MEASUREMENTS AND EFFECTS OF TRANSPORT DELAYS IN A STATE-OF-THE-ART F-16C FLIGHT SIMULATOR

Capt Scott J. Horowitz, PhD
Chief Engineering Evaluation Pilot
Air Force Human Resources Laboratory
Operations Training Division
Williams AFB AZ 85240-6457

ABSTRACT

In recent years the military community has developed advanced simulators for high performance fighter-type aircraft. These devices not only simulate high performance aircraft but also complex tasks such as air to air combat, aerial refueling, air to ground combat, and formation flying. With the increases in the sophistication of these simulators has come a corresponding increase in computational complexity. This complexity has negated the effects of higher computational speeds available in today's computers; thus the transport delays have remained essentially constant. What has not remained constant, however, are the effects these transport delays have on the training effectiveness of these complex simulators. Since these modern simulators tend to be very complex in nature and consist of many computers interfaced with each other, the determination and measurements of the transport delays is often difficult. The effects these delays have on the simulation of a high performance fighter-type aircraft are also difficult to determine. The Air Force Human Resources Laboratory, Operations Training Division (AFHRL/OT) is currently completing the development of a new F-16C simulator with full field-of-view visual display and no motion system. This paper describes the methods utilized to measure the transport delays that exist in this system and some of their effects on the training effectiveness of the simulation.

INTRODUCTION

To perform research involving transport delay, it is first convenient to define a few terms. Refer to Figure 1 which shows graphically the definitions of transport delay, equivalent time delay, and effective time delay. These delays are shown as a system's response to a step input and have the following physical significance: (1) Transport Delay - this is the type of delay that is associated with pure delay where the response is zero until the end of the delay period, sometimes referred to as "time to wiggle"; (2) Equivalent Time Delay - this delay is determined by assuming a functional form of the system response (usually a *simpler model of the actual system*) and determining the delay due to the term exp(1/t); and (3) Effective Time Delay - this delay is achieved graphically and is defined as the intercept on the time axis of the maximum slope of the system's response. For the simulator experiments conducted at AFHRL/OT the type of delays investigated were pure transport delays.

The objectives of this experiment were: (1) develop a method for easily measuring the transport delays of an existing simulator system, (2) measure the delays due only to the computer hardware, (3) measure the delays due to software and hardware combinations, and (4) determine the effects on training in the simulator, if any, of transport delay (refer to Figure 2). To accomplish these objectives, a three step process was outlined as follows: (1) design and construct hardware to interface with the cockpit and aero computations (basic side), and interface with the visual computer and display system (visual side) to provide a recording of inputs and responses, (2) analyze the delays expected using a system block diagram and equipment performance specifications and compare to the data collected in step 1, and (3) compare objective flight evaluations of the simulator using data collected during F-16C transition training conducted at AFHRL/OT.

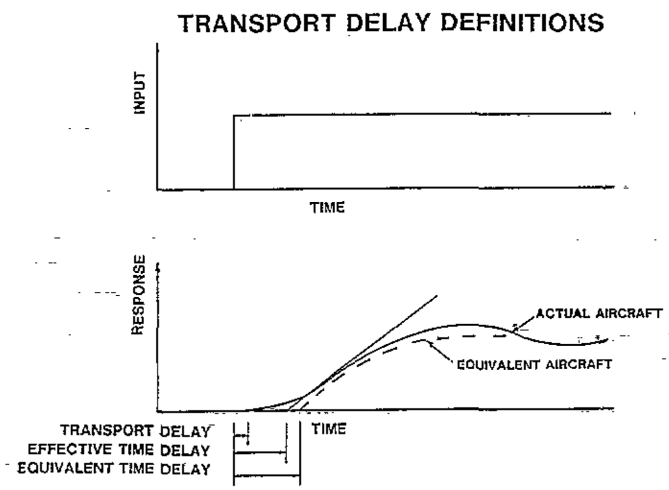


Figure 1

EXPERIMENTAL SETUP

The interface designed and constructed for the basic side integration was relatively straightforward, due in part to the fact that the F-16C is a fly-by-wire aircraft and all information is available in analog or digital form.

upon predetermined ECM effectiveness criteria such as jamming techniques, J/S ratio, EXCM dispense type, EXCM dispense timing and penetrator maneuvers.

As shown in Figure 5, the threat data base simulation architecture is partitioned into two data structure types: scenario and characteristics. Scenario data provides the training mission related data including system/site assignments, weapon release doctrine, and initial radar start-up conditions. Characteristics data defines the non-mission related data parameters that define the rest of the EW characteristics needed to simulate a particular threat.

The scenario data structure is partitioned into ground site and AI platform data. Ground site data definitions include site NATO name, site position (latitude, longitude, altitude), site networked designations, auxiliary support radar identification, and initial radar operating conditions such as RF, PRF, mode antenna scan, pulse width, activation range, and deactivation range. AI platform data defines the AI NATO name, radar system assignment, weapon types and quantity on-board, release doctrine, AI attack profile type, reattack options, and the AI initial conditions, such as range and angle to penetrator, heading, airspeed, activation time, and deactivation time.

The characteristics data structure is further decomposed into electromagnetic, weapon system, and miscellaneous data.

The electromagnetic data describes the radar system emitter characteristics. Definitions for the four major threat categories (EW, SAM, AI and AAA) include parameters for type, beam, mode, ECCM, and radar profile. Type parameters include quantities for beams and modes, delay times for sector and occulting, and tracking angle limits. Beam parameters include transmitter power, antenna gain, antenna beamwidths, boresight angles, RF limits and separation, and detection range. Mode parameters include antenna scan definition, sidelobe definition, and PRF limits and agility. ECCM parameters include definition for frequency change, frequency slide without dummy load, and frequency slide with dummy load. Radar profile parameters include definition for initial mode, next higher and next lower modes, mode durations, and mode change delay times. Definitions for auxiliary emitters (IFF, command guidance, and communications data link) include a subset (beam and mode parameters) of those described above.

The weapon system data describes the weapon system characteristics for SAM's, AI and AAA. Weapons characteristics address radar missiles, Infrared (IR) missiles, and guns. Profile data includes definition of the AI attack and reattack parameters. The system characteristic definition includes weapon loads, weapon launch doctrine, radar type, IFF type, system tracking parameters, and non-radar sensor identification. The missile characteristic definitions include seeker type and sensitivity, launch contours, and launch angular limits. Gun characteristic definitions include type, range limits, and elevation limits. The profile data definition provides the initial conditions for position, heading, gravity-limits and speed ratios between the AI and the penetrator.

The miscellaneous data structure describes simulation characteristics for IR and Electro-optical systems. These parameters include system name, sensitivity, angular limits, detection range, and day/night capability.

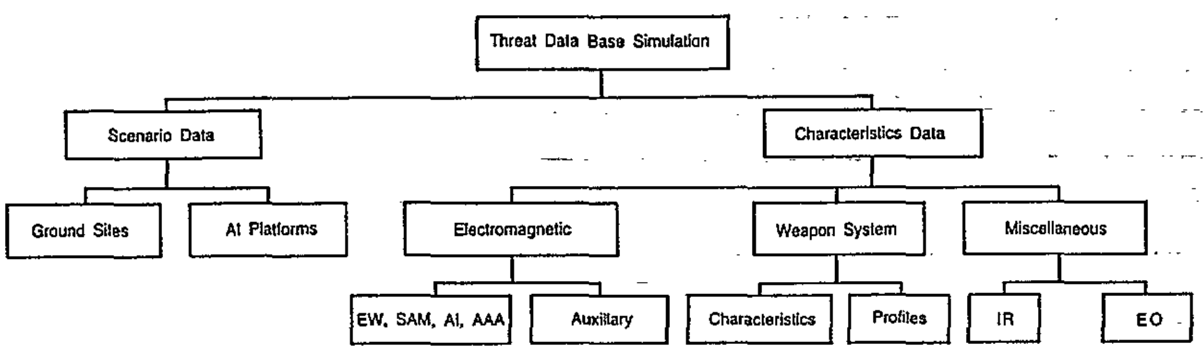
CONCLUSION

A threat environment software-intensive simulation has been developed by BMAC that adheres to training objectives and provides an optimal blend of fidelity, simplicity, and flexibility. Through attention to top-down, modular software and database architectures in the threat design process, capability is inherent to increase fidelity, if necessary, for a project application where crew member equipment permits more detailed observation or where enhancements occur in future threat capabilities.

ABOUT THE AUTHOR

Mr. Jerry W. Denver is manager of Defensive Training Simulation Technology for BMAC's Simulation and Training Systems. He is responsible for development of software and database requirements and performance of software system integration for electronic warfare simulations. These simulations have been applied to BMAC's B-52 and B-1B Weapon System Trainer programs. He holds a B. S. Degree in Electronic Engineering from Wichita State University in 1972.

FIGURE 5. THREAT DATA BASE SIMULATIONS STRUCTURE



SIMULATOR FIDELITY

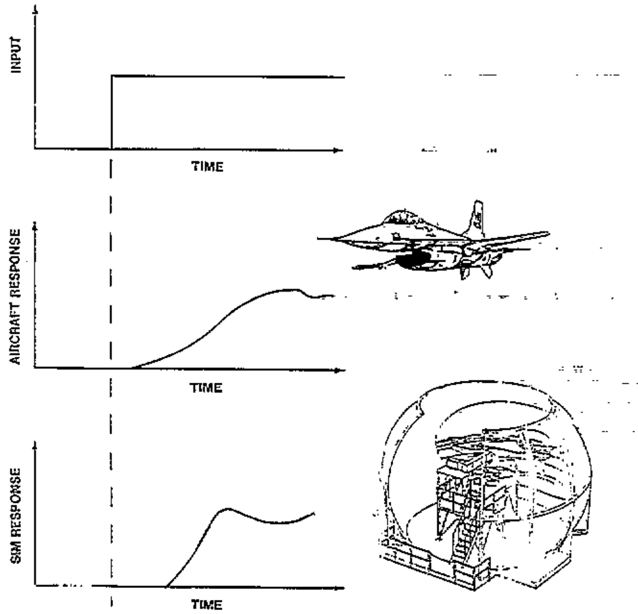


Figure 2

An interface card was designed and inserted in the system as seen in Figure 3.

EXPERIMENTAL SETUP

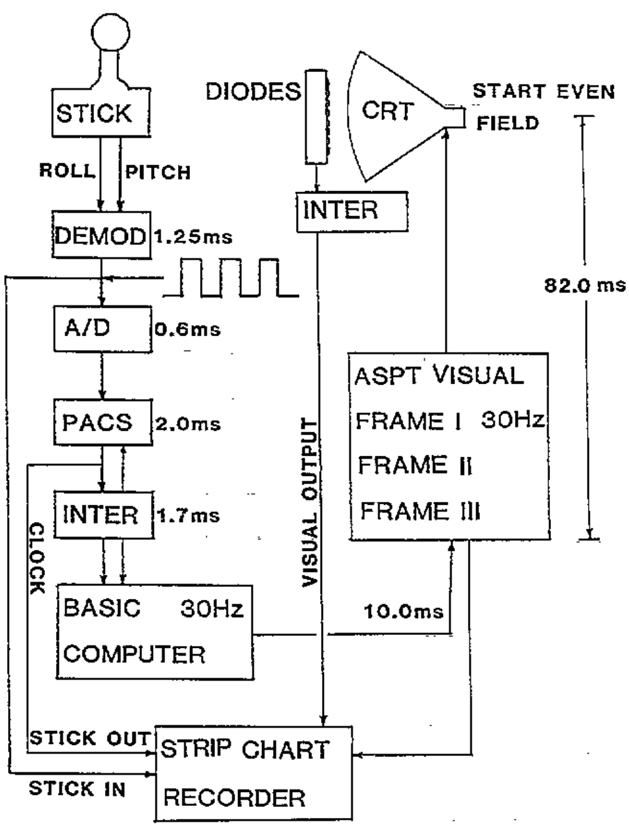


Figure 3

This card provided the pilot input as a stick voltage due to stick pressure as well as the signal the basic computer sent to the visual computer (pitch angle and roll angle). This information was all sent as analog signals to the strip chart recorder. The interface to the visual output was not as straightforward. Since there are delays associated with the calculation of the visual scene as well as delays due to the projection system (cathode ray tube (CRT), light valves, etc.), it was decided the best way to measure the visual output was directly from where the pilot would detect the visual scene, i.e., the display system itself. In order to accomplish this, a device was designed and constructed by Mr Bill Leinenwever of General Electric to convert the moving image on a CRT to an analog signal. This system is shown in Figure 4.

VISUAL TO ANALOG (V-A) CONVERTER

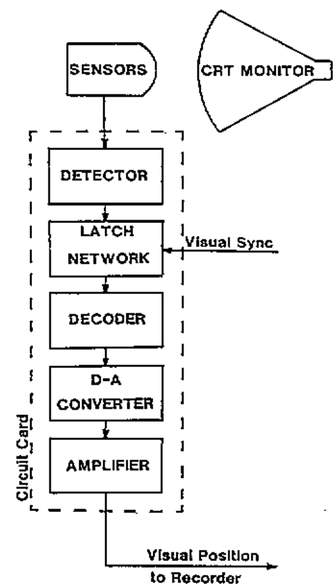


Figure 4

The visual detector consists of five main components: (1) detector, (2) latch network, (3) decoder, (4) D/A converter, and (5) amplifier. The detector consists of 16 photo sensitive transistors mounted on a PC board. Each sensor covers approximately 12 raster lines on the CRT monitor (6 odd field and 6 even field). The window definitions for the monitor were changed in the software so that several scan lines on the monitor correspond to one scan line in the actual viewing field. The latching network is connected to the visual computer and is run by the 60Hz pulse that runs the system which synchronizes the latching network with the visual output. The latching network "holds" the information from the photo transistors for the whole field since the raster only momentarily illuminates the photo transistor and would result in a momentary spiked output rather than a continually increasing output as the horizon on the monitor moved. The information then is decoded, converted to an analog signal, and finally amplified for use by the strip chart recorder. The delay in the electronics of the visual sensor account for approximately 12 nanoseconds and is not considered significant when compared to the quantities being measured.

The software used in the F-16C simulation had to be altered slightly to provide the outputs necessary for determining when the system had received the input from the control stick and sent the information to the visual computer. This presents a bit of a problem not unlike the Heisenburg uncertainty principle: you want to measure some quantity but in the process of measuring it you introduce changes and are now measuring a changed system. Much care needs to be exercised when making changes to the software of a flight simulator to ensure that the system is altered as little as possible. One of the major problems encountered in the software modifications for this program centered around the fact that the simulator did not have a "perfect" trim condition when taken off "freeze." This resulted in the aircraft changing attitude without any stick input and thus made it difficult to determine where the beginning of the measurement of delay began. Another problem encountered was that in order to have a signal to show when the basic computer finished its calculations required the software to send a signal to the D/A converter which sends a signal to the strip chart. The placement of this code in the simulation software is critical and should be as close to the actual shipment of data to the visual computer as possible. Modifications to the software were also introduced to study two types of delay: (1) delay due only to hardware and (2) delay due to hardware and software combined. To measure the delay due only to hardware, the code is changed to allow the stick input to be received by the basic computer while the entire aerodynamics package is skipped. At the system interrupt, the basic computer simply sends the visual computer either a 90 degree pitch up or down signal corresponding to whether the stick was pulled or pushed. This results in a step input at the stick providing a step output of the visual system. If the stick is driven by a square wave generator, then the output through the visual system will also be a square wave with a phase shift corresponding to the delay of the system. The setup that includes the software for the aerodynamics of the aircraft should yield the same transport delays as long as everything is working correctly. The only difference will be that a square wave input will not result in a square wave output, as the aerodynamics of the aircraft will act as a filter and distort the results, but the onset or "time to wiggle" should remain the same. If the test shows that the delays were increased, it is expected the increase will be in increments equal to the frame time of the system, as the software package may not have completed before the end of the frame. In this case the "frame drop" will be detected as a lack of output for one or more frame times (33.3 ms for a 30Hz system). In testing the software it is important to "exercise" the aero package by running the tests with the aircraft in different configurations.

Two methods of collecting data were used for these tests. To measure hardware delay only, the control stick input was replaced with a square wave generator. To select the frequency at which to drive the system one must first determine the expected delay time and corresponding frequency of the system being measured. In this case, a maximum delay of about 150ms was expected which is equivalent to 6.66Hz. Since each cycle of the square wave will input both an up and down pitch (right and left roll), the signal frequency must be no larger than half the system frequency or 3.33Hz. It should also be noted that if the frequency selected is exactly an integer fraction of the system frequency the measured delay will be a constant because the input will always be in sync with the system. In order to measure the range of delays, the signal frequency should be slightly offset. The second method used to measure delay was the use of a step input. This was accomplished with the aid of a pilot and a stick cutout switch. The pilot first trimmed the aircraft to fly straight and level. The next step was to turn the cutout switch on which removed the control inputs from the system. The pilot then input max stick deflection (actually pressure on the F-16C) and the cutout switch was deactivated. The result was for the system to receive a step input from a trimmed condition which bypassed the problem of the simulator's inability to come off "freeze" in a trimmed state.

DISCUSSION OF RESULTS

The first analysis to be accomplished when measuring a simulator system for transport delay is to determine what delays are expected. In order to accomplish this, several items must be determined: (1) the delays of each of the systems components, (2) the type of visual perception model to be employed, and (3) how the components are interfaced. A schematic of the system used in this experiment is shown in Figure 3, which includes the transport delays for each of the system components. The different perception models basically refer to when to assume the pilot has perceived a change in the visual scene i.e., the beginning of the first field, the end of the first field, or the end of the second field. In this study the beginning of the even field (the second field) was used as the moment of perception. Using these definitions and the data in Figure 3, it is a relatively straightforward task to add up all of the delays in the system. Adding all of the delays in Figure 3 yields a total maximum transport delay of 130.9 ms. Unfortunately, things are not quite so simple. In order to understand the internal workings of this simulator system one must look at a timing diagram which shows how all of the devices are related to the system clock. Figure 5 shows the timing diagram for this simulator system. The most important item to note is that the location of the software commands for reading the control input is critical in determining the expected transport delay.

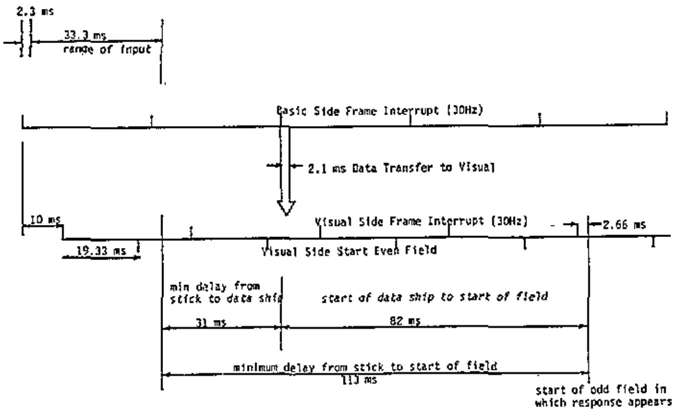
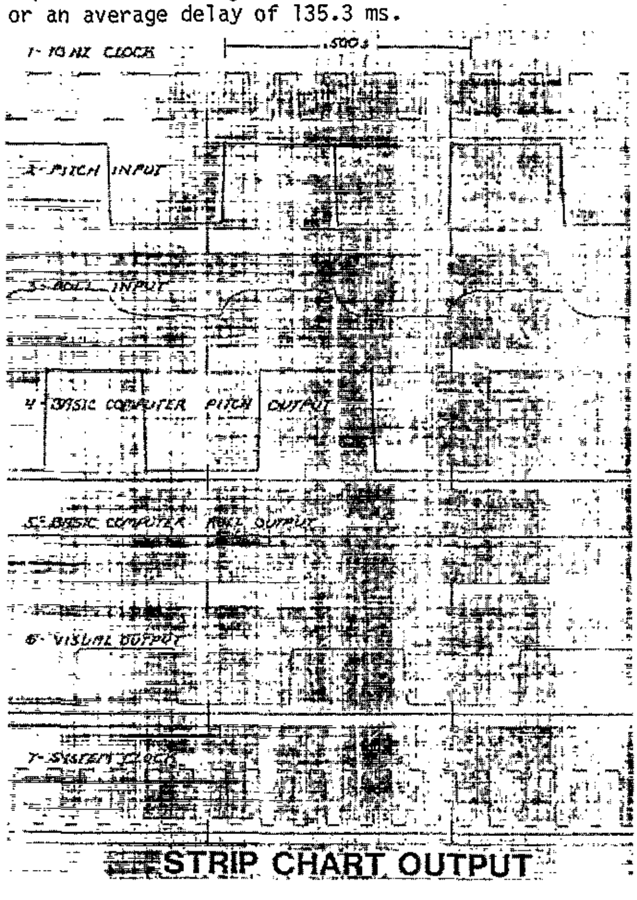


Figure 5

As shown in Figure 5, this reading occurs 10 ms after the beginning of the basic computer interrupt, also the stick read occurs at 2.3 ms into the basic frame and thus the expected delay should be 12.3 ms less or 118.6 ms. Since the stick input may occur at any time there is a range of delays that will be encountered. This range is determined by the length of the basic computer calculations which for our system is 33.3 ms. So the total delay that should be expected will range from 118.6 ms to 151.9 ms, or an average delay of 135.3 ms.



The raw data for this experiment was collected on an eight channel strip chart recorder running at 200 mm/sec. The channels used for this experiment were 1-7 and had the following information: (1) 10Hz reference clock, (2) pitch input, (3) roll input, (4) basic computer pitch output, (5) basic computer roll output, (6) visual output, and (7) system clock (frame interrupt). An example of the raw data collected is shown in Figure 6. A sample of the reduced data for these tests is presented in Figure 7.

DELAYS WITHOUT FLIGHT EQUATIONS

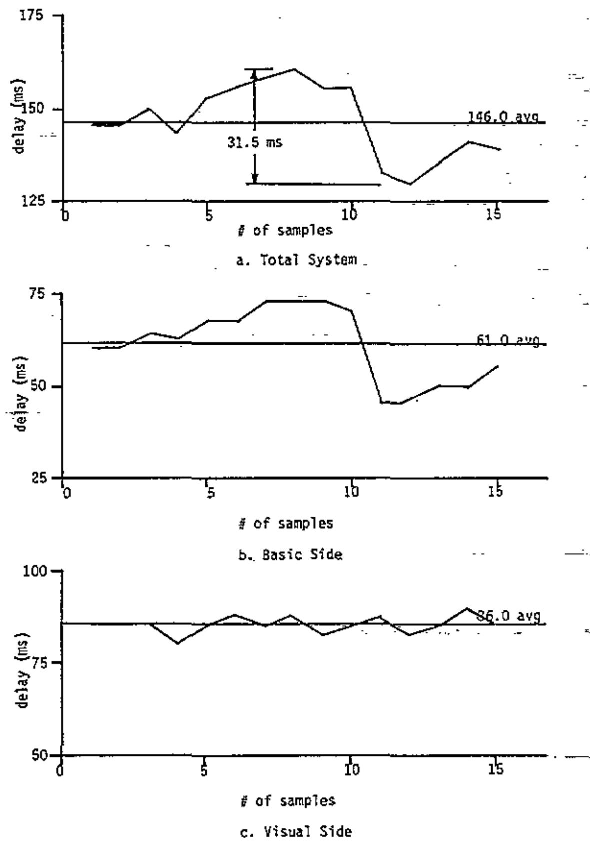


Figure 7

These graphs show the transport delay as a function of sample number for the no flight equations case. Graph 7A shows the results for the entire system, Graph 7B shows the results for the basic computer side only, and Graph 7C shows the results for the visual side only. Graph 7A shows the measured delay varies by approximately 31.5 ms, versus 33.3 ms expected, and shows the average total transport delay to be 147 ms. This average delay is about 11.7 ms over the expected result. Graph 7B shows the transport delay from the control input to the basic output is averaging 61 ms, which is approximately 10.1 ms greater than the 50.9 ms delay expected. Graph 7C shows the visual system is almost constant at 86 ms, which compares favorably to the 84.1 ms expected. It

should also be pointed out at this point that the delays listed in Figure 3 for the devices between the control stick and the basic computer are considered worse-case delays. Therefore the 10.1 ms discrepancy is a minimum and the actual difference may be as high as 15 ms. This discrepancy was researched in some detail, and *the only conclusion that could be drawn was* that the interface between the PACS and the basic computer is not operating as advertised.
Figure 8 shows the same results except that the software for the flight equations was included. For the flight condition tested, the software did not always complete in time for the data transfer to the visual system, as can be seen by the spikes in Graph 8B.

DELAYS WITH FLIGHT EQUATIONS

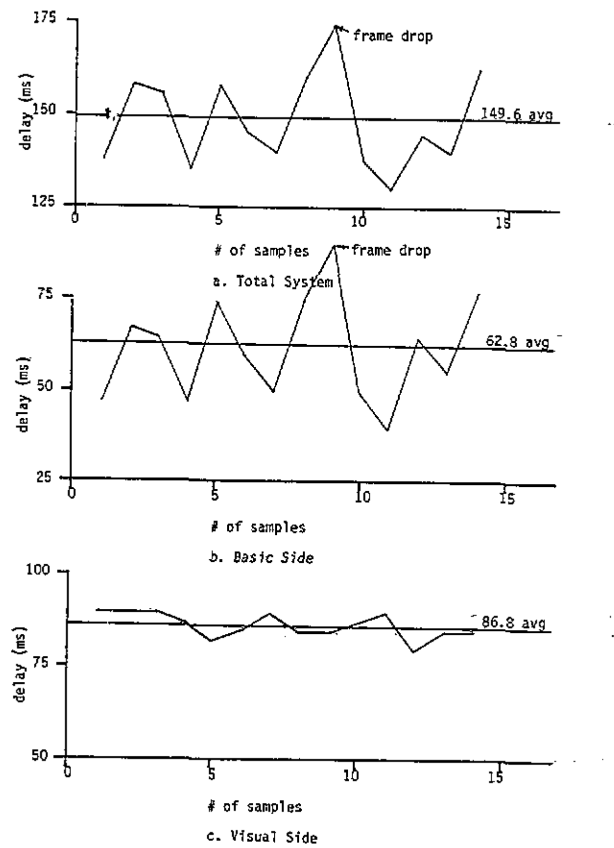


Figure 8

The final figure presented is Figure 9. This graph shows the average total transport delay as a function of the time since the beginning of the experiment. It is interesting to note that the results show the system operating well outside of specifications at the beginning of the experiment and asymptotically approaching the design specs near the end of the experiment. What makes this result even more interesting is the fact that the contractors working on the simulator maintain that no changes to the system were made. It is left to the reader to draw any conclusions he/she wishes from this figure.

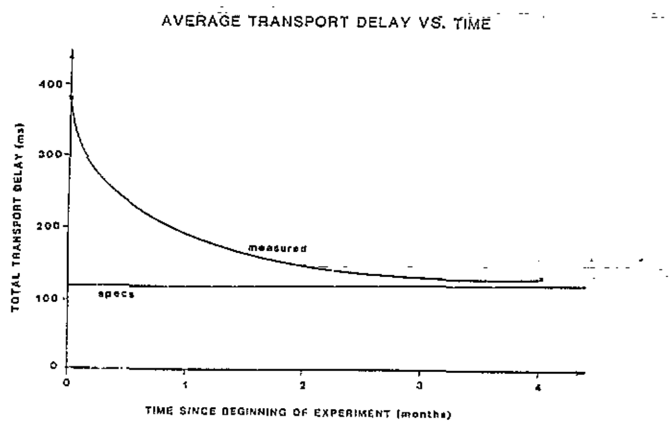


Figure 9

The final result that was obtained concerns the pilots' acceptance of the simulator. Before this experiment was begun the simulator was being used for transition training and familiarization. The basic response from the pilots was that the simulator did not fly like the aircraft and did not handle well at all. By the end of the experiment the pilots who were interviewed still said that the simulator did not fly like the aircraft, but stated that the simulator was not any harder to fly than the aircraft and that all tasks could be accomplished without much trouble. While this hardly constitutes an in depth examination of the handling qualities of a simulator as a function of the transport delay it does indicate that a properly operating simulator with minimal transport delay will be more acceptable to the pilot.

CONCLUSIONS

These results indicate the importance of making transport delay measurements on simulation equipment. These measurements verify if the device is actually performing according to specifications and if not where the potential bottleneck is occurring. The method developed for measuring these transport delays is relatively simple and includes a unique and innovative technique for determining the output from a simulator visual display. Poor simulator handling qualities can be directly attributed to excessive transport delays and further studies are required to quantify these effects. In order to determine the effects of transport delay on simulator handling qualities, another experiment is in progress that will utilize an in-flight simulator and ground based simulators. This experiment will look at the effects of varying transport delay on the acceptance of a flight simulation.

ACKNOWLEDGMENTS

The efforts of the following people made this research possible: Lt Paul Choudek and J. C. Gainer of AFHRL/OT, Mary Bollman and Nancy Heen of Singer/Link Division, Bill Leinenwever and David Lerman of General Electric, and visiting professor Dr Sybil de Groot.

ABOUT THE AUTHOR

CAPT SCOTT J. HOROWITZ is the chief engineering evaluation pilot for the Air Force Human Resources Laboratory, Operations Training Division (AFHRL/OT) at Williams Air Force Base, Arizona. He is also a T-38 instructor pilot for the 82 Flying Training Wing at Williams. He is currently responsible for AFHRL's work in simulator handling qualities research, Fighter Lead-In Trainer development, and Man-In-Space efforts. He holds a B.S. in Engineering from California State University at Northridge, Master's degree in Aerospace Engineering from Georgia Tech, and a PhD in Aerospace Engineering from Georgia Tech. After working for Lockheed Georgia Co., as an associate engineer he entered the US Air Force to attend pilot training and was assigned as a T-38 instructor pilot after graduation. Scott has published numerous articles on a variety of aerospace topics including: Numerical Techniques for Aeroacustic Analyses, Composite Aircraft Design, Finite Element Analysis, Experimental Aircraft Design, and Simulator Handling Qualities Studies.