

NASA Technical Memorandum 107600

1N-05  
115121  
P.11

**THE MULTIPLE-FUNCTION  
MULTI-INPUT/MULTI-OUTPUT  
DIGITAL CONTROLLER SYSTEM  
FOR THE AFW WIND-TUNNEL MODEL**

**Sherwood T. Hoadley and Sandra M. McGraw**

**July 1992**

(NASA-TM-107600) THE  
MULTIPLE-FUNCTION  
MULTI-INPUT/MULTI-OUTPUT DIGITAL  
CONTROLLER SYSTEM FOR THE AFW  
WIND-TUNNEL MODEL (NASA) 11 p

N92-32536

Unclass

G3/05 0115121



National Aeronautics and  
Space Administration

**Langley Research Center**  
Hampton, Virginia 23665-5225



# THE MULTIPLE-FUNCTION MULTI-INPUT/MULTI-OUTPUT DIGITAL CONTROLLER SYSTEM FOR THE AFW WIND-TUNNEL MODEL

Sherwood T. Hoadley\*  
Aeroservoelasticity Branch  
NASA Langley Research Center  
and  
Sandra M. McGraw  
Lockheed Engineering and Sciences Co.  
Hampton, Virginia

## Abstract

A real-time multiple-function digital controller system was developed for the Active Flexible Wing (AFW) Program. The digital controller system (DCS) allowed simultaneous execution of two control laws: flutter suppression and either roll trim or a rolling maneuver load control. The DCS operated within, but independently of, a slower host operating system environment, at regulated speeds up to 200 Hz. It also coordinated the acquisition, storage, and transfer of data for near real-time controller performance evaluation and both open- and closed-loop plant estimation. It synchronized the operation of four different processing units, allowing flexibility in the number, form, functionality, and order of control laws, and variability in selection of sensors and actuators employed. Most importantly, the DCS allowed for the successful demonstration of active flutter suppression to conditions approximately 26% (in dynamic pressure) above the open-loop boundary in cases when the model was fixed in roll and up to 23% when it was free to roll. Aggressive roll maneuvers with load control were achieved above the flutter boundary. The purpose of this paper is to present the development, validation and wind-tunnel testing of this multiple-function digital controller system.

## Nomenclature

<b>B</b>	Matrix for blending sensor input signals into control law inputs
<b>D</b>	Matrix for distributing control law outputs to various actuator commands
<b>F,G,H',E'</b>	Control law quadruple
<b>K<sub>EU/V</sub></b>	Scaling parameter for control law sensor input to convert from A/D integer voltage counts to engineering units
<b>K<sub>V/EU</sub></b>	Scaling parameter for control law actuator command to convert from engineering units to D/A integer voltage counts
<b>K</b>	Overall feedback gain
<b>p</b>	Roll rate
<b>p<sub>c</sub></b>	Roll-rate command
<b>p<sub>cap</sub></b>	Capture roll rate = 5 deg/sec.
<b>p<sub>L</sub></b>	Peak hold roll rate
<b>t<sub>B</sub></b>	Break time when p <sub>c</sub> reaches peak hold roll

\* Associate Fellow

<b>t<sub>cap</sub></b>	rate Capture time when p = p <sub>cap</sub>
<b>t<sub>T</sub></b>	Termination time when $\phi = \phi_T$
<b>u<sub>k+1</sub></b>	k+1 <sup>st</sup> control law output
<b>x<sub>k</sub></b>	k <sup>th</sup> control law state
<b>y<sub>k</sub></b>	k <sup>th</sup> time-step sensor inputs to control law (in A/D integer voltage counts)
<b>y'<sub>k</sub></b>	Blended k <sup>th</sup> time-step, scaled, floating point control law input
<b>δ<sub>k+1</sub></b>	k+1 <sup>st</sup> control law actuator command (in integer counts for D/A)
<b>φ</b>	Roll angle (measured)
<b>φ<sub>c</sub></b>	Roll-angle command
<b>φ<sub>cap</sub></b>	Roll angle at time t <sub>cap</sub>
<b>ω</b>	frequency

## Subscripts

<b>A</b>	Antisymmetric
<b>S</b>	Symmetric

## Notation

<b>{•}</b>	Vector
<b>[•]</b>	Matrix
<b>A/D</b>	Analog-to-digital
<b>D/A</b>	Digital-to-analog

## Acronyms

<b>AFW</b>	Active Flexible Wing
<b>AP</b>	Array Processor
<b>CPE</b>	Controller Performance Evaluation
<b>CPU</b>	Central processing unit
<b>C1</b>	A SKY Computers, Inc. Challenger-1 integer DSP board
<b>C30</b>	A SKY Computers, Inc. Challenger-C30/V multiprocessor board
<b>DCS</b>	Digital controller system
<b>DSP</b>	Digital Signal Processor
<b>FFT</b>	Fast Fourier Transform
<b>FSS</b>	Flutter suppression system
<b>HBS</b>	Hot Bench Simulation
<b>MIMO</b>	Multi-input/multi-output
<b>RMLA</b>	Rolling Maneuver Load Alleviation System
<b>RRTS</b>	Roll Rate Tracking System
<b>RTS</b>	Roll-Trim System

## I. Introduction

The Active Flexible Wing (AFW) Program<sup>1,2</sup> was a cooperative effort between the NASA Langley Research Center and the Rockwell International Corporation. One

of the specific program objectives was the validation of analysis and synthesis methodologies as applied to the multiple-function control of a sophisticated full-span, free-to-roll, aeroelastically scaled wind-tunnel model.

The control functions being investigated included the suppression of flutter, roll trim control, rolling maneuvers with load control and load alleviation. Figure 1 depicts the multiple-function control requirements, indicating that the flutter suppression system (FSS) had to be capable of being switched in with any of three different roll control systems: Roll-Trim System (RTS), Roll Rate Tracking System (RRTS), or Rolling Maneuver Load Alleviation System (RMLA).

Meeting the primary objectives of the AFW Wind-Tunnel Program required gaining practical experience in designing, fabricating, and implementing a real-time multiple-function multi-input/multi-output (MIMO) digital controller and interfacing hardware. As a result, a versatile digital controller system was developed which operated at 200 Hz within a slower operating system environment allowing for simultaneous multiple function control.

The purpose of this paper is to present the development, validation, and the wind-tunnel testing of the AFW digital controller system (DCS). The generic forms of the control laws developed to suppress flutter, the forms of roll and maneuver load alleviation control laws used to maximize roll performance, techniques employed to verify control laws, and procedures used to validate the DCS are described.

## II. Digital Controller System Design

Most modern computers operate within a time-share operating system, capable of performing many tasks which share the central processing unit (CPU). These operating systems are not designed to enable one task to operate at regulated frequencies in a real-time fashion, oblivious to other tasks being performed. Consequently, the DCS was designed using a separate dedicated processor as the real-time system executor to perform real-time control functions independent of the host CPU. This real-time system executor had to be capable of controlling data transfer over the data BUS and to start and stop processes. An integer digital signal processor with an internal clock which could be used for regulating speeds was selected to perform this task. However, an integer processor does not lend itself to changing control law parameters quickly because scaling to avoid overflow and underflow for each control law must be included in the code. The design requirement that control laws be easily modified or changed was a driving constraint. It forced the use of separate floating point processors to execute the various control laws. Hence, a dedicated integer processor was used as executor of the real-time system and dedicated floating point processors were used to execute individual control laws.

A SUN 3/160 Workstation, driven by a Unix Operating System, was selected as the "shell" of the DCS. The DCS had three special purpose processing units linked via a data BUS which included an integer Digital Signal Processor (DSP), a floating point DSP board with two microprocessors, and an Array Processor (AP).

Unlike its analog counterpart comprised primarily of fixed hardware circuitry, the DCS was designed to allow flexibility: in the number, functionality and form of control laws to be implemented; in the selection of sensors and actuators employed; in the number of states in the state-space representations; and in the size and number of tables used for control laws using table-lookup and interpolation. The DCS was also designed to coordinate data acquisition, storage, and transfer.

Components of the DCS and interfacing hardware either to the near real-time simulation of the wind-tunnel model or to the wind-tunnel model itself are depicted in figure 2. The DCS itself, on the left side of figure 2, depicts schematically that the host CPU, the disk and tape drives, and the added boards communicated across the data BUS. The host CPU and the Status Display Panel provided user interface to the real-time system. A SKY Computers, Inc. Challenger-1 (C1) integer DSP board controlled the real-time processing. Most control law computations were performed on a SKY Challenger-C30/V (C30). This was a high speed, floating point, 32-bit systems oriented digital signal two-processor board. The AP was another SKY board which provided high-speed direct memory access for the DCS and vectorized floating point processing. In case the C30 failed, the AP board performed floating-point calculations for the FSS and RMLA control laws as a backup<sup>3</sup>. Backup for the RTS was performed by the C1 using integer arithmetic. There was no backup for the RRTS system. Two analog-to-digital (A/D) and two digital-to-analog (D/A) converter boards, manufactured by Data Translation, Inc., provided the link between the digital and analog worlds, providing all of the analog/digital data conversions required between the model and the DCS. They converted the incoming analog voltages from the sensors to 12-bit digital values and 12-bit digital signals such as the control surface actuator commands into outgoing analog voltages. The entire real-time operation from sensor input to actuator command output was repeated at regulated speeds up to a maximum requirement of 200 times each second.

The Interface Electronics hardware components are shown schematically on the right side of figure 2 in a rack labeled Interface Electronics. This rack contained the analog circuitry for processing the analog signals coming from or going to either the wind-tunnel model or the simulator. The Filter Box housed analog antialiasing filters, analog notch filters, and electrical isolation networks. The analog antialiasing filters were configured to provide either first-order roll-off or fourth-order roll-off with either a 25 Hz break frequency or a 100 Hz break frequency. The sensor signals coming to the DCS or the commands going to the model could also be filtered through analog notch filters, if desired, to filter out undesired frequency ranges. The Patch Box allowed direct input/output of analog signals by the DCS without additional filtering.

The Status Display Panel, designed and built in-house by NASA, displayed, through status lights, the real-time status of various control parameters such as the feedback switch. It also displayed the system pulse.

Although not shown in the figure, a second SUN Workstation, configured similarly to the DCS was used

not only as a back-up for the DCS, but also as a near real-time multi-signal digital analyzer to evaluate controller performance and estimate both the open- and closed-loop plants<sup>4,5,6</sup>. It was linked to the DCS via an Ethernet line, allowing for fast data transfer of blocks of sampled data.

A separate Digital Signal Analyzer, was also used to verify analog signals and to debug problems associated with the DCS plus Interface Electronics.

### III. Functionality of Software Components

As functions of the DCS were identified, separate program components were developed which performed the various functions. These components, outlined in figure 3, are described below. Except for the commands required to perform the actual calculations on the AP, all of the software was written in a high level (C) programming language. Operation code command blocks were generated for the AP.

Three user interface programs for the DCS were executed on the host computer. They are: the User/Controller Interface program; the Data Transfer program; and the Information Display program. The User/Controller Interface program provided the DCS operator with communication links to the real-time system. All user options, control law arrays, control parameters, and excitation definitions were specified through this program and downloaded into C1, C30, or AP memory as required. The array-processor command blocks were also determined and downloaded by this program into C1 memory. Through this program, the DCS operator was able to perform such tasks as: selecting the mode of operation; changing gains; selecting control laws; opening and closing control law loops; selecting excitations; changing excitation amplitudes; and selecting the control surfaces to be excited. The Data Transfer program controlled the formatting and sending of the digitized data to external files or tapes for control law verification, performance, and Controller Performance Evaluation (CPE)<sup>4,5,6</sup>. The Information Display program displayed all pertinent DCS information such as control-surface deflections, errors between commanded and actual deflections, and switch settings verifying operator selections.

The real-time system was controlled by a Real-Time Executor program which resided on the C1 board. As BUS master, this Real-Time Executor provided management of all real-time activities and tasks. It controlled all of the real-time processing: digitizing of analog input signals; converting of output signals to analog voltages; starting control-law execution; summing of digitized excitations with bias commands to statically position and excite control surfaces. The Real-Time Executor also provided the interface to the Status Display Panel lights, checked for faults, and instructed the host computer when blocks of data were stored and could be transferred.

Primary control law processing was performed by programs residing on the C30 board, referred to herein and in figure 3, as the Control Law Processor. This board was comprised of shared global memory and two complete processing nodes, referred to herein as Nodes 1 and 2.

Each node had its own memory in which arrays defining the control laws and control law execution code were stored. The globally shared memory could be accessed by the Real-Time Executor, the User/Controller Interface and by both node processors without interrupting node processing. Two programs, each residing on separate nodes, performed the different control law calculations as follows:

#### Node 1:

RTS control law transfer function

RRTS table look-up and interpolation

RMLA state-space matrix computations

#### Node 2:

FSS state-space matrix computations.

Computations included all unit conversions, scaling, matrix computations, and interpolations. A bivariate table-lookup capability with linear interpolation was developed for the RRTS with a fixed number of inputs and outputs, but variably sized tables so that interpolations could be improved by enlarging the number of elements in the tables.

Vectorized floating-point computations used by the backup system were performed on the AP, referred to as the Backup Processor in figure 3. Because this was a single-processor, however, the simultaneous control law calculations needed for multiple-function control could not be performed by the backup system within the time constraints imposed by a 200 Hz sampling rate.

### IV. Generic Forms of FSS and RMLA Control Laws

A generic form of the control laws for FSS and RMLA was identified such that one set of software would accommodate both types of control laws while imposing minimal constraints on the control law designers. The generic structure allowed the designers: choice of sensors with the option to blend them; freedom of control law order with upper limits; scheduling of control law parameters with respect to dynamic pressure; and selection of various control surfaces with or without distribution of control law outputs to different actuators.

The FSS and RMLA control laws were implemented using the following difference equations:

$$\begin{aligned} \{x_{k+1}\} &= F\{x_k\} + G\{y'_k\} \\ \{u_{k+1}\} &= H\{x_{k+1}\} + E\{y'_{k+1}\} \\ &= H'\{x_k\} + E'\{y'_k\} \\ \{\delta_{k+1}\} &= D\{u_{k+1}\}, \end{aligned} \quad (1)$$

where

$$\{y'_k\} = B\{y_k\},$$

$$H' = HF \text{ and}$$

$$E'\{y'_k\} = (HG\{y'_k\} + E\{y'_{k+1}\}).$$

If  $E=0$  or  $\{y'_k\} = \{y'_{k+1}\}$ , then  $E'\{y'_k\} = (HG+E)\{y'_k\}$ , and these equations could be implemented with a one time-step delay between input and output instead of a two

time-step delay. The DCS implemented the equations assuming a one time-step delay. The vectors  $\{y\}$  and  $\{\delta\}$  in eq. (1) are subsets, selected by the control law designer, of the analog sensor input signals and actuator command signals.

RMLA and FSS designers provided the control law quadruples:

$$[F, G, H', E'],$$

sensor blending matrix,  $B$ , and the control law output distribution matrix,  $D$ , if desired, and a list of desired sensor inputs and actuator outputs for each control law.

FSS designers provided these for both symmetric and antisymmetric control laws, separately. These were then combined into a single system defined by the following equations:

$$\begin{aligned} F &= \begin{bmatrix} F_S & 0 \\ 0 & F_A \end{bmatrix}, \quad G = \begin{bmatrix} G_S & 0 \\ 0 & G_A \end{bmatrix}, \\ H' &= \begin{bmatrix} H'_S & 0 \\ 0 & H'_A \end{bmatrix}, \quad E' = \begin{bmatrix} E'_S & 0 \\ 0 & E'_A \end{bmatrix}, \end{aligned} \quad (2)$$

$$\begin{aligned} B &= \begin{bmatrix} B_S & 0 \\ 0 & B_A \end{bmatrix}, \\ D &= \begin{bmatrix} D_S & 0 \\ 0 & D_A \end{bmatrix}. \end{aligned}$$

and

#### V. Digital Controller System Modes of Operation

There were seven basic modes of operation defined for the DCS, labeled MAINTENANCE, MANUAL, STATIC, RTS, FSS, RMLA, and RRTS. In all modes of operation, the conversion of the 12-bit signals from the DT boards to 16-bit integers was performed by the Real-Time Executor using masking operations. In the first two modes, no data was stored or saved, and in the first three modes, no control laws were executed. In the last four modes, the averaging of the signals for the FSS control law were performed by the Real-Time Executor using binary shift operations. Signal data to be saved were sent to a block of AP memory for temporary storage.

In each mode of operation, there was a block of "slow-cycle" code performing 10 different secondary communication tasks, each executed once in every 10 iterations. Included in this was code to read switch settings downloaded from the host User/Controller Interface program and code to send DCS parameters to the Information Display program. Types of parameters sent included:

- Selected mode of operation,
- Desired sampling frequency,
- Selected control law,
- Selected control law scheduling option,
- Selected option of feedback loop, opened or closed,
- Selected excitation and symmetry,
- Selected point for adding excitation to the system (actuator commands, control law outputs, or sensor inputs), and
- Tunnel parameters.

The primary functions of each mode of operation are described below.

The primary function of the MAINTENANCE mode was the check-out of all analog-digital links and hardware. This mode allowed for the individual checking of each input and output signal line and allowed the most basic hardware debugging, without interference from code designed for control law execution or data saving.

The primary function of the MANUAL mode was static positioning of control surfaces and checking of scale factors for each signal with a minimal amount of code involved. This, too, was primarily a debugging mode.

The primary function of the STATIC mode was the static positioning and/or excitation of control surfaces while saving of data from the different sensors. Excitation signals could be sent individually or to pairs of control surfaces, either symmetrically or antisymmetrically. This mode was designed primarily for obtaining data about the model to develop improved plant models.

The next four modes of operation involved execution of various control laws separately or simultaneously. One roll control law could be operated simultaneously with flutter suppression (both switch selectable) in all four modes. The basic differences between the four modes were defined by which control law was dominant. The data which was sampled and saved, the types of commands which could be executed, the points at which excitations could be added varied with each mode. For instance, in the FSS mode, the FSS control law was dominant, and the data which were saved in this mode were those related directly to FSS control law execution, verification and performance.

Figure 4 is a detailed schematic of the blocks of code and signal flow involved in the execution of these last four modes. All blocks of code and paths which are not delineated by bold rectangles were executed by the Real-Time Executor. Code delineated by bold rectangles were performed by the specified processor. Software flags were sent to each processor to initiate desired control laws. (If the backup system was employed, commands blocks to operate the AP were sent by the Real-Time Executor to the AP during each execution time cycle.)

The actuator commands for performing roll trim,  $\delta_{TRIM}$ , or a rolling maneuver,  $\delta_{RRTS}$  or  $\delta_{RMLA}$ , were combined with static positioning or bias commands,  $\delta_{BIAS}$ , and then limited to a maximum deflection of  $\pm 10^\circ$ . Actuator excitations (if any) and FSS actuator commands,  $\delta_{FSS}$ , were then combined with the deflection-limited commands to form the final actuator commands as depicted in figure 4. These were then converted to analog signals to be sent to the model.

Each control system feedback was individually switch selectable; i.e., each control law loop could be individually opened or closed, but no two roll control laws could be operated simultaneously. Bias commands, roll-trim commands, and the roll-rate commands were implemented using a ramping procedure which ramped in (or ramped out) the command rather than introducing a command as an instantaneous step.

In both RMLA and RRTS modes of operation, the RTS was used to hold the model at an initial roll angle until the roll-rate command was invoked. Both the

RMLA and RRTS modes of operation were coded in such a way that an RMLA or RRTS control law was invoked simultaneously with a specified roll-rate command when the RMLA or RRTS control law was enabled by the operator. Once the desired termination angle was passed, the roll-rate command was ramped out. When the roll-rate was below a specified "capture rate", the RTS was then re-invoked automatically to trim the model to the current roll angle of the model at the time the capture rate was achieved. The roll-rate command used in both modes is shown in figure 4 as a ramp-in/hold/ramp-out command and is detailed in figure 5. The initial roll-trim angle and termination angle after which the roll-rate command was ramped back to 0, and the on and off ramp rates for each command were specified at the time of each maneuver.

#### VI. Timing

Each of the different tasks performed by the Real-Time Executor required a varying amount of time depending on the mode and options selected, the signals employed, and the size of each control law being executed. The approximate amounts of time involved in performing the various tasks in the FSS mode for executing FSS + (RTS, RRTS, or RMLA) are delineated on a time line shown in figure 6. The time required to perform timing tasks and obtain all the control law output commands from the C30 was approximately 0.4 ms from the start of a cycle. It took about 0.6 ms to sum the various control law outputs, add in an excitation, compute the final actuator commands and send all the output signals to the DIA's. The sampling of the sensors required up to approximately 1.7 ms. Computation of control law inputs and the starting of execution required less than 0.4 ms. Data storage required up to 1 ms. The slow-cycle code required about 0.2 ms to execute. It was determined that the entire set of commands had to be completed within 4.7 ms in order to avoid BUS interference problems and intermittent loss of data.

In the primary system, even the slowest FSS control law could be executed with any roll control law at the required 200 Hz frequency. Many of the options available in the primary system had to be eliminated or reduced in scope when employing the backup system in order to operate at 200 Hz<sup>3</sup>.

#### VII. Validation of Digital Controller System

Validation of the DCS was performed in various stages: during system development using Hot Bench Simulation (HBS)<sup>3,7,8</sup>, during end-to-end tests, and in the wind tunnel. Figure 1 identifies the basic locations of signals at which verification was performed. In each case, the first step was to verify the correct values of all input and output signals against known values at locations A, B, and C in figure 1. This was done first without, and then with, the Interface Electronics (analog filters) in the loop. An oscilloscope was used to check each output signal at A. Once output signals were verified, a fixed voltage was hooked to each input, separately, at location C and then at B. As this was done, each input value was checked and verified.

Once signals were verified, initial validation of implemented control laws was performed. To do this,

open-loop (no plant) control law time responses (at E) due to some excitation into one of the control law inputs (at D) were generated. For FSS and RMLA control laws, the input was usually a unit step. In the case of accelerometer inputs this equaled a 1g step excitation. For the RRTS control law, one cycle of a sine wave, whose range encompassed the input range of the control law, was used. This was done for each combination of control/sensor pairs.

Once the time responses compared to corresponding plots provided by the control law designers, FSS control laws were further validated by generating control law transfer functions for each control/sensor pair. These were generated by inputting an excitation with a specified frequency range content into each control law input, performing FFT calculations, and generating corresponding transfer functions. These had to compare favorably to transfer functions generated analytically by the control law designers. These transfer functions were generated between various points without a model in the loop: E/D, E/C, E/B, and A/B. These last transfer functions for A/B were obtained without the model in the loop (loop open) but with the software feedback switch closed, using a Digital Signal Analyzer which could be hooked directly to the analog signals to debug problems associated with the DCS plus Interface Electronics. During HBS validation, transfer functions for E/D with the simulation model in the loop were extracted from the closed-loop system. The next stage was to repeat these verification procedures during end-to-end testing. Final verification was performed in the wind-tunnel with the model in the loop.

Comparisons of some of these transfer functions for a control/sensor pair used for one of the FSS control laws are shown in figure 7. The analytically generated transfer function in this case, shown by the dashed line, was for a 200 Hz, digitized control law with a 1-time step delay built in. The output for all of these transfer functions is at the control law output point, E, and does not include the negative sign for negative feedback. As should be expected in figures 7(a) and (b), the two curves are coincident. Figure 7(a) shows the initial open-loop, control law only transfer function (E/D) in which the excitation is added internally at the control law input point, D; (b) shows the same transfer function after being extracted from the closed-loop HBS data; and (c) shows the open-loop DCS + analog filters (E/B) in which the excitation is added at the sensor input point, B, and includes the analog antialiasing filters. Phase differences due to the analog antialiasing filters are seen in (c), but do not show up in (b) because the filters in that case are seen as part of the plant. Any differences in phase and gain between the control law transfer functions generated analytically and experimentally had to be accounted for by the control law designer before testing the control law in the wind-tunnel. Figure 7(d) shows the control law transfer function (E/D) extracted from the closed-loop system at 260psf, 11% above the open-loop flutter. Although noisy, because signal-to-noise ratios were low, it verifies that the control law was operating as prescribed, thus further validating the FSS controller during wind-tunnel testing.

The RTS was verified by extensive use with HBS. Figure 8(a) shows a comparison plot of the measured simulation roll angle and the commanded roll angle generated during an HBS run. Plots such as these were generated with the RTS loop closed. Figure 8(b) shows a roll-trim maneuver in the wind-tunnel at a dynamic pressure of 205psf, verifying RTS performance in the wind-tunnel. Plots such as this were generated for various roll commands to verify the RTS prior to any testing of the RRTS, RMLA, or simultaneous FSS and roll control, since the RTS was used in all these modes of operation.

The RRTS and RMLA systems were further verified by generating closed-loop HBS plots of roll-rate commands and actual measured roll-rate. Sensor and control law output data were saved and plotted for various commands. These then were verified by the control law designers.

### VIII. Controller Performance

The final "validation" of the DCS was its demonstrated use in the wind-tunnel. Figure 9 depicts the different combinations and complexity of control laws which were tested in the tunnel. Roll control involving all three roll control systems was achieved simultaneously with flutter suppression up to 23% above the open-loop boundary. Rolling maneuvers with load control were performed up to 17% above the open-loop boundary. All these tests were performed while saving data for on-line analysis within a total combined operating time of less than 5ms, allowing the DCS to operate at the required 200 Hz sampling frequency.

### IX. Concluding Remarks

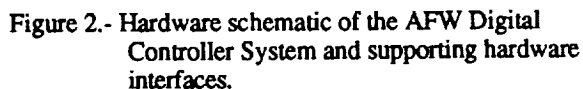
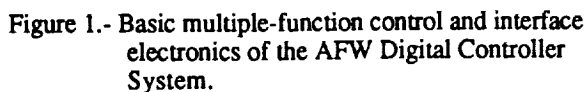
A versatile digital controller system which operated at 200 Hz within a 60 Hz host operating system environment was developed to actively control an aeroelastic wind-tunnel model. It allowed for: simultaneous execution of two control laws; data acquisition, storage, and transfer; flexibility in the form and order (number of degrees of freedom) of control laws implemented; and flexibility in the number and selection of sensors and actuators employed. The system design coordinated and synchronized the operation of four different computing units: a host SUN 3/160 central processing unit, an integer Digital Signal Processor, a floating point Digital Signal Multi-Processor, and an Array Processor.

Most importantly, the DCS allowed the successful demonstration of active flutter suppression while performing aggressive roll maneuvers up to 17% (in dynamic pressure) above the flutter boundary. It allowed the successful demonstration of active flutter suppression to a point 23% above the open-loop boundary when the model was free to roll. By the simultaneous execution of both symmetric and anti-symmetric flutter suppression control laws, it also allowed the successful demonstration of active flutter suppression up to 26% above the open-loop boundary when it was fixed in roll. While executing control laws, the digital controller system acquired data for near real-time controller performance evaluation as well as open- and closed-loop plant estimation.

### References

1. Perry, B. III; Cole, S.; and Miller, G.: *A Summary of the Active Flexible Wing Program*, AIAA Paper No. 92-2080-CP, presented at the AIAA Dynamics Specialists Conference, Dallas, Texas, April 16-17, 1992.
2. Perry, B. III; Mukhopadhyay, V.; Hoadley, S.T.; et al: *Digital-Flutter-Suppression-System Investigations for the Active Flexible Wing Wind-Tunnel Model*, AIAA Paper No. 90-1074-CP, presented at the AIAA/ASME/ASCE/AHS 31<sup>st</sup> Structures, Structural Dynamics, and Materials Conference, Long Beach, California, April 1990. Also published as NASA TM-102618, March 1990.
3. Hoadley, S.H.; Buttrill, C.S.; McGraw, S.M.; and Houck, J.A.: *Development, Simulation Validation, and Wind-Tunnel Testing of a Digital Controller System for Flutter Suppression*. Paper presented at the Fourth Workshop on Computational Control of Flexible Aerospace Systems, July 11-13, 1990, Williamsburg, Virginia. NASA CP 10065, March 1991, pp.583-613.
4. Pototzky, A.S.; Wieseman, C.D.; Hoadley, S.T.; and Mukhopadhyay, V.: *Development and Testing of Methodology for Evaluating the Performance of Multi-Input/Multi-Output Digital Control Systems*, AIAA Paper No. 90-3501, presented at the AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, August 20-22, 1990.
5. Pototzky, A.S.; Wieseman, C.D.; Hoadley, S.T.; and Mukhopadhyay, V.: *On-Line Performance Evaluation of Multi-Loop Digital Control Systems*, AIAA Journal of Guidance, Control and Dynamics, pending publication.
6. Wieseman, C.D.; Hoadley, S.T.; McGraw, S.M.: *On-line Analysis Capabilities Developed to Support the AFW Wind-Tunnel Tests Wing Program*, AIAA Paper No. 92-2084-CP, presented at the AIAA Dynamics Specialists Conference, Dallas, Texas, April 16-17, 1992.
7. Buttrill, C.S.; Bacon, B.; Heeg, J.; Houck, J.A.; and Wood, D.: *Simulation and Model Reduction for the AFW Program*, AIAA Paper No. 92-2081-CP, presented at the AIAA Dynamics Specialists Conference, Dallas, Texas, April 16-17, 1992.
8. Buttrill, C. and Houck, J.: *Hot Bench Simulation of the Active Flexible Wing Wind-Tunnel Model*, AIAA Paper No. 90-3121 presented at the 1990 AIAA Flight Simulation Technologies Conference, Dayton, Ohio, July 17-19, 1990.





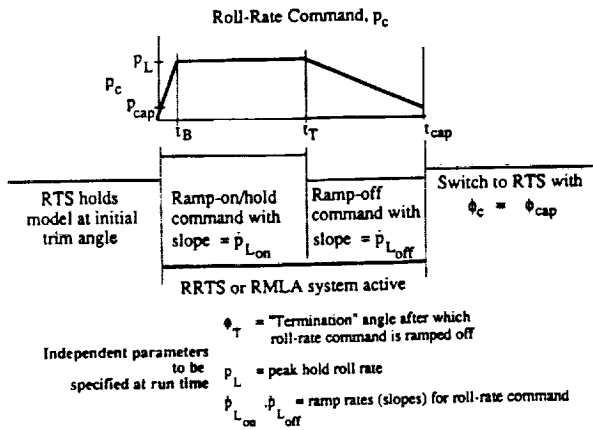


Figure 5.- Ramp-on/hold/ramp-off roll-rate command for RRTS and RMLA control systems.

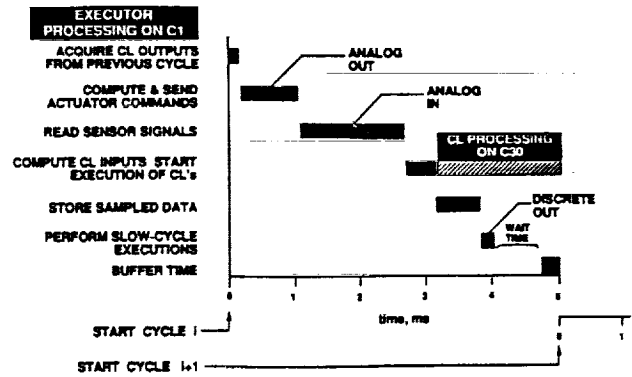


Figure 6.- Time-line diagram of FSS+(RTS, RRTS, or RMLA) control law execution with FSS mode dominant.

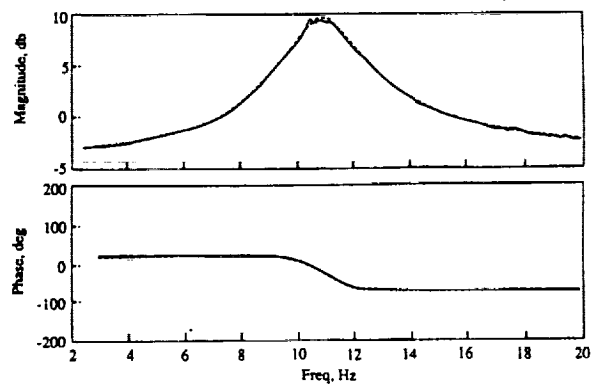
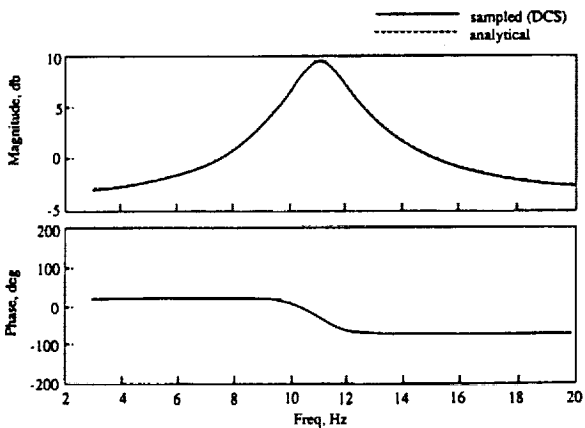
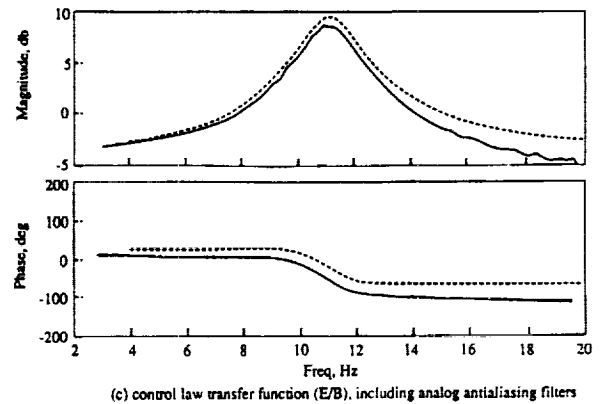
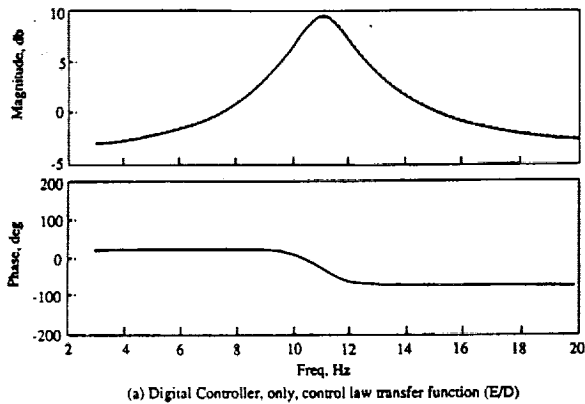


Figure 7.- Transfer function comparisons for validation of Digital Controller flutter suppression system, comparing digitally generated transfer functions with analytically generated ones.

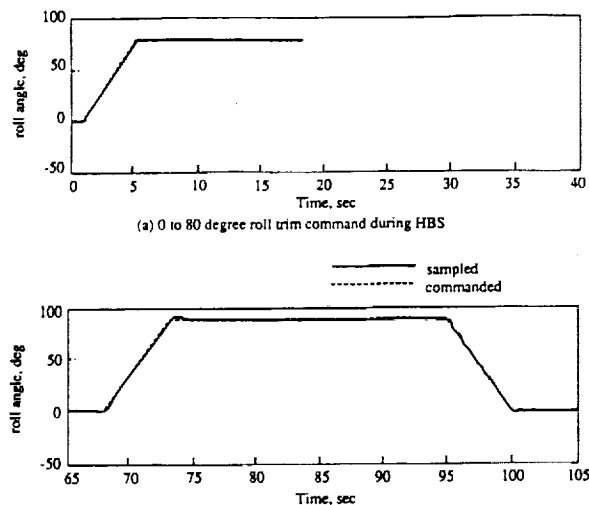


Figure 8.- Measured versus commanded roll-angle comparisons for validation of Digital Controller Roll Trim System.

Multiple Functions	Roll Control		Flutter Suppression		Maximum Test Condition %q above open-loop flutter	
	controls/sensors	control law	controls/sensors	control law	crab	roll maneuver
RMLA + FSS 1		9th order state-space		5th order state-space	23%	11%
RRTS + FSS 2		6 dof bilinear table lookup		3rd order state-space	23%	17%
RRTS + FSS 3		6 dof bilinear table lookup		11th order state-space	23%	11%
RRTS + FSS 4		6 dof bilinear table lookup		3rd order state-space	23%	11%

Figure 9.- Multiple-function MIMO testing accomplished with AFW Digital Controller System.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 1992	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE  The Multiple-Function Multi-Input/Multi-Output Digital Controller System for the AFW Wind-Tunnel Model			5. FUNDING NUMBERS  WU 505-63-50-15	
6. AUTHOR(S)  Sherwood T. Hoadley Sandra M. McGraw				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  NASA Langley Research Center Hampton, VA 23665-5225			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING / MONITORING AGENCY REPORT NUMBER  NASA TM-107600	
11. SUPPLEMENTARY NOTES Presented in the Active Flexible Wing I Session at the AIAA Dynamic Specialists' Conference, Dallas, Texas, April 16-17, 1992. Figures 1, 3, 6, and 9 have been altered from those in the conference paper. Hoadley: NASA Langley Research Center, Hampton, VA; McGraw: Lockheed Engineering and Sciences Co., Hampton, VA.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Unclassified - Unlimited  Subject Category 05			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  A real-time multiple-function digital controller system was developed for the Active Flexible Wing (AFW) Program. The digital controller system (DCS) allowed simultaneous execution of two control laws: flutter suppression and either roll trim or a rolling maneuver load control. The DCS operated within, but independently of, a slower host operating system environment, at regulated speeds up to 200 Hz. It also coordinated the acquisition, storage, and transfer of data for near real-time controller performance evaluation and both open- and closed-loop plant estimation. It synchronized the operation of four different processing units, allowing flexibility in the number, form, functionality, and order of control laws, and variability in selection of sensors and actuators employed. Most importantly, the DCS allowed for the successful demonstration of active flutter suppression to conditions approximately 26 percent (in dynamic pressure) above the open-loop boundary in cases when the model was fixed in roll and up to 23 percent when it was free to roll. Aggressive roll maneuvers with load control were achieved above the flutter boundary. The purpose of this paper is to present the development, validation, and wind-tunnel testing of this multiple-function digital controller system.				
14. SUBJECT TERMS Multiple function digital control; real-time digital control system; multi-input/multi-output (MIMO); Active Flexible Wing (AFW); digital controller system (DCS)			15. NUMBER OF PAGES 10	
			16. PRICE CODE A02	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	