin to fmax. Set either Nw or Fratio and the other quantity is automatically updated.

"Frequency Domain Averaging", sets the number of spectra bins averaged across frequency. See Section 6.7 for examples and further information on smoothing.

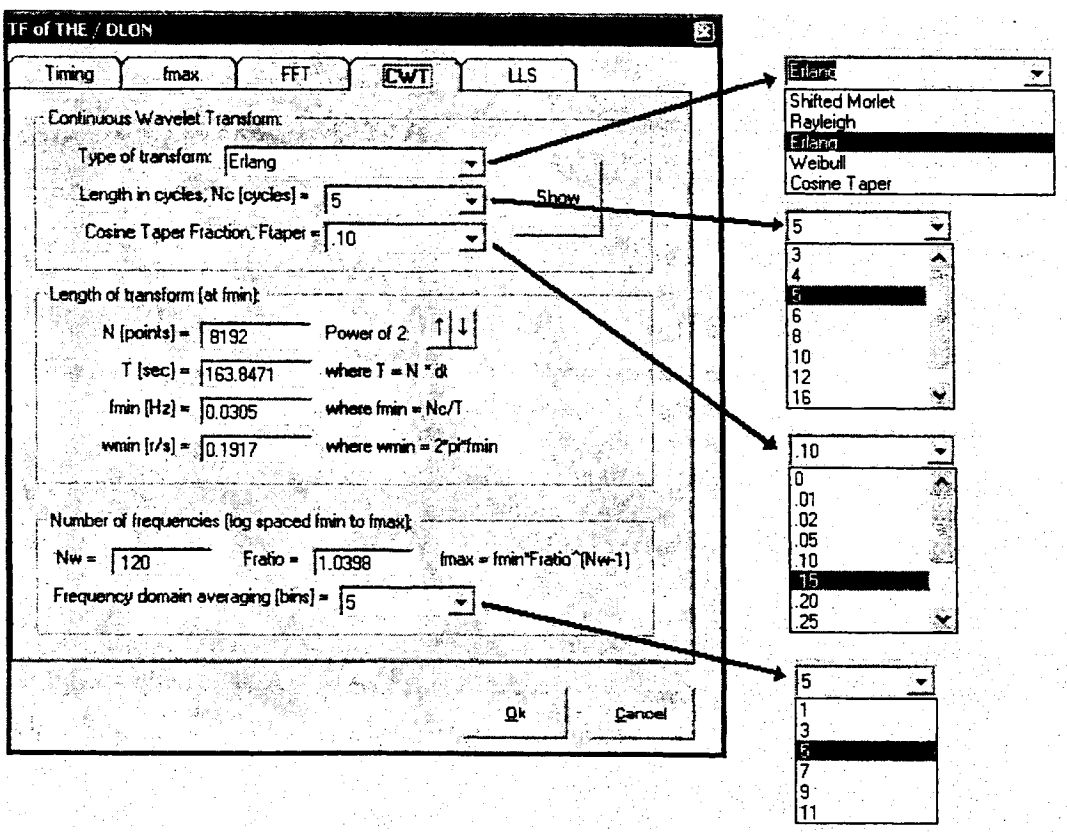


Figure 29. CTW Tab

6.6 Mother Wavelets

Mother Wavelet Dialog Box

The "Show" push button on the "CWT" tab pops up the "Continuous Wavelet Transform" (a.k.a. "Mother Wavelet") Dialog Box shown in Figure 30. The mother wavelet and its Fourier transform are displayed.

The parameters that define the wavelet are the same as in the "CWT" tab. Each time a change is made the new wavelet is displayed.

Check the "Show previous magnitudes" box to compare envelopes in time and frequency. To reset the comparison cancel back to the tabbed dialog box and re-press the "Show" button.

Check the "Show imaginary part" box the show both the real and imaginary parts of the wavelet.

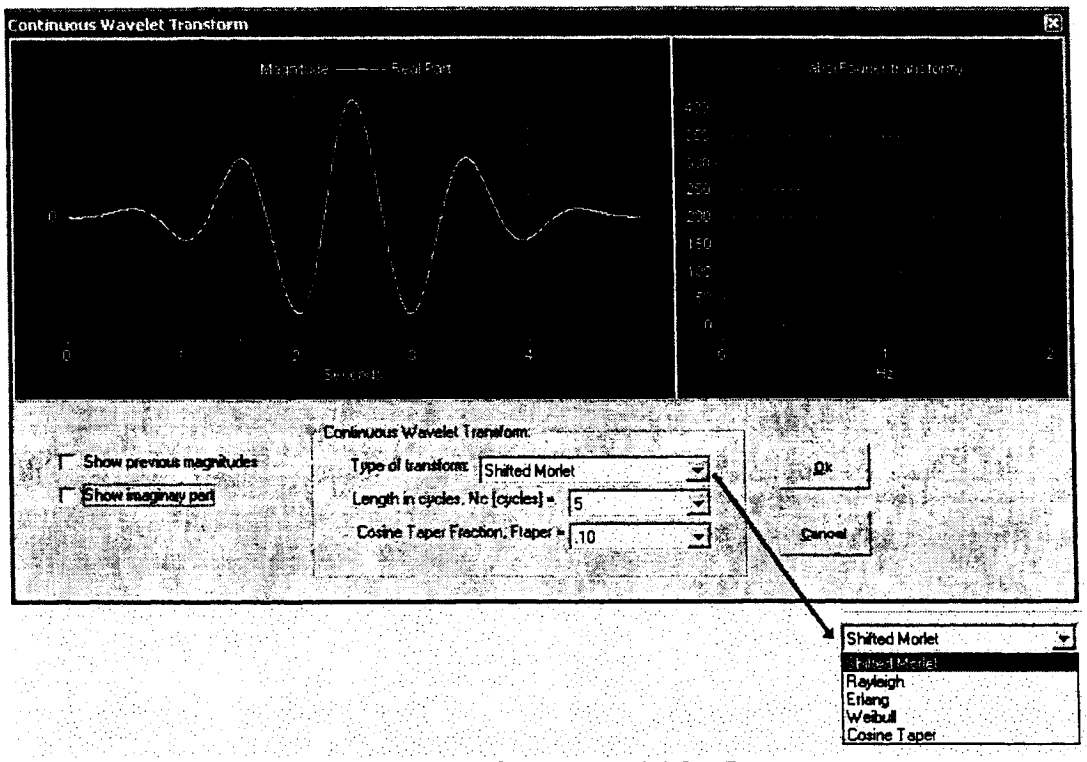


Figure 30. Mother Wavelet Dialog Box

Mother Wavelets:

All of the wavelets are a complex exponential with an envelope:

$$\psi(f,t)=\frac{1}{\sqrt{f}}g\left(f\times(t-u)\right)e^{j2\pi f(t-u)}$$

where f Hz is the frequency of the wavelet, u is the shift in time, and $g(t)$ is the envelope. The mother wavelet has a frequency of $f=1$ Hz and a shift of $u=0$ seconds.

Morlet: The envelope for the Morlet wavelet is:

$$\text{Morlet: } g(t)=\frac{1}{\sqrt{\pi\beta}}e^{-\left(t-t_{shift}\right)^2/\beta}$$

This is also called a "shifted Morlet" wavelet, because it is shifted to the right to make it causal. (Wavelets are causal if they are non-zero only for positive time). The envelope is a Gausian bell-shaped curve with a standard deviation of $\sigma = \sqrt{\beta/2}$. The shift is $t_{shift} = 3\sigma$ and the enveloped is truncated at $t_{shift} \pm 3\sigma$. This range is set equal to the length of N_c cycles, so that $6\sigma = N_c / f$. The Morlet wavelets with 5 and 10 cycles are compared in Figure 31a. The wavelet with 10 cycles is half the amplitude in both time and frequency.

Rayleigh: The envelope for the Rayleigh wavelet is:

$$\text{Rayleigh: } g(t) = \frac{2}{\beta} t e^{-t^2/\beta}$$

The name comes from the Rayleigh probability density. The Rayleigh and Morlet wavelets are compared in Figure 31b. The frequency response of the Rayleigh has a slightly narrower bandwidth, but this difference is not significant. It is the faster rise of the Rayleigh in the time domain that makes this wavelet significantly different from the Morlet. The faster rise results in faster tracking of changes in the spectra estimated using this wavelet. The maximum value in the Rayleigh envelope occurs at time $t_{peak} = \sqrt{\beta/2} \times (1/f)$. The envelope is truncated at time zero and four times the peak, and this range is set equal to the length of N_c cycles, so that $4t_{peak} = N_c / f$.

Erlang: The envelope for the Erlang wavelet is:

$$\text{Erlang: } g(t) = \frac{a^n t^{n-1}}{(n-1)!} e^{-at}$$

The name comes from the Erlang probability density. The Erlang and Morlet wavelets with 5 cycles are compared in Figure 31c. The order parameter $n = 3$ is fixed and not made available to the user for change. With this order the Erlang and Rayleigh are almost the same. The Erlang has a slightly smaller bandwidth than the Rayleigh in the frequency range, and hence is slightly broader in the time domain. Our experience with the Erlang wavelet has yielded slightly better results at estimating parameters and that is why we have been using the Erlang wavelet and not the Rayleigh for most of the examples in this user's guide. The maximum value of the Erlang envelope occurs at time $t_{peak} = (n-1)/a$. The envelope is truncated at time zero and four times the peak, and this range is set equal to length of N_c cycles, so that $4t_{peak} = N_c / f$.

Weibull: The envelope for the Weibull wavelet is:

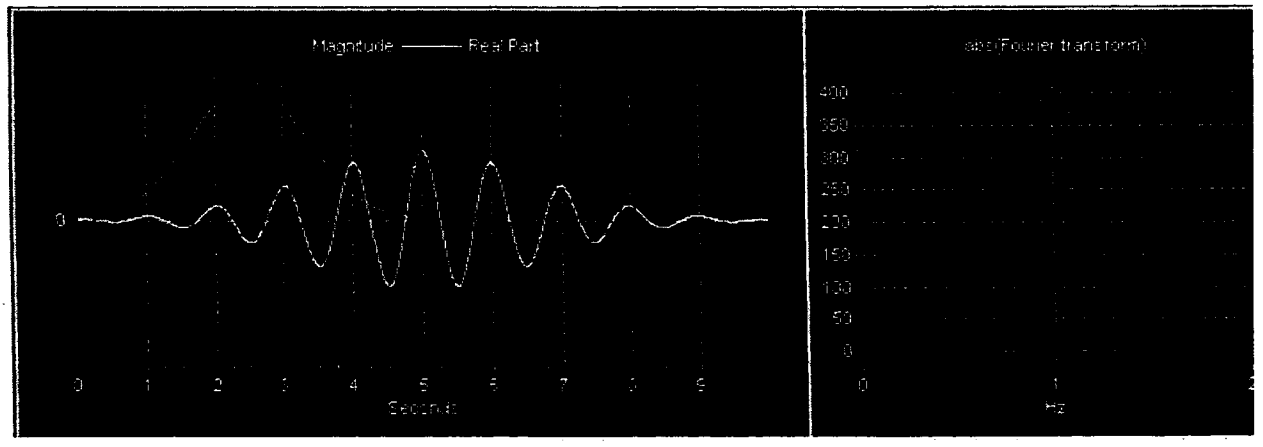
$$\text{Weibull: } g(t) = abt^{b-1} e^{-at^b}$$

The name comes from the Weibull probability density. The Weibull and Morlet wavelet with 5 cycles are compared in Figure 31d. The order parameter $b = 3$ is fixed and not made available to the user for change. With this order it turns out that the Weibull with $2N_c$ cycles is a close match to the Morlet with N_c cycles. The maximum value of the Weibull occurs at $t_{peak} = [(b-1)/(ab)]^{1/b}$. The envelope is truncated at time zero and four times the peak, and this range is set equal to length of N_c cycles, so that $4t_{peak} = N_c / f$. (In retrospect truncating the envelope at two times the peak would be better, but the choice was made to use four times to be consistent with the other skewed envelopes).

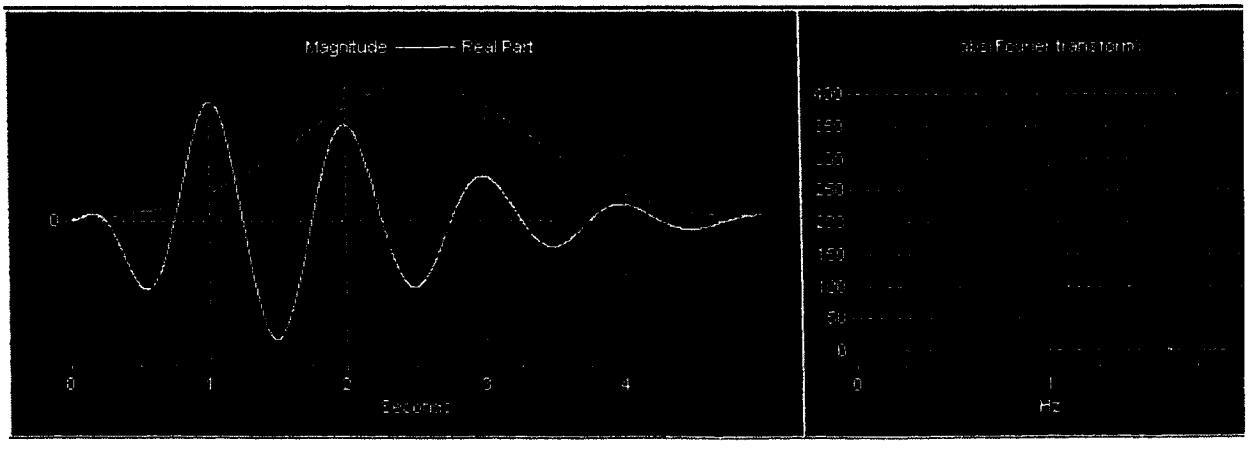
Cosine Taper: The envelope for the cosine taper wavelet is:

$$\text{Cosine taper: } g(t)=\begin{cases}\dfrac{\alpha}{2}\left(1+\cos\dfrac{\pi t}{t_{taper}}\right) & 0\le t<t_{taper}\\ \alpha & t_{taper}\le t<T-t_{taper}\\ \dfrac{\alpha}{2}\left(1+\cos\dfrac{\pi(t-T)}{t_{taper}}\right) & T-t_{taper}\le t<T\end{cases}$$

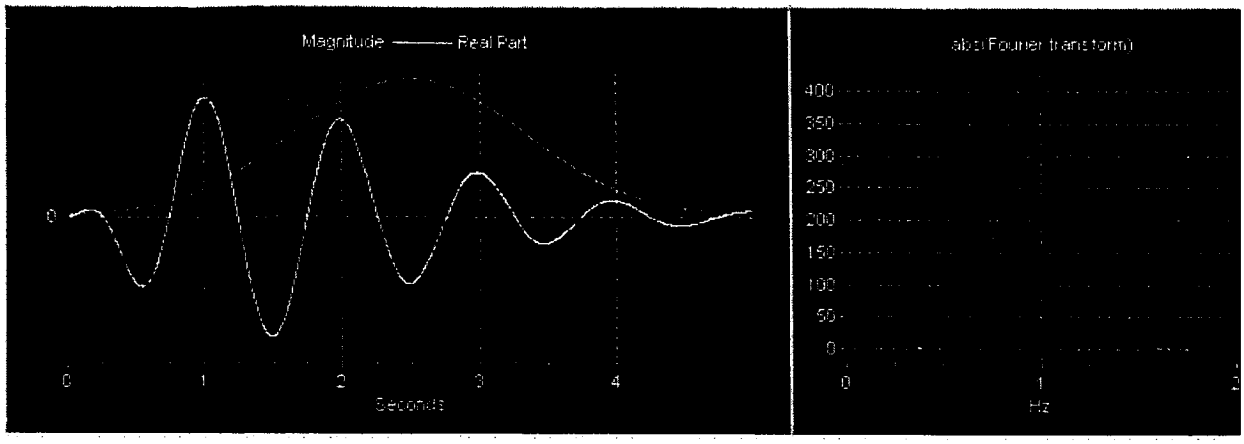
where the taper time is a given fraction of the total time T, and $\alpha=1/[(1-t_{taper})T]$ so that the area under the envelope is unity. The cosine taper and Morlet wavelets with 5 cycles are compared in Figure 31e. All of the other Fourier transforms had smooth tails, but here the Fourier transform has scallops, reminiscent of FFT windows, which is actually what the cosine tapered wavelet is designed to be. The difference, however, as with all wavelets is that the wavelet scales with frequency. For a given frequency the cosine taper has better frequency resolution than the Morlet wavelet at the expense of more leakage in the side lobes. The cosine taper wavelet is an interesting idea, but we have not found in our analysis to date better estimation results. There may be cases we haven't yet discovered where this wavelet is the one to use, so it remains in the stockpile.



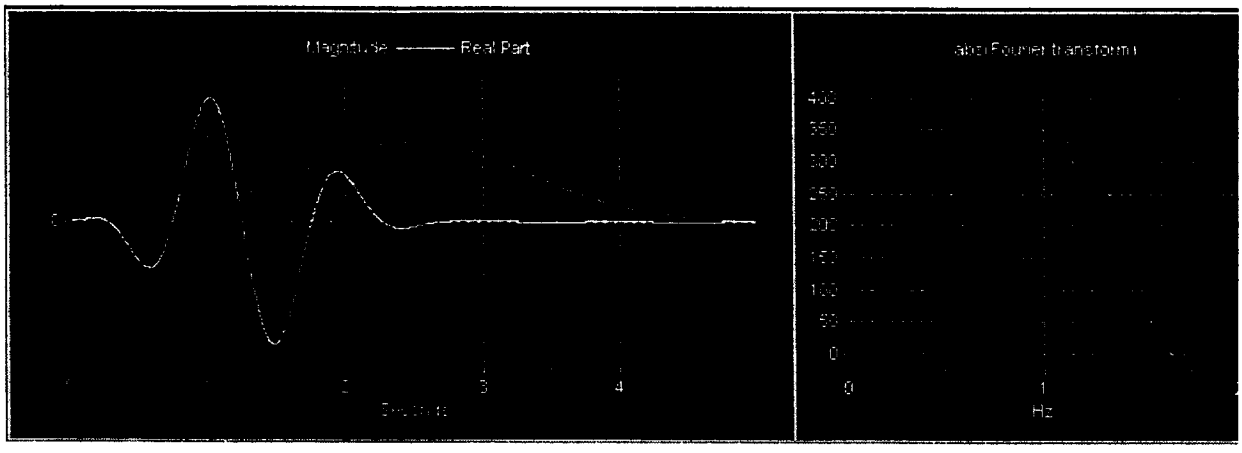
a) Morlet Wavelet with 10 cycles compared with 5 cycles



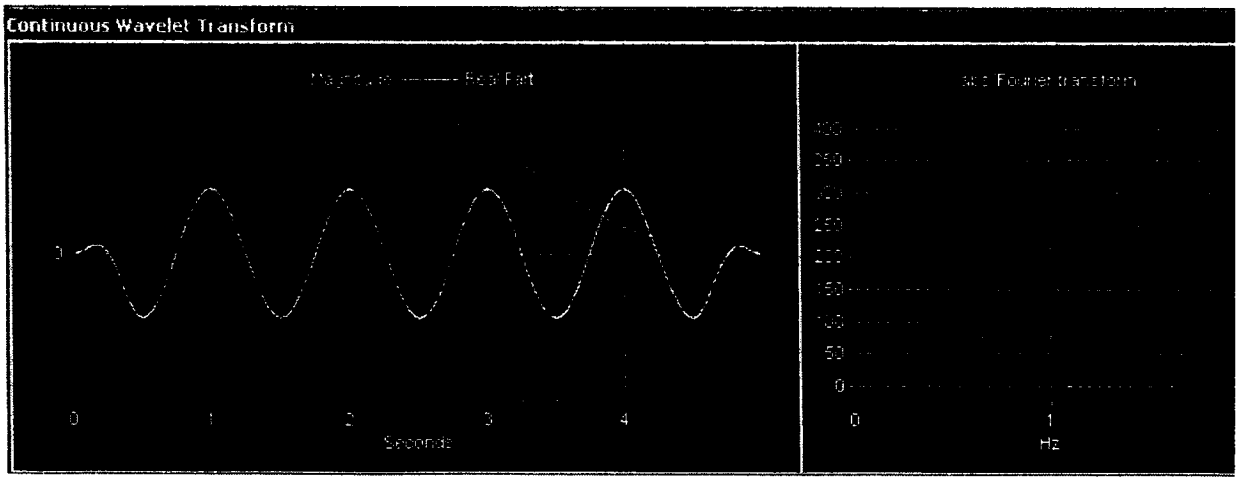
b) Rayleigh Wavelet with 5 cycles compared with Morlet 5 cycle



c) Erlang Wavelet with 5 cycles compared with Morlet 5 cycle



d) Weibull Wavelet with 5 cycles compared with Morlet 5 cycle



e) Cosine Taper Wavelet with 5 cycles and 10% taper compared with Morlet 5 cycle

Figure 31. Mother Wavelet Comparison

6.7 LLS Tab

Parameters for the Linear Least Square (LLS) estimation problem are set using the LLS tab shown in Figure 32. Coefficients of a z-transform are estimated and then used to compute the frequency response.

Numerator and Denominator Order:

Enter the numerator and denominator orders, respectively m and n of the estimated z-transform. The implementation is general and high orders can be entered, but generally n should be in the range 2 to 6 and the relative order $n-m$ should be zero or one.

Length of Sliding Window:

Set the length $T = N \times dt$ [sec] of the sliding window. Enter either the length in seconds or points, and the other parameter is automatically updated. The length in the "Timing" tab is also changed. The number of points can be changed by the arrows to be powers of 2, though there is nothing special here about these powers.

Frequencies:

The z-transform is estimated and then the frequency response computed and returned for plotting. The z-transform coefficients are not returned or displayed The frequency response is computed using Nw log-spaced points from fmin to fmax Hz. Enter Nw points and fmin Hz or wmin rad/sec in the "LLS" tab. Enter fmax Hz or wmax rad/sec in the "fmax" tab.

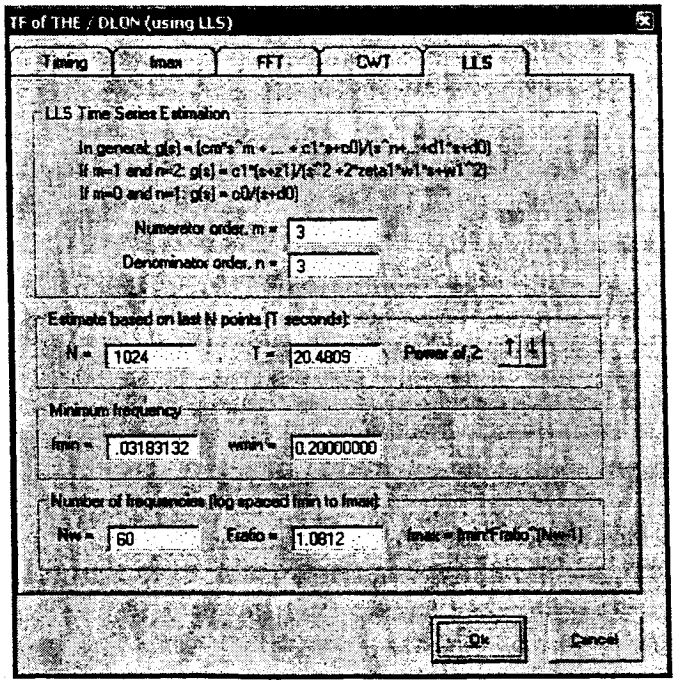


Figure 32. LLS Tab

6.8 LLS Estimation Method

The transfer function parameter estimation method implemented here is the one-step prediction method based on the input and output time series. This is a "bare-bones" parameter estimation method that is used for comparison with the transform methods

The LLS parameter estimation method is briefly described. The starting point is the input and output times series x_i and y_i defined for points in the sliding window. The objective is to estimate the parameters of the z-transform:

$$G(z) = \frac{c_m z^m + \cdots + c_0}{z^n + d_{n-1} z^{n-1} + \cdots + d_0}$$

The one-step prediction of the output is based on the previous inputs and outputs:

$$\hat{y}_k = -\left(d_{n-1} y_{k-1} + \cdots + d_0 y_{k-n}\right) + \left(c_m u_{k+m-n} + \cdots + c_0 u_{k-n}\right)$$

The LLS estimation problem finds the parameters that minimize the least square of the error $e_i = y_i - \hat{y}_i$ for points in the sliding window. The error is linear in the z-transform coefficients, and hence the name "Linear Least Squares."

The example in Figure 33 compares the transfer function magnitude computed using the Erlang wavelet transform and using the LLS estimation problem (Run 69 from the LOCATS data). The LLS parameters are those shown in Figure 32. Here the comparison looks good. The LLS estimate is smoother because the selected order of three is not large enough to represent jagged peaks and valleys. Generally, however, the LLS estimate does not do as well, having a larger variance and too often giving nonsense.

There are many variations of the LLS estimation problem that will improve the transfer function estimate, but only the "bare-bones" version is implemented in LOCATS. Several LLS variations were explored and implemented in Matlab, the point being that the simple version in LOCATS is not good enough, and in fact the LLS variations that do the best job of estimating the transfer function use transforms of the time series as an initial step in the z-transform coefficient estimation.

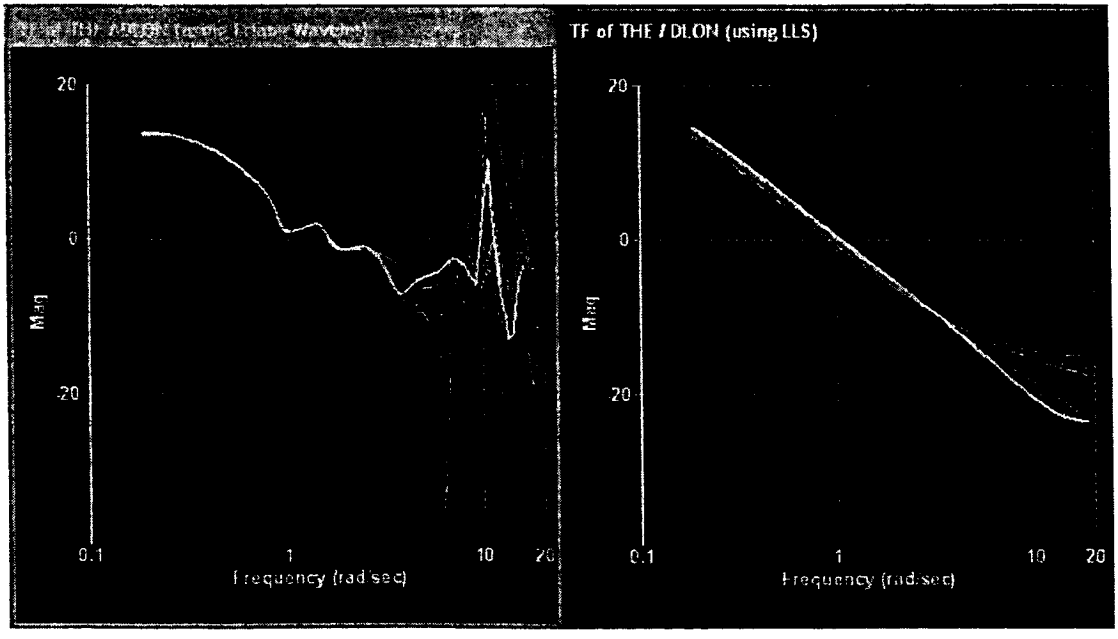


Figure 33. LLS Example

6.9 Smoothing

Smoothing Parameters:

The spectra are averaged in time and frequency as defined by the parameters:

"Time Domain Averaging" in the "Timing" tab is the number of spectra averaged in time.

"Nmin" and "Fratio" in the "FFT" tab defines frequency averaging for FFTs.

"Frequency Domain Averaging" in the "CWT" tab is the number of spectra averaged in frequency for wavelet transforms.

The average is of the auto-spectra and the cross-spectra. The transfer function is the ratio of the averaged cross-spectra and the input auto-spectra. The coherence also uses the averaged quantities, and if no averaging in time or frequency the coherence is unity.

The Smoothing Window:

The cross-spectrum is averaged in time and frequency:

$$\hat{S}_{xy}(\omega_i, t_j) = \frac{1}{N_i N_j} \sum_{k=i-(N_i-1)/2}^{i+(N_i-1)/2} \sum_{\ell=i+1-N_j}^{j} S_{xy}(\omega_k, t_\ell)$$

The same averaging is also done for the auto-spectra $\hat{S}_{xx}$ and $\hat{S}_{yy}$ .The N_i frequencies, where N_i is odd, are centered about the frequency ω_i, and the N_j times are at the current time t_j and previous times (and hence is average is causal). The window used for the average is shown in Figure 34.

The total time and frequency range define a rectangle, and at the edges of this rectangle the averaged spectra have less than the maximum number of times and frequencies available. The numbers N_i and N_j are reduced to the number actually used in the average. Hence, the expected values of averaged and un-averaged spectra are the same.

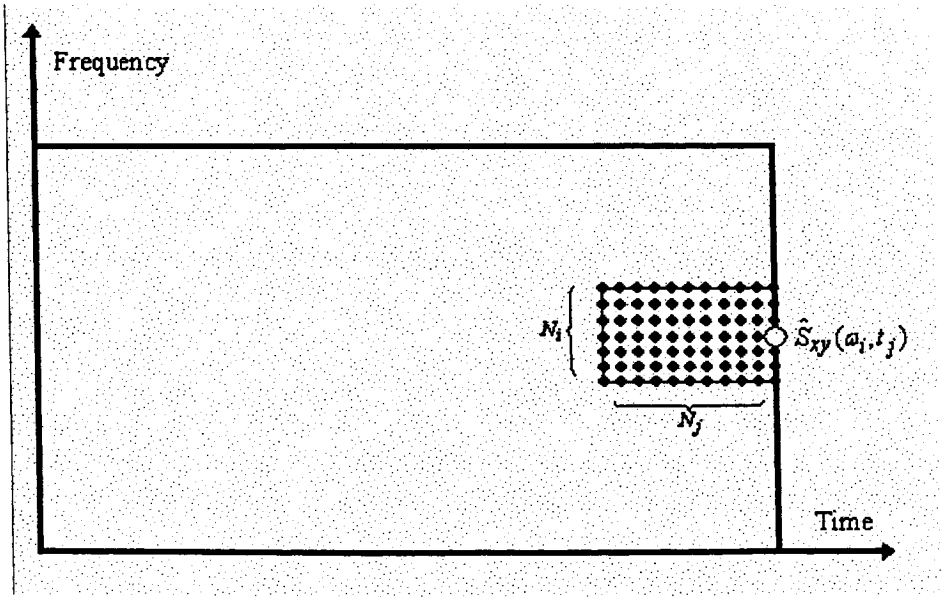


Figure 34. Smoothing Window

An example of time and frequency domain smoothing for a Erlang 5-cycle wavelet transform is shown in Figure 35. The smoothed PSD is of pitch (Run 69 from the LOCATS database, averages 41 adjacent times 0.16 seconds apart, and 5 adjacent frequencies for 120 frequencies from 0.19 to 20 rad/sec).

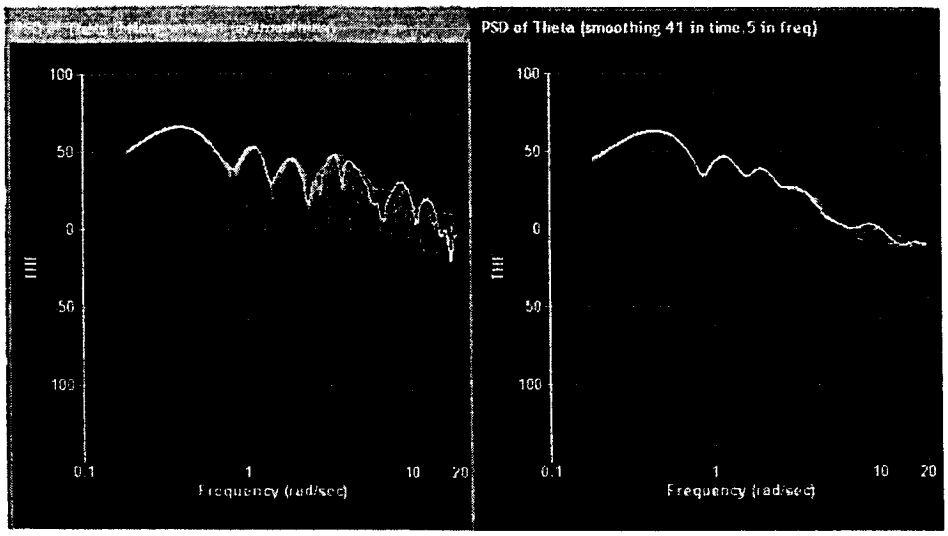


Figure 35. Wavelet Smoothing

FFT Log Binning:

Adjacent FFT frequency bins are averaged by checking the "Use log binning" box in the FFT tab. The dc component is not used in the average, the low frequency bin is at 1/T Hz, and the maximum frequency is fmax Hz as set in the fmax tab. The number of frequency bins used for the average (usually) increases with frequency so that the adjacent averaged frequencies have approximately equal ratio. See Section 6.4 for the log binning parameters. An example of log binning is shown in Figure 36, the PSD of pitch, using the same time series as in Figure 35. There is no smoothing in time.

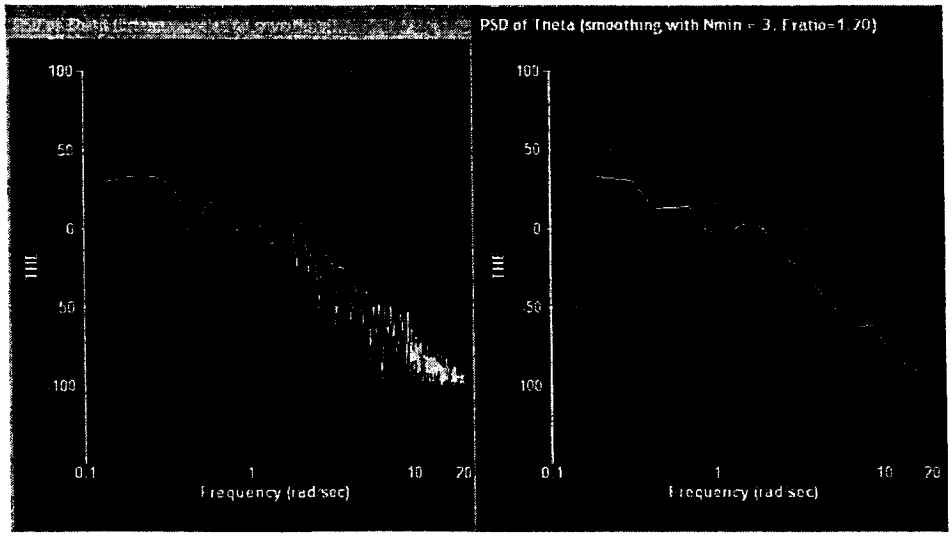


Figure 36. FFT Log Binning

6.10 LOCATS Timing

6.10.1 Timing Loop

The real time signals used by LOCATS are obtained from the Mil 1553 bus via software data requests. These requests are inside a timing loop as follows:

Top of timing loop
- Start 20 msec timer and record current time
- Request signals
- Store signals
- Transform calculations (if any)
- Plotting (if any)
- Loop to use up remaining part of 20 msec (if any)

End of timing loop

The calculations depend on the choice of plots. Strip charts are updated each cycle. The time series for PSD and TF plots are stored every cycle, but the transform calculations and plotting are updated every Nincr samples, where Nincr is a specified number of increments. For example, if one transform has Nincr = 5 and a second has Nincr = 7, then they are computed and plotted respectively every 5 and 7 samples, overlapping every 35 samples.

Another factor in the timing is the number of lines drawn on each plot. The PSD and TF plots can have multiple lines, as determined by the "persistence" parameters. Drawing multiple lines turned out to have a larger timing penalty then computing transforms.

If the calculations and plotting take longer than 20 msecs then the final part of the timing loop is bypassed, and then next cycle starts by resetting the timer to zero and requesting the next batch of signals. The time at the start of each cycle is stored and the delta time indicates whether or not a slip occurs. There is not a master clock with 20 msec increments. The 20 msec timer starts when the previous cycle is completed.

6.10.2 Timing and Quantization of Buffered Signals

The signals on the Mil 1553 bus are updated according to the simulator software. The signals are not under the control of LOCATS and their updates are not synchronized with LOCATS. The Mil 1553 bus holds the previous value of each signal until it is updated.

A detail of signals from LOCATS Run 41 is shown in Figure 37. (There figures use post-processed data collected by LOCATS, but the plots are from Matlab and are not part of LOCATS). Each dot is a recorded value for that signal. The signal timing and quantization can be determined from these plots. Comments:

- The first line is the delta time between successive signals, and in this 2.5 second segment of data the delta time holds steady at 20 msec indicating there is no slippage.
- Each sample of the Q signal is held for 2 cycles, indicating the update rate is about half of the 50 Hz rate set for LOCATS.
- Each sample of the VCAS signal is held for at least 4 cycles, indicating an update rate about one-forth of 50 Hz. Sometime the signal is held for more than 4 cycles, due to quantization.
- Most samples of the THE signal are held for 1 cycle, and occasionally 2 cycles, indicating an update rate of about 50 Hz but not quite synchronized with the LOCATS update.

- The DLON samples are all held for one cycle, perfect synchronization over this segment of data.
- The lateral axis signals (PHI, P, and DLAT) are all held for many cycles, due mainly to quantization. The pilot input is primarily pitch for this run.

6.10.3 Timing Tests

LOCATS requests data from the Mil 1553 bus every 20 msec, but slips when the calculations and/or plotting takes more than this amount. The time of each sample is recorded from a precision timer built into the PC, and the delta time can therefore be used to test the computation time.

Several LOCATS runs were used primarily to record calculation times. The type of input is not relevant to the timing, but for the record each of these timing test runs was a single pitch doublet with data collected for 20 to 30 seconds. The number of plots and the length of the transform calculations were varied for each run. The transform calculations were either FFTs or wavelets, computed every Nincr samples, one time for each plot. The "persistence" is the number of lines drawn for each PSD plot, a persistence of 10, for example, plots the current PSD line and the 9 previously calculated PSD lines.

The results of the timing tests are summarized in Table 1. The delta time plots for most of the runs are included in Figure 38 (the runs not included have flat, 20 msec delta time plots).

Conclusions:

- No slippage with just strip charts.
- No slippage for a single PSD plot using either FFTs or wavelets up to length 4096. The number 4096 can be increased some, but the tests are not sufficient to determine an exact bound.
- Plotting with persistence results in severe timing slippage, taking as much as 10 times longer than then the transform calculations.

Recommendations:

- Use the current version of LOCATS data with just strip charts. This is the safest way to obtain data with no slippage (this was done for the LOCATS runs above R8).
- Change the LOCATS software to separate the data collection from the calculations and plotting. Do this either with a single computer using interrupt driven multiple threads, or with separate computers. Writing this type of software was beyond the scope of the current project.

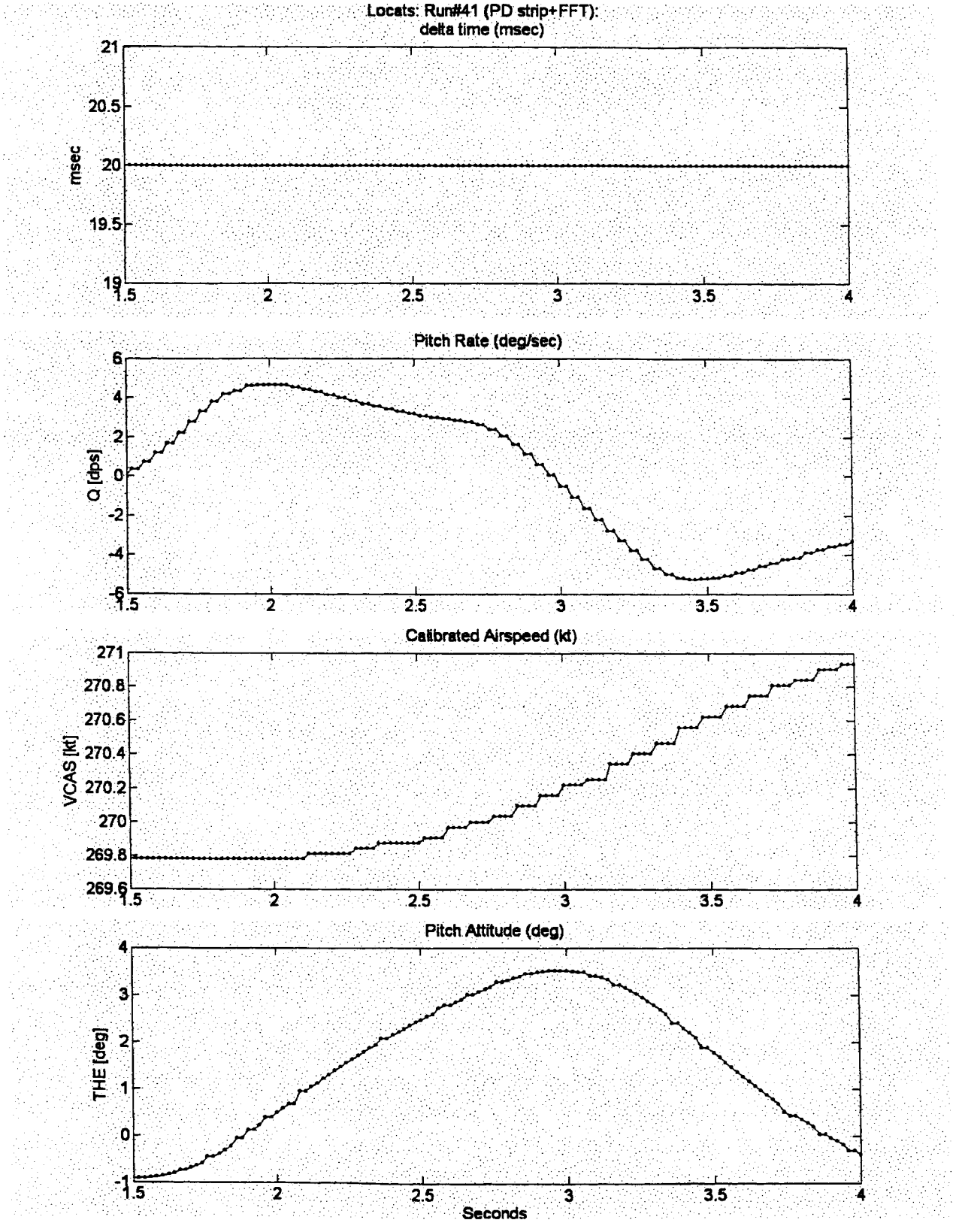


Figure 37. (Part 1 of 2) Detail of Signals Showing Timing and Quantization

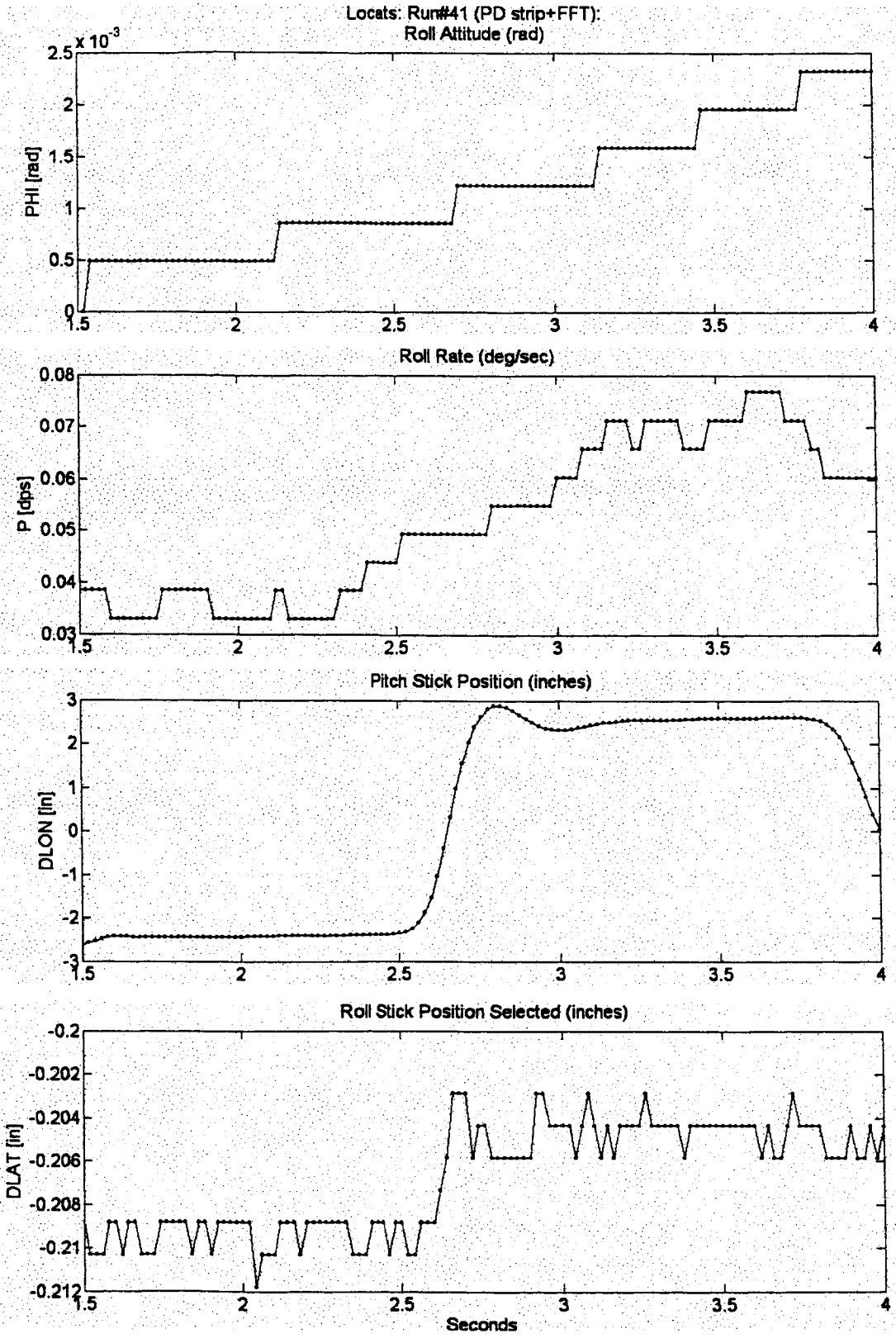


Figure 37. (Part 2 of 2) Detail of Signals Showing Timing and Quantization

Table 1. LOCATS Timing Tests

LOCATS Run #	Description Of Run	Real Time Plots	# Samples out of first 1000 that slip more than 0.1 msec (nominal value is 20 msec)
39	Pitch doublet	Strip chart only	0
41	Pitch doublet	Strip chart 4096 point FFT (Persistence = 1)	0
42	Pitch doublet	Strip chart 4096 point FFT (Nincr = 6, Persistence = 1) 4096 point Wavelet (Nincr = 8, Persistence=1))	158 (slips to about 30 msec every 24 samples when transforms line up)
43	Pitch doublet	Strip chart 4096 point FFT (Nincr = 6, Persistence = 10) 4096 point wavelet (Nincr = 8, Persistence = 10)	196 (slips to 40 to 60 msec for each transform plot, slips to 140 to 160 msec every 24 samples when transform plots line up)
60	Pitch doublet	Strip chart 8192 point wavelet (Nincr = 6, Persistence = 1)	136 (slips 0 to 4 msec for each transform)
61	Pitch doublet	Strip chart 4096 point wavelet (Nincr = 6, Persistence = 1)	0
62	Pitch doublet	Strip chart 6144 point wavelet (Nincr = 6, Persistence = 1)	3 (slips more than 0.1 msec 3 times, with a max slip of 0.6 msec)
63	Pitch doublet	Strip chart 8196 point FFT (Nincr = 6, Persistence = 1) 6144 point wavelet (Nincr = 7, Persistence = 1)	35 (slips 6 to 10 msec every 42 samples when transform line up)

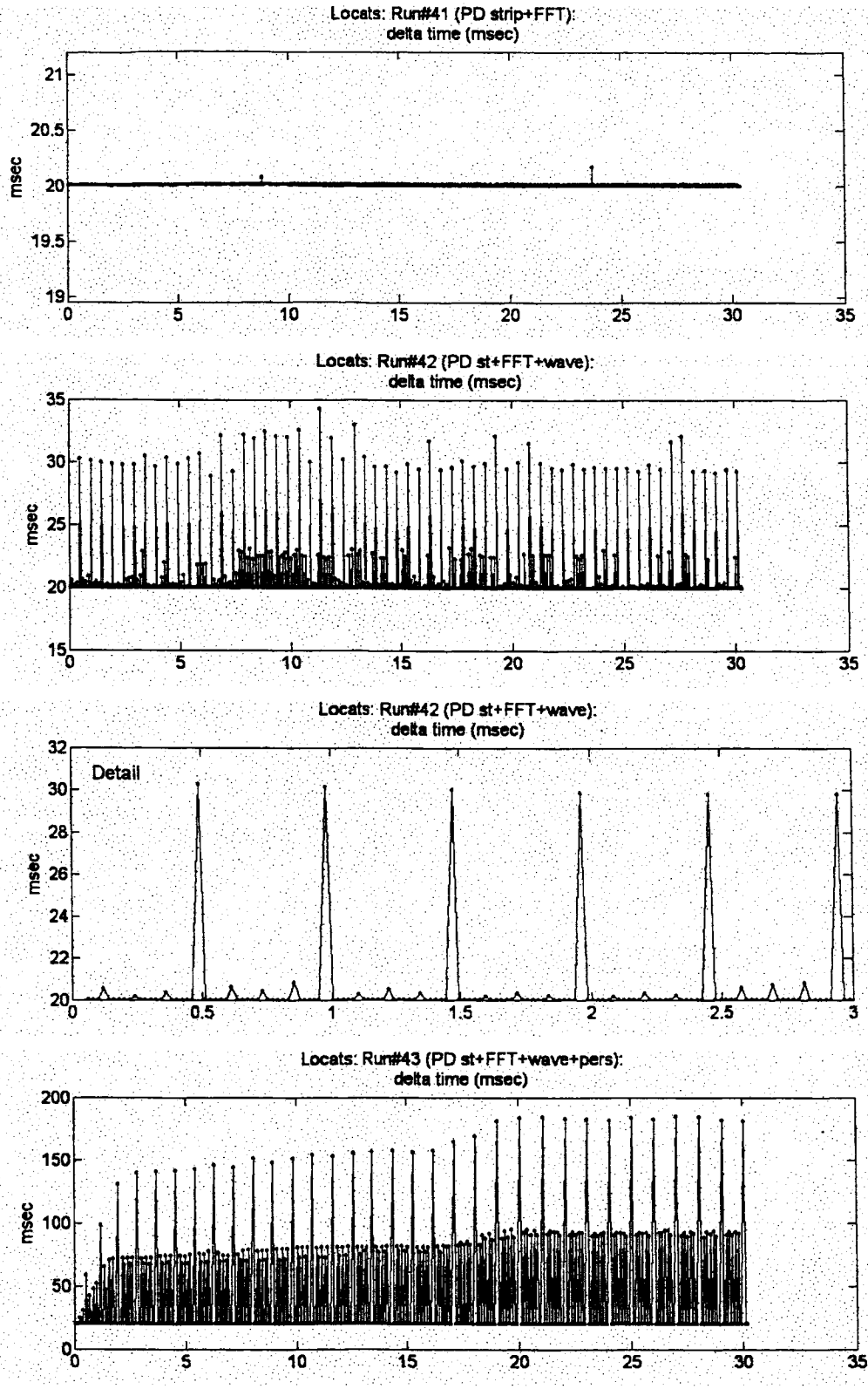


Figure 38. (Part 1 of 2) LOCATS Timing Tests

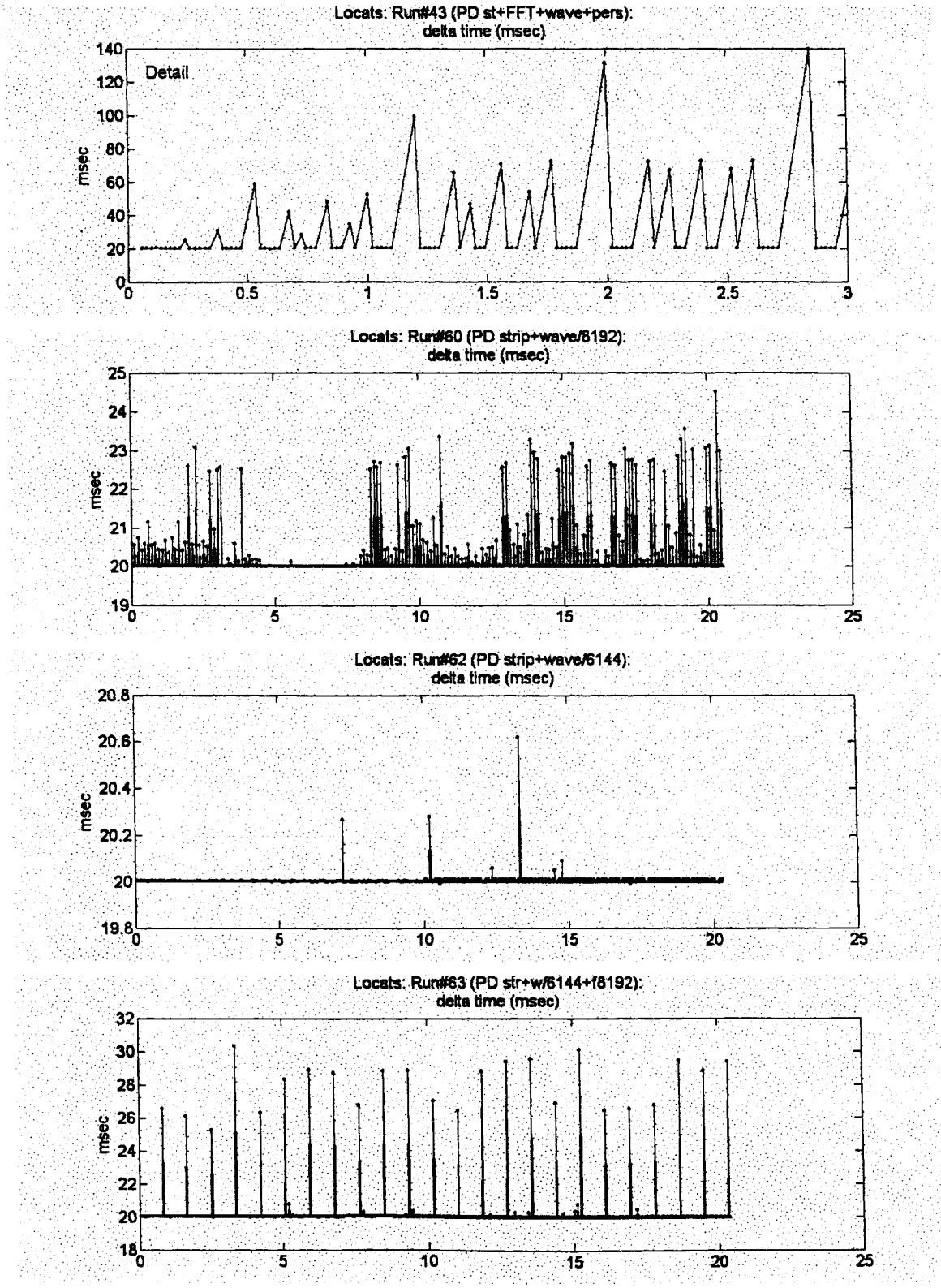


Figure 38. (Part 2 of 2) LOCATS Timing Tests

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 30, 2003	Final Technical Report 01/31/01 – 09/30/03

4. TITLE AND SUBTITLE
On-Line Loss of Control Detection Using Wavelets
Volume III: LOCATS User's Manual

5. FUNDING NUMBERS
NAS4-010004

6. AUTHORS
Peter M. Thompson, Ph.D., David H. Klyde, Theodore J. Rosenthal, Edward N. Bachelder, Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Technology, Inc.
13766 S. Hawthorne Blvd.
Hawthorne, CA 90250

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-1341-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASA
Dryden Flight Research Center
P.O. Box 273
Edwards, CA 93523-0273

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Wavelet transforms are used for on-line detection of aircraft loss of control. Wavelet transforms are compared with Fourier transform methods and shown to more rapidly detect changes in the vehicle dynamics. This faster response is due to a time window that decreases in length as the frequency increases. New wavelets are defined that further decrease the detection time by skewing the shape of the envelope. The wavelets are used for power spectrum and transfer function estimation. Smoothing is used to tradeoff the variance of the estimate with detection time. Wavelets are also used as front-end to the eigensystem reconstruction algorithm. Stability metrics are estimated from the frequency response and models, and it is these metrics that are used for loss of control detection. A Matlab toolbox was developed for post-processing simulation and flight data using the wavelet analysis methods. A subset of these methods was implemented in real time and named the Loss of Control Analysis Tool Set or LOCATS. A manual control experiment was conducted using a hardware-in-the-loop simulator for a large transport aircraft, in which the real time performance of LOCATS was demonstrated. The next step is to use these wavelet analysis tools for flight test support.

14. SUBJECT TERMS
Loss of control, wavelets, time varying transfer functions, Fourier transforms, eigensystem reconstruction algorithm

15. NUMBER OF PAGES
49

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-07-2005	Contractor Report	01/31/01 to 09/30/03

4. TITLE AND SUBTITLE
On-Line Loss of Control Detection Using Wavelets

5a. CONTRACT NUMBER
NAS4-01004

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Peter M. Thompson, Ph.D.; David H. Klyde; Edward N. Bachelder, Ph.D.; and Theodore J. Rosenthal

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Technology, Inc.
13766 S. Hawthorne Blvd.
Hawthorne, California 90250-7083

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-1341-1, TR-1341-2, TR-1341-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/CR-2005-212873

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified -- Unlimited
Subject Category 08 Availability: NASA CASI (301) 621-0390

13. SUPPLEMENTARY NOTES
Technical Monitor, Martin J. Brenner, Aerospace Engineer, NASA Dryden Flight Research Center, P.O. Box 273, MS/4840D, Edwards, California.

14. ABSTRACT

Wavelet transforms are used for on-line detection of aircraft loss of control. Wavelet transforms are compared with Fourier transform methods and shown to more rapidly detect changes in the vehicle dynamics. This faster response is due to a time window that decreases in length as the frequency increases. New wavelets are defined that further decrease the detection time by skewing the shape of the envelope. The wavelets are used for power spectrum and transfer function estimation. Smoothing is used to tradeoff the variance of the estimate with detection time. Wavelets are also used as front-end to the eigensystem reconstruction algorithm. Stability metrics are estimated from the frequency response and models, and it is these metrics that are used for loss of control detection. A Matlab toolbox was developed for post-processing simulation and flight data using the wavelet analysis methods. A subset of these methods was implemented in real time and named the Loss of Control Analysis Tool Set or LOCATS. A manual control experiment was conducted using a hardware-in-the-loop simulator for a large transport aircraft, in which the real time performance of LOCATS was demonstrated. The next step is to use these wavelet analysis tools for flight test support.

15. SUBJECT TERMS
Aircraft simulation, Eigensystem realization algorithm (ERA), Stability and health monitoring, System identification, Wavelets

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	U	472	19b. TELEPHONE NUMBER *(Include area code)* (301) 621-0390