Development, Simulation Validation, and Wind-Tunnel Testing of a Digital Controller System for Flutter Suppression

by

Sherwood T. Hoadley
Aeroservoelasticity Branch
NASA Langley Research Center

Carey S. Buttrill
Aircraft Guidance and Control Branch
NASA Langley Research Center

Sandra M. McGraw
Lockheed Engineering and Sciences Co.

and Jacob A. Houck
Analysis and Simulation Branch
NASA Langley Research Center

presented at

The Fourth NASA Workshop on Computational Control of Flexible Aerospace Systems
Williamsburg, VA
July 11-13, 1990

Introduction

The Active Flexible Wing (AFW) Program (refs.1 and 2) is a cooperative effort between the NASA Langley Research Center and Rockwell International Corporation. The program objective is the validation of analysis and synthesis methodologies through the development of real-time digital multi-input multi-output (MIMO) control laws for a sophisticated aeroelastic wind-tunnel model. This model was tested in the Langley Transonic Dynamics Tunnel during the Fall of 1989.

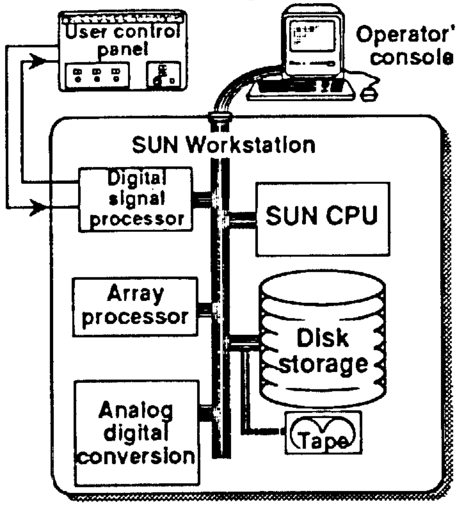
Flutter suppression (FS) is one of the active control concepts being investigated in the AFW Program. The design goal for FS control laws was to increase the passive flutter dynamic pressure 30 percent. In order to meet this goal, the FS control laws had to be capable of suppressing both symmetric and antisymmetric flutter instabilities simultaneously. In addition, the FS control laws had to be practical and of low-order, robust, and capable of real-time execution within a 200 hz. sampling rate.

The purpose of this paper is to present an overview of the development, simulation validation, and wind-tunnel testing of a digital controller system for flutter suppression.

L-90-2912

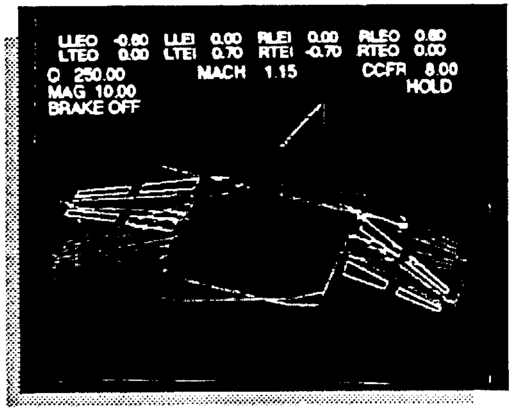
AFW DIGITAL CONTROLLER DESIGNED, ASSEMBLED, CODED, VALIDATED, AND TESTED

Designed



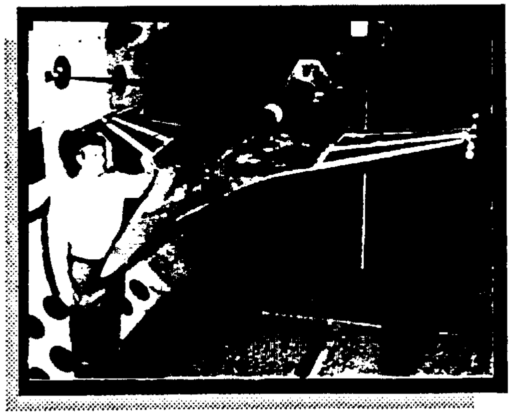
Assembled/Coded

Validated



```
LLEO  -0.80  LLEI  0.00  RLEI  0.00   RLEO  0.80
LTEO   0.00  LTEI  0.70  RTEI  -0.70  RTEO  0.00
Q  250.00          MACH  1.15         CCFR  8.00
MAG  10.00                                HOLD
BRAKE OFF
```

Tested



ORIGINAL PAGE IS
OF POOR QUALITY

AFW Digital Controller Designed, Assembled,Coded, Validated, and Tested

The AFW digital controller was designed, assembled, coded, validated, and tested completely in-house. This represents a "first" for the agency. The accompanying figure illustrates each of these accomplishments.

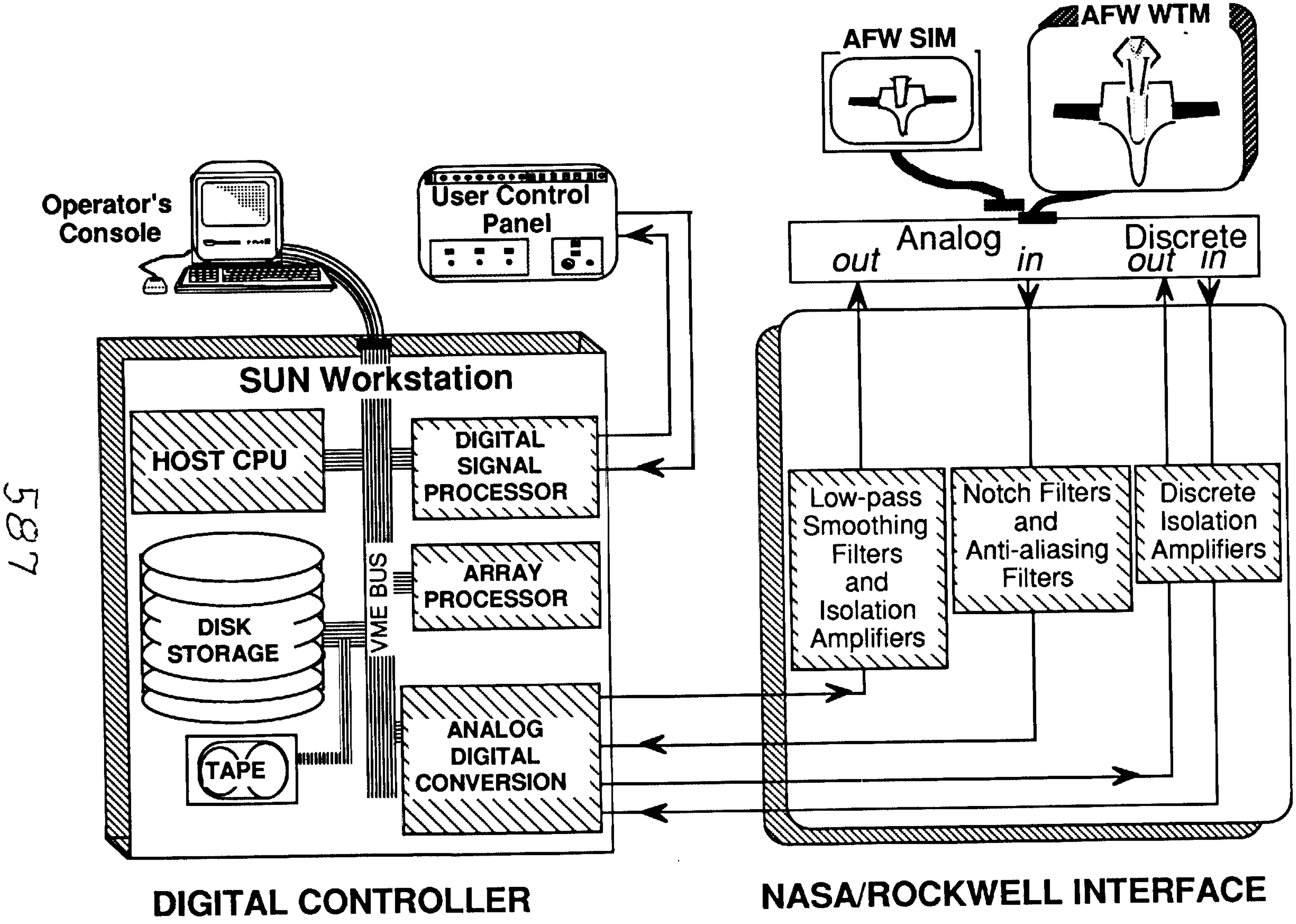
DESIGNED shows schematically a digital controller, comprised of various computers, tape drive, disks, array processor, analog-to-digital conversion boards all residing within the same chassis as a SUN host central processing unit. Design specifications required that the controller have the capability of receiving and providing analog and discrete signals from/to the model and user control panel. The Digital Controller controls the wind-tunnel model by digitizing the incoming sensor signals, processing the currently-implemented control law, and then providing the appropriate control-surface actuator commands to effect these laws.

ASSEMBLED/CODED shows an operator sitting at the SUN 3/160 Workstation. Over 22,000 lines of code in C-language were written during 1989 for the 1989 Wind-Tunnel test . The implementation of control laws into the control computer is a time-critical path leading up to the hot-bench simulation and wind-tunnel tests. This means that the control laws need to be generic in form. The digital controller software can be modified easily and quickly as required, and the generic form of the control systems allows for changes in a design to be implemented easily and reliably.

VALIDATED shows a color-coded three-dimensional wireframe outline of the AFW model generated by the NASA/LaRC Advanced Real-Time Simulation (ARTS) facility. The ARTS facility was used to verify and validate the functionality of the digital controller.

TESTED shows the AFW wind-tunnel model in the NASA/LaRC Transonic Dynamics Tunnel during its October/November 1989 entry. The digital controller operating an FS control law took the AFW wind-tunnel model 24% (in dynamic pressure) above its open-loop boundary.

CONTROL SYSTEM HARDWARE SCHEMATIC



Control System Hardware Schematic

Digital Controller

One of the primary objectives of the AFW Program is to gain practical experience in designing, fabricating, and implementing a real-time MIMO digital controller and in developing the hardware interface between the controller and the actual wind-tunnel model and simulator. The hardware components of the digital controller, on the left side of the figure, show schematically how the host central processing unit (CPU), the disk and tape drives, and the added boards communicate across the VME BUS. During closed-loop operation, the ADC boards convert analog sensor signals to digital data; the DAC boards convert digital actuator commands to analog signals; the host CPU and the user control panel provide user interface to the signal processing board; the signal processing board ("the controller") controls the real-time processing; and the array processing board performs floating-point calculations of the flutter suppression control laws. The entire operation is repeated 200 times a second for real-time operation. To meet these requirements with reasonable resources, a SUN 3/160 workstation driven by a Unix Operating system was selected as the "shell" of the Digital Controller.

NASA/Rockwell Interface

The hardware components of the interface box are shown schematically on the right side of the figure. The interface box contains the analog circuitry for processing the analog signals coming from or going to either the wind-tunnel model or the simulator. The circuitry includes low-pass filters (break frequencies of 1000 hz) to reduce the high-frequency noise and limit voltage spikes, antialiasing filters, and electrical isolation networks. The antialiasing filters are 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 controller can also be filtered through notch filters, specified with each control law to prevent signals with undesired frequencies from being input to the control law. The isolation amplifiers provide optical isolation between two electronic systems.

COMPONENTS OF DIGITAL CONTROLLER

Host Computer
SUN 3/160

Provides user interface to controller
- Initializes options, arrays, control parameters
- Controls saving of data to external files and tapes

Digital Signal Processor
SKY Challenger

Controls real-time processing
- Analog input/output
- Control law execution
- Interface through User Control Panel
- Sending of signals for Controller evaluation

Array Processor
SKY Warrior I

Performs floating-point processing
- Unit conversions and scaling
- Control-law matrix computations

ADC's and DAC's
Data Translation
(2-DT 1401's and 2-DT 1406's)

Performs analog and digital conversions
- Converts analog (sensor) signals to digital data
- Converts digital (actuator) commands to analog signals

User Control Panel
Designed and built in-house

Provides real-time interface to controller
- Allows option selection
- Displays current status of selectable options
- Provides warning for critical conditions

Components of Digital Controller

Besides the host computer, the Digital Controller consists of several special purpose processors linked to the workstation via a data bus. These processors include a digital signal processor, a high speed array processor, and four data translation boards.

The host computer is a SUN 3/160 workstation. It provides the user interface to the digital signal processor board, which is the heart of the real-time digital controller. All user options, control law arrays, control parameters, and excitation definitions are specified through the host user interface. The host downloads signal-processor software and determines and downloads the array-processor command code to implement a currently-selected control law. It allows real-time changes in selection of mode of operation, selection of gains, excitation amplitudes and the control surfaces to be used. The host controls the saving of the digitized data to external files and tapes and provides the display of important parameters such as control-surface deflections, errors between commanded and actual deflections, overall control-law gain, and switch selections.

The digital signal processor (DSP) is a Challenger-I board manufactured by SKY Computers, Inc. and is composed primarily of two TMS 32020 microcomputers and 64K integer words (one word equals 2 bytes) of memory. The DSP is the "real-time digital controller" because it provides the management of all signal processing and scheduling of control laws. As bus master, the DSP controls, directs, and sequences the real-time activities and tasks. It controls all the real-time processing of analog input and output signals. It controls control-law execution by sending commands to the array processor to implement a desired control law and adds digitized model excitations or bias commands to statically position control surfaces. It provides the interface to the user control panel lights and switches and checks for faults; and it sets switches (software flags) for the host computer which specify when blocks of data can be stored and transferred.

The array processor (AP) is a SKY Warrior I board with 16Mbytes of memory which provides the high-speed floating-point arithmetic computations required in executing a particular control law. Included in these computations are unit conversions, scaling, and all matrix computations.

The data translation boards consist of two DT-1401 analog-to-digital converter (ADC) boards and two DT-1406 digital-to-analog converter (DAC) boards manufactured by Data Translation, Inc. They provide all the analog data conversions required between the model and the controller. The ADC's are used to convert the incoming analog sensor signals to digital integer values which can be processed by the DSP. The DAC's are used to convert integer actuator command signals sent by the DSP into analog voltages which are then sent to the control surface actuators.

The user control panel, designed and built in-house by NASA, provides the real-time interface to "the controller". It allows real-time selection of certain options via lighted switches and provides real-time status of various control parameters through status display lights. These switches are simulated in the host interface software for use with the simulator and as a backup.

NASA

90-01865

Langley Research Center
Hampton, Virginia 23665-5225

Digital Controller Simulation Lab

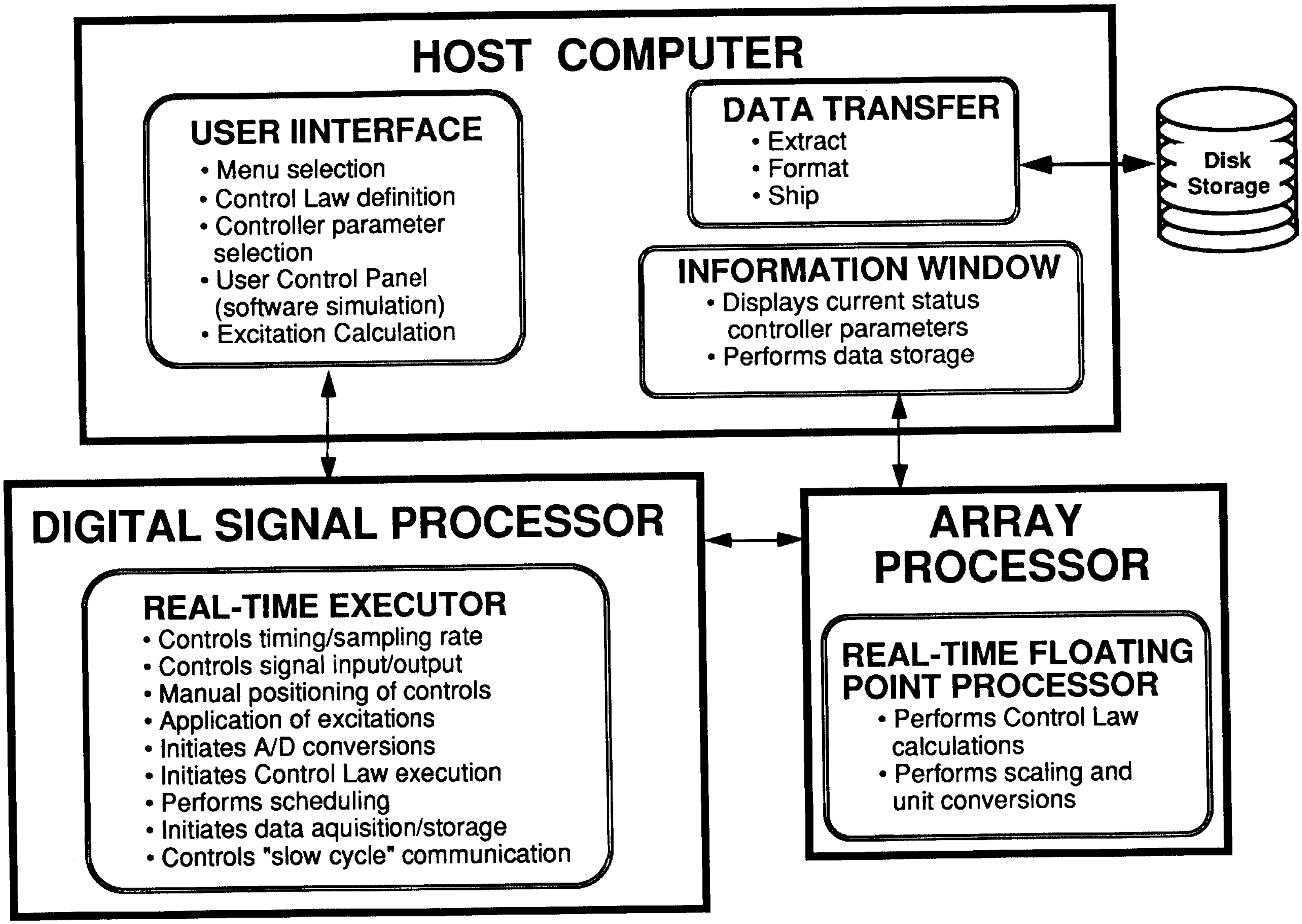
The accompanying figure is a photograph of the Digital Controller Simulation Lab. The components in Lab include:

Two SUN 3/160's (housed in rolling cabinets under the table)
NASA/Rockwell Interface Box with User Control Panel and Patch Box (shown on the extreme left). The User Control Panel and Patch Box are located in the top of the box. The power supply is in the bottom, the anti-aliasing, notch filters, etc. are in the box above the power supply.
Oscilloscope and Simulation Video Monitor with wireframe image displayed (on table)
Operator's console (to the right of the monitor)
Printer

The Patch Box is used to bypass the hardware in the NASA/Rockwell Interface Box when hooked to the simulator when software-implemented anti-aliasing filters are used and when trouble-shooting problems with the hardware.

The oscilloscope is used to monitor specific analog signals for checkout, debugging, and trouble-shooting. The Simulation Video Monitor provides the controller operator with a visual wireframe image of the model which includes the dynamics of the model and the motion of the control surfaces.

DIGITAL CONTROLLER SOFTWARE



HOST COMPUTER

USER IINTERFACE
- Menu selection
- Control Law definition
- Controller parameter selection
- User Control Panel (software simulation)
- Excitation Calculation

DATA TRANSFER
- Extract
- Format
- Ship

INFORMATION WINDOW
- Displays current status controller parameters
- Performs data storage

DIGITAL SIGNAL PROCESSOR

REAL-TIME EXECUTOR
- Controls timing/sampling rate
- Controls signal input/output
- Manual positioning of controls
- Application of excitations
- Initiates A/D conversions
- Initiates Control Law execution
- Performs scheduling
- Initiates data aquisition/storage
- Controls "slow cycle" communication

ARRAY PROCESSOR

REAL-TIME FLOATING POINT PROCESSOR
- Performs Control Law calculations
- Performs scaling and unit conversions

Digital Controller Software

A generic form of the control law function was identified such that one set of software would accommodate a given control law while imposing minimal constraints on the designers. The generic structure allows the designers to choose sensors with options to blend them, freedom of controller order with upper limits, scheduling of controller parameters with respect to dynamic pressure, and selection of various control surfaces with or without distribution of controller outputs to different actuators. Components of the Digital Control System were identified and separate program modules were developed. Their various functions are outlined below. All the Digital Controller software is written in the high level C programming language except for the commands required to perform the actual calculations on the array processor. Operation code command blocks were generated for these.

HOST computer:

There are three primary HOST programs, all of which run simultaneously:

1. HOST INTERFACE providing menus for
   Control Law Definition
   Controller parameter selection
   Calculation of excitation signals for Controller Performance Evaluation
   User Control Panel software simulation
   Calculation of the excitations for Control Law verification
2. DATA TRANSFER providing capability to
   Extract sampled experimental data which is in main memory and/or stored on disk
   Format data for external use
   Ship data to disk, tape, or external computer
3. INFORMATION WINDOW displays current status of controller parameters:
   Actual Control Surface Deflections and percent errors
   Mach and q
   Roll angle, roll-rate, pitch
   Current Sampling Speed
   Current switch selections
   Type, size, amplitude, and frequency of excitation
   Status of data storage
   Control Law inputs and outputs

Digital Signal Processor (DSP):

There is one program residing on the Digital Signal Processing board written in C which controls the real-time execution for

Specifying timing and sampling rate
Controlling sampling of specified input signals
Initiating A|D and D|A conversions of signals
Manually positioning the control surfaces
Sending excitation signals to various control surfaces
Initiating Control Law execution by sending command codes to array processor
Performing scheduling of control laws based on dynamic pressure
Initiating data acquisition and storage
Controlling all the "slow cycle communications between the host INTERFACE program, the Array Processor and the host INFORMATION WINDOW program.

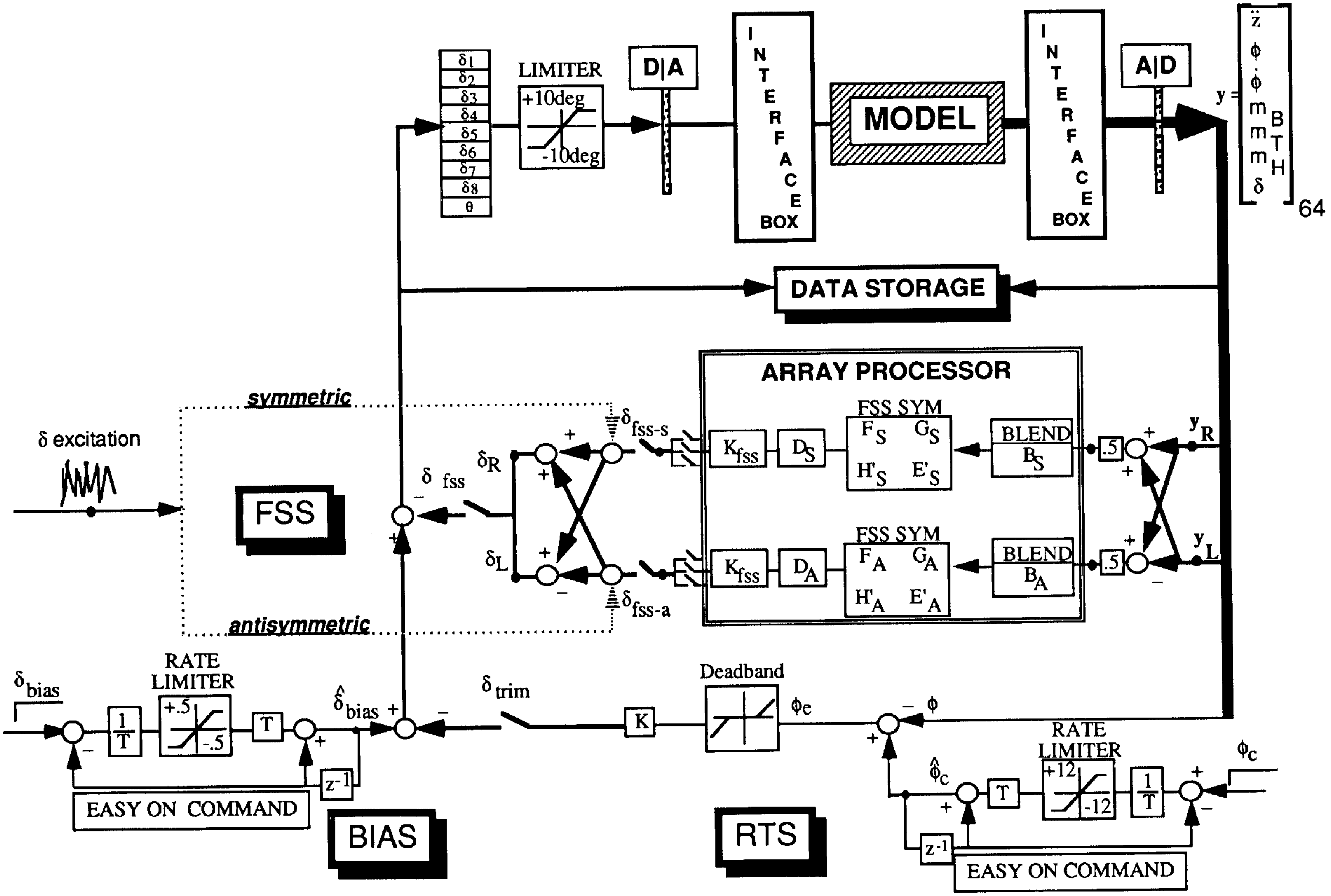
Array Processor (AP):

Control law execution code written using the array processor command language which is stored by the HOST computor in the Digital Signal Processor memory. When control laws are executed, the DSP sends these commands to the array processors. This code performs:

Control Law calculations
Scaling and unit conversions

FLUTTER SUPPRESSION SYSTEM

with ROLL TRIM CONTROL as coded for 1989 Wind-Tunnel test

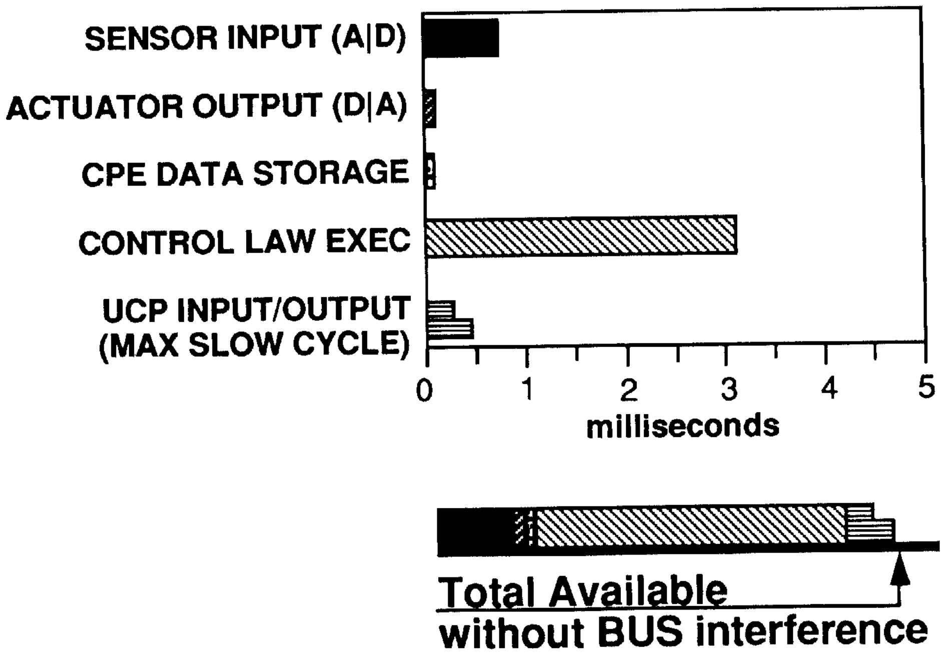


Flutter Suppression System Flow Diagram

This figure is a detailed schematic of the various blocks of code involved in the actual FSS control execution. All blocks of code reside on the Digital Signal Processor (DSP). Commands to operate the Array Processor (AP) are sent from the DSP to the AP during execution each time cycle. Commands for adding bias to a control surface or for performing roll trim are added to the flutter suppression commands. The Roll Trim System feedback is switch selectable. Both the bias commands and the roll trim command are implemented using an "easy on" procedure. The AP performs the actual floating-point calculations of the Control Law matrix operations, indicated by the "boxed-in" area in the figure. The conversion to 16 bit integers and the averaging of the signals for the FSS control law are performed on the DSP using fast masking and binary shifts operations. A subset of the sampled incoming signals and outgoing signals are sent to memory located on the array processor by the DSP.

TIMING SCHEDULE

(200 samples / sec = 5 milliseconds)



Execution Times
for FSS controller
components
as implemented
Nov. 1989

Timing Schedule

The different amounts of time involved in performing the FSS control law functions as delineated in the block diagram in the previous figure are shown in this figure. The first operation is the sampling of the required sensors and shipping them to the array processor memory. This block of code requires approximately 0.68 ms. The actuator commands are then summed and sent to the D|A converters. This requires approximately 0.06 ms. Data storage for Controller Performance Evaluation is then performed. This takes approximately .03 ms. Sending the command blocks of code to the Array Processor to execute the control law requires the most time and is dependent on the number of inputs and outputs, or on the number of states and blended signals. The maximum time required is approximately 3.2ms. There are 10 different "slow-cycle" blocks of code, each executed every 10 iterations (if operating at 200 hz, this translates to 20 times a second). Code for performing communication between different programs or devices is executed during each of these "slow" cycles. Included in this is code to read switch settings from the HOST Interface program and to send parameters to the HOST Information Window program which do not need to be updated every iteration. Types of communications parameters passed are:

mode of operation selected,
excitation and symmetry selected,
desired sampling frequency,
whether or not to perform scheduling of the control law,
whether to open or close the feedback loop,
sampling time left at end of cycle,
whether or not add excitation to actuator commands,
whether or not to save data,
type of excitation selected,
Mach number and dynamic pressure.

The sum of each of these blocks of code must be less than approximately 4.5ms in order to operate without BUS interference. One of the controls laws could not quite meet this goal and had to be slowed by 5% in order to operate without BUS interference.

SUMMARY OF PROBLEMS RESOLVED

- **Real-Time controller operation 3.3 times HOST *required***
  - ~ code development to give system control to DSP
  - ~ supplied software modification so controller operations could be initiated by DSP

- **DSP and AP multiple memory access lock-out to HOST *required***
  - ~ program code specialization on HOST
  - ~ passing of parameters between various HOST programs via the DSP

- **16-bit integer arithmetic on DSP *required***
  - ~ performance of necessary floating-point calculations on AP
  - ~ slow integer division algorithms to be implemented on DSP
  - ~ 'approximate' fast divisions using binary shifts
  - ~ solution of overflow and underflow problems

- **12-bit data translation boards *caused***
  - ~ undesirable overhead for data conversion
  - ~ sign extension problems
  - ~ limiting signals using special code 'compare' operations
  - ~ consideration quantization errors

- **Time budget/BUS interference *resulted in***
  - ~ reduced number of signals saved
  - ~ reduced sampling rate of one control law by 5%
  - ~ removal of some deflection limiting safety features

Summary of Problems Resolved

The primary difficulties involved in designing, coding, and assembling this one-of-a-kind digital controller revolved around four basic problems. The first was that the real-time controller had to operate at 200 hz within a Unix-based operating system which runs at 60 hz. This necessitated obtaining code from SKY Computers, Inc. which gave control of the data bus to the Challenger (DSP) computer which was able to run as fast as calculations would permit. Furthermore, modifications to software library routines supplied by the SKY Computers, Inc. had to be modified so that controller operations could be initiated by the DSP.

The second major problem incurred was that no two host software codes could communicate with the array processor memory or the Challenger memory simultaneously. Since the primary functions of the host computer fell into three categories:

1) selection of control options, definitions of control laws and excitations, and setting of various parameters for the controller;
2) display of current sensor, actuator command, and control law parameters employed by the controller; and
3) controller performance data storage and transfer (refs. 3 and 4),

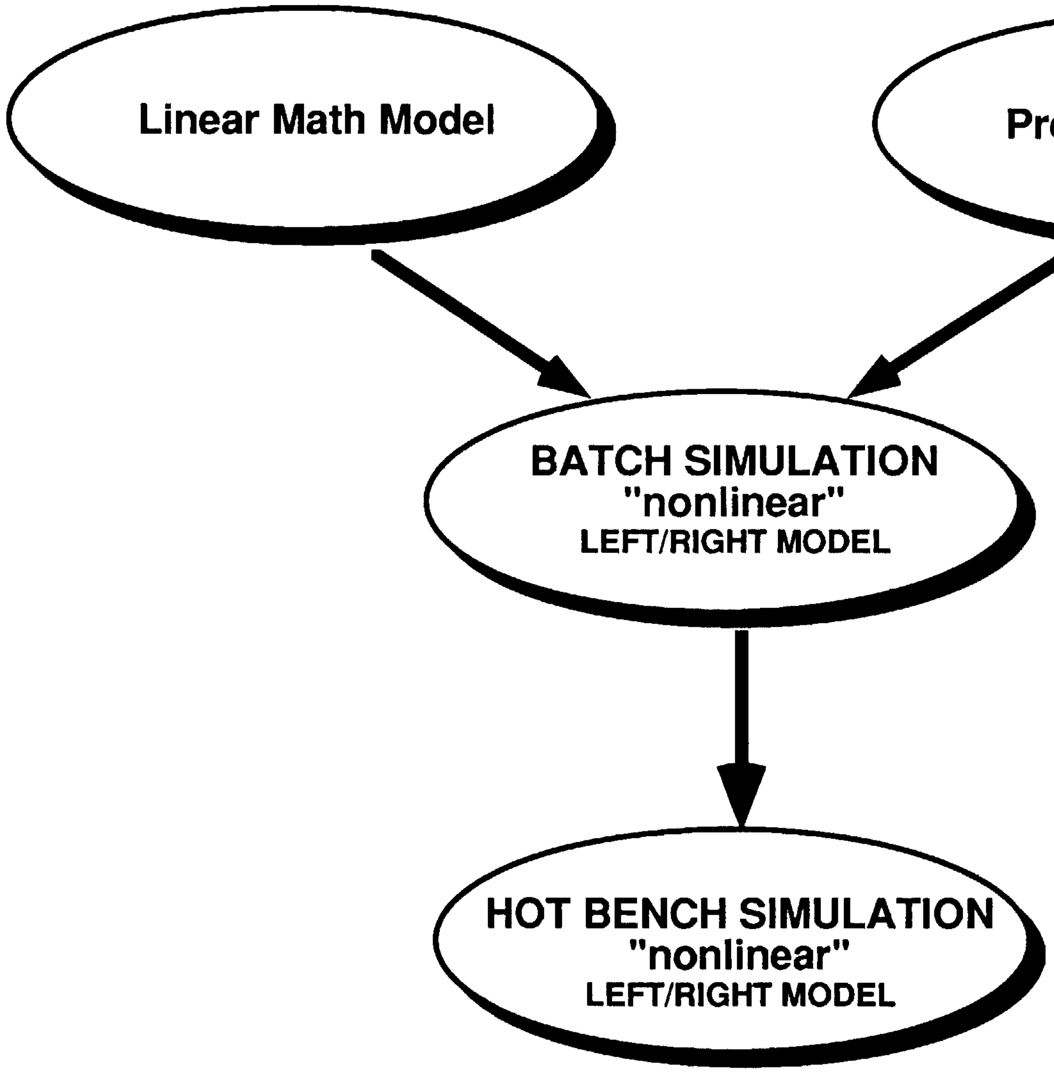
different software packages were developed for each. The interfacing of these various packages with the controller provided an interesting challenge which was met by using the Challenger to pass information between the various host programs.

The third problem resulted from the fact that the DSP was only capable of performing integer arithmetic. It had no floating point registers. This was solved by performing most floating point arithmatic on the AP; however, this entailed transferring data and command codes to the AP. Some processes only required a crude integer division capability which was implemented within the DSP. It would have been preferable to have a DSP which was capable of some floating point arithmetic. The 16-bit address registers along with 32K-byte memory map also caused problems in storing data.

The fourth problem revolved around the fact that the data translation boards which were used only generated 12 bits of resolution. This not only caused some voltage resolution loss, but also necessitated careful handling of sign extensions and truncations from and to 16-bit integer data by the DSP and used a significant portion of the 5ms time budget allowed by the 200 hz sampling rate. It also forced special code to implement voltage limiters on signals which required comparisons of 12-bit 'signed' data with 16-bit compare registers. It would have been highly desirable to have had 16-bit data translation boards.

As a result of time budget and BUS interference problems, some problems were resolved in a fashion which was less than desired. The number of signals to be save had to be reduced, the sampling rate of one control law was reduced by 5 percent, and some deflection-limiting safety features had to be removed.

SIMULATION OVERVIEW



PURPOSE :

- Validate Control Laws "off-line"
- Provide "nonlinear" model for HOT BENCH simulation

PURPOSE :

- Validate functionality of Digital Controller

Simulation Overview

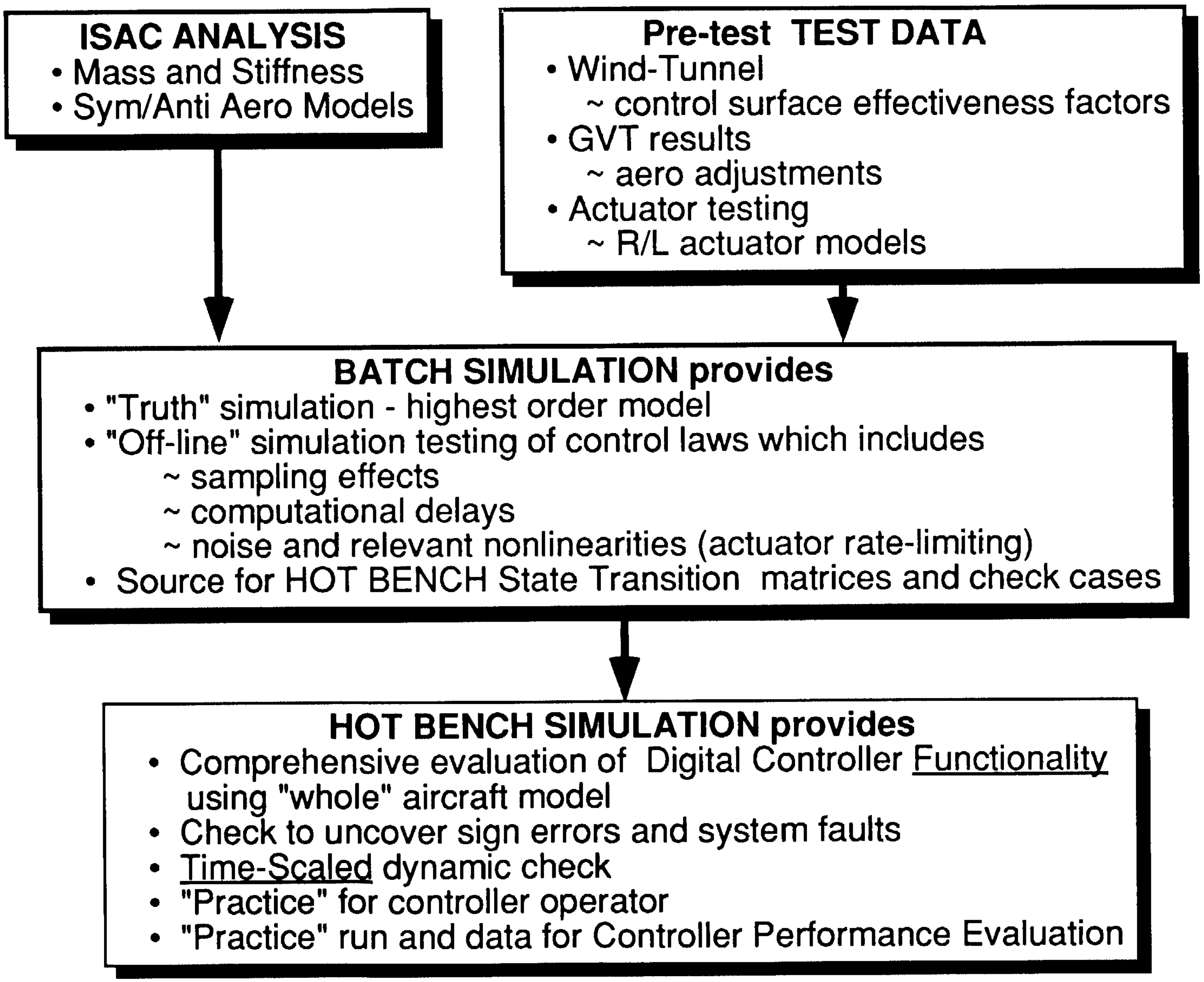
Pre-test analytical open-loop flutter results for the model indicated that the onset of flutter would be very rapid. At test conditions of 0.5 Mach number and 300 psf, the predicted flutter frequency was 7.2 hz. This means that the time-to-double amplitude for flutter was about 0.12 seconds. For closed-loop testing above the open-loop flutter boundary, any digital control system failure might result in very rapid loss of the model before "flutter stopper" mechanisms: releasing the tip-ballast store brake and/or effectively reducing tunnel conditions by opening the tunnel bypass valves - could be effected. Also, the effectiveness of the use of the tip-ballast store as a "flutter stopper" was unknown prior to the test. Because there was a lot of concern for the safety of the wind-tunnel model, it was felt that it was essential to do pre-test verification of the digital controller to gain confidence that the systems functioned properly. This verification is performed by coupling the digital controller to a computer simulation of the model being tested in the tunnel. Because the computer simulation sends signals to and receives signals from the hardware setup, it is referred to as a hot bench simulation (HBS).

The data used to build up the batch and hot bench simulations come from both linear system theory analysis from which a linear math model is developed and from pre-test test data such as experimentally generated actuator transfer functions, aerodynamic correction factors, etc.

These are combined to make up a "nonlinear" BATCH Simulation model which includes the linear model of the plant along with nonlinear rate limiting of the actuators, and represents the whole aircraft, both left and right. There are two primary purposes for the BATCH simulation. First is to provide a mechanism for control law designers to validate their control laws "off-line" from the actual hardware of the HOT BENCH simulator. Computational time delays and sampling effects are included in the model for this purpose. The second purpose is to provide a "nonlinear", whole aircraft model for HOT BENCH simulation.

The purpose of the HOT BENCH simulation is to validate the functionality of the Digital Controller.

SIMULATION DETAILS



ISAC ANALYSIS
- Mass and Stiffness
- Sym/Anti Aero Models

Pre-test TEST DATA
- Wind-Tunnel
  ~ control surface effectiveness factors
- GVT results
  ~ aero adjustments
- Actuator testing
  ~ R/L actuator models

BATCH SIMULATION provides
- "Truth" simulation - highest order model
- "Off-line" simulation testing of control laws which includes
  ~ sampling effects
  ~ computational delays
  ~ noise and relevant nonlinearities (actuator rate-limiting)
- Source for HOT BENCH State Transition matrices and check cases

HOT BENCH SIMULATION provides
- Comprehensive evaluation of Digital Controller <u>Functionality</u> using "whole" aircraft model
- Check to uncover sign errors and system faults
- <u>Time-Scaled</u> dynamic check
- "Practice" for controller operator
- "Practice" run and data for Controller Performance Evaluation

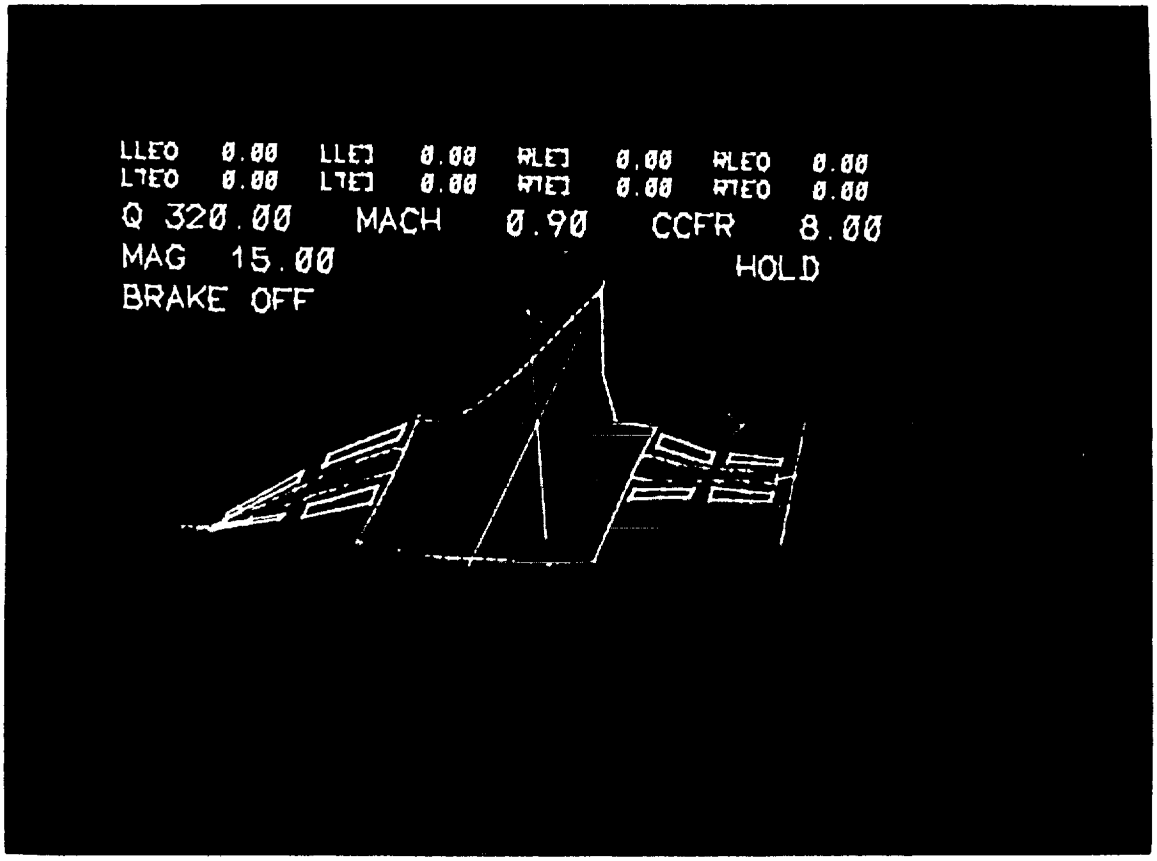
Simulation Details

The data used to build up the batch and hot bench simulations come from three sources, (1) a collection of aeroservoelastic analysis programs known as ISAC , ref. 5, (2) some vibration codes to calculate natural frequencies, and (3), measured data. From ISAC come the generalized mass and stiffness matrices, and the generalized aerodynamic forces (GAF's). The generalized aerodynamic forces are calculated in ISAC by linear lifting surface theory. Complex-valued matrices of GAF's are produced as tabular functions of reduced frequency ($k = b\omega/v$). These tabulated aerodynamic matrices can be approximated in ISAC as rational functions of the parameter "p" where "p"="jk" and j = sqrt(-1). These rational function approximations (RFA's) can be formulated in a variety of ways. A good summary of the various methods of forming RFA's can be found in refs 6 and 7.

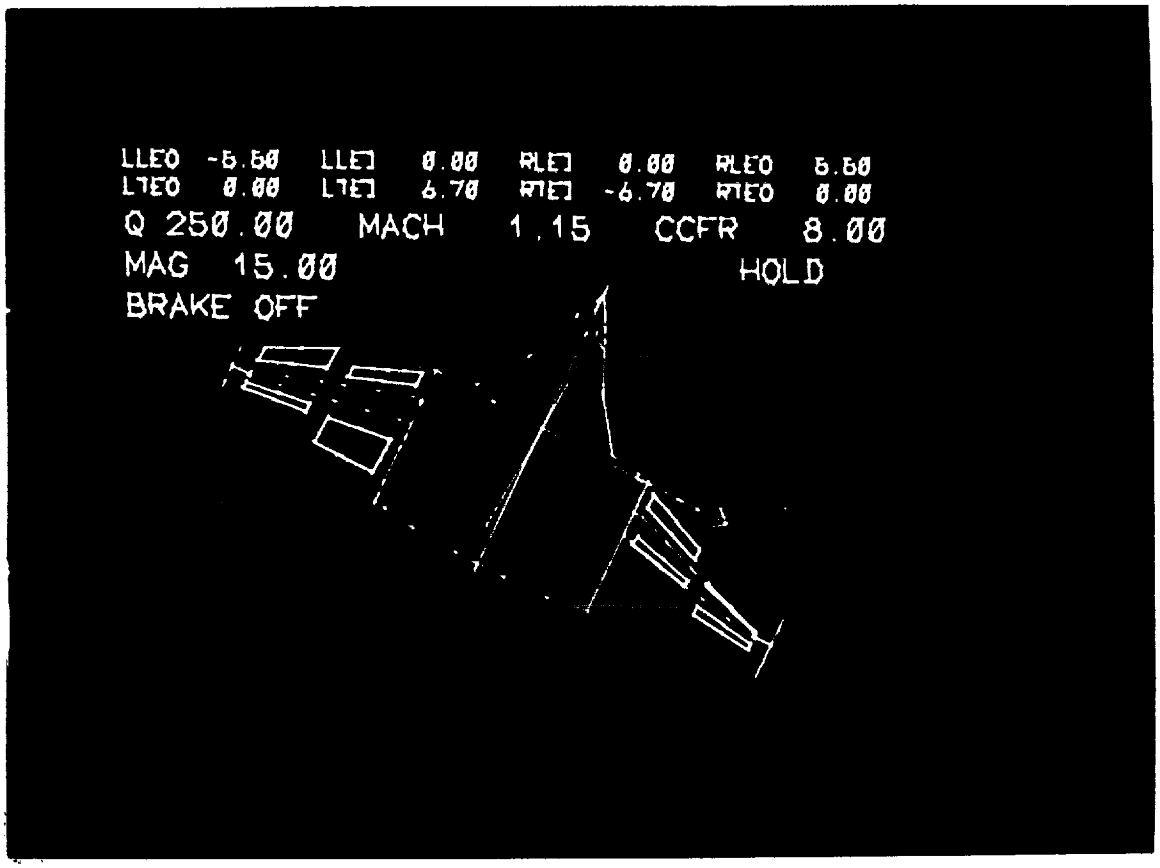
The other source of data for the batch simulation is experimental. The elastic mode frequencies resulting from a vibration analysis are replaced with measured GVT frequencies where applicable. The actuator transfer functions are the result of fitting measured frequency response data with third order transfer functions. As a result, the right and left actuator models are not equal for actuator pairs. These actuator models are implemented in the simulations. In addition, the simulated response of the actuators is rate-limited according the published specifications. Extensive static data was taken in the last wind-tunnel entry of the roll moments and lift force produced by control surface deflections. When roll (lift) per unit deflection, both measured and predicted, are plotted as functions of dynamic pressure, they are not the same. The predictions come from lifting surface theory and the lack of agreement is no surprise. By judicious use of "effectiveness factors", the predicted roll(lift) can be brought into agreement with the measured data. Two points of interest where agreement in predicted and measured control effectiveness is sought is (1), the limiting value as dynamic pressure goes to zero and, (2), the dynamic pressure where control surface deflection produces no change in roll(lift) due to elastic deformation of the wing, the reversal point. These effectiveness factors are implemented in the simulations.

Both the batch and hot bench simulations are "whole" aircraft models. The inputs are right and left actuators and the outputs are right and left measurements. The GAF's, mass and stiffness matrices are in terms of symmetric and anti-symmetric modes, which are combined in the simulation models.

The batch simulation is intended to be the "truth" (or most correct) model. The hot bench simulation model will typically be simplified in some fashion to reduce the required computational time. Currently the hot bench simulation is the same order as the batch simulation, but it is anticipated that as the batch simulation is updated, the order will increase from 115 to 196. Various methods of model reduction are being examined to create a reduced order hot bench simulation.



```
LLEO   0.00   LLEI   0.00   RLEI   0.00   RLEO   0.00
LTEO   0.00   LTEI   0.00   RTEI   0.00   RTEO   0.00
Q 320.00    MACH     0.90    CCFR    8.00
MAG   15.00                          HOLD
BRAKE OFF
```

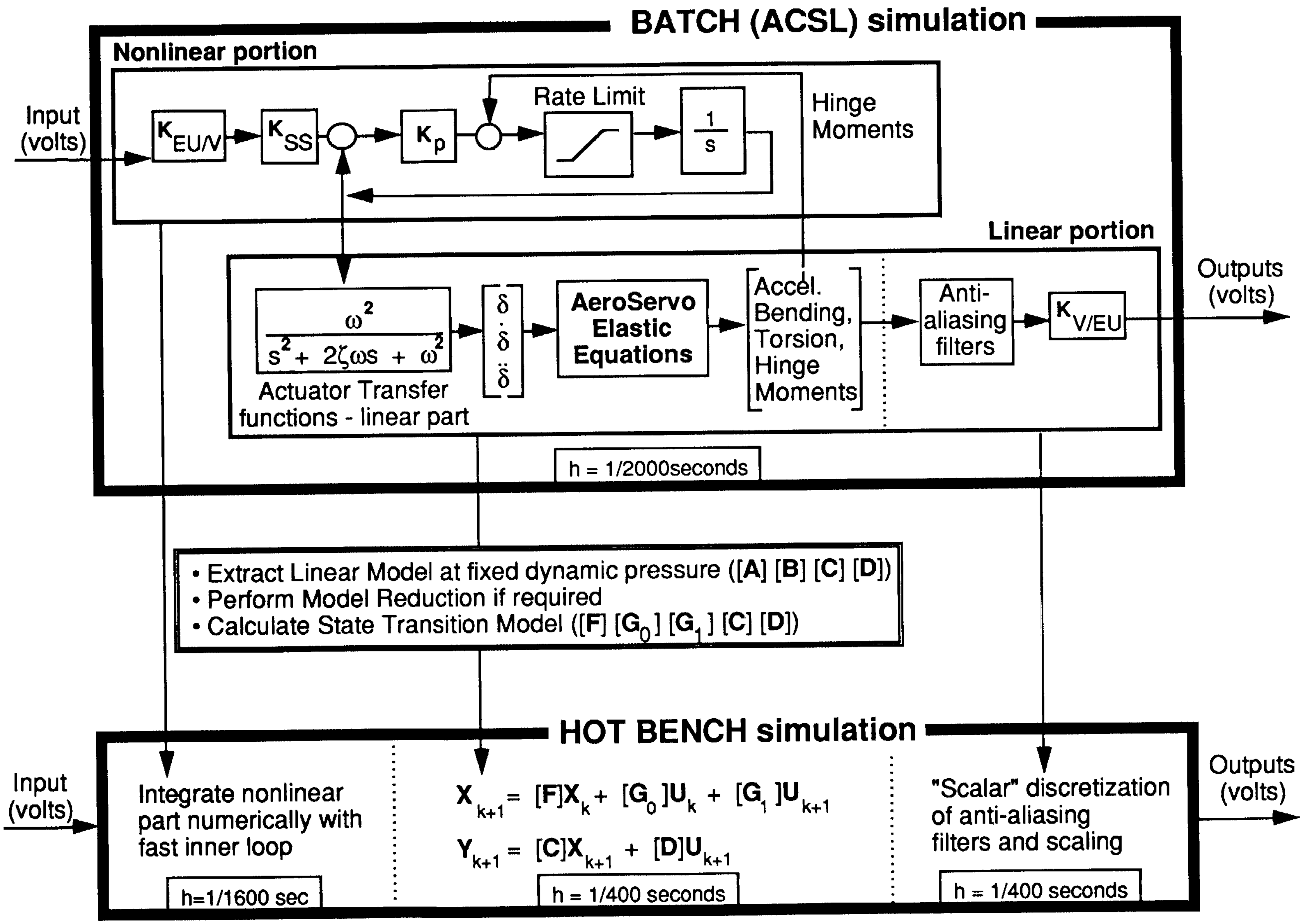


```
LLEO  -6.60   LLEI   0.00   RLEI   0.00   RLEO   6.60
LTEO   0.00   LTEI   6.70   RTEI  -6.70   RTEO   0.00
Q 250.00    MACH     1.15    CCFR    8.00
MAG   15.00                          HOLD
BRAKE OFF
```

Wireframe Simulation Image

An ADAGE (Eagle1000) graphics computer generates a color-coded, three-dimensional articulating wireframe image of the flexing AFW model. The display presents model pitch, roll and yaw, control surface deflections and total model deformation which can be magnified for visual clarity. A blue shadow wireframe of the undeformed model is drawn so that deformations are more easily seen. Examples of both an aerodynamically deforming model and a flexible/rigid rolling model are displayed in this figure. The undeformed model can be seen as a horizontal, undistorted "shadow" image.

FROM BATCH TO HOT BENCH SIMULATION



BATCH (ACSL) simulation

Nonlinear portion

Input (volts)

$K_{EU/V}$ K_{SS} K_p

Rate Limit

$\frac{1}{s}$

Hinge Moments

Linear portion

$$\frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

Actuator Transfer functions - linear part

$\begin{bmatrix} \delta \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix}$

AeroServo Elastic Equations

Accel. Bending, Torsion, Hinge Moments

Anti-aliasing filters

$K_{V/EU}$

Outputs (volts)

h = 1/2000seconds

- Extract Linear Model at fixed dynamic pressure ([A] [B] [C] [D])
- Perform Model Reduction if required
- Calculate State Transition Model ([F] $[G_0]$ $[G_1]$ [C] [D])

HOT BENCH simulation

Input (volts)

Integrate nonlinear part numerically with fast inner loop

h=1/1600 sec

$$X_{k+1} = [F]X_k + [G_0]U_k + [G_1]U_{k+1}$$

$$Y_{k+1} = [C]X_{k+1} + [D]U_{k+1}$$

h = 1/400 seconds

"Scalar" discretization of anti-aliasing filters and scaling

h = 1/400 seconds

Outputs (volts)

From Batch to Hot Bench Simulation

The batch simulation is implemented as if it were a nonlinear system. Dynamic pressure is a parameter that can be varied during a run. All the states are collected in a large state vector and integrated with a Runge-Kutta second order integration scheme. The integration step used in the batch simulation is 1/2000 seconds. Sensitivity studies indicate a small degradation in accuracy with an integration step of 1/1600 seconds and significant degradation for larger steps.

If dynamic pressure is held fixed, the batch simulation is linear except for the rate limit imposed on the single pole portion of the actuator transfer functions. There are eight actuators modeled with third order transfer functions. The second-order part of the third-order actuator models can be lumped with the remaining linear dynamics. The aeroservoelastic (ASE) equations are highly coupled and currently contain 49 states, broken down as follows:

8	symmetric elastic mode positions
8	symmetric elastic mode velocities
8	symmetric aerodynamic lag states (1 lag formulation)
2	symmetric gust states (modified Dryden)
7	anti-symmetric elastic mode positions
7	anti-symmetric elastic mode velocities
7	anti-symmetric aerodynamic lag states (1 lag formulation)
2	anti-symmetric gust states (modified Dryden)
49	

Together with the 16 states associated with the second-order part of eight actuator models, a coupled linear system of 65 states, 10 inputs (8 actuator and 2 noise), and 40 outputs can be extracted from the "linear" portion of the batch simulation. Order reduction techniques can be applied to this dynamic system followed by conversion to a state transition model based on an integration step of 1/400 seconds.

The anti-aliasing filters are applied to each output signal and result in a diagonal system. The anti-aliasing filters are therefore not lumped with the actuator-ASE coupled system to avoid making full matrix-multiply operations if they can be avoided. The anti-aliasing filter dynamics are digitized in a sequential scalar manner in the hot bench simulation with an integration step of 1/400 seconds.

The nonlinear portion is integrated numerically with an integration step of 1/1600 seconds. Four integration steps are made to predict the value of the input to the coupled linear system at time (k+1)h where h = 1/400 seconds. Since input to the coupled linear system at time (k+1)h is now available, a trapezoidal state transition scheme can be employed. Let $\{u_k\}$ denote the quantity $\{u(t=kh)\}$ where the vector $\{u\}$ is a function of time, t. Given the linear dynamic system

$$\{\dot{x}\} = [A]\{x\} + [B]\{u\}$$

if the ramp input signal ,

$$\{u(t)\} = \{u_k\} + (t-kh)\frac{\{u_{k+1}\} - \{u_k\}}{h}$$

over the interval

$$kh \leq t < (k+1)h$$

then the following exact solution for $\{x\}$ at time $t = (k+1)h$ exists:

$$\{x_{k+1}\} = [F]\{x_k\} + [G_0]\{u_k\} + [G_1]\{u_{k+1}\}$$

where,

$$[F] = e^{[A]h}$$

$$[G_0] = (e^{[A]h} - e^{[A]h}[A]^{-1}/h + [A]^{-1}/h)\,[A]^{-1}[B]$$

$$[G_1] = \left(e^{[A]h}[A]^{-1}/h \; - \; [I] \; - [A]^{-1}/h \right) [A]^{-1}[B]$$

The above equations are not valid if [A] is not invertible (i.e. if [A] has eigenvalues equal to 0, such as occurs with rigid-body modes); however, using Taylor series identities, the equations for $[G_0]$ and $[G_1]$ can be put into a form that can be calculated:

$$[G_0] = \left(\sum_{p=2}^{\infty} \frac{1}{p!} (-1)^p \, ([A]h)^{p-2} \right) h \, e^{[A]h} \, [B]$$

$$[G_1] = \left(\sum_{p=2}^{\infty} \frac{1}{p!} ([A]h)^{p-2} \right) h \, [B]$$

The matrices $[G_0]$ and $[G_1]$ are calculated by summing the above series until the next term is under some tolerance. This Taylor series approximation approach may not work in general if [A] is ill conditioned.

It can be shown that integrating the nonlinear portion of the equations with an integration step of 1/1600 and the linear equations in the fashion above with an integration step of 1/400 gives only a small degradation in the accuracy of the solutions to the differential equations of motion.

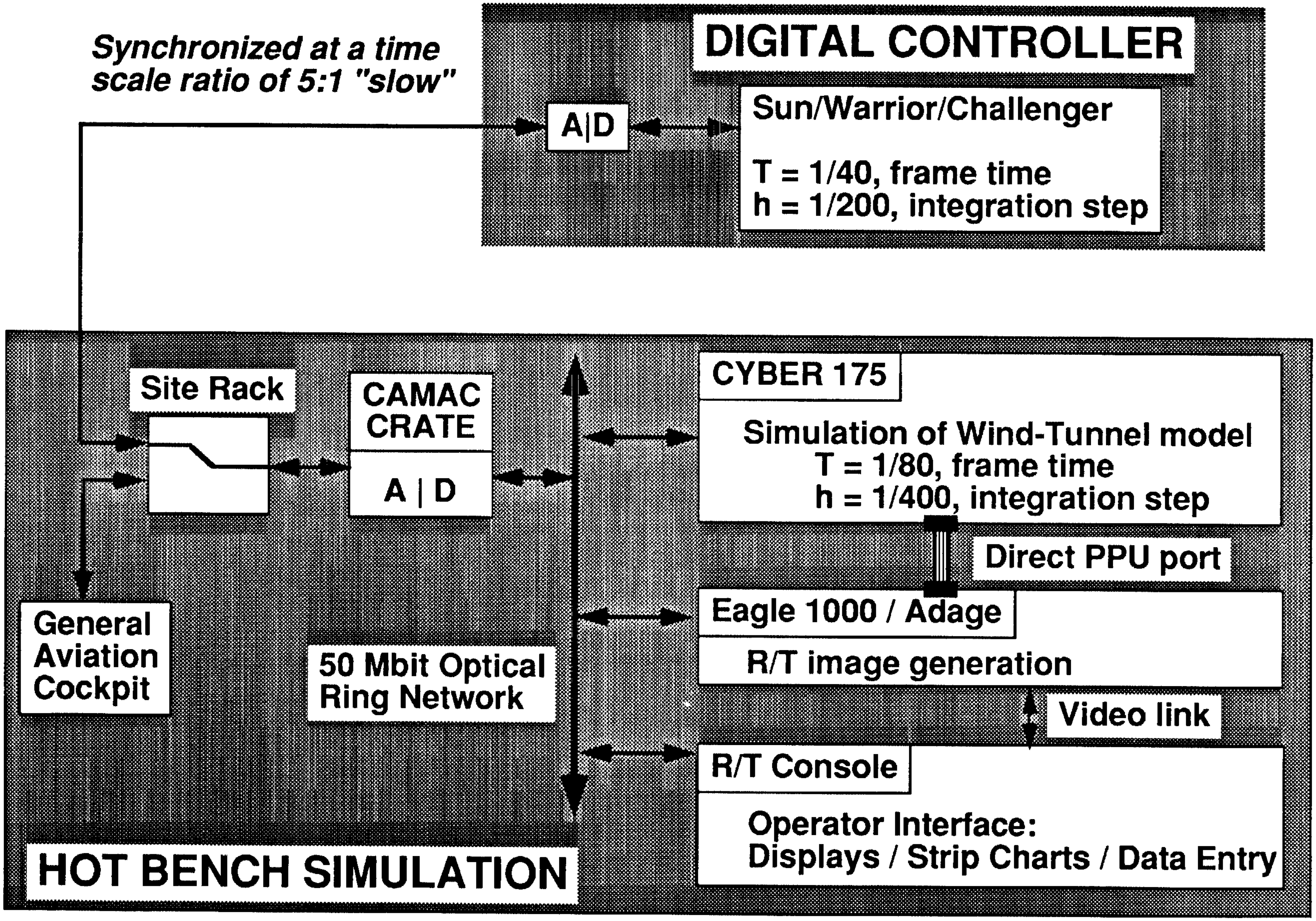
The issue of time scaling deserves some explanation. The Cyber can only integrate the equations of motion of the plant at 80 frames per second without losing time synchronization. This means that implementation of the hot bench simulation in a similar fashion as the batch simulation creates an unacceptably slow time scale ratio (2000:80 = 25:1) due to the 80 frames/second rate of the Cyber (on which the hot bench simulation runs). Since the linear simulation equations can be integrated with a time step of 1/400 seconds, this means the Cyber simulation is only running 5:1 (400:80) slow. The control laws are digitized for an integration time step of 1/200 seconds. Thus the digital controller must be clocked at 40 frames per second (200/5) to be dynamically equivalent. Since there is no human operator in the loop, a slow time scale can be accommodated.

Currently no model reduction is being performed on the extracted 65 state model. This will change in the near future. The ASE dynamic models being formulated as a result of the Fall 1989 tunnel entry will have the following set of states:

10	symmetric elastic mode positions
10	symmetric elastic mode velocities
40	symmetric aerodynamic lag states (4 lag formulation)
2	symmetric gust states (modified Dryden)
11	anti-symmetric elastic mode positions
11	anti-symmetric elastic mode velocities
44	anti-symmetric aerodynamic lag states (4 lag formulation)
2	anti-symmetric gust states (modified Dryden)
130	

Combined with 16 actuator states, this leads to a 196 state coupled linear system to be integrated with matrix state transition equations. Some model reduction will be necessary to retain the current 5:1 time scale ratio and is being investigated.

SCHEMATIC OF HOT BENCH SIMULATION



Synchronized at a time scale ratio of 5:1 "slow"

DIGITAL CONTROLLER

A|D

Sun/Warrior/Challenger

T = 1/40, frame time
h = 1/200, integration step

Site Rack

CAMAC CRATE

A | D

General Aviation Cockpit

50 Mbit Optical Ring Network

CYBER 175

Simulation of Wind-Tunnel model
T = 1/80, frame time
h = 1/400, integration step

Direct PPU port

Eagle 1000 / Adage

R/T image generation

Video link

R/T Console

Operator Interface:
Displays / Strip Charts / Data Entry

HOT BENCH SIMULATION

Schematic of Hot Bench Simulation

The digital controller communicates with the central Langley Advanced Real Time Simulation (ARTS) System via analog lines that are connected to a site rack (ref.8). The ARTS consists of two Cyber 175 computers connected to an array of simulation sites by means of a 50-megabit/second fiber optic digital data network called Computed Automated Measurement and Control (CAMAC). The CAMAC interface converts outgoing Cyber 175 digital signals to analog signals and incoming analog signals to digital signals for the Cyber 175. The simulation of the AFW wind-tunnel model consists of: (1), an engineering console that controls the simulation, (2), a Cyber 175 wherein the equations of motion are integrated, and (3), an ADAGE graphics computer that generates a color-coded, three-dimensional articulating wireframe image of the flexing AFW model.

Both the Cyber and the Adage are dated and are in the process of being replaced. The Cyber communicates directly with the ADAGE graphics computer through a PPU port on the Cyber. The ADAGE will soon be replaced with an Eagle 1000. The Eagle will communicate over the the 50 Mbit optical ring just as the Cyber and the real time console do.

SUMMARY OF DIGITAL CONTROLLER ACHIEVEMENTS

DEVELOPED VERSATILE SYSTEM WHICH

- Operates at 200 Hz within a 60 Hz operating system environment
- Allows simultaneous FSS and data aquisition, storage, and transfer
- Allows flexibility in control law implementation
- Coordinates and synchronizes the operation of three different computers

ALLOWED SUCCESSFUL DEMONSTRATION OF

- Active flutter suppression
- Near real-time Controller Performance Evaluation

Digital Controller Achievements

An extremely versatile system has been developed which operates at 200 hz. within a 60 hz operating system environment. It allows simultaneous Flutter suppression and data acquisition, storage, and transfer. Normally, a controller system would not be expected to also provide for data acquisition. It allows not only flexibility in control law implementation both in the number of sensors and actuators employed, but also in the number of states, and the selection of sensors. It coordinates and synchronizes the operation of three different computers: a host SUN 3/160, a Digital Signal Processor, and an Array Processor.

Most importantly, it allowed the successful demonstration of active flutter suppression, and provided the data for near real-time controller performance evaluation.

References

1. Noll, Thomas E.; Perry, Boyd III; Tiffany, Sherwood Hoadley; et al: *Aeroservoelastic Wind-Tunnel Investigations Using the Active Flexible Wing Model - Status and Recent Accomplishments*, AIAA Paper No. 89-1168-CP, April 1989. Also published as NASA TM-101570, April 1989.

2. Perry, Boyd III; Mukhopadhyay, Vivek; Hoadley, Sherwood T.; et al: *Digital-Flutter-Suppression-System Investigations for the Active Flexible Wing Wind-Tunnel Model*, AIAA Paper No.90-1074-CP March 1990. Also published as NASA TM-102618, March 1990.

3. Pototzky,A.S.; Wieseman*,C.D.; Hoadley, S.T.; Mukhopadhyay, V.: *Development and Testing of Controller Performance Evasluation Methodogy for Multi-Input/Multi-Output Digital Control Systems.* Fourth Workshop on Computational Control of Flexible Aerospace Systems, July 11-13, 1990.

4. Pototzky*,A.S.; Wieseman,C.D.; Hoadley, S.T.; Mukhopadhyay, V.: *Development and Testing of Methodology for Evaluating the Performance of Multi-Input/Multi-Output Digital Control Systems.* Paper presented at AIAA Guidance, Navigation, and Control Conference, AIAA Paper No. 90-3501, August 20-22, 1990, Portland, Oregon.

5. Peele, E.L. and Adams, W.M., Jr.: *A Digital Program for Calculating the Interaction Between Flexible Structures, Unsteady Aerodynamics, and Active Controls*, NASA TM-80040, January 1979.

6. Tiffany*, S. H.; and Adams, W. M., Jr.: *Nonlinear Programming Extensions to Rational Function Approximations of Unsteady Aerodynamics*, AIAA Paper 87-0854-CP, April 1987.

7. Tiffany, Sherwood.H. and Adams, William.M.Jr.: *Nonlinear Programming Extensions to Rational Function Approximation Methods for Unsteady Aerodynamic Forces Which Allow Variable Selection of Constraints,* NASA TP 2776, May 1988.

8. Crawford, Daniel J.; Cleveland, Jeff I. II; and Staib, Richard O.: *The Langley Advanced Real-Time Simulation (ARTS) System Status Report*, AIAA-88-4595-CP, September 1988.

*Presenter of paper