

Retrospective Theses and Dissertations

1993

Measuring and Compensating for Transport Delay in Real-time Interactive Driving Simulation

Joseph D. Dumas II

University of Central Florida, Joe-Dumas@ucf.edu

 Part of the [Electrical and Computer Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/rtd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Dumas, Joseph D. II, "Measuring and Compensating for Transport Delay in Real-time Interactive Driving Simulation" (1993). *Retrospective Theses and Dissertations*. 3649.
<https://stars.library.ucf.edu/rtd/3649>



MEASURING AND COMPENSATING FOR TRANSPORT DELAY IN A
REAL - TIME INTERACTIVE DRIVING SIMULATOR

By
Joseph D. Dumas II

1993

UCF



UNIVERSITY OF CENTRAL FLORIDA
COLLEGE OF ENGINEERING

DISSERTATION APPROVAL

DATE: April 1, 1993

BASED ON THE CANDIDATE'S SUCCESSFUL ORAL DEFENSE, IT IS
RECOMMENDED THAT THE DISSERTATION ENTITLED:

"Measuring and Compensating for Transport Delay in a Real-Time Interactive
Driving Simulator"

PREPARED BY: Joseph D. Dumas II

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY IN ENGINEERING

FROM THE DEPARTMENT OF Electrical and Computer Engineering

COMMITTEE:

Harold I. Klee
Harold I. Klee, Chair

Christian S. Bauer
Christian S. Bauer

Soheil Khajenoori
Soheil Khajenoori

Charles E. Nuckolls
Charles E. Nuckolls

FOR E.J. Rinalducci - H Klee
Edward J. Rinalducci

APPROVED:

Nicolaos S. Tzannes
Nicolaos S. Tzannes
Department Chair

Fred S. Gunnerson
Fred S. Gunnerson
Director of Graduate Affairs

Gary E. Whitehouse
Gary E. Whitehouse
Dean

The material presented within this report does not necessarily reflect the opinion of the committee, the College of Engineering or the University of Central Florida

UNIVERSITY OF CENTRAL FLORIDA
COLLEGE OF ENGINEERING

DEFENSE OF DISSERTATION

THE UNDERSIGNED VERIFY THAT THE FINAL ORAL DEFENSE OF THE
DOCTOR OF PHILOSOPHY DISSERTATION OF Joseph D. Dumas II
HAS BEEN SUCCESSFULLY COMPLETED ON April 1, 1993
TITLE OF DISSERTATION: "Measuring and Compensating for Transport Delay
in a Real-Time Interactive Driving Simulator"

FROM THE DEPARTMENT OF: Electrical and Computer Engineering

COMMITTEE:

Harold I. Klee
Harold I. Klee, Chair

Christian S. Bauer
Christian S. Bauer

Soheil Khajenoori
Soheil Khajenoori

Charles E. Nuckolls
Charles E. Nuckolls

FOR E.J. Rinalducci - H. Klee
Edward J. Rinalducci

APPROVED:

Nicolaos S. Tzannes
Nicolaos S. Tzannes
Department Chair

Fred S. Gunnerson
Fred S. Gunnerson
Director of Graduate Affairs

Gary E. Whitehouse
Gary E. Whitehouse
Dean

**MEASURING AND COMPENSATING FOR TRANSPORT DELAY IN A
REAL-TIME INTERACTIVE DRIVING SIMULATOR**

by

JOSEPH D. DUMAS II
B.S., University of Southern Mississippi, 1984
M.S., Mississippi State University, 1989

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Engineering
Department of Electrical and Computer Engineering
College Of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1993

Copyright 1993

by

Joseph D. Dumas II

ABSTRACT

Real-time, man-in-the-loop simulators are important tools for operator training as well as human performance research. Simulator implementation using digital computers offers many important advantages but may also cause problems. One of the most significant and troublesome artifacts of digital computer simulation is the presence of transport delays in the operator/vehicle control loop. Transport delays have been shown to destabilize the system, resulting in poorer control of the simulated vehicle. They may also contribute to an increased likelihood of "simulator sickness" in human operators. Therefore, it is desirable to be able to quantify simulator transport delays and to compensate the system in such a way that delay effects on operator performance and well-being are minimized.

The research presented in this dissertation involved the measurement of simulator transport delay using two different methods: a time-domain approach involving the detection of a response to a simulated step control input, and a frequency-domain approach involving the measurement of phase shift from a simulated sinusoidal input. Algorithmic compensators (digital filters) were developed to provide phase lead to counteract the system transport delay. Two compensators designed using approaches previously described in the literature canceled out delay reasonably well; however, a new compensator design developed by the author

provided more nearly ideal phase performance without introducing unwanted side effects such as visual jitter.

The transport delay measurement and compensation techniques were applied to a low-cost, real-time interactive automobile driving simulator developed at the University of Central Florida. The investigations using both measurement techniques revealed that a substantial amount of delay was present in the system. The three delay compensators implemented in the simulator were found (by reapplication of the frequency-domain or steady-state delay measurement technique) to operate approximately as designed. Finally, a driver-in-the-loop experiment was conducted to assess the effect of delay compensation on driver/vehicle performance. While the small size of the experiment allowed no definite conclusions to be drawn regarding the efficacy of compensation, trends in the data were generally indicative of better performance with compensation.

ACKNOWLEDGEMENTS

Many people contributed directly and indirectly to this work. The UCF Driving Simulator would not exist in its present form without the efforts of Mike Garnsey, Chris Nichols, Lew Alexander, and Pete Polasek, among others. Nor would it exist without the leadership of Dr. Harold Klee, who brought us all together and oversaw our efforts, or Brian Goldiez of the Institute for Simulation and Training, who helped Dr. Klee get the project started four years ago. I wish to thank all who have participated in the driving simulator project and helped make this work possible.

One of the delay compensators implemented in the simulator, as well as the inspiration for my own compensation algorithm, came from the work of Richard McFarland at NASA Ames Research Center, Moffett Field, California. Mr. McFarland's published papers were very interesting and instructive; he also offered additional information and advice through personal communications. I'd like to acknowledge his work in advancing the state of the art in transport delay compensation and his assistance with my efforts to do the same.

I would like to take this opportunity to thank the members of my doctoral committee. Dr. Chris Bauer and Dr. Soheil Khajenoori, Computer Engineering faculty members, gave of their time to help oversee this work. The service of Dr. Ed Rinalducci of Psychology is also appreciated. Dr. Charles Nuckolls of Mechanical Engineering played an important role in the development of the driving simulator in addition to serving on the dissertation committee. Most of all, I'd like to thank Dr. Harold Klee for cheerfully and professionally playing the roles of committee chairman, advisor, proofreader, supervisor, occasional tennis opponent, and friend.

Support for this research was provided by the Link Foundation through the Link Fellowship in Advanced Simulation and Training. This generous assistance is gratefully acknowledged by the author.

Even more important than the technical contributions acknowledged above is the personal support given by family and friends. Tennis with Mike and trivia with Carolyn and Craig helped keep my morale high. So did other friends too numerous to name here. My father, Joseph Dumas Sr., has long been a source of support and inspiration in my academic (and other) efforts. I thank him, my sisters Carrie and Jennifer, and their families for their love and encouragement. Most of all, I would like to express my loving thanks to my wonderful wife, Chereé, for all she is, does, and has done over the past several years. She and my two stepsons, Johnathan and Joshua, put up with my moods and my absences during the writing of this dissertation and offered little complaint. We did it, hon! (Your turn now.)

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER 1 — INTRODUCTION	1
Limitations of Low-Cost Real-Time Simulators	3
The University of Central Florida Driving Simulator	12
The Simulator Hardware	13
The Simulator Programs	15
Host Computer Programs	16
Image Generation Computer Programs	21
CHAPTER 2 — THE EFFECTS OF TRANSPORT DELAY IN SIMULATORS	22
Problems With Control	22
Simulator Sickness	31
CHAPTER 3 — MEASURING TRANSPORT DELAY	34
Time-Domain Approaches in the Literature	35
Steady-State Approaches in the Literature	38
Time-Domain Delay Measurements in the UCF Driving Simulator	39
Time-Domain Delay Measurement Procedure	40
Results of Time-Domain Delay Tests	44
Steady-State Delay Measurements in the UCF Driving Simulator	49
Setup and Procedure for Steady-State Delay Measurements	49
Video Detection Circuit	50
Host Computer Software Modifications	51
Image Generator Software and Database Modifications	58
Measurement Procedure	59
Results of Steady-State Delay Tests	61
Analysis of Simulator Delay Measurements	69
CHAPTER 4 — TRANSPORT DELAY COMPENSATION	74
Transport Delay Compensation Algorithms in the Literature	74
Predictive (Lead) Filtering	76
Lead/Lag Filters	79
McFarland's Compensator	84

Application of Delay Compensation Algorithms to the UCF Simulator ..	88
Lead/Lag Compensation in the UCF Simulator	88
Application of the McFarland Compensator to the UCF Simulator ..	91
Other Compensation Approaches Considered	94
A New Compensation Algorithm	96
A Compensator Using Five Velocity Terms	96
A Compensator Using Four Velocity Terms	102
 CHAPTER 5 — TESTING THE EFFECTIVENESS OF DELAY COMPENSATION	114
Steady-State Delay Measurements of the Compensated System	115
Driver-In-The-Loop Experiment	130
Experimental Design and Procedure	130
Data Analysis	137
 CHAPTER 6 — SUMMARY AND CONCLUSIONS	146
 APPENDICES	153
A. Real-Time C Program Code for Velocity-Based Compensator	154
B. Real-Time C Program Code for Lead/Lag Compensator	162
C. C Program to Compute Constraint Equation Coefficients	169
 REFERENCES	173

LIST OF TABLES

1. Duration of Host Computer Operations and Routines	44
2. Transport Delay Measured With Oscilloscope	46
3. Transport Delay Measured With Logic Analyzer	47
4. Vehicle Dynamics Model Phase Measurements From Offline Simulation .	63
5. Steady-State Delay Measurements (No Moving Objects)	64
6. Steady-State Delay Measurements (With Moving Objects)	65
7. Transport Delay Measurements (No Moving Objects)	67
8. Transport Delay Measurements (With Moving Objects)	68
9. Total Effective Simulator Transport Delay	72
10. Steady-State Measurements (No Moving Objects, Lead/Lag Compensator)	116
11. Transport Delay (No Moving Objects, Lead/Lag Compensator)	117
12. Steady-State Measurements (Moving Objects, Lead/Lag Compensator) . .	118
13. Transport Delay (Moving Objects, Lead/Lag Compensator)	119
14. Steady-State Measurements (No Moving Objects, 3-term Compensator) .	121
15. Transport Delay (No Moving Objects, 3-term Compensator)	122
16. Steady-State Measurements (Moving Objects, 3-term Compensator)	123
17. Transport Delay (Moving Objects, 3-term Compensator)	124
18. Steady-State Measurements (No Moving Objects, 4-term Compensator) .	126
19. Transport Delay (No Moving Objects, 4-term Compensator)	127

20. Steady-State Measurements (Moving Objects, 4-term Compensator)	128
21. Transport Delay (Moving Objects, 4-term Compensator)	129

LIST OF FIGURES

1. Block Diagram of Simulator Hardware Components	14
2. Explicit and Implicit First-Order Integration Methods	18
3. Explicit and Implicit Second-Order Integration Methods	19
4. Driving Simulator Host Computer Frame Timing	20
5. Flow Chart for Time-Domain Delay Tests	41
6. Timing Diagram for Time-Domain Delay Tests	43
7. Video Output Detection Circuit	52
8. YCAR Response with Various Values of STEERW Phase	55
9. Flow Chart for Steady-State Delay Tests	56
10. Timing Diagram for Steady-State Delay Tests	57
11. Comparison of 1000 Hz and 60 Hz Solutions of Vehicle Dynamics Model	60
12. Vehicle Dynamics Phase Shift with 1 rad/s Steering Input	62
13. Vehicle Dynamics Phase Shift with 1 Hz Steering Input	63
14. Single Frame Lead from Numerical Integration	71
15. Block Diagram of Simulator with Predictive Delay Compensator	75
16. Block Diagram of Lead/Lag Delay Compensator	81
17. Block Diagram of McFarland's Delay Compensator	84
18. Bode Plot of Lead/Lag Compensator Gain Magnitude and Phase	90
19. Bode Plot of McFarland's Compensator ($P = 0.083$ s, $T = 0.02$ s)	92

20. Bode Plot of McFarland's Compensator ($P = 0.174$ s, $T = 0.01667$ s) . . .	92
21. Comparison of Phase Characteristics (McFarland vs. Lead/Lag)	93
22. Comparison of Gain Characteristics (McFarland vs. Lead/Lag)	94
23. Bode Plot of Phase Characteristics, Five-velocity-term Compensator	101
24. Bode Plot of Gain Characteristics, Five-velocity-term Compensator	101
25. Bode Plot of Phase Characteristics, McFarland's Four-term Compensator	104
26. Bode Plot of Gain Characteristics, McFarland's Four-term Compensator .	105
27. Bode Plot of Phase Characteristics, Author's Four-term Compensator . . .	109
28. Bode Plot of Gain Characteristics, Author's Four-term Compensator	110
29. ANOVA Table and Confidence Intervals for Lateral Position Deviations	138
30. ANOVA Table and Confidence Intervals for Steering Wheel Reversals . .	140
31. ANOVA Table and Confidence Intervals for Steering Input Deviations . .	142
32. ANOVA Table and Confidence Intervals for Lateral Acceleration	143
33. ANOVA Table and Confidence Intervals for Vehicle Speed Deviations . .	145

CHAPTER 1

INTRODUCTION

Vehicle simulation has evolved over the past several decades from primitive beginnings into a valuable tool for research and operator training. The earliest, and still the most common, vehicle simulators were flight simulators. The idea of training pilots cheaply and safely has been the common denominator from the days of Edward Link's first trainer to the high-technology systems that simulate the operation of today's multimillion-dollar high-performance fighter aircraft and even the Space Shuttle orbiter.

In recent years, the once astronomical price tag of simulator hardware components has been reduced by orders of magnitude. It has thus become possible to consider building simulators for ground vehicles such as automobiles and trucks as well as aircraft. While some aircraft and automobile research simulators cost in the millions of dollars, it is now possible to construct an interactive simulator suitable for basic driver training applications for well under \$100,000. Obviously in the course of developing such a simulator certain tradeoffs must be made which affect the fidelity of the system. However, innovative solutions to the problems imposed by low-cost constraints can drastically improve performance and training effectiveness.

The research presented here focused on ways to improve the handling qualities of a low-cost, real-time interactive automobile driving simulator developed at the University of Central Florida. During system development, difficulties in controlling the simulated vehicle were observed in demonstrations and informal tests conducted by project personnel. It was believed that these handling problems were due in large part to the presence of transport delay (to be defined below) in the simulation loop, especially in the computer graphics system used to provide visual feedback to the driver. Prior to the conduct of this research, however, this assumption had never been adequately tested. The scope of the research performed included measurement of the simulator transport delay to determine the extent of the problem, the development and implementation of delay compensation algorithms for the simulator, and the testing of the compensated versus uncompensated system.

The remainder of this chapter discusses the limitations of low-cost interactive vehicle simulators in general, including factors that contribute to delay. An example of a low-cost simulator, the UCF Driving Simulator, is described in some detail. The second chapter describes some of the deleterious effects of transport delay in interactive simulators. Chapter 3 outlines procedures for measuring transport delay using time- and frequency-domain based techniques and discusses the application of those techniques to the UCF Driving Simulator. Chapter 4 discusses past delay compensation research and describes the application of some delay compensators, including a newly-developed predictive algorithm, to the UCF simulator. Chapter

5 discusses the results of testing the compensated versus uncompensated system. Conclusions drawn from this work are presented in Chapter 6.

Limitations of Low-Cost Real-Time Simulators

Digital computers have largely replaced analog and hybrid computers in simulation for a number of reasons. Among these are dynamic range, noise immunity, reliability, and — perhaps most important of all — flexibility (or ease of reprogramming). These advantages, however, come at the expense of certain limitations. For example, while analog signals — and thus analog computers — have theoretically infinite resolution, digital devices (including computers) have finite word lengths and thus finite resolution. While the resolution of any individual computation can be very high, solution of complex sets of equations describing real-world processes can involve large numbers of operations and thus introduce roundoff errors. In addition, integration and/or differentiation of signals cannot be done directly, as in analog computers, but must be approximated using numerical techniques. This introduces truncation errors related to accuracy in addition to errors in dynamic response when comparing solutions obtained by digital approximation and analog (exact) methods. The most fundamental limitation of digital computers, particularly from the point of view of real-time simulation, is the fact that they are serial processing devices. "Because digital computers are serial devices," Casali and Wierwille [1] explained, "they introduce delays ... in every type of computation performed. High-speed machines can perform simple computations

rapidly but not instantaneously. The more complex the computation and the slower the speed of the machine, the longer the computation time"

Serial digital processing causes time-delayed responses, or transport delays, in simulator systems. A transport delay is a "pure" time delay, "where the output of the system is a faithful representation of its input, only it appears after a fixed amount of time [2]." A transport delay of T seconds is represented in Laplace transform notation as e^{-sT} (in contrast to dynamic lags, which are represented by transfer functions with polynomial denominators in s). These transport delays "are inappropriate in the simulator if [they contribute to lags] in excess of the normal control response lags inherent in the actual system dynamics [1]." In the words of Johnson and Middendorf [3], "simulator transport delay is defined as the time delay between pilot input and pilot cueing solely due to simulator implementation. Delay due to the dynamics inherent in the real [system is] not part of the transport delay." While transport delays may result from other aspects of simulator implementation such as communications, "inertial effects in motion ... systems, control input sampling rates, iteration rates of motion cuing algorithms and visual display generators, and analog-to-digital or digital-to-analog conversion rates [1]" as well as "digital integration techniques ... and anti-aliasing filters [4]", the longest and most objectionable delays are due to serial processing for vehicle dynamics and computer image generation. To further complicate matters, the amount of delay in a digital simulation may be variable "depending on the instantaneous load on the computational systems and memory storage capabilities", among other factors.

Delays in the man-machine visual feedback loop are most serious since human beings obtain most of the information needed for almost any task, particularly control, visually. Allen [5] noted that "visual displays are the primary means for providing feedback to the human operator in vehicle control tasks such as car driving and aircraft piloting." Various problems have been encountered and dealt with as simulator visual systems have evolved. "The advent of computer-generated imagery ... has mostly overcome previous limitations [of oscilloscope displays, closed-circuit TV model boards, and other more primitive visual technologies], but has added a host of new concerns including computational delay" In the words of Crane [6], "CGI visual systems offer important advantages, including large field of view, ease of scene modification, and independent motion of scene elements" However, "the image construction time, though short ... introduces a delay into the pilot-aircraft system."

In most modern air or land vehicle simulators, the largest component of pure time delay is due to the computer image generation subsystem. This is despite the fact that it is often the most expensive, or one of the most expensive, simulator components [7]. These "delays occur because typically the computer calculates the simulated vehicle's current position before it calculates (usually serially) the CIG visual scene. This problem can be exacerbated even further by the current practice of using separate computers of differing update frequencies for the motion and visual subsystems [8]." The visual scene is updated at intervals (not necessarily corresponding to the vehicle dynamics frame rate) rather than continuously. There

may be additional delays due to pipelining of the graphics computer. For example, Crane [6] described a CGI system that updated the image every 33.3 milliseconds; the three-stage pipeline structure, however, resulted in a transport delay of 100 ms. To further complicate matters, this delay is not necessarily constant. "For example, a Computer Generated Image (CGI) that updates at varying rates causes a range of possible delay times [3]." Also, the "delay values may vary slightly depending upon when the dynamics processor makes the results of its calculations available to the CGI system [9]."

Allen, citing McRuer's description of lead generation by a human operator [10], listed "smooth-appearing motion" as a primary requirement of a visual display system for a vehicle control simulation. The illusion of smooth motion "is essential ... in order that the ... operator can anticipate vehicle movement [5]." The specification of visual update rates required for acceptable smoothness depends on roll, pitch, and yaw rates as well as translational velocities. In particular, forward velocity relative to the distances to observable objects is of concern. "It can be shown geometrically that [angular velocity of objects moving toward the edge of the display] is not a matter of absolute velocity, but of velocity relative to the range of an object. Thus, ground vehicles ... can generate just as high scene expansion rates as high speed aircraft because they typically move much closer to scene elements." Due to pipelining effects, fast update rates do not necessarily correspond to small transport delays. (It may be possible in some systems to rapidly update the screen with images, each of which took a long time to calculate.) However, one can draw the

inference from Allen's observation that CIG transport delay, which (like update rate) affects the timing of visual feedback to the simulator operator, can have an equally bad (or possibly worse) effect on the control of ground vehicles as it does on the control of aircraft.

Digital computer processing speed limitations require tradeoffs to be made in any real-time simulator design. For example, the fidelity of the mathematical model is necessarily limited by the processing power of the simulation host computer. While some high-end simulators run detailed, highly nonlinear multiple degree of freedom vehicle dynamics models, low-cost simulators (such as the UCF Driving Simulator) that use minicomputers or microcomputers to run in real-time must necessarily employ simpler models. The UCF simulator, in fact, currently models the simulated vehicle in only three degrees of freedom, incorporating some nonlinearities to describe simple aerodynamic forces, engine/transmission dynamics, and limiting of vehicle performance characteristics (braking, speed). In general, it is possible to run a more detailed vehicle dynamics model at the same hardware cost (on the same computer), but only by cutting into spare frame time (if any) or by reducing the simulation frame rate and thereby increasing transport delay. Thus, there is a tradeoff between temporal fidelity, or timeliness of the simulated response, and modeling fidelity (realism of the response). Likewise, for a given image generator system, it is possible to model the visual scene in greater or less detail (in general, using more or fewer polygons to represent varying numbers of objects). "These factors result in tradeoffs between visual fidelity and temporal fidelity [11]." Going

too far in either direction can result in a poor simulation experience. The consequences of inadequate modeling fidelity and visual fidelity are, obviously, vehicles that do not handle realistically moving through "terrain" that appears barren or perhaps even "cartoonish". On the other hand, poor temporal fidelity (excessive lag) can result in degradation of stability margins and operator control performance. In some cases, simulator sickness (a phenomenon similar to motion sickness) may result. The subsequent chapter on effects of transport delay in simulators treats the subject of simulator delay effects in more detail.

If money is no object, one can overcome almost all the limitations imposed on real-time simulation by digital computing. With sufficient funds, one can buy fast enough computer hardware to generate photorealistic images at a very small transport delay and run highly detailed dynamics models at an iteration rate that very nearly duplicates an analog computer solution. Higher-performance communications and analog interface devices can be added as well. Sophisticated motion systems which further enhance the fidelity of the simulation can be built — for a price. This high-cost, high-technology approach may have been (and may even still be) feasible for full-task military flight simulators which are funded by governmental agencies and which are used to train pilots to fly multimillion-dollar aircraft. It may even work for one-of-a-kind automobile research simulators like the Daimler-Benz driving simulator [12] [13] and the proposed National Driving Simulator. However, tradeoffs must be made when designing a simulator which is intended to be replicated many times and used to teach drivers to operate \$10,000.00 cars.

Deyo, Briggs, and Doenges [7] stated that "a driving simulator should ideally match the lag, from control input to visual result, at a few tens of milliseconds typical of responsive cars." The most advanced currently operational driving simulator, operated by Daimler-Benz, has a visual system transport delay of 80 ms [12]. Casali and Wierwille [1] proposed a general specification for a research simulator (aircraft or ground vehicle) which "budgets" a total of only 25 milliseconds for the maximum allowable overall system transport delay, most of which is allocated for the various tasks involved in updating the visual scene. They cited the work of researchers in manual (man-in-the-loop) control system design, most of whom "would agree that a total loop delay of 25 msec would not appreciably affect system performance or handling. However, they would also indicate that delays greater than 25 msec would probably affect performance. Therefore maximum allowable delay should not be greater than 25 msec." They went on to explain that "manipulating [visual] data quickly and displaying them with only small delays ... is only now becoming possible. Most visual systems ... have delays that approach 100 msec, which are too long If necessary, parallel processing can be used to bring delay times down to acceptable levels." Of course, parallel processing implies replication of hardware, which increases system cost. In situations where this added cost would be prohibitive, for example in the UCF Driving Simulator, some other, less expensive approach must be adopted.

There is evidence to suggest that motion cues may mitigate somewhat the effects of delayed visual feedback. Ricard and Harris [14], citing previous research done at

the Naval Training Equipment Center (NTEC), stated that the activation of a motion system in a flight simulator caused the range of tolerable delays (for a given aircraft) to be extended. The "best tracking was associated with a complete set of ... motion cues. When this set was reduced, tracking performance deteriorated and delays had more of an effect on control performance." The conclusion drawn from these observations was that "motion cues provide additional [information] which enables pilots to generate low-frequency lead."

This topic was explored further by Hosman and van der Vaart [15]. These researchers indicated that because "some time delay [100 to 200 milliseconds] occurs in visual motion perception" by a human operator, while physical motion is sensed much more quickly "due to the differentiating action of certain sensory cells in the vestibular organs", there is "an important advantage of vestibular motion perception when compared to visual motion perception." The incorporation of appropriate physical motion cues in a simulator, where the human operator's visual processing delays are often compounded by serial processing of the visual scene, improves operator performance in "disturbance compensation tasks and target following tasks". Unfortunately, addition of a motion platform to a fixed-base simulator such as the UCF Driving Simulator can be very expensive, both directly (in terms of hardware cost) and indirectly, in the form of presenting a safety hazard that must be dealt with and insured against. Addition of a motion platform may not be cost-effective, since in one particular study the addition of full motion cues (by in-flight simulation using a variable stability aircraft) improved pilot error scores by only about 10 percent [16].

Thus, in many instances other alternatives must be pursued to reduce the undesirable effects of CIG transport delay.

Another attempt at mitigating visual transport delay effects by modifying the system hardware configuration was investigated by Merriken, Johnson, Cress, and Riccio [17]. Their approach used supplementary visual cues rather than motion cues. Experiments were conducted in which a number of non-pilot subjects flew a simulated aircraft. The primary CGI display provided heading, (roll and pitch) attitude, and altitude cues (with a transport delay of 200 ms) while secondary displays located to the left and right of the main display (in the pilot's peripheral vision area) provided attitude (horizon) cues only, at delays of either 67 or 200 milliseconds. Improvements were not found to be statistically significant; however, when the faster-updating secondary cues were provided, "in all cases, RMS error performance was better with the faster updating secondary cues than with the control condition." There was no indication from this study that the 133 ms cue mismatch resulted in any performance degradation. Hosman and van der Vaart [15] inferred from similar experiments that "peripheral field displays and cockpit motion have a similar influence on tracking performance and control behavior when added to a central display", although "changes in performance and subject's dynamic behavior are, as a rule, larger due to cockpit motion than due to peripheral visual cues."

In the case of the UCF simulator, financial realities have prohibited the acquisition of significantly faster CIG hardware, additional display channels, and/or

a motion platform. The remaining alternative offering the best hope for improved system performance was, and is, compensation for transport delay in software.

Although little had been published regarding the use of software compensation for transport delays in automobile simulators, there was reason to expect favorable results. To quote the conclusions of Ricard, Norman, and Collyer, "while a compensation for ... delays is potentially useful in all areas of flight simulation, we would expect that improvements resulting" from its use to be more dramatic for tasks which "demand narrow performance tolerances over an extended period of time [9]." While this comment was directed toward flight tasks such as aerial refueling and formation flying, it also could be taken to apply to common automobile driving tasks such as maintaining lane position or following a lead vehicle. A review of the literature revealed several works describing computationally inexpensive, yet more or less effective software-based methods used to compensate for transport delays in flight simulators. The application of some of these techniques, as well as a new one, to the UCF Driving Simulator is described in Chapter 4.

The University of Central Florida Driving Simulator

The research presented here was carried out using the University of Central Florida Driving Simulator. This fully interactive, digital computer-based automobile simulator was described in two papers by Klee [18] [19]. The UCF simulator was conceived and constructed as a low-cost (approximately \$60,000 total hardware outlay) training simulator prototype that would be replicated for use in instructing

students in high school driver education classes, elderly/handicapped driver rehabilitation programs, or other driver training applications. It was also envisioned that this type simulator might have sufficient fidelity at a low enough cost to be used by state agencies for administering driver licensing and/or recertification examinations. More recently, the driving simulator has been used to support other University research projects, for example the development of expert system software to evaluate driver performance and "intelligent training systems" that will lead drivers through practice scenarios of various difficulty levels without the need for an in-vehicle, human instructor. The components and operation of the simulator are described in more detail below.

The Simulator Hardware

The simulator hardware consists of the vehicle cab and cab interface hardware, two digital computer systems, a video projector, and a sound generation/amplification system used to provide audio feedback to the operator. The basic hardware configuration is shown in Figure 1. The primary driver control inputs (steering, throttle, and brake) are sampled every frame (or iteration of the simulation software), using an incremental rotary position encoder for the steering input and analog-to-digital (A/D) converters for the other signals. These quantities serve as inputs to a mathematical model program running on the first computer system (an Intel 80386/387 based platform) which simulates the vehicle dynamics and computes current values for the simulator velocities in three degrees of freedom: longitudinal,

lateral, and rotational (yaw). These velocities are translated into the absolute or world coordinate frame, integrated again to obtain the vehicle position coordinates, and then output to the second computer (also 80386-based) over a serial communications link.

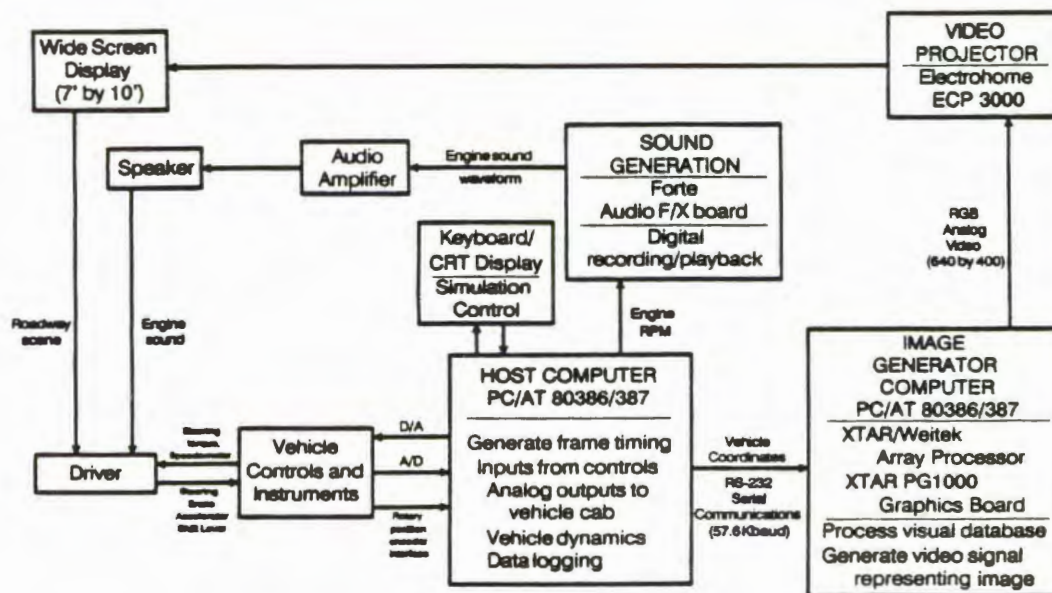


Figure 1. Block Diagram of Simulator Hardware Components

This second computer contains a set of special-purpose boards which accept the vehicle coordinates and perform Computer Image Generation (CIG) for the simulator system. The end product of the CIG computer system is a video signal representing the visual scene from the driver's point of view. The video image is projected on a screen in front of the vehicle cab where it can be viewed by the driver

through the front windshield of the car. This visual feedback to the driver closes the primary man-machine control loop. Thus "the driver, presented with visual ... cues from the roadway environment, responds with appropriate actions involving the steering, accelerator, and brake to control the vehicle's position and heading [3]."

Secondary cues provided by the simulator include the vehicle speedometer, which is calibrated to display vehicle speed sent as an analog signal (D/A converter output) from the simulation host computer, a simple force-feel system using a DC torque motor attached to the steering column and also driven by an analog output from the simulation host, and an engine RPM sound generated by a special-purpose board resident in the host. Because of cost constraints, the simulator does not provide physical motion feedback to the operator. Nevertheless, it is a full-task, fully interactive simulation of routine automobile driving that provides a high degree of realism relative to cost.

The Simulator Programs

The host and image generator computer programs used in the course of this research are based on the software developed for demonstration of the simulator to university faculty, students, and guests. The host computer program, in particular the portion which models the vehicle dynamics, was adapted from code provided by the Federal Highway Administration (FHWA) as used in their Highway Driving Simulator (HYSIM). The program was provided as a Fortran code listing and was converted to C by UCF personnel, who also added I/O routines and other custom

software needed to operate in the IBM-PC compatible environment. The image generator C program and databases were generated entirely by UCF personnel [20] [21], who utilized some routines and tools provided by XTAR, the manufacturer of the image generator board set.

Host Computer Programs. At the time these investigations began, the host computer simulator demonstration program included a vehicle dynamics routine based on the FHWA model with modifications to allow for manual shifting of the automatic transmission, including a reverse gear capability. Integrations of acceleration to velocity were done using (explicit) Euler integration, followed by implicit Euler integration of velocity to obtain position. Implicit Euler integration, also known as backward rectangular or simply rectangular integration, is identical to explicit Euler except that the newly updated velocity value v_{n+1} is used as the state derivative rather than v_n . The diagrams in Figure 2 illustrate the difference between the two techniques. Transformation from body axis coordinates to absolute ("world", or "inertial") coordinates was (and is) done on the translational velocity values before the second integration, so that the integrator outputs represent the absolute (X, Y) position of the simulated vehicle. Because of timing restrictions imposed by a previous host computer system, the host frame rate was set at 20 Hz ($T = 0.05$ second).

The first step in adapting the host computer program for delay measurement and compensation purposes was the removal (for purposes of clarity and

compactness) of all code unnecessary to the simulator's functioning. This included display windowing and other user interface routines. The calling of certain functions by the simulator real-time executive was reordered slightly to optimize for minimum transport delay from control input through visual output. In particular, the calculation of steering wheel restoring torque was removed from the vehicle dynamics routine and placed in a separate function which is not executed until after the vehicle coordinates have been sent to the IG. Also, because of the higher computational performance of the 80386-based host computer (which replaced a previous 80286-based machine), it was found that the host frame rate could be increased from 20 Hz to 60 Hz while still retaining some spare time during each frame. The 60 Hz host computer frame rate was used throughout the course of these investigations.

As part of the analysis of system timing characteristics, the author investigated the phase characteristics of several real-time numerical integration algorithms. The work of Howe [22] and Panzitta [23] indicated that second-order algorithms such as Adams-Bashforth (AB-2) perform better than Euler integration. AB-2 integration has been shown to have considerably less phase error for conditions common in real-time simulation [23] than does Euler integration. For this reason as well as its simplicity of implementation, "the Adams-Bashforth predictor integration routines are generally the most effective for real-time simulation [22]."

With the foregoing results in mind, it was decided to abandon the explicit/implicit Euler integration scheme described previously in favor of using more

accurate second-order algorithms. The program versions used in conducting this research employ the AB-2 method for the acceleration to velocity integrations, followed by trapezoidal integration of velocity to obtain position. Again, it is possible to use an implicit method (trapezoidal) for the second integration because the updated value (at time $(n+1)T$) of velocity is available as the output of the first integrator. The diagrams in Figure 3 illustrate the AB-2 and trapezoidal integration techniques.

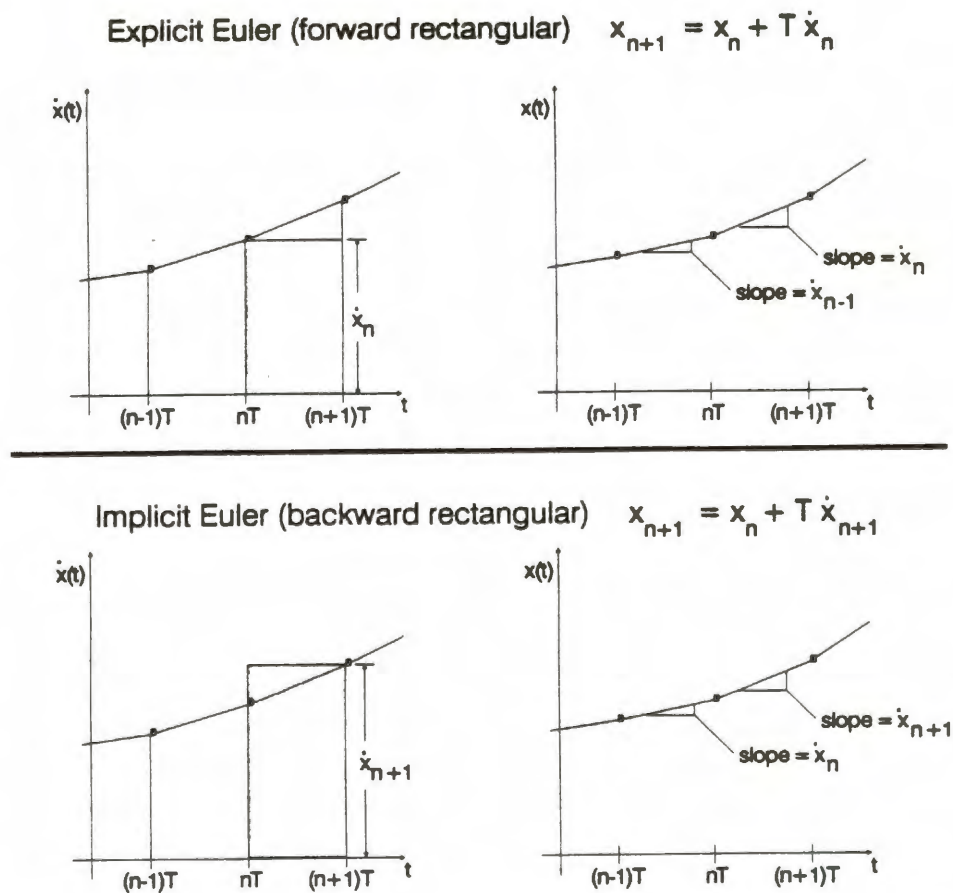
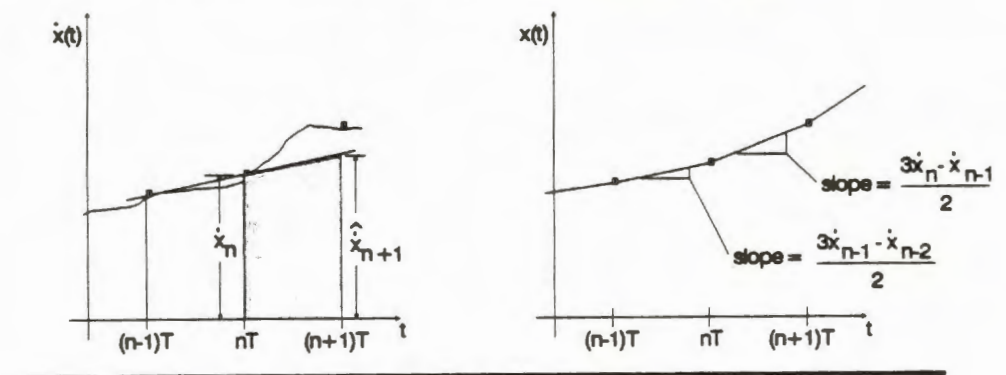


Figure 2. Explicit and Implicit First-Order Integration Methods

$$\text{AB-2 (explicit)} \quad x_{n+1} = x_n + (T/2)(3\dot{x}_n - \dot{x}_{n-1})$$



$$\text{Trapezoidal (implicit)} \quad x_{n+1} = x_n + (T/2)(\dot{x}_n + \dot{x}_{n+1})$$

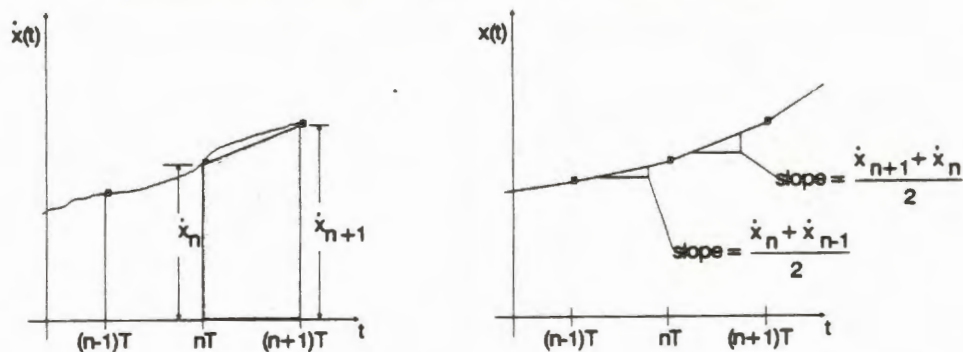
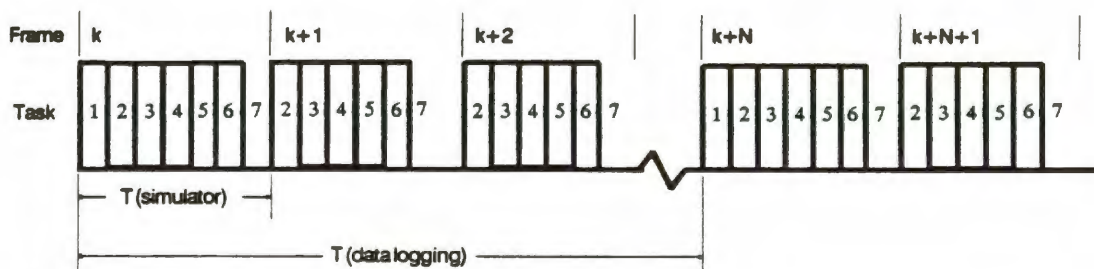


Figure 3. Explicit and Implicit Second-Order Integration Methods

The second-order integration techniques adopted for use in the simulator add little to the computational cost but provide significantly better accuracy than Euler integration. Even more importantly, they possess nearly ideal (identical to continuous integrators) phase characteristics at frequencies of interest. Therefore, the numerical integration itself contributes essentially nothing to the system transport delay. The nature of the integrations, however, when coupled with the timing characteristics of the system (see Figure 4), does allow for some mitigation of

transport delay. A small amount of effective lead is produced by passing to the IG the coordinates for frame $(n+1)$ as soon as they are computed (on completion of the vehicle dynamics routine) rather than waiting until the end of frame (n) /beginning of frame $(n+1)$. This concept is similar to what Gum and Albery [24] termed "single-interval lead"; it is explained in more detail in Chapter 3.



1. Log Data
2. Sample Control Inputs
3. Compute Vehicle Dynamics
4. Send Vehicle Coordinates to IG
5. Compute and Update Vehicle Cab Outputs
6. Update Engine Sound
7. Spare Frame Time

Figure 4. Driving Simulator Host Computer Frame Timing

Informal tests of the simulator revealed no noticeable improvement in vehicle handling characteristics due to reducing the step size and adopting the second-order integration algorithms. Of course, little or no difference was anticipated since transport delay was believed to be the chief cause of the problems previously

observed with simulator operation, and transport delay is only slightly reduced by increasing the host frame rate. (The $T/2$ delay associated with zero-order holds is reduced in length, but no other delay components are changed.) So the program using the modified FHWA dynamics model with AB-2 and trapezoidal integration at a time step of $1/60$ second (60 Hz frame rate) was adopted as the baseline case for the measurement and compensation of simulator transport delay.

Image Generation Computer Programs. The simulator CIG program and databases used in this research were adapted from the latest demonstration versions which included several thousand feet of roads, three-dimensional objects such as trees, signs, and other objects, and two other visible moving cars besides the simulated vehicle. All nonessential code, including routines allowing alternate viewpoints (other than the driver's view) was removed for the sake of simplicity. In order to demonstrate the ability to measure and compensate for transport delays of various duration, two test program versions were created, one with and one without the movable car object databases. A third version of the program, used in early tests, had all three-dimensional object data bases removed, leaving only the two ground (flat) databases (terrain and lines). A fourth test database contained just sixteen flat polygons. This was used in order to establish an approximate lower bound on the XTAR IG transport delay. The IG's color palette was changed to black and white for better contrast in making delay measurements but remained as originally designed for all driving tests.

CHAPTER 2

THE EFFECTS OF TRANSPORT DELAY IN SIMULATORS

There have been a number of papers published over the past twenty-odd years dealing with the problem of transport delay in real-time digital computer simulation of real-world systems, in particular human-controlled vehicles. Some of this literature discusses the simulation of ground-based vehicles such as automobiles, although most of the available sources (no doubt due to funding from defense-related agencies) deal with flight simulators, both for research and pilot training. In general, transport delay effects have been found to be similar whatever the type of vehicle simulated; they fall primarily into the categories of control degradation and simulator sickness. Both types of effects can influence training effectiveness as well as the validity of research data obtained in a simulator.

Problems With Control

McRuer [10] presented a structural model of the human operator or controller of a man-machine system which emphasized "three different types of control operations on ... visually presented system inputs." These operations are classified as compensatory, pursuit, and precognitive modes of control. "With [the compensatory] pathway operational, continuous closed-loop control is exerted on the

machine so as to minimize system errors in the presence of commands and disturbances." The pursuit and precognitive control efforts represent "open-loop control in conjunction with the compensatory closed-loop error-correcting action." McRuer mentioned that "a rather complete example, in which all of the fundamental pathways are involved in the various maneuvers, is driving."

In the same paper, McRuer went on to develop a model of the compensatory control action of a human operator based on observed time histories in a system with a random forcing function (typical of formation flying, lead vehicle following in an automobile, et cetera). He inferred an approximate operator transfer characteristic of the form

$$Y_p = K_p(Ts + 1)e^{-\tau s} \quad (2.1)$$

where K_p represents a (variable) gain factor associated with the operator, τ represents his equivalent pure time delay, and T is the dominant time constant associated with the system dynamics. In other words, the operator's response lags his stimulus by τ seconds (his neuromuscular system delay or "processing and actuation time") but he "develops a lead which is approximately equal to the first-order lag component of the controlled element dynamics." In a simulator with transport delays, however, the system delay adds to the inherent operator delay, requiring the operator (driver or pilot) to generate an increased amount of "lead compensation" in his control effort. The generation of this additional lead compensation in itself imposes a time penalty; in the researcher's words, "the effective time delay τ ... is not a constant. It depends primarily on the amount of

lead compensation required of the operator." Since greater total delays in the control loop (including transport delays) demand more lead compensation from the operator, but increases in lead themselves require more time for him to generate, there is a maximum overall delay in the system beyond which the operator will be unable to compensate. Even at smaller values of transport delay, the operator's workload in controlling the simulator will increase beyond what is required in the actual vehicle.

Allen [5] presented a model of operator/vehicle system control stability similar to (though somewhat simpler than) that developed by McRuer [10]. In this model, there is an effective overall system pure time delay which is the sum of the operator's effective delay and the system transport delay, if any. "As system equivalent time delay increases in going from real vehicles to fixed base simulators, human operators maintain a consistent stability margin by reducing system bandwidth [5]." This reduction in bandwidth "would certainly have consequences in system response and performance." These consequences would be manifested in part by poor opinion ratings of the simulator in question, which have "been shown to degrade with an increase in equivalent system time delay", and also by increased operator workload. Allen also made the point that "delays in feedback of angular motion are much more serious than delays in translational motion which are one integration further removed from the human operator's control actions. This suggests that angular transformations need to be updated most frequently"

Research by Levison and Papazian [16], using an optimal control model for the pilot/vehicle system, specified a typical value for the human operator's effective transport delay of $\tau = 0.2$ seconds. Of course, this value may vary somewhat between subjects, or even for the same subject given the level of workload and attention to the task [10]; nevertheless, it is an approximation more or less agreed on in the literature, for example in references [2] [15] [25]. Hosman and van der Vaart [15], like McRuer, attributed this delay to "all time delays in perception, mental processing and control input generation."

The theoretical effects of transport delay on a man-machine control system can be verified with off-line simulation using a suitable driver describing function. Dumas [26] performed an ACSL simulation study using the same vehicle dynamics model employed in the University of Central Florida driving simulator in combination with driver models similar to those developed by Allen. "For all three [driver] models, increasing simulator delay ... had the expected effect of degrading overall system response." Another finding of the simulation study was that the destabilizing effect of transport delay, as evidenced by increased overshoot during a simulated lane change maneuver, worsened as vehicle forward velocity was increased (from 60 to 90 feet per second).

Practical experience with human control of systems subject to transport or computation delays (both simulators and actual vehicles with digital controls and/or avionics displays) agrees well with theoretical predictions. It has been known for years that delay contributes to degradation of control in real-world systems. For

example, Cooper, Harris, and Sharkey [27] performed an experiment in 1975 using the Naval Training Equipment Center's "TRADEC" F-4 flight simulator to determine the effects of time delay in the simulator's visual system on pilot performance. The performance of 16 pilots and former pilots, all but two of whom were carrier-qualified, was measured during aircraft carrier landing approach tasks. The carrier was visually depicted on a line drawing CRT display system; delay was set to either minimum (12.5 to 25 milliseconds) or minimum plus an artificial 100 ms added delay. The researchers found that there were statistically significant differences between certain pilot control inputs, specifically aileron control displacement and aileron control force, made under the minimum delay and increased delay conditions. Furthermore, the researchers examined the frequency spectra of the pilot control inputs under both delay conditions and found that "the major difference between the Delayed and Non-Delayed spectra typically occurred in the range 0 to 2 Hz." The researchers did not find statistically significant differences in pilot learning performance between minimum-delay (12.5 to 25 ms) and increased delay (100 ms delay added) conditions in the F-4 simulator with a carrier landing task. They conjectured, however, that this result (no significant increase in "trials-to-criterion" for increased delay) "could be due to pilot subjects responding, with extra effort, to the delayed task conditions, i.e., they may have 'tried harder'." This speculation agrees with McRuer's theory regarding increased operator workload due to delays [10].

Another study conducted at the Naval Training Equipment Center by Ricard, Norman, and Collyer [9] concluded that "controllability gets worse with long delays, and good control is harder for the higher performance aircraft." Their research revealed that system controllability is affected by any type of time lags, including the first- and second-order lags examined in a previous Air Force study. However, it was found that "transport delays, which allow no system response for the duration of the delay, are most disruptive."

A succeeding study by Ricard and Harris [14] elaborated on the effects of transport-type delays. The researchers stated (as in the previous study) that when such delays are present, "the time between pilot control input and system response will increase. Poorer pilot control performance results." The reason for this degraded control performance, they go on to explain, is the reduction of stability margins due to the increased delay. "When pilots attempt to maintain a constant method of control in the presence of delayed feedback, they are forced to reduce their phase margins In the ... region of the spectrum where the gain vs. frequency curve ... crosses over from greater to less than unity gain, pilots like to maintain a phase margin of 25-45 degrees. Computer generation of images ... takes about 100 ms ... this time would reduce the pilot's phase margin by 17-28 degrees Human controllers exposed to delayed visual feedback will attempt to generate more of a phase lead for their control inputs and, failing this, will then reduce their crossover frequency and possibly increase low-frequency gain in order to minimize system

error." Again, these actions can be taken to be indicative of increased operator workload as described by McRuer.

Woltkamp, Ramachandran, and Branson [28] analyzed pilot control activity and pilot opinion ratings in a fixed-base helicopter simulator with variable transport delays. These researchers found that as delay was increased, "pilot control activity increased in the low speed, high gain tasks." Also, "the Cooper-Harper rating increased indicating degradation in perceived handling qualities. However, for the type of helicopter simulated, there was not a definite time delay at which the ratings changed abruptly. This indicates that for engineering design purposes, while it is desirable to keep the delay to the absolute minimum, there is sufficient flexibility in the design of the simulator to permit cost/capability tradeoffs."

Bailey, Knotts, Horowitz, and Malone [4] also observed this degradation of pilot opinion ratings with added transport delays. "For the two transport [aircraft] configurations ... (C-21 and C-141), pilot ratings appear to degrade at a constant rate with added delay For the two aggressively flown aircraft (F-16 and C-17), the rate of pilot rating degradation with time delay appears to be slightly higher than the transport vehicles, although not significantly so. For each aircraft, as the time delay became significant, control problems became evident with increasing tendencies toward overshoots, oscillations, and PIO [pilot-induced oscillations] To achieve any degree of pilot-vehicle performance with additional delay, each pilot adopted his own particular compensation techniques." While such compensation allowed the pilots to fly the simulated aircraft reasonably well, the necessity for using a modified

control strategy "may result in the acquisition of skills (e.g., pulsatile control) that are inappropriate for a system with small delays [29]", in particular, the actual vehicle.

Middendorf, Lusk, and Whiteley [30] performed a Fourier analysis of data obtained from a sidestep landing maneuver experiment performed in a fixed-base flight simulator. Their "power spectral analysis on lateral stick activity showed that power in a narrow band (0.4 to 0.5 Hz) increased as time delay increased" from 90 to 200 to 300 milliseconds. "As time delay increased, the man-machine system became less stable and less damped ... the subjects needed to make additional control inputs to correct for overshoot and degraded stability."

Researchers have found that the effects of delays vary between simulated vehicles due to their different characteristics. For example, Ricard, Norman, and Collyer found that while even small delays were found to be detrimental in the flying of high-performance aircraft, "when the subjects flew simulations of aircraft with better [more forgiving] handling qualities, longer delays had less of an adverse effect on performance [9]." The conclusion was drawn that "as we try to simulate more sophisticated aircraft or to teach the more complicated flying tasks, we might expect delays to be increasingly detrimental." Later work by Levison and Papazian [16] confirmed this observation; they found that "delay had a larger effect on [tracking] performance with the simulated 'F-16' than with the simulated 'C-141'." Allen and DiMarco [31] note that "ground vehicles typically have faster response dynamics than aircraft in terms of path control, and it is suspected that the problem may be even more serious for driving simulators."

Not only do delays of a given length affect the simulation of different vehicles differently, but delay effects have been found to vary with the operator's assigned task as well. In particular, Ricard and Harris [14] found that while "often no change is seen in the pilot's control of tasks ... such as free flight", performance of more demanding maneuvers is adversely affected by the system transport delay. Specifically, "the requirement of maintaining an orientation or position relative to an external object close to the simulated aircraft has caused pilots to induce oscillations when a significant delay was present." These oscillations occur "usually along the lateral axis, but sometimes along the longitudinal one as well [14]."

These problems had also been observed in experiments by Gum and Albery [24] during early formation flying evaluations performed in the Advanced Simulator for Undergraduate Pilot Training. They noted that "the pilots would close in on the lead aircraft but not maintain precise control of their position with respect to the lead ... the end result was usually an induced oscillation mode of control." They further commented that "the greatest impact of iteration rates and transport delays seemed to be in the control of aircraft roll position."

The above-described undesirable effects of system delays are not necessarily equal for all axes of operator control. For example, the experiments by Ricard, Norman, and Collyer using the TA4J Operational Flight Trainer and the Advanced Simulator for Undergraduate Pilot Training revealed that both simulators exhibited roll-axis instability during certain maneuvers, to the point (in the TA4J) "that flyers of the trainer tended to produce pilot-induced oscillations on the last leg of the

carrier approach task [9]." In each case the observed instability was attributed to the addition of CIG system delays to the delays already present in the dynamics processors. Of interest is the fact, pointed out by the authors, that while roll-axis instability was notable in these flight simulators, "almost no differences were seen between delay and no-delay conditions for the control of the pitch axis." Lusk, Martin, Whiteley, and Johnson [32] also found that altitude maintenance was not affected by a variation of primary display delay from 67 ms to 300 ms, "indicating that altitude control was not sensitive to delay." A possible reason for these observed differences in the effect of delays has been suggested by Merriken, Riccio, and Johnson [11]. According to those authors, the fact that "the effects of delay consistently were greater for roll axis control than for pitch axis control" in certain experiments "may be due, in part, to greater system bandwidth of the roll axis."

Simulator Sickness

Hettinger, McCauley, Cook, and Voorhees [33], reporting on the first meeting of the NASA Ames Simulator Sickness Steering Committee, defined simulator sickness as "the constellation of signs and symptoms of motion sickness and related perceptual aftereffects that occurs in ground-based vehicular simulators. The simulator sickness syndrome is characterized by adverse symptomatology experienced either during or after exposure to simulated motion scenarios that would not produce sickness in the actual vehicle." The symptoms may include disorientation, dizziness, headache, and nausea while operating the simulator as well as prolonged nausea,

fatigue, visual dysfunction and even flashbacks for up to ten hours after the simulated experience [34].

Some of the problems associated with simulator sickness include safety and health concerns for users, effects on training effectiveness, validity of R&D data, and impact on the scheduling and utilization of simulator-trained personnel. "Pilots who suffer from severe symptoms may need to be removed from flight duties temporarily. Their sudden unavailability detracts from flight training schedules and, perhaps, from general flight readiness [33]." The Steering Committee report went on to mention that "the commonly accepted theory of motion sickness is the sensory conflict theory, sometimes known as 'neural conflict' or 'neural mismatch'." Applied to real-time, man-in-the-loop simulators, this implies that "a temporal and/or spatial mismatch of information about one's orientation or motion through space", possibly due to the delayed onset of visual and/or motion cues, may result in an increased tendency for operators to experience symptoms of simulator sickness.

Kennedy, Allgood, and Lilienthal [35] explored a number of factors believed to contribute to simulator sickness, which has been widely observed "in both fixed- and motion-base devices, and in devices with a variety of projection and image generation techniques." One of the chief contributors to simulator sickness was lag, or lack of synchronization, in visual and/or motion stimuli presented to the operator. "Computational limitations generally produce temporal lags between operator control input and subsequent changes in position as indicated by the visual display and motion base It is known that lags may cause pilot-induced oscillations which can

have two consequences: 1) overtax the visual system and create dynamic visual distortions, and 2) produce nauseogenic inertial energy around .2 Hz." Casali and Wierwille also stated that among "the major elements of vehicular simulators that are believed to contribute to simulator sickness" are various "system lags and delays (transport, exponential and second-order lag, phasing between visual and motion update) [1]."

Frank, in his doctoral dissertation [36] and in a 1988 article with Casali and Wierwille [8], presented the results of experiments in a moving-base automobile driving simulator which explored varying amounts of transport delay in both the CIG visual and motion feedback systems. The driving performance of male and female subjects, in terms of steering wheel reversals and yaw standard deviation, was measured for various combinations of visual and motion system transport delays including minimum delay, equal added delays, and conditions of visual information leading motion and vice versa. The results clearly indicated that "visual and motion system delays are detrimental to both an individual's control performance and well-being [8]." Furthermore, the conclusion was reached that "visual delay is far more disruptive to a simulator operator's control performance and physical comfort than is motion delay." In attempting to explain this finding, the authors hypothesized that, with experience, individuals develop filtering mechanisms for disregarding unimportant visual and motion-related information, and further, that "under normal everyday driving conditions, visual cues provide the primary information for vehicle control and motion cues provide the secondary information."

CHAPTER 3

MEASURING TRANSPORT DELAY

It is recognized that transport delay causes problems in man-in-the-loop simulators and that the problems increase with increased delay. It is therefore important to be able to quantify simulator delays. One must be able to measure delay to know the extent of the problem and, perhaps even more importantly, to correct it if possible. The literature describes two basic approaches to measurement of simulator delays: examining the time history of the response to a command given at a particular instant (the time-domain approach) or inferring the delay from phase measurements taken from the system in response to a sustained sinusoidal input (the frequency-domain or steady-state approach). Each method has been described in various papers as applied to several different simulators. To the author's knowledge, however, no reference heretofore has compared the results obtained from applying both delay measurement approaches to the same simulator. Details of the application of both delay measurement strategies, in general and with respect to the UCF Driving Simulator, are given in this chapter.

Time-Domain Approaches in the Literature

Butrimas and Browder [37] performed a detailed "cue synchronization study" of the Naval Training Equipment Center's Visual Technology Research Simulator. They measured "linkage throughput delays" (transport delays) over several transmission routes, including "cockpit control to CIG input and to CIG video output" as part of this study. According to the researchers, the two throughput delays related to computer image generation "were determined by first measuring the total throughput from control stick to video output, and then by measuring the segment from CIG input to video output. The remaining ... segment ... was then computed by subtracting the partial throughput from the total throughput." For their purposes, "video output was defined as the presence of video signal in the first horizontal line of the raster (the first pixel of CIG video)." However, when stating a figure for overall system response, "an additional time period to generate one TV field (17 msec) was added as necessary for reaching a visual cueing threshold" or minimum amount of video information usable by the operator for control purposes.

The system time delay measurements were made using the same software actually used by the simulator during run time. The one exception to this was that "the flight computer software was modified to cause the aircraft to instantly change position ... upon recognition of change of polarity of control stick input [37]." The "aerodynamic lags were removed ... so that all transport delay measurements reflected only computer and linkage related throughput lags." The results of the experiment produced a range of measured transport delay times due to the "walking"

or asynchronicity of control stick inputs with respect to sampling time. Since the computations were performed at a 30 Hz rate, "the difference between minimum and maximum throughput was 33.3 milliseconds." Their overall timing measurements were thus "referenced to an average between worst case ... and best case Thus, an average transport delay time is defined as having tolerance of ± 17 msec."

Woltkamp, Ramachandran, and Branson [28] described the measurement of time delays in a helicopter simulator in response to step control inputs. In order to measure overall system transport delay, the system software "was modified so that any detected change in the stick input resulted in an in-plane rotation of the visual of 90 degrees [to face an area of the scene containing a contrasting color]. The aero model would continue to integrate a response, but the output was ignored since the model delay was not being measured as part of the hardware delay." To measure output changes, "a sensitive photo sensor was built to detect response of the visual system by sensing the change in pixel brightness. Analog output from the photo sensor device was assigned to a strip chart recorder channel alongside the controller [stick] analog signal channels." Using this method, the researchers found the total system transport delay for a McDonnell Douglas helicopter research simulator to be approximately 87 milliseconds.

McFarland and Bunnell [38] performed a detailed analysis of time delays in a moving-base flight simulator at the NASA Ames Research Center. In their paper, the authors discussed some time-domain techniques for measurement of delay in a real-time simulator environment. Perhaps more importantly, they presented a

theoretical model (called the equivalent systems model or ESM) for evaluating the delay contributions from the various simulator components and combining them to get a valid measure of overall system delay. The model developed can be used to analyze multirate as well as single-rate simulations. Z-transform representations were derived for each component; the overall "system model may be created as the product of individual simulation components expressed as z-transforms,

$$W(z) = \Pi W_i(z) \quad (3.1)$$

Because of this relationship, the individual phases and component time delays are additive. This permits the independent investigation of various delay sources in flight simulation."

McFarland and Bunnell took issue with methods that had been used by some previous investigators, for example, the use of step inputs to test system time delays. While convenient to generate, step functions are not typical of operator inputs; their use "constitutes a major difference from the actual simulation environment, where inputs are generally sampled such that significant activity does not occur between sample intervals" Techniques for measuring delay which involve removing the dynamics equations and their integrations from the software also distort the overall delay picture, the authors observed, since various numerical integration methods contribute to overall phase lead or lag. Finally, the authors concluded that "although procedures for handling multiple loops have been established, the use of a multirate model is not encouraged because of the I/O time-delay penalty and because of problems with aliasing [38]."

Steady-State Approaches in the Literature

Researchers at the U. S. Air Force's Armstrong Aerospace Medical Research Laboratory devised a method for estimating system transport delays by measuring phase relationships rather than time lag from a given input. Johnson and Middendorf [3] discussed some of the drawbacks of delay measurement techniques that involve measuring the system response to a step input (as described above). "Methods that rely on measurement of transients can give misleading results when applied to all-digital simulations. For instance, a step response measurement on a CGI could miss the delay due to the holding effects of a multi-refresh image which become apparent only in steady-state excitation. Transient methods also suffer from the intrinsic dependence on the definition of the stimulus and response. Frequency domain measurement techniques, on the other hand, provide more consistent results." Merriken, Riccio, and Johnson [11] added that "frequency-domain techniques have been used for verification ... since it measures any combination of dynamic and pure delays and can be easily replicated."

Delay measurement using the Armstrong Laboratory steady-state technique was a relatively simple procedure. "To measure the transport delay, several test frequencies [chosen in the range of normal pilot control activity] were substituted for stick command inputs. A photocell was used to measure the differences in display luminance. The phase difference between the input to the aircraft dynamics and the output measured by the photocell was determined by a frequency analyzer The phase lag due to the aircraft dynamics was subtracted from the measured phase

difference at each of the test frequencies. The transport delay was then calculated by dividing the adjusted phase difference by the product of the test frequency and one revolution [11]." Johnson and Middendorf [3] gave a more detailed description of the measurement process, including the method for determining the phase lag of the aircraft dynamics, the modeling of the display monitor, and the modifications made to the display software to provide suitable contrast for light output measurement. Construction of the optical sensor was detailed by the authors, as were the characteristics of the primary test instrument (the frequency response analyzer). In particular, they noted that the analyzer's "excellent harmonic rejection capability meant that fundamental response could be accurately read through a non-linear system response." The results of the steady-state delay measurement experiments were very consistent across the range of input frequencies used; the delay times computed for the simulator of interest ranged only from 119 to 122 milliseconds. The researchers asserted, based on their success with this approach, that "the exacting nature of frequency domain delay measurement is proving invaluable in verifying and accurately quantifying simulator delay."

Time-Domain Delay Measurements in the UCF Driving Simulator

Considerable work, including software development for the host and image generator computer systems, was required in order to adapt the time-domain delay measurement techniques described in the first section of this chapter to the

particular environment of the UCF driving simulator. The following subsections discuss the details of the measurement procedure and the results obtained.

Time-Domain Delay Measurement Procedure

The time-domain delay measurements were conducted in a similar manner to those described in the references [28] [37] [38] cited in the first section of this chapter. Most of the changes that had to be made were prompted by the difficulty of providing a step input via the rotary position encoder interface normally used to sample driver steering commands. If these changes are understood and taken into account in determining numerical values for system transport delay from the test results (see the subsequent section on analysis of delay measurements), accuracy is not compromised.

To facilitate the time-domain transport delay tests, the host computer program was modified to simulate the occurrence of a step steering input command (see Figure 5). For test purposes, program logic in the analog input routine ignores the value read from the steering position encoder and (normally) sets the value of the steering input (STEERW) to zero. Periodically, however, (at an interval chosen for easy triggering of an oscilloscope or logic analyzer) the value of STEERW is set to 1. Logic in the main program examines the STEERW value passed by the I/O routine and, based on it, sets the simulator position coordinates. The two sets of coordinates, which are passed to the XTAR IG for display, were chosen such that the viewpoint faces either an all-black or all-white polygon. (The vehicle dynamics

code, which ordinarily determines the motion of the simulated vehicle, is bypassed for this test [37] [39].)

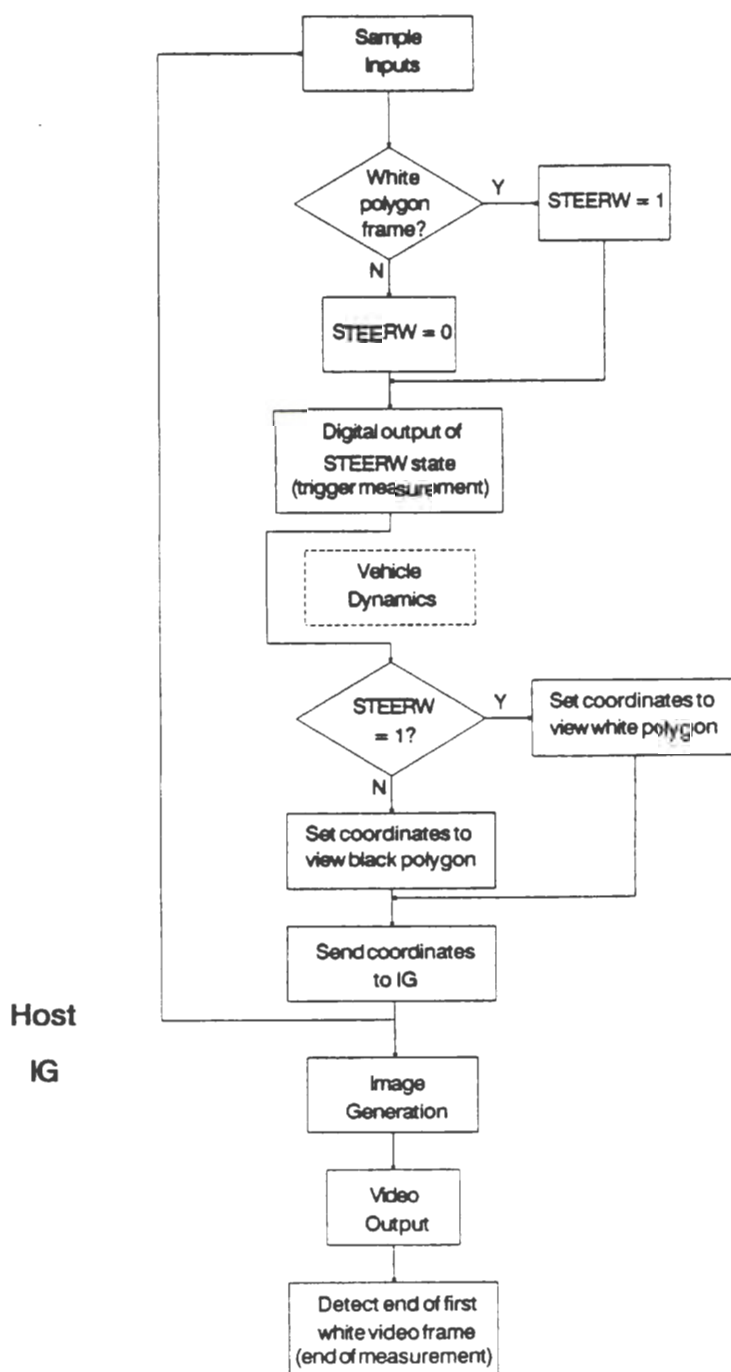


Figure 5. Flow Chart for Time-Domain Delay Tests

At the beginning of each frame, just before the steering wheel position encoder is read, the input routine updates a digital output line with the state which the STEERW input is about to assume. This digital output line mimics the state of the input command being "sampled" and thus provides a timing reference for comparison to the video signals for the corresponding video frame(s) during which the screen turns white. Delay is recorded as the time between the leading or positive-going edge of the pulse from the host computer I/O board and the trailing edge of the first corresponding pulse on the red (or blue or green) video output. This latter time corresponds with the end of the first frame of white video to be displayed. The delay measured in this manner thus includes the time taken by the input routine during and after the sampling of the steering input, the host-IG communications, and image generation and video processing by the XTAR system. The delays due to expression evaluation and numerical integration for the vehicle dynamics and the (average) $T/2$ delay in sampling an external step input are not included in these measurements. A timing diagram which illustrates the time-domain delay measurement process is shown in Figure 6.

The time-domain transport delay measurements were made using a storage oscilloscope as well as a logic analyzer capable of sampling digital inputs at a high rate. The results for each visual database yielded a range of transport delay times due to the asynchronous host-IG interface. The results are discussed further in the next subsection.

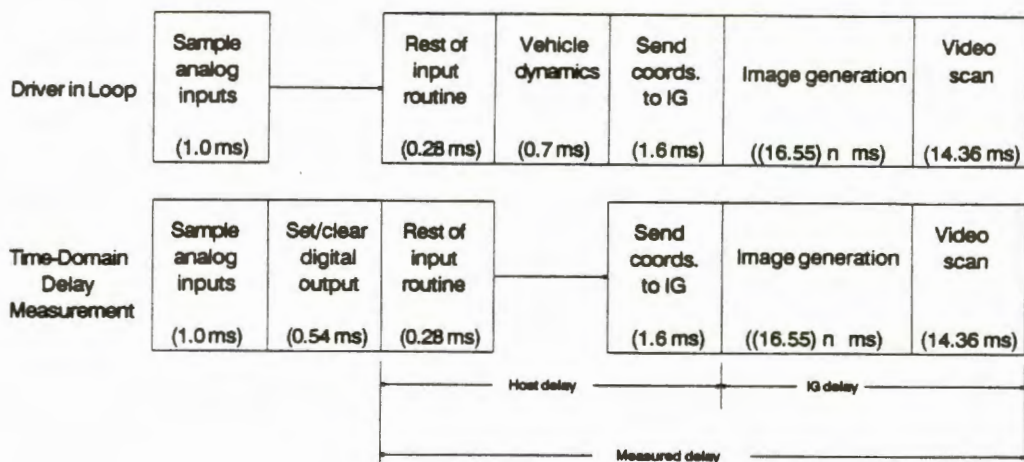


Figure 6. Timing Diagram for Time-Domain Delay Tests

Finally, in order to get a more precise picture of the operation of the simulator in real time, measurements were made of the duration of various tasks performed by the host computer real-time program. This was done using the same instruments and time-domain techniques employed to measure overall delay. The results, which are reported below, show (among other things) that measurement-induced delays, including the time taken to send out the pulse from the host's I/O board and subsequently to force the IG coordinates, are short enough to be virtually insignificant when compared with the overall system transport delay being measured.

A more detailed analysis of the total system delay is presented in the final section of this chapter.

Results of Time-Domain Delay Tests

Measurements of the duration of various I/O operations and software routines used by the host computer real-time program were made using time-domain techniques. Since the results pertain to the measurement and analysis of overall simulator delay, they are summarized here for reference:

Table 1.

Duration of Host Computer Operations and Routines

Operation or Software Routine	Time Consumed (ms)
Digital output	0.54
Analog output	0.54
Digital input	0.24
Analog input	0.33
Steering encoder input	0.01
Input routine (complete)	1.27
Vehicle dynamics routine	0.69
CIG communications routine	1.60

The complete set of frame tasks (including sound generation, calculation of steering wheel torque, and other operations which are done after the transmission of vehicle coordinates to the IG) takes 5.46 to 7.16 milliseconds to complete. Since

the allowed frame time is $T = 1/60$ second = 16.67 ms, there is significant spare frame time available for adding detail to the vehicle dynamics model, performing any additional I/O operations which might be needed, or implementing delay compensation in the software. The I/O operations, in particular, can be seen to be considerably more time-consuming than numerical computations. This implies that the most effective way to improve performance in the host system might be to acquire faster analog and digital I/O boards (rather than a faster CPU or floating-point coprocessor).

Time-domain transport delay tests, using the simulated step input technique previously described, were run for four IG database cases ranging from the simplest (a 16-polygon test database) to the full driving simulator demonstration database with stationary flat and three-dimensional objects plus two movable objects (other vehicles).

For each experimental condition (database) used in these tests a range of observed delay times was recorded. This is due to the fact that the host/IG communication is asynchronous. The IG can "just miss" the coordinate update from the host and have to sample it again on its next opportunity (an integral number of video frames later, resulting in maximum transport delay), can read it just after it is updated (resulting in minimum transport delay), or can get the new coordinates anywhere between these two extremes. This causes the variation in the delay times shown in Tables 2 and 3 below. In addition, sampling the driver controls (including STEERW) causes an additional variation in transport delay which does not show up

in the measured values. At the original host computer frame rate (20 Hz) the total delay can be as small as the shortest time observed (both host and IG sample at "just the right time") or as large as the longest time observed plus 50 milliseconds (representing worst-case timing for both the host and IG). The average additional delay due to sampling is $T/2$ [37] or, in this case, 25 ms. The new, higher host frame rate of 60 Hz not only reduces the magnitude of the average system delay (because $T/2$ is smaller) but also reduces its variability.

System transport delay times were originally measured using a Tektronix 5103N storage oscilloscope. Thirty measurements were obtained for each of the four cases described above. The results obtained for each database from simplest to most detailed are shown in the following table:

Table 2.

Transport Delay Measured With Oscilloscope

Database Configuration	Measured Transport Delay (milliseconds)			
	Minimum	Maximum	Midpoint	Average of 30
Test (16 polygons)	50	66	58	59
Flat only (ground, road, background)	84	116	100	101
Flat plus stationary 3-D (buildings, trees)	117	164	140	138
Flat plus all 3-D (includes moving objects)	153	219	186	192

In order to confirm these results and establish the delay times with greater precision, the time-domain delay measurements were repeated for all four databases using a Tektronix 1230 Logic Analyzer. An average of 30 delay times was computed for each database. Additional runs were also made in order to determine the extremes (maximum and minimum) of delay for each graphical database. The results of these trials are summarized in Table 3:

Table 3.
Transport Delay Measured With Logic Analyzer

Database Configuration	Measured Transport Delay (milliseconds)			
	Minimum	Maximum	Midpoint	Average of 30
Test (16 polygons)	50.0	65.9	57.95	58.42
Flat only (ground, road, background)	82.9	115.6	99.25	100.27
Flat plus stationary 3-D (buildings, trees)	116.0	164.8	140.40	144.67
Flat plus all 3-D (includes moving objects)	149.8	214.8	182.30	180.84

By comparing the preceding tables, it can be seen that the measurements agreed within 1 to 2 milliseconds for the shorter delays and 4 to 5 ms for the longer delays. In fact, the midpoints between the longest and shortest observed delays agree within less than 1 ms except for the last (longest delay) case. For comparison, we can

develop expected values for delay based on the measured duration of the software operations and the frame time of the IG, as follows:

The remaining duration of the input routine (during and after the sampling of the steering input) is approximately $(1.27 - 3(0.33)) = 0.28$ milliseconds. The vehicle dynamics routine is bypassed (0 ms execution time). Communication to the IG takes 1.6 milliseconds. The IG takes an integer number of video frames at 16.55 ms each (operations are synchronized to the 60.42 Hz, non-interlaced display) to draw the scene corresponding to each set of coordinates it reads. The last frame, however, is considered to be complete for measurement purposes when the last pixel of the display is updated; the vertical retrace period (2.19 ms) is not part of the transport delay. Thus the length of the last frame is $(16.55 - 2.19) = 14.36$ ms. Taking into account that the duration of each delay, while not random, should be more or less uniformly distributed between the maximum and minimum times, the actual figures for the IG delay alone may be computed as:

16-polygon test database: 47.46 to 64.01 ms (3 to 4 frames); average = 55.74 ms.
DTS database (flat only): 80.56 to 113.66 ms (5 to 7 frames), average = 97.11 ms.
DTS database (full/no cars): 113.66 to 163.31 ms (7 to 10 frames), avg. = 138.49 ms.
DTS database (full+2 cars): 146.76 to 212.96 ms (9 to 13 frames), avg. = 179.86 ms.

Adding in the extra 1.88 milliseconds for the remainder of the input routine (after sampling) and the communications routine along with the above times for image generation, the theoretical average values for transport delay using this method would be (in milliseconds) 57.62, 98.99, 140.37, and 181.74 respectively.

These figures are in excellent agreement with the values shown in Tables 2 and 3, and are used below to help determine the values of system transport delay to be compensated for by the algorithms developed in Chapter 4.

Steady-State Delay Measurements in the UCF Driving Simulator

Some work was required in order to adapt the steady-state delay measurement techniques described earlier in this chapter to the UCF driving simulator and available test equipment. For instance, hardware and software had to be developed to allow tracking of the video display output of the XTAR computer image generation system in order for phase measurements to be taken. In addition, a workable method of providing a sinusoidal input to the system had to be devised, since the direct connection of an electronic function generator to a system A/D input (as done by previous researchers [3] [37]) was not practical in the UCF simulator. The details of the measurement procedures and test setup, and the results obtained, are discussed in the following subsections.

Setup and Procedure for Steady-State Delay Measurements

To apply steady-state delay measurement techniques to the UCF Driving Simulator, means had to be devised to allow instruments to monitor the video output of the image generator and to provide a sinusoidal input to the system while constraining the motion of the simulated vehicle along the boundary of polygons of

contrasting colors. The steps taken to solve these problems and perform the delay measurements are outlined in the following paragraphs.

Video Detection Circuit. In order to monitor the CIG video output using an oscilloscope, spectrum analyzer or similar equipment, it was necessary to devise a circuit that would convert the video signals provided by the XTAR board set into a single signal indicating whether a white or black display is being shown at any given time. The available signals were the red, green, and blue analog video outputs as well as /CSYNC (composite synchronization pulse output), which goes low for both horizontal (line) and vertical (frame) synchronization. A further constraint was that the analog video must be sampled somewhat prior to the vertical retrace pulse on /CSYNC, as this occurs at a time when the color signals are not active.

The video detection circuit was designed in three parts: a smoothing filter for the red video signal (green or blue could have been used interchangeably since the test databases use a grayscale color palette) to eliminate the effects of horizontal retrace; a sample-and-hold to capture the level of the filtered video; and the control/timing circuit for the sample-and-hold.

The sample-and-hold consists of a capacitor that is charged through a 4066 CMOS analog switch (chosen for low ON resistance and very high OFF resistance). The switch is closed by a control pulse (SAMPLE command), gating the signal to be sampled (in this case the filter output) onto the capacitor, which is connected to an operational amplifier voltage follower for buffering.

The filter circuit was configured as two inverting integrators in series. It was found experimentally that a filter cutoff frequency of 159 Hz ($R = 10\text{ K}\Omega$, $C = 0.1\ \mu\text{F}$) worked reasonably well.

The control/timing circuit for the sample-and-hold consists of three retriggerable monostable multivibrators or "one-shots" (each 1/2 74LS123). The first one-shot is retriggered by each horizontal retrace pulse on /CSYNC until the longer, vertical retrace pulse occurs. At this time the one-shot changes state and triggers a second one-shot which produces a pulse of sufficient duration to last until nearly the end of the next frame. The time-out of this pulse then triggers a third one-shot which generates the SAMPLE pulse as its output. Component values for proper pulse lengths were calculated using manufacturer's specifications; potentiometers were used to make fine adjustments to the circuit to allow for sampling as close as possible (within 1-2 milliseconds) to the end of each video frame. Thus the "visual cueing threshold" [37] for the steady-state delay tests is approximately the same as for the step-input tests (the end of the last visible line of the video frame). A circuit diagram for the video capture circuit is shown in Figure 7.

Host Computer Software Modifications. Ideally, the steering input to the simulator during steady-state delay tests would be provided through the same input interface used by a human subject during real-time simulator operation; however, due to the nature of the signal produced by the rotary position encoder, this proved to be impractical. The next choice would be to provide input from a function generator

or similar instrument via an A/D port such as those used to sense accelerator and braking inputs. This approach, however, presented difficult if not insurmountable problems with initializing the simulation and ensuring that the "car" would stay aligned with the "road", since the amplitude and frequency of the input could not be held precisely constant (nor could the road be moved in real time).

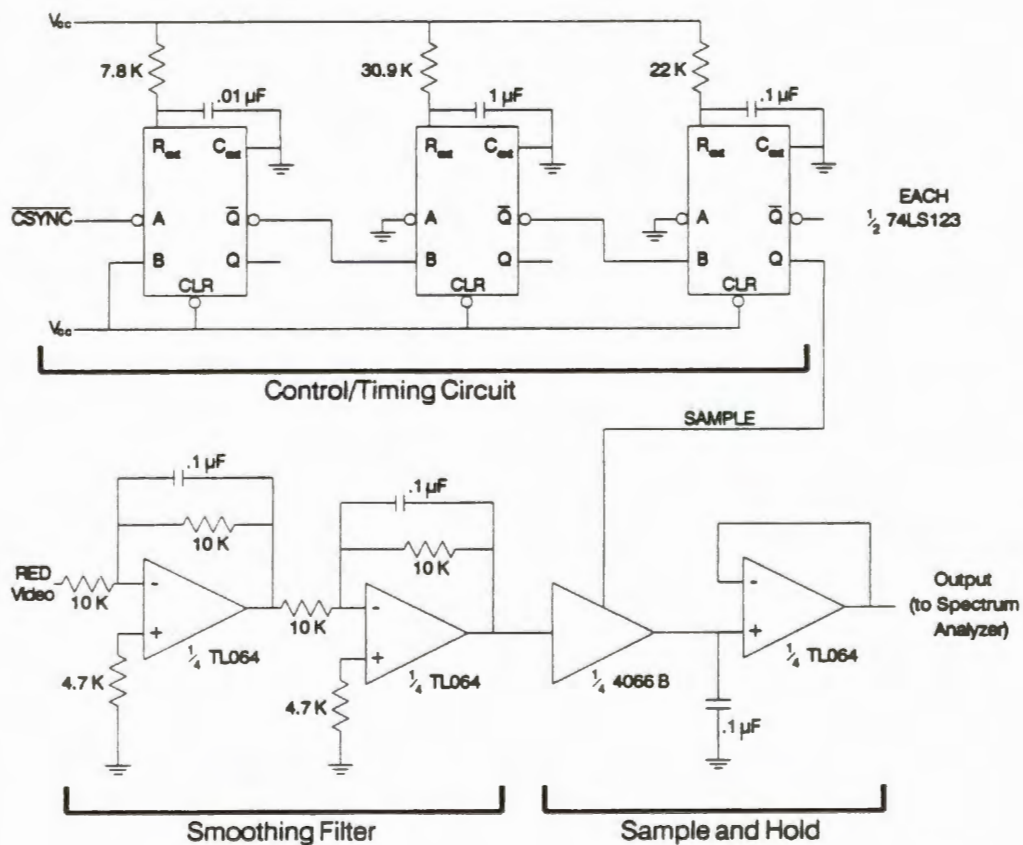


Figure 7. Video Output Detection Circuit

Initializing the sinusoid at a precise value and phase would also have been a problem with an external input. Offline testing of the dynamics program showed that good steady-state behavior of YCAR was quite sensitive to the initial value (and thus the phase angle) of STEERW; if initial conditions were not set exactly, the simulated automobile traveled away from the road at an angle. (See Figure 8 for an example of this effect with a 2 rad/s steering input at a forward speed of 60 ft/s.) This masked the sinusoidal oscillations of vehicle lateral position, which (in order to facilitate delay measurements) should occur about the inertial Y value corresponding to the color boundary. This was not mentioned as a problem in the articles describing flight simulator delay measurements [3] [11]; in those cases delay could presumably be observed by rolling or pitching the aircraft against a distant horizon. In the UCF simulator environment, however, there was a need for precise control of the vehicle's absolute position in order to perform these measurements.

Given the problems associated with applying an external sinusoidal input to the system for delay tests, it was decided to modify the host computer program to simulate the occurrence of a sinusoidal input. The current value of the "input" sinusoid is calculated by the program just before the analog inputs are sampled; the calculated value is then substituted for the value actually obtained from the encoder interface board. (The board is still read in order to change the I/O timing as little as possible.) Calculating the sinusoidal function makes the "input" deterministic, an important modification to the steady-state delay measurement technique as described in the literature [3] [11]. This innovation allowed the simulated vehicle to be

stabilized over the road color boundary, which exists at a particular Y value in the world coordinate system, rather easily.

Initial conditions on lateral position and the driver steering input that would provide proper operation of the vehicle (sustained oscillations about a reference lateral position) for the steady-state delay test program were determined iteratively using an offline test program. Values were determined for operation at forward speeds of 30, 60, and 90 ft/s (20.5, 40.9, and 61.4 mph) and steering input frequencies of 1, 2, and 6.283 rad/s (0.1592, 0.3183, and 1 Hz). These frequencies were chosen as representative of the spectral range of steering activity needed for most driving tasks; the several combinations of input frequency and velocity would provide a good variety of test cases for steady-state delay measurements.

An estimate of approximately two radians per second for the driver/vehicle system crossover frequency was developed from various sources in the literature [14] [30] [31] [40]. McMillan [41] also noted research results indicating that pilots are most sensitive to changes in dynamics in the 2-3 rad/s range. These findings lent further support to the optimization of delay compensators for this band of frequencies. Since compensation algorithms were to be optimized for the region of crossover, 2 rad/s was chosen as one of the test frequencies. The lowest test frequency (one radian per second) is in the system pass band — the range of frequencies in which operators provide the bulk of the input for controlling the vehicle. Finally, the 1 Hertz test frequency was chosen to represent inputs near the upper limit of the driver's input spectrum [42].

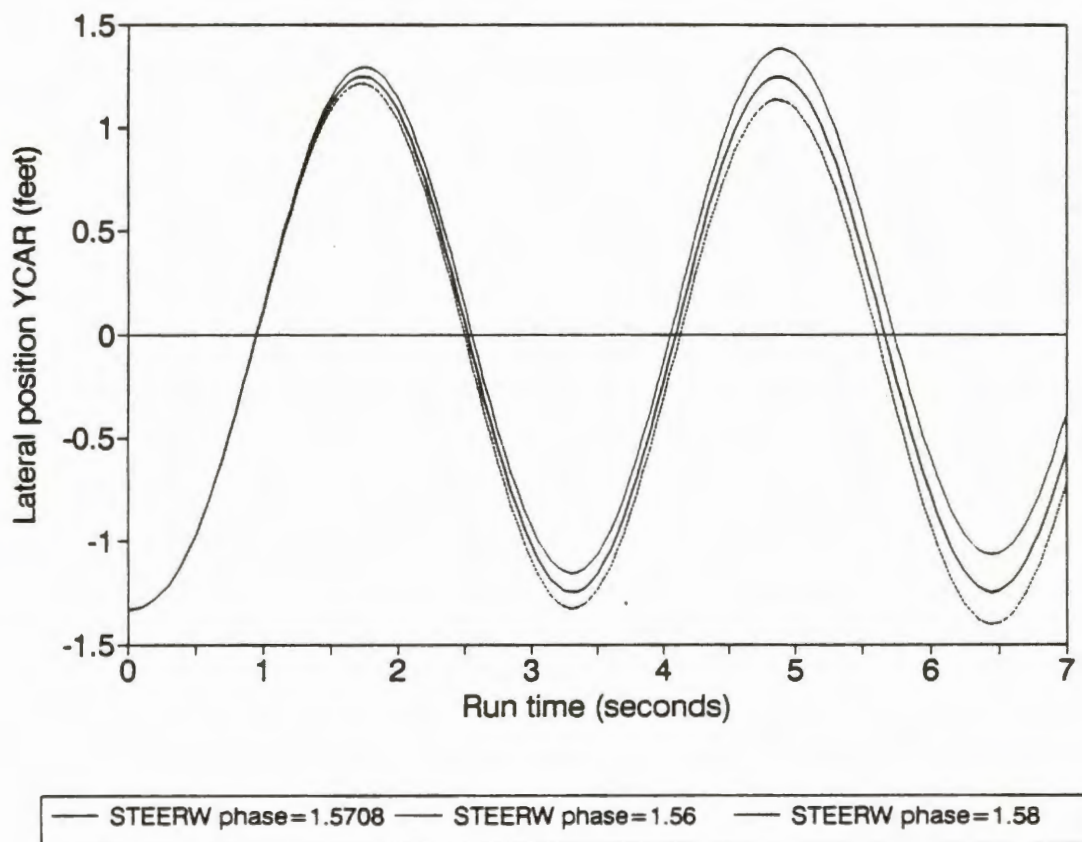


Figure 8. YCAR Response with Various Values of STEERW Phase

In order to perform steady-state delay tests on the driving simulator, one channel of the phase-measuring equipment was connected to a D/A converter output from the host computer (this required a program modification) representing the current value of the STEERW "input" signal. Of course, the zero-order hold effect of the D/A output (a lag of $T/2$ seconds on average, which is small but not insignificant compared to the size of the total system delay) must be accounted for in calculating delay from the measurements obtained in this manner. The YCAR value was also

scaled and output through a second D/A port to facilitate phase measurements on either the dynamics model or IG alone. (See Figure 9 for a flow chart illustrating the measurement procedure.)

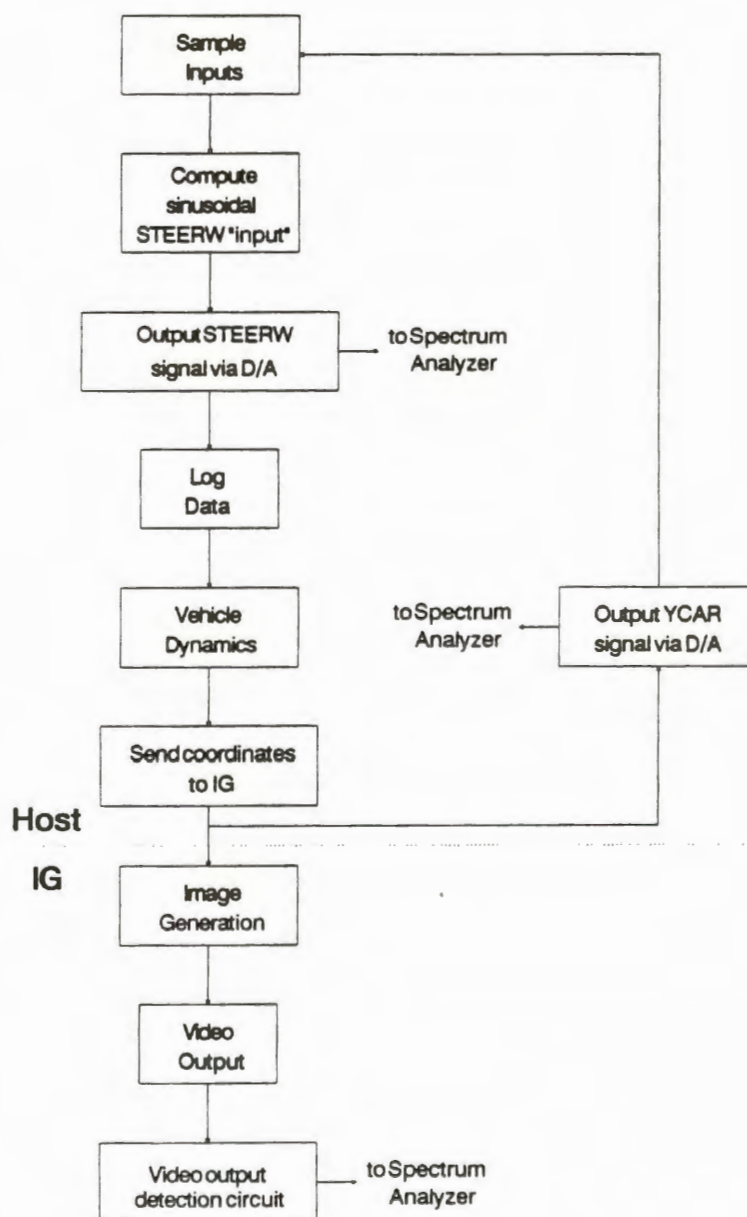


Figure 9. Flow Chart for Steady-State Delay Tests

The additional operations required to implement the steady-state delay tests (computation and analog output of the simulated STEERW input, forcing UCAR to a constant value, analog output of YCAR) take only about one millisecond. This measurement-induced delay is small compared to the magnitude of the total system delays to be measured (see the results of the time domain delay measurements above) and the resolution capabilities of the instruments used in the tests. A timing diagram detailing the steady-state measurement procedure is shown in Figure 10.

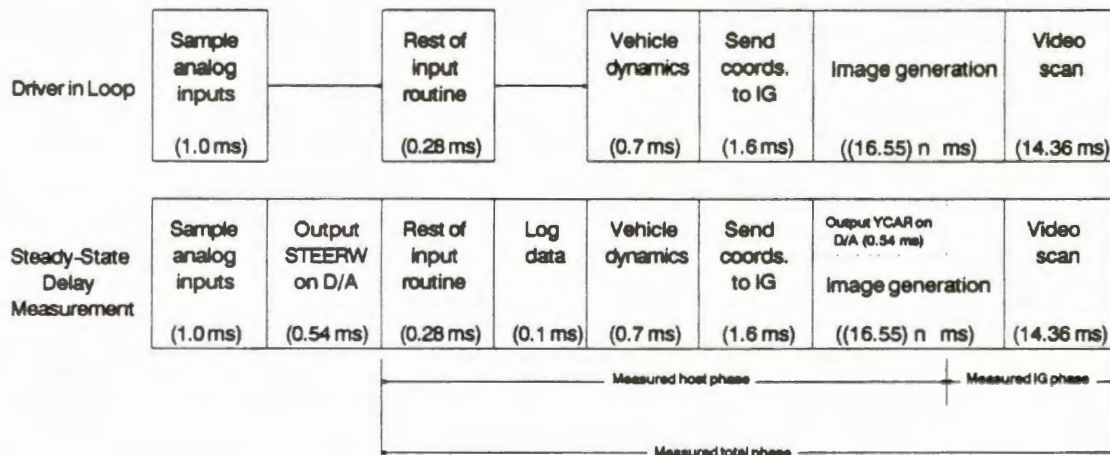


Figure 10. Timing Diagram for Steady-State Delay Tests

One other modification to the program, alluded to above, was made in order to facilitate the delay tests: the integration of UDOT to UCAR (vehicle forward velocity) is performed to preserve timing relationships, but UCAR is subsequently fixed at the chosen value (30, 60, or 90 ft/s) for experimental purposes. The final version of the host computer steady-state delay test program was tested for all combinations of UCAR and steering input frequency and found to operate satisfactorily (i.e. oscillate about $YCAR = 0$ without drifting off line during a period of time sufficient to make phase shift measurements) in all cases.

Image Generator Software and Database Modifications. Some changes also had to be made to the original demonstration databases and IG software in order to facilitate the steady-state delay measurements. These changes included widening the polygons comprising the chosen straight section of roadway to allow for the excursions of YCAR as well as changing the colors of the ground, roadway, lines, and several three-dimensional objects in this area so that the display would be completely white or black at all times during the test. This last constraint also required that the viewpoint be pitched down toward the ground, lowered, and (only when the computed position of the vehicle was very close to the color boundary) adjusted slightly away from the boundary. The CIG program performs these functions and also translates the XCAR and YCAR position before the scene is drawn in order to place the car on the correct section of road. In the IG as well as the host, the time

required for these operations is negligible compared to the overall system transport delay.

In order to demonstrate the ability to measure delays of varying length and compare these measurements against the time-domain delay measurements, two versions of the simulator driving database were used: the full three-dimensional demonstration database (including terrain, roads, stationary objects, and two movable "cars") and an identical road/stationary object configuration without the moving objects.

Measurement Procedure. Before the delay measurements were performed, an offline version of the steady-state test program was used (along with a spreadsheet program for graphing) to determine the actual vehicle dynamics model phase lags for each test condition (input frequencies of 1, 2, and 6.283 rad/s coupled with UCAR values of 30, 60, and 90 feet per second). First, the output of the model iterated at 60 Hz was checked against a 1000 Hz solution for the 60 ft/s cases; the two graphs were found to be essentially identical in phase for all three frequencies of interest. (See Figure 11 for an example with a 1 Hz steering input.) The 30 and 90 ft/s cases were thus tested only at the 60 Hz frame rate (the 1000 Hz, "pseudo-continuous" case was assumed to be identical). Results are discussed in the following subsection.

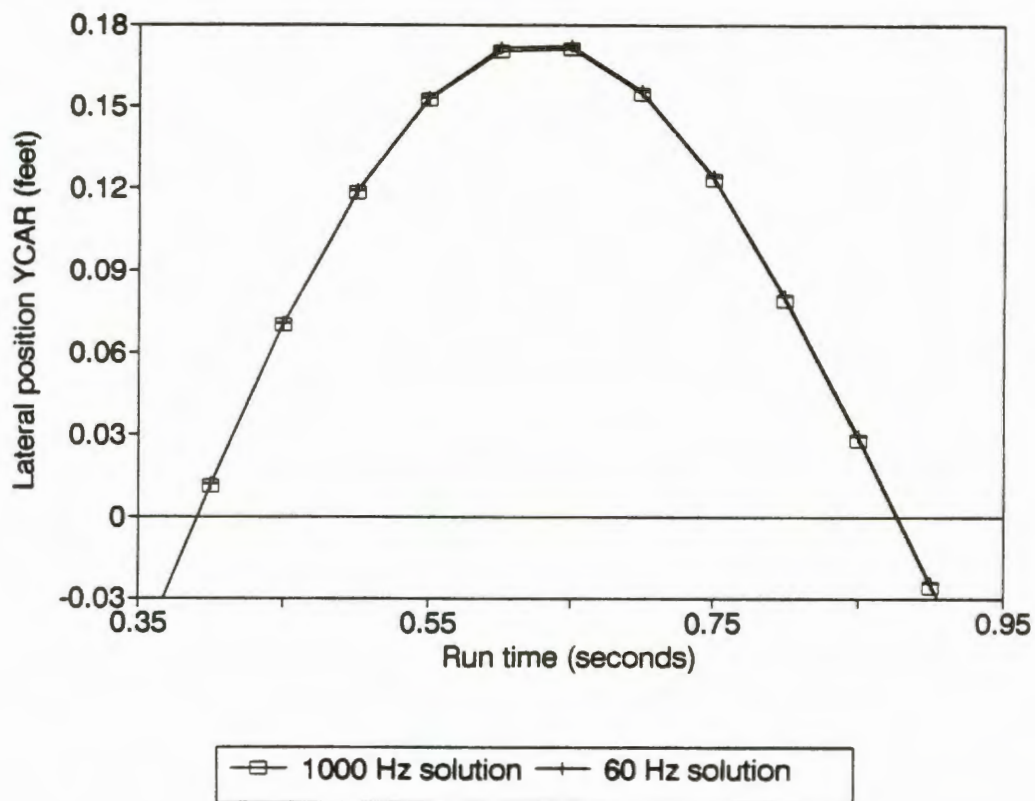


Figure 11. Comparison of 1000 Hz and 60 Hz Solutions of Vehicle Dynamics Model

Following these preliminary tests, the delay measurements were performed using the host and IG software described above. Some phase measurements were attempted using a storage oscilloscope, but it proved very difficult to get good results; and in any case, the displays obtained were only instantaneous "snapshots in time", more like the step input delay measurements made previously (which had to be made repeatedly and then averaged) than the steady-state measurements described in the literature.

The spectrum analyzer ultimately used for the delay measurements samples two signals over a period of time which varies depending on switch settings but, for the

investigations reported here, ranged from less than one minute to about two minutes. The resulting display shows a phase transfer function over some frequency range of interest. It was possible to read out the phase difference at a particular frequency (in this case, the test frequencies of 0.16, 0.32, and 1.00 Hz) by positioning a marker on the trace at that point. Phase measurements (from simulated STEERW input to model output (YCAR), STEERW input to CIG output, and model output to CIG output) were made for all combinations of the test parameters (forward speed and input frequency) with each of the IG database configurations. It proved necessary to average four to six observations for each measurement because of some variation (approximately ± 1 to 2 degrees) observed in the displayed readings. These average readings, along with the delays calculated from them, can be found in Tables 5 through 8 in the following subsection.

Results of Steady-State Delay Tests

The following results were obtained from offline simulation to obtain the vehicle dynamics model phase characteristics at the same conditions used in the steady-state delay tests. The YCAR (lateral position) responses for each case were recorded, graphed, and examined using a spreadsheet program. (Typical graphs for 1 rad/s and 1 Hz steering inputs at UCAR = 60 ft/s are shown in Figures 12 and 13.) The values obtained are shown in Table 4 for comparison to the values obtained via use of the spectrum analyzer in phase measurement mode (see Tables 5 and 6).

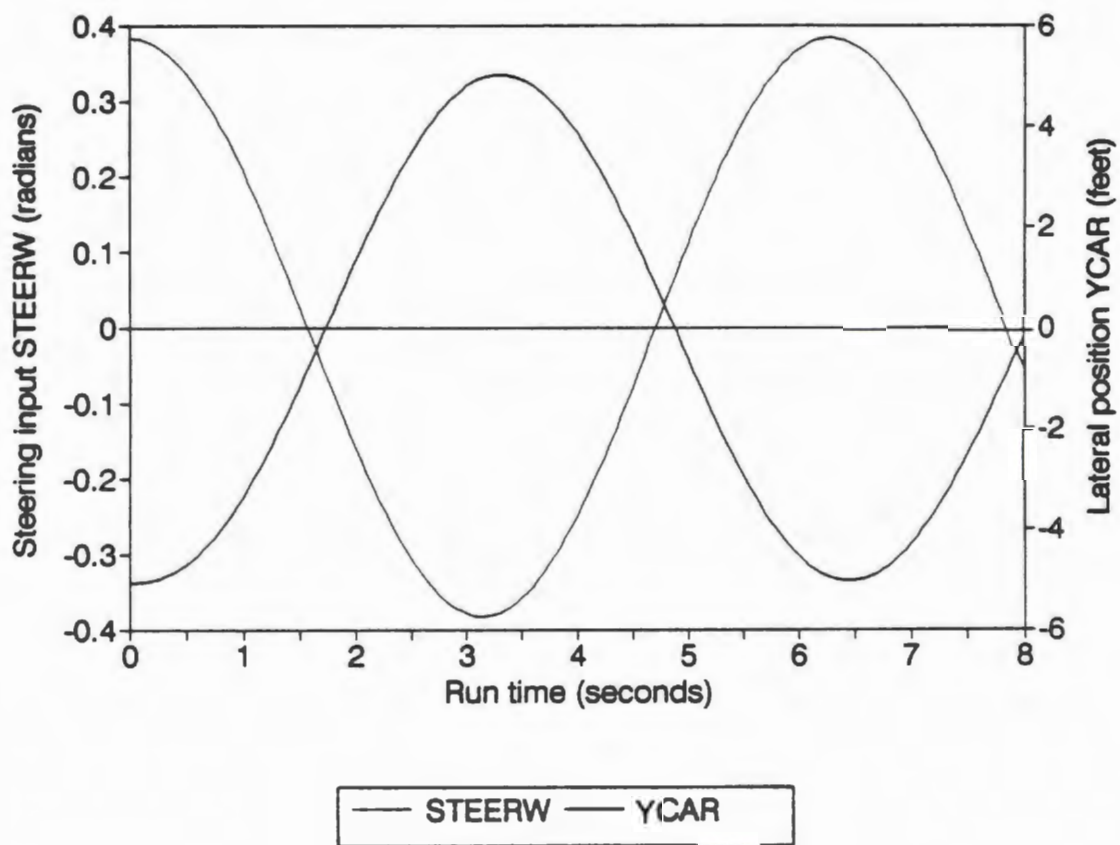


Figure 12. Vehicle Dynamics Phase Shift with 1 rad/s Steering Input

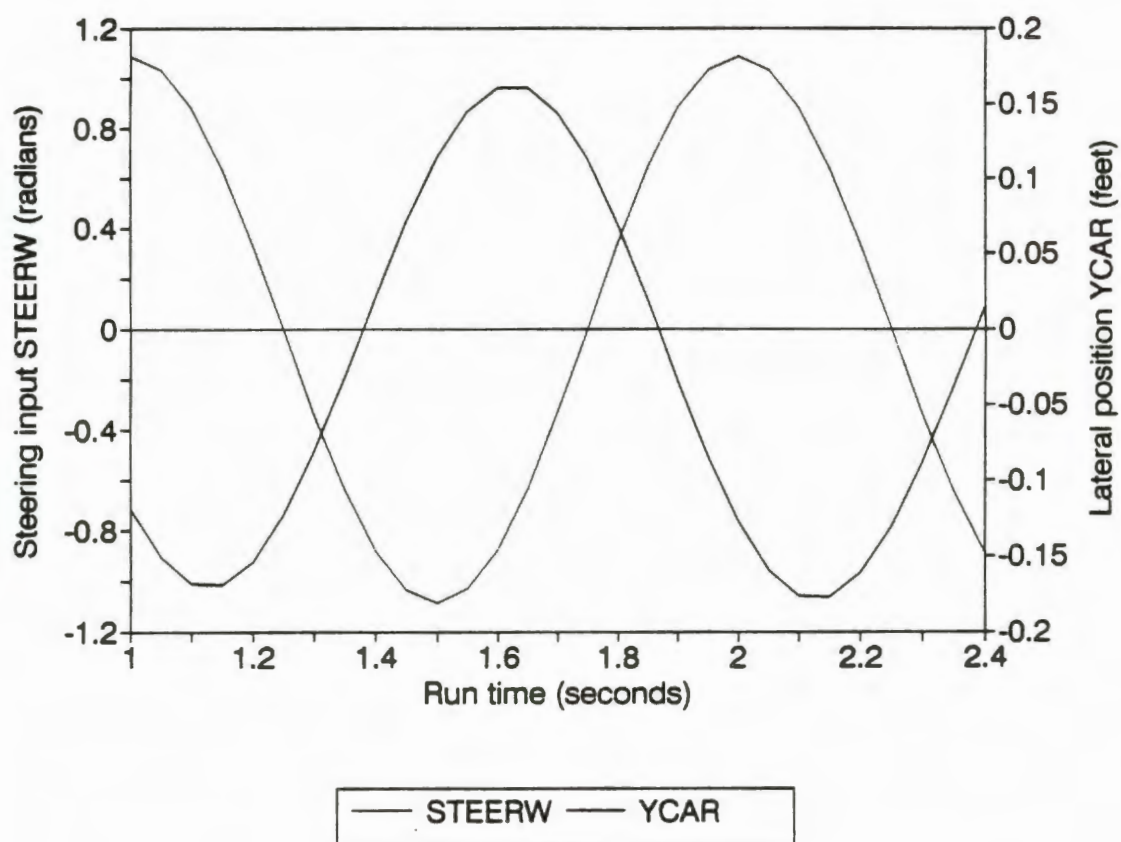


Figure 13. Vehicle Dynamics Phase Shift with 1 Hz Steering Input

Table 4.

Vehicle Dynamics Model Phase Measurements From Offline Simulation

UCAR (ft/s)	Model Phase Lag (STEERW to YCAR) (degrees)		
	STEERW Input Frequency (radians/second)		
	1.0	2.0	6.283
30	181.3	182.7	177.1
60	189.1	199.4	225.4
90	191.4	204.0	253.8

The results presented below were obtained from steady-state transport delay tests conducted on the actual system using the HP spectrum analyzer. Because of variability in the instrument readings several tests were run for each experimental condition; in each case the average was recorded to the nearest degree.

Table 5.

Steady-State Delay Measurements (No Moving Objects)

Steering Input Freq. (rad/s)	Phase Shift (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total (STEERW to IG)	Measured Model (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG delay (YCAR to IG)	Actual Model (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	189	181	8	9	181	8
	197	189	8	9	189	8
	200	191	9	9	191	9
2 (0.318 Hz)	198	181	17	17	183	15
	215	198	17	17	199	16
	219	202	17	17	204	15
6.283 (1.000 Hz)	224	173	51	51	177	47
	271	220	51	51	225	46
	301	250	51	52	254	47

Table 6.

Steady-State Delay Measurements (With Moving Objects)

Steering Input Freq. (rad/s)	Phase Shift (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total (STEERW to IG)	Measured Model (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG delay (YCAR to IG)	Actual Model (from offline simulation)	Meas. Total minus Actual Model
1 (0.159 Hz)	192	181	11	12	181	11
	200	189	11	11	189	11
	202	191	11	12	191	11
2 (0.318 Hz)	203	181	22	22	183	20
	218	198	20	21	199	19
	223	202	21	22	204	19
6.283 (1.000 Hz)	239	173	66	66	177	62
	286	220	66	66	225	61
	316	250	66	66	254	62

Tables 5 and 6 show the total simulator phase shift (measured from the STEERW simulated input to the IG video output) at each test condition, along with the measured vehicle dynamics model phase (STEERW to YCAR) and the actual model phase (determined from offline simulation) from Table 4. Also shown for each case is the phase shift due to transport delay, determined in three ways: measured from the YCAR output to the IG output, calculated by subtracting the measured dynamics model phase lag from the measured total phase lag, and

calculated by subtracting the dynamics model phase lag based on simulation (see Table 4) from the measured total.

The vehicle dynamics model phase lags from offline simulation should agree closely with those measured using the spectrum analyzer since the zero-order hold effects on the output of STEERW and YCAR cancel. However, discrepancies of 1-2 degrees (at 2 rad/s) and 4-5 degrees (at 1 Hz) were noted, with the measured values always less than or equal to the values obtained from simulation. This means that the transport delay values (see Tables 7 and 8) calculated from the figures for (total measured phase - simulated model phase) are always equal to or smaller than the values found by subtracting measured model phase from measured total phase. The transport delays calculated from the direct measurement of IG phase lag are very close to those obtained from (measured total - measured model); this is not surprising since both these sets of figures come entirely from spectrum analyzer measurements.

Table 7.

Transport Delay Measurements (No Moving Objects)

Steering Input Freq. (rad/s)	Transport Delay (milliseconds) (at UCAR = 30, 60, 90 feet per second)				
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady- State Total minus Actual Model	Measured Value from Time- Domain Test	Best Estimate (from Frame Timing Analysis)
1	140	157	140	140	133
(0.159 Hz)	140	157	140		
	157	157	157		
2	148	148	131	140	133
(0.318 Hz)	148	148	140		
	148	148	131		
6.283	142	142	131	140	133
(1.000 Hz)	142	142	128		
	142	144	131		

Table 8.

Transport Delay Measurements (With Moving Objects)

Steering Input Freq. (rad/s)	Transport Delay (milliseconds) (at UCAR = 30, 60, 90 feet per second)				
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady- State Total minus Actual Model	Measured Value from Time- Domain Test	Best Estimate (from Frame Timing Analysis)
1 (0.159 Hz)	192	209	192	182	174
	192	192	192		
	192	209	192		
2 (0.318 Hz)	192	192	175	182	174
	175	183	166		
	183	192	166		
6.283 (1.000 Hz)	183	183	172	182	174
	183	183	169		
	183	183	172		

All of the transport delay values calculated from the steady-state measurements (shown in Tables 7 and 8) are consistent within a few milliseconds of each other; they also agree quite well with the values obtained from the time-delay tests. Much of the discrepancies can be attributed to the measuring apparatus: the HP spectrum analyzer displays readings only to the nearest degree and guarantees accuracy only to ± 2 degrees. Two degrees of phase at a frequency of 1 radian per second

corresponds to approximately 35 milliseconds; even at 1 Hz, the accuracy of this instrument translates to about ± 6 ms. A more complete comparison of the time-domain and steady-state delay measurements, culminating in a best estimate of the total effective delay for each of the two IG database configurations (shown in the last column of Tables 7 and 8), appears in the following section.

Analysis of Simulator Delay Measurements

The data observed and presented in the previous sections allow us to construct a relatively accurate picture of the overall transport delay in the UCF Driving Simulator as made up of several components. The time-domain delay measurements include all components of delay except the computation time for the vehicle dynamics, numerical integration advances/delays, and the T/2 penalty due to zero-order hold effects in a sampled-data system.

The first element ignored by the time-domain delay measurements, namely the time required to perform the calculations for the vehicle dynamics, was isolated and determined to be approximately 0.69 millisecond. Because of the nature of the vehicle dynamics code (few branches or options) this figure should be fairly consistent from frame to frame. The average T/2 penalty for zero-order hold effects, given a 60 Hz frame rate, is 1/120 second or approximately 8.33 ms. These two effects, therefore, require us to add 9.02 milliseconds to the average figures determined from the time-domain measurements.

One of the main reasons mentioned in Chapter 1 for use of the AB-2/trapezoidal integration scheme in the simulator is its very good phase performance at frequencies of interest. The integrator outputs, when graphed versus time, matched up extremely well with the outputs that would be produced by continuous integrators. (Since an analog computer solution was not practical, comparisons were actually made with a "pseudo-continuous" solution found by integrating the model off-line using a very small time step; refer to Figure 11.) This is not to say, however, that numerical integration has no temporal effects in the UCF simulator. In fact, since there is significant spare frame time, and since all operations not critical to calculating position are done after updated coordinates are sent to the IG, there is a small but significant net time advance due to passing the position coordinates calculated for time $(n+1)T$ before the end of frame n (which began at time nT).

To understand this concept in more detail, recognize that we are given $V(0)$ as the initial output condition for an acceleration-to-velocity integrator. $A(0)$, the initial acceleration, is not known beforehand but is calculated from the inputs sampled at time $t=0$. We perform the required computations for numerical integration and get $V(0+T) = V(T)$. But, for the reasons enumerated above, this process (as well as the subsequent update of position from time 0 to time T) is completed before time T . (See Figure 14 for a graphical representation.) Extending this argument indefinitely, one can see that the (estimated) position coordinates for time $(n+1)T$ are always known as soon as the inputs at time T are sampled and the vehicle dynamics are computed. Thus, a lead of somewhat less than T seconds is introduced

into the system, compensating for a small portion of the transport delay. The remainder of the delay must be dealt with by a compensation algorithm (see Chapter 4).

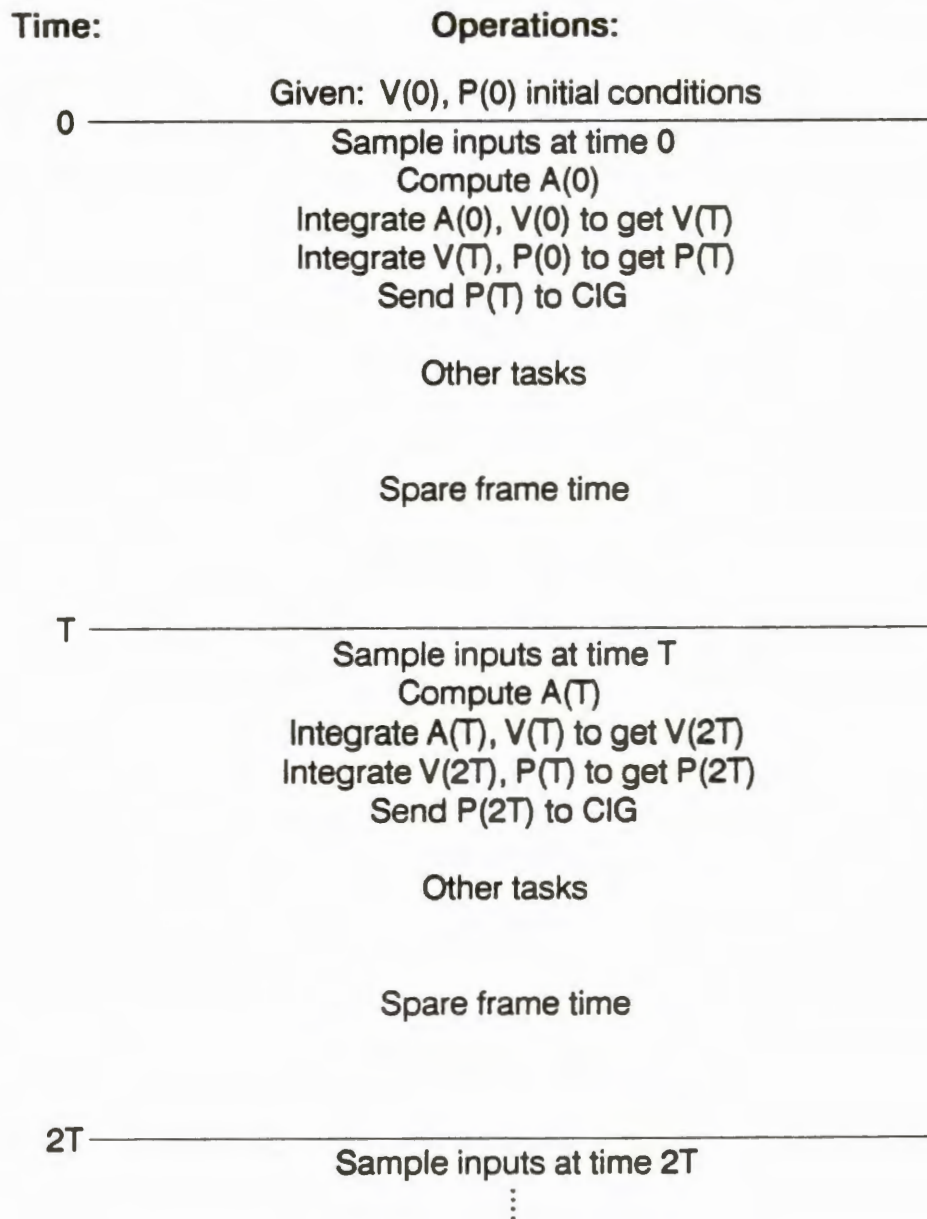


Figure 14. Single Frame Lead from Numerical Integration

The net time advance (with respect to the values obtained from time-domain measurements) is thus

$$T_{net\ advance} = T - \frac{T}{2} - t_{vehicle\ dynamics} \quad (3.2)$$

where T is the numerical integration prediction span, $T/2$ is the ZOH penalty, and $t_{vehicle\ dynamics}$ is the time required to run the dynamics routine. Using measured values, this works out to equal $(8.33 - 0.69)$ ms = 7.64 milliseconds. Applying this analysis and the values obtained from the time-domain measurements gives average total transport delay values for the simulator system using each of the four visual databases as follows:

Table 9.

Total Effective Simulator Transport Delay

Database Configuration	Transport Delay (ms)
Test (16 polygons)	49.98
Flat only (ground, road, background)	91.35
Flat plus stationary 3-D (buildings, trees)	132.73
Flat plus all 3-D (includes moving objects)	174.10

The last two figures in Table 9, rounded to 133 and 174 milliseconds, respectively, are the values used in designing delay compensation for the two IG database configurations used in driving tests. The first two databases were used for illustrative purposes only.

The analysis presented thus far has relied on the time-domain delay measurements primarily because they could be made more precisely with available instruments. The only instrument available for making phase measurements provided readings to the nearest degree, with accuracy guaranteed only to ± 2 degrees. By examining Tables 7 and 8 in the previous section it can be seen that the results of the steady-state delay measurements do agree with the results of the system delay analysis just presented within these bounds. Further investigations comparing the time-domain and steady-state delay measurement techniques using more precise instrumentation would be instructive but are not necessary to achieve the main objective of this research: improving the simulator by compensating for transport delay.

CHAPTER 4

TRANSPORT DELAY COMPENSATION

A number of attempts have been made to mitigate or compensate for the effects of transport delays in real-time, man-in-the-loop simulators. These have taken the form of modifications to the simulator hardware (for example the addition of motion platforms or displays of peripheral horizon cues discussed in Chapter 1) as well as software changes (the addition of delay compensation algorithms). The emphasis in the literature, and here as well, is on algorithmic compensation, since hardware enhancements are not always technically or financially feasible. The first section of this chapter describes in some detail several delay compensation algorithms reported in the literature. The second section addresses the application of some of these algorithms to the UCF Driving Simulator. Finally, the development of a new delay compensation algorithm for use in the UCF simulator is discussed in the last section.

Transport Delay Compensation Algorithms in the Literature

The idea of compensating for transport delays with digital filters or other algorithms has been around for a number of years. Published papers have described a number of approaches from the simple (linear prediction or pure lead) to the complex (for example McFarland's method described below). The common thread

of all these approaches is the calculation of a separate set of "vehicle coordinates" used only by the image generator. (See Figure 15.) These values computed by the algorithm represent not the present location and orientation of the simulated vehicle, but its anticipated future position at the time the image is actually displayed to the operator. The differences between the algorithms lie in how the future vehicle position is predicted. The following subsections describe the particulars of each approach to this problem.

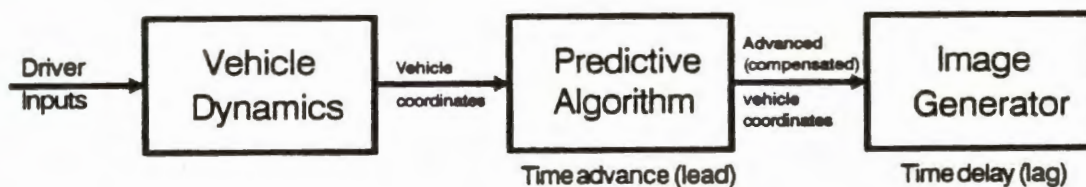


Figure 15. Block Diagram of Simulator with Predictive Delay Compensator

Predictive (Lead) Filtering

As early as 1975, Cooper, Harris, and Sharkey [27] suggested the use of "predictive filters designed based upon the frequency spectra of the differences in the delayed and non-delayed pilot control input performance", indicating that this form of compensation "could be expected to reduce the effects of the delayed visual presentation." Subsequently, Ricard, Norman, and Collyer [9] presented a simple linear prediction scheme for compensating for CIG system delays in real-time simulators by adjusting the simulated vehicle's position values before passing them to the CIG system. In this scheme, which had been studied by previous researchers, the estimated instantaneous values of a function, X_n , and its slope, \dot{X}_n , "are combined such that the predicted value is a linear extension of the function from point X_n :"

$$X_{n+1} = X_n + K\dot{X}_n \quad (4.1)$$

where K is the prediction span." This type of compensation is of value for real-time simulators because "both X_n and \dot{X}_n are usually available for each iteration of the dynamics processor, and little processor time is used to calculate X_{n+1} ." (Author's note: the notation used by Ricard, Norman, and Collyer is strictly correct only for the case where the prediction span is equal to the frame time. More generally, if the prediction span is K seconds and the frame time is T seconds, Equation 4.1 should be written as:

$$X_{n+\frac{K}{T}} = X_n + K\dot{X}_n \quad (4.2)$$

These researchers went on to point out some of the problems with the use of a linear prediction delay compensation scheme in a real-time simulator. Because the term $K\dot{X}_n$ is a product, they explained, "if either K or \dot{X}_n becomes large, the adjustment to X_n can also become large." \dot{X}_n can be large in any rapidly maneuvering vehicle, such as a high-performance aircraft, and of course K becomes larger with increased delay present in the system. The net result of prediction using the derivative, or pure lead compensation, is "an amplification of the high-frequency responses" of the simulated vehicle. This increased gain at high frequencies "produces values of X_{n+1} that can fluctuate quite a bit, causing a 'jitter' in the visual display." To reduce this annoying visual noise, it proved necessary to adjust K to some fraction of the true CIG system delay. In other words, it was not possible to satisfactorily compensate for all of a large transport delay by use of a pure linear predictive compensator. There proved to be a "best compromise" or tradeoff point, empirically determined for a given simulator, at which a certain amount of display jitter was acceptable for a given reduction of system delay. Above this level of compensation, noise in the visual presentation distracted the pilots to the point of interfering with the control task (the pilots began tracking noise fluctuations with their inputs); below this level, the display was stable enough but delay was not adequately compensated for.

Gum and Albery [24] investigated alternative methods of compensating for transport delay in the Advanced Simulator for Undergraduate Pilot Training (ASUPT). The primary source of delay in that flight simulator was the time (100

milliseconds) taken by the line-drawing, raster-scan image generation system to generate a video frame from an input of positional data. This was the only delay for which compensation was deemed critical; "the delay inherent in digital simulation from control input to output [to the display] was not considered ... the visual compensation value was chosen to lead the instruments by a value equal to the CIG system transport delay."

The compensation scheme chosen by Gum and Albery involved a combination of two adjustments to position values sent to the image generator. The first adjustment, termed "single-interval lead" by the researchers, was based on the particular real-time numerical integration method used in the ASUPT simulation software. This method (second-order Adams) has "enough information in a given frame to determine the exact position of the simulator one full frame ahead." Since the position (though not velocity or acceleration) information was known one full frame time, or 66.7 ms, ahead, simply passing the advanced position coordinates to the IG in place of the current coordinates allowed for the elimination of two-thirds of the total CIG transport delay. Without going into the details, the researchers stated that "the remainder of the delay ... was compensated for by a two-term Taylor Series." They emphasized the difference between the two methods by explaining, with diagrams, that "the single-interval lead is a true time compensating technique whereas the Taylor series extrapolation is a position compensating technique [24]."

Gum and Albery found, in trials with test pilots, that their attempt to augment the single-interval lead compensation with a Taylor Series extrapolation in order to

eliminate the remaining CIG transport delay "resulted in an objectionable lack of smoothness in the visual scene." This was particularly apparent "under rapid control-reversal conditions." Their study concluded by recommending "that the Taylor series extrapolation be abandoned in favor of using only the single-interval lead with a variable integration interval incorporated to provide some delay compensation variation capability [24]." Thus, once again it was demonstrated that pure lead compensation was unsuitable for compensating for all of a lengthy simulator transport delay.

Lead/Lag Filters

Ricard, Norman, and Collyer attempted to improve the linear prediction scheme discussed in [9] by developing a rule for limiting the difference between X_n and the predicted value X_{n+1} . In control systems terminology, they decided to "reduce the amount that a compensation scheme can amplify high-frequency responses by passing adjusted parameters through a low-pass filter." They performed a series of experiments using naive and trained subjects, flying two different types of aircraft, under conditions of varying delay with and without compensation. The compensation, when applied, was in the form of linear prediction cascaded with a first-order low-pass filter with variable break frequency. One conclusion of their study was that the optimal break frequency (from the point of view of pilot training) for the first-order low-pass filter was in the range of 4 to 5 radians per second. For significant delay conditions (up to 400 ms), an approximate 40 percent reduction in

pilot training time was achieved when delay was compensated for using the prediction/filtering approach. The researchers concluded that by using the low-pass filter to remove most of the "jitter" associated with the linear prediction scheme for delay compensation, they had overcome its chief limitation and "significantly increased its training usefulness." They further conjectured that by changing to a second-order low-pass filter, the break frequency could be set higher (allowing more of the phase lead effect to remain) "while still attenuating the high-frequency responses strongly enough to reduce the display 'jitter' to acceptable levels [9]."

In their 1980 paper, Ricard and Harris [14] elaborated on the delay compensation scheme described above. The compensator, which used prediction in the form of a first-order lead followed by a first-order low-pass filter to reduce high-frequency noise (see Figure 16), was equivalent to the insertion of lead/lag transfer functions in the pitch and roll control loops of the pilot/simulator system. The authors described the results of experiments in which subjects' control performance was evaluated for various ratios of T_d (low-pass filter time constant) to T_n (lead time constant, set equal to the delay of the visual display system). There proved to be an optimal low-pass filter break frequency setting (in terms of pilot performance in nulling errors and reduction of control deflection needed) below which the detrimental effects of delay were increasingly evident and above which pilots began trying to null display jitter rather than actual system error. The researchers concluded from their experiments that the "lead/lag form of delay compensation seems to be useful, even though it may produce smaller amounts of phase lead than

other methods of adjusting signals for visual displays." Pilots were able to control the simulated aircraft significantly better when delays were compensated for, and "filtering clearly produced a more acceptable image than no filtering"

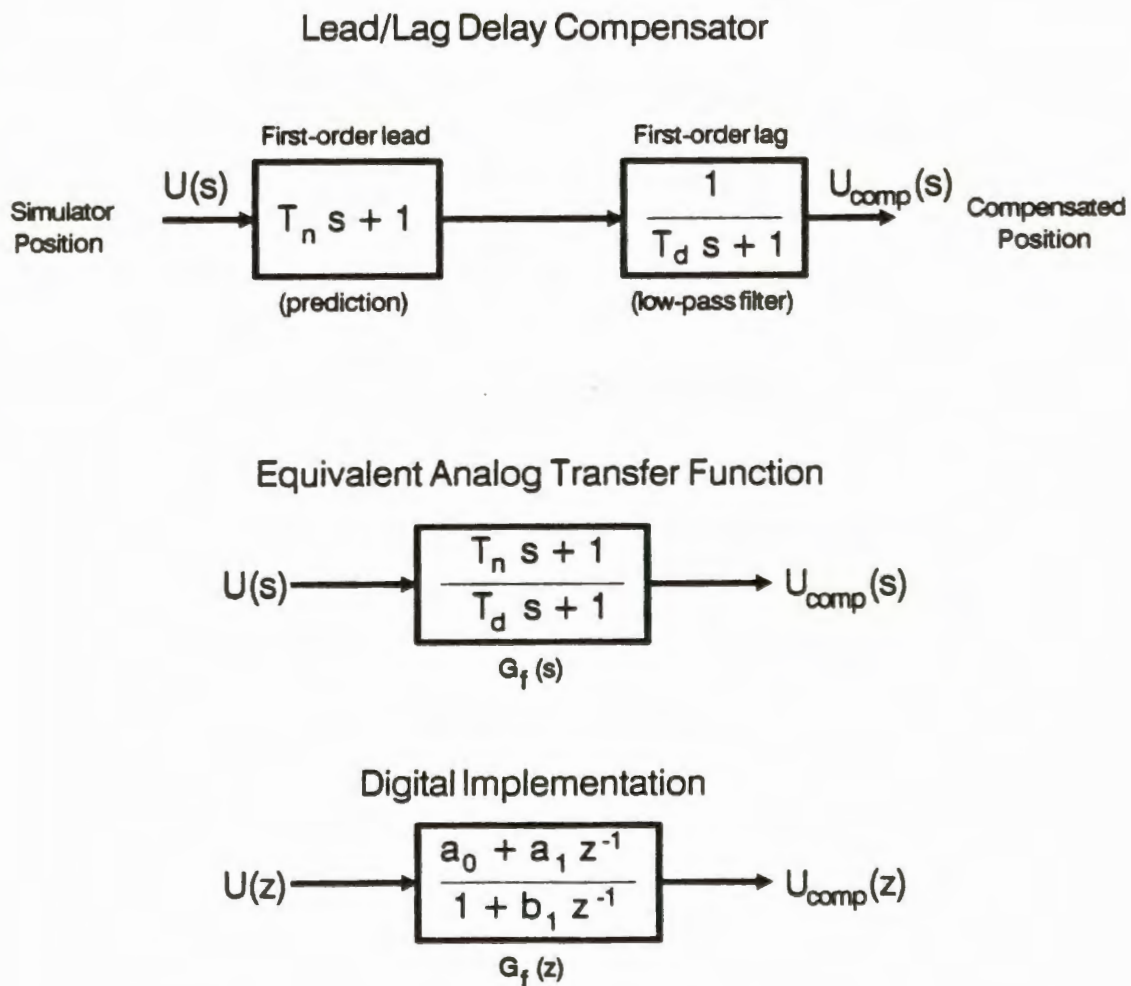


Figure 16. Block Diagram of Lead/Lag Delay Compensator

Crane [6] [43] further refined lead/lag delay compensation by proposing a set of design rules to govern the choice of the filter pole and zero in order to achieve the necessary lead while minimizing the filter's gain distortion. While "it is especially important to note that phase lead is purchased at the cost of gain distortion" (amplification of high-frequency signals) when using a lead/lag filter, Crane's design guidelines "attempt to minimize change in pilot-aircraft dynamics in the region of crossover in order to restore system stability and maintain system responsiveness." This is accomplished by careful selection of the parameters K_D , T_n , and T_d in the filter transfer function:

$$G_f = K_D \frac{T_n s + 1}{T_d s + 1} \quad (4.3)$$

When designing the lead/lag compensator according to Crane's approach, the zero is located at the (measured or assumed) crossover frequency by setting $T_n = 1/\omega_c$. This is done in order to provide the greatest lead near (and just above) the crossover frequency, where the operator needs it most in order to combat the delay. The transfer function pole serves to roll off the high-frequency gain of the filter in order to avoid the problem of display jitter (a manifestation of high-frequency noise). The pole is placed just "above" (higher in frequency than) the zero; the exact position depends on the length of the delay and the location of the zero. It is chosen by equating the filter phase lead (ϕ_f) at ω_c to the phase lag produced by the transport delay at ω_c and solving the resulting equation:

$$\Phi_f|_{\omega=\omega_c} = \tan^{-1}(\omega_c T_n) - \tan^{-1}(\omega_c T_d) = \omega_c t_{delay} \quad (4.4)$$

The K_D term is chosen such that the magnitude of G_f at ω_c is unity. This reduces the gain distortion above ω_c (and eliminates it at ω_c) at the cost of reducing system responsiveness somewhat at frequencies below ω_c .

Some problems with lead/lag compensation for transport delays include the abovementioned gain distortion (amplification) at higher frequencies, which corrupts the simulation, and the fact that one must know (or be able to estimate) the operator/vehicle system crossover frequency in order to place the lead at the frequency where it will do the most good. Above the design frequency (chosen by Crane's method to be equal to the crossover frequency) there will be insufficient lead; below it there will be slightly too much lead. This excess low-frequency lead compensates for a small part of the vehicle dynamics lags in addition to the transport delay and thus artificially makes the vehicle slightly easier to drive or fly. Despite these problems, however, lead-lag filters have been used in several simulators to compensate for display delays, with mostly favorable results. In experiments described by Crane which used an oscilloscope-type display as well as a CIG display, "the compensation was effective; improvements in pilot performance and workload or HQR [handling-qualities rating] were observed [6]."

McFarland's Compensator

McFarland, in [42] and [44], described a compensation approach that predicts the vehicle's future position (at the time the scene will finally be displayed) using the current position and the three most recent velocity terms. Acceleration terms are avoided because they are not concurrent with the position and velocity terms in real-time and because they contain higher-frequency components not attenuated by the vehicle dynamics [44]. The form of McFarland's compensator (see Figure 17 for a block diagram) is:

$$u_{k+\frac{P}{T}} = u_k + b_0 v_k + b_1 v_{k-1} + b_2 v_{k-2} \quad (4.5)$$

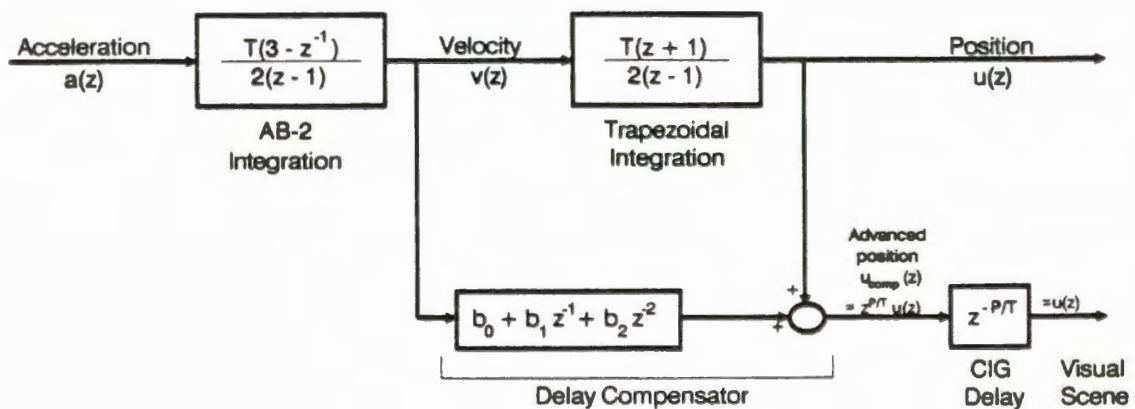


Figure 17. Block Diagram of McFarland's Delay Compensator

where P is the prediction interval (generally chosen equal to the transport delay) and T is the simulator frame time. The term u_k represents any one of the most recently computed set of vehicle position or orientation coordinates; $u_{k+P/T}$ is the predicted coordinate P seconds in the future which is passed to the image generator instead of u_k . v_k , v_{k-1} , and v_{k-2} are the current and two immediately previous velocity values corresponding to the u coordinate. The three weighting coefficients b_0 , b_1 , and b_2 are found by solving three simultaneous equations which constrain the performance of the compensator under certain conditions. The first equation (eq. (10) in reference [42]) is formed by constraining for perfect prediction at DC ($\omega=0$) or constant velocity:

$$b_0 + b_1 + b_2 = P \quad (4.6)$$

In order to derive the other two constraint equations, McFarland begins by expressing the z -domain transfer function representing the compensator difference equation. If the velocity-to-position integration is done using a trapezoidal integrator, this transfer function is given by his equation (11):

$$f_A(z) = \frac{u(z)}{v(z)} = z^{-\frac{P}{T}} \left[\frac{T}{2} \left(\frac{1+z^{-1}}{1-z^{-1}} \right) + b_0 + b_1 z^{-1} + b_2 z^{-2} \right] \quad (4.7)$$

Because in a continuous system position is the integral of velocity, "the total delay-plus-prediction process may be compared to the perfect velocity-to-position

transfer function by forming the relative error function" (equation (12) of reference [42]):

$$h(\omega) = j\omega f_A(z)|_{z=e^{j\omega T}} \quad (4.8)$$

This equation is used to derive the additional two constraint equations necessary to compute the b_i 's. These constraints are obtained by "tuning" the preceding equation for zero error at a particular frequency f_0 (or equivalent angular frequency $\omega_0 = 2\pi f_0$). To accomplish this, the expressions for the magnitude and angle of the relative error function (Equation 4.8) are set to equal one and zero, respectively, at ω_0 . The equations $|h(\omega_0)| = 1$ and $\angle h(\omega_0) = 0$ (or, equivalently, $\text{Re}[h(\omega_0)] = 1$ and $\text{Im}[h(\omega_0)] = 0$) provide the other two constraints needed to determine the three unknowns b_0 , b_1 , and b_2 . Defining the parameters $\theta_0 = \omega_0 T$ (the cyclic angle) and $\psi_0 = \omega_0 P$ (the projection angle) and solving the three equations simultaneously for the weighting coefficients, McFarland [42] obtains the following results:

$$b_0 = \frac{[\psi_0 + \sin \psi_0 (1 - 2 \cos \theta_0)] \sin \theta_0 + [\frac{1}{2} \theta_0 \sin \theta_0 - \cos \psi_0 (1 - \cos \theta_0)] (1 + 2 \cos \theta_0)}{2 \omega_0 \sin \theta_0 (1 - \cos \theta_0)} \quad (4.9)$$

$$b_1 = \frac{\sin \theta_0 [2 \sin (\theta_0 + \psi_0) - 2 \psi_0 \cos \theta_0 - \theta_0 (1 + \cos \theta_0)]}{2 \omega_0 \sin \theta_0 (1 - \cos \theta_0)} \quad (4.10)$$

$$b_2 = \frac{\sin \theta_0 [\psi_0 - \sin \psi_0 + \frac{1}{2} \theta_0] - \cos \psi_0 (1 - \cos \theta_0)}{2 \omega_0 \sin \theta_0 (1 - \cos \theta_0)} \quad (4.11)$$

Above the design frequency ω_0 there is rather severe gain distortion (amplification of high-frequency inputs); however, if ω_0 is well chosen there is little

or no visible effect since the "gains" of both the operator and vehicle roll off above the system crossover frequency, and consequently little energy exists in the system in the spectral region where the distortion occurs. Between 0 Hz and the design frequency, performance is good (considerably better than that of the lead/lag filter) for the range of transport delays investigated by McFarland. This is in contrast to the lead/lag approach, the performance of which is only good near the design frequency (crossover frequency).

Jewell, Clement, and Hogue [45] performed a frequency-domain analysis of pilot control movements in the NASA-Ames Vertical Motion Simulator with and without the use of McFarland's delay compensation scheme. Their results showed that the compensation algorithm did correct the phase lag due to the 112 ms of overall system transport delay, but also produced "a slight hint of gain distortion at about 13 rad/s" for control of pitch. For a subsequent evaluation of control in the yaw axis, again with the compensation scheme in use, "there is no evidence of phase lag accompanying a CGI delay and virtually no magnitude distortion up to frequencies of about 10 rad/s. Above 10 rad/s there is a definite trend toward phase lead and gain amplification. This appears to be the 'price' one has to pay for correcting the phase lag at lower frequencies" using this technique. The price is acceptable if no noticeable visual artifacts are created.

Describing the effect of this scheme in a more recent article, McFarland and Bunnell [38] stated that "because of the compensation algorithm ... delay is not observed in flight simulation (over the frequency range pertinent to handling qualities

research)." While the actual transport delay is "on the order of 150 msec", in the "flight simulation environment at Ames the effective delay is more like 15 msec."

Application of Delay Compensation Algorithms to the UCF Simulator

Two types of delay compensation described in the previous section (a lead/lag filter and McFarland's predictor) were implemented in the UCF Driving Simulator. Linear prediction ("pure lead", see Equation 4.2) had already been shown to be ineffective for compensating for transport delays of the magnitude found in the UCF simulator, so that method was eliminated from consideration. The following subsections provide details of the application of the two chosen delay compensation algorithms as well a new compensator that was developed by the author.

Lead/Lag Compensation in the UCF Simulator

A C program was written to allow automatic calculation of lead/lag filter transfer functions in the s - and z -domains using Crane's design approach [6] and the Tustin bilinear transform (which corresponds to trapezoidal integration). The program accepted values for the estimated crossover frequency ω_c , the time delay P to be compensated for, and the host frame rate T ; it printed out $G_f(s)$ and $G_f(z)$, the filter transfer functions. The compensator is implemented as a difference equation derived from $G_f(z)$.

Code which implements the lead/lag compensation filter difference equation was inserted into the real-time host program for the driving simulator. Initial testing

revealed a problem with the operation of the compensator: the nonzero "DC gain" of the filter (the term K_D in Equation 4.3), applied to the values sent to the IG (which are in absolute or world coordinates), resulted in the viewpoint being shifted significantly in the X and Y directions and pointed at the wrong heading angle. With the filter in the form of Equation 4.3, after initial transients died out, the X, Y, and heading values sent to the IG were at $K_D * 100$ percent of the corresponding values in the host. This is obviously not satisfactory for real-time simulator operation.

The obvious alternatives to alleviating this problem involved eliminating the K_D gain term from the difference equation feeding the IG. The K_D term could either be left out completely or put somewhere "upstream" of the final processing (before the coordinate transformations and integration to absolute X and Y positions and heading). Of course, this latter approach would affect the output of the vehicle model itself, effectively reducing overall low-frequency gain and therefore the response of the car in all axes. For this reason it was decided to simply eliminate the K_D term from the compensator. This does have the effect of increasing the gain (and therefore adding gain distortion) at higher frequencies (near and above the system crossover frequency); however, it leaves the low-frequency gain correct, which was deemed more important. A Bode plot which shows the gain magnitude and phase characteristics of the lead/lag compensator designed for the 174 millisecond transport delay may be found in Figure 18.

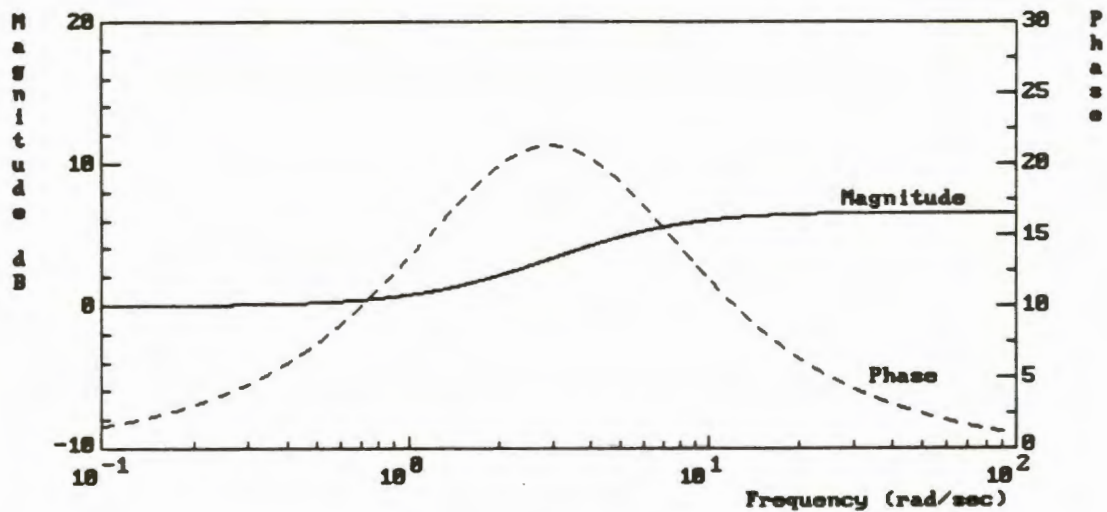


Figure 18. Bode Plot of Lead/Lag Compensator Gain Magnitude and Phase

After the above corrections were made, the program was re-tested. Informal evaluation of the simulated car's handling characteristics by project personnel indicated that the lead/lag compensator had enough of an effect (apparently beneficial) on simulator driving to be worth testing formally. These tests, which involved a reiteration of the steady-state delay measurements (previously performed on the system without compensation) as well as driving tests by human subjects, are described in Chapter 5.

Application of the McFarland Compensator to the UCF Simulator

A C program was written to compute the coefficients b_0 , b_1 , and b_2 for McFarland's algorithm using equations 4.9, 4.10, and 4.11. This program was used to find the coefficient values for McFarland's example system and also for the UCF simulator, both with a design frequency $f_0 = 3 \text{ Hz} \approx 18.849556 \text{ rad/s}$. For the example system in [42], $T = 1/50$ and $P = 83$ milliseconds; the resulting compensator had excellent phase characteristics (compared to the lead/lag filter) below f_0 , and especially below 3 rad/s . (See Figure 19 for a Bode plot.) Performance was less exemplary when the values $T = 1/60$ and $P = 174 \text{ ms}$ (typical of the UCF simulator) were used. Phase errors of 0.83 , 2.59 , and 15.96 degrees were calculated at frequencies of 2 rad/s , 3 rad/s , and 1 Hz (6.283 rad/s). The maximum phase error in the prediction band was determined to be about 62.65 degrees at about 14.49 rad/s . All of these errors were in the direction of not providing enough phase lead to compensate for the delay.

While the largest errors computed were above the assumed crossover frequency, the errors in the range of $2\text{-}3 \text{ rad/s}$ were sufficient to warrant efforts at improvement. This could be attempted by choosing f_0 lower (say, 1 Hz instead of 3 Hz); however, if f_0 is chosen too low, the amplitude distortion (which is quite severe above f_0) could become noticeable to the driver through the visual display and thus affect the simulation adversely. These observations suggested the possible need for development of a compensation method that would give better phase performance in the prediction band than McFarland's algorithm. Such a compensator would

ideally allow for a longer "prediction interval" relative to T without increasing the phase error so much in the compensation band.

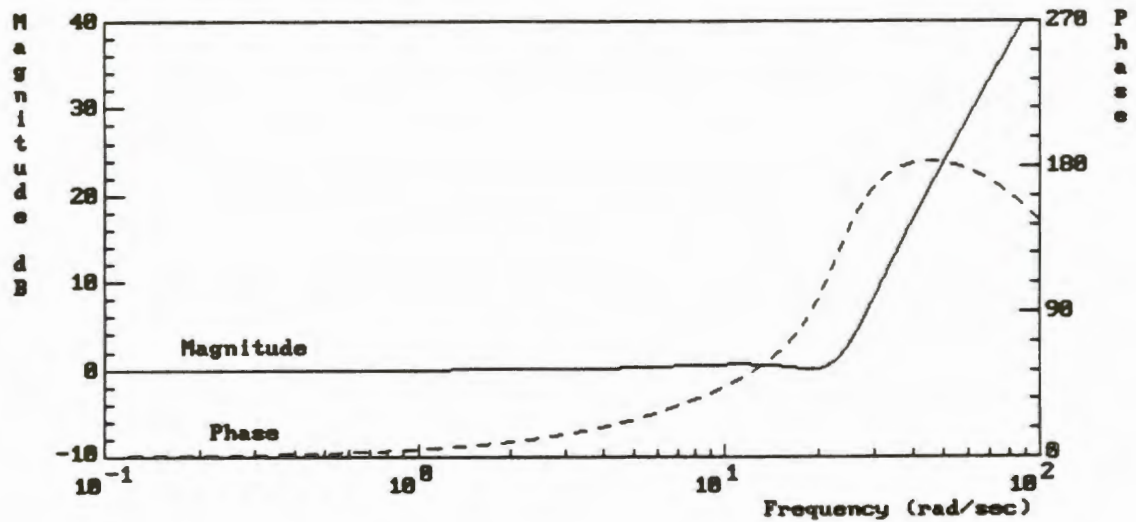


Figure 19. Bode Plot of McFarland's Compensator ($P = 0.083$ s, $T = 0.02$ s)

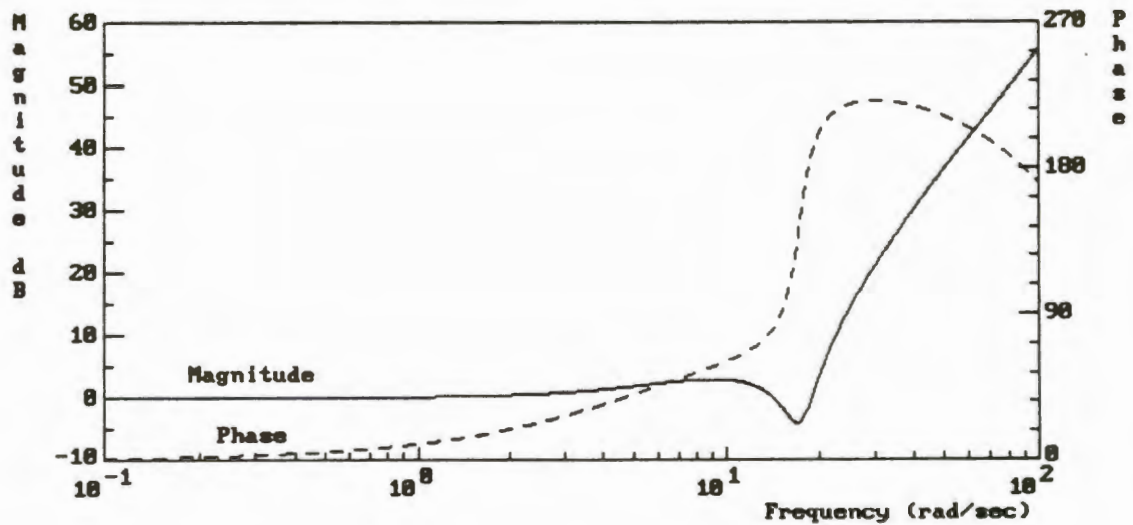


Figure 20. Bode Plot of McFarland's Compensator ($P = 0.174$ s, $T = 0.01667$ s)

Despite the phase error inherent to McFarland's compensator when designed for the frame time and transport delay of the UCF simulator, its theoretical performance in the prediction band is considerably better than that of the lead/lag filter. Figure 21 shows a comparison of the phase characteristics of the two compensators, while Figure 22 compares their gain magnitude characteristics. Accordingly, the simulator host computer programs (both the real-time interactive program and the steady-state delay test program) were modified to compute the projected values for X and Y position and heading (per Equation 4.5) and pass these values (the compensator outputs) to the image generator. Informal testing revealed apparently beneficial effects on vehicle handling which was checked using steady-state delay measurements as well as driver-in-the-loop tests. The results of these tests are discussed in Chapter 5.

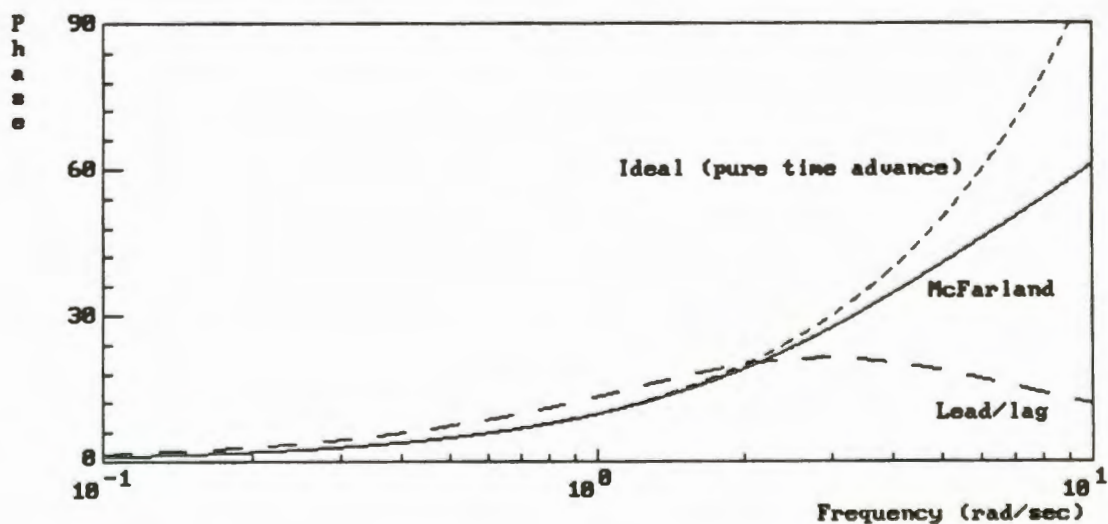


Figure 21. Comparison of Phase Characteristics (McFarland vs. Lead/Lag)

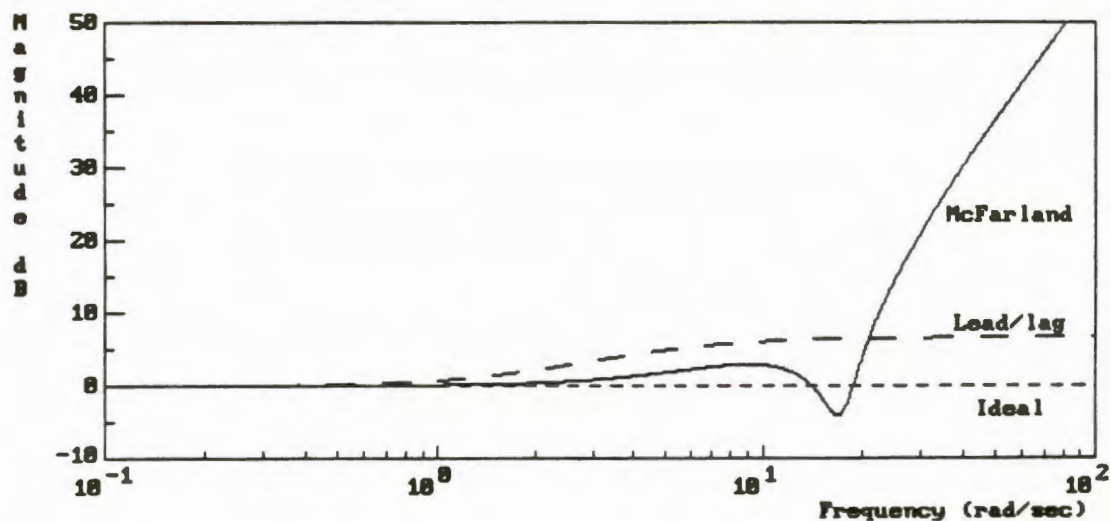


Figure 22. Comparison of Gain Characteristics (McFarland vs. Lead/Lag)

Other Compensation Approaches Considered

Some articles located in a search of the literature described attempts to improve on the basic lead or lead/lag compensators. For example, Ricard, Norman, and Collyer [9] suggested the possibility of trying a second-order lag section cascaded with the lead compensator. This filter, as well as a similar design with a second-order numerator and denominator, was investigated and found not to be as promising as other approaches (for example, McFarland's prediction algorithm). Excess phase lead is still generated at low frequencies, while the greater attenuation of high-frequency noise is not really necessary since jitter was not a significant problem with the basic lead/lag filter in the UCF simulator.

Hess and Myers [46] described a compensator which produced phase lead without the amplitude distortion associated with the lead/lag filter. Their "SPAN" (Split PAth Nonlinear) filter consists of a lead/lag transfer function with associated nonlinear elements used to cut the gain at higher frequencies. It would also be possible to extend this approach by substituting another type of compensator for the lead/lag filter component used to provide the phase lead in Hess and Myers' design. In any case, however, modifications would need to be made to their approach to allow the nonlinear elements to work with absolute position coordinates (as used in the UCF simulator). For that reason as well as the relative success and ease of implementation of McFarland's compensator, the author decided not to pursue implementation of the SPAN filter in the driving simulator.

McMillan [41], in his review article, described an approach to delay compensation involving prediction of the vehicle's future position using a weighted combination of velocity and acceleration terms. McFarland's approach and the new algorithm designed by the author (to be discussed below) are examples of this type of compensator where the acceleration terms are given weights of zero (only velocity terms are used). Some theoretical investigations were performed in an attempt to develop a compensator using both velocity and acceleration terms, but no workable solutions were found. Thus, it was decided to further investigate the McFarland algorithm and try to develop a similar compensator for use in the UCF simulator.

A New Compensation Algorithm

While the McFarland compensation algorithm worked well in the NASA-Ames flight simulator [38] [45] and was shown to have good characteristics with a transport delay of 83 milliseconds [42], it is not quite as accurate for longer delays and/or when compensation must be applied over a wide bandwidth (ω_0 is large). With these limitations in mind, an attempt was made to generalize and extend McFarland's compensation approach in order to achieve better performance and/or compensate for longer delays.

A Compensator Using Five Velocity Terms

The general form of the difference equation for a delay compensator similar to McFarland's using several velocity values is:

$$u_{k+\frac{P}{T}} = u_k + b_0 v_k + b_1 v_{k-1} + b_2 v_{k-2} + b_3 v_{k-3} + b_4 v_{k-4} + \dots \quad (4.12)$$

where the symbology is the same as for Equation 4.5. If we retain the constant velocity constraint

$$b_0 + b_1 + b_2 + b_3 + b_4 + \dots = P \quad (4.13)$$

and restrict ourselves to considering compensators with an odd number of velocity terms, it is possible to "tune" the compensator at more than one frequency. In general, forcing perfect prediction at m frequencies (excluding DC, which is covered by the constant velocity constraint) will require the use of $2m+1$ velocity values in the compensator difference equation.

The simplest extension of McFarland's compensation approach, then, uses five velocity terms for the prediction of future position and is designed by "tuning" at two frequencies rather than one. The two design frequencies may now be denoted ω_0 and ω_1 . The compensator difference equation has the form:

$$u_{k+\frac{P}{T}} = u_k + b_0 v_k + b_1 v_{k-1} + b_2 v_{k-2} + b_3 v_{k-3} + b_4 v_{k-4} \quad (4.14)$$

The first of the five constraint equations necessary for determining the coefficients b_0 - b_4 is the constant velocity constraint

$$b_0 + b_1 + b_2 + b_3 + b_4 = P \quad (4.15)$$

The other four constraint equations are determined in a manner similar to McFarland's method, by developing the relative error function $h(\omega)$ and setting its real part to 1 and imaginary part to 0 at each of the two design frequencies ω_0 and ω_1 . $H(\omega)$ is found from the z-domain transfer function

$$f_A(z) = z^{-\frac{P}{T}} \left[\frac{\frac{T}{2}(1+z^{-1}) + b_0(1-z^{-1}) + b_1 z^{-1}(1-z^{-1}) + b_2 z^{-2}(1-z^{-1}) + b_3 z^{-3}(1-z^{-1}) + b_4 z^{-4}(1-z^{-1})}{1-z^{-1}} \right] \quad (4.16)$$

in the same manner used to derive Equation 4.8 from Equation 4.7. Once again defining $\theta = \omega T$ and $\psi = \omega P$, the relative error function can be shown to be

$$h(\omega) = \frac{\{(1 - \cos\theta)[(\omega \sin\psi)d_0 + (\omega \cos\psi)d_1] + (\sin\theta)[(\omega \cos\psi)d_0 - (\omega \sin\psi)d_1]\} + j\{(1 - \cos\theta)[(\omega \cos\psi)d_0 - (\omega \sin\psi)d_1] - (\sin\theta)[(\omega \sin\psi)d_0 + (\omega \cos\psi)d_1]\}}{2(1 - \cos\theta)} \quad (4.17)$$

where

$$d_0 = \left(\frac{r}{2} + b_0\right) + \left(\frac{r}{2} + b_1 - b_0\right)(\cos\theta) + (b_2 - b_1)(\cos 2\theta) + (b_3 - b_2)(\cos 3\theta) + (b_4 - b_3)(\cos 4\theta) + (-b_4)(\cos 5\theta) \quad (4.18)$$

and

$$d_1 = \left(\frac{r}{2} + b_1 - b_0\right)(\sin\theta) + (b_2 - b_1)(\sin 2\theta) + (b_3 - b_2)(\sin 3\theta) + (b_4 - b_3)(\sin 4\theta) + (-b_4)(\sin 5\theta) \quad (4.19)$$

Setting the real part of $h(\omega)$ equal to 1 we obtain

$$b_0(K_0 - K_A) + b_1(K_A - K_B) + b_2(K_B - K_C) + b_3(K_C - K_D) + b_4(K_D - K_E) = \frac{2(1 - \cos\theta)}{\omega} - \frac{r}{2}(K_0 + K_A) \quad (4.20)$$

where

$$K_0 = \sin \psi - \sin \psi \cos \theta + \cos \psi \sin \theta \quad (4.21)$$

$$K_1 = \cos \psi - \cos \psi \cos \theta + \sin \psi \sin \theta \quad (4.22)$$

$$K_A = K_0 \cos \theta + K_1 \sin \theta \quad (4.23)$$

$$K_B = K_0 \cos 2\theta + K_1 \sin 2\theta \quad (4.24)$$

$$K_C = K_0 \cos 3\theta + K_1 \sin 3\theta \quad (4.25)$$

$$K_D = K_0 \cos 4\theta + K_1 \sin 4\theta \quad (4.26)$$

$$K_E = K_0 \cos 5\theta + K_1 \sin 5\theta \quad (4.27)$$

Equation 4.20 applies to each of the two design frequencies ω_0 and ω_1 and thus provides two more of the five necessary constraints for determining b_0 - b_4 .

Now, setting the imaginary part of $h(\omega)$ equal to zero we obtain

$$b_0(K_1 - K_F) + b_1(K_F - K_G) + b_2(K_G - K_H) + b_3(K_H - K_J) + b_4(K_J - K_K) = -\frac{T}{2}(K_1 + K_F) \quad (4.28)$$

where K_0 and K_1 are defined above and the remaining constants are given by

$$K_F = K_1 \cos \theta - K_0 \sin \theta \quad (4.29)$$

$$K_G = K_1 \cos 2\theta - K_0 \sin 2\theta \quad (4.30)$$

$$K_H = K_1 \cos 3\theta - K_0 \sin 3\theta \quad (4.31)$$

$$K_J = K_1 \cos 4\theta - K_0 \sin 4\theta \quad (4.32)$$

$$K_K = K_1 \cos 5\theta - K_0 \sin 5\theta \quad (4.33)$$

Again, Equation 4.28 is applied at each of the two design frequencies ω_0 and ω_1 , providing the final two constraints necessary to solve for b_0 - b_4 .

Since the expressions for the coefficients of b_0 - b_4 in Equations 4.20 and 4.28 are too cumbersome for hand calculations, a C program was written to compute these coefficients for any given T , P , ω_0 , and ω_1 . Once all the appropriate values are obtained, the five equations can be expressed in matrix form and solved simultaneously using a matrix-solving program such as MATRIXx. The output of the equation-solving program is the set of compensator coefficients b_0 - b_4 .

In order to check the derivation of Equations 4.20 and 4.28 and allow the construction of Bode plots of compensator frequency response, the z-domain transfer

function of the compensator alone ($U_{\text{comp}}(z)/U(z)$) was evaluated in terms of the coefficients b_0 - b_4 and the sampling period T :

$$\frac{U_{\text{comp}}(z)}{U(z)} = \frac{(1 + \frac{2b_0}{T})z^5 + (1 + \frac{2b_1}{T} - \frac{2b_0}{T})z^4 + (\frac{2b_2}{T} - \frac{2b_1}{T})z^3 + (\frac{2b_3}{T} - \frac{2b_2}{T})z^2 + (\frac{2b_4}{T} - \frac{2b_3}{T})z - (\frac{2b_4}{T})}{z^5 + z^4} \quad (4.34)$$

Another program was written to aid in computing the transfer function numerator coefficients in Equation 4.34. Finally, a control systems design program, CC, was used to draw Bode plots (Figures 23 and 24) for the new compensator.

The five-term compensator designs for the driving simulator were done using an upper design frequency f_0 of 3 Hz ($\omega_0 = 18.849556$ rad/s), the same design frequency used by McFarland for his compensator using three velocity terms. The second, lower design frequency ω_1 was chosen to be 2.0 rad/s in order to optimize performance near the estimated crossover frequency and allow a direct comparison with the lead/lag compensator, which was designed to exactly cancel the phase shift due to delay at that frequency. Given these choices, the algorithm approximates a pure time advance of P seconds much better (within the compensation band) than does McFarland's three-term compensator. The tradeoff, readily apparent in the Bode plots, is that beyond f_0 the gain of the five-term compensator increases more rapidly and to a much higher value for this design than for McFarland's compensator. In other words, the price paid for better behavior at low frequencies is increased gain distortion at higher frequencies.

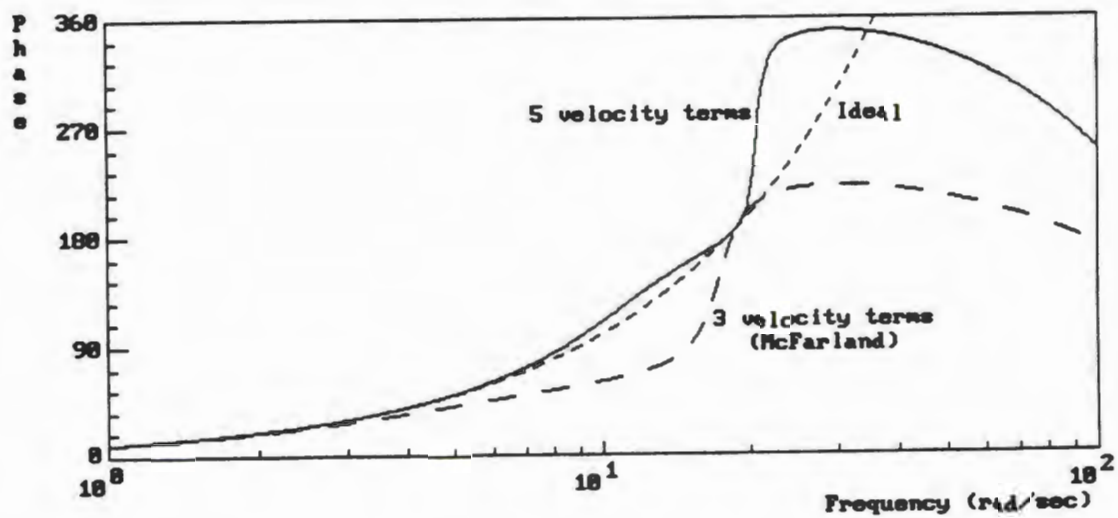


Figure 23. Bode Plot of Phase Characteristics, Five-velocity-term Compensator

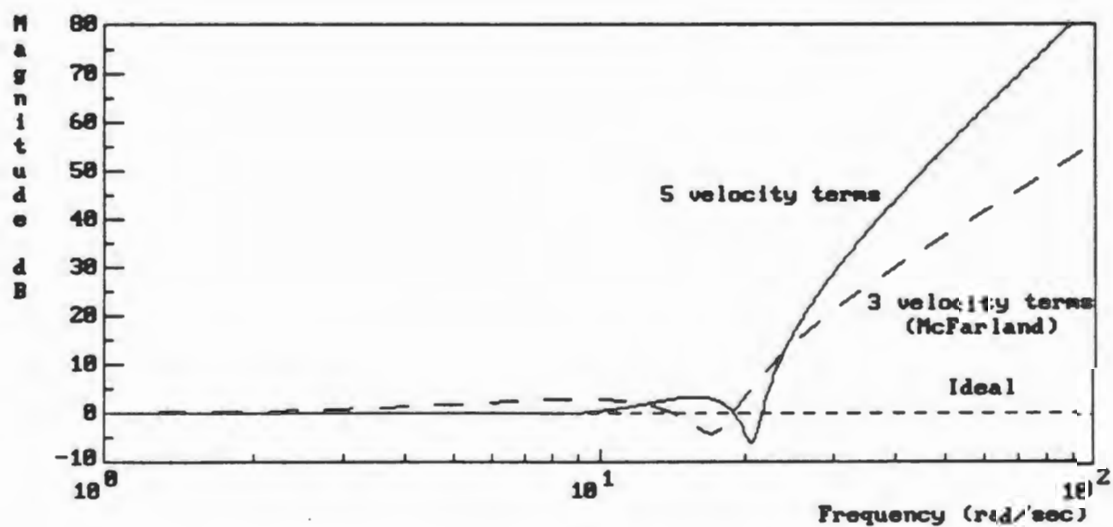


Figure 24. Bode Plot of Gain Characteristics, Five-velocity-term Compensator

For comparison purposes, code implementing this new compensator was added to the real-time host computer software (just as had been done for the lead/lag filter and McFarland compensators) and informal driver tests were conducted. The five-term algorithm did not perform well. In particular, there was considerable high-frequency noise or "jitter" in the visual display, particularly when steering wheel position was changed quickly. This jitter is almost certainly a manifestation of the amplitude distortion above 3 Hz, probably due to harmonic content in the driver input. It was concluded from this test that the compensator using five velocity terms was not suitable for use in the UCF simulator, as jitter in the visual display would surely be objectionable to drivers.

A Compensator Using Four Velocity Terms

While McFarland developed the theory of a general velocity-based compensator only for the case of an odd number of velocity terms [38], there is no reason why a similar algorithm could not make use of an even number of velocity values. The main difference in the design process is the way the constraints are chosen. Using a constant velocity constraint and tuning the compensator for correct magnitude and phase at one or more frequencies, as outlined in the previous section, requires an odd number of terms. On the other hand, any of several methods could be used to specify the constraints needed to solve for an even number of velocity coefficients.

One approach to designing a compensator with an even number ($2n$) of constraints is to use McFarland's design procedure, as extended in the previous

section, to specify the first $2n-1$ constraints. The last constraint could then be chosen to optimize either magnitude (gain) or phase, but not both, at some other design frequency. Both of these procedures (setting the magnitude of $h(\omega)$ to one and setting the phase of $h(\omega)$ to zero at an intermediate design frequency ω_1) were tried by the author in an attempt to formulate a workable compensator using four velocity terms. Neither design produced a good compensator response; fixing either the gain or phase at a frequency below the upper design frequency ω_0 resulted in the other response parameter varying far from the desired value, thus degrading rather than enhancing the compensator's performance as compared with the three-term design.

McFarland [47] also developed a compensator using four velocity terms using this type of approach, namely setting the compensator gain magnitude at a given frequency. However, this frequency was chosen above rather than within the "compensation band" (in this case, 0 - 3 Hz) in an attempt to address the amplitude distortion problem. "The additional constraint was that the magnitude at the Nyquist frequency vanish, independent of what the phase was doing [47]." A compensator was designed for the UCF simulator using this four-coefficient approach; its theoretical performance was compared to that of the corresponding design using three velocity terms. The Bode plots (Figures 25 and 26) show that the improvements in performance gained by using the additional velocity term in the compensator are marginal. While the amplitude distortion is decreased by 5-6 dB (to about +57 dB) at 20 Hz and by over 40 dB (to about +61 dB) near 30 Hz, it is actually increased slightly in the compensation band (below about 2 Hz) and just

above it (from about 3 to 7 Hz). These latter frequency ranges are much more critical to simulator operation than is the extreme high frequency range where the gain rolloff occurs. Moreover, it can be seen that the phase performance of the three-coefficient compensator is better than that of McFarland's four-coefficient formulation at all frequencies from DC to 30 Hz (except at the design points of 0 and 3 Hz, where the phase responses are identical). For these reasons, this method of setting the "extra" constraint for a compensator using an even number of velocity terms is of little help in compensating for transport delay in the driving simulator.

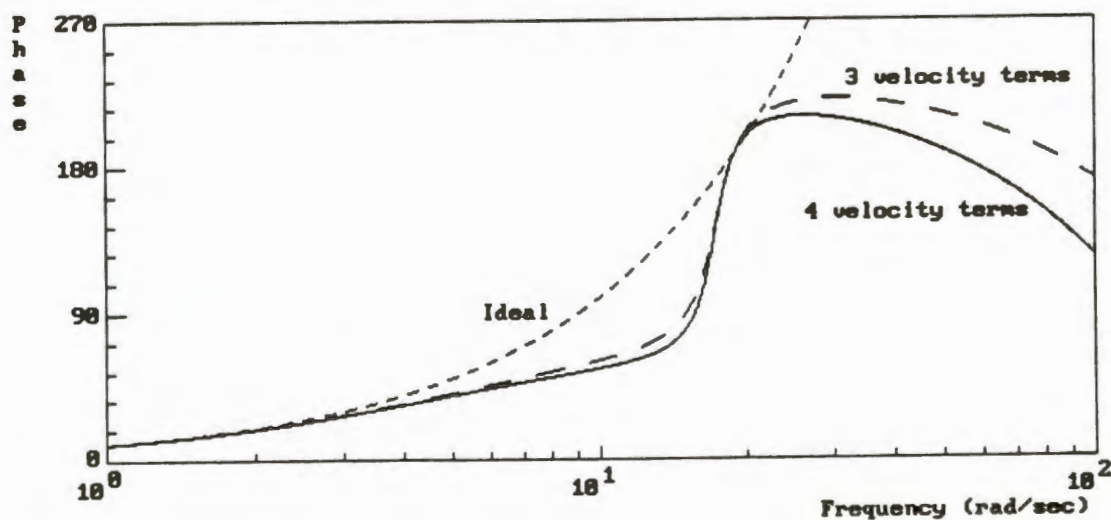


Figure 25. Bode Plot of Phase Characteristics, McFarland's Four-term Compensator

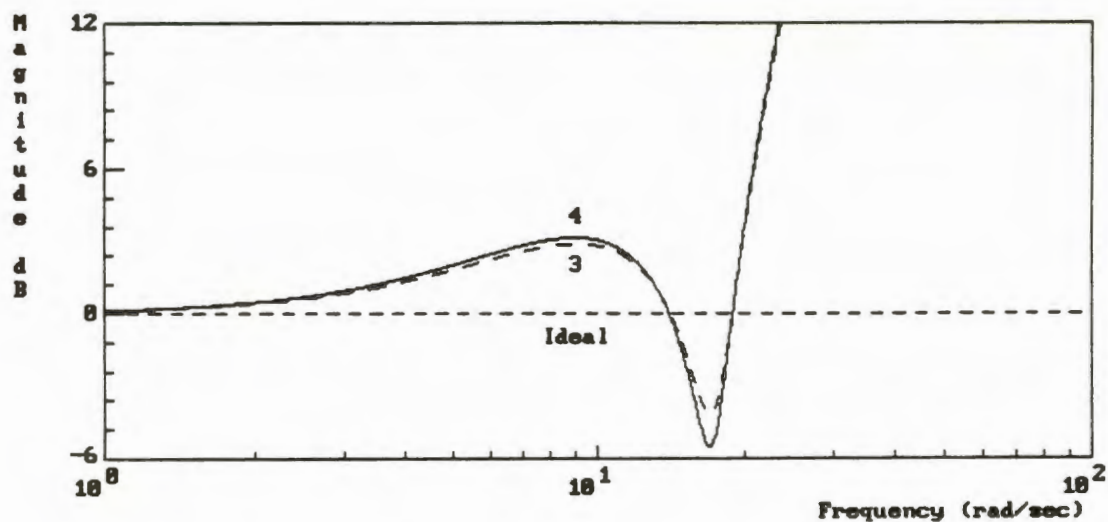


Figure 26. Bode Plot of Gain Characteristics, McFarland's Four-term Compensator

Another approach to developing an even number of coefficient constraint equations not mentioned in McFarland's work [38] [42] [44] [47] or in any of the other references is to ignore the constant velocity constraint expressed in Equation 4.13. With this constraint removed $\omega=0$ is no longer a design point; however, the designer may choose any number of design frequencies $\omega_0, \omega_1, \dots, \omega_n$. Two constraints are formed for each of these frequencies by setting the magnitude and phase of the relative error function $h(\omega)$ to one and zero as before. Simultaneous solution of the $2n$ equations thus formed yields the coefficients b_i for the velocity terms in the compensator.

Since McFarland's compensator with three velocity terms left some room for improvement while excessive amplitude distortion precluded the use of the five-term compensator in the driving simulator, the design approach just outlined was investigated for the case of two nonzero design frequencies ω_0 and ω_1 and four velocity terms with associated coefficients b_0 - b_3 . The compensator difference equation has the form:

$$u_{k+\frac{r}{T}} = u_k + b_0 v_k + b_1 v_{k-1} + b_2 v_{k-2} + b_3 v_{k-3} \quad (4.35)$$

The four constraint equations are determined in a manner similar to that outlined in the previous subsection, with the exception that the constant velocity constraint is omitted. The relative error function $h(\omega)$ is developed and its real and imaginary parts are set to 1 and 0, respectively, at each of the two design frequencies ω_0 and ω_1 . $H(\omega)$ is found from the z -domain transfer function

$$f_A(z) = z^{-\frac{r}{T}} \left[\frac{\frac{r}{2}(1+z^{-1}) + b_0(1-z^{-1}) + b_1 z^{-1}(1-z^{-1}) + b_2 z^{-2}(1-z^{-1}) + b_3 z^{-3}(1-z^{-1})}{1-z^{-1}} \right] \quad (4.36)$$

which is the same as Equation 4.16 except for the absence of the b_4 term in the numerator. Evaluating $j\omega f_A(z)$ at $z = e^{j\omega T}$ we get:

$$h(\omega) = \frac{\begin{aligned} & (1 - \cos\theta)[(\omega \sin\psi)d_0 + (\omega \cos\psi)d_1] + (\sin\theta)[(\omega \cos\psi)d_0 - (\omega \sin\psi)d_1] \\ & + j[(1 - \cos\theta)[(\omega \cos\psi)d_0 - (\omega \sin\psi)d_1] - (\sin\theta)[(\omega \sin\psi)d_0 + (\omega \cos\psi)d_1] \end{aligned}}{2(1 - \cos\theta)} \quad (4.37)$$

where $\theta = \omega T$, $\psi = \omega P$, and the other terms are defined as:

$$d_0 = \left(\frac{T}{2} + b_0\right) + \left(\frac{T}{2} + b_1 - b_0\right)(\cos \theta) + (b_2 - b_1)(\cos 2\theta) + (b_3 - b_2)(\cos 3\theta) + (-b_3)(\cos 4\theta) \quad (4.38)$$

and

$$d_1 = \left(\frac{T}{2} + b_1 - b_0\right)(\sin \theta) + (b_2 - b_1)(\sin 2\theta) + (b_3 - b_2)(\sin 3\theta) + (-b_3)(\sin 4\theta) \quad (4.39)$$

Setting the real part of $h(\omega)$ equal to 1 we obtain

$$b_0(K_0 - K_A) + b_1(K_A - K_B) + b_2(K_B - K_C) + b_3(K_C - K_D) = \frac{2(1 - \cos \theta)}{\omega} - \frac{T}{2}(K_0 + K_A) \quad (4.40)$$

where

$$K_0 = \sin \psi - \sin \psi \cos \theta + \cos \psi \sin \theta \quad (4.41)$$

$$K_1 = \cos \psi - \cos \psi \cos \theta + \sin \psi \sin \theta \quad (4.42)$$

$$K_A = K_0 \cos \theta + K_1 \sin \theta \quad (4.43)$$

$$K_B = K_0 \cos 2\theta + K_1 \sin 2\theta \quad (4.44)$$

$$K_C = K_0 \cos 3\theta + K_1 \sin 3\theta \quad (4.45)$$

$$K_D = K_0 \cos 4\theta + K_1 \sin 4\theta \quad (4.46)$$

Equation 4.40 applies to each of the two design frequencies ω_0 and ω_1 and thus provides the first two of the four necessary constraints for determining b_0 - b_3 .

Now, setting the imaginary part of $h(\omega)$ equal to zero we obtain

$$b_0(K_1 - K_F) + b_1(K_F - K_G) + b_2(K_G - K_H) + b_3(K_H - K_J) = -\frac{T}{2}(K_1 + K_F) \quad (4.47)$$

where K_0 and K_1 are defined above and the remaining constants are

$$K_F = K_1 \cos \theta - K_0 \sin \theta \quad (4.48)$$

$$K_G = K_1 \cos 2\theta - K_0 \sin 2\theta \quad (4.49)$$

$$K_H = K_1 \cos 3\theta - K_0 \sin 3\theta \quad (4.50)$$

$$K_J = K_1 \cos 4\theta - K_0 \sin 4\theta \quad (4.51)$$

Equation 4.47 is also applied at each of the two design frequencies ω_0 and ω_1 , providing the other two constraints necessary to solve for the four coefficients b_0 - b_3 .

A program was written in C to compute the coefficients of the b_0 - b_3 terms in the four constraint equations for any given T , P , ω_0 , and ω_1 . Once all the appropriate values are obtained, the four equations can be expressed in matrix form and solved simultaneously using a matrix-solving program, yielding the set of compensator coefficients b_0 - b_3 .

Because the constant velocity or DC constraint was omitted from the design process for this new compensator, it is necessary to examine Bode plots of the compensator's transfer function in order to verify that it will perform acceptably with low-frequency vehicle motions. Therefore, the compensator's z-domain transfer

function ($U_{\text{comp}}(z)/U(z)$) was evaluated in terms of the coefficients b_0 - b_3 and the sampling period T :

$$\frac{U_{\text{comp}}(z)}{U(z)} = \frac{(1 + \frac{2b_0}{T})z^4 + (1 + \frac{2b_1}{T} - \frac{2b_0}{T})z^3 + (\frac{2b_2}{T} - \frac{2b_1}{T})z^2 + (\frac{2b_3}{T} - \frac{2b_2}{T})z - (\frac{2b_3}{T})}{z^4 + z^3} \quad (4.52)$$

Another program was written to aid in computing the transfer function numerator coefficients in Equation 4.52. Finally, a control systems design program, CC, was used to draw Bode plots (Figures 27 and 28) for the compensator.

The four-term compensator designs for the driving simulator were done using an upper design frequency f_0 of 3 Hz ($\omega_0 = 18.849556$ rad/s), the same design frequency used in the previous examples for the three- and five-coefficient compensators. The second, lower design frequency ω_1 was chosen to be 2.0 rad/s, the estimated system crossover frequency, to allow direct performance comparisons with the other compensator designs.

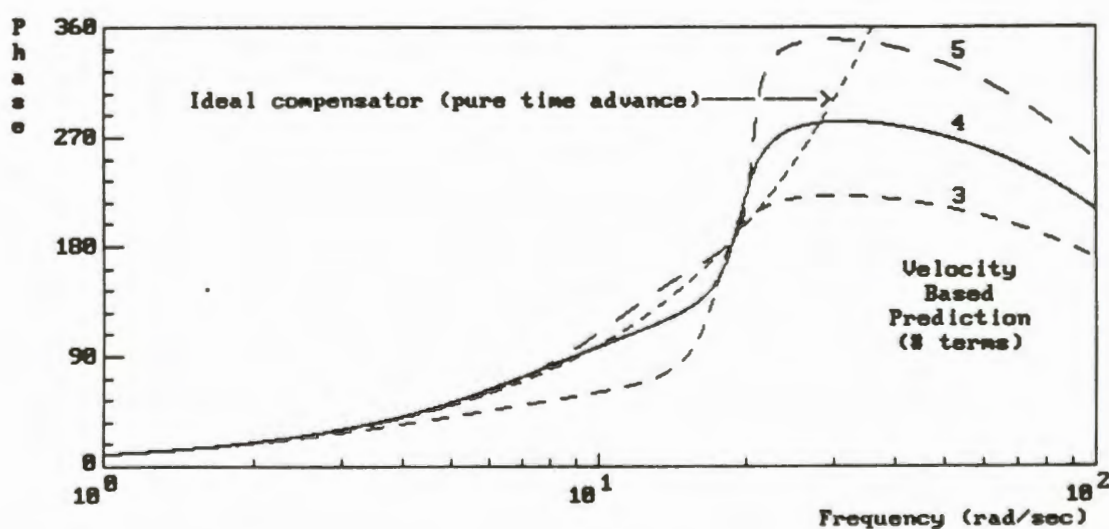


Figure 27. Bode Plot of Phase Characteristics, Author's Four-term Compensator

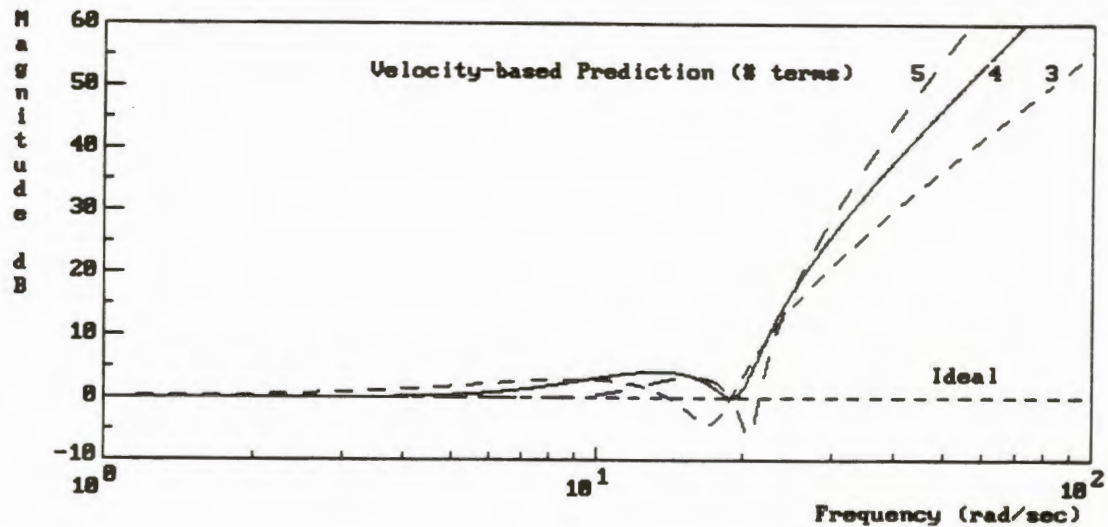


Figure 28. Bode Plot of Gain Characteristics, Author's Four-term Compensator

The first thing to note from the gain magnitude Bode plot (Figure 28) is that the concern regarding possible poor performance at low frequencies was unfounded. Gain magnitude is essentially a constant 0 dB at very low frequencies (over the range of 0.00001 to 0.1 rad/s) for the design parameters chosen for the driving simulator. Thus, low-frequency amplitude distortion should not be noticeable during simulator operation.

Another check on the low-frequency performance of the compensator is to check how closely the constant velocity constraint (which was not imposed in the design process) is satisfied. For perfect performance at constant velocity ($\omega=0$) we would require $b_0 + b_1 + b_2 + b_3 = P$, the transport delay. For the case of the shorter transport delay ($P = 0.133$ second) the sum of b_0 - b_3 is 0.13209; for the longer delay

case ($P = 0.174$ second) their sum is 0.17139. The errors in each case are very small (0.7% and 1.5%, respectively). Thus, no problems were anticipated with using the four-term compensator in the simulator.

Also notable from the Bode plot of compensator gain magnitude is the behavior of the four-term algorithm at high frequencies. It can be seen from Figure 28 that the amplitude distortion induced by this compensator above f_0 (3 Hz in this case) is more than that of the three-term compensator but less than that displayed by the five-term compensator. This is an important but not surprising result, since it opens up the possibility that the four-term compensator might be usable in the driving simulator where the five-term compensator was not.

Looking at the Bode plot of compensator phase angle (Figure 27) it is apparent that the compensator using four velocity terms performs much better in this respect than McFarland's three- or four-coefficient designs, and nearly as well as the five-term compensator that was rejected due to excessive amplitude distortion. For frequencies up to about 7-8 radians per second the four-term compensator just derived is virtually indistinguishable from the five-term design. Above this frequency (up to 3 Hz) the four-coefficient scheme does not provide quite as much lead or match the ideal response quite as closely as does the five-coefficient compensator, but it still performs much better than McFarland's original compensator using three terms. Overall, the Bode plots suggested the possibility that the new compensator design might be more desirable than McFarland's compensator for use in the driving

simulator, assuming that the amplitude distortion at high frequencies did not cause noticeable visual effects.

Code implementing the four-coefficient compensator was added to the real-time host computer software (as had been done for the other designs) and informal driver tests were conducted. The results were very encouraging. There was very little "jitter" or other noise in the visual display during normal driving maneuvers. It was concluded from these informal tests that the compensator using four velocity terms did show promise for use in the UCF simulator and was therefore worth testing more fully (see Chapter 5).

One problem was observed with the operation of all the velocity-based compensation algorithms in the real-time simulator environment: erratic behavior of the visual display at very low vehicle speeds. This problem was not due to the operation of the compensator itself but rather to the fact that it operates using the velocity terms as well as the position computed by the vehicle dynamics model. The problem was traced to a numerical instability in the acceleration-to-velocity integrations in the dynamics code. This oscillation does not appear in the position values computed by the second integrator; it damps out with increasing speed and does not present a problem (other than to the compensator) during normal operations. This effect of this condition on the compensator was averted by the simple means of bypassing the compensator at low vehicle speeds. At very low speeds, compensation is not important since the vehicle's position changes very little between frames; thus, the transport delay has little effect on the system and can be

ignored. Therefore, the uncompensated vehicle coordinates are sent to the IG. At speeds above those where the model instability occurs, the compensated coordinates are transmitted. At intermediate speeds, a weighted combination of the actual and projected vehicle coordinates is sent to the IG in order to provide a smooth transition that is not noticeable to the driver. This "fix" worked very well in informal tests and was incorporated in the final version of the real-time software which was used to conduct driver testing of the three- and four-velocity-term compensators.

CHAPTER 5

TESTING THE EFFECTIVENESS OF DELAY COMPENSATION

While theoretical analysis of the various forms of delay compensation using such tools as Bode plots is a necessary task, it is not sufficient for proving the worth of the compensators in a real-time system. The real "proof of the pudding" is found in actually exercising the system in real time, with and without compensation, to determine if significant improvement in handling of the simulated vehicle has resulted.

The results of using the delay compensation techniques developed for the UCF Driving Simulator were tested in two different ways. First, measurements of effective delay in the compensated system were made using the same steady-state technique used to measure delay in the uncompensated system. The time-domain approach is not appropriate for testing the system with compensation since the vehicle dynamics model, which generates the velocity terms used in the compensator, is bypassed. Even the lead/lag compensator, which does not use velocity terms explicitly, is designed to operate under steady-state conditions. The steady-state delay measurements were performed at several test frequencies to determine how effective each compensator is across the spectral range of typical driver inputs. The results of these tests are discussed below.

In addition to measuring the actual time delay reduction (or improved phase characteristics) achieved with the various delay compensation schemes, it was considered important to try to determine whether the simulated vehicle with compensation "drives better" in terms of improved control performance by drivers. Therefore, experiments using human subjects were performed to determine the practical effectiveness of delay compensation. Experiments similar to those described in references [8], [17], and other sources, in which subjects were asked to perform simple driving tasks, were conducted. Suitable measures of driving performance such as steering wheel reversals, steering input and lateral position standard deviations, etc. were recorded and analyzed using statistical techniques. The conduct and results of these tests are described below.

Steady-State Delay Measurements of the Compensated System

The steady-state transport delay tests described in Chapter 3 were repeated with the lead/lag delay compensation filter added to the host computer program. The results are recorded in Tables 10 through 13 for the visual databases with and without the movable car models.

Table 10.

Steady-State Measurements (No Moving Objects, Lead/Lag Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-180	-181	+1	+2	-181	+1
	-187	-189	+2	+2	-189	+2
	-188	-191	+3	+2	-191	+3
2 (0.318 Hz)	-182	-181	-1	-2	-183	+1
	-199	-198	-1	-2	-199	0
	-204	-203	-1	-1	-204	0
6.283 (1.000 Hz)	-212	-173	-39	-40	-177	-35
	-259	-221	-38	-39	-225	-34
	-289	-250	-39	-40	-254	-35

Table 11.

Transport Delay (No Moving Objects, Lead/Lag Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	+17 [157]	+35 [192]	+17 [157]	185.2
	+35 [175]	+35 [192]	+35 [175]	
	+52 [209]	+35 [192]	+52 [209]	
2 (0.318 Hz)	-9 [140]	-17 [131]	+9 [140]	133.0
	-9 [140]	-17 [131]	0 [140]	
	-9 [140]	-9 [140]	0 [131]	
6.283 (1.000 Hz)	-108 [33]	-111 [31]	-97 [33]	31.8
	-106 [36]	-108 [33]	-94 [33]	
	-108 [33]	-111 [33]	-97 [33]	

Table 12.

Steady-State Measurements (Moving Objects, Lead/Lag Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-179	-181	+2	+2	-181	+2
	-187	-189	+2	+3	-189	+2
	-188	-191	+3	+2	-191	+3
2 (0.318 Hz)	-182	-181	-1	-1	-183	+1
	-200	-198	-2	-1	-199	-1
	-203	-203	0	-1	-204	+1
6.283 (1.000 Hz)	-221	-173	-48	-49	-177	-44
	-268	-221	-47	-48	-225	-43
	-298	-250	-48	-49	-254	-44

Table 13.

Transport Delay (Moving Objects, Lead/Lag Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	+35 [227]	+35 [244]	+35 [227]	234.0
	+35 [227]	+52 [244]	+35 [227]	
	+52 [254]	+35 [244]	+52 [244]	
2 (0.318 Hz)	-9 [183]	-9 [183]	+9 [183]	174.0
	-17 [157]	-9 [175]	-9 [157]	
	0 [183]	-9 [183]	+9 [175]	
6.283 (1.000 Hz)	-133 [50]	-136 [47]	-122 [50]	46.1
	-131 [53]	-133 [50]	-122 [47]	
	-133 [50]	-136 [47]	-122 [50]	

The steady-state test results show that the lead/lag delay compensator operates approximately as designed. A small amount of net lead (overcompensation) is apparent at the lowest test frequency (1 radian per second), while the net compensated IG delay is close to zero at 2 rad/s (the design crossover frequency). Phase lag due to IG delay was observed with the highest frequency (1 Hz) input, but it was less (by about 11 to 12 degrees) than the lag observed without compensation.

The observed reductions in delay (or elimination of delay with some net lead at the 1 rad/s test frequency) were close to the values theoretically provided by the lead/lag filters. For example, the filter used with the longer delay theoretically provides leads of 234 ms at 1 rad/s, 174 ms at 2 rad/s, and 46.1 ms at 1 Hz. The observed improvements were in the range of 227-254 ms at 1 rad/s, 157-183 ms at 2 rad/s, and 47-53 ms at 1 Hz. Considering that the HP Spectrum Analyzer displays readings only to the nearest degree and that several readings (at forward speeds of 30, 60, and 90 ft/s) were averaged to come up with each of these figures, the results are quite good. Within the limitations of available instrumentation, the operation of the lead/lag compensator was considered a success pending the results of driver testing (see below).

Steady-state delay measurements made with McFarland's three-velocity-term compensator are recorded in Tables 14 through 17 for both visual database configurations. The steady-state test results show that McFarland's compensator, like the lead/lag compensator, operates approximately as designed. The compensation is very accurate at the two lower test frequencies (1 and 2 radians per second). Some phase lag due to IG delay was observed with the highest frequency (1 Hz) input, but it was much less than the lag observed without compensation and considerably smaller than the lag observed with lead/lag compensation. The observed reductions in delay were close to the values theoretically provided by the compensators.

Table 14.

Steady-State Measurements (No Moving Objects, 3-term Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-182	-181	-1	-2	-181	-1
	-191	-189	-2	-1	-189	-2
	-191	-191	0	-1	-191	0
2 (0.318 Hz)	-183	-181	-2	-2	-183	0
	-200	-198	-2	-2	-199	-1
	-204	-203	-1	-2	-204	0
6.283 (1.000 Hz)	-182	-173	-9	-9	-177	-5
	-229	-221	-8	-8	-225	-4
	-259	-250	-9	-10	-254	-5

Table 15.

Transport Delay (No Moving Objects, 3-term Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	-17 [122]	-35 [122]	-17 [122]	132.3
	-35 [105]	-17 [140]	-35 [105]	
	0 [157]	-17 [140]	0 [157]	
2 (0.318 Hz)	-17 [131]	-17 [131]	0 [131]	130.5
	-17 [131]	-17 [131]	-9 [131]	
	-9 [140]	-17 [131]	0 [131]	
6.283 (1.000 Hz)	-25 [117]	-25 [117]	-14 [117]	114.9
	-22 [119]	-22 [119]	-11 [117]	
	-25 [117]	-28 [117]	-14 [117]	

Table 16.

Steady-State Measurements (Moving Objects, 3-term Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-181	-181	0	-1	-181	0
	-191	-189	-2	-1	-189	-2
	-193	-191	-2	-1	-191	-2
2 (0.318 Hz)	-183	-181	-2	-3	-183	0
	-201	-198	-3	-2	-199	-2
	-204	-203	-1	-2	-204	0
6.283 (1.000 Hz)	-192	-173	-19	-19	-177	-15
	-238	-221	-17	-18	-225	-13
	-268	-250	-18	-19	-254	-14

Table 17.

Transport Delay (Moving Objects, 3-term Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	0 [192]	-17 [192]	0 [192]	172.1
	-35 [157]	-17 [175]	-35 [157]	
	-35 [157]	-17 [192]	-35 [157]	
2 (0.318 Hz)	-17 [175]	-26 [166]	0 [175]	167.0
	-26 [148]	-17 [166]	-17 [148]	
	-9 [175]	-17 [175]	0 [166]	
6.283 (1.000 Hz)	-53 [131]	-53 [131]	-42 [131]	130.9
	-47 [136]	-50 [133]	-36 [133]	
	-50 [133]	-53 [131]	-39 [133]	

The compensator used with the shorter delay theoretically provides leads of 132.3 ms at 1 rad/s, 130.5 ms at 2 rad/s, and 114.9 ms at 1 Hz; the observed improvements were in the range of 105-157 ms at 1 rad/s, 131-140 ms at 2 rad/s, and 117-119 ms at 1 Hz. In the case of the longer delay, the compensator's theoretical lead time was 172.1 ms at 1 rad/s, 167.0 ms at 2 rad/s, and 130.9 ms at 1 Hz. The observed ranges of reduction in lag were 157-192 ms at 1 rad/s, 148-175 ms at 2

rad/s, and 131-136 ms at 1 Hz. Again, considering the HP Spectrum Analyzer's accuracy and resolution limitations, the results are quite good. Within the limitations of available instrumentation, the operation of McFarland's three-velocity-term compensator was considered a success pending the results of driver testing (see below).

Steady-state delay measurements made with the author's four-velocity-term compensator are recorded in Tables 18 through 21 for both visual database configurations. The steady-state test results show that this compensator, like the previous ones, operates approximately as designed. The results are quite similar to those obtained for the three-coefficient compensator except that they show a much closer match to the phase characteristics of an ideal compensator at the highest test frequency (one Hertz). In fact, the compensation is very nearly ideal at all three test frequencies. The observed reductions in delay were close to the values theoretically provided by the compensators.

Table 18.

Steady-State Measurements (No Moving Objects, 4-term Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-182	-181	-1	-1	-181	-1
	-190	-189	-1	-1	-189	-1
	-192	-191	-1	-1	-191	-1
2 (0.318 Hz)	-183	-181	-2	-2	-183	0
	-199	-198	-1	-2	-199	0
	-204	-203	-1	-1	-204	0
6.283 (1.000 Hz)	-174	-173	-1	-2	-177	+3
	-223	-221	-2	-3	-225	+2
	-253	-250	-3	-4	-254	+1

Table 19.

Transport Delay (No Moving Objects, 4-term Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	-17 [122]	-17 [140]	-17 [122]	132.3
	-17 [122]	-17 [140]	-17 [122]	
	-17 [140]	-17 [140]	-17 [140]	
2 (0.318 Hz)	-17 [131]	-17 [131]	0 [131]	133.0
	-9 [140]	-17 [131]	0 [140]	
	-9 [140]	-9 [140]	0 [131]	
6.283 (1.000 Hz)	-3 [139]	-6 [136]	+8 [139]	137.7
	-6 [136]	-8 [133]	+6 [133]	
	-8 [133]	-11 [133]	+3 [133]	

Table 20.

Steady-State Measurements (Moving Objects, 4-term Compensator)

Steering Input Freq. (rad/s)	Phase Lead (+) or Lag (-) (degrees) (at UCAR = 30, 60, 90 feet per second)					
	Measured Total Phase (STEERW to IG)	Measured Model Phase (STEERW to YCAR)	Meas. Total minus Meas. Model	Measured IG Phase (YCAR to IG)	Actual Model Phase (from offline simu- lation)	Meas. Total minus Actual Model
1 (0.159 Hz)	-183	-181	-2	-1	-181	-2
	-189	-189	0	-1	-189	0
	-192	-191	-1	-1	-191	-1
2 (0.318 Hz)	-184	-181	-3	-1	-183	-1
	-200	-198	-2	-1	-199	-1
	-204	-203	-1	-2	-204	0
6.283 (1.000 Hz)	-173	-173	0	+1	-177	+4
	-220	-221	+1	0	-225	+5
	-250	-250	0	0	-254	+4

Table 21.

Transport Delay (Moving Objects, 4-term Compensator)

Steering Input Freq. (rad/s)	Effective System Advance (+) or Delay (-) (ms) [Improvement due to Compensator (ms)] (at UCAR = 30, 60, 90 feet per second)			
	From Steady- State Difference (Total minus model)	From Steady- State Direct Meas. (YCAR to IG)	Steady-State Total minus Actual Model	Theoretical Improvement from Compensator (ms)
1 (0.159 Hz)	-35 [157]	-17 [192]	-35 [157]	172.1
	0 [192]	-17 [175]	0 [192]	
	-17 [175]	-17 [192]	-17 [175]	
2 (0.318 Hz)	-26 [166]	-9 [183]	-9 [166]	174.0
	-17 [157]	-9 [175]	-9 [157]	
	-9 [175]	-17 [175]	0 [166]	
6.283 (1.000 Hz)	0 [183]	+3 [186]	+11 [183]	183.4
	+3 [186]	0 [183]	+14 [183]	
	0 [183]	0 [183]	+11 [183]	

The filter used with the shorter delay theoretically provides leads of 132.3 ms at 1 rad/s, 133.0 ms at 2 rad/s, and 137.7 ms at 1 Hz; the observed improvements were in the range of 122-140 ms at 1 rad/s, 131-140 ms at 2 rad/s, and 133-139 ms at 1 Hz. In the case of the longer delay, the filter's theoretical lead time was 172.1 ms at 1 rad/s, 174.0 ms at 2 rad/s, and 183.4 ms at 1 Hz. The observed ranges of reduction in lag were 157-192 ms at 1 rad/s, 157-183 ms at 2 rad/s, and 183-186 ms at 1 Hz.

Once again, considering the HP Spectrum Analyzer's accuracy and resolution limitations, the results are very good. The operation of the four-velocity-term compensator was considered a success pending the results of driver testing (see the following section).

Driver-In-The-Loop Experiment

The results of early informal driver testing as well as the steady-state testing described in the previous section indicated some possible improvement in the driveability of the simulated car due to the addition of delay compensation. The final step in evaluating the merits of the compensator was a series of performance tests using volunteer drivers not experienced with the simulator. Performance measures described in the literature [8] [48] [49] [50] [51] were chosen to evaluate the ability of the drivers to control the simulated vehicle. For each experimental run, these measures of performance were derived from logged data. The design, conduct, and results of the driver-in-the-loop experiment are described in the following subsections.

Experimental Design and Procedure

The objective of the human factors experiment was to determine significant differences, if any, between the four experimental conditions or "treatments": no delay compensation, lead/lag compensation, compensation by prediction using three velocity terms (McFarland), and the author's predictive compensator using four

velocity terms. A blocked design was used in order to isolate possible learning effects from the effects of compensation. Each of twelve drivers was to make one test run over a course to be described below in each of the four conditions, with the order of presentation counterbalanced [52] in order to cancel out learning effects. (Each of the four treatments was scheduled to occur on the first run three times, on the second run three times, on the third run three times, and on the final run three times.) Practice runs (given to each driver before the first data collection run in order to familiarize him/her with the equipment) used the same condition as the first test run in every case. This was intended to equalize overall practice time across the four treatments [52] and thus remove a possible source of bias. Drivers and treatments were assigned to the schedule randomly.

Due to circumstances beyond the author's control it proved impossible to complete the full experimental schedule. A critical simulator component (the video projection system) failed while the eighth subject was driving and could not be repaired in a timely fashion. Thus, the experimental results collected and analyzed below consist of only 28 test runs (four for each of seven driver/subjects). This, of course, had a negative impact on the experimental design: not only was the intended counterbalancing effect (which depended on the number of subjects being a multiple of four) compromised, but the statistical power of the experiment was adversely affected due to the reduction in the total number of experimental runs from 48 to 28. Nevertheless, statistical analysis of the data was attempted (see the following

subsection) with the experiment treated as a randomized block design using seven blocks and four treatments.

The driving course used in the experiment had to be chosen as a subset of the available driving scenario database developed by other researchers [20] [21] for the simulator. In order to evaluate driver handling of the vehicle under a range of conditions, the test course was made up of both straight and curved portions. There were four straight sections of road with varying lengths, each followed by a "curved" section (approximated by several polygons). The test commenced with the simulated vehicle parked in the center of its lane at the beginning of the first straight road segment and finished on a straight road segment following the fourth curve. A more detailed description of each section of the test course follows:

<u>Section</u>	<u>Description</u>
1.	Straight road, dashed center line, 2600' long.
2.	Moderate 90 degree curve to right, nearly constant radius, approximated by 18 polygons.
3.	Straight road, unmarked, 900' long.
4.	Moderate 90 degree curve to right (similar to section 2).
5.	Straight road, unmarked, 600' long.
6.	Easy 90 degree curve to left (approximated by 15 polygons).
7.	Straight road, dashed center line, 1050' long.
8.	Sharp 90 degree curve to right (approximated by 9 polygons).
9.	Exit section 8 onto 1170' straight unmarked section of road. Terminate test when stable on this section.

A number of measures of driver/vehicle performance that had previously been used in experiments involving simulators and instrumented vehicles are described in the literature. Based on availability of data and suitability for the experiment, several of these measures were chosen to be computed for each experimental run. These measures are described in more detail in the following paragraphs.

Standard deviation of lateral (lane) position deviations is a direct measure of the driver's lateral control of the vehicle. It has been used as a performance index by a number of human factors researchers including Wierwille and others [48] [49]. This measure was selected instead of the mean lateral position because drivers tend to choose various lane positions as "centered". Some like to stay closer to the centerline while some prefer to hug the edge of the road. A driver who maintains a slightly off-center position well is arguably controlling the vehicle "better" than one whose average position is in the exact center of the lane but who weaves back and forth about this position significantly. Standard deviation was made even more desirable (versus average position) by the fact that some portions of the test course had no marked centerline, thus making it difficult for the driver to determine his/her absolute lane position.

Because of the polygonal nature of the simulator visual database it proved difficult to quickly compute lane position on the "curved" portions of road. On the other hand, lateral position standard deviation was easily calculated for the straight portions of the test course since lateral motion on those sections is exclusively in the X or Y direction at any given time. For simplicity it was decided to monitor lane

position only on the four straight road sections; the final performance index was computed as the average of these four values, weighted by the length of each section as a fraction of the total length of all four straight sections. A smaller value of this lateral standard deviation performance index indicates better control of the vehicle.

Steering wheel reversals is a commonly used [8] [48] measure of the control effort required of the driver while negotiating the test course. This performance index is computed simply as the number of times that the steering wheel position moves more than a certain amount in the opposite direction to its previous movement. Two degrees (0.03491 radian), a value reported in the literature for other driving studies [48], was chosen as the threshold value for this experiment. This performance index was easily determined for both straight and curved sections of the test course; the score recorded for each driver was the total number of reversals over the entire course (beginning of section 1 to end of section 8). Fewer steering reversals does not necessarily imply better control of the vehicle in the sense of smaller lane position deviations, but does indicate that less control effort was required. Thus, the number of steering reversals is a possible indicator of the handling qualities of a given configuration.

Steering wheel angle standard deviation is another measure of the steering effort required to control the vehicle over the test course. Steering deviations were monitored in the experiments described in [48] and [50]. It would be possible to compute a single value for steering wheel angle standard deviation over the entire test course; however, since straight and curved road sections will have different mean

values of steering wheel position, this approach may give misleading results. Accordingly, it was decided to compute the mean and corresponding standard deviation of steering angle separately for each section of the course. The average of the eight standard deviation values (weighted by the length of each section as a fraction of the total course length) was used as a performance index. Obviously there is a minimum amount of steering effort necessary to negotiate the test course; however, excess steering input may be indicative of handling problems with a particular vehicle configuration. Therefore, a smaller value of this performance index was taken to indicate better vehicle handling characteristics.

Lateral acceleration standard deviation may also be used as a measure of performance. A number of driver experiments reported in the literature (reference [51], for example) included monitoring of lateral acceleration. Ideally, the vehicle's lateral acceleration would be zero on straight sections of road and constant (or nearly so) on curves. Excessive variations in lateral acceleration, like those in steering angle, may indicate problems with handling the vehicle. For this experiment, the standard deviation of lateral acceleration was computed separately for each straight and curved section of the test course. As in the case of steering wheel angle standard deviation, the weighted average of these eight values was taken as the performance index for each experimental run.

Longitudinal velocity standard deviation is a measure of the ability of the driver to control the vehicle's speed while negotiating the test course. Speed control was not the primary assigned driving task but can be treated as a secondary task which

may be indicative of the load imposed by the primary task (staying on the road). If maintaining directional control is difficult because of transport delay or compensator effects, driver performance in controlling the secondary variable (speed) can be expected to deteriorate. Thus, even though two different test conditions may not show significant differences in directional control (because it proved possible for the driver, by increasing concentration/effort, to control the vehicle as well in the less optimal configuration), the difference between the two conditions may show up indirectly as an increase in vehicle speed deviations. This measure was computed separately for each of the eight sections of the test course. The values for the first straight section were found to be extremely large because of the necessary acceleration from the initial condition (parked) to cruising speed. Therefore, the performance index recorded for each run was computed as the weighted average of the longitudinal velocity standard deviations for only the remaining seven sections of road.

In order to be able to compute the performance measures described above it was necessary to "log" or record several of the simulation variables in real time. The six quantities listed below were sampled at an interval of 0.1 second for the duration of each run (approximately 2 to 3 minutes). Data were saved in a text file at the end of each run and used to compute the performance indices described above during post-processing.

<u>Variable Name</u>	<u>Quantity</u>	<u>Units</u>
sim_time	simulation run time	seconds
STEERW	steering wheel angle	radians
LATACC	lateral acceleration	ft/s ²
UCAR	longitudinal velocity	ft/s
XCAR	inertial X position	feet
YCAR	inertial Y position	feet

Data Analysis

The main subject of interest in this experiment was the difference in driver/vehicle performance attributable to the four "treatments" (three different compensation algorithms plus the uncompensated case). Variations in performance between drivers were believed to exist but were not of interest in this study; thus the experiment was designed as a randomized block experiment (described in Box, Hunter, and Hunter [53] and McClave and Dietrich [54]). For each of the 28 data collection runs, a value was computed for each performance measure discussed in the previous subsection. Each collection of 28 data values (for example, the values for number of steering wheel reversals) was subjected to a statistical analysis similar to that detailed in [53], chapter 7. By referring the ratio s^2_T/s^2_R (treatment mean square/residual (error) mean square) to the appropriate F-distribution, the null hypothesis that all the treatment means are equal (no significant effects due to delay compensation) was tested for each performance measure. The statistical analysis was performed using the MINITAB package [55] to compute the appropriate means, sample standard deviations, and F-statistics.

The data for lateral position standard deviation (a direct measure of control effectiveness) were input to the MINITAB statistical program and subjected to a two-way analysis of variance. Little could be said about the assumption of normality given the limited amount of data, but sample variances were fairly consistent across the four treatments. A plot of residuals versus fitted values (not shown) also revealed no apparent evidence of nonadditivity between driver (block) and compensator (treatment) effects. Therefore, statistical inferences can reasonably be drawn from analysis of the ANOVA table and related plots generated by MINITAB.

ANALYSIS OF VARIANCE, lateral position st. dev.

SOURCE	DF	SS	MS	F	p
drivers	6	5.1647	0.8608	16.120	0.000
comps	3	0.0920	0.0307	0.575	0.639
ERROR	18	0.9609	0.0534		
TOTAL	27	6.2176			

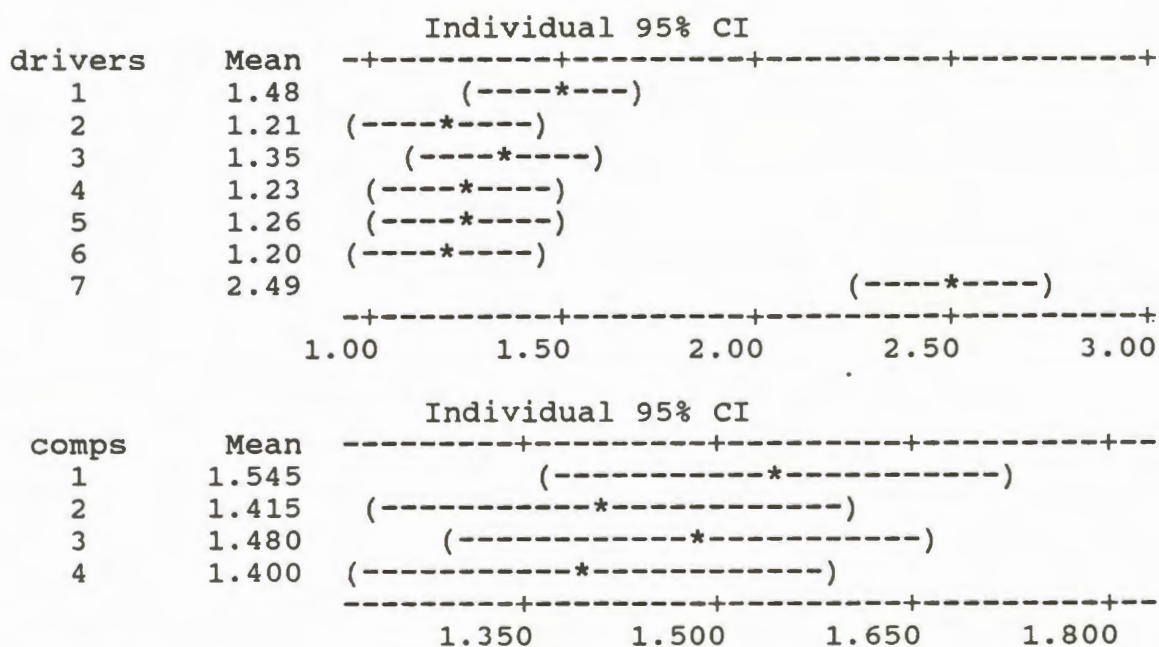


Figure 29. ANOVA Table and Confidence Intervals for Lateral Position Deviations

The first observation apparent from the ANOVA table and plot of driver confidence intervals (Figure 29) is that the decision to block by drivers was definitely justified. The high F value (16.12) for block effects lets us reject the null hypothesis that the driver effects are equal with near certainty. A large portion of the overall variability in the data can be attributed to the drivers rather than random chance.

With regard to the subject of primary interest (compensator effects), however, the relatively low F value for treatment effects does not allow us to reject with any degree of statistical confidence the null hypothesis that all treatment means are equal. There does seem to be a trend toward smaller lateral position deviations with compensation since the mean for treatment 1, no compensation, was higher than that for any of treatments 2 through 4 (the delay-compensated configurations). However, the large amount of variability in the data (note the overlap of the individual confidence intervals for the treatment means) does not allow us to draw the conclusion that delay compensation definitely improves lateral control of the simulated vehicle. In fact, there is approximately a 64% probability that differences of the magnitude observed in the experiment could be due solely to chance.

The data for steering wheel reversals were also input to MINITAB and subjected to a two-way analysis of variance. Once again, the checks for homogeneous variance between treatments and additivity of block and treatment effects indicated that analysis of the randomized block experiment using the ANOVA table and confidence interval plots was reasonable. The MINITAB printout is shown below in Figure 30.

ANALYSIS OF VARIANCE, steering wheel reversals

SOURCE	DF	SS	MS	F	p
drivers	6	5253	875	7.000	0.001
comps	3	147	49	0.392	0.760
ERROR	18	2245	125		
TOTAL	27	7644			

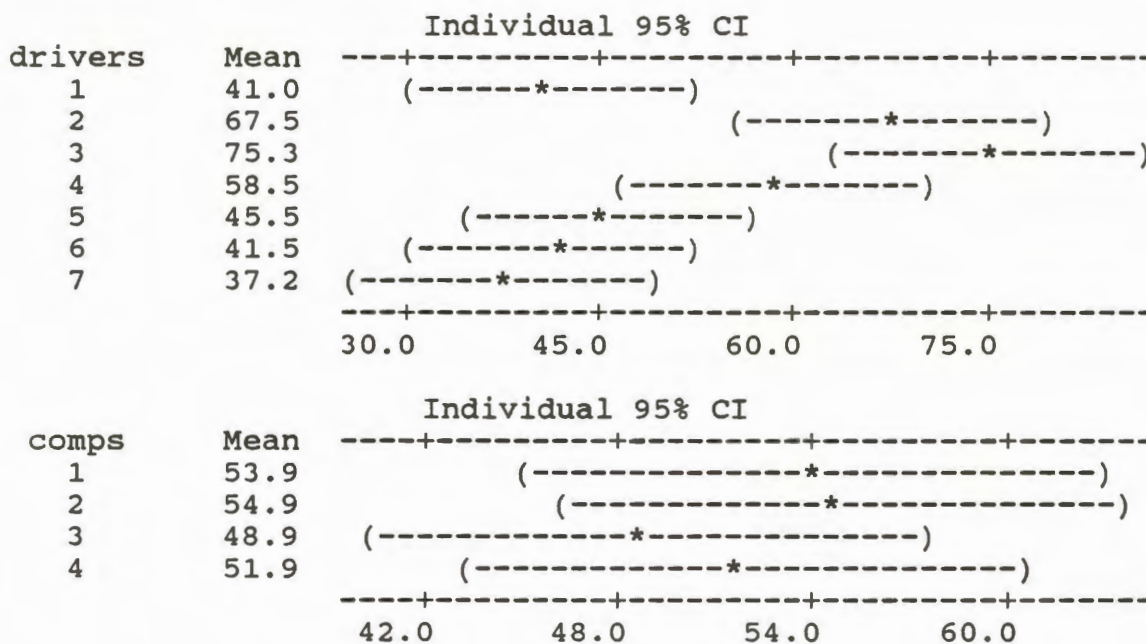


Figure 30. ANOVA Table and Confidence Intervals for Steering Wheel Reversals

Again, it is apparent from the ANOVA table and plot of driver confidence intervals that the blocked design was appropriate. We can reject the null hypothesis of no driver effects with 99.9% confidence. A large portion of the overall variability in the data can be attributed to the drivers rather than to chance.

There appears to be a trend in the data in favor of the velocity-based compensators. (Treatment 3 is McFarland's three-term compensator and treatment 4 is the four-velocity-term compensator designed by the author.) On the other hand,

the means for treatments 1 (no compensation) and 2 (lead/lag compensation) are nearly equal; in fact, the lead/lag compensator appears to be slightly "worse than nothing" with regard to steering reversals. Once again, however, the low F value (0.392) for treatment effects does not allow us to reject the null hypothesis that all treatment means are equal. The large amount of variability in the data (note the almost complete overlap of the individual confidence intervals for the treatment means) indicates a high probability that the observed differences in the number of steering reversals could be due to chance rather than any real effects of compensation.

Another measure of steering effort computed for each experimental run was the standard deviation of steering wheel angle. The data for steering input standard deviation were input to MINITAB for a two-way analysis of variance after diagnostic checks indicated that the analysis was reasonable. The ANOVA printout for steering angle standard deviation is shown in Figure 31.

The F-test for driver (block) effects indicates that the blocked design was appropriate in this case since driver effects are statistically significant. Analysis of compensator effects does not yield such a definite conclusion. Once again, though there was a definite trend toward reduced steering effort (in terms of standard deviation of steering angle) with compensation, we cannot reject the null hypothesis that the treatment means are equal with 95%, 90%, or even 80% confidence. The greatest observed difference between any two treatment means corresponded to an approximate 13 percent reduction in steering deviations from treatment 1 (no

compensation) to treatment 3 (McFarland's compensator). Even in this case, however, the respective confidence intervals exhibit considerable overlap. Thus the experimental data allow us to draw no firm inferences regarding compensator effects on steering effort.

ANALYSIS OF VARIANCE, steering input st. dev.

SOURCE	DF	SS	MS	F	p
drivers	6	0.016967	0.002828	6.639	0.001
comps	3	0.002082	0.000694	1.629	0.218
ERROR	18	0.007674	0.000426		
TOTAL	27	0.026722			

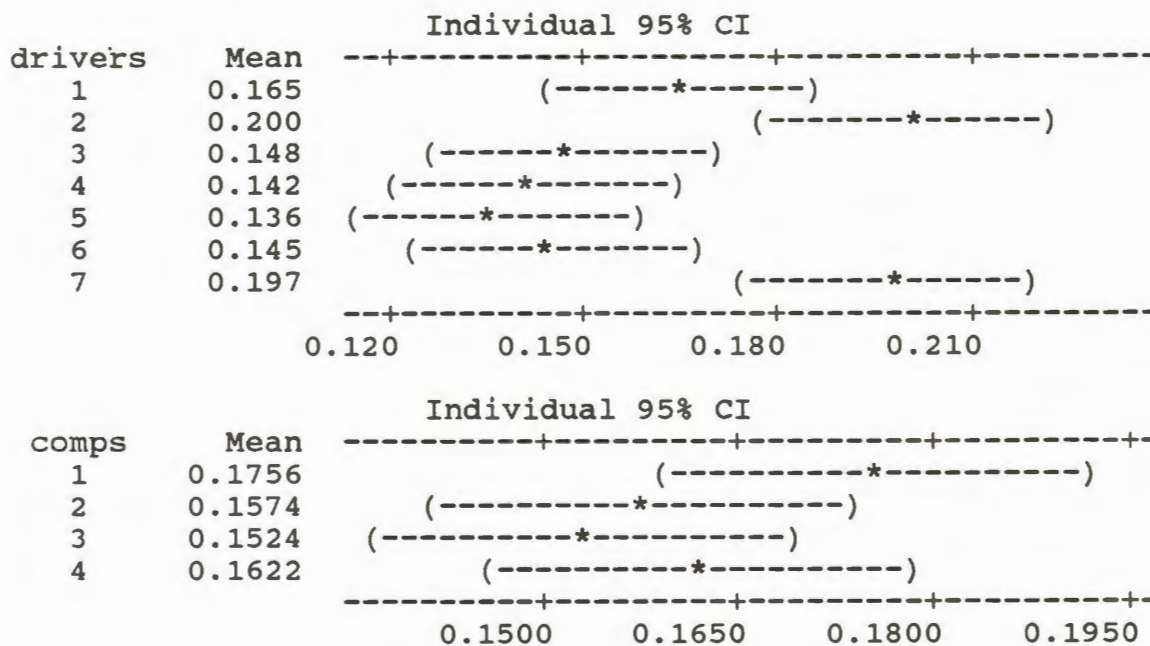


Figure 31. ANOVA Table and Confidence Intervals for Steering Input Deviations

The data for lateral acceleration standard deviation were input to MINITAB. Diagnostic checks revealed no major departures from model assumptions and a two-way analysis of variance was performed. The ANOVA table is shown in Figure 32.

ANALYSIS OF VARIANCE, lateral acceleration st. dev.

SOURCE	DF	SS	MS	F	p
drivers	6	1.6190	0.2698	2.939	0.035
comps	3	0.0829	0.0276	0.301	0.825
ERROR	18	1.6516	0.0918		
TOTAL	27	3.3535			

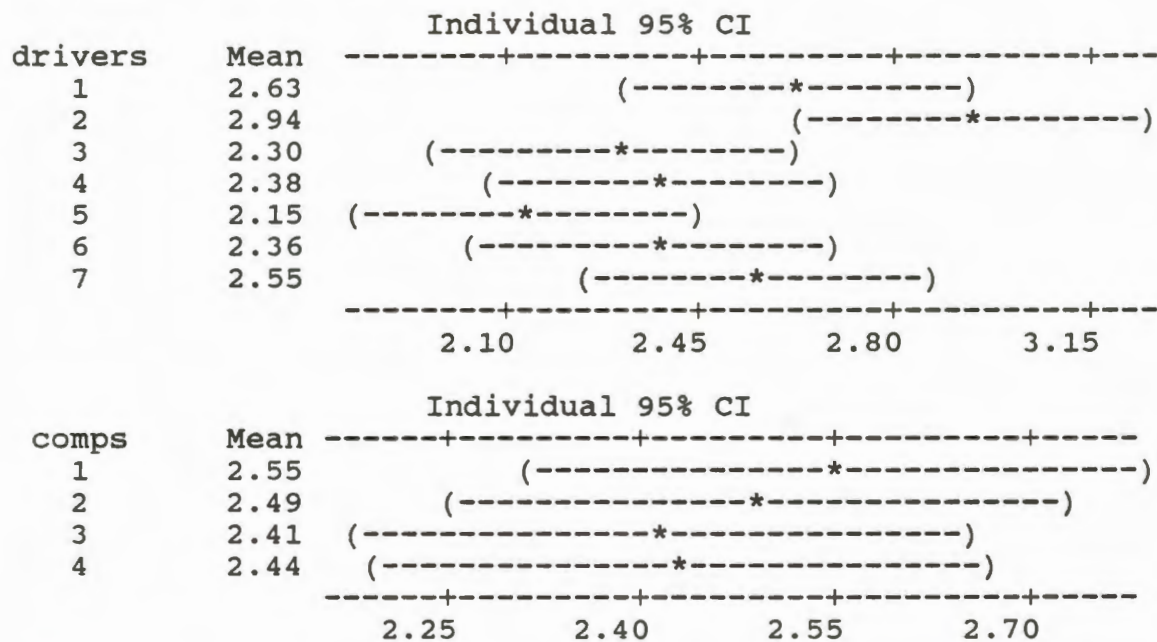


Figure 32. ANOVA Table and Confidence Intervals for Lateral Acceleration

Examination of the ANOVA table for possible compensator effects reveals a low F value (0.301) for treatments. The F-test does not allow us to reject with any degree of statistical confidence the null hypothesis that all treatment means are equal. Once again, there does seem to be a trend toward smaller lateral acceleration deviations with compensation: the mean for treatment 1, no compensation, is higher than that for any of treatments 2 through 4 (the delay-compensated configurations). However, the large amount of variability in the data (which appears in Figure 32 as

an overlap of the individual confidence intervals for the treatment means) does not allow us to draw the conclusion that delay compensation reduces lateral acceleration deviations in the simulated vehicle. There is approximately an 82% probability that differences of the magnitude observed in the experiment could be due solely to chance.

Finally, the data for standard deviation of longitudinal velocity (vehicle speed) were analyzed. Diagnostic checks revealed no serious departures from model assumptions and a two-way analysis of variance was performed. The ANOVA table and individual confidence intervals generated by MINITAB are shown in Figure 33.

In the case of vehicle speed deviations, as for steering wheel reversals, there was a trend toward better performance with the two velocity-based compensators as opposed to lead/lag or no compensation. In this case, however, the effect was more pronounced. The observed treatment means for no compensation and lead/lag compensation were virtually identical. The mean effect of the four-term compensator was to reduce the standard deviation of vehicle speed by about 20%, while McFarland's compensator was observed to reduce speed deviations by approximately 26% when compared to no compensation.

Though we cannot reject the null hypothesis that the treatment means are equal with 95% confidence, there is nearly a 90 percent probability that the observed variation is not due to chance. In fact, if confidence intervals are constructed for individual differences in means according to the procedure outlined in Appendix 6C

ANALYSIS OF VARIANCE, speed st. dev.

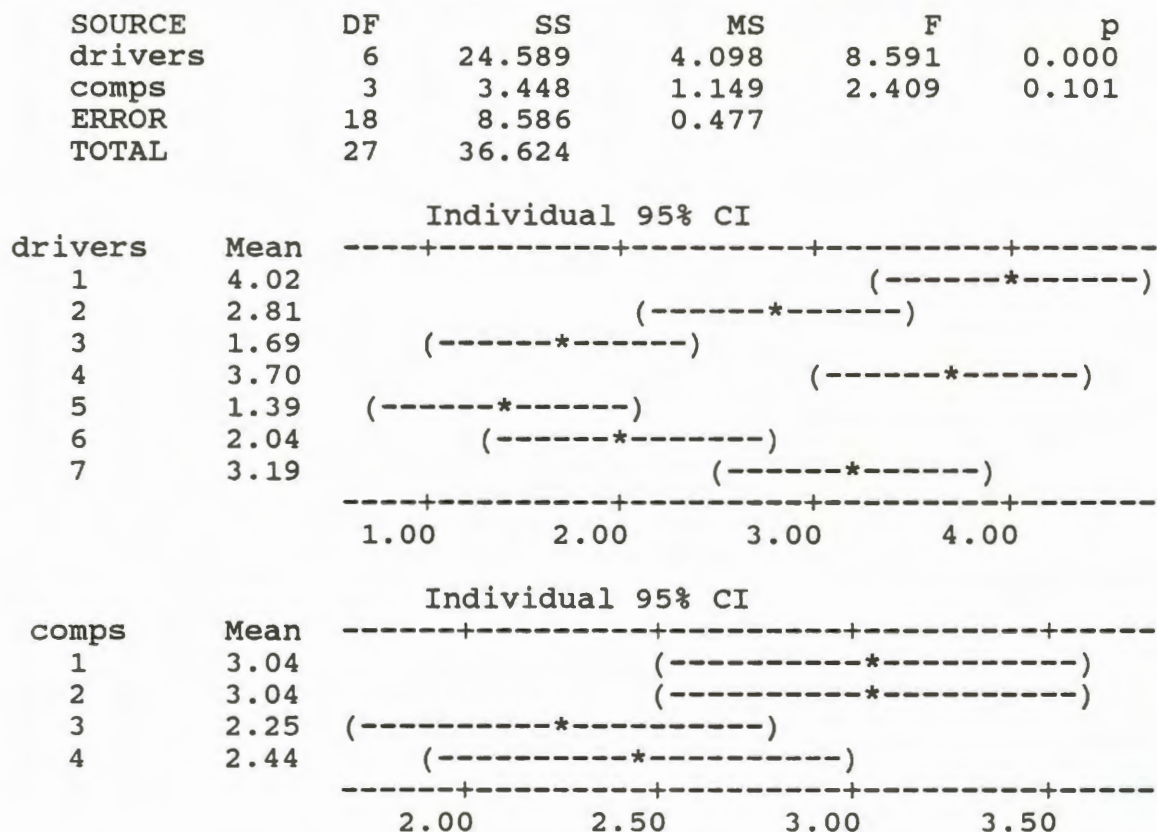


Figure 33. ANOVA Table and Confidence Intervals for Vehicle Speed Deviations

of reference [53], we have at least 90% confidence that the treatment means for McFarland's compensator and no compensation are different and nearly 90% confidence that the means for the four-term compensator and no compensation differ. Thus, while the analysis of vehicle speed deviation data does not offer overwhelming evidence of beneficial compensator effects, the possible indication of such effects is stronger in this case than for any of the other performance indices.

CHAPTER 6

SUMMARY AND CONCLUSIONS

The research presented in the preceding chapters focused on improving the handling qualities of an interactive driving simulator. This was to be done by reducing the effects of system transport delays, which were believed to be adversely affecting driver/vehicle control performance. Investigations using both time- and frequency-domain based measurement techniques revealed that substantial (and somewhat variable) transport delay was present in the system. Algorithmic compensators (digital filters) were designed to provide phase lead sufficient to counteract system delays at frequencies important to driver control. Three of these compensators were implemented in the simulator and found (by reapplication of the frequency-domain or steady-state delay measurement technique) to operate approximately as designed. Finally, a driver-in-the-loop experiment was conducted to assess the effect of delay compensation on driver/vehicle performance. While the small size of the experiment allowed no definite conclusions to be drawn regarding the efficacy of compensation, trends in the data were generally indicative of better performance with compensation.

A significant feature of the transport delay measurement performed as part of this research was the application of both time-domain and steady-state techniques

to the same system. While some of the references cited expressed a preference for one delay measurement approach over the other, none presented a comparison of the two techniques based on actual application of both to the same system. In this case, limitations of available instrumentation (namely the accuracy and resolution of phase measurements made with the available spectrum analyzer) inhibited the comparison to some degree. However, the results of the time-domain tests (which in this case gave more precise results) did in general fall within the wider range of delay times calculated from the steady-state phase measurements. It would be interesting to repeat the comparison using a more accurate instrument to measure phase. In the meantime, the two methods have been shown to be in approximate agreement, at least as applied to the UCF Driving Simulator environment.

With regard to the subject of applying delay compensation techniques to the UCF simulator, it is worth noting that certain modifications had to be made to both the time-domain and steady-state delay measurement techniques described by other researchers in order to adapt them to the system hardware and software. For example, a video sampling circuit had to be devised in order to couple the output of the image generator to the spectrum analyzer used to measure phase in steady-state. Of more significance is the fact that both approaches made use of system "inputs" computed in software, and output through digital or analog channels to instruments, rather than actual externally-generated inputs. This innovation simplified the delay testing processes, particularly the steady-state measurements, considerably. There is little or no effect on the accuracy of the measurement process

as long as the operations involved in generating the "inputs" are analyzed for timing effects and these effects are taken into account when determining effective transport delay times from the measurement data. The values finally reported for transport delay should be, and in this case were, the result of a theoretical analysis of system delays coupled with physical measurements of delay.

The design of a filter to compensate for transport delay in a system can be relatively simple or quite arduous and complex depending on the frequency range of interest and the quality of compensation (as compared to an ideal pure time advance) desired. The lead/lag filter, hardly a novel concept, provides a good "match" only at one frequency. However, if the system crossover frequency is known (or can be estimated reasonably closely) and if transport delay effects much above that frequency are unimportant, this approach may be cost-effective since it is quite easy to implement.

The velocity-based prediction methods first developed by McFarland and extended as part of this work are also digital filters, but of a higher order (and thus more complex design) than the simple lead/lag filter. By incorporating knowledge of the simulator's motion over several (3, 4, or 5) frames and using precisely-tuned filter coefficients, these algorithms are able to produce phase lead which is very close to ideal over a wide range of frequencies, typically from DC to several Hertz. This is of particular importance where a wide range of input frequencies are important and/or the system crossover frequency is variable or not known precisely — all likely scenarios for a training simulator.

The price that must be paid for the improved cancellation of delay effects using velocity-based compensation is amplitude distortion at frequencies above the compensation band. McFarland documented this effect as exhibited by his compensator using three velocity terms; this research further revealed that as the in-band phase match was improved (by using four or five velocity terms rather than three), the out-of-band amplitude distortion worsened. The five-term compensator could not be used in the driving simulator because of the visual "jitter" induced by the extreme amplification of high-frequency components in the simulation. The author's four-term compensator represented the highest-order filter that could be used in the driving simulator application without objectionable effects. In some situations (for example helicopter flight simulation) where significant high-frequency content is present in the system, even the three-velocity-term compensator may be unusable. In these cases lead/lag compensation could be tried or other lower-order filtering schemes could be investigated.

Three compensation algorithms (lead/lag, McFarland's three-velocity-term predictive filter, and the author's predictor using four velocity terms) gave the impression of handling improvement in informal trials and were implemented in the simulator. The operation of each was verified at three separate frequencies by reapplication of the steady-state delay measurement technique. The reductions observed in system phase lag were very close to the phase leads theoretically produced by each compensator at each test frequency. This result further illustrates the value of the steady-state technique: in addition to helping quantify system delay

for the purpose of designing delay compensation filters, it can be used to debug and/or verify the correct operation of the filters after they are implemented.

With the correct operation of all three compensators verified, an experiment was devised using human subjects to drive the simulator in an attempt to establish whether any differences in performance could be attributed to compensation. Unfortunately, the already modest number of experimental runs in the design was further reduced by unforeseen equipment failure. Thus it is not surprising that the trends exhibited by the chosen performance indices, while almost universally in favor of better performance with compensation than without, were not strong enough to establish statistically significant benefits attributable to compensation. Further, while for each of the five performance indices computed one or both of the two velocity-based compensators were the best performers (and in general the lead/lag compensator appeared to be third best), there is little that can be concluded from this experiment about the relative merits of the three compensators if one assumes that compensation is beneficial.

Further research into the performance effects of delay compensators would definitely be worthwhile. An obvious possibility would be to repeat the experiment described in the second section of Chapter 5 with at least the originally intended number of subjects. Considering the variability in the performance data actually obtained, it might be necessary to conduct even more trials than were originally planned to establish significant compensator effects, if any. Alternatively, or perhaps additionally, one could compute different performance measures or alter the driving

task (which was quite simple) in an attempt to highlight differences between the compensators. Many more examples of performance measures and driving tasks are described in the literature than could be included in this investigation. For example, steering wheel reversals and steering angle standard deviation tell only part of the story with regard to driver control effort. It might also be illuminating to perform a spectral (Fourier) analysis of the logged steering inputs to see if any differences exist due to compensation or the lack thereof. The experimental data could also be compared to instrumented car data, if available, to determine which compensator makes the vehicle drive most "realistically". The results of such a comparison could be used to choose the compensator which most improves the validity of the simulation.

Other potential experiments might examine the effect of delay compensation algorithms in areas besides driver/vehicle performance and simulator validation. In particular, since the UCF Driving Simulator was intended as a training simulator, it might be worthwhile to investigate the effects of delay compensation on training effectiveness. Naive subjects such as 15-year-old beginning drivers could be trained in the simulator using various compensators (or none at all) and the transfer of training to driving an actual automobile evaluated for each case. With a well-designed experiment it might be possible to prove or disprove the intuitive notion that "anything that makes the simulator drive more like a real car (for example, reducing transport delay) should improve its ability to train people to drive a real car."

Results reported in the literature [9] [11] [32] suggest that transport delay can affect some axes of control more than others, at least for flight simulators. If this is also true for ground vehicle simulation, it may be necessary (or worthwhile) to apply compensation only to position, or only to orientation (yaw angle) rather than to the entire set of vehicle coordinates. Though time did not permit this hypothesis to be investigated as part of the research effort, it might prove instructive for future investigators to try to determine whether driver performance and/or training are affected by applying a given form of delay compensation to just part of the information sent to the image generator.

Finally, although one of the major adverse effects of transport delay is believed to be an increased tendency to produce simulator sickness, it does not follow that reducing or eliminating transport delay will prevent subjects from experiencing distress. In fact, during the driver-in-the-loop experiment two subjects had to be rejected because of the onset of simulator sickness during practice runs using delay compensation. Two others who were able to complete all four runs complained afterward of some symptoms. It is not known whether these examples of simulator sickness were due to lack of a motion platform (perceptual mismatch between apparent visual motion and lack of physical motion cues) or some other cause. But it is apparent that delay compensation, while potentially very useful in improving the realism of a given simulator, is not a "cure for everything that ails it."

APPENDICES

APPENDIX A

REAL-TIME C PROGRAM CODE FOR VELOCITY-BASED COMPENSATOR

```

/*****
*      Name: MODEL.C
*      Purpose: This contains the functions needed to perform the
*               vehicle dynamics mathematics
*      Author: Mike Garnsey, Joe Dumas, Chris Nichols
*      Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration)
*                  10/16/92 (added compensator with 3, 4, or 5
*                          velocity terms to correct for delay)
*                  11/19/92 (modified for driver experiments)
*****/

/*****
* Below are the variables and descriptions for values in the HYSIM
* math model
*****/

float FRMTIM;      /* Frame time (sec) */
float HDGCAR;     /* Vehicle heading (rad) */
float STEERW;     /* Steering wheel angle (rad) */
float UCAR;       /* Vehicle forward velocity (fps) */
float VCAR;       /* Lateral velocity (fps) */
float XCAR;       /* X-position, map (inertial) coordinates (ft) */
float XDOT;       /* X velocity component, map coords (ft) */
float YCAR;       /* Y-position, map (inertial) coordinates (ft) */
float YDOT;       /* Y velocity component, map coords (ft) */
float YAWRAT;     /* Vehicle yaw rate (rad/sec) */

float XDOT_old,YDOT_old; /* Saved values of velocities at time */
float YAWRAT_old;      /* (n), needed for Trapezoidal int. of */
                        /* velocity to position */
                        /* (also used in delay compensation) */

/* ----- Variables added to implement delay compensation ----- */

float XDOT_old2;      /* Saved values of car inertial velocity, */
float XDOT_old3;      /* needed to implement delay compensator */
float XDOT_old4;

float YDOT_old2;      /* Saved values of car inertial velocity, */
float YDOT_old3;      /* needed to implement delay compensator */
float YDOT_old4;

float YAWRAT_old2;    /* Saved values of yaw rate (rotational */
float YAWRAT_old3;    /* velocity) needed to implement delay */
float YAWRAT_old4;    /* compensator */

float b0, b1, b2, b3, b4; /* Coefficients for digital filter */

float XCOMP;          /* Current values of vehicle position */
float YCOMP;          /* compensated by prediction */
float HDGCOMP;

float XOUT;           /* Current values of vehicle position */
float YOUT;           /* sent to the IG */
float HDGOUT;

```

```

int delflag;          /* indicates which delay value to allow */
                    /* for (which CIG database is being run) */
int vel_terms;      /* number of terms (3, 4, or 5) to use */
                    /* in delay comp. difference equation */

/*****
/*----- Vehicle Dynamics Module -----*/
/*****
void veh_dyn()
{
    /* [Vehicle model details not shown] */

/*----- Trapezoidal Integration of Vehicle Heading -----*/
    HDGCAR = HDGCAR + (0.5 * FRMTIM) * (YAWRAT + YAWRAT_old);
    /* Now we have heading(n+1) (computed to end of current frame) */
/* ----- Keep heading limited for trig. functions -----*/
    if ( HDGCAR > M_PI ) HDGCAR = HDGCAR - PIx2;
    if ( HDGCAR < -M_PI ) HDGCAR = HDGCAR + PIx2;

    SINHDG = sin(HDGCAR);
    COSHDG = cos(HDGCAR);

/* ----- Transform velocity from body to inertial coordinates -----*/
    XDOT = (UCAR * COSHDG) - (VCAR * SINHDG);
    YDOT = (UCAR * SINHDG) + (VCAR * COSHDG);

/* Now we have XDOT and YDOT(n+1) (computed from UDOT and VDOT(n+1)) */
/*----- Trapezoidal Integrations (Velocity to Position) -----*/
    XCAR = XCAR + (0.5 * FRMTIM) * (XDOT + XDOT_old);
    YCAR = YCAR + (0.5 * FRMTIM) * (YDOT + YDOT_old);

    /* Now we have position(n+1) (computed to end of current frame) */
}
/* ----- End of veh_dyn() ----- */

```

```

/*****
*      Name: IO.C
*      Purpose: The purpose of these functions is to perform all I/O *
*               functions including reading analog and digital inputs*
*      Author: Joe Dumas, Mike Garnsey, Chris Nichols
*      Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration) *
*                  10/16/92 (added compensator with 3, 4, or 5
*                  velocity terms to correct for delay)
*                  11/19/92 (modified for driver experiments)
*****/

/* External variables */

extern int delflag;
extern int vel_terms;
extern float b0, b1, b2, b3, b4;

/* ----- */
/* ----- */

void io_init()
{

    printf("\n\nUsing 3-D database with cars (enter 1) or without cars
(0)?");
    scanf("%d",&delflag);

    printf("\n\nNumber of velocity terms to use in compensation (3, 4,
5): ");
    scanf("%d",&vel_terms);

    /* Set parameters for delay compensator depending on database used */
    /* and number of terms to be used in delay compensator (3, 4, or 5) */

    if(delflag == 1)          /* Using the 3-D database with cars */
    {
        if(vel_terms == 5)  /* Using 5 velocity terms */
        {
            b0 = 34.0533958760706;
            b1 = -120.2015904563864;
            b2 = 163.6398787354032;
            b3 = -101.2926101993455;
            b4 = 23.9749260442580;
        }
        if(vel_terms == 4)  /* Using 4 velocity terms */
        {
            b0 = 10.078469831629158;
            b1 = -26.675355435196618;
            b2 = 24.534653431488909;
            b3 = -7.766375178504606;
            b4 = 0.0;
        }
    }
}

```

```

if(vel_terms == 3) /* Using 3 velocity terms (McFarland) */
{
    b0 = 2.338730942226848;
    b1 = -4.187122050116626;
    b2 = 2.022391102882988;
    b3 = 0.0;
    b4 = 0.0;
}
}
else /* Using the 3-D database without cars */
{
    if(vel_terms == 5) /* Using 5 velocity terms */
    {
        b0 = 13.320659045063927;
        b1 = -45.209450962572625;
        b2 = 59.760578173270460;
        b3 = -36.116131490478700;
        b4 = 8.377345234716937;
    }
    if(vel_terms == 4) /* Using 4 velocity terms */
    {
        b0 = 4.943313810426728;
        b1 = -12.529410247560927;
        b2 = 11.154276149118376;
        b3 = -3.436090775315308;
        b4 = 0.0;
    }
    if(vel_terms == 3) /* Using 3 velocity terms (McFarland) */
    {
        b0 = 1.516530348045595;
        b1 = -2.575189990269217;
        b2 = 1.191659643534925;
        b3 = 0.0;
        b4 = 0.0;
    }
}
}
}

```

```

/*****
* Name: MAIN.C *
* Purpose: This is the main calling program for the DTS program *
* All initial conditions are set, realtime exec is *
* started, and main loop is run *
* Authors: Mike Garnsey, Joe Dumas, Chris Nichols *
* Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration) *
* 10/16/92 (added compensator with 3, 4, or 5 *
* velocity terms to correct for delay) *
* 11/19/92 (modified for driver experiments) *
*****/

```

```

/* ---- externally declared veh. dyn. model variables used here ---- */
extern float YAWRAT;
extern float XCAR, XDOT, YCAR, YDOT, HDGCAR, FRMTIM;
extern float UCAR, VCAR, STEERW;

```

```

extern float XDOT_old, YDOT_old, YAWRAT_old;
extern float XDOT_old2, YDOT_old2, YAWRAT_old2;
extern float XDOT_old3, YDOT_old3, YAWRAT_old3;
extern float XDOT_old4, YDOT_old4, YAWRAT_old4;

extern float XOUT;
extern float YOUT;
extern float HDGOUT;
extern float XCOMP;
extern float YCOMP;
extern float HDGCOMP;
extern float b0, b1, b2, b3, b4;

/* ----- Variables for communication to visual display computer ----- */

int cgi_yaw;
float xcgi;
float ycgi;

void main()
{

    /*----- MATH MODEL UPDATE RATE -----*/
    FRMTIM = 1./60.;          /* Set integration step size to 60 Hz */
    io_init();

    /*----- INITIAL STATE CONDITIONS -----*/
    YCAR      = 52.5;
    YDOT      = 0.;
    XCAR      = 100.0;
    XDOT      = 0.;
    HDGCAR    = 0.;
    UCAR      = 0.;
    VCAR      = 0.;
    YAWRAT    = 0.;
    STEERW    = 0.;

    XDOT_old = 0.0; /* Initialize AB-2 integrator outputs (inputs to */
    YDOT_old = 0.0; /* trapezoidal integrators for velocity to pos.) */
    YAWRAT_old = 0.0; /* These values are also used in delay comp. */

    /* Initialize rest of terms to be used in delay compensator */

    XDOT_old2 = 0.0;
    YDOT_old2 = 0.0;
    YAWRAT_old2 = 0.0;

    XDOT_old3 = 0.0;
    YDOT_old3 = 0.0;
    YAWRAT_old3 = 0.0;

    XDOT_old4 = 0.0;
    YDOT_old4 = 0.0;
    YAWRAT_old4 = 0.0;

```

```

/* -- Main execution loop for simulation up and running -- */
do
{
/* -- log data every sixth frame (10 Hz when running at 60 Hz) -- */
  if((file_full == FALSE) && (log_count == 0)) log_data();

  do_in();
  veh_dyn();

/* delay compensator difference equation filters data before sending */

/* Compensate X position */

  XCOMP=XCAR + b0*XDOT + b1*XDOT_old + b2*XDOT_old2 + b3*XDOT_old3 +
b4*XDOT_old4;

  XDOT_old4 = XDOT_old3;          /* Bump all values to one step older */
  XDOT_old3 = XDOT_old2;          /* for next time */
  XDOT_old2 = XDOT_old;
  XDOT_old = XDOT;

/* Compensate Y position */

  YCOMP=YCAR + b0*YDOT + b1*YDOT_old + b2*YDOT_old2 + b3*YDOT_old3 +
b4*YDOT_old4;

  YDOT_old4 = YDOT_old3;          /* Bump all values to one step older */
  YDOT_old3 = YDOT_old2;          /* for next time */
  YDOT_old2 = YDOT_old;
  YDOT_old = YDOT;

/* Compensate heading */

  HDGCOMP = HDGCAR + b0*YAWRAT + b1*YAWRAT_old + b2*YAWRAT_old2 +
b3*YAWRAT_old3 + b4*YAWRAT_old4;

  YAWRAT_old4 = YAWRAT_old3;      /* Bump all values to one step older */
  YAWRAT_old3 = YAWRAT_old2;      /* for next time */
  YAWRAT_old2 = YAWRAT_old;
  YAWRAT_old = YAWRAT;

/* -- set coordinates to send out to IG -- */

if(UCAR > 20.0) /* Send comp. values to IG (normal operation) */
{
  XOUT = XCOMP;
  YOUT = YCOMP;
  HDGOUT = HDGCOMP;
}
else if(UCAR > 12.0) /* Use weighted combination of actual/comp. */
{
  XOUT = XCAR + ((UCAR - 12.0) / 8.0) * (XCOMP - XCAR));
  YOUT = YCAR + ((UCAR - 12.0) / 8.0) * (YCOMP - YCAR));
  HDGOUT = HDGCAR+((UCAR-12.0) / 8.0) * (HDGCOMP - HDGCAR));
}
}

```

```
else          /* Use uncompensated values at very low speeds */
{
    XOUT = XCAR;
    YOUT = YCAR;
    HDGOUT = HDGCAR;
}

/* convert veh. model coord. sys. to XTAR CGI coord. sys. and send */

    xcgi = -YOUT;
    ycgi = -XOUT;
    cgi_yaw = -HDGOUT * 651.89865;

    out_cgi(&xcgi,&ycgi,&cgi_yaw);

    do_out();

    sim_time = sim_time + FRMTIM;
    log_count++;
    if (log_count == 6) log_count = 0;

} while (kbhit() == 0);

safe_exit();
}
/* ===== end of main() ===== */
```


APPENDIX B

REAL-TIME C PROGRAM CODE FOR LEAD/LAG COMPENSATOR

```

/*****
*      Name: MODEL.C
*      Purpose: This contains the functions needed to perform the
*               vehicle dynamics mathematics
*      Author: Mike Garnsey, Joe Dumas, Chris Nichols
*      Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration)
*                  07/31/92 (added lead/lag digital filter for delay
*                          compensation testing)
*                  11/19/92 (modified for driver experiments)
*****/

/*****
* Below are the variables and descriptions for values in the HYSIM
* math model
*****/

float FRMTIM;      /* Frame time (sec) */
float HDGCAR;     /* Vehicle heading (rad) */
float STEERW;     /* Steering wheel angle (rad) */
float UCAR;      /* Vehicle forward velocity (fps) */
float VCAR;      /* Lateral velocity (fps) */
float XCAR;      /* X-position, map (inertial) coordinates (ft) */
float XDOT;     /* X velocity component, map coords (ft) */
float YCAR;      /* Y-position, map (inertial) coordinates (ft) */
float YDOT;     /* Y velocity component, map coords (ft) */
float YAWRAT;   /* Vehicle yaw rate (rad/sec) */

float XDOT_old, YDOT_old; /* Saved values of velocities at time */
float YAWRAT_old;        /* (n), needed for Trapezoidal int. of */
                        /* velocity to position */

/* ----- Variables added to implement delay compensation ----- */

float XCAR_old; /* Saved value of car inertial position, */
               /* needed to implement delay compensator */
float YCAR_old; /* Saved value of car inertial position, */
               /* needed to implement delay compensator */
float HDGCAR_old; /* Saved value of car heading angle, */
                 /* needed to implement delay compensator */

float kd, a0, a1, b1; /* Coefficients for digital filter */

float XOUT, XOUT_old; /* Current and past values of vehicle */
float YOUT, YOUT_old; /* position (compensated) for CIG use */
float HDGOUT, HDGOUT_old;

int delflag; /* indicates which delay value to allow */
             /* for (which CIG database is being run) */

/*****
/----- Vehicle Dynamics Module -----*/
/*****
void veh_dyn()
{
    /* [Vehicle model details not shown] */

```

```

/*----- Trapezoidal Integration of Vehicle Heading -----*/
HDGCAR = HDGCAR + (0.5 * FRMTIM) * (YAWRAT + YAWRAT_old);
/* Now we have heading(n+1) (computed to end of current frame) */
YAWRAT_old = YAWRAT;          /* Save YAWRAT for next frame */
                               /* (will be YAWRAT(n)) */

/* ----- Keep heading limited for trig. functions ----- */
if ( HDGCAR > M_PI )
{
  HDGCAR = HDGCAR - PIx2;
  HDGCAR_old = HDGCAR_old - PIx2;
  HDGOUT = HDGOUT - PIx2;
  HDGOUT_old = HDGOUT_old - PIx2;
}
if ( HDGCAR < -M_PI )
{
  HDGCAR = HDGCAR + PIx2;
  HDGCAR_old = HDGCAR_old + PIx2;
  HDGOUT = HDGOUT + PIx2;
  HDGOUT_old = HDGOUT_old + PIx2;
}

SINHHDG = sin(HDGCAR);
COSHHDG = cos(HDGCAR);

/* ----- Transform velocity from body to inertial coordinates ----- */
XDOT = (UCAR * COSHHDG) - (VCAR * SINHHDG);
YDOT = (UCAR * SINHHDG) + (VCAR * COSHHDG);

/* Now we have XDOT and YDOT(n+1) (computed from UDOT and VDOT(n+1)) */
/* ----- Trapezoidal Integrations (Velocity to Position) ----- */
XCAR = XCAR + (0.5 * FRMTIM) * (XDOT + XDOT_old);
YCAR = YCAR + (0.5 * FRMTIM) * (YDOT + YDOT_old);

/* Now we have position(n+1) (computed to end of current frame) */
XDOT_old = XDOT;    /* Save XDOT for next frame (will be XDOT(n)) */
YDOT_old = YDOT;    /* Save YDOT for next frame (will be YDOT(n)) */
}
/* ----- End of veh_dyn() ----- */

```

```

/*****
*      Name: IO.C
*      Purpose: The purpose of these functions is to perform all I/O *
*              functions including reading analog and digital inputs*
*      Author: Joe Dumas, Mike Garnsey, Chris Nichols
*      Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration)
*                  07/31/92 (added lead/lag digital filter for delay
*                          compensation testing)
*                  11/19/92 (modified for driver experiments)
*****/

/* External variables */

extern int delflag;
extern float kd, a0, a1, b1;

/* ----- */
/* ----- */

void io_init()
{

    printf("\n\nUsing 3-D database with cars (enter 1) or without cars
(0)?");
    scanf("%d",&delflag);

/* Set parameters for delay compensator depending on database used */
    if(delflag == 1)          /* Using the 3-D database with cars */
    {
        kd = 0.7805944;
        a0 = 2.0993686;
        a1 = -2.0305368;
        b1 = -0.9311682;
    }
    else                      /* Using the 3-D database without cars */
    {
        kd = 0.8145284;
        a0 = 1.7277625;
        a1 = -1.6711146;
        b1 = -0.9433520;
    }
}

```

```

/*****
*      Name: MAIN.C
*      Purpose: This is the main calling program for the DTS program
*              All initial conditions are set, realtime execution is
*              started, and main loop is run
*      Authors: Mike Garnsey, Joe Dumas, Chris Nichols
*      Last Update: 07/21/92 (changed to AB-2/Trapezoidal integration)
*                  07/31/92 (added lead/lag digital filter for delay
*                  compensation testing)
*                  11/19/92 (modified for driver experiments)
*****/

/* ---- externally declared veh. dyn. model variables used here ---- */

extern float YAWRAT;
extern float XCAR, XDOT, YCAR, YDOT, HDGCAR, FRMTIM;
extern float UCAR, VCAR, STEERW;

extern float XDOT_old, YDOT_old, YAWRAT_old;

extern float XCAR_old, XOUT, XOUT_old;
extern float YCAR_old, YOUT, YOUT_old;
extern float HDGCAR_old, HDGOUT, HDGOUT_old;
extern float kd, a0, a1, b1;

/* ----- Variables for communication to visual display computer ----- */

int cgi_yaw;
float xcgi;
float ycgi;

void main()
{

    /*----- MATH MODEL UPDATE RATE -----*/

    FRMTIM = 1./60.;          /* Set integration step size to 60 Hz */

    io_init();

    /*----- INITIAL STATE CONDITIONS -----*/

    YCAR      = 52.5;
    YDOT      = 0.;
    XCAR      = 100.0;
    XDOT      = 0.;
    HDGCAR    = 0.;
    UCAR      = 0.;
    VCAR      = 0.;
    YAWRAT    = 0.;
    STEERW    = 0.;

```

```

XDOT_old = 0.0; /* Initialize AB-2 integrator outputs (inputs to */
YDOT_old = 0.0; /* trapezoidal integrators for velocity to pos.) */
YAWRAT_old = 0.0;

/* Initialize terms to be used in delay compensator */

XCAR_old = 100.0;
XOUT      = 100.0;
XOUT_old  = 100.0;

YCAR_old = 52.5;
YOUT      = 52.5;
YOUT_old  = 52.5;

HDGCAR_old = 0.0;
HDGOUT      = 0.0;
HDGOUT_old  = 0.0;

/* -- Main execution loop for simulation up and running -- */
do
{
/* -- log data every sixth frame (10 Hz when running at 60 Hz) -- */
  if((file_full == FALSE) && (log_count == 0)) log_data();

  do_in();
  veh_dyn();
/* delay compensator difference equation filters data before sending */
  XOUT = a0 * XCAR + a1 * XCAR_old - b1 * XOUT_old;

  XCAR_old = XCAR;
           /* Save XCAR value this frame as old value for next */
  XOUT_old = XOUT;
           /* Save XOUT value this frame as old value for next */

  YOUT = a0 * YCAR + a1 * YCAR_old - b1 * YOUT_old;

  YCAR_old = YCAR;
           /* Save YCAR value this frame as old value for next */
  YOUT_old = YOUT;
           /* Save YOUT value this frame as old value for next */

  HDGOUT = a0 * HDGCAR + a1 * HDGCAR_old - b1 * HDGOUT_old;

  HDGCAR_old = HDGCAR;
             /* Save HDGCAR to use as old value next frame */
  HDGOUT_old = HDGOUT;
             /* Save HDGOUT to use as old value next frame */

/* convert veh. model coord. sys. to XTAR CGI coord. sys. and send */

  xcgi = -YOUT;
  ycgi = -XOUT;
  cgi_yaw = -HDGOUT * 651.89865;

  out_cgi(&xcgi,&ycgi,&cgi_yaw);

```

```
do_out();

sim_time = sim_time + FRMTIM;
log_count++;
if (log_count == 6) log_count = 0;

} while (kbhit() == 0);

safe_exit();
}
/* ===== end of main() ===== */
```

APPENDIX C

C PROGRAM TO COMPUTE CONSTRAINT EQUATION COEFFICIENTS


```

/*****
*      Name: 4TERMEQS.C
*      Purpose: This program calculates the terms on the left and
*              right sides of the four equations that must be
*              solved simultaneously to obtain the filter
*              coefficients b0-b3.
*      Author: Joe Dumas
*      Last Update: 10/16/92
*****/

#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <process.h>
#include <time.h>

/* -- Global variables -- */

float P, w, w0, w1;
/* Transport delay, angular freq., two design freqs. */
int framerate;
/* Number of frames per second */
double T;
/* Frame time (seconds) */

double theta, psi;
/* Cyclic angle and projection angle */
double sintht, costht, sin2tht, cos2tht;
/* Sines and cosines of theta */
double sin3tht, cos3tht, sin4tht, cos4tht;
/* and its multiples */
double sinpsi, cospsi;
/* Sine and cosine of psi */

double k0, k1, ka, kb, kc, kd, kf, kg, kh, kj;
/* Intermediate constants */

double m0, m1, m2, m3;
/* Coefficients in left sides */
double n0, n1, n2, n3;
/* of constraint equations */
double rsm, rsn;
/* Right sides of constraint equations */

main()
{
    printf("\nProgram to calculate terms of equations that must be
solved");
    printf("\nto get delay filter coefficients, using 4 velocity
terms.");

    printf("\nEnter the first design frequency in rad/s: ");
    scanf("%f", &w0);

    printf("\nEnter the other design frequency in rad/s: ");
    scanf("%f", &w1);

    printf("\nEnter the delay to be compensated in seconds: ");
    scanf("%f", &P);

    printf("\nEnter the frame rate in Hz: ");
    scanf("%d", &framerate);

    T = 1.0 / ((double)(framerate));

    printf("\n");
}

```

```

/* -- Calculate first 8 numbers (coef. of 1st/2nd equations) -- */

w      = w0;
theta  = w0 * T;
psi    = w0 * P;

sintht = sin(theta);
sin2tht = sin(2.0 * theta);
sin3tht = sin(3.0 * theta);
sin4tht = sin(4.0 * theta);
sinpsi  = sin(psi);

costht  = cos(theta);
cos2tht = cos(2.0 * theta);
cos3tht = cos(3.0 * theta);
cos4tht = cos(4.0 * theta);
cospsi  = cos(psi);

k0 = sinpsi - (sinpsi * costht) + (cospsi * sintht);
k1 = cospsi - (cospsi * costht) - (sinpsi * sintht);

ka = (k0 * costht) + (k1 * sintht);
kb = (k0 * cos2tht) + (k1 * sin2tht);
kc = (k0 * cos3tht) + (k1 * sin3tht);
kd = (k0 * cos4tht) + (k1 * sin4tht);

kf = (k1 * costht) - (k0 * sintht);
kg = (k1 * cos2tht) - (k0 * sin2tht);
kh = (k1 * cos3tht) - (k0 * sin3tht);
kj = (k1 * cos4tht) - (k0 * sin4tht);

m0 = k0 - ka;
m1 = ka - kb;
m2 = kb - kc;
m3 = kc - kd;

n0 = k1 - kf;
n1 = kf - kg;
n2 = kg - kh;
n3 = kh - kj;

rsm = ((2.0 * (1.0 - costht)) / w) - (0.5 * T * (k0 + ka));
rsn = -(0.5 * T) * (k1 + kf);

printf("%23.16E %23.16E\n      %23.16E %23.16E\n\n",m0,m1,m2,m3);
printf("%23.16E %23.16E\n      %23.16E %23.16E\n\n",n0,n1,n2,n3);
printf("Right side coefs = %23.16E      %23.16E\n\n",rsm,rsn);

/* -- Calculate second 8 numbers (coef. of 3rd/4th equations) -- */

w      = w1;
theta  = w1 * T;
psi    = w1 * P;

sintht = sin(theta);
sin2tht = sin(2.0 * theta);
sin3tht = sin(3.0 * theta);
sin4tht = sin(4.0 * theta);
sinpsi  = sin(psi);

```

```

costht = cos(theta);
cos2tht = cos(2.0 * theta);
cos3tht = cos(3.0 * theta);
cos4tht = cos(4.0 * theta);
cospsi = cos(psi);

k0 = sinpsi - (sinpsi * costht) + (cospsi * sintht);
k1 = cospsi - (cospsi * costht) - (sinpsi * sintht);

ka = (k0 * costht) + (k1 * sintht);
kb = (k0 * cos2tht) + (k1 * sin2tht);
kc = (k0 * cos3tht) + (k1 * sin3tht);
kd = (k0 * cos4tht) + (k1 * sin4tht);

kf = (k1 * costht) - (k0 * sintht);
kg = (k1 * cos2tht) - (k0 * sin2tht);
kh = (k1 * cos3tht) - (k0 * sin3tht);
kj = (k1 * cos4tht) - (k0 * sin4tht);

m0 = k0 - ka;
m1 = ka - kb;
m2 = kb - kc;
m3 = kc - kd;

n0 = k1 - kf;
n1 = kf - kg;
n2 = kg - kh;
n3 = kh - kj;

rsm = ((2.0 * (1.0 - costht)) / w) - (0.5 * T * (k0 + ka));
rsn = -(0.5 * T) * (k1 + kf);

printf("%23.16E %23.16E\n          %23.16E %23.16E\n\n",m0,m1,m2,m3);
printf("%23.16E %23.16E\n          %23.16E %23.16E\n\n",n0,n1,n2,n3);
printf("Right side coefs = %23.16E          %23.16E\n",rsm,rsn);
}

```

REFERENCES

- [1] Casali, J. G., and Wierwille, W. W. (1986). "Potential Design Etiological Factors of Simulator Sickness and a Research Simulator Specification", Transportation Research Record 1059, pp. 66-74, Transportation Research Board, National Research Council.
- [2] Ricard, G. L., and Puig, J. A. (1977). "Delay of Visual Feedback in Aircraft Simulators", technical note NAVTRAEQUIPCEN TN-56, Naval Training Equipment Center, Orlando, Florida, March 1977.
- [3] Johnson, W. V., and Middendorf, M. S. (1988). "Simulator Transport Delay Measurement Using Steady-State Techniques", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-88-4619-CP).
- [4] Bailey, R. E., Knotts, L. H., Horowitz, S. J., and Malone, H. L. III (1987). "Effect of Time Delay on Manual Flight Control and Flying Qualities During In-Flight and Ground-Based Simulation", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2370-CP).
- [5] Allen, R. W. (1984). "Computational Considerations in Real Time Simulation Computer Graphics", Systems Technology, Inc. paper no. 341, presented at Computer Graphics '84, Anaheim, CA.
- [6] Crane, D. F. (1983). "Compensation For Time Delay in Flight Simulator Visual-Display Systems", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-83-1080-CP).
- [7] Deyo, R., Briggs, J. A., and Doenges, P. (1988). "Getting Graphics in Gear: Graphics and Dynamics in Driving Simulation", Computer Graphics, Vol. 22, No. 4, August 1988, pp. 317-326.
- [8] Frank, L. H., Casali, J. G., and Wierwille, W. W. (1988). "Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator", Human Factors, Vol. 30, No. 2, pp. 201-217.
- [9] Ricard, G. L., Norman, D. A., and Collyer, S. C. (1976). "Compensating for Flight Simulator CGI System Delays", Ninth NTEC/Industry Conference Proceedings (NAVTRAEQUIPCEN IH-276), pp. 131-139.

- [10] McRuer, D. (1980). "Human Dynamics in Man-Machine Systems", Automatica, Vol. 16, No. 3, May 1980, pp. 237-253.
- [11] Merriken, M. S., Riccio, G. E., and Johnson, W. V. (1987). "Temporal Fidelity in Aircraft Simulator Visual Systems", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2372-CP).
- [12] Drosdol, J., and Panik, F. (1985). "The Daimler-Benz Driving Simulator: a Tool for Vehicle Development", SAE Technical Paper Series (paper no. 850334), Society of Automotive Engineers, Inc.
- [13] Hahn, S., and Käding, W. (1988). "The Daimler-Benz Driving Simulator — Presentation of Selected Experiments", SAE Technical Paper Series (paper no. 880058), Society of Automotive Engineers, Inc.
- [14] Ricard, G. L., and Harris, W. T. (1980). "Lead/Lag Dynamics to Compensate for Display Delays", Journal of Aircraft, Vol. 17, No. 3, March 1980, pp. 212-217.
- [15] Hosman, R. J. A. W., and van der Vaart, J. C. (1988). "Visual-Vestibular Interaction in Pilot's Perception of Aircraft or Simulator Motion", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-88-4622-CP).
- [16] Levison, W. H., and Papazian, B. (1987). "The Effects of Time Delay and Simulator Mode on Closed-Loop Pilot/Vehicle Performance: Model Analysis and Manned Simulation Results", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2371-CP).
- [17] Merriken, M. S., Johnson, W. V., Cress, J. D., and Riccio, G. E. (1988). "Time Delay Compensation Using Supplementary Cues in Aircraft Simulator Systems", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-88-4626-CP).
- [18] Klee, H. I. (1990). "Progress Report on Development of a Low Cost Driving Simulator", Proceedings of the 1990 Society for Computer Simulation Western Multiconference (January, 1990, San Diego, CA), pp. 105-110.
- [19] Klee, H. I. (1991). "The University of Central Florida Interactive Driving Simulator", Proceedings of the 1991 Society for Computer Simulation Western Multiconference (January, 1991, Anaheim, CA).

- [20] Alexander, L. A. (1990). "The Development of the Real Time Software, Data Base, and Data Base Editing Tool for the Computer Image Generator Associated With a Driver Training Simulator", research report, University of Central Florida, May 1990.
- [21] Polasek, P. J. (1990). "Development of a Database and Database Compiler in Support of a Low Cost Driver Trainer Simulation System Design", research report, University of Central Florida, May 1990.
- [22] Howe, R. M. (1983). "Special Considerations in Real-Time Digital Simulation", Proceedings of the 1983 Summer Computer Simulation Conference, Society for Computer Simulation, pp. 66-71.
- [23] Panzitta, M. J. (1991). "Limitations of Asymptotic Root Error Analysis for Real-Time Integration Algorithms", technical report, Evans and Sutherland Computer Corporation.
- [24] Gum, D. R., and Albery, W. B. (1977). "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training", Journal of Aircraft, Vol. 14, No. 4, April 1977, pp. 327-332.
- [25] Levison, W. H. (1989). "Model-Based Guidelines for Simulator Temporal Fidelity Requirements", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-89-3271-CP).
- [26] Dumas, J. D. (1990). "Investigation of Driver-Vehicle System Behavior Using ACSL", course term report (unpublished), University of Central Florida, March 1990.
- [27] Cooper, F. R., Harris, W. T., and Sharkey, V. J. (1975). "Effects of Visual System Time Delay on Pilot Performance", Eighth NTEC/Industry Conference Proceedings (NAVTRAEQUIPCEN IH-250), pp. 35-51.
- [28] Woltkamp, J., Ramachandran, S., and Branson, R. (1988). "Determination of Helicopter Simulator Time Delay and Its Effects on Air Vehicle Development", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-88-4620-CP).
- [29] Riccio, G. E., Cress, J. D., and Johnson, W. V. (1987). "The Effects of Simulator Delays on the Acquisition of Flight Control Skills: Control of Heading and Altitude", Proceedings of the Human Factors Society: 31st Annual Meeting, Vol. 2, pp. 1286-1290, Human Factors Society, Santa Monica, CA.

- [30] Middendorf, M. S., Lusk, S. L., and Whiteley, J. D. (1990). "Power Spectral Analysis to Investigate the Effects of Simulator Time Delay on Flight Control Activity", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-90-3127-CP).
- [31] Allen, R. W., and DiMarco, R. J. (1985). "Effects of Transport Delays on Manual Control System Performance", technical report N85-14498, NASA Ames-Moffett Technical Library, National Aeronautics and Space Administration, 1985.
- [32] Lusk, S. L., Martin, C. D., Whiteley, J. D., and Johnson, W. V. (1990). "Time Delay Compensation Using Peripheral Visual Cues in an Aircraft Simulator", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-90-3129-CP).
- [33] Hettinger, L. J., McCauley, M. E., Cook, A. E., and Voorhees, J. W. (1989). "Summary of Proceedings of the First Meeting of the NASA Ames Simulator Sickness Steering Committee", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-89-3268-CP).
- [34] Casali, J. G., and Frank, L. H. (1986). "Perceptual Distortion and its Consequences in Vehicular Simulation: Basic Theory and Incidence of Simulator Sickness", Transportation Research Record 1059, pp. 57-65, Transportation Research Board, National Research Council.
- [35] Kennedy, R. S., Allgood, G. O., and Lilienthal, M. G. (1989). "Simulator Sickness on the Increase", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-89-3269-CP).
- [36] Frank, L. H. (1986). Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator, Ph. D. dissertation, Virginia Polytechnic Institute and State University.
- [37] Butrimas, S. K., and Browder, G. B. (1983). "Simulator Performance Definition by Cue Synchronization Analysis", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-83-1092-CP).
- [38] McFarland, R. E. and Bunnell, J. W. (1990). "Analyzing Time Delays in a Flight Simulation Environment", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-90-3174-CP).
- [39] Butrimas, S., and Browder, B. (1987). "A Unique Approach to Specification and Testing of Simulators", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2570-CP).

- [40] Sobiski, D. J., and Cardullo, F. M. (1987). "Predictive Compensation of Visual System Time Delays", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2434-CP).
- [41] McMillan, G. R. (1991). "Cue Integration and Synchronization", technical report, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, January, 1991.
- [42] McFarland, R. E. (1988). "Transport Delay Compensation for Computer-Generated Imagery Systems", NASA Technical Memorandum 100084, National Aeronautics and Space Administration, January, 1988.
- [43] Crane, D. F. (1980). "Time Delays in Flight Simulator Visual Displays", Proceedings of the 1980 Summer Computer Simulation Conference, pp. 552-557, Society for Computer Simulation, La Jolla, CA.
- [44] McFarland, R. E. (1986). "CGI Delay Compensation", NASA Technical Memorandum 86703, National Aeronautics and Space Administration, January, 1986.
- [45] Jewell, W. F., Clement, W. F., and Hogue, J. R. (1987). "Frequency Response Identification of a Computer-Generated Image Visual Simulator With and Without a Delay Compensation Scheme", Proceedings of AIAA Flight Simulation Technologies Conference (paper no. AIAA-87-2425-CP).
- [46] Hess, R. A., and Myers, A. A. (1985). "A Nonlinear Filter for Compensating for Time Delays in Manual Control Systems", technical report N85-14493, NASA Ames-Moffett Technical Library, National Aeronautics and Space Administration, 1985.
- [47] McFarland, R. E. (1992). Personal communication (letter with accompanying diagrams), November 10, 1992.
- [48] Wierwille, W. W., and Gutmann, J. C. (1978). "Comparison of Primary and Secondary Task Measures as a Function of Simulated Vehicle Dynamics and Driving Conditions", Human Factors, Vol. 20, No. 2, pp. 233-244.
- [49] Skipper, J. H., and Wierwille, W. W. (1986). "Drowsy Driver Detection Using Discriminant Analysis", Human Factors, Vol. 28, No. 5, pp. 527-540.
- [50] Repa, B. S., and Wierwille, W. W. (1976). "Driver Performance in Controlling a Driving Simulator with Varying Vehicle Response Characteristics", SAE Technical Paper Series (paper no. 760779), Society of Automotive Engineers, Inc.

- [51] Lincke, W., Richter, B., and Schmidt, R. (1973). "Simulation and Measurement of Driver Vehicle Handling Performance", SAE Technical Paper Series (paper no. 730489), Society of Automotive Engineers, Inc.
- [52] Wierwille, W. W., Casali, J. G., and Repa, B. S. (1983). "Driver Steering Reaction Time to Abrupt-Onset Crosswinds, as Measured in a Moving-Base Driving Simulator", Human Factors, Vol. 25, No. 1, pp. 103-116.
- [53] Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978). Statistics For Experimenters. John Wiley & Sons, New York, 1978.
- [54] McClave, J. T., and Dietrich, F. H. (1988). Statistics. Dellen Publishing Company, San Francisco, 1988.
- [55] Schaefer, R. L., and Anderson, R. B. (1989). The Student Edition of Minitab. Addison-Wesley, Reading, Mass., 1989.

