NASA/CR-2007-215095

Advanced Control Algorithms for Compensating the Phase Distortion Due to Transport Delay in Human-Machine Systems

Liwen Guo and Frank M. Cardullo
Department of Mechanical Engineering
State University of New York, Binghamton, New York

Lon C. Kelly
Unisys Corporation, Hampton, Virginia

December 2007

The NASA STI Program Office . . . in Profile

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2007-215095

Advanced Control Algorithms for Compensating the Phase Distortion Due to Transport Delay in Human-Machine Systems

Liwen Guo and Frank M. Cardullo
Department of Mechanical Engineering
State University of New York, Binghamton, New York

Lon C. Kelly
Unisys Corporation, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL06AA74T

December 2007

Acknowledgments

Jacob Houck of the Simulation Development and Analysis Branch at the NASA Langley Research Center assisted in the preparation of this report.

Available from:

NASA Center for AeroSpace Information (CASI)
7115 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Abstract

The desire to create more complex visual scenes in modern flight simulators outpaces recent increases in processor speed. As a result, simulation transport delay remains a problem. New approaches for compensating the transport delay in a flight simulator have been developed and are presented in this report. The lead/lag filter, the McFarland compensator and the Sobiski/Cardullo state space filter are three prominent compensators. The lead/lag filter provides some phase lead, while introducing significant gain distortion in the same frequency interval. The McFarland predictor can compensate for much longer delay and cause smaller gain error in low frequencies than the lead/lag filter, but the gain distortion beyond the design frequency interval is still significant, and it also causes large spikes in prediction. Though, theoretically, the Sobiski/Cardullo predictor, a state space filter, can compensate the longest delay with the least gain distortion among the three, it has remained in laboratory use due to several limitations.

The first novel compensator is an adaptive predictor that makes use of the Kalman filter algorithm in a unique manner. In this manner the predictor can accurately provide the desired amount of prediction, while significantly reducing the large spikes caused by the McFarland predictor. Among several simplified online adaptive predictors, this report illustrates mathematically why the stochastic approximation algorithm achieves the best compensation results. A second novel approach employed a reference aircraft dynamics model to implement a state space predictor on a flight simulator. The practical implementation formed the filter state vector from the operator's control input and the aircraft states. The relationship between the reference model and the compensator

performance was investigated in great detail, and the best performing reference model was selected for implementation in the final tests.

Theoretical analyses of data from offline simulations with time delay compensation show that both novel predictors effectively suppress the large spikes caused by the McFarland compensator. The phase errors of the three predictors are not significant. The adaptive predictor yields greater gain errors than the McFarland predictor for short delays (96 and 138 ms), but shows smaller errors for long delays (186 and 282 ms). The advantage of the adaptive predictor becomes more obvious for a longer time delay. Conversely, the state space predictor results in substantially smaller gain error than the other two predictors for all four delay cases.

Preface

This report is the first of two NASA contractor reports documenting the research on flight simulator transport delay compensation, undertaken in the Man-machine Systems Research Laboratory at the State University of New York at Binghamton and supported by the NASA Langley Research Center, in Hampton, Virginia. Loosely speaking, the two reports cover the theoretical research and the experimental testing of the research, respectively.

This report begins with a theoretical investigation of the effects of pure time delay on a control system consisting of an aerodynamic model, a pilot model and the Pade approximation of time delay. It then summarizes the literature study of transport delay causes in, and effects on, a flight simulator. This report continues with the introduction of three existing transport delay compensators—the lead/lag filter, the McFarland predictor and the Sobiski/Cardullo predictor, including intensive analyses of the strengths and limitations of each compensator. After a brief description of an expedient algorithm, designed to reduce the large spikes by the McFarland predictor, it presents the main body of research, i.e., development of two novel compensators. This report then thoroughly develops the adaptive predictor and the state space predictor. The adaptive predictor is a special Kalman filter that recursively updates the coefficients so that accurate prediction can be achieved. Among several versions of the adaptive algorithms, the Stochastic Approximation algorithm is mathematically demonstrated to achieve the best compensation results. The state space predictor makes use of the state transition matrix and its integral of a reference aircraft model. Several aircraft models were tested and the landing model of a large commercial transport in pitch achieved the best compensation

results as a reference model. By simplifying the state space predictor, the relationship between the compensation quality and the reference model was intensively investigated. Offline compensation results are presented to compare the McFarland predictor and the two novel predictors. The final part of the first report draws conclusions and suggests possible future research.

The second in the series, i.e., NASA CR 2007-215096[1] is presented in three parts: transport delay measurement in the NASA Langley Research Center's Visual Motion Simulator (VMS), piloted testing of the time delay compensators, and conclusions. The time delay measurement was conducted to verify the actual transport delay prior to the application of compensation in the final piloted tests. The average transport delay from the pilot control input to the visual display update was measured to be 90 ms. The second part of the report treats the final piloted experiment design, added time delay, test subjects, compensators, data collection, and evaluation metrics. It then presents the results of the final piloted tests in terms of performance errors, task load index, handling quality and power spectral density of the pilot controls. The final part of the report draws conclusions on the delay measurement and piloted simulation tests, and includes suggestions for future research. The appendices of the report include resultant graphs of all 13 pilots in terms of the four metrics, and the source code and flowcharts of some of the algorithms used in the research.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}$	Quadruple of a continuous state space control system
$\mathbf{A}_c$	Observer feedback gain matrix
$b_0 - b_2$	Coefficients of the McFarland predictor and the adaptive predictor
$c_0 - c_4$	Coefficients of the simplified state space predictor
e	Gain error metric of a predictor
$\mathbf{e}$	Error vector of a feedback state space system
$\dot{\mathbf{e}}$	Derivative of $\mathbf{e}$
E	Mathematical expectation
G	Feed forward gain of a control system with a state space predictor
$\mathbf{G},\mathbf{H},\mathbf{C},\mathbf{D}$	Quadruple of a discrete state space control system
h	Aircraft altitude
I	Cost function
$\mathbf{I}$	Identity matrix
$\mathbf{j}$	Velocity vector $\begin{bmatrix} v(k) & v(k-1) & v(k-2) \end{bmatrix}^T$
$\hat{k}_x$	Autocorrelation sequence of signal x
$\mathbf{K}$	Feedback gain matrix, or the Kalman matrix
l	Aircraft longitude (arc length)
l_θ	Aircraft longitude (angle)
$\mathbf{L}$	Observer matrix
L, L^{-1}	Laplace transform and inverse Laplacetransform
P_x	Exact power spectral of signal x from discrete Fourier Transform

$\hat{P}_x^f(\theta)$	Estimated power spectral of signal x
$\mathbf{P}$	Intermediate matrix to update the Kalman matrix
$\mathbf{P}_{B2E}^A$	Angular velocity transformation matrix from body frame to the earth frame
$\mathbf{P}_{B2E}^T$	Translational velocity transformation matrix from body frame to the earth frame
p,q,r	Aircraft angular velocities in the body frame
t_d	Time delay
T	Sampling period
tr	Trace operation of a matrix
u	Pilot control input
v	Velocity of y ; reference input
$\mathbf{v}_K^B$	Aircraft angular velocity vector in the body frame
$\mathbf{v}_N^B$	Aircraft translational velocity vector in the body frame
$\mathbf{v}_K^E$	Aircraft angular velocity vector in the earth or topodetic frame
$\mathbf{v}_N^E$	Aircraft translational velocity vector in the earth or topodetic frame
w	Digital window sequence
$\mathbf{x}$	State vector of a control system
$\dot{\mathbf{x}}$	Derivative of $\mathbf{x}$
$\tilde{\mathbf{x}}$	Observed $\mathbf{x}$
$\dot{\tilde{\mathbf{x}}}$	Derivative of $\tilde{\mathbf{x}}$
$\mathbf{x}_a$	Aircraft state vector
$\mathbf{x}_f$	Filter state vector
$\mathbf{x}_p$	Predicted filter state vector

y	Undelayed aircraft state (output of the EOM)
$\tilde{y}$	Observed y
$\dot{\tilde{y}}$	Derivative of $\tilde{y}$
y_c	Compensated aircraft state (i.e., y_p delayed by t_d)
y_d	Delayed aircraft state
y_p	Predicted aircraft state
Z, Z^{-1}	Z-transform and inverse Z-transform
ϕ	Aircraft roll angle
$\mathbf{\Phi}$	State transition matrix for continuous system
$\mathbf{\Phi}_d$	State transition matrix for discrete system
λ	Aircraft latitude (arc length); forgetting factor;
λ_i	Eigenvalues
λ_θ	Aircraft latitude (angle)
θ	Aircraft pitch angle
$\tilde{\mathbf{\theta}}$	Estimate of $[b_0 \quad b_1 \quad b_2]^T$ (output of the Kalman filter algorithm)
ψ	Aircraft yaw angle
$\mathbf{\Psi}$	Integral matrix of $\mathbf{\Phi}$
$\mathbf{\Psi}_d$	Integral matrix of $\mathbf{\Phi}_d$
ω	Angular frequency
ω_c	Crossover frequency
ω_n, ζ	Natural frequency and damping ratio

Acronyms

AP	Adaptive Predictor
CHR	Cooper-Harper Rating
CRT	Cathode Ray Tube
DFT	Discrete Fourier Transform
DP	Display Processor
DOF	Degree Of Freedom
EOM	Equations Of Motion
EVDAS	Electronic Visual Data Acquisition System
FEC	Front End Computer
FRA	Frequency Response Analyzer
GP	Geometry Processor
GS	Glide Slope
GSE	Glide Slope Error
HQR	Handling Quality Rating
LMS	Least Mean Square
LSF	Least Squares Fitting
LTI	Linear Time-Invariant
MF	McFarland predictor
MFR	McFarland predictor with spike Reduction
NC	No Compensation
NEU	North East Up frame

ODE	Ordinary Differential Equation
PIO	Pilot-Induced Oscillation
PS	Pitch Stick
PSD	Power Spectral Density
FPSD	Frequency of the highest PSD peak
IPSD	Integrated PSD
RMSE	Root Mean Squared Error
RP	Rudder Pedal
RS	Roll Stick
SA	Stochastic Approximation
SIMES	SIMulator Evaluation System
SISO	Single Input Single Output
SS	State Space predictor
SSQ	Simulator Sickness Quantity
STD	STandard Deviation
TD	TouchDown
TDE	TouchDown Error
TLX	Task Load Index
VA	Visual to Analog
VMS	Visual Motion Simulator
ZOH	Zero-Order Hold

1. Introduction

1.1. Transport Delay in Vehicle Simulation

The transport delay in a vehicle simulator is the time elapsed from an operator's control input until an appropriate stimulus is presented to the operator by the associated hardware[2]. In a real vehicle, the transport delay is negligible because the vehicle responds to the operator command almost instaneously. Unfortunately, this is not true for a vehicle simulator. As an example, Fig.1.1 shows a motorcycle simulator. Unlike a driver on a real moving motorcycle who directly feels the motion of the motorcycle relative to the street, the driver on this simulator perceives the motion primarily based on the visual display showing the movement of the road and the surroundings. The time it takes for the simulator computers to generate a new visual image on the screen based on the operator's control input is the transport delay. To illustrate the sources of the transport delay, Fig.1.2 shows the architecture of an ordinary vehicle simulator with a visual system.

The transport delay comes primarily from three sources: sampling delay, processing time and data transfer time. Sampling delay results because the simulator dynamics computer only samples the operator's control input at the beginning of each computation frame whereas the actual control input arrives stochastically. Therefore the change of input between two consecutive sampling events is delayed. It may be as long as almost a full frame, or as short as zero, but the average of the sampling uncertainty is a half frame. The processing time consists of two parts—the time taken by the dynamics computer to calculate the vehicle states from the sampled operator's control input, and the time for the computers in the visual system to prepare the visual image. The

processing time usually dominates the total transport delay. Data transfer time is the time it takes for the visual system to receive the updated vehicle state computed by the dynamics computer. If the update rates of the vehicle dynamics computer and the visual system are not equal and the latter is not an integer multiple of the former, communication asynchrony occurs which results in additional delay. If the transfers are asynchronous, the data transfer delay affects the sampling delay. As long as the transfer time is less than the sampling interval (i.e., the frame length), transfer time may be considered the same as processing time. Although the simulator time delay consists of several components from different subsystems, the origin makes no difference to the operator, who only feels the total effect.

Fig. 1.1. A motorcycle simulator

In Fig. 1.2, the sampling delay occurs between the hand and the plant, the processing delay occurs in the plant and between the output and the display, and the delay

due to data transfer may arise if there is a difference in update rate between the plant and the display system.

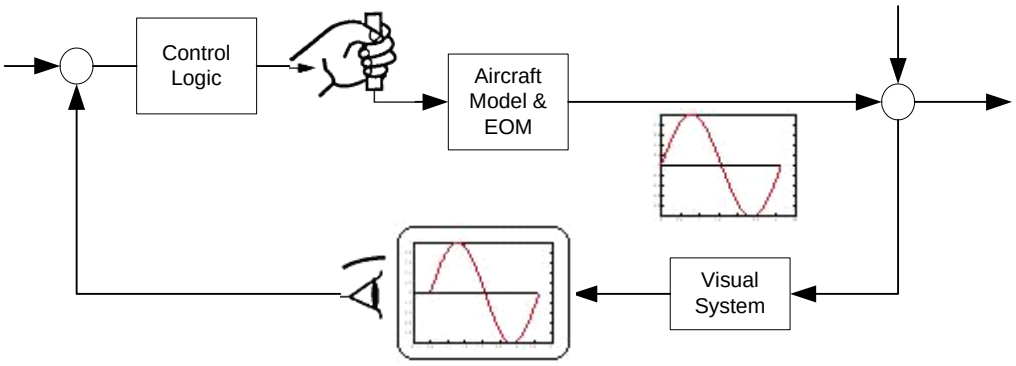


Fig. 1.2. Architecture of a vehicle simulator with a visual system

If the overall delay reaches a noticeable level, when the operator tries to perform a task, say a left turn, she will see insufficient response from the display relative to her expectations; hence the operator's cognitive control logic causes her to maneuver further until the expected display is observed; but because of the delay, the display will show the operator that she has already over controlled, resulting in a compensation or a modification, and so on. The resulting locus of the motorcycle positions might resemble the dashed curve in Fig. 1.3.

This example demonstrates that one of the immediate effects of long transport delay is Pilot Induced Oscillation (PIO). As the time delay gets longer, the oscillation is expected to be more severe—with a larger magnitude and a slower decay, which may even become unstable. In other words, time delay makes the system's response slower and undermines the system stability. As a result, the virtual vehicle is harder to control with time delay, indicating that the operator's perception of the handling quality becomes worse, and the control workload is increased. Using Fig 1.3, it is easy to visualize the degradation in performance by comparing the actual trajectory, represented by the dashed

curve with the ideal trajectory, represented by the solid curve. In summary, the following problems caused by the time delay are expected:

1) The man-machine system performance is degraded;
2) The operator's control workload is increased due to over control and modification;
3) The operator's assessment of the handling quality of the system is diminished.

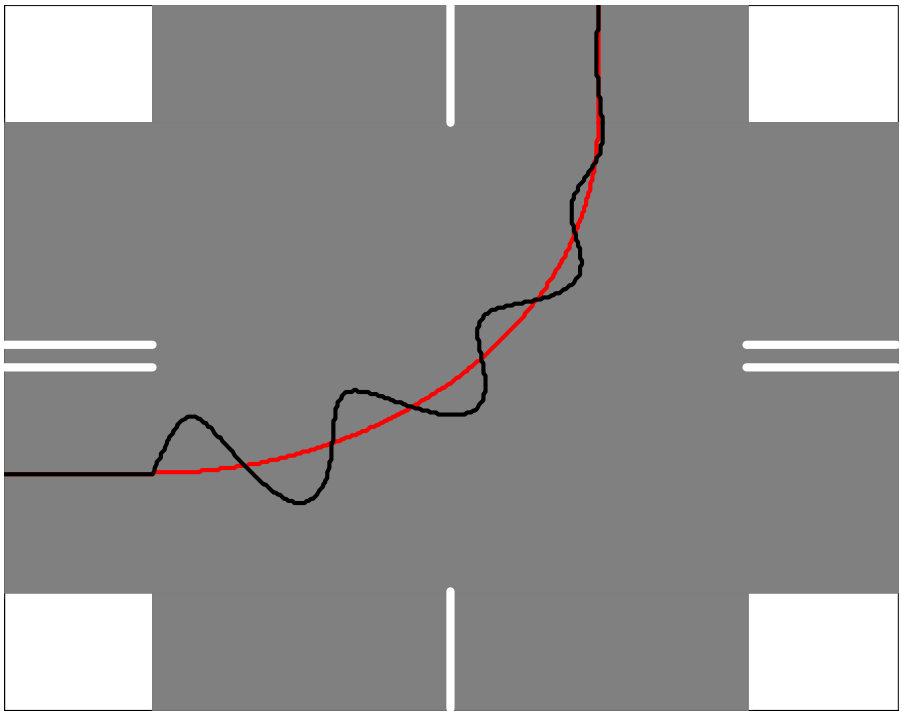


Fig. 1.3. Ideal turn of a real motorcycle versus an actual turn in a simulator

In the frequency domain, time delay shifts the time-line of the simulated vehicle to the right, with respect to the response of the real vehicle, causing a phase lag in the simulation system. This phase lag is proportional to the frequency components of the operator's control input. The phase lag at the system crossover frequency decreases the system phase margin; it also contributes to the PIO and undermines system stability. To

restore the system phase margin, the operator tries to increase the control gain or lower the system crossover frequency, resulting in an increase in control workload, and degrading the handling quality. The frequency analysis agrees with the time domain analysis.

The literature supports the above analyses of the effects of transport delay on a man-in-the-loop flight simulator system. Several metrics indicate that transport delay degrades the man-machine system performance. Transport delay increases the system Root Mean Square Error (RMSE) associated with various tasks (Riccio, et, al, Bailey, et al); the Power Spectral Density (PSD) analyses of the operator controls demonstrate that the time delay makes the operator's workload increase, especially in the high frequencies (Middendorf, et al, Guo, et al); the Cooper-Harper Rating (CHR) also shows that the operator's handling quality assessment is affected by the delay (Cooper and Harris). Large transport delays may also induce simulator sickness (Zaychik, et al). (The literature study of the time delay effects is elaborated in Chapter 2.)

1.2. Delay Compensation

Because the impact is undesirable in flight simulations, simulator transport delay must be minimized in order to reduce its effects. If, after minimization, the transport delay still exceeds the tolerable threshold for maintaining desirable simulator performance, algorithms to compensate for the delay should be employed. Delay compensation usually makes use of prediction of the aircraft states before they are output to the cueing channels. This is illustrated in Fig. 1.4, where, in the small plot to the right of the predictor block, the black dashed curve is the predicted aircraft state. Images based on the predicted aircraft state can be used to offset the transport delay in the visual system.

The purpose of prediction is to restore the phase margin, which would be reduced by the transport delay.

Prediction is achieved by making use of past and current system information, including the aircraft displacement, velocity and acceleration, the operator control input, the dynamic model, and so on. What information is used for the prediction and how to use it lead to various ways of designing the compensator. The most prominent three compensators are the lead/lag filter, the McFarland predictor and the Sobiski/Cardullo predictor.

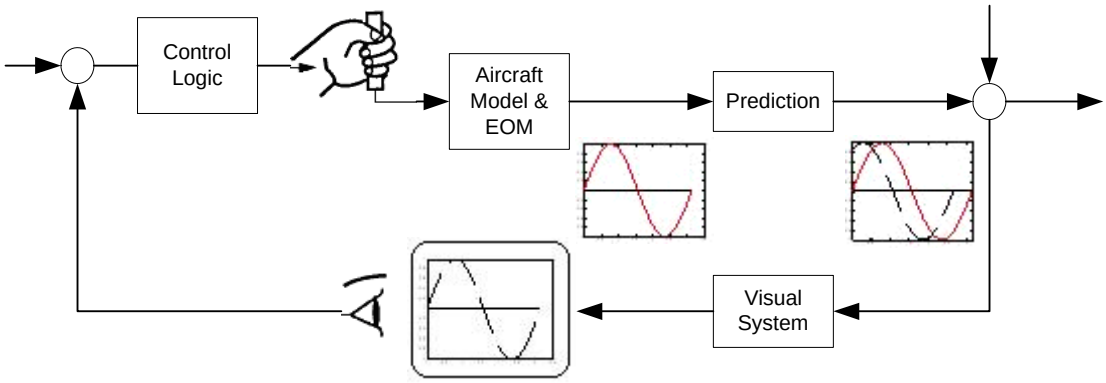


Fig. 1.4. Delay compensation based on prediction

The lead/lag filter had long been used in industry before Ricard/Harris introduced it to the flight simulator to compensate for the transport delay. Having a single pole and a single zero, the lead/lag filter provides some phase lead in a certain frequency range while introducing gain distortion. In order to properly design a lead/lag filter, the designer must determine the pole, the zero and the gain appropriate for the transport delay to be compensated such that the phase lead and gain distortion are well balanced. Both Ricard/Harris and Crane proposed methods of designing the lead/lag filter, and they tested the compensation against the performance of piloted simulations. Though both methods show some advantages, the lead/lag compensator has been replaced by other

more powerful predictive filters, primarily due to its limited ability to provide phase lead and the undesirable introduction of significant gain distortion. McFarland developed a discrete filter, which extrapolates the future aircraft displacement, from three consecutive iterations of velocity. This special integration algorithm is a type of finite impulse response filter because it only has poles at the origin. Because the current prediction does not involve the past predictions, the prediction error is not passed to the next iteration, and therefore, there is no error accumulation. The large gain distortion, which would be present when using the lead/lag filter is significantly reduced while phase lead is substantially increased. The challenge in designing this type of filter is to determine the three coefficients that multiply the three steps of velocity. McFarland introduced a method known as sinusoidal tuning, which makes use of boundary conditions of the so-called “pass band”. The pass band is defined to be the primary frequency band for most pilot operations. While the McFarland filter works well within this pass band, the gain distortion and phase lead deficiency are significant, and the gain distortion leads to very disturbing spikes in the prediction. The spikes originate from the constant coefficients, which were determined using the sinusoidal tuning, and are not adjusted during the simulation.

The Sobiski/Cardullo predictor is the first state space filter used for compensating the transport delay in a flight simulation. It was derived from the solution of a linear time-invariant (LTI) differential equation in state space format. By using more information, in each iteration of prediction, theoretically the full order Sobiski/Cardullo filter should achieve better compensation (sufficient phase lead and less gain error) than prior techniques, provided that the aerodynamics are also LTI and known.

However, there are three practical problems which prevent the widespread application of the Sobiski/Cardullo predictor to flight simulators. First, most modern flight simulators include complex, nonlinear, time variant aircraft models. Second, when using the Sobiski/Cardullo predictor, the extrapolated state is only valid if the operator's control input is piece-wise constant, sinusoidal, or exponentially decaying, etc. Third, the matrix operations used to implement the Sobiski/Cardullo predictor make it computationally intensive, and simplifying the algorithm would make it more practical.

1.3. Scope of Research

This is a comprehensive study of the transport delay in a vehicle simulator, from its sources, to its effects, measurement and compensation. In Chapter 2, a theoretical analysis of a pure time delay—its effects on a control system in both the time and frequency domain—is presented. The second part of Chapter 2 is a summary of a literature study on the causes and effects of the transport delay in a flight simulator.

Chapter 3 describes the three prominent compensation techniques, the lead/lag, McFarland and Sobiski/Cardullo filters, which were briefly introduced in this chapter (Section 1.2), in much more detail. The basic principles, the formulation, and the advantages and disadvantages of each filter will be presented in this chapter. Analyses of these filters in both the time and frequency domains are also presented.

The equation proposed by Crane for positioning the pole of the lead/lag filter has been revised, and a filter designed with the revised equation shows obvious improvement over those designed using Crane's original equation. The revision is introduced in Chapter 3.

Chapter 4 introduces two novel predictors for compensating transport delay. Section 4.1 presents a simple spike reduction algorithm to alleviate the gain distortion caused by the McFarland compensator. In section 4.2, a new adaptive predictor is introduced. The Kalman estimator, which is an online recursive least squares method has been adopted to design the coefficients of a predictive compensator, which also uses three consecutive velocities to extrapolate, similar to the McFarland compensator. While simplifying the Kalman filter algorithm, a forgetting factor, the Kaczmarz algorithm, the stochastic approximation algorithm and the Least Mean Squares algorithm are introduced. This section also mathematically demonstrates why the stochastic approximation algorithm stands out above all the other adaptive algorithms in compensating the transport delay.

A second novel approach employes a reference aircraft dynamic model to implement a state space predictor for use on a flight simulator. The practical implementation formed the filter state vector from the operator's control input and the aircraft states. Among several reference models tested, the landing model of a large commercial transport in the pitch axis achieves the best compensation result, and was selected for the final piloted tests. The relationship between the reference model and the compensation performance is also investigated in detail in chapter 4.

Theoretical analysis of the two novel compensators is the main topic of Chapter 5. It covers an evaluation and comparison of the errors caused by a compensator with different predictors, and includes a sensitivity analysis, which demonstrates how the compensation errors change as the time delay increases. The chapter begins by defining two error metrics, and then compares the compensation errors in terms of the two error

metrics using offline tests (see Chapter 5). The chapter concludes with a sensitivity analysis.

2. Background Information

2.1. Theoretical Description of Time Delay

A pure time delay simply causes a signal to shift right on the time line. As an example, Fig.2.1 shows a sinusoid signal ($y(t) = sin(\omega t)$) and the resulting signal when it is delayed by t_d . The delayed signal, $y_d(t) = sin(\omega t - \omega t_d)$, has a phase lag of ωt_d with respect to the initial signal. This example demonstrates that time delay can be described in both time domain and frequency domain. The transform of a time delay t_d between the time domain and the frequency domain is given in Eq. (2.1).

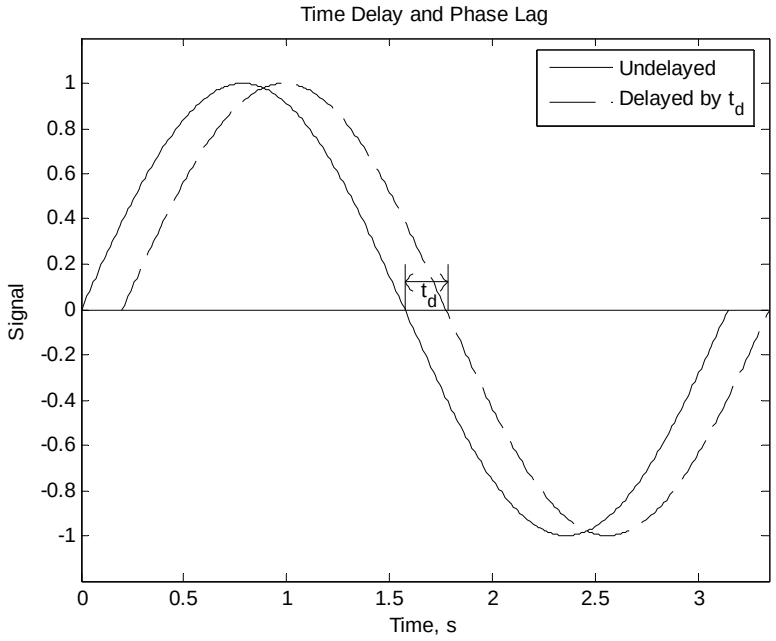


Fig. 2.1. A sinusoid signal and its delayed result by t_d

$$f(t - t_d) \Leftrightarrow e^{-j\omega t_d} F(\omega t_d) \tag{2.1}$$

If the signal is continuous, the delayed signal is simply given by

$$y_d(t) = y(t - t_d) \tag{2.2}$$

And for a continuous signal, the relationship between it and its delayed partner in the frequency domain is given by the Laplace transfer function

$$\frac{Y_d(s)}{Y(s)} = e^{-t_d s} \tag{2.3}$$

Because $e^{-t_d s}$ is nonlinear, it is usually approximated by the 2nd-order Pade approximation

$$e^{-t_d s} \approx \frac{s^2 - \frac{6}{t_d}s + \frac{12}{t_d^2}}{s^2 + \frac{6}{t_d}s + \frac{12}{t_d^2}} \tag{2.4}$$

In a discrete system, if the time delay is an integer multiple of the frame time T, say $t_d = n_d T$, where n_d is an integer, the counterparts of Eq. (2.2) and (2.3) are simply given by Eq. (2.5) and (2.6) respectively

$$y_d(k) = y(k - n_d) \tag{2.5}$$

$$\frac{Y_d(z)}{Y(z)} = z^{-n_d} \tag{2.6}$$

However, if the ratio $r = t_d / T$ is not an integer, these relationships become much more complicated. Substituting the trapezoidal integration, $\frac{1}{s} = \frac{T}{2}\frac{z+1}{z-1}$ into Eq. (2.4), the Pade approximation becomes

$$\frac{Y_d(z)}{Y(z)} \approx \frac{\beta_0 + \beta_1 z^{-1} + z^{-2}}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} \tag{2.7}$$

where $\alpha_2 = \beta_0 = \frac{3r^2 - 3r + 1}{3r^2 + 3r + 1}$ and $\alpha_1 = \beta_1 = \frac{6r^2 - 2}{3r^2 + 3r + 1}$. And the difference equation corresponding to Eq. (2.5) is given by

$$y_d(k) = -\alpha_1 y_d(k-1) - \alpha_2 y_d(k-2) + \beta_0 y(k) + \beta_1 y(k-1) + \beta_2 y(k-2) \tag{2.8}$$

It follows from Eq. (2.3) that the transfer function of a pure time delay has unity magnitude at all frequencies, but has negative phase angle, calculated by

$$\phi_d = t_d \omega \tag{2.9}$$

This can be verified by using the Bode diagram of a time delay as shown in Fig. 2.2, in which both the exact calculation and the 2nd-order Pade approximation are plotted. When a time delay is added to an open loop system, it only delays the output without causing any gain distortion. Take Fig. 2.1, for example, if a time delay of t_d is applied to the solid sinusoid signal, it moves the curve to the right and becomes the dashed curve.

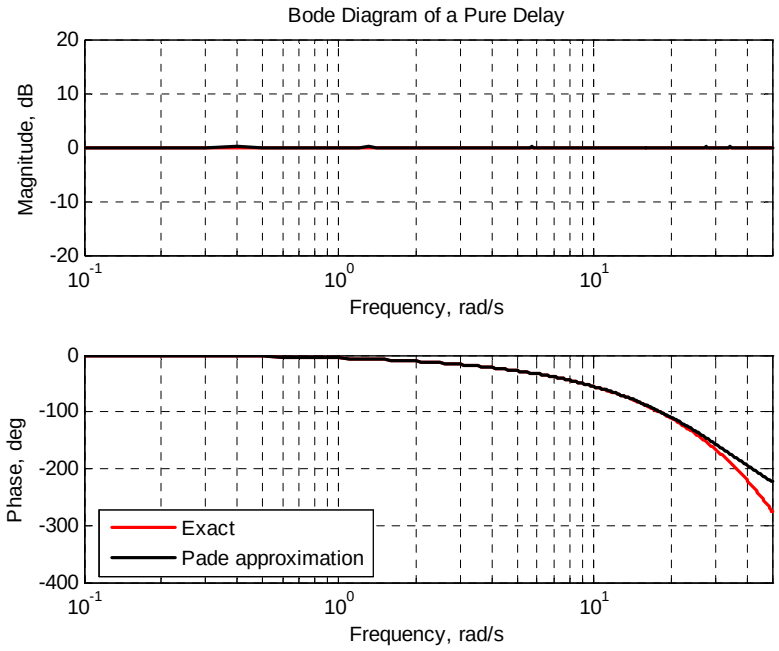


Fig. 2.2. Bode diagram of a pure delay: exact calculation and Pade approximation

However, if time delay is introduced in a closed loop system, the system output is shifted to the right, and the gain changes, because the system feedback is also delayed. Delayed feedback makes the system sluggish so that it becomes more oscillatory and its stability is undermined. If the delay is sufficiently large, the system could become

unstable. This can be illustrated by modeling a flight simulation task, as shown in Fig. 2.3. The pilot model, given by Eq. (2.10), matches a lateral control task performed with a rate controller cascading a delay term representing the neuromuscular and cognitive time delay, which were lumped into the predictor. The aircraft model, given by Eq. (2.11) represents the change in the roll angle per unit of deflection of the control stick, at a flight condition of 430 knots airspeed and 30,000 feet altitude. The time delay block refers to the artificially inserted transport delay (denoted by t_d) represented by the 2nd-order Pade approximation (Eq. (2.12)). Three values of the artificial delay were tested: 0, 200, or 400 ms, and the closed-loop step responses and the open loop frequency responses of these three cases are given in Fig. 2.4 and Fig. 2.5.

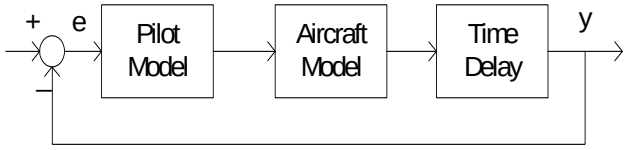


Fig. 2.3. Block diagram of a simulation with a man-in-the-loop control

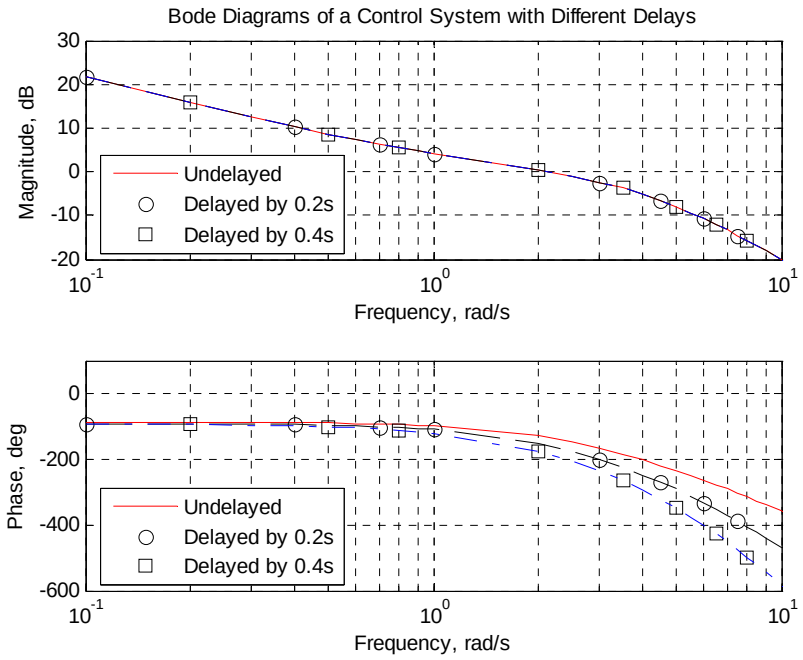


Fig. 2.4. Bode Diagrams of a closed loop system with different delays

$$H_p(s) = \frac{18(s+1)e^{-0.3s}}{(s+3)(s+9)} \tag{2.10}$$

$$\frac{\varphi(s)}{\delta_e(s)} = 5.57\frac{\frac{s^2}{3.46}+\frac{0.48s}{1.86}+1}{s(0.16s+1)(s^2+0.8929s+1.8788^2)} \tag{2.11}$$

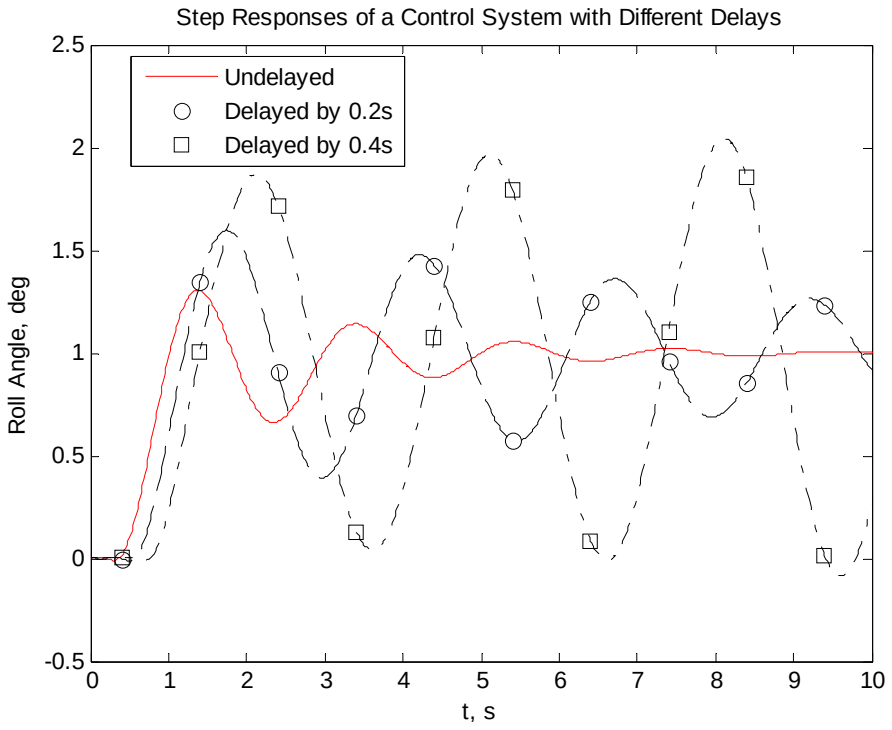


Fig. 2.5. Step responses of a closed loop system with different delays

From the Bode diagrams of the open loop system, it is appearent that the magnitude is not changed by the time delay, but the phase angle is decreased, which agrees with the unity magnitude and negative phase angle properties of the delay. According to Eq. (2.9), the phase margin decrease at the crossover frequency is proportional to the amount of time delay

$$\phi_{PM} = t_d \omega_c \tag{2.12}$$

For a 200 ms delay, the phase margin is reduced considerably but is still positive, which means the system is still stable, yet becomes more oscillatory. When the delay is 400 ms, the phase margin is negative, which indicates an unstable system. The step responses of the closed-loop system confirm the frequency domain analysis (Fig. 2.5). Note that although for 200 ms delay the step response has more oscillations, it still converges to the same steady state value.

The impact of time delay on the man-machine system is not only dependent upon the amount of the delay, but also the system dynamics. Specifically, the system bandwidth or crossover frequency impacts the effects of time delay on the system. To illustrate, a second system is created by varying the real part of the dominant poles of the aircraft model given in Eq. (2.11) (i.e., by changing the natural frequency from 1.8788 rad/s to 1.95 rad/s) with the operator model unchanged. The bandwidth of the new system (System II) is larger than that of the original system (System I). For comparison, some properties of the systems are listed in Table 2.1. The table shows that time delay decreases the system closed-loop bandwidth, and the longer the time delay, the larger the decrease in close-loop bandwidth. In addition, the system with higher bandwidth tends to suffer faster closed-loop bandwidth reduction with time delay.

Table 2.1. Properties of two dynamic systems

Properties	System I	System II
Damping ratio	0.2376	0.2376
Natural frequency (rad/s)	1.8788	1.9500
Crossover frequency (rad/s)	2.19	2.54
Closed-loop bandwidth (rad/s)	4.3238	4.4864
Closed-loop bandwidth (rad/s) with 150 ms delay	3.9131	4.0338
Closed-loop bandwidth (rad/s) with 300 ms delay	3.4760	3.5801
Phase margin (deg)	44.1922	34.4364
Maximal tolerable delay (s)	0.352	0.237

Fig. 2.6 shows the closed-loop step responses of these two systems with 0, 150 and 300 ms added delay. With zero added delay, system I (upper figure) has slower responsiveness than system II because its bandwidth is lower. However, with a 150 ms delay, system I responds faster than system II, showing that a dynamic system with

higher bandwidth tends to be impacted more by the same amount of delay. With a delay of 300 ms, system I becomes more oscillatory but is still stable, whereas system II is no longer stable. Further analysis shows that although the two systems have the same phase margin, system I can tolerate much longer maximal time delay than system II (344 ms vs. 238 ms). This can be interpreted using Eq. (2.12). Usually, the system with higher bandwidth tends to have higher crossover frequency (ω_c). It follows from Eq. (2.12) that the system with the higher crossover frequency suffers a larger phase lag ϕ_d with the same time delay. And because $t_d = \phi_{PM} / \omega_c$, given the same phase margin (ϕ_{PM}), the system with lower crossover frequency tolerates longer delay.

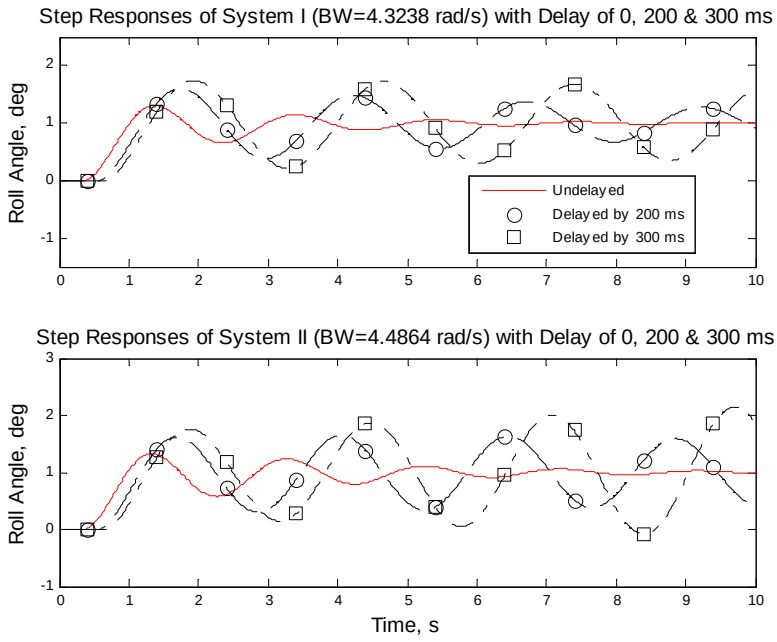


Fig. 2.6. Step responses of the two dynamic systems to different delay

2.2. Sources of Transport Delay

This section is a summary of the sources of transport delay in a flight simulator, and some proposed methods to reduce the delay from each source.

Fig. 2.7 is a view of part of a flight simulator cockpit. The out-of-the-window display, i.e., the sea, the aircraft carrier and the lead helicopter, provides the visual cue to the pilot; the gauges, dials and photodiodes on the panel in front of the pilot provide the instrument cues, such as the airspeed, altitude and attitude; and the dynamic seat and the motion of the cockpit (if any) provide the motion cues. These are the three primary cueing channels, which serve as feedback to the operator in response to his input. There may be other cues available, e.g., the sound system may indicate the aural environment surrounding the pilot.

Fig. 2.7. Cockpit view of a flight simulator

When the pilot issues a command, the basic cues provide feedbacks to the operator, who continues to make control inputs based on these cues. The operator loop closes the ***man-machine system***—the flight simulation. Because the simulator computers need time to generate the cues, they do not respond to the operator's command instantaneously. The time it takes for the simulator to generate the basic cues is called

transport delay. Although the processing speed of computers has been improving, the increasing complexity of the simulator display negates the potiential to lessen the transport delay. As a result, transport delay still exists on state-of-the-art-flight simulators. This transport delay has a number of sources:

1) Sampling delay—the time the simulation computer takes to sense a control input;
2) Plant delay—the time the simulation computer uses to calculate the aircraft states;
3) Data transfer delay—the time lapse from when the aircraft states are available to the point when the motion system, the visual system and/or the instrument system receives the signal;
4) Cueing delay—the time it takes each individual system to respond to aircraft states.

All these systems are likely to contribute transport delays, though these delays are not always present in concert, and more than likely, they will all be different from each other.

2.2.1. Sampling Delay

In a digital simulation, sampling delay arises from the fact that the simulator regularly senses the control input at the beginning of each frame, whereas the input arrives stochastically. For the zero-order hold (ZOH[*]) system, the sampled value is held till the next sampling. Therefore, an input change in the middle of a frame is either lost or

---

[*] Though the accuracy of the first-order hold is higher than that of the ZOH, ZOH is used in the flight simulator because the first-order hold introduces an extra frame of delay.

delayed until the beginning of the next frame. The amount of delay varies in each frame, resulting in a sampling uncertainty. Fig. 2.8 illustrates the cause of sampling uncertainty.

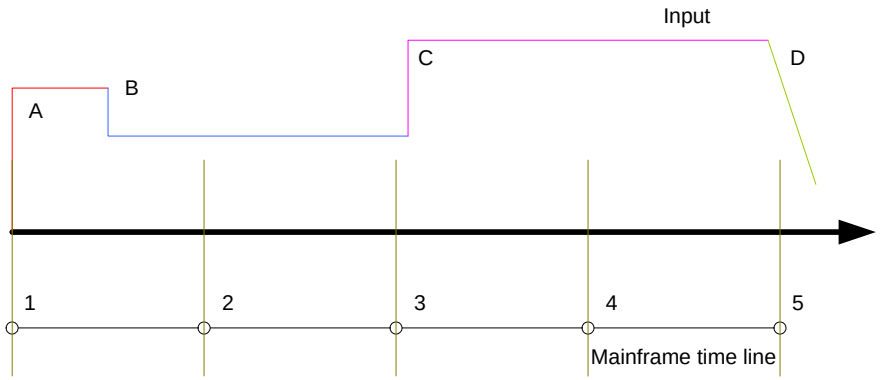


Fig. 2.8. Illustration of sampling delay

The input change at point A coincides with the first sampling by chance, and thus the sampling delay is zero, representing the best-case scenario; the input at point B is not sensed by the computer until the second sampling, and the sampling delay is half a frame; point C is right after the 3rd sampling, and it will be sensed in the 4th sampling with a delay of almost a full frame, representing the worst-case scenario; for input at point D, the sampling delay is just a small part of a frame. With the exception of the best-case scenario, the sampled value represents the past input. Because the input may arrive at any time point during a frame, some sampling uncertainty results. As the number of frames becomes sufficiently large, the statistical average of the sampling delay is a half frame.

Although the sampling delay can be reduced by using shorter frames, the frame length is driven primarily by the aerodynamics computation, rather than the sampling. Another way to reduce the average sampling delay is the multi-rate sampling proposed by R. M. Howe[3], but this results in higher hardware and software costs.

2.2.2. Plant Delay

The plant in a flight simulator refers to the process of sampling the pilot control inputs, and then calculating the updated aircraft state using a specific aerodynamic model. The model first calculates the aircraft accelerations in all six degrees of freedom (DOF) in the body frame, then integrates the accelerations to get the velocities and displacements, and finally transforms the updated aircraft states (displacements, velocities and accelerations) to other frames, such as the world frame and the topodetic frame, for use by the individual cueing systems. Usually the process of calculating the accelerations, using the non-linear acceleration functions, is computationally intensive.

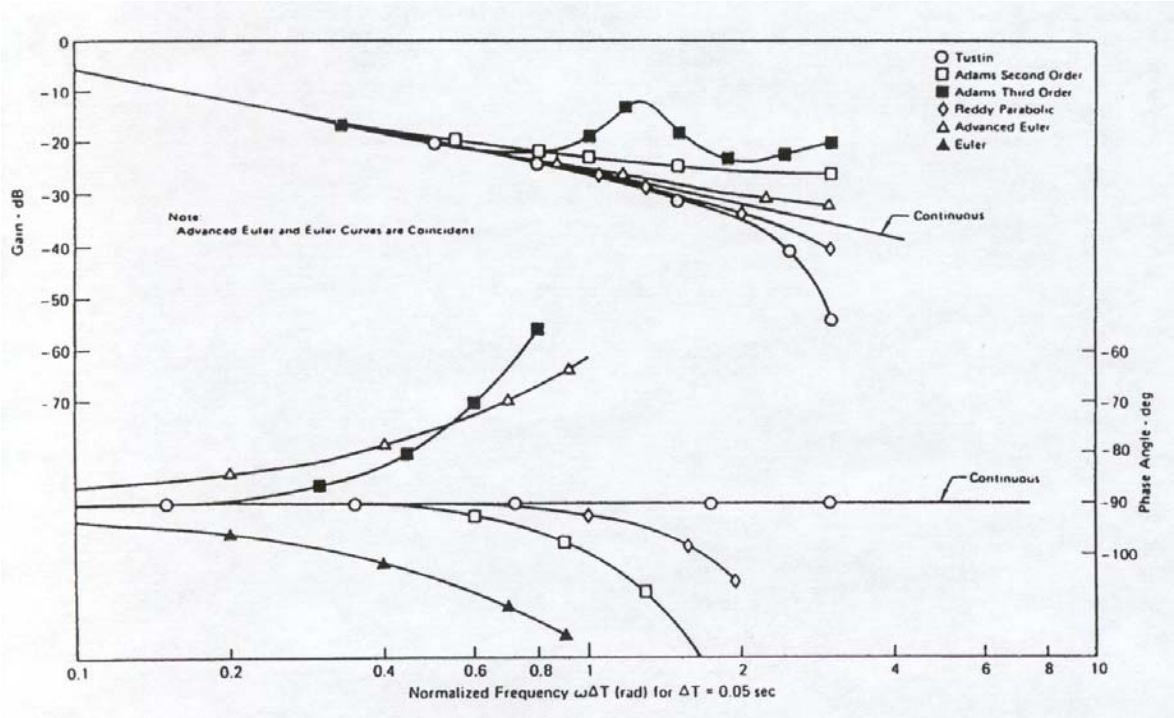


Fig. 2.9. Bode diagrams of several numerical integration algorithms

Although the two numerical integration processes themselves take little time to perform, they introduce phase lag and latency. Different integration methods have different characteristics in introducing phase lag, which is a function of the working

frequencies. Fig. 2.9 presents a comparison of some numerical integration methods in terms of both phase lag and gain distortion[4].

When two numerical integration methods are carried out one after another, they may cause latency due to the dependency of the latter on the outputs of the former for several iterations. For example, two 2nd-order Adams-Bashforth integrations cause an extra frame of latency. If the second integration (from velocity to displacement) is replaced with the trapezoidal integration, this extra frame of latency could be removed. Therefore, the aerodynamics processing delay can be reduced by judiciously choosing the integration combination, and by rearranging the order of computation within each integration frame.

2.2.3. Data Transfer Delay

Data transfer time is the time it takes for the visual system (or the motion or instrument systems) to get the output signals computed by the simulation computer[5]. If the update rate of the simulation computer and the visual system are not the same and the latter is not an integer multiple of the former, communication asynchrony occurs which results in delay. Asynchronous delay occurs when the vehicle states are available from the host computer but the image processor is not ready to receive them until the beginning of the next frame. According to Richard McFarland[6], the asynchronous delay between an image processor and a simulation computer appears to be random, but it is actually periodic. The periodicity can be demonstrated with the time lines shown in Fig. 2.10. The periodic component of the asynchronous delay may be given as

$$t_d(m) = T\left\{ m\frac{M}{N} - \left\lfloor m\frac{M}{N} \right\rfloor \right\} \tag{2.13}$$

where m is the iteration index of the visual computer, T is the update period of the host computer, and $\frac{M}{N}$ is the minimal fraction equal to the ratio of $\frac{T'}{T}$ (T' is the update period of the visual computer), or M and N must be relatively prime. For the above example

$$t_d(1)=30, t_d(2)=20, t_d(3)=10, t_d(4)=0, t_d(5)=30, \cdots$$

And the average asynchronous delay is

$$\bar{t}_d = \frac{1}{N}\sum_{m=1}^{N} t_d(m) \tag{2.14}$$

For the above example,

$$\bar{t}_d = \frac{1}{4}(30+20+10+0) = 17.5(ms)$$

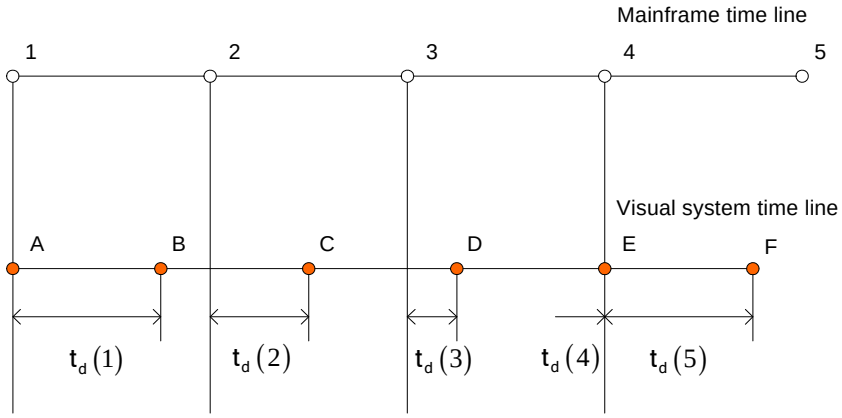


Fig. 2.10. Illustration of periodical asynchronous time delay

If the transfers are asynchronous, the data transfer delay affects the sampling delay. As long as the transfer time is less than the sampling interval (i. e., the frame length), transfer time may be considered the same as processing time[7]. To reduce or even eliminate the asynchronous time delay, the update rates of the subsystems must be equal to each other, or at least the update rate of the down stream subsystems should be an integer multiple of that of the upstream one.

2.2.4. Cueing Delay

When the aircraft states are available from the simulation computer, the three basic visual, motion and instrument systems use them to generate the individual cues in the hardware systems, as depicted in Fig. 2.11. Each of the three cueing systems requires a different amount of time to generate and execute the hardware commands, which leads to cueing mismatches.

A. Instrument Delay

The quantities to be displayed on the cockpit instruments, such as the position and orientation of the aircraft, are available from the simulation computer, and hence there are very few time-consuming computations in the instrument system. The calculations necessary to transform the variables into the positions of dials of the indicators, and the digital-to-analog conversion contribute no more than one frame of transport delay.

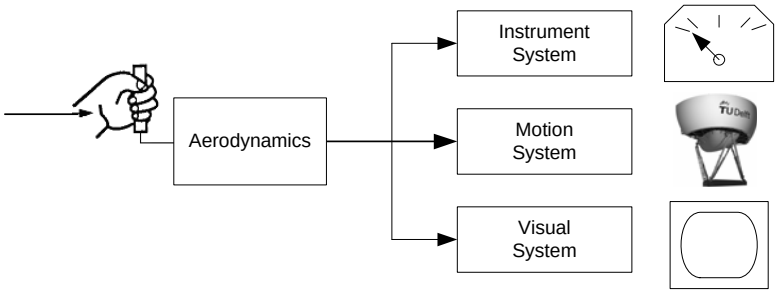


Fig. 2.11. Three basic cueing channels

B. Motion Delay

The purpose of the motion system of a flight simulator is to impart motion cues to the pilot, which are very similar to those he would experience in an actual aircraft. Because of the physical limits of any ground based simulator motion system, (i.e., it can not reproduce the large motions of a real aircraft), the simulator motion commands (attitude and rate) must be calculated from the aircraft states. This process is the core of

the motion cueing algorithm, which makes use of an operator motion sensation model and control theory. Various combinations of operator motion models and control theories lead to different types of motion cueing algorithms. The more advanced motion cueing algorithms may need to solve a Riccati equation that is computationally intensive. Telban and Cardullo have developed a neuro-computing approach to solve the Riccati equation in real time. After the motion base trajectory is computed, the simulator motion commands are then transformed from the degree-of-freedom space to actuator space (kinematic transformation), and the desired commands to the six actuators are generated. A block diagram illustrating the function of the motion system is given in Fig. 2.12[8].

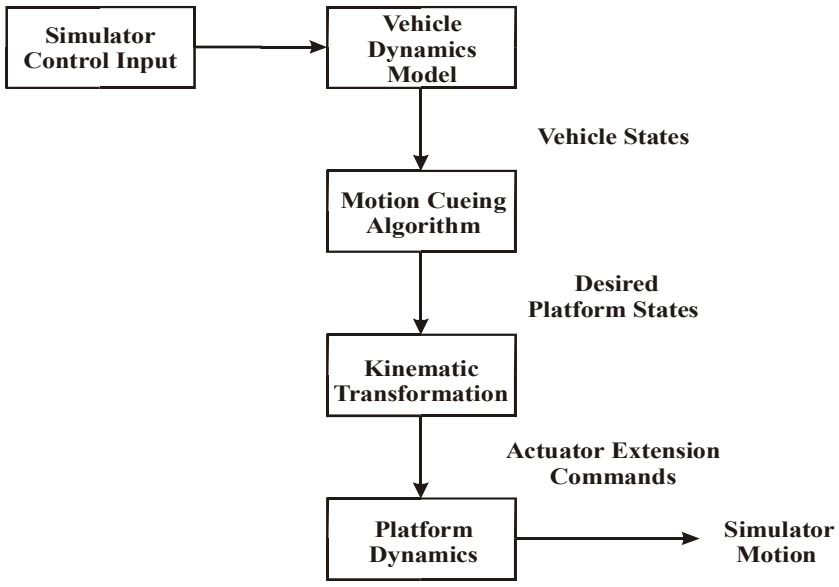


Fig. 2.12. Block diagram of a motion cueing system (courtesy of Telban)

The motion cueing algorithm and the kinematic transformation may take as long as one frame to complete, and the D/A conversion takes half a frame on average. Therefore, the motion system can cause as much as one and half frames of transport delay, although, usually the motion cueing algorithm is executed in the same frame as the flight dynamics.

C. Visual Delay

The visual system consists of four parts in series: the front-end computer (FEC), the geometry processor (GP), the display processor (DP) and the display. The FEC takes care of the scene management and retrieving the image from the database; the geometry processor calculates the geometric transformations and the perspective transformation; and the display processor pixelates the data. Each of these three may consume one full frame. The fourth part (display) consumes a half frame on average, with the worst case being a full frame. Fig. 2.13 illustrates two scenarios of how these four steps in the pipeline may be implemented. In the worst case they work serially without overlap, causing 4 frames of transport delay; in the best case, parallel computation is employed between the FEC, GP and the DP, reducing the visual delay by one frame.

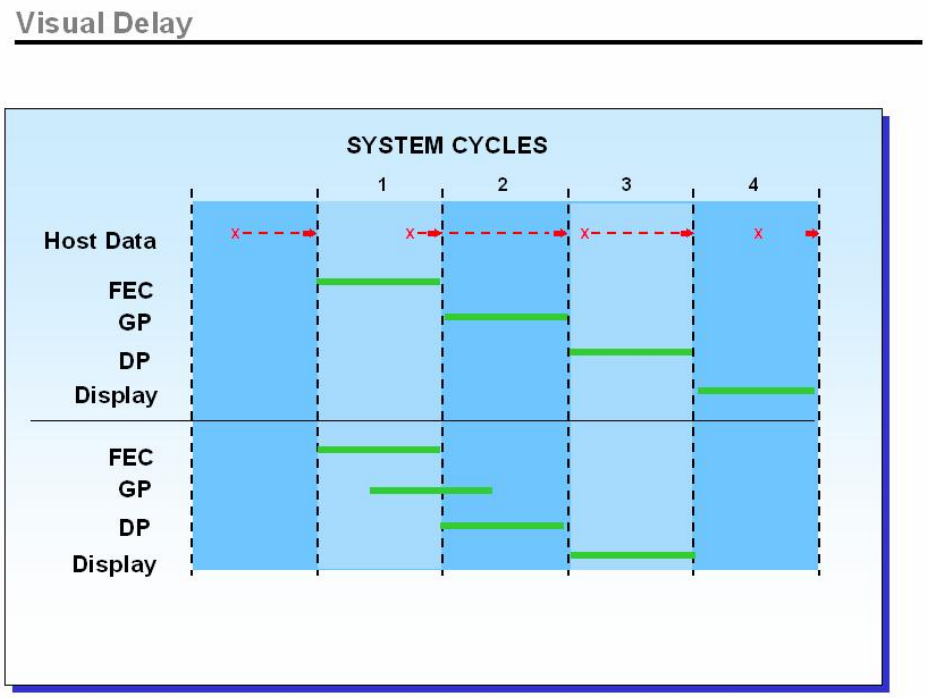


Fig. 2.13. Two scenarios for delay in the visual system

D. Cueing Mismatches

The previous analysis shows that the instrument system causes the shortest transport delay, the motion system the medium, and visual system the longest. The discrepancy in the delay between the cueing systems is referred to as cueing mismatch. Fig. 2.14 illustrates the cueing mismatch between the motion and the visual systems. Frame 1 is the initial state; in frame 2, the pilot initiates a control input, but neither the motion nor the display changes; in frame 3, the response of the dynamic seat begins, but the display still remains unchanged; and finally, in the 4th frame, the response of visual display begins. Though the research is not clear on the maximum mismatch a human can tolerate in a flight simulator, the cueing mismatch is believed to be the main cause of simulator sickness. Therefore, steps to sufficiently reduce the mismatch are necessary.

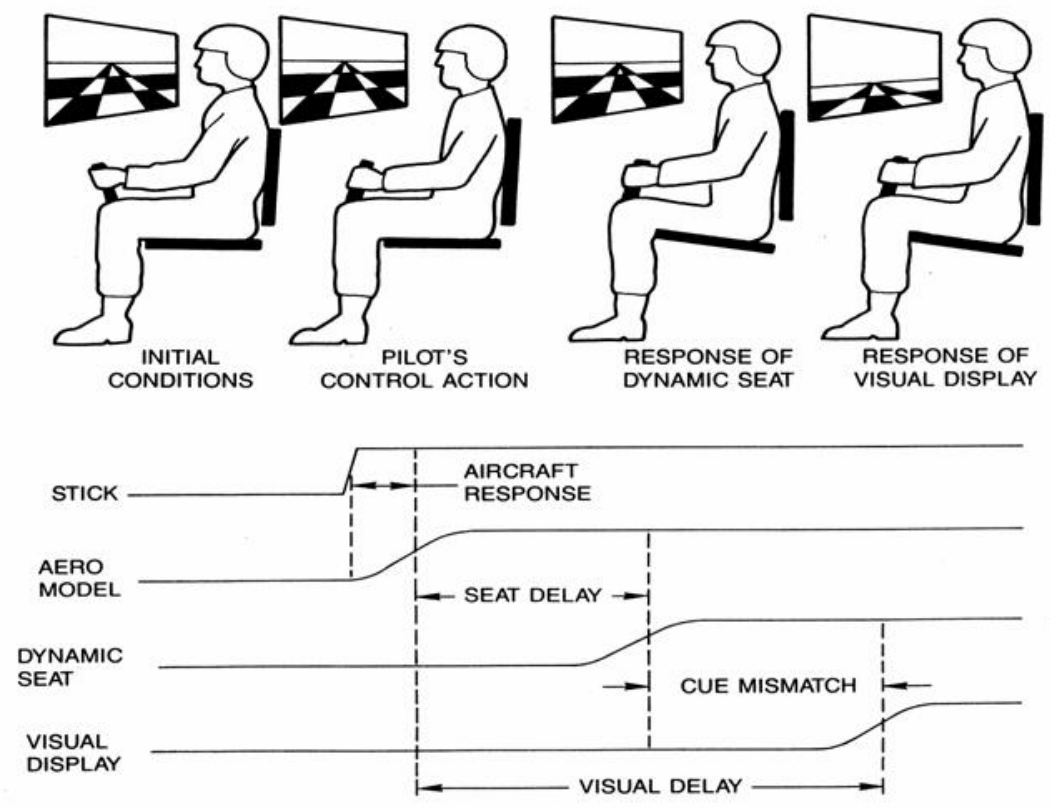


Fig. 2.14. Cueing mismatch between the motion and the visual systems

2.2.5. Summary

Therefore, a typical flight simulator may incur, on average, 1.5 frames of transport delay in the simulation computer, and 3.5 frames of delay in the visual system. Assuming the two subsystems have the same frame length, and assuming an average 0.5 frames of data transfer delay, the average total transport delay is then 5.5 frames, and may vary from 4 to 7 frames.

If the transfers are asynchronous, the data transfer delay affects the sampling delay. As long as the transfer time is less than the sampling interval (i. e., the frame length), transfer time may be considered to be the same as processing time. Although the simulator time delay consists of several components from different subsystems, they make no difference to the operator, who only feels the total effect.

2.3. Effects of Transport Delay

It follows from the theoretical analyses on time delay in section 2.1 that time delay introduces oscillation to the closed-loop control system. In a man-machine system such as a flight simulator, this causes pilot induced oscillation (PIO). Increased oscillation means more and larger swings in response, which also mean larger variance of system output and degraded simulation performance. Conversely, the system output is the feedback on which the operator action is based. Increased output variance makes the operator compensate for the error, and this compensation causes the operator workload to go up. And for this reason, the operator's perception of handling quality is degraded. Therefore, three effects of time delay on the flight simulation are expected:

1) The system performance is degraded;
2) The operator's workload is increased;

3) The operator's perception of handling quality is decreased.

The three effects of transport delay are usually evaluated with three different metrics: namely, the Root Mean Square Error (RMSE) for the system performance, the Power Spectral Density (PSD) for the pilot workload, and the Cooper-Harper Rating (CHR) for the handling quality.

The RMSE is a measure of variance of the simulator output, and is used as an objective metric of the system performance. Larger RMSE means poorer performance. The RSME is usually applied to simulation tracking tasks, such as compensatory tracking, pursuit tracking, or a gliding task.

The power spectrum or power spectral density (PSD) of a digital signal is the discrete Fourier transform (DFT) of the autocorrelation sequence of a signal. In mathematical terms, the PSD is proportional to the square of the magnitude of the process. Hence, it is closely related to the energy of a signal as a function of the frequency, or it represents the frequency distribution of energy of a signal. The PSD of the pilot control input deflection (the pitch stick, roll stick, rudder pedal or throttle) is related to the energy the pilot put into the simulator, often referred to as the workload. It is also an objective metric.

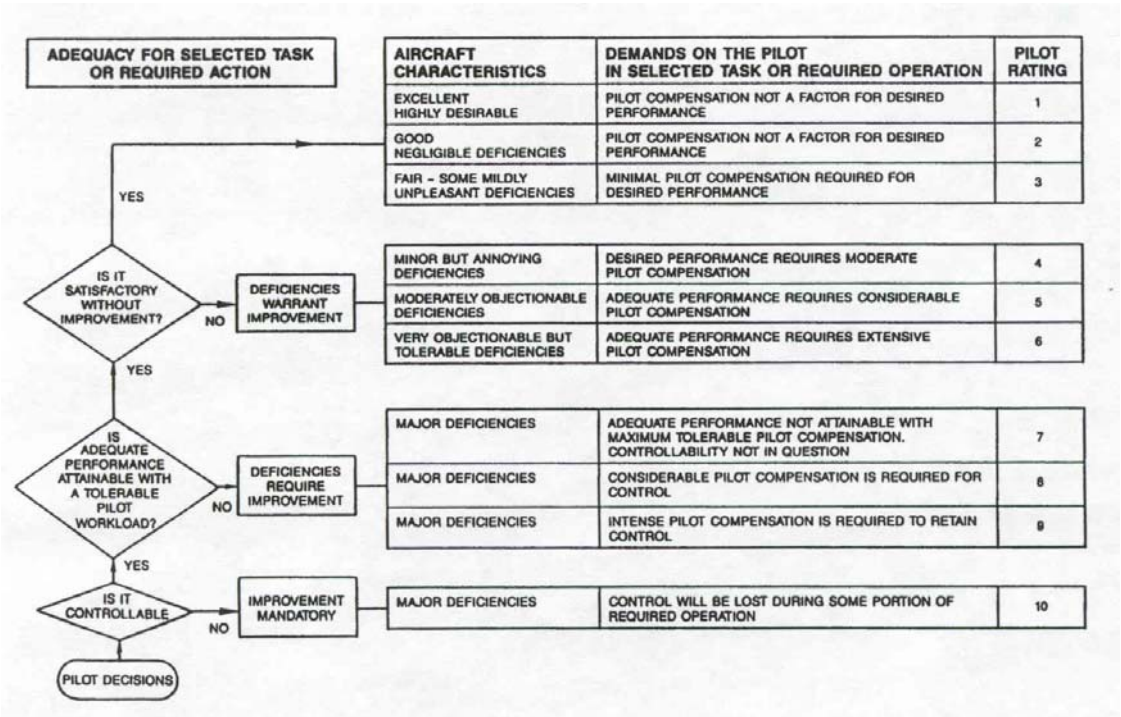
The Cooper-Harper rating scale[9] includes ten values, listed in Table 2.2, in ascending order corresponding to decreasing handling qualities, i.e., a scale value of one represents the best handling quality, and a value of 10 represents the worst deficiency. Ratings of 1, 2 and 3 fall in level I, ratings of 4, 5 and 6 are in level II, and ratings of 7, 8, 9 and 10 are in level III. Jumping from one level to another is considered to be a significant change of handling quality. A CHR is a quasi-objective evaluation because the

operator is required to make a series of decisions concerning the difficulty of controlling the aircraft. The Cooper-Harper rating is an evaluation of the overall system, rather than a single factor such as responsiveness; hence two simulations with close CHRs may have quite different dynamic responses.

In summary, the RMSE is an objective metric of the simulator output, the CHR is a quasi-objective metric from the operator's cognitive judgment, and the PSD is an objective metric of the simulator input (or the operator output).

Many researchers have extensively explored the effects of time delay on a flight simulation using these three metrics. A study of the available literature is summarized as follows.

Table 2.2. The Cooper-Harper scale



By analyzing the altitude errors in different delay conditions, Crane[10] reported that pilot dynamic response and system performance change as the pilot attempts to

compensate for the decrease in system stability caused by the transport delay. More importantly, the changes in pilot dynamic response and the system performance can bias the results of a simulation by influencing the pilot's rating of handling qualities of the simulated aircraft.

Bailey, et al[11], conducted some experiments on a simulator including four simplified aircraft models—fighter, small cargo, medium cargo and large cargo, with pilots flying some demanding tasks in both ground based and in-flight modes. 0-240 ms artificial delay was added over the 100 ms baseline delay. Bailey noted:

1) Except for the fighter in the in-flight mode, the time delay significantly increases the RMSE, or degrades the performance.
2) The time delay causes the regression line of the Cooper-Harper rating to cross the Level I and Level II boundary, indicating significant handling quality degradation.
3) The difference in the effects of time delay among the four aircraft models is small.
4) In the in-flight mode, the negative effects of time delay are not as serious as in the ground-based mode. Because the in-flight mode could be thought of including the motion cue, it suggests that the motion cue makes the operator less sensitive to the time delay.

Similar simulation experiments were conducted by Riccio, et, al[12], in which 0-350 ms time delay was added in addition to a baseline delay of 50 ms, with the pilot controlling a fighter or a large transport aircraft to maintain a constant heading and 100 ft altitude over flat terrain, and with turbulence. In this experiment only the RMSE evaluation was carried out. Both heading RMSE and altitude RMSE increased significantly with the time delay. There was no statistically significant difference between

the two aircraft models, however, the RMSE curve of the transport appeared to have a larger curvature.

Middendorf, et al[13], on the other hand, used the power spectral analysis to investigate the effects of the simulator time delay on flight control activities. The simulation was conducted on a fix-base (no motion cue) simulator. Time delays of 90, 200 and 300 ms were inserted in addition to the 90 ms baseline delay. The lateral stick movement was recorded for a spectral analysis while the pilot was completing an offset approach. For this task, the aircraft was initially lined up with one runway, then transitions to an adjacent parallel runway within a certain distance. They reported the following findings:

1) There was a peak in the power spectrum at approximately 0.08 Hz, which is, as they stated, related to the inverse of the period of the sidestep maneuver.
2) There was another peak at approximately 0.25 Hz, appearing to be a direct result of the maneuver itself.
3) The reduced damping ratio resulting from the time delay made the closed-loop system less stable at high delay conditions.

Cardullo conducted two phases of simulator experiments with his graduate students, Telban and Guo[14], and some colleagues at the NASA Langley Research Center. The experiments were conducted on the Visual Motion Simulator, and they were designed to test novel motion cueing algorithms as well as the effects of time delay and the McFarland compensation algorithm. Artificial delays of 0, 50, 100, and 200 ms were added to the simulation. These delays, combined with the three motion cueing algorithms

—the adaptive, optimal and nonlinear algorithms, generated 16 different conditions. Each pilot was assigned three tasks —a straight-in approach, an offset approach and a takeoff. Half of the takeoff test runs included an engine failure. Three pilots took part in the first phase of the experiment (also referred to as the preliminary experiment), and eleven took part in the second phase. Cooper-Harper ratings were logged only in the preliminary experiments. In both phases of the experiment, the time domain data of the pilot's control deflections and some other simulator state variables were recorded for power spectrum analysis. Although most of the results are outside the scope of this section, the results on the effects of time delay and compensation are summarized here.

1) The time delay slightly increased the overall integrated PSD of the roll control, pitch control and rudder pedal control, and significantly increased the integrated PSD of the roll and pitch controls in the frequency range of [0.17 0.4] Hz.
2) The time delay moved the highest peak of the PSD plot to higher frequency area. The effects of the engine failure in the takeoff maneuver dominated the total PSD of the pilot control inputs, obscuring the effects of the time delay on PSD.
3) The Cooper-Harper ratings did not increase with time delays of up to 100 ms, in the two approach maneuvers.

Transport delay may also cause simulation sickness. The term "Simulation Sickness" is usually reserved for situations that are nauseogenic in the simulator but not in the corresponding aircraft. Simulation Sickness shares many symptoms, such as drowsiness, dizziness nausea, vomiting, etc, with the motion sickness that occurrs in the real systems. It is difficult to conduct simulator sickness research because it is difficult to objectively evaluate it. Literature research suggests that cue mismatch is the primary

cause of simulator sickness. Duh, et al, recently proposed a visual-vestibular crossover frequency concept, and hypothesized that conflicting visual and inertial self-motion cues at the frequency of maximum crossover would be more likely to evoke simulator sickness than conflicting cues at a high frequency. This hypothesis is supported by experiments. Sobiski states, "Temporal cue mismatch can contribute to a malaise known as 'simulator sickness' and it may be due to the frantic efforts of the brain to resolve two conflicting sources of motion related information." Because the cue mismatch is caused by the difference in time delay, and simulator sickness is induced by the cue mismatch, it may be inferred that simulator sickness is indirectly caused by the difference in time delay. It seems logical that time delay may cause simulator sickness. Long time exposure to PIO caused by time delay is expected to make the operator feel tired and to induce some instantaneous or simulator sickness. However, to date, the direct relationship between time delay and simulator sickness is not completely understood, probably because there has been insufficient research on this issue. Zaychik [15] reported that Nelson et al, investigated the influence of time delay, time on task and task complexity on subjective ratings of simulator sickness. It was shown that the subjective ratings varied directly with duration of task. The delay variance affected the performance and workload of a subject but had no effect on the simulator sickness questionnaire (SSQ) ratings. Zaychik suggests that this interesting observation can be attributed to the subjective nature of the simulator sickness assessing technique used by the researchers.

Likewise, Uliano, et al[16], reported that visual lag had no effect on illness in their experiments. Although longer lags were somewhat disruptive of performance, there was no evidence that they contributed to illness. These results are surprising because the range

of asynchrony studied was fairly large (126 ± 17 ms to 215 ± 70 ms). Based on these results, Uliano et al suggested that within the range of operationally useful simulators, visual asynchrony does not appreciably contribute to simulator sickness. Lags approaching 300 ms in flight simulators become too unrealistic and/or too difficult to fly.

3. Current Techniques of Compensating Transport Delay

As stated in Chapter 2 (section 2.3), transport delay in a flight simulator reduces the system phase margin, and can cause pilot-induced oscillation (PIO), which in turn leads to poorer system performance and handling quality, higher control workload, and even simulator sickness. Transport delay may be reduced, by rearranging the order of the calculations, choosing more efficient algorithms, increasing the sampling rate, or synchronizing the communications. If the delay is still exceeds the tolerable level, for maintaining desirable simulator performance, algorithms to compensate for the delay may be employed. The purpose of compensation of time delay, from a time-domain perspective, is to provide prediction to counteract the system latency; and in terms of the frequency-domain, is to create phase lead to offset the reduction in system phase margin.

When transport delay exists in a visual system, the image that is displayed is actually the delayed image representing a past aircraft state. Since one cannot generate an undelayed image from a delayed one, compensation must be applied to the aircraft state. Therefore, the idea of compensation is to predict the future aircraft state using the currently available state information. Therefore, an image, based on the predicted aircraft states, can be used to offset the transport delay in the visual system. This idea is illustrated in Fig. 1.4, where the dashed line in the small plot to the right of the predictor block, illustrates the prediction.

The dashed predicted curve in Fig. 1.4 is an ideal prediction, i.e., it has exactly the same shape as the original solid one, with only a pure phase shift. Unfortunately, an ideal predictor does not exist in the real world. A practical predictor generally introduces error,

and the longer the time delay, the greater the error. Mathematical analysis illustrates that pure time delay brings phase lag to the system without changing the magnitude (Section 2.1). Therefore, a good predictor must satisfy two basic criteria:

1) It must be able to provide sufficient phase lead to offset the phase lag caused by the time delay.
2) It must introduce minimum gain distortion.

In addition, the compensation must be simple enough that it does not introduce extra time delay due to computation. Therefore, a third criterion is:

3) The computation workload of the predictor must be minimal.

To date, many compensation techniques have been developed to mitigate the transport delay in the flight simulator. The lead/lag filter, the McFarland compensator and the Sobiski/Cardullo state space predictor are the three most prominent current techniques, and will be considered here.

3.1. Lead/lag Filter

The transfer function of the lead/lag filter is given by

$$\frac{Y_p(s)}{Y(s)} = k\frac{s+\omega_n}{s+\omega_d}, (\omega_n < \omega_d), \text{ or} \tag{3.1}$$

$$\frac{Y_p(s)}{Y(s)} = k\frac{\tau_n s+1}{\tau_d s+1} = k\frac{\tau_n s+1}{\alpha\tau_n s+1}, (0<\alpha<1) \tag{3.2}$$

where $Y_p(s)$ and $Y(s)$ are the Laplace transforms of the predicted aircraft state and the undelayed aircraft state, respectively; and ω_n and ω_d are the two corner frequencies of the filter. If $\omega_n < \omega_d$, the Bode asymptotes of both the magnitude and the phase of the filter given by Eq. (3.1) are illustrated in Fig.3.1.

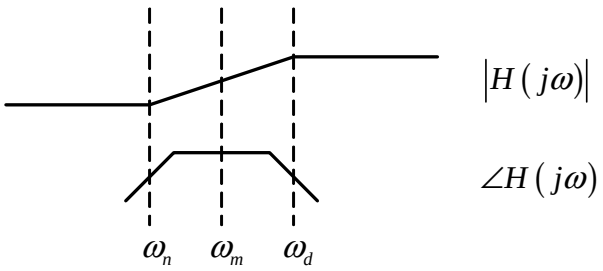


Fig. 3.1. Bode asymptotes of the lead/lag filter

The hump of phase asymptote in $[\omega_n\ \omega_d]$ provides the phase lead to the system, and the maximal phase lead occurs at a frequency ω_m called the medium frequency, which is the geometric mean of the two corner frequencies as given by

$$\omega_m = \sqrt{\omega_n \omega_d} \tag{3.3}$$

and the maximal phase lead is

$$\sin\phi = \frac{1-\alpha}{1+\alpha} \tag{3.4}$$

This phase lead is obtained at the expense of gain distortion because the magnitude $|H(j\omega)|$ is not unity. And, since the high-frequency gains are increased, any system using phase lead compensation may be subjected to high-frequency noise problems. Nevertheless, as Crane[10] states, the resulting increase in system gain at frequencies greater than the crossover frequency (ω_c) is not normally a problem, because the system amplitude ratio and the power of the input and disturbance signal usually decrease rapidly at frequencies greater than ω_c.

Designing a lead/lag filter involves choosing the gain *k* and the two corner frequencies ω_n and ω_d (or the two time constants τ_n and τ_d). The lead/lag filter had long been used in other control systems to provide phase modification before it was first

applied in a flight simulator. Using the classical frequency-domain design method can meet the design specifications accurately. However, because the classical method assumes linearity, and the aerodynamic models of a flight simulator are usually non-linear, the classical approach does not lend itself to the design of a lead/lag filter for compensating the transport delay in the flight simulator.

Because the pilot crossover frequency region has been shown to be the most critical for pilot control, and for pilot ratings (Fig. 3.2[17]) of the fidelity of a dynamic simulation, an ideal design would be to make the medium frequency ω_m, the frequency at which the maximal phase lead occurs, at the pilot crossover frequency ω_c. Unfortunately, ω_c is usually unknown. One approach would be to assume an estimated $\hat{\omega}_c$, and let $\omega_m = \hat{\omega}_c$. Then calculate the maximal phase lead necessary to counteract the phase lag caused by the delay t_d

$$\phi = t_d \hat{\omega}_c \tag{3.5}$$

Then calculate the ratio α according to Eq. (3.4), and finally determine the two time constants. Though simple, this design approach has two problems. First, the estimated $\hat{\omega}_c$ is not the real crossover frequency ω_c, which varies in a simulation due to many factors. The further $\hat{\omega}_c$ (i.e., ω_m) departs from ω_c, the less phase lead at ω_c compared to the maximal lead at ω_m. In other words, the phase compensation at the crossover frequency is not sufficient. Second, since $0 < \alpha < 1$, it follows from Eq. (3.4), that ϕ must be less than $\pi/2$. Then, according to Eq. (3.5), for a long delay t_d, the estimated $\hat{\omega}_c$ must be very small, and hence it also diverges from ω_c, resulting in insufficient phase

compensation. This explains why the lead/lag filter cannot be used to compensate long time delay.

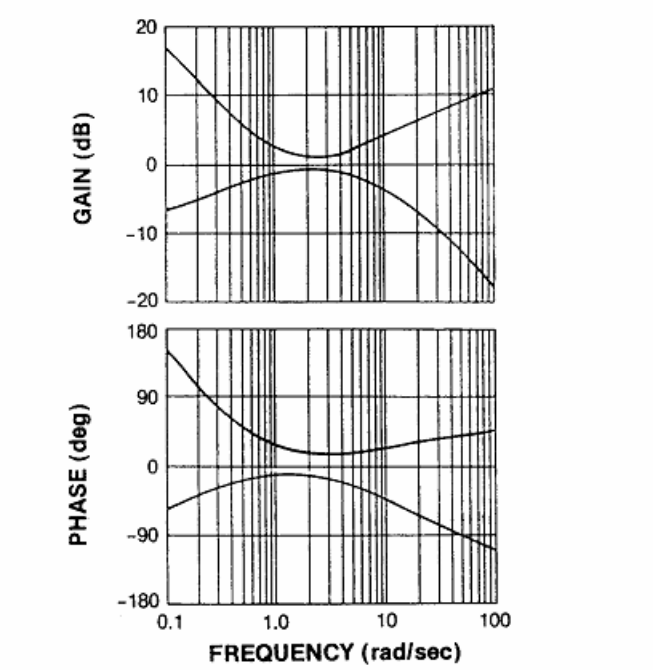


Fig. 3.2. Pilot sensitivity envelops in the frequency domain

Therefore, the design of a lead/lag filter to compensate the time delay in a flight simulator with unknown or nonlinear aerodynamic model cannot be exact, and therefore approximate methods are usually used. Two such approximate methods were introduced: one by Crane, and the other by Ricard and Harris[18]. In both methods, the numerator time constant τ_n is set to be equal to the time delay t_d, whereas the denominator time constant τ_d is chosen in a different manner. In Crane's approach, τ_d is the solution of the following equation

$$\Phi_f\Big|_{\omega=\omega_c} = tan^{-1}(\omega_c t_d) - tan^{-1}(\omega_c \tau_d) = \omega_c t_d \tag{3.6}$$

The gain k can be chosen so that the filter gain is unity at the crossover frequency. Eq. (3.6) then becomes problematic because, for the widely-accepted value of ω_c of 2-3 Hz

and t_d up to 800 ms, the term $tan^{-1}(\omega_c t_d)$ is always less than $\omega_c t_d$. Hence the term $tan^{-1}(\omega_c \tau_d)$ is negative. Any τ_d that makes $tan^{-1}(\omega_c \tau_d)$ negative is either negative (meaningless) or greater than τ_n (in this case, the filter becomes a phase lag filter). As stated previously, the actual phase lead of the filter is usually less than the maximum, and one way to resolve this difficulty is to multiply the right side of Eq. (3.6) by a coefficient, thereby changing it to

$$\Phi_f\Big|_{\omega=\omega_c} = tan^{-1}(\omega_c t_d) - tan^{-1}(\omega_c \tau_d) = \eta \omega_c t_d, (0 < \eta < 1) \tag{3.7}$$

where η is dependent upon the crossover frequency and the time delay. For example, for a crossover frequency of 2 rad/s and time delay up to 800 ms, η=0.5 produces very good results. Table 3.1 gives the design results obtained using the modified Crane method for time delays of 200, 400 and 800 ms and an estimated crossover frequency of 2 rad/s.

Table 3.1. The lead/lag filter coefficients designed with Crane's method

t_d (ms)	τ_n	τ_d	ω_m (rad/s)	Φ_m (deg)
200	0.2	0.0912	7.4044	21.94
400	0.4	0.1410	4.2108	28.60
800	0.8	0.1080	3.4021	49.65

In Ricard's approach, τ_d is chosen such that the best pilot flight control performance is achieved. Using Ricard's method, the two time constants for lead/lag filters used to compensate for delays of 200, 400 and 800 ms are listed in Table 3.2 (obtained from Sobiski[19]). Column 4 gives the frequencies at which these filters provide the maximal phase lead. The maximal phase lead is listed in column 5. Assuming the crossover frequency of the flight simulator system is 2 rad/s, the decrease in phase

margin caused by time delays of 200, 400 and 800 ms are 22.9°, 45.8° and 91.7°, respectively. These values are greater than the corresponding maximal phase lead of the lead/lag filter (column 5 in Table 3.1). This shows that none of these filters provide sufficient compensation. Although other design methods may yield better compensation, the improvement is limited. The lead/lag filter simply lacks the ability to provide large phase lead.

Table 3.2. The lead/lag filter coefficients designed with Ricard's method

t_d (ms)	τ_n (s)	τ_d (s)	ω_m (rad/s)	ϕ_m (deg)
200	0.2	0.1859	5.2	2
400	0.4	0.2105	3.5	18
800	0.8	0.1695	2.5	40

Reguarding the effects of the lead/lag compensation, Crane reported, "The compensation is effective; improvements in pilot performance and workload or HQR were observed. The delay compensation approach attempts to minimize changes in pilot-aircraft dynamics in the region of the crossover frequency."

Ricard and Harris reported that there were effects due to the presence of the lead/lag filter, and the error score indicated that varying τ_d was significant, but not significant for the crossover power. However, Ricard and Harris did not reveal whether the pilot performance, similar to that in the undelayed system, was achieved when using lead/lag compensation.

To test the effectiveness of the lead/lag filter in compensating the transport delay, a lead/lag compensator was added to the control system shown in Fig. 2.3 in front of the added time delay. The new system block diagram is given as Fig. 3.3, where y is the

undelayed aircraft state, y_p is the predicted aircraft state and y_c is the delayed prediction, or the compensated state. The added time delay is 200, 400 or 800 ms, and for each delay case, both the lead/lag filters designed with the modified Crane method and the Ricard method (see Table 3.1 and Table 3.2) were applied. The frequency responses and step responses are given in Fig. 3.4 and Fig. 3.5.

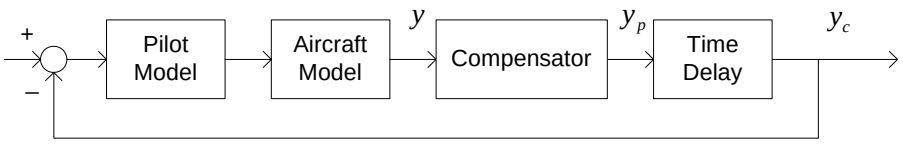


Fig. 3.3. Block diagram of a delayed control system with a compensator

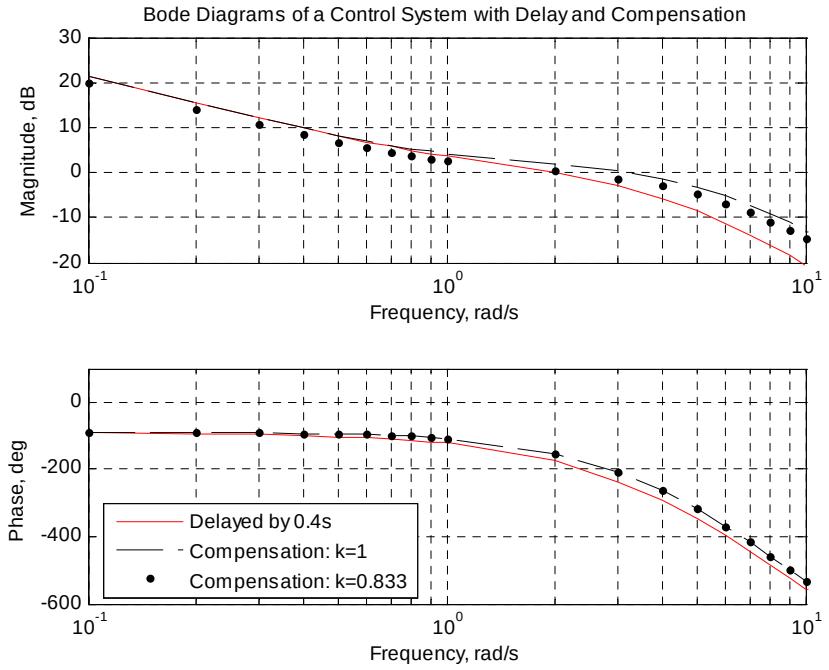


Fig. 3.4. Bode diagrams with delay (400 ms) and with/without lead/lag compensation of different gains

Fig 3.4 shows the 400 ms delay compensation using the modified Crane design. When the filter gain is unity (dashed curve), it provides 20° phase lead at 2 rad/s. But because of the gain distortion, the crossover frequency is shifted to 3 rad/s, corresponding

to a phase margin of -38.44°, which is even lower than the delayed system without compensation. To move the crossover frequency back to 2 rad/s, the pilot can decrease his gain, or the filter designer can decrease the lead/lag filter gain from unity to 0.833 (dotted curve). The resulting lead/lag filter provides a phase lead of 12.43°, which is equivalent to a 108 ms delay, and much less than desired 400 ms.

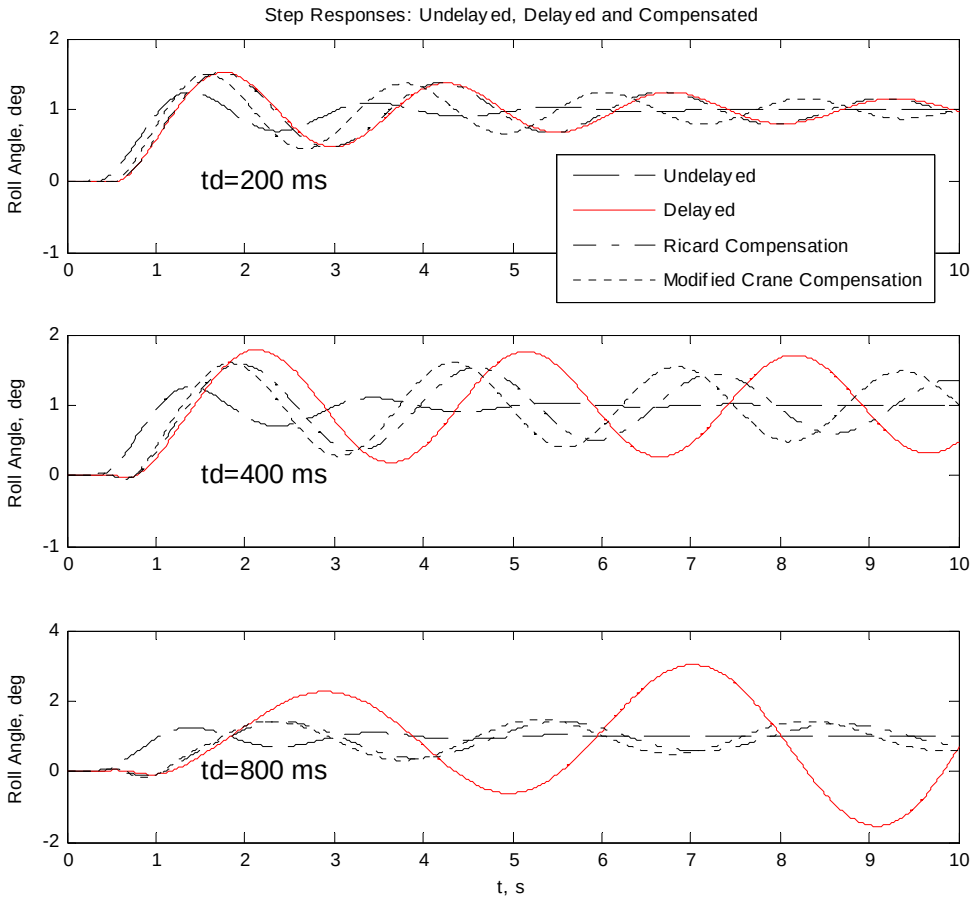


Fig. 3.5. Step responses of the undelayed, delayed and compensated systems

From the step responses, a lead/lag filter designed using Crane's method achieves better results than one designed with Ricard's method in all delay cases. The results are greatly improved for a delay of 200 ms, in which Ricard's filter provides little phase lead.

In all cases, a lead/lag filter cannot compensate the entire delay: no matter how long the delay is. The lead/lag filter is only beneficial when used to compensate for delays of less than approximately 100 ms. This should be considered the upper limit of the lead/lag filter.

In short, although the lead/lag compensator does meet the third criterion of a good compensator, it does not meet the first two criteria..

The pulse transfer function and the difference equation of the lead/lag filter are given by Eq. (3.8) and (3.9), respectively

$$\frac{Y_p(z)}{Y(z)} = \frac{z + c_n}{z + c_d} \tag{3.8}$$

$$y_p(k+1) = -c_d y_p(k) + y(k+1) + c_n y(k) \tag{3.9}$$

As an Infinite Impulse Response (IIR) filter (since $c_d \neq 0$), the lead/lag compensator makes use of the previous prediction to calculate the current prediction, thus the error from one iteration is passed on to the next iteration, resulting in error accumulation. This error accumulation is the primary cause of the gain distortion.

3.2. McFarland Filter

Richard McFarland[20] developed a Finite Impulse Response (FIR) filter to avoid the weaknesses of the lead/lag filter that cause the prediction error to accumulate. Its pulse transfer function and difference equations are given by

$$Y_p(z) = Y(z) + \left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right) V(z) \tag{3.10}$$

$$y_p(k) = y(k) + b_0 v(k) + b_1 v(k-1) + b_2 v(k-2) \tag{3.11}$$

where y is the aircraft state to be predicted, v is the corresponding velocity, y_p is the predicted aircraft state, k is the iteration index, and $Y_p(z)$, $Y(z)$ and $V(z)$ are the z-transforms of y_p, y and v, respectively. Clearly, the McFarland compensator is a special integrator making use of three consecutive steps of velocity. The three coefficients b_0, b_1 and b_2 determine its ability to compensate the time delay. McFarland uses a method known as sinusoidal tuning to determine these three coefficients, which are the solutions to three equations derived from the boundary conditions of the pass band [0 ω_0] by assuming a sinusoidal input to the filter. The pass band comes from the assumption that the pilot operates primarily within this frequency interval and the operation beyond ω_0, about 6-20 rad/s, is insignificant. At the zero frequency condition, the velocity is constant, and Eq. (3.11) becomes $y_p(k) = y(k) + (b_0 + b_1 + b_2)v(k)$. For an ideal prediction of $y_p(k)$ that is t_d ahead of $y(k)$, the relation $y_p(k) = y(k) + v(k)t_d$ holds. Comparing this with the previous equation, and the first equation from the boundary conditions is obtained

$$b_0 + b_1 + b_2 = t_d \tag{3.12}$$

In addition, by substituting the relations $Y_p(z) = z^{\frac{t_d}{T}} Y(z)$ and $Y(z) = \frac{T}{2}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)V(z)$ (trapezoidal integration) into Eq. (3.10), the relationship becomes

$$\frac{Y(z)}{V(z)} = z^{-\frac{t_d}{T}}\left\{\frac{T}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right) + \left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right)\right\} \tag{3.13}$$

Let $Y(z)/V(z)$ be equal to the ideal integration $1/(j\omega)$ at the other boundary frequency ω_0, and the magnitude and phase angle equalities give the remaining two equations. Combining these with Eq. (3.13), and the three coefficients are given as follows, where $\theta_0 = \omega_0 T$ and $\psi_0 = \omega_0 t_d$.

$$b_0 = \frac{\left[\psi_0 + \sin\psi_0 (1 - 2\cos\theta_0)\right]\sin\theta_0 + \left[\tfrac{1}{2}\theta_0 \sin\theta_0 - \cos\psi_0 (1 - \cos\theta_0)\right](1 + 2\cos\theta_0)}{2\omega_0 \sin\theta_0 (1 - \cos\theta_0)} \quad (3.14)$$

$$b_1 = \frac{2\sin(\theta_0 + \psi_0) - 2\psi_0 \cos\theta_0 - \theta_0 (1 + \cos\theta_0)}{2\omega_0 (1 - \cos\theta_0)} \quad (3.15)$$

$$b_2 = \frac{\left[\psi_0 - \sin\psi_0 + \tfrac{1}{2}\theta_0\right]\sin\theta_0 - \cos\psi_0 (1 - \cos\theta_0)}{2\omega_0 \sin\theta_0 (1 - \cos\theta_0)} \quad (3.16)$$

McFarland states that "the algorithm delivers high-fidelity, compensated CGI drive signals over the human-factors bandwidth, and can dramatically improve the pilot control for high gain tasks such as precision hovering and station keeping." However, he does not show experimental results to substantiate it. A system similar to the one depicted in Fig. 3.3 was used to test the McFarland compensator. In this test, the lead/lag filter was replaced with the McFarland filter. The system was then transformed into a discrete form before the compensator was applied because the McFarland filter was in discrete format. The frequency responses and step responses are given in Fig. 3.6 and Fig. 3.7, respectively.

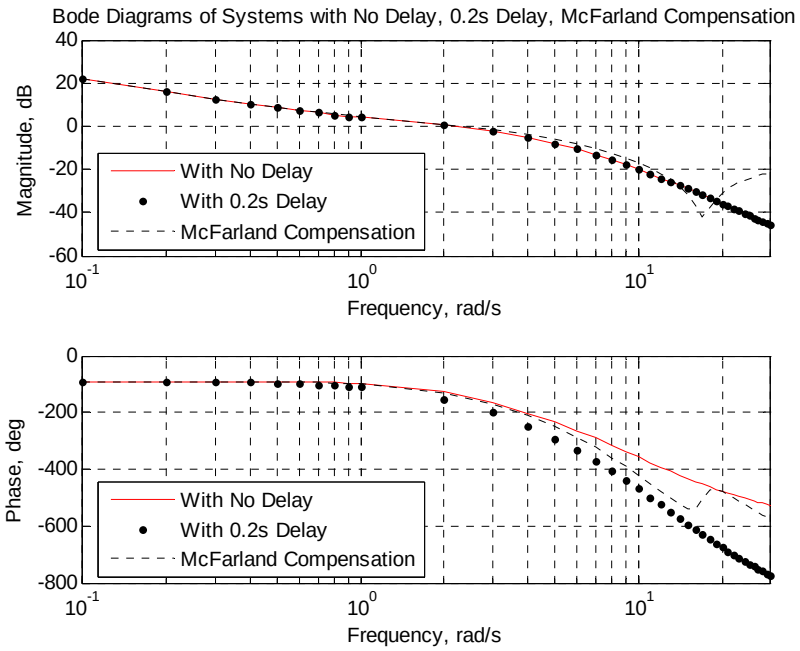


Fig. 3.6. Bode diagram of McFarland compensation for delay of 200 ms

The Bode diagram shows that for time delay of 200 ms, the McFarland filter provides satisfactory phase compensation when the frequency is below 5 rad/s, but the phase lead is not sufficient in higher frequencies. The gain distortion is small when the frequency is within the pass band, but the gain distortion escalates when the frequency is beyond the pass band. For comparison, the compensation by the lead/lag filter designed with the modified Crane's method is also plotted in Fig. 3.7. This plot shows that the McFarland compensator can provide more phase lead than the lead/lag filter, however, the McFarland filter still cannot provide 100 percent compensation.

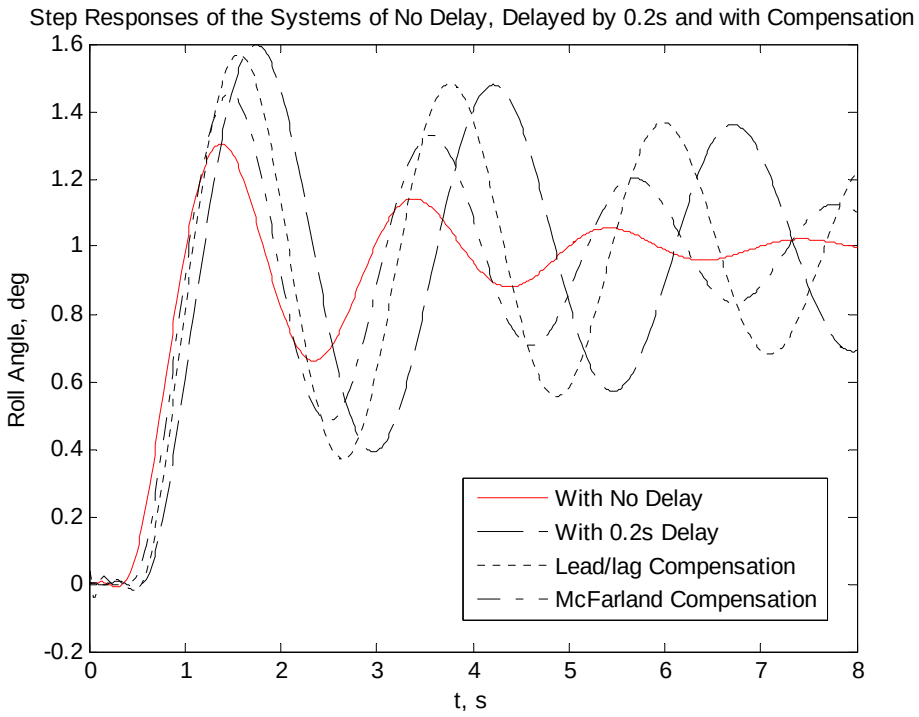


Fig. 3.7. Undelayed, delayed by 0.2s and compensated responses

The McFarland predictor can also be applied to simulation data. Fig. 3.8 shows a time history of roll angle and roll velocity recorded on the Visual Motion Simulator at the NASA Langley Research Center. By using the roll angle as y and the velocity as v, and with a prediction time of 192 ms, the McFarland compensator described by Eq. (3.11) produced the dashed curve shown in Fig. 3.9, this is the signal y_p in Fig. 3.3. The prediction is then delayed by the same amount for ease of comparision with the uncompensated roll angle. For ideal compensation, the delayed, compensated curve (dashed line) should be exactly the same as the undelayed uncompensated curve (solid line). However, this plot shows that the McFarland prediction has two problems: the actual prediction is only 176 ms, less than the design prediction by almost a frame (16.5 ms), and more seriously, it causes very large spikes.

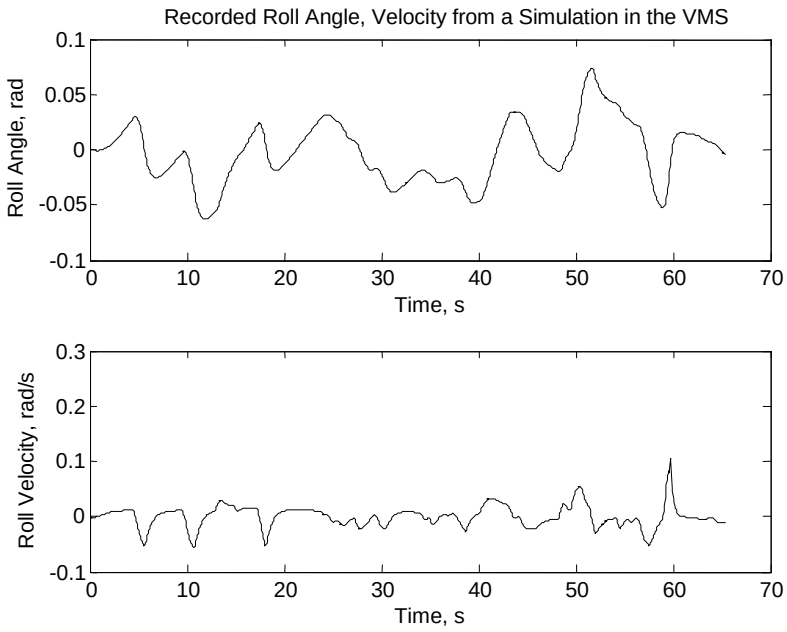


Fig. 3.8. Roll angle and velocity of a real simulation

The gain distortion and insufficient phase lead at high frequencies, demonstrated here, were verified by the performance of the pilots in the simulations conducted in the VMS at the NASA Langley Research Center in 2002. Fig. 3.10 shows the power spectral densities of the roll control averaged across the pilots for both delayed and compensated (McFarland compensation) cases in an offset landing maneuver with 200 ms artificial delay added to the visual system. The McFarland compensator significantly reduces the PSD in the frequency range of [0.5 2.3] rad/s, whereas it increases the PSD in the range of [2.3 4.4] rad/s. In addition, the Cooper-Harper ratings indicate that for long added delay (200 ms), the pilot handling qualities tend to be degraded by the McFarland compensation.

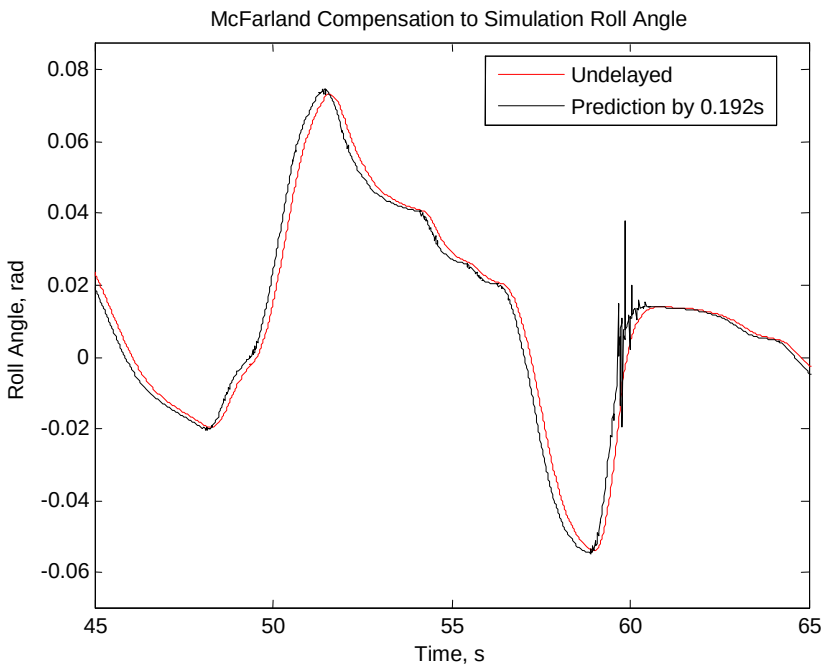


Fig. 3.9. Prediction by the McFarland filter of the simulation data

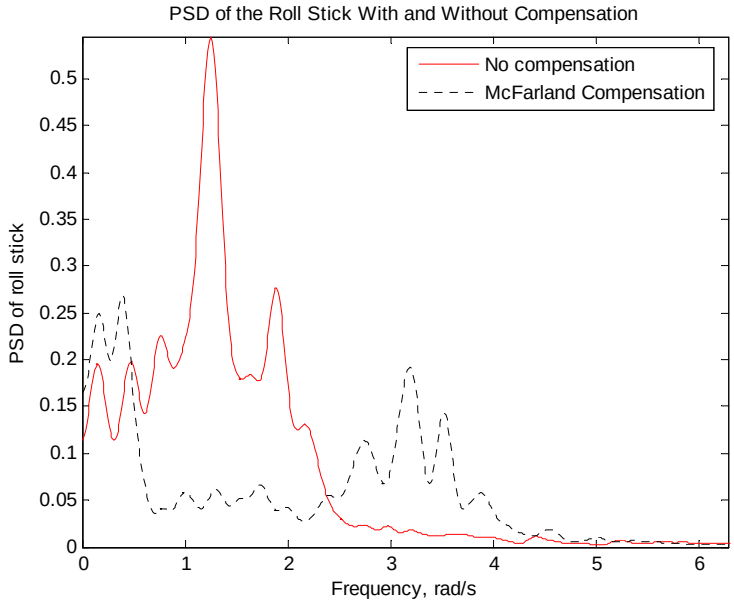


Fig. 3.10. PSD of the roll stick with and without McFarland compensation

Apparently, the biggest problem with the McFarland filter is the annoying spikes it causes. The longer the delay is, the larger the magnitude of the spikes. The McFarland

algorithm is a special integrator or an extrapolator. The simplest extrapolator to provide a prediction of t_d is

$$y_p(k) = y(k) + t_d v(k) \tag{3.17}$$

If the velocity changes slowly (low frequency), this works well, but if the velocity changes quickly, the prediction introduces error because the velocity may be quite different t_d later. The McFarland filter is superior to the extrapolator given by Eq. (3.17) because it uses three consecutive steps of velocities that can extrapolate the future velocity better than a single velocity. For moderate frequencies (around 1 Hz), let the average of these three velocities be $\bar{v}$, then (3.11) reduces to $y_p(k) = y(k) + t_d \bar{v}(k)$, which is similar to Eq. (3.17), but the average velocity is used, resulting in a better prediction. Therefore, its working frequency range is wider than the pure extrapolator. However, if the velocity changes abruptly (the frequency is even higher), such as the velocity plotted in Fig. 3.8 near time = 60 seconds, spikes occur. No matter what the delay is, the three coefficients b_0, b_1 and b_2 are always positive, negative and positive, respectively, and the absolute value of b_1 is always the largest, e.g., for t_d =0.2s, b_0 =2.9979, b_1 =-5.5197 and b_2 =2.7219. The absolute values of the coefficients are at least 10 times greater than t_d. If the velocity changes by more than 10 percent in several iterations, spikes are likely to occur. Table 3.3 gives an example, where the two spikes are highlighted. From this example, it can be shown that the spikes from the McFarland compensation are caused by: first, the three coefficients change sign alternatively, and second, they do not change value in relation to changing simulation conditions. A better choice for the three coefficients would be $b_0 = b_1 = b_2 = t_d/3$, however, if the coefficients

could be made to change with the simulation conditions, the prediction could be even better.

Table 3.3. Several iterations of McFarland prediction with spikes

t (s)	v_0 (rad/s)	v_1 (rad/s)	v_2 (rad/s)	y_p (rad)
59.776	0.1000	0.1015	0.1029	0.0115
59.792	**0.0877**	**0.1000**	**0.1015**	**-0.02**
59.808	0.0754	0.0877	0.1000	0.0085
59.824	0.0630	0.0754	0.0877	0.0072
59.840	0.0507	0.0630	0.0754	0.0056
59.856	**0.0504**	**0.0507**	**0.0630**	**0.0402**
59.872	0.0502	0.0504	0.0507	0.0080

3.3. Sobiski/Cardullo Filter

In 1987, Sobiski and Cardullo[21] proposed a state space predictor for compensating the transport delay. It is based on the equation

$$\mathbf{x}(t+t_d) = e^{\mathbf{A}t_d}\mathbf{x}(t) + \int_0^{t_d} e^{\mathbf{A}(t_d-\tau)}\mathbf{B}u(t+\tau)\,d\tau \tag{3.18}$$

which is derived from the solution of the state space differential equation $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$. This equation shows that the predicted state vector $\mathbf{x}(t+t_d)$ may be calculated from the current state vector $\mathbf{x}(t)$ provided that the future input u is known between t and $t+t_d$. Unfortunately, this is an obviously impossible condition with stochastic operator's control input u. Therefore, Sobiski made some assumptions about the form that the input might take, i.e., piece-wise constant, sinusoidal, exponentially decaying, etc, so that the future input may be approximated by the current input. Then the prediction is given by

$$\mathbf{x}(t+t_d) = \left[e^{\mathbf{A}t_d} \right] \mathbf{x}(t) + \left[\int_0^{t_d} e^{\mathbf{A}(t_d-\tau)} d\tau \right] \mathbf{B}u(t) \tag{3.19}$$

By denoting

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} \tag{3.20}$$

and

$$\mathbf{\Psi} = \int_0^{t_d} e^{\mathbf{A}(t_d-\tau)} d\tau \tag{3.21}$$

Eq. (3.19) is simplified to

$$\mathbf{x}_p = \mathbf{\Phi x} + \mathbf{\Psi B}u \tag{3.22}$$

$\mathbf{\Phi}$ and $\mathbf{\Psi}$ are called the state transition matrix and the convolution integral matrix, respectively. Assuming the matrix $\mathbf{D}$ is zero (this is true for most control systems), then the predicted output is calculated by

$$y_p = (\mathbf{C\Phi})\mathbf{x} + (\mathbf{C\Psi B})u \tag{3.23}$$

Because $\mathbf{C\Psi B} \neq 0$, the matrix $\mathbf{D}$ of the compensated system is usually not zero even though the matrix $\mathbf{D}$ of the undelayed system is zero. The problem caused by the non-zero $\mathbf{D}$ matrix of the compensated system will be addressed later in this section. Directly from Eq. (3.19), the structure of the Sobiski/Cardullo filter is illustrated in Fig. 3.11. The Sobiski/Cardullo state space predictive filter is an original approach for compensating the time delay. Theoretically it can compensate longer delay than the McFarland compensator because it uses more system information, i.e., the full state vector, though it requires complicated calculations.

The state space filter may be expressed in discrete form. Let the discrete state equation be $\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}u(k)$, where $\mathbf{G} = e^{\mathbf{A}T}$ and $\mathbf{H} = \left(\int_0^T e^{\mathbf{A}\tau} d\tau \right) \mathbf{B}$, where T

is the sampling period. The discrete state space filter that predicts l iterations in the future is then given by Eq. (3.24)

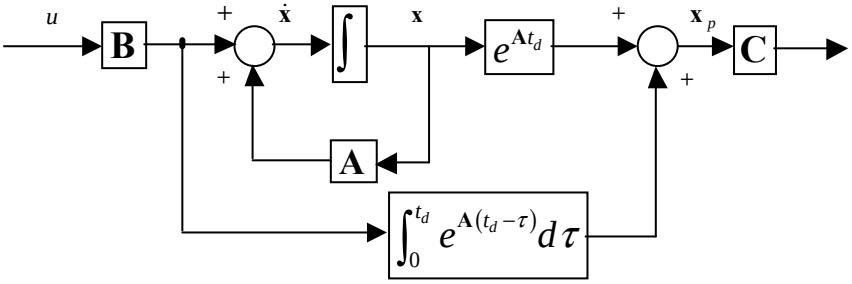


Fig. 3.11. Sobiski/Cardullo compensator

$$\mathbf{x}(k+l)=\left[\mathbf{G}^{l}\right]\mathbf{x}(k)+\left[\sum_{j=0}^{l-1}\mathbf{G}^{j}\mathbf{H}\right]u(k) \tag{3.24}$$

In deriving Eq. (3.24), the same assumptions used by Sobiski reguarding continuous system input u are also held, so that the future inputs $u(k+j)$ can be approximated by the current input $u(k)$. Define

$$\mathbf{\Phi}_{d}(l)=\mathbf{G}^{l} \tag{3.25}$$

and

$$\mathbf{\Psi}_{d}(l)=\sum_{j=0}^{l-1}\mathbf{\Phi}_{d}(j)\mathbf{H} \tag{3.26}$$

These are the discrete state transition matrix and convolution integral matrix, respectively. The drawback of the discrete state space filter (Eq. (3.24)) is that it can only predict delay in integer multiples of the frame time T. If the time delay t_d cannot be divided exactly by T, say $t_d = lT+\tau, (0<\tau<T)$, the exact amount of prediction may be calculated with the interpolation method

$$\mathbf{x}\left((k+l)T+\tau\right)=\left[e^{\mathbf{A}\tau}\right]\mathbf{x}\left((k+l)T\right)+\left[\int_{0}^{\tau}e^{\mathbf{A}\lambda}d\lambda\right]\mathbf{B}u(k) \tag{3.27}$$

Comparing Eq. (3.17) and (3.19), one can see that the Sobiski/Cardullo algorithm is a type of extrapolator in state space format. The primary difference is that the Sobiski/Cardullo filter uses more system information, including the control input information. In the continuous state space predictor, more derivative or integration information is used. In the discrete state space predictor, more past information is used. Inclusion of the control input is a distinguishing characteristic of the Sobiski/Cardullo filter when compared to the lead/lag or the McFarland filter.

To test the Sobiski/Cardullo compensator, the same system as depicted in Fig. 3.3 is used, except the lead/lag filter is replaced with the Sobiski/Cardullo filter. Because it is in the state space format, the system is redrawn as in Fig. 3.12. The pilot model and the aircraft model are cascaded together to form the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, and to calculate $\mathbf{\Phi}$ and $\mathbf{\Psi}$. As was done for the other two filters, 200, 400 or 800 ms delay and compensation are tested, and the frequency responses and step responses are shown in Fig. 3.13 and Fig. 3.14.

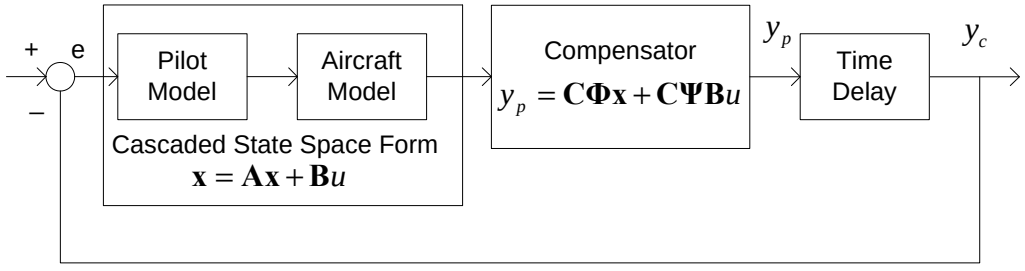


Fig. 3.12. Block diagram of a control system with a state space compensator

The Bode diagrams show that the Sobiski/Cardullo filter can compensate a 200 ms delay with unnoticeable gain distortion and phase error up to 30 rad/s. In

compensating a 400 ms delay, the gain distortion is very slight, and the phase error is unnoticeable for frequencies lower than 10 rad/s, but the compensation displays slight phase lead that increases with frequency above 10 rad/s. In the 800 ms delay compensation case, gain distortion is significant in high frequency area, and the phase lead is insufficient for frequencies above 5 rad/s, causing the system to be unstable. For this reason, the step response of the 800 ms delay case is not presented.

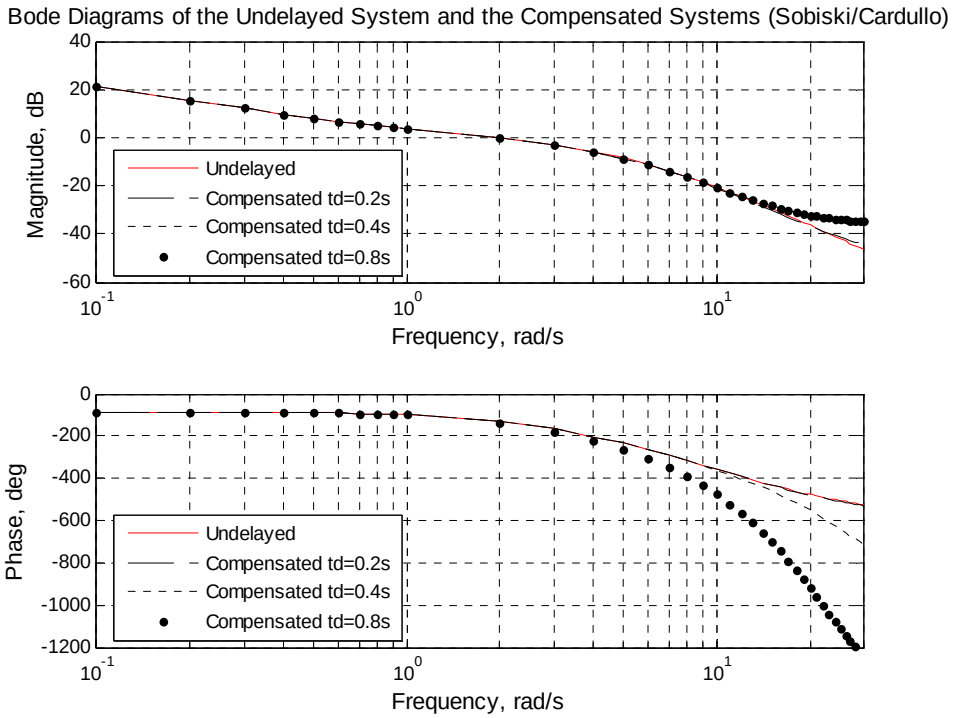


Fig. 3.13. Bode Diagrams of the Compensated Systems (Sobiski/Cardullo)

The step responses shown in Fig. 3.14 verify the frequency response analysis. The "compensated" response (thick dashed curve) refers to y_c in Fig. 3.12, and the result obtained by delaying the prediction by t_d is illustrated by the dashed dot curve. For good compensation, the compensated response must be as close to the response of the undelayed system (solid curve) as possible. The Sobiski/Cardullo filter achieves perfect compensation for 200 ms delay in this system, and for delay of 400 ms, the compensation

error is insignificant. Since these are all closed-loop step responses, they illustrate the change in responsiveness caused by the prediction and delay. Fig. 3.15 directly compares the lead/lag filter, the McFarland filter and the Sobiski/Cardullo filter to step responses. The superiority of the Sobiski/Cardullo filter is obvious in terms of both gain distortion and phase compensation.

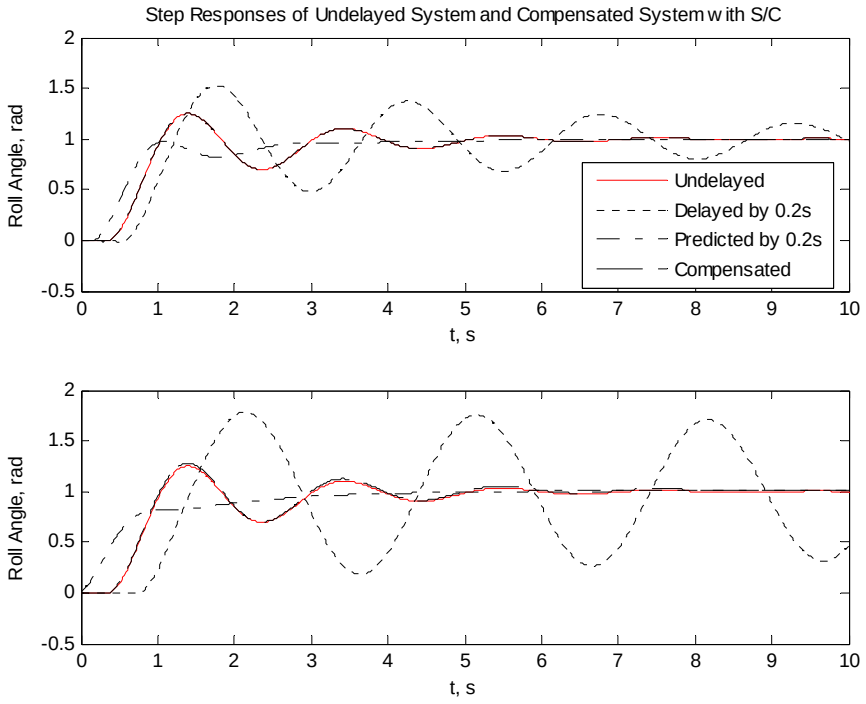


Fig. 3.14. Step responses of the Compensated Systems (Sobiski/Cardullo)

However, the Sobiski/Cardullo filter has several problems. First, since the state space equations are derived from linear differential equations, and they are constant in Sobiski's implementation, the filter can only be applied to linear time-invariant (LTI) system. As previously stated, the aerodynamics of a flight simulator are usually nonlinear and time variant, and thus the matrices $\mathbf{A}$, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are not available. This is the primary reason the Sobiski/Cardullo filter has remained in laboratory use since its advent almost 20 years ago. Second, Sobiski's implementation has some limitations. In his

implementation, the Pade approximation of the time delay is added before the compensator, as is shown in Fig. 3.11., (where Pade approximation of the time delay is necessary and the time delay model is cascaded with the operator model and the aircraft model so that the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ carry the information of the time delay) in contrast, the author's approach is shown in Fig. 3.12. In Sobiski's system, the delayed variable to be compensated must have the same dimension as the undelayed one. In a flight simulator, the transport delay appears to be the delayed image displayed on the screen. Because the image and the corresponding aircraft state do not have the same dimension, the real transport delay is not linear. In the author's implementation (shown in Fig. 3.12) the delay occurs after the prediction which more closely represents the compensation in a real flight simulator. In a real system, the "Time Delay" may be a pure transport delay in the visual system, and may not necessarily match the Pade approximation of the time delay. Therefore, Sobiski's implementation has more theoretical significance than practical usefulness. Third, the assumptions for approximating the future control input with the current one, such as piece-wise constant, sinusoidal, exponentially decaying, etc, do not apply to the real simulation conditions. As given by Eq. (3.23), the matrix $\mathbf{D}$ for the compensated system is not zero, and hence the high frequency components of the stochastic pilot control input u will be added to the compensation y_p. Finally, the state space filter involves a large computation burden because of the matrix operations, especially the power series operations in the calculations of the state transition matrix $\mathbf{\Phi}$ and the convolution integral matrix $\mathbf{\Psi}$. Because of these limitations it would be desirable to develop a more practical state space predictor that can compensate longer delay than

currently available time delay compensators, and simplify the algorithm to provide minimal computation cost.

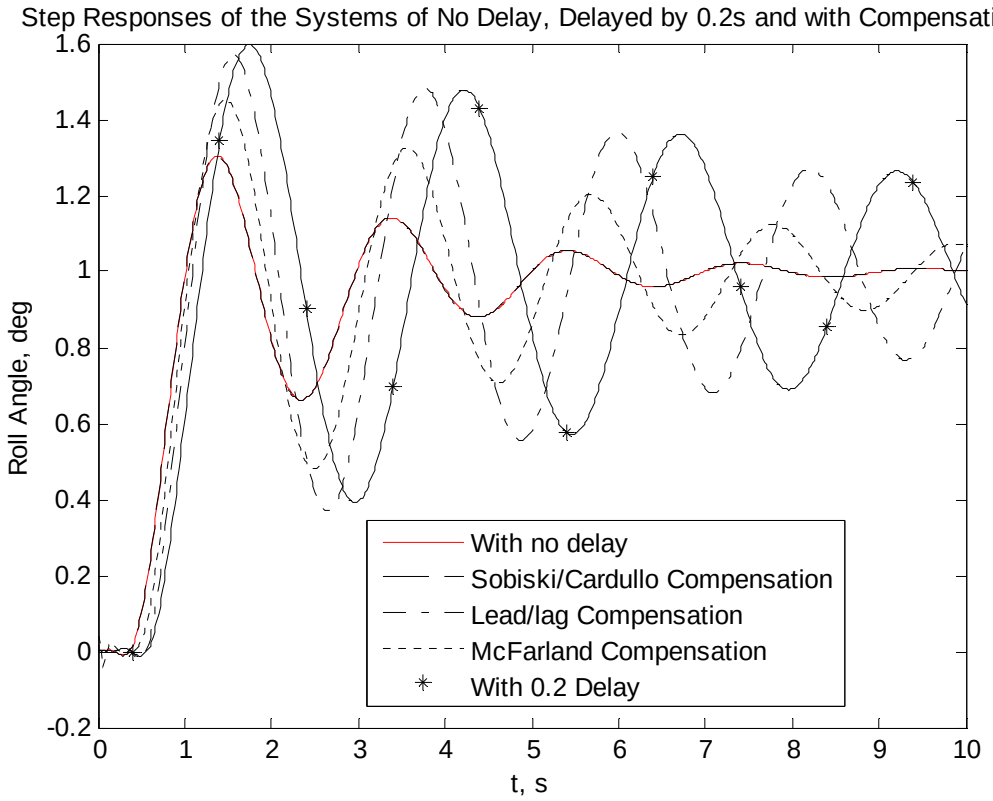


Fig. 3.15. Comparison of the Three Prominent Compensators

4. Novel Approaches to Compensation of Time Delay

As stated in Chapter 3 (section 3.2), there are numerous problems with the currently available compensators. The McFarland predictor offers insufficient phase lead and large gain distortion at high frequencies due to the constant coefficients that do not change with the simulation conditions. The Sobiski/Cardullo filter requires a linear time-invariant system model, whereas the aerodynamic models used in modern flight simulators are usually non-linear, time-variant and are frequently not readily available. Another major problem with the Sobiski/Cardullo filter is its computational burden. This chapter presents some new techniques to address these problems. First, a simple spike-reduction algorithm for the McFarland filter is introduced. Next, this chapter discusses using least squares methods in both frequency and time domains to design a three-velocity predictor. After that, the author will present two novel compensation algorithms that have been developed—an adaptive predictor which uses a Kalman filter and a state space predictor which uses a linear model of the aircraft dynamics to predict future states. The well-known Kalman filter technique is used in a unique manner so that the predictor can accurately provide the desired amount of prediction. From five different implementations of the Kalman estimator, the best option was selected based on the results of theoretical analyses. The state space filter with a linear reference model is the first practical model referenced state space predictor applied in a flight simulator to compensate the time delay. Several currently available linear aircraft dynamic models were tested, and the one that achieved the best prediction, based on the offline tests (see Chapter 5, p106), was chosen. The relationship between the reference model and the

quality of prediction was also investigated. By simplifying the state predictor to an ordinary predictor in a transfer function format, the computational workload is reduced significantly. The two new compensators are described below.

4.1. Reduction of Spikes in the McFarland Compensator

Close examination of the spikes caused by the McFarland compensation shows that wherever the spike occurs, the absolute value of the difference in the prediction of the aircraft state between two adjacent iterations is abnormally large (Fig. 4.1), much greater than the maximum of the corresponding absolute difference of the undelayed aircraft state. This fact leads to the definition of a criterion to decide if a spike will occur.

If y and y_p are the undelayed and predicted aircraft states, let

$$md = Max_{i=1}^{k}\left(\left|y[i]-y[i-1]\right|\right),$$

where k indicates the current iteration, and let $d_p = y_p[k]-y_p[k-1]$, then, if

$$r_d = \frac{\left|d_p\right|}{md} >= \mu \tag{4.1}$$

a noticeable spike will occur (μ varys with the time delay). For a delay up to 200 ms, $\mu = 2.5$ works well. If there is a noticeable spike, recalculate the prediction using

$$y_p[k] = y_p[k-1] + {d_p}/{r_d}, \tag{4.2}$$

and the spike will be reduced. Fig. 4.2 shows an example.

Pilots flying the VMS at the NASA Langley Research Center commented that their performance was much better with this algorithm. Nonetheless, it is worth mentioning that this simple spike-reduction algorithm is only an expedient measure. From Fig. 4.2, it's obvious that the algorithm still causes prediction error, and more

importantly, the spike reduction algorithm reduces the phase lead in the spike-concentrated area.

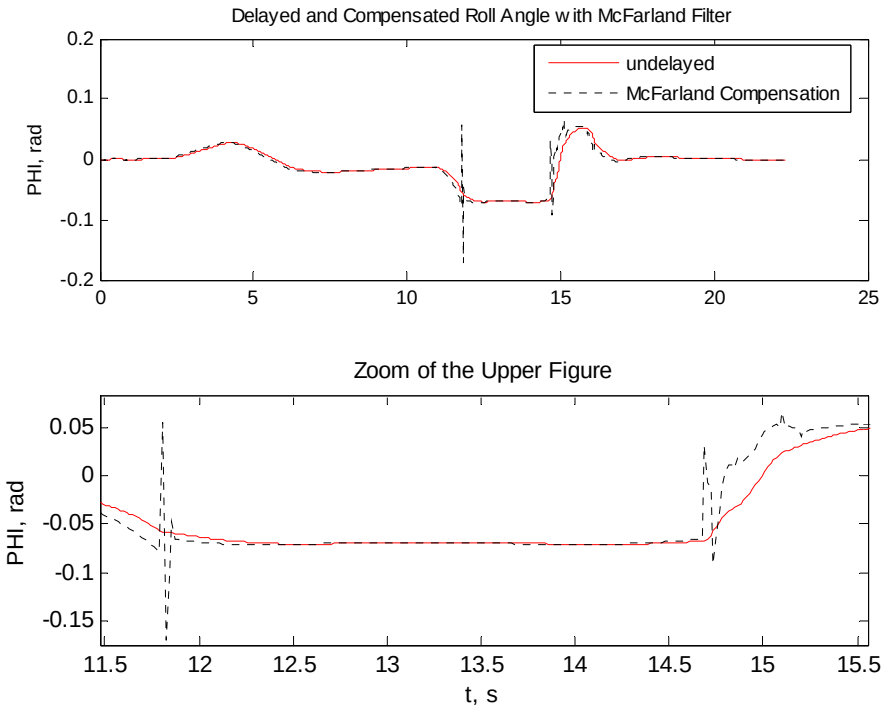


Fig. 4.1. Spikes caused by the McFarland compensation

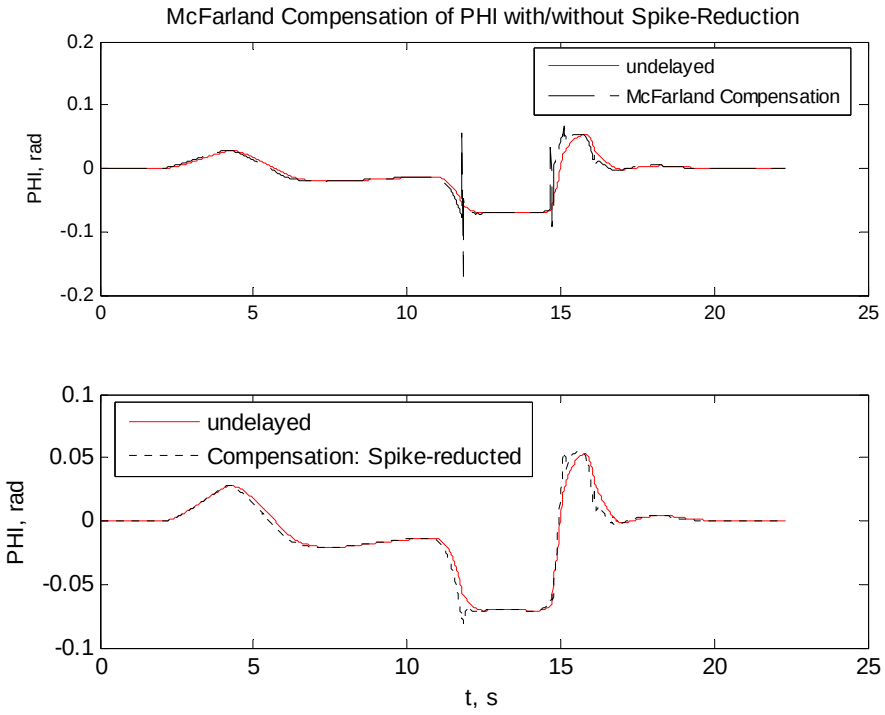


Fig. 4.2. Spikes are reduced

4.2. Frequency Domain Least Squares Method to Design McFarland Predictor[†]

In deriving the three coefficients of his predictor with sinusoidal tuning, McFarland used the system bandwidth, but he did not include a system dynamics model. For this reason, the McFarland compensator may work well for some systems with certain types of input, but it may not yield satisfactory results for other systems. The problem can be minimized if the three coefficients are designed by taking a system dynamics model into account. One way to achieve this is to make sure the frequency

---

[†] If the coefficients are not determined with the sinusoidal tuning, the predictor is no longer technically a McFarland predictor, however, it is still referred to as a McFarland predictor for convenience in this discussion.

characteristics of the compensated system are as close to that of the undelayed system as possible, based on some defined criterion. The least squares method can be used for this purpose. As an example, a McFarland filter, designed with the frequency least squares method for the system depicted in Fig. 3.3, is presented. The design criterion is to minimize the cost function

$$I = \frac{1}{2}\sum_{i=1}^{N}\left|H_d(z_i)H_{McF}(z_i) - H_0(z_i)\right|^2 \tag{4.3}$$

where $H_d(z)$, $H_{McF}(z)$ and $H_0(z)$ are, respectively, the pulse transfer functions of the delayed system, the McFarland filter and the undelayed system, given by

$$H_d(z) = \mathbf{Z}\left\{\frac{1-e^{-Ts}}{s}\left[sH_{ac}(s)\right]\left[H_{op}(s)\right]\left[H_{td}(s)\right]\right\} \tag{4.4}$$

$$H_{McF}(z) = \frac{T}{2}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + \left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right) \tag{4.5}$$

$$H_0(z) = \mathbf{Z}\left\{\frac{1-e^{-Ts}}{s}\left[H_{ac}(s)\right]\left[H_{op}(s)\right]\right\} \tag{4.6}$$

where $z = e^{-jT\omega}$. The pulse transfer functions are used because the McFarland filter is discrete. In Eq. (4.4) and (4.6), the operation $\mathbf{Z}\left\{\frac{1-e^{-Ts}}{s}[...]\right\}$ is used to obtain the discrete transfer function using a zero-order hold (ZOH), with a sampling period *T*. $H_{ac}(s)$, $H_{op}(s)$ and $H_{td}(s)$ are given in Eq. (2.11), (2.10) and (2.4), respectively. Because the output of $H_{ac}(s)$ is the displacement (roll angle), and the McFarland filter uses the velocities, the differentiator s is pre-multiplied to change the output of $H_d(z)$ to velocity. By choosing a suitable number N of frequency points ω_i within the pilot

working frequency range, the cost function (4.3) is minimized, and satisfactory results can be achieved. The final result is

$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix}^T = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{4.7}$$

with

$$\mathbf{A} = real\left(\mathbf{Z}_V\right) + imag\left(\mathbf{Z}_V\right), \mathbf{Z}_V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \\ z_1^2 & z_2^2 & \cdots & z_N^2 \end{bmatrix}^T \tag{4.8}$$

$$\mathbf{b} = real\left(\mathbf{D}_T\right) + imag\left(\mathbf{D}_T\right), \mathbf{D}_T = \begin{bmatrix} H_T\left(z_1\right) & H_T\left(z_2\right) & \cdots & H_T\left(z_N\right) \end{bmatrix}^T \tag{4.9}$$

where $H_T\left(z\right) = H_0\left(z\right) - H_d\left(z\right)\frac{T}{2}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$.

Although the cost function (4.3) seems to put constraints on the magnitude only, it actually minimizes the errors in both the real part and the imaginary part, so that the phase error is minimized also. The Bode diagram (Fig. 4.3) and step response (Fig. 4.4) of the compensated system with the McFarland filter designed with this method also shows that both the gain distortion and phase error are decreased compared with the ordinary McFarland filter.

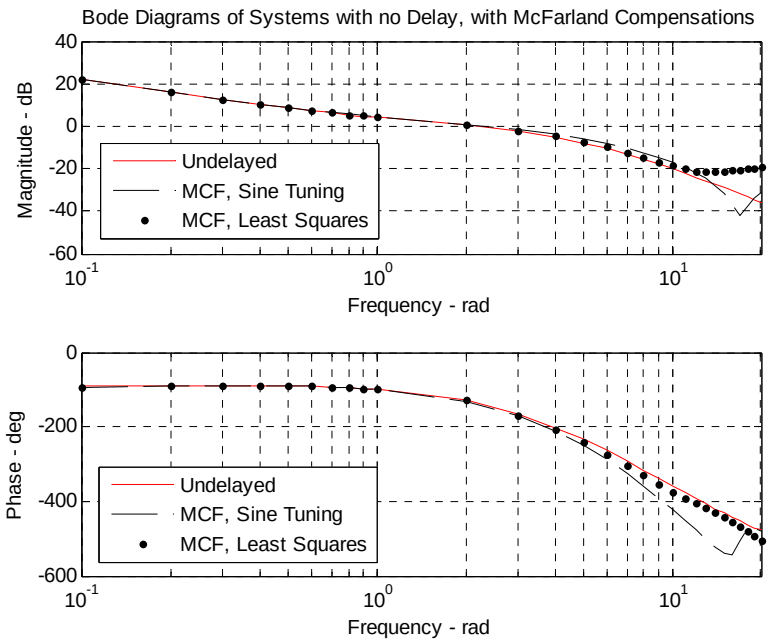


Fig. 4.3. Frequency responses of compensated systems with McFarland filters

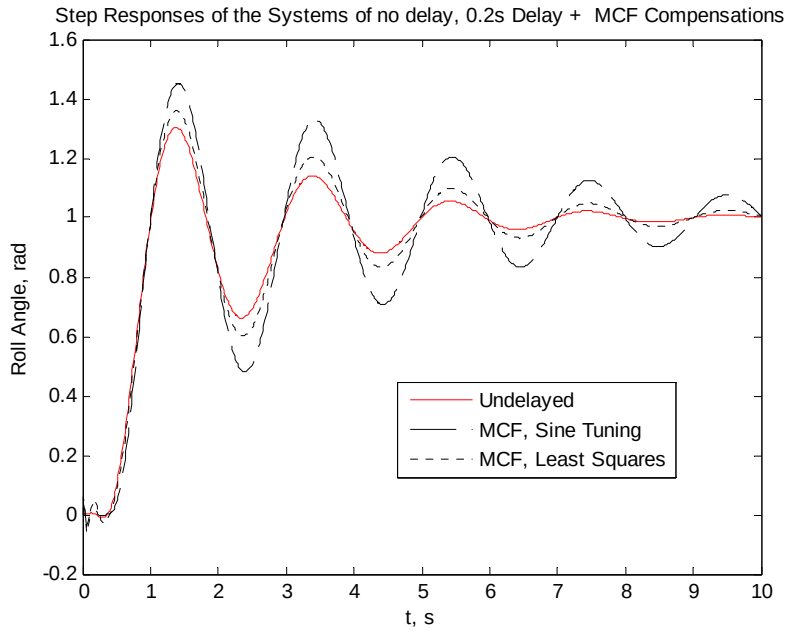


Fig. 4.4. Step responses of compensated systems with McFarland filters

4.3. Time Domain Least Squares Method to Design McFarland Predictor

The frequency domain least squares method introduced in the previous section is suitable for designing a McFarland predictor in a simple simulation system similar to Fig. 3.3. However, in a high fidelity, piloted flight simulator, $H_{op}(s)$ is replaced with a human pilot, and the flight dynamics are much more complicated than $H_{ac}(s)$ and may not be expressed as a simple transfer function. The frequency least squares method does not apply in this situation. Nevertheless, because the McFarland filter involves only the displacement and its velocity, which are available in the flight simulator, they can be used to design the McFarland filter coefficients that best fit the simulation condition according to similar quadratic criteria. A sample of roll axis displacement and velocity data recorded during an offset approach flown in the VMS at the NASA Langley Research Center, is plotted in Fig. 3.8, and is reused here to derive this predictor. Defining the roll angle and its velocity as y and v, respectively, the time domain quadratic cost function is given by

$$I = \frac{1}{2}\sum_{k=1}^{N}\left[y_d(k) + b_0 v(k) + b_1 v(k-1) + b_2 v(k-2) - y(k) \right]^2 \tag{4.10}$$

with k being the iteration index, and y_d the roll angle y delayed by t_d, which may be obtained from y using the Pade approximation. Minimization of the cost function I in the last equation results in a left pseudo-inverse as given in Eq. (4.7), with different matrices $\mathbf{A}$ and $\mathbf{b}$.

$$\mathbf{A}^T = \begin{bmatrix} v(0) & v(1) & v(2) & v(3) & \cdots & v(N) \\ 0 & v(0) & v(1) & v(2) & \cdots & v(N-1) \\ 0 & 0 & v(0) & v(1) & \cdots & v(N-2) \end{bmatrix} \tag{4.11}$$

$$\mathbf{b} = \begin{bmatrix} y(0) - y_d(0) \\ y(1) - y_d(1) \\ \vdots \\ y(N) - y_d(N) \end{bmatrix} \tag{4.12}$$

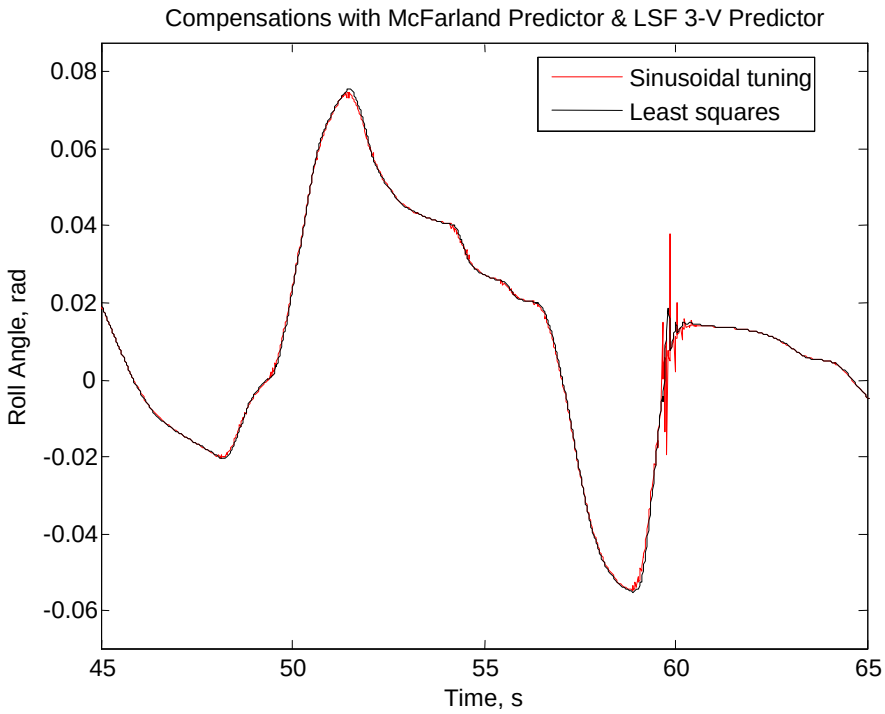


Fig. 4.5. Roll angle, velocity and roll stick of a simulation

After the three coefficients b_0, b_1 and b_2 are designed, calculate the predicted displacement with Eq. (3.11). A comparison of the McFarland compensators designed with the sinusoidal tuning and the least squares method, on the recorded data plotted in Fig. 3.8 is shown in Fig. 4.5. (In this case 192 ms prediction was applied.) With the least squares design, the spikes are significantly reduced, and the high frequency artifacts in the roll angle peak areas caused by the sinusoidally-tuned McFarland compensator are also removed. However, the calculation workload is substantially increased, and another serious drawback of the least squares methods is the three coefficients are not available

until the end of the simulation. Thus, this method is referred to as an offline least squares method.

4.4. Adaptive Predictor

The time domain least squares method discussed in the previous section can be implemented, for each iteration, so that the predictor coefficients are updated instantly. This involves calculating the pseudo-inversion (Eq. (4.7)) in each iteration, requiring intensive computation and storage. However, these calculations can be carried out in a recursive manner such that the current update uses only the current data and the results of the previous calculation. In this manner both redundant calculation and the storage of large quantaties of data are avoided. The coefficients are updated with the process so that the large prediction error caused by the constant filter coefficients is reduced. A schematic diagram of the adaptive predictor is illustrated in Fig. 4.6.

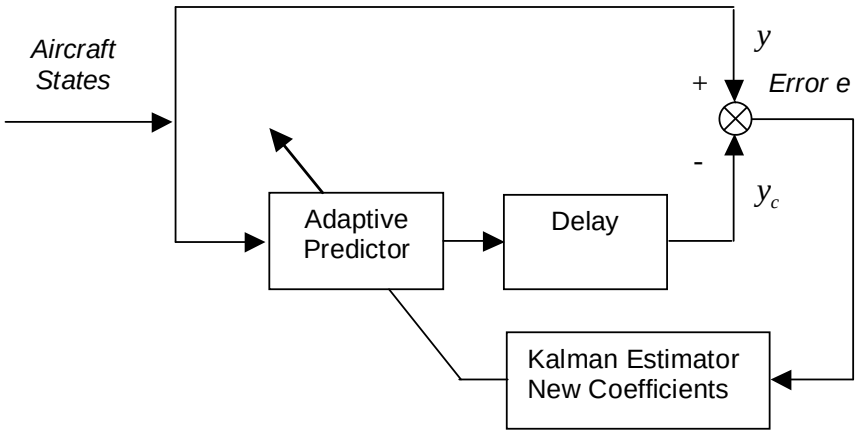


Fig. 4.6. Structure of the adaptive predictor with the Kalman estimator

To conveniently compare the adaptive predictor with the McFarland predictor, three consecutive steps of velocity will be used as an example. Therefore, using Fig. 4.6, the compensated aircraft state y_c is given by $y_c = y_d + b_0 v + b_1 v_{-1} + b_2 v_{-2}$, where y_d is the

delayed aircraft state, and v, v_{-1} and v_{-2} are the velocities of y in three consecutive iterations. Minimizing the quadratic loss function given by either Eq. (4.10) or

$$I = \frac{1}{2} E\left[y - y_c\right]^2 \tag{4.13}$$

where operator "E" refers to the expectation. (Dividing Eq. (4.10) by N yields Eq. (4.13). The same results are obtained by setting the derivatives to zero. Eq. (4.13) was designed to conveniently introduce the ODE given in Eq. (4.30)). By making use of the Kalman matrix inversion theorem, the final recursive least squares method is given as

$$\mathbf{P}(k) = \left[\mathbf{I}_{3\times 3} - \mathbf{K}(k)\mathbf{j}^T(k)\right]\mathbf{P}(k-1) \tag{4.14}$$

$$\mathbf{K}(k) = \mathbf{P}(k-1)\mathbf{j}(k)\left[1 + \mathbf{j}^T(k)\mathbf{P}(k-1)\mathbf{j}^T(k)\right]^{-1} \tag{4.15}$$

$$\tilde{\boldsymbol{\theta}}(k) = \tilde{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\left[y(k) - y_d(k) - \mathbf{j}^T(k)\tilde{\boldsymbol{\theta}}(k-1)\right] \tag{4.16}$$

where

$$\tilde{\boldsymbol{\theta}}(k) = \left[b_0(k) \quad b_1(k) \quad b_2(k)\right]^T \tag{4.17}$$

which gives the three coefficients of the adaptive predictor, and

$$\mathbf{j}^T(k) = \left[v(k) \quad v(k-1) \quad v(k-2)\right] \tag{4.18}$$

is a vector consisting of the three consecutive velocities. The algorithm starts with $\mathbf{P}(k_0) = \left(\mathbf{j}(k_0)\mathbf{j}^T(k_0)\right)^{-1}$ and $\tilde{\boldsymbol{\theta}}(k_0) = \mathbf{P}(k_0)\mathbf{j}^T(k_0)\left(y(k_0) - y_d(k_0)\right)$, where k_0 corresponds to the first time when the quantity $\mathbf{j}^T(k_0)\mathbf{j}(k_0)$ is nonsingular. Notice that since the quantity inside the brackets is a scalar, a matrix inversion is avoided, and the algorithm is considerably simplier than the original left pseudo inverse given by Eq. (4.7). This is the widely used Kalman filter algorithm. Because the coefficients are updated

each iteration, the prediction error is also reduced significantly. Simulations show that the new adaptive predictor substantially reduces the high frequency gain distortion and spikes caused by the McFarland filter. The three-step recursive least squares algorithm (Eq. (4.14) - (4.16)) can be formulated in a compact form as:

$$\tilde{\boldsymbol{\theta}}(k)=\tilde{\boldsymbol{\theta}}(k-1)+\frac{\mathbf{j}(k)}{\sum_{i=k_0}^{k}\mathbf{j}(k)\mathbf{j}^T(k)}\left[y(k)-y_d(k)-\mathbf{j}^T(k)\tilde{\boldsymbol{\theta}}(k-1)\right] \tag{4.19}$$

with $\sum_{i=k_0}^{k}\mathbf{j}(i)\mathbf{j}^T(i)=\mathbf{P}^{-1}(k)=\mathbf{P}^{-1}(k-1)+\mathbf{j}(k)\mathbf{j}^T(k)$.

The algorithm given by Eqs. (4.14)-(4.16) allocates even weight to the data available no matter how old they are in the data history. A forgetting factor $\lambda\,(0<\lambda\le 1)$ would allow the designer to assign larger weight to the more recent data so that they contribute more than the older data in the least squares algorithm. The new cost function with the forgetting factor becomes $I=\frac{1}{2}\sum_{i=k_0}^{k}\lambda^{k-i}\left(y(i)-y_c(i)\right)^2$, and the update formulas of the matrices $\mathbf{P}$ and $\mathbf{K}$ are changed to

$$\mathbf{P}(k)=\left[\mathbf{I}_{3\times 3}-\mathbf{K}(k)\mathbf{j}^T(k)\right]\mathbf{P}(k-1)/\lambda \tag{4.20}$$

$$\mathbf{K}(k)=\mathbf{P}(k-1)\mathbf{j}(k)\left[\lambda+\mathbf{j}^T(k)\mathbf{P}(k-1)\mathbf{j}^T(k)\right]^{-1} \tag{4.21}$$

The recursive formula for updating $\tilde{\boldsymbol{\theta}}$ is unchanged (the same as Eq. (4.16). An optimal forgetting factor range has been determined by trial-and-error and an algorithm with a forgetting factor in this range demonstrated a reduction of the predicting error.

The recursive least squares method can be simplified for different approximation algorithms. The simplification comes from avoiding updating the matrix $\mathbf{P}$, since

updating the matrix **P** dominates the computing effort for a large number of iterations. The first approximation considered was the Kaczmarz's algorithm. But because the recursive least-square algorithm updates the current estimate $\boldsymbol{\theta}^T(k)$ based on the previous estimate $\tilde{\boldsymbol{\theta}}(k-1)$ and the new measurement $y(k)=\mathbf{j}^T(k)\boldsymbol{\theta}(k)$, which contains information only in the direction of $\mathbf{j}^T(k)$ in the parameter space, Kaczmarz proposed the normalized projection algorithm that minimizes $\left|\tilde{\boldsymbol{\theta}}(k)-\tilde{\boldsymbol{\theta}}(k)\right|^2$ subject to the constraints, $y(k)=\mathbf{j}^T(k)\tilde{\boldsymbol{\theta}}(k)$. The cost function for the Kaczmarz's algorithm (the normalized projection) is $I=\left|\tilde{\boldsymbol{\theta}}(k)-\tilde{\boldsymbol{\theta}}(k)\right|^2+\bar{\alpha}\left[y(k)-\mathbf{j}^T(k)\tilde{\boldsymbol{\theta}}(k)\right]$, which may be considered as a function of the vector variable $\boldsymbol{\theta}(k)$ with $\bar{\alpha}$, the Lagrange multiplier, as a parameter. Taking derivatives with respect to $\boldsymbol{\theta}(k)$ and $\bar{\alpha}$ and invoking the stationary results in the Kaczmarz's algorithm, also called the normalized projection algorithm, $\boldsymbol{\theta}(k)$ becomes

$$\boldsymbol{\theta}(k)=\boldsymbol{\theta}(k-1)+\frac{\mathbf{j}(k)}{\mathbf{j}^T(k)\mathbf{j}(k)}\left[y(k)-y_d(k)-\mathbf{j}^T(k)\boldsymbol{\theta}(k-1)\right] \tag{4.22}$$

Comparing the Kaczmarz's algorithm with the original least squares algorithm given in Eq. (4.19) shows that the denominator of the Kaczmarz's algorithm is changed from a matrix $\sum_{i=k_0}^{k}\mathbf{j}(i)\mathbf{j}^T(i)$ to a scalar $\mathbf{j}^T(k)\mathbf{j}(k)$. This is where the simplification comes in. Two other similar approximation least squares algorithms, with scalar denominators are the Stochastic Approximation algorithm and the Least Mean Square (LSM) algorithm

$$\boldsymbol{\theta}(k)=\boldsymbol{\theta}(k-1)+\frac{\mathbf{j}(k)}{\sum_{i=k_0}^{k}\mathbf{j}^T(i)\mathbf{j}(i)}\left[y(k)-y_d(k)-\mathbf{j}^T(k)\boldsymbol{\theta}(k-1)\right] \tag{4.23}$$

$$\boldsymbol{\theta}(k)=\boldsymbol{\theta}(k-1)+\frac{\mathbf{j}(k)}{\gamma}\left[y(k)-y_d(k)-\mathbf{j}^T(k)\boldsymbol{\theta}(k-1)\right] \tag{4.24}$$

The formula of the stochastic approximation (SA) algorithm ((4.23)) looks similar to the one-step basic least squares algorithm given in Eq. (4.19), with only the summation element being changed from a matrix $\mathbf{j}(i)\mathbf{j}^T(i)$ to a scalar $\mathbf{j}^T(i)\mathbf{j}(i)$. Inspired by the similarity, the stochastic approximation algorithm may also be rewritten in three steps: first Eq. (4.14) is rewritten as

$$\mathbf{P}(k)=\left[1-\mathbf{j}(k)\mathbf{K}(k)\right]\mathbf{P}(k-1) \tag{4.25}$$

and the remaining two steps are exactly the same as Eqs. (4.15) and (4.16). If the exponential forgetting factor is applied to the three-step stochastic approximation algorithm, Eq. (4.20) becomes

$$\mathbf{P}(k)=\left[1-\mathbf{j}(k)\mathbf{K}(k)\right]\mathbf{P}(k-1)/\lambda \tag{4.26}$$

and the formulas for the second and third steps are the same as Eqs. (4.21) and (4.16), respectively. Finally, the stochastic algorithm with a forgetting factor is formulated as

$$\boldsymbol{\theta}(k)=\boldsymbol{\theta}(k-1)+\frac{\mathbf{j}(k)}{\sum_{i=k_0}^{k}\lambda^{k-i}\mathbf{j}^T(i)\mathbf{j}(i)}\left[y(k)-y_d(k)-\mathbf{j}^T(k)\boldsymbol{\theta}(k-1)\right] \tag{4.27}$$

Offline tests of the five adaptive prediction algorithms, namely the Kalman filter algorithm, Kalman algorithm with a forgetting factor, the Kaczmarz's algorithm, the stochastic approximation algorithm and the least mean square algorithm, on recorded roll angle data are illustrated in Fig. 4.7 and Fig. 4.8 (which is a blowup of Fig. 4.7)

It is obvious from Fig. 4.8 that the stochastic approximation works best among these five adaptive algorithms. The next chapter will show the superiority of the

stochastic approximation by using three criteria. The stochastic approximation algorithm was chosen to implement the adaptive predictor in the final piloted tests.

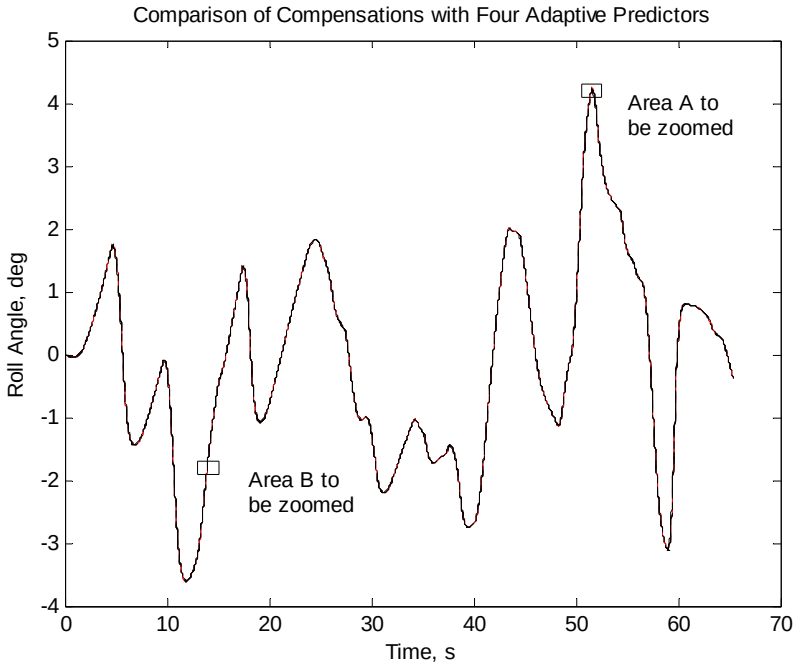


Fig. 4.7. Adaptive compensations applied to the roll angle using different algorithms

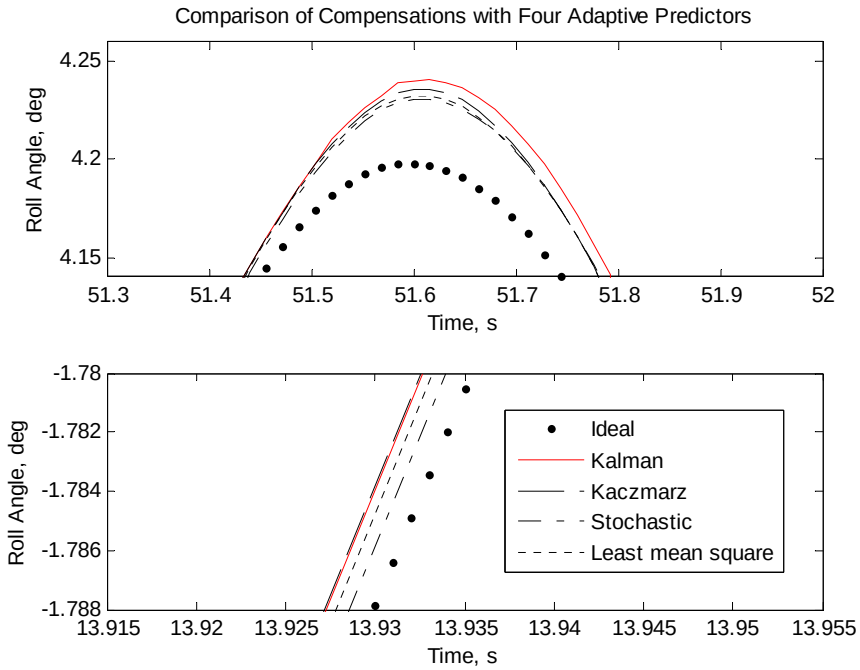


Fig. 4.8. Zoom of Figure 4.7

The recursive least squares methods have long been applied in system identification. In the past, the simplified recursive least squares algorithms, such as the stochastic approximation algorithm, were favored over the basic Kalman filter algorithm, because the former were more useful in real time when the processing speed of the digital computer was low. However, the simplified algorithms usually give biased identification, and they are vulnerable in the presence of noise. Therefore, as new generations of computers appeared with much higher processing speed, the basic algorithm became the preferred method once processing time was not a primary concern. Nevertheless, the stochastic approximation method is more suitable to application in a fight simulator when compensating for transport delay than the Kalman filter algorithm for three reasons.

First, the application here is not system identification, rather it is a method to design a predictor aimed at providing accurate phase lead with small error. The input variables (the aircraft state and its velocity) contain little noise because they are results of

the aerodynamic computation, rather than measurement, and one is simply the integration of the other.

Second, the processing time is a concern. In order to minimize the transport delay, it is required that all the aircraft states in different coordinate systems be calculated in one frame after the operator control input is sampled. Unfortunately in the Langley VMS these computations consume almost the whole frame, leaving very limited time available for implementing the compensation.

Third, the stochastic approximation algorithm yields the least predicting error (most accurate phase lead with least gain distortion) among all these recursive least squares algorithms. The explanation is given as follows:

The basic requirement for a predictor of this type, i.e., using three consecutive iterations of velocity to predict, is $b_0 + b_1 + b_2 = t_d$. This guarantees that the prediction is close to the time delay t_d if the velocity change is not abrupt (low frequencies). But as analyzed in Chapter 2, if the velocity changes quickly (which frequently happens in real flight simulations), the filter causes a large error when the differences among the three coefficients are large. The benefit of using three steps of velocity instead of one is that a weighted average of three past values is less likely to cause a spike than just one past value. Another requirement of a three-velocity predictor is that the differences among the three coefficients must be small. A first choice might be $b_0 = b_1 = b_2 = t_d/3$, but this is not necessary because the velocity changes irregularly. This is why an online recursive update of coefficients is employed. The coefficients may vary from simulation to simulation, but ideally they are as close to each other as possible. The smaller their difference, the better the prediction of the filter at high frequencies. Investigation shows

that, of the previously introduced recursive least squares methods, the stochastic approximation algorithm always generates the least difference among the coefficients. Table 4.1 shows the converged filter coefficients calculated with five different algorithms averaged across 16 sets of simulation data in the roll axis. Only the stochastic approximation method yields three coefficients that are close to each other. Note that the least mean squares algorithm alone gives all-positive coefficients in this case, but it has been found in other simulations that the coefficients have alternative sign changes.

Table 4.1. Three coefficients calculated with different methods (t_d =0.192s)

Algorithm	b_0	b_1	b_2	Sum
McFarland Filter	2.8613	-5.2342	2.5650	0.192
Kalman Filter	-0.0030	-0.5894	0.7811	0.189
Stochastic Approximation	0.0525	0.0571	0.0789	0.188
Kaczmarz Algorithm	-0.2064	0.0503	0.3485	0.192
Least Mean Square	0.0284	0.0700	0.1066	0.205

The algorithm expressed in Eq. (4.23) belongs to a large family called Stochastic Approximation Algorithms. Two other prominent versions are the Saridis and Stein's algorithm and the Kwatny's form[22]. A general expression of the stochastic approximation algorithm is

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1) + \varepsilon_k \mathbf{j}(k)\left[y(k) - y_d(k) - \mathbf{j}^T(k)\boldsymbol{\theta}(k-1)\right] \tag{4.28}$$

where the scalar coefficient ε_k must be monotonically decreasing. Specifically Robins and Monro show that if

$$\varepsilon_k > 0, \quad \lim_{k\to\infty} \varepsilon_k = 0, \quad \sum_{k=1}^{\infty} \varepsilon_k = \infty \ , \quad and \sum_{k=1}^{\infty} \varepsilon_k^2 < \infty \tag{4.29}$$

the algorithm Eq. (4.28) converges. For the algorithm given in Eq. (4.23),

$$\varepsilon_k = \frac{1}{\sum_{i=k_0}^{k} \mathbf{j}^T(i)\,\mathbf{j}(i)}$$

where $\mathbf{j}^T(k) = \begin{bmatrix} v(k) & v(k-1) & v(k-2) \end{bmatrix}$. In a flight simulation, the velocity does not change exponentially, so the coefficient $\varepsilon_k = \sum_{i=k_0}^{k} \mathbf{j}^T(i)\,\mathbf{j}(i)$ meets the conditions given in Eq. (4.29), and hence the algorithm Eq. (4.23) is indeed a stochastic approximation algorithm. Conversely, in the Kaczmarz's algorithm $\varepsilon_k = \mathbf{j}^T(i)\,\mathbf{j}(i)$, and in the least mean square algorithm ε_k is a constant, both of which do not meet the $\lim_{i\to\infty} \varepsilon_i = 0$ requirement. Therefore, these are not stochastic approximation algorithms.

The ODE that characterizes the asymptotic behavior of the stochastic approximation algorithm given in Eq. (4.28) is[23]

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}} E\left\{ \left[y - y_d \right] - \mathbf{j}^T \boldsymbol{\theta} \right\}^2 \\ &= E\left(\mathbf{j}\mathbf{j}^T \right) - E\left(y\mathbf{j}^T \right) \boldsymbol{\theta} \end{aligned} \tag{4.30}$$

where E is the mathematical expectation. The right side of the first equality of Eq. (4.30) is the negative gradient of the cost function, indicating that the stochastic approximation algorithm may be interpreted as a stochastic gradient descent algorithm. Though the gradient of the cost function in Eq. (4.13) with respect to $\boldsymbol{\theta}$ is unknown, the gradient at the current sample of $-\left\{ \left[y(k) - y_d(k) \right] - \mathbf{j}^T(k)\boldsymbol{\theta}(k) \right\}^2$ is just $\mathbf{j}(k)\left[y(k) - y_d(k) - \mathbf{j}^T(k)\boldsymbol{\theta}(k-1) \right]$, which is the dynamic term from Eq. (4.28). Because

the quantity inside the brackets $y(k)-y_d(k)-\mathbf{j}^T(k)\boldsymbol{\theta}(k-1)$ is a scalar, from the asymptotic ODE, $\dot{\boldsymbol{\theta}}(k)$ is in the direction of $\mathbf{j}(k)$. This means the modifying term of $\boldsymbol{\theta}(k+1)$ is in the direction of $\mathbf{j}(k)$. And since the recursive algorithm starts with an initial zero vector, it is logical that $\boldsymbol{\theta}$ is in a direction close to the average of $\mathbf{j}$. Because $\mathbf{j}^T(i)=\left[v(i) \quad v(i-1) \quad v(i-2)\right]$, and the average velocity does not change much within two iterations, it follows that the three elements of $\boldsymbol{\theta}$ (or the coefficients of the adaptive predictor) do not differ much either. This demonstrates why the stochastic approximation algorithm gives the best prediction.

4.5. A Practical State Space Compensator with a Reference Model

As stated in section 3.3, although the Sobiski/Cardullo predictor shows some desirable advantages, it also has limitations in its implementation that prevent its practical application in a flight simulator. The basic state space prediction equation $\mathbf{x}_p=\boldsymbol{\Phi}\mathbf{x}+\boldsymbol{\Psi}\mathbf{B}u$ requires a linear time invariant (LTI) system, while the simulator flight dynamics are usually nonlinear, time-variant and coupled in different degrees of freedom, and are frequently not available. Instead of being expressed in state space equations, they are often expressed in coupled non-linear differential equations. However, employing an aircraft reference model in the predictor algorithm can solve this problem. This results in the development of a novel practical state space predictor, which is discussed as follows.

4.5.1. Basic Implementation

A reference model is an approximate linear aircraft dynamics model that is used to form the predictor states from the operator control input and the aircraft states, as well as to provide the state transition matrix and the convolution integral. This reference

model is used so that the state space prediction equation $\mathbf{x}_p = \mathbf{\Phi x} + \mathbf{\Psi B}u$ may be applied in a flight simulator when the aerodynamics are not available. Fig. 4.9 illustrates this approach, where $\mathbf{x}_a$ is the aircraft state vector (including the aircraft displacement, velocity and acceleration, etc), $\mathbf{x}$ is the filter state vector and $\mathbf{x}_p$ is the predicted filter state vector. Because the aircraft state vector $\mathbf{x}_a$ includes the aircraft state y to be predicted and its velocity and acceleration, the filter state vector $\mathbf{x}$ is also in terms of y. Therefore, the predicted filter state $\mathbf{x}_p$ calculated with $\mathbf{x}_p = \mathbf{\Phi x} + \mathbf{\Psi B}u$ contains the predicted information of y. Then using the matrix $\mathbf{C}$ to retrieve y_p from $\mathbf{x}_p$, the desired prediction is achieved.

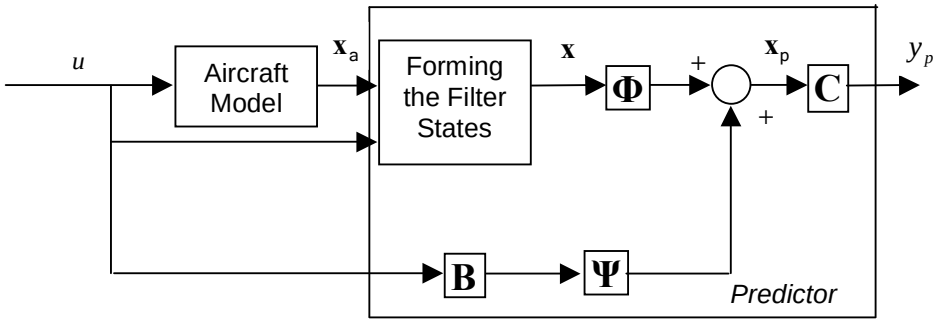


Fig. 4.9. Structure of the state space compensator using a reference model

Four 4th-order reference models were tested. The first two models give the relationship between the pitch angle and the roll angle, respectively, and the corresponding stick deflections of a fixed wing jet flying at an altitude of 30,000ft and an airspeed of 430 knots. They will be called Model I and Model II (Eq. (2.11)). The other two models are for a large commercial transport, in the pitch mode, one for cruise, and one for landing. They will be called Model III and Model IV. These four models share the same general form of the transfer function

$$H_{AC}(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.31}$$

The observable state space matrices of this general model are given in Eq. (4.32). Selection of the observable canonical form is made because the output is desired to be the first state variable. (In other words, $y = x_1$; and after the predicted filter state vector $\mathbf{x}_p$ is obtained, the predicted aircraft state is just the first element of $\mathbf{x}_p$, i.e., $y_p = x_{p1}$.) The expressions of the four state filter variables are directly derived from $\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}u \end{cases}$, and the result is given in Eq. (4.33). Note that $\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \mathbf{x}$ is the predictor state vector. This is an artificial state vector because, aside from the first element x_1, the remaining elements do not exist in the simulator and have no physical significance. They are formed from the reference model.

$$\mathbf{A} = \begin{bmatrix} -a_3 & 1 & 0 & 0 \\ -a_2 & 0 & 1 & 0 \\ -a_1 & 0 & 0 & 1 \\ -a_0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{D} = 0 \tag{4.32}$$

$$\begin{cases} x_1 = y \\ x_2 = \dot{y} + a_3 y \\ x_3 = \ddot{y} + a_3 \dot{y} + a_2 y - b_2 u \\ x_4 = b_0 \int_0^T u dt \end{cases} \tag{4.33}$$

Note that, in addition to Eq. (4.33), alternative formulas for calculating the state variables x_2, x_3 and x_4 exist. Namely, $x_2 = \int_0^T \{-a_2 x_1 + x_3 + b_2 u\} dt$, $x_3 = \int_0^T \{-a_1 y + x_4 + b_1 u\} dt$ and $x_4 = \dot{x}_3 + a_1 y - b_1 u$. Though the final prediction, when using the alternatives is only slightly different, these alternative equations are not used

because they require an extra integration, they include the jerk term $\dot{x}_3$, which introduces high frequency artifacts.

In most aircraft simulations, the aerodynamic model is processed first, and then the accelerations and velocities of the vehicle are calculated in the body frame. These velocities and accelerations are then transformed to other necessary frames. For the visual display, the cueing channel for which the time delay compensation is designed, the topodetic coordinate frame is normally used (occasionally the geodetic frame[24], or north-east-up frame is used). In this coordinate frame, the six axes are roll, pitch, yaw, altitude, latitude and longitude. Prior to this study, the accelerations in these six axes were not normally available. The formulas to calculate these six accelerations are given below without their derivations (the derivations are in Appendix F).

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{4.34}$$

where ϕ, θ and ψ are the roll, pitch and yaw angles (Euler angles), and $\dot{p}$, $\dot{q}$ and $\dot{r}$ are the three angular accelerations in the body frame. The other three accelerations are

$$\begin{bmatrix} \ddot{\lambda} \\ \ddot{l} \\ \ddot{h} \end{bmatrix} = \mathbf{P}_{B2E}^{T} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \mathbf{T} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.35}$$

where λ, l and h are the latitude, longitude and altitude, and $\dot{u}$, $\dot{v}$ and $\dot{w}$ are the three translational accelerations in the body frame, and the matrices $\mathbf{P}_{B2E}^{A}$ and $\mathbf{T}$ are

$$\mathbf{P}_{B2E}^{T} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ \sin\theta & -\sin\phi\cos\theta & -\cos\phi\cos\theta \end{bmatrix} \tag{4.36}$$

$$\mathbf{T} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{31} \\ \alpha_{21} & \alpha_{22} & \alpha_{32} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \tag{4.37}$$

with the elements being given in Table 4.2.

Another problem is on the input u in the six axes, since there are only four control inputs: the roll stick, the pitch stick, the rudder pedal and the throttle. For the roll and pitch compensation, the input is the roll stick and pitch stick, respectively. For yaw, both the roll stick and the rudder pedal contribute, but the input is chosen to be the rudder pedal because its control is more direct. Altitude is changed indirectly by the pitch angle or throttle, thus the pitch stick can be employed as its input. Longitude is affected by the heading, hence the rudder pedal may be used as its input.

Table 4.2. Elements of the matrix T

Element	Expression
α_{11}	$v(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)+w(\cos\phi\sin\psi-\sin\phi\sin\theta\cos\psi)$
α_{12}	$-u\sin\theta\cos\psi+v\sin\phi\cos\theta\cos\psi+w\cos\phi\cos\theta\cos\psi$
α_{13}	$-u\cos\theta\sin\psi-v(\sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi)+w(\sin\phi\cos\psi-\cos\phi\sin\theta\sin\psi)$
α_{21}	$v(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)-w(\cos\phi\cos\psi+\sin\phi\sin\theta\sin\psi)$
α_{22}	$-u\sin\theta\sin\psi+v\sin\phi\cos\theta\sin\psi+w\cos\phi\cos\theta\sin\psi$
α_{23}	$u\cos\theta\cos\psi+v(\sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi)+w(\sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi)$
α_{31}	$-v\cos\phi\cos\theta+w\sin\phi\cos\theta$
α_{32}	$u\cos\theta+v\sin\phi\sin\theta+w\cos\phi\sin\theta$
α_{33}	0

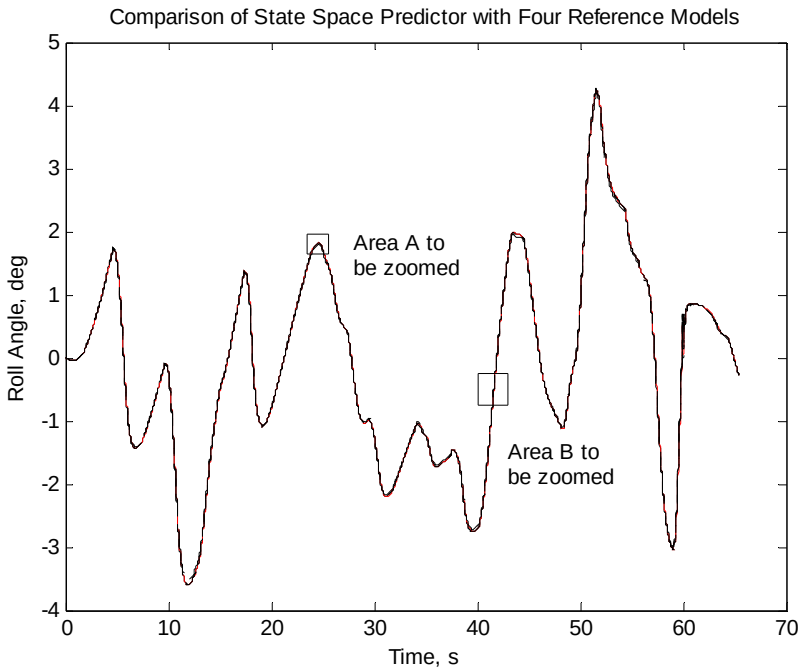


Fig. 4.10. Comparison of the state space predictors with four reference models

After the filter state vector $\mathbf{x}$ is available from Eq. (4.33), apply the state space prediction formula $\mathbf{x}_p = \mathbf{\Phi x} + \mathbf{\Psi B}u$ to calculate the predicted filter state vector, and finally calculate the predicted aircraft state with $y_p = \mathbf{C}\mathbf{x}_p$ ($\mathbf{D} = 0$). Compensated data, from the state space filters using the four formerly introduced reference models on recorded roll angle data are illustrated in Figs. 4.10 and 4.11 (an enlargement of Fig. 4.10). They show that the state space filter with reference Model IV achieves the best prediction among the four reference models. The next chapter will present a detailed comparison of the effectiveness of the four reference models. Therefore, Model IV, or the landing model of a large commercial transport, in pitch has been chosen as the reference model for the state space compensation in the final tests.

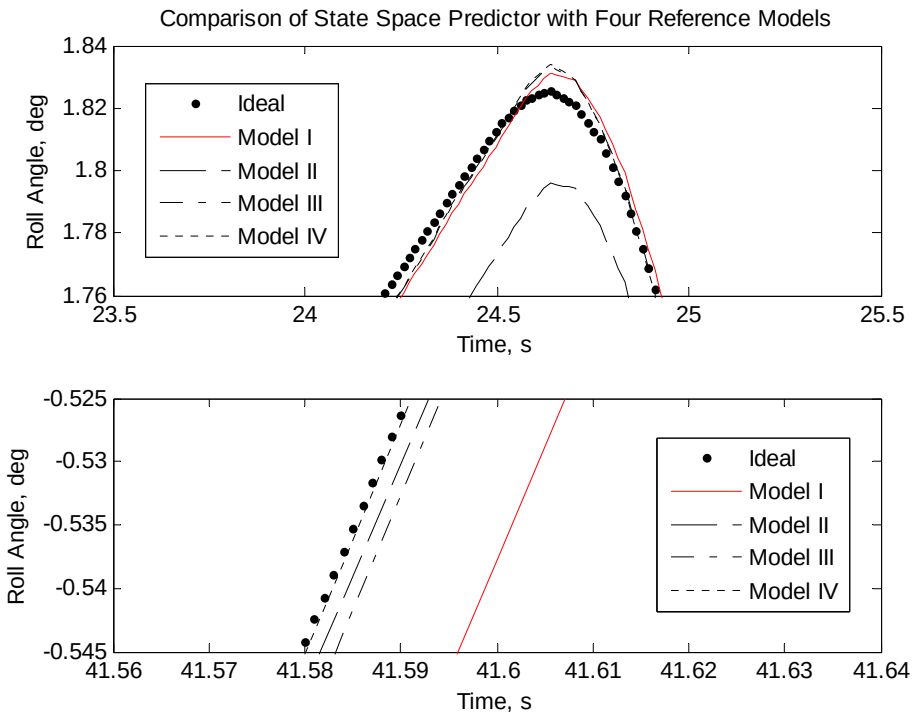


Fig. 4.11. Zoom of Figure 4.10

The reference models used in the state space compensation are not necessarily 4th-order. Two 3rd-order models[25] were also evaluated. The formulation of the filter state vector with the 3rd-order reference model is similar to that with the 4th-order, and hence only the result is listed here. The general transfer function and the state space expression of the model, and the filter state vector are, respectively

$$H_{AC}(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.38}$$

$$\mathbf{A} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \mathbf{D} = 0 \tag{4.39}$$

$$\begin{cases} x_1 = y \\ x_2 = \dot{y} + a_2 y - b_2 u \\ x_3 = \int_0^T \{-a_0 y + b_0 u\}\, dt \end{cases} \tag{4.40}$$

There are also alternative expressions for some elements of the predictor state vector, but only the formulation that resulted in better compensation is shown. Although compensation achieved by the state space filter with a 3rd-order reference model is not as good as with the 4th-order models (the 3rd-order reference model produces greater phase error but smaller gain distortion in the prediction), it is still better than the McFarland compensation. A detailed comparison among all these reference models will be presented in the next chapter.

4.5.2. Simplification and Essence of the State Space Compensator

Calculating the four predicted filter states as given by $\mathbf{x}_p = \mathbf{\Phi x} + \mathbf{\Psi B}u$ involves many matrix operations. However, what is really needed is the predicted aircraft state y_p given by $y_p = \mathbf{C}\mathbf{x}_p$, and because $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$, y_p is just the first element of $\mathbf{x}_p$. Therefore, calculation of the last three elements of $\mathbf{x}_p$ is not necessary, and this shows that the algorithm can be simplified. The simplification is given in Eq. (4.41), where ϕ_{ij} and ψ_j are elements of matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ (Eqs. (3.20) and (3.21)). By substituting the expressions of the four elements of the filter state vector in Eq. (4.33), the final simplified state space compensator is obtained as Eq. (4.42)

$$
\begin{aligned}
y_p &= \mathbf{C}\mathbf{x}_p = \left(\mathbf{C}\boldsymbol{\Phi}\right)\mathbf{x} + \left(\mathbf{C}\boldsymbol{\Psi}\mathbf{B}\right)u \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \left\{ \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix} u \right\} \\
&= \phi_{11}x_1 + \phi_{12}x_2 + \phi_{13}x_3 + \phi_{14}x_4 + \psi_1 u
\end{aligned} \tag{4.41}
$$

$$
y_p = \left(\phi_{11} + \phi_{12}a_3 + \phi_{13}a_2\right) y + \left(\phi_{12} + \phi_{13}a_3\right) \dot{y} + \phi_{13}\ddot{y} + \left(\psi_1 - \phi_{13}b_2\right) u + \phi_{14}b_0 \int_0^T u dt \tag{4.42}
$$

This is the essence of the state space filter! When compared with the previous compensator, this shows that, while the previous compensators use three consecutive steps of velocity to predict, the state space filter uses the current velocity, acceleration, the control input and its integral to predict.

Likewise, the state space predictor based on the 3rd-order reference model is simplified, and the final result is similar to Eq. (4.42):

$$
y_p = \left(\phi_{11} + \phi_{12}a_2\right) y + \phi_{12}\dot{y} - \phi_{13}a_0 \int_0^T y dt + \left(\psi_1 - \phi_{12}b_2\right) u + \phi_{13}b_0 \int_0^T u dt \tag{4.43}
$$

Eq. (4.43) implies that the state space predictor may also be interpreted as a general PID controller with two modification terms on the control input *u*.

The prediction of the state space filter depends solely on the five coefficients in Eq. (4.42) or (4.43), which are functions of the time delay and the reference model. Therefore, an algorithm based on the state transition matrix and convolution integral with a reference model can be used as a design tool—to design the coefficients of the compensator. Because the reference model is time invariant, these coefficients are constants that may be calculated offline. Therefore, during each iteration of the simulation, only five multiplications and four additions are required—computation is simplified significantly. Because only the first rows of the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are needed

to calculate the predictor coefficients, calculating the remaining rows is unnecessary. This is significant in applications where a time-variant reference model is required, and the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ must be updated every iteration. The state space filter may be implemented as depicted in Fig. 4.12 and Fig. 4.13, where the 90 ms of delay in the visual system is the baseline transport delay determined in the VMS at the NASA Langley Research Center (artificial delay was also added for the experiments to determine the general applicability of the predictor to compensate other delays), and the coefficients are given in Table 4.3 and Table 4.4, respectively.

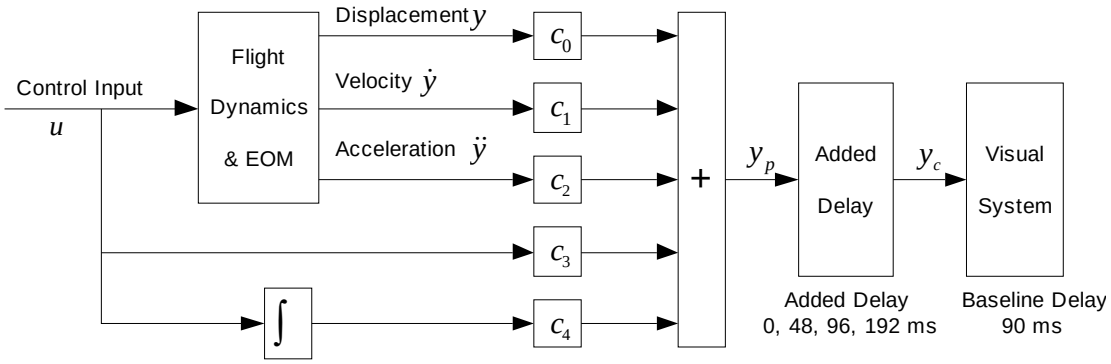


Fig. 4.12. Simplified state space compensator using a 4th-order reference model

If the reference model is of 4th-order, the coefficients of the last two terms of the control input u (c_3 and c_4) are so small compared to the first three terms without u that they may be neglected. In other words, a good reference model attenuates the contribution of the second term of $\mathbf{x}(t+t_d) = e^{\mathbf{A}t_d}\mathbf{x}(t) + \int_0^{t_d} e^{\mathbf{A}(t_d-\tau)}\mathbf{B}u(t+\tau)\,d\tau$, on which the state space predictor is based. The trivial contribution of the input u makes it easier to justify approximating the future input with the current input than Sobiski's assumptions. On the other hand, the small contribution of the control input u is desirable because its high frequency jumps are attenuated. As the time delay gets longer, the coefficients c_3

and c_4 become larger so that more high frequency content is added to the prediction, resulting in a larger predicting error. Note also that the coefficient c_0 of term y is unity, a property the McFarland predictor also possesses ($y_p(k) = y(k) + b_0 v + b_1 v_{-1} + b_2 v_{-2}$).

Table 4.3. Coefficients of the state space predictor shown in Fig. 4.12

Coefficient	Expression	For 4th-order Model I, t_d =186 ms
c_0	$\phi_{11} + \phi_{12} a_3 + \phi_{13} a_2$	1
c_1	$\phi_{12} + \phi_{13} a_3$	0.1850
c_2	ϕ_{13}	0.0159
c_3	$\psi_1 - \phi_{13} b_2$	4.0052e-04
c_4	$\phi_{14} b_0$	3.2020e-05

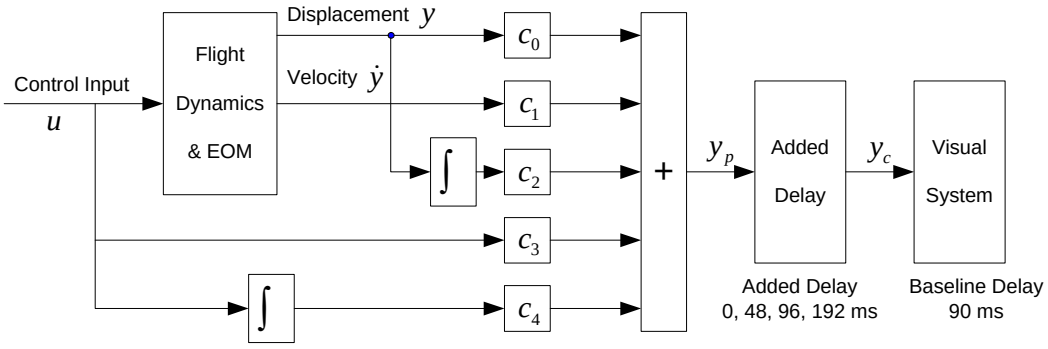


Fig. 4.13. Simplified state space compensator using a 3rd-order reference model

Conversely, if the reference model is of 3rd-order, the contribution to the prediction from the pilot control input u is greater from the integral term ($c_3 > c_2$), and it may explain why the 3rd-order reference model cannot achieve as good a prediction as the 4th-order reference model. The coefficient of y is no longer unity, and the longer the time delay, the farther it deviates from unity. This is a desirable property because as the time delay gets longer, the future y will resemble the current y less, and therefore the current y should contribute less to the prediction.

Table 4.4. Coefficients of the state space predictor shown in Fig. 4.13

Coefficient	Expression	For 3rd-order model A, t_d =186 ms
c_0	$\phi_{11}+\phi_{12}a_2$	0.9949
c_1	ϕ_{12}	0.1750
c_2	$-\phi_{13}a_0$	-7.3938e-004
c_3	$\psi_1-\phi_{12}b_2$	0.0036
c_4	$\phi_{13}b_0$	7.3938e-004

4.5.3. State Space Predictor with a Discrete State Transition Matrix

Chapter 3 introduced the discrete format of the state space predictor, and the discrete state transition matrix and convolution integral matrix (Eqs. (3.24)-(3.26)). A prediction algorithm based on the discrete state space filter using a reference aircraft model has also been tested in a manner similar to the continuous one. In changing the prediction algorithm from Eq. ((3.19) to Eq. (3.24), the formation of the filter state vector is also changed—it must be formed with the discrete state space equations:

$$\begin{cases}\dot{\mathbf{x}}(k+1)=\mathbf{G}\mathbf{x}(k)+\mathbf{H}u(k)\\ y(k)=\mathbf{C}\mathbf{x}(k)+\mathbf{D}u(k)\end{cases}$$

The general pulse transfer function corresponding to the continuous transfer function Eq. (4.31), its state space matrices and the discrete filter state vector are given as follows

$$H_{AC}(z)=\frac{b_3z^3+b_2z^2+b_1z+b_0}{z^4+a_3z^3+a_2z^2+a_1z+a_0} \tag{4.44}$$

$$\mathbf{G}=\begin{bmatrix}-a_3 & 1 & 0 & 0\\ -a_2 & 0 & 1 & 0\\ -a_1 & 0 & 0 & 1\\ -a_0 & 0 & 0 & 0\end{bmatrix},\ \mathbf{H}=\begin{bmatrix}b_3\\ b_2\\ b_1\\ b_0\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix},\ \mathbf{D}=0 \tag{4.45}$$

$$\begin{cases} x_1(k) = y(k) \\ x_2(k) = -a_2 x_1(k-1) + x_3(k-1) + b_2 u(k-1) \\ x_3(k) = -a_1 x_1(k-1) + x_4(k-1) + b_1 u(k-1) \\ x_4(k) = -a_0 x_1(k-1) + b_0 u(k-1) \end{cases} \tag{4.46}$$

Because the integrator in the continuous state space equation becomes the delay operator in the discrete equations, the derivative or integral terms are replaced with the past values of the aircraft state. Lack of aircraft velocity and acceleration information in prediction is the primary drawback of the discrete state space filter. Inclusion of the current and several past values may result in the derivative terms indirectly (due to numerical differentiation) introducing errors. Therefore, the discrete state space filter is expected to be inferior to a continuous one that makes use of the derivative terms directly. Application of the discrete state space predictor to the recorded simulation data has proven this. This is more obvious from the simplified algorithm. Corresponding to Eq. (4.42), the simplified algorithm with a 4th-order reference model is given by

$$\begin{aligned} y(k+l) = g_{11} y(k) - (a_2 g_{12} + a_1 g_{13} + a_0 g_{14}) y(k-1) + g_{12} x_3(k-1) \\ + g_{13} x_4(k-1) + (b_2 g_{12} + b_1 g_{13} + b_0 g_{14}) u(k-1) + h_1 u(k) \end{aligned} \tag{4.47}$$

where g_{ij} and h_i are the elements of the matrices **G** and **H** given in Eqs. (3.25) and (3.26). There are more contributions from the control input u, and conversely, the states $x_3(k-1)$ and $x_4(k-1)$ have to be calculated recursively with the 3rd and 4th equations of Eq. (4.46), which results in error accumulation. Finally, as stated in Chapter 3, the time delay to be compensated with the discrete state space filter must be an integer multiple of the frame cycle. After considering all these factors, the discrete state space filter was not used during the final simulation tests.

4.5.4. Relationship Between Prediction and Reference Model

Section 4.5.3 mentions that the four 4th-order reference models give slightly better compensation than the 3rd-order reference model, and the large commercial transport landing pitch model achieves the best compensation compared to the other three 4th-order reference models. This raises some questions: which factors of the reference model make the most difference in compensation? Does the order of the reference model affect the compensation? What is the relationship between the prediction quality and the reference model? This section is a summary of the research designed to answer these questions.

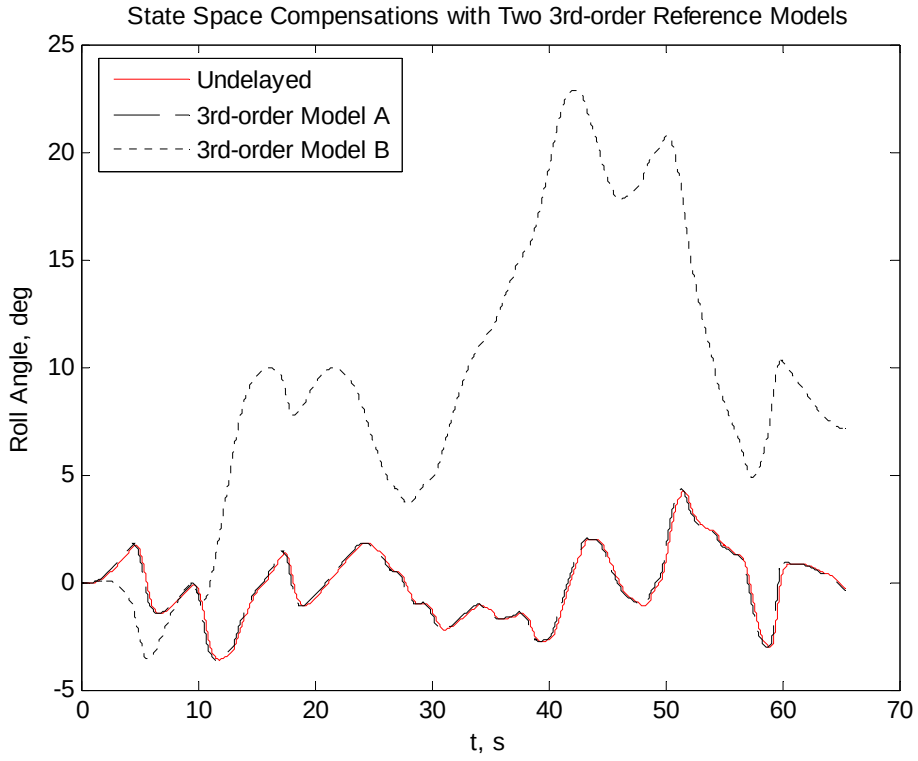


Fig. 4.14. State space compensations with two 3rd-order reference models

The 3rd-order reference model that achieved the best results is not an aerodynamic model, instead it was borrowed from a book on control theory (for convenience call it 3rd-order Model A). Another 3rd-order model (3rd-order Model B) was also chosen, which

takes the same form of transfer function as given in Eq. (4.38). The state space compensations (t_d =0.192s) using these two reference models on the same recorded roll data that was used for the other compensators are shown Fig. 4.14.

The 3rd-order model B gives a completely wrong prediction. Checks of the state transition matrix **Φ** and the convolution matrix **Ψ** do not reveal the problem. However, if the five coefficients $c_0 - c_4$ of the simplified filter shown in Fig. 4.13 are checked and compared with those of Model A, the problem becomes clear. These coefficients (t_d =0.192s) are listed in Table 4.5. The coefficients of the 4th-order Model I, and the McFarland filter are also included for comparison.

Table 4.5. Coefficients of different compensators for t_d =0.192s

Filter	Proportion	Derivative	Integral	Input	Integration of input
	c_0	c_1	c_2	c_3	c_4
3rd-order model A	0.9946	0.1803	-0.0008	0.0039	0.0008
3rd-order model B	0.4604	0.0480	-0.5299	0.5659	0.5299
4th-order model IV	1	0.1909	0.0169	4.3972e-004	3.5149e-005
McFarland	1	0.192 ($b_1 + b_1 + b_2$)			

In Table 4.5, only the state space predictor with model B does not work. For other models, which work well, the coefficient of the proportional term is either unity or very close to unity, the coefficient of the derivative term is equal to or very close to the time delay, and all other coefficients are much smaller. This shows that the derivative extrapolation is either the only compensation or dominates the compensation. But this is not true for the 3rd-order model B: its coefficient of the proportional term is far less than unity, and the derivative compensation contributes even less than the integral

compensation and the terms in the control input. This explains why the prediction quickly drifts away from the undelayed aircraft state.

The values of the coefficients of the simplified state space compensator have direct relationships with the reference model. Table 4.4 shows that the coefficients are functions of the elements of the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, which are in terms of $a's$, $b's$ and t_d. Deriving these expressions involves the evaluation of the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$. The definition of the state transition matrix $\mathbf{\Phi}$ is an infinite Taylor series

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} = \sum_{i=0}^{\infty} \frac{\left(\mathbf{A}t_d\right)^i}{i!} \tag{4.48}$$

Because this series is always convergent, it is usually approximated by truncating it to a finite series. The number of terms required to get satisfactory approximation depends on the matrix $\mathbf{A}$. For the five reference models that work well, five terms seem to be sufficient. Therefore,

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} \approx \mathbf{I} + \mathbf{A}t_d + \frac{1}{2}\left(\mathbf{A}t_d\right)^2 + \frac{1}{6}\left(\mathbf{A}t_d\right)^3 + \frac{1}{24}\left(\mathbf{A}t_d\right)^4 \tag{4.49}$$

In fact, the last term contributes little to the final result. Another way to compute the state transition matrix $\mathbf{\Phi}$ is the exact method making use of the Cayley-Hamilton theorem[26]. That algorithm is given by

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} = \sum_{i=0}^{n-1} \alpha_i \mathbf{A}^i \tag{4.50}$$

where n is the order of the matrix $\mathbf{A}$, and $\alpha_0 - \alpha_{n-1}$ are the solutions of the following linear equations

$$\sum_{i=0}^{n-1} \alpha_i \lambda_j^i = e^{\lambda_j t_d}, \left(j = 1...n\right) \tag{4.51}$$

with the λ_j's being the eigenvalues of the matrix $\mathbf{A}$, provided that all the eigenvalues are distinct. The analytical expressions of $\alpha_0 - \alpha_{n-1}$ with/without repeated eigenvalues are given in Appendix A. For the five working reference models, it has been found that

$$\alpha_0 \approx 1, \alpha_1 \approx t_d, \alpha_2 \approx \frac{1}{2}t_d^2, \alpha_3 \approx \frac{1}{6}t_d^3 \tag{4.52}$$

Note that only the 4th-order models have α_3. The approximations in Eq. (4.52) can be verified by the Taylor expansion of $e^{\lambda_j t_d}$: $e^{\lambda_j t_d} = 1 + \left(\lambda_j t_d\right) + ... + \left(\lambda_j t_d\right)^i / i! + ...$ For t_d up to 0.3 seconds, and the absolute values of the real part of the maximal eigenvalue around 1, the fifth term $\left(\lambda_j t_d\right)^4 / 24$ is only 3.3750e-004, and the higher order terms are even smaller. Therefore, the scalar Taylor series can be truncated to four terms, or

$$e^{\lambda_j t_d} \approx 1 + \left(\lambda_j t_d\right) + \left(\lambda_j t_d\right)^2 / 2 + \left(\lambda_j t_d\right)^3 / 6 .$$

Then Eq. (4.51) becomes

$$1 + \left(\lambda_j t_d\right) + \left(\lambda_j t_d\right)^2 / 2 + \left(\lambda_j t_d\right)^3 / 6 \approx \alpha_0 + \alpha_1 \lambda_i + \alpha_2 \lambda_i^2 + \alpha_3 \lambda_i^3 \tag{4.53}$$

Comparing the coefficients of both sides gives Eq. (4.52). In this sense, the state space compensation may be viewed as a Taylor series extrapolation in the state space form.

The convolution integral matrix approximated by making use of Eq. (4.49) is given in the last equation of Eq. (4.54). Substituting the elements ϕ_{ij} and ψ_j of the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ calculated with Eq. (4.49) and Eq. (4.54) into the expressions of the five coefficients listed in Table 4.4 (for state space predictor based on a 3rd-order reference model; application on a 4th-order reference model will be discussed later), the

final expressions of $c_0, c_1, \ldots, c_4$ are given in Table 4.6. Due to the approximations in Eqs. (4.49) and (4.54), these expressions are also only approximate .

$$\begin{aligned}\boldsymbol{\Psi} &= \int_0^{t_d} e^{\mathbf{A}(t_d-\tau)} d\tau \\ &= \left(\boldsymbol{\Phi}-\mathbf{I}\right)\mathbf{A}^{-1} \\ &\approx t_d + \frac{t_d^2}{2}\mathbf{A} + \frac{t_d^3}{6}\mathbf{A}^2 + \frac{t_d^4}{24}\mathbf{A}^3\end{aligned} \tag{4.54}$$

Table 4.6. Approximate expressions of the coefficients of the simplified state space predictor with a 3rd-order reference model

Coefficient	Expression
c_0	$1-\frac{a_1}{2}t_d^2+\frac{a_1a_2-a_0}{6}t_d^3+\frac{a_0a_2-a_1a_2^2+a_1^2}{24}t_d^4$
c_1	$t_d-\frac{a_2}{2}t_d^2+\frac{a_2^2-a_1}{6}t_d^3+\frac{2a_1a_2-a_2^3+a_0}{24}t_d^4$
c_2	$-\frac{a_0}{2}t_d^2+\frac{a_1a_2}{6}t_d^3+\frac{a_0a_2^2-a_0a_1}{24}t_d^4$
c_3	$\left(\frac{1}{2}t_d^2+\frac{a_1}{6}t_d^3+\frac{a_2^2-a_1}{24}t_d^4\right)b_1+\left(\frac{1}{6}t_d^3-\frac{a_2}{24}t_d^4\right)b_0$
c_4	$\left(\frac{1}{2}t_d^2-\frac{a_1}{6}t_d^3+\frac{a_2^2-a_1}{24}t_d^4\right)b_0$

Further examination of the expression of c_0, indicates that the third and fourth terms are much smaller than the second term for a time delay shorter than 0.3s. And the second term, a_1 must be less than 0.5 for a delay of 0.2s or less than 0.25 for a delay of 0.3s for c_0 to be unity within 1%. For example, for model A, a_1=0.3050 results in c_0=0.9946 which would be satisfactory; for model B, a_1=62.1314 results in c_0=0.4406, which is much less than unity. With a similar analysis, a_2 must be less than one to make c_1 fairly close to t_d. Thus, the values of a_1 and a_2 are critical to the applicability of a 3rd-

order model as a reference model. The restrictions on a_1 and a_2 indicate that there must also be some restrictions on the eigenvalues of the reference model because, for a 3rd-order model, the following relationships between the three eigenvalues $\lambda_1 - \lambda_3$ and a_1, a_2 are held

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 = -a_2 \\ \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 = a_1 \end{cases} \tag{4.55}$$

For a stable model, all eigenvalues must have negative real parts. Then Eq. (4.55) implies that $|\lambda_i| \le a_2 \le 1$ and $\lambda_i\lambda_j \le a_1 \le 0.5$. Therefore, the absolute values of all eigenvalues must be at least less than unity to make the reference model work. Model A meets this requirement, but model B does not. The eigenvalues of model B are located much farther from the imaginary axis of the s-plane than model A, and therefore, model B responds much faster than model A (Fig. 4.15). In other words, the bandwidth of model B is much larger than that of model A.

Therefore, it appears that a good reference model can be formed by merely choosing suitable $a's$ and $b's$ in Table 4.6 so that the five coefficients have desirable values. Although it appears that the reference model does not have to match the flight dynamics, in actuality this is not true. In fact, the reason the 3rd-order model B does not work for the roll angle recorded from the VMS is because it has totally different dynamic properties than those of the aircraft model running in the VMS. In other words, the reason that Model B does not work as well as Model A does is that Model A has frequency characteristics close to the aircraft model used in the simulation from which the data was collected, but the Model B does not. Likewise, the four 4th-order reference models and the 3rd-order model A work well because their bandwidths are close to the bandwidth of

the large commercial transport model running on the VMS. This can be verified by simply comparing the bandwidths of the 3rd-order models A and B with those of the four 4th-order aerodynamic models, as shown in Table 4.7. The bandwidth of the 4th-order model II is not available because it has a pure integrator (Type I system, Eq. (2.11)).

Table 4.7. Bandwidths of six reference models

Model	3rd-order		4th-order			
	A	B	I	II	III	IV
Bandwidth (rad/s)	0.6069	8.0561	0.6523	NA	0.1036	0.0324

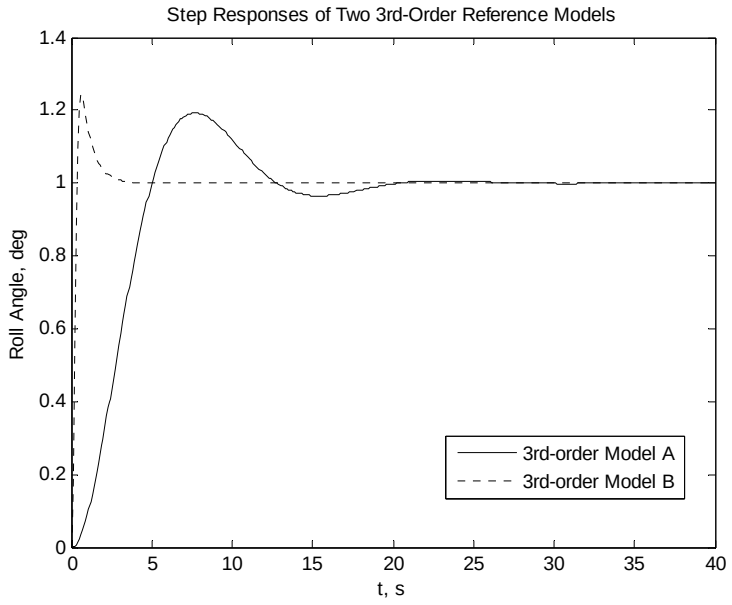


Fig. 4.15. Step responses of two 3rd-order reference models

The bandwidth of the 3rd-order model B is much too high compared to the other aircraft models. One way to determine whether the model bandwidth plays an important role in compensation quality would be to reduce the bandwidth of model B gradually, and apply it to the state space prediction, and see how the compensation changes. The results are demonstrated in Fig. 4.16. Investigation shows that satisfactory compensation can be

achieved when the bandwidth of the reference model is below 1 rad/s. This proves that the bandwidth of the reference model must be close to that of the simulated vehicle dynamics.

Originally, it was assumed that the damping ratio of the reference model made a significant difference in compensation. Some of the reference models that do not work well are over-damped, whereas the aerodynamics of the large commercial transport are under-damped. A study of the impact of varying the damping ratio was conducted, similar to the study in which the bandwidth was varied, and that study shows that the damping ratio of the reference model does not make a significant difference in the state space compensation.

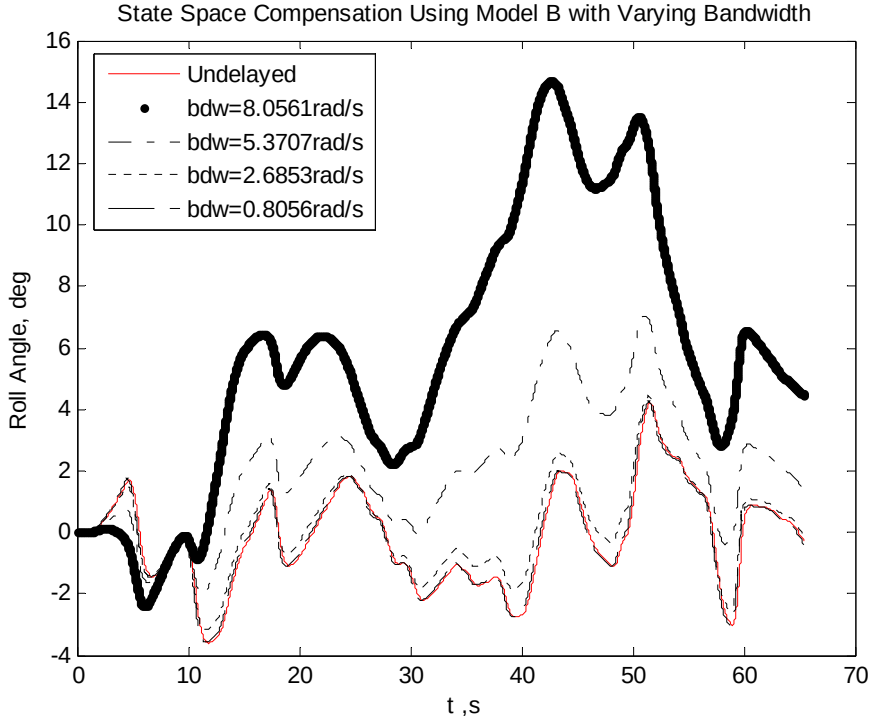


Fig. 4.16. State space compensation using model B with varying bandwidth

With a derivation similar to that of the 3rd-order reference model, the five coefficients of the simplified state space predictor with a 4th-order reference model as

depicted in Fig. 4.12 were obtained in terms of the parameters of the model transfer function. These coefficients are listed in Table 4.8. By comparing Table 4.8 with Table 4.6, it becomes obvious that there is no term in t_d^2. Because t_d^3 is smaller than t_d^2 for t_d <1s, the proportional coefficient c_0, when using a 4th-order model, is closer to unity for the same value of a_1, and the derivative coefficient c_1 is closer to t_d for the same value of a_2. Furthermore, because the coefficients a_1 and a_2 are higher order functions of the eigenvalues when using a 4th-order model (compare Eqs. (4.55) with (4.56)), even though the eigenvalues of a 4th-order model have magnitudes close to those of a 3rd-order model, and the a_1 and a_2 of the 4th-order model are smaller (the absolute values of the aircraft model eigenvalues are usually less than 1), this makes c_0 and c_1 closer to unity and t_d, respectively. Thus, the 4th-order reference model is expected to achieve better prediction than the 3rd-order reference model, and the comparison between compensations with the 3rd-order model A and 4th-order model IV proves this.

$$\begin{cases} \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_1\lambda_4 + \lambda_2\lambda_3 + \lambda_2\lambda_4 + \lambda_3\lambda_4 = a_2 \\ \lambda_1\lambda_2\lambda_3 + \lambda_2\lambda_3\lambda_4 + \lambda_1\lambda_2\lambda_4 + \lambda_1\lambda_3\lambda_4 = -a_1 \end{cases} \tag{4.56}$$

Table 4.8. Approximate expressions of the coefficients of the simplified state space predictor with a 4th-order reference model

Coefficient	Expression
c_0	$1-\frac{a_1}{6}t_d^3+\frac{a_1a_3-a_0}{24}t_d^4$
c_1	$t_d-\frac{a_2}{6}t_d^3+\frac{a_2a_3-a_1}{24}t_d^4$
c_2	$\frac{1}{2}t_d^2-\frac{a_3}{6}t_d^3+\frac{a_3^2-a_2}{24}t_d^4$
c_3	$\left(\frac{a_1}{6}t_d^3-\frac{a_3}{24}t_d^4\right)b_2+\frac{b_0}{24}t_d^4$
c_4	$\left(\frac{1}{6}t_d^3-\frac{a_3}{24}t_d^4\right)b_0$

The comparison between these two reference models may not be convincing because the 3rd-order model A is not an aircraft model. So, a 3rd-order model was formed by reducing the 4th-order model, such that the reduced-order model shares similar frequency characteristics with the original model. Then this reduced-order model was used to carry out the state space compensation. The compensation error was indeed considerably greater than that of the original 4th-order model.

It has been shown that the 4th-order reference model is superior to a 3rd-order one. But, what about a 2nd-order model, or a model of order higher than 4? If the reference model is of 2nd-order, the simplified state space filter is given as

$$y_p=\alpha_0 y+\alpha_1\dot{y}+(1-\alpha_0)u \tag{4.57}$$

where α_0 and α_1 are solutions of the coupled equations $\sum_{i=0}^{n-1}\alpha_i\lambda_j^i=e^{\lambda_j t_d}$, $(j=1,2)$,

where λ_0 and λ_1 are the two eigenvalues of the model. Eq. (4.57) is inferior to a reduced state space predictor with a 4th-order reference model because less system information is used for prediction. On the other hand, if a 5th-order reference model is employed, the filter state vector will contain either a high frequency jerk component (derivative of the acceleration) or triple integration of the aircraft state, which is likely to introduce computational artifacts. In short, the 4th-order reference model is the best choice.

As stated previously, the bandwidth of the reference model plays a major role in the compensation quality of the state space predictor. The bandwidth of a model is affected by its poles and zeros, but not by its gain. The gain of the reference model also influences the compensation quality. From Tables 4.6 and 4.8, the coefficient c_3 of the control input u is a linear function of the model gain. A large gain of the reference model will amplify the high frequency components in the control input u, and distort the prediction. However, the sensitivity of a reference model's compensation quality to the variation of its gain depends on the individual model; some models are more sensitive than others. The effects of the gain of a reference model are much less significant than the effects of its bandwidth.

The large commercial transport landing model in pitch (4th-order model IV) has been found to work successfully on data recorded from the large commercial transport simulation runining in the LaRC VMS. If the vehicle dynamics were quite different, e.g., a fighter rather than a transport, one possible method of finding a suitable reference model with which to apply the state space predictor would be to use system identification techniques to get a model that closely matches the aircraft dynamics. During the course of a simulation run, the dynamics may change dramatically. For instance, the dynamics

before and after an offset maneuver may be quite different. In such cases, fuzzy logic may be employed to identify the mode. Unfortunately, applying these algorithms increases the computation workload greatly. These issues merit further investigation.

In summary, the following points are conclusions of the study of the state space predictor, and the relationships between the compensation quality and the reference model:

- The state space predictor can be simplified to a general PID filter;
- The state space compensation may be viewed as a Taylor series extrapolation in the state space form;
- The bandwidth of the reference model must be close to that of the simulated vehicle;
- The damping ratio of the reference model does not make a significant difference in the state space compensation;
- The 4^{th}-order reference model is the best choice;
- The effects of the gain of a reference model are much less significant than the effects of its bandwidth.

5. Results of Theoretical Analysis

In Chapter 4, two novel predictors for compensating the transport delay in a flight simulator are compared from a theoretical perspective: the adaptive predictor based on the Kalman filter algorithm, and the state space predictor using an aircraft dynamic reference model. Chapter 4 demonstrated that these two new types of compensators show improved performance over the McFarland predictor. It also mentioned that the stochastic approximation adaptive predictor achieves the best compensation among the five different Kalman filter algorithms, and the large commercial transport landing model in pitch works best as a reference model for the state space predictor. All these conclusions are made without sufficient verification while focusing only on the principles of the novel compensators. In order to give a convincing comparison among all those compensators, quantitative approaches are necessary to analyze the respective compensation results. Because different predictors of the same type (i.e., the five adaptive predictors, and the state space predictors using different reference models) may demonstrate slight differences in compensation quality, the quantitative methods must be sensitive enough to highlight the differences and give accurate results.

This chapter starts by defining two metrics, which can be used to evaluate the compensation errors caused by different predictors. It then applies these metrics to the compensation results from offline tests, and finally quantitatively compares the compensation qualities among different predictors in terms of the two metrics. In the offline tests, compensations were applied to the aircraft state data recorded during a previously completed simulation study, generating predicted aircraft states. Offline tests

were implemented on a personal computer running MATLAB rather than the Visual Motion Simulator. No visual images were generated and no pilot was involved. The predicted aircraft states were used only for analysis. The last section investigates the sensitivity of the prediction errors with respect to the amount of time delay.

5.1. Error Metrics

Compensation error may appear as either phase error or magnitude error (gain distortion) or both. Fig 5.1 illustrates an example of phase error and magnitude error of a compensation applied to an ideal sinusoidal signal.

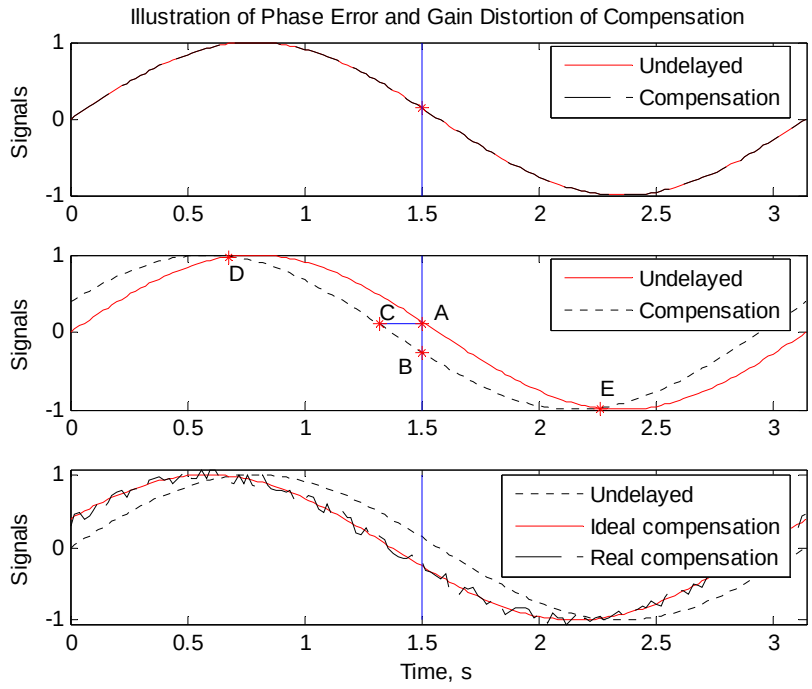


Fig. 5.1. Illustration of phase error and gain distortion of compensation

The function of a predictor is to generate a phase lead designed to be equal to the transport delay in the succeeding subsystem so that the delay can be compensated by the phase lead. In offline tests, the transport delay may be simulated by simply delaying the state information by the desired amount. In Fig. 5.1, the "compensation" signal is the

result of the delayed prediction. The ideal compensation produces no error (either phase error or distortion) at all, as depicted in the upper subplot of Fig. 5.1, where the compensated signal lines up exactly on top of the undelayed one. In other words, at every discrete point, the values of the two signals are exactly the same. Define an error function as the sum of square difference between the compensation and the undelayed signal

$$E = \sum_{i=1}^{N} \left[y_c(i) - y(i) \right]^2 \tag{5.1}$$

where y and y_c are the undelayed and compensated signals. This error function would be zero for the ideal compensation.

Assume that the predictor brings no gain distortion, but the phase lead it provides is not exactly equal to the phase lag resulting from the time delay. This is shown in the middle subplot of Fig. 5.1, in which the compensated signal does not match the undelayed signal. The phase error causes the predicted value to not match the undelayed signal at most time points (except those intersections D and E). For example, when t=1.5s, the phase error is $\overline{\mathrm{AC}}$, and the corresponding magnitude difference is $\overline{\mathrm{AB}}$. Because there is no gain distortion, the greater the phase error, the larger the magnitude discrepancy if the phase error is less than π (A phase lag of π results in an opposite phase, representing the maximal magnitude error *E*. As the phase changes from 0 to π, *E* increases, but as the phase lag exceeds π, *E* starts to decrease). Therefore, if there is no gain distortion, either the phase error or the error function defined by Eq. (5.1) may be used as a metric to evaluate the compensation error.

A real compensator usually results in both phase error and gain distortion, as shown in the dashed dot curve in the bottom subplot in Fig. 5.1. Evaluating the error

index *E*, as given by Eq. (5.1), between the dashed curve and the dashed dot curve, both phase and gain errors are present, but, the error index does not indicate which factor contributes more. For the transport delay compensation, the phase error usually is given more attention, and therefore it is necessary to define two metrics of error so that the effects of the two factors can be separated.

One way to separate the two is to fit the compensated signal with a smooth curve (such as the thick dashed curve in Fig. 5.2). Determine the phase difference between the fitted curve and the undelayed curve, and define this phase difference as the phase error of compensation. Then calculate the error index *E* between the actual signal (dashed) and the fitted one (dashed dot) using Eq. (5.1). In this example, the undelayed signal is a known function. Because the fitted signal is also analytical, it is easy to determine the phase difference between these two analytical functions by simply drawing a horizontal line intersecting both curves, and the distance between the two intersection points is the absolute value of the phase error.

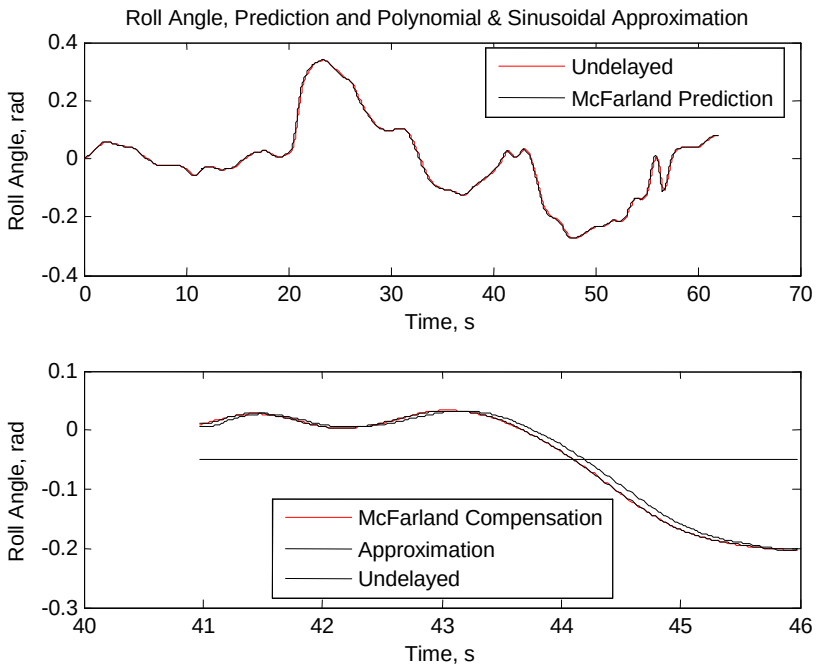


Fig. 5.2. Roll angle, prediction and polynomial & sinusoids approximation

In most situations, the real signal to be compensated is not an analytical function. In such situations, the undelayed signal also needs to be fitted with an analytical function. Because the actual aircraft state in a flight simulator is much more complicated than a sinusoidal signal, the fitting function is a combination of many simple components (a 4^{th}-order polynomial and 13 sinusoids of various magnitudes, frequencies and initial phases).

5.2. Comparison of Predictors Based on Offline Tests

Based on the analyses of the compensator used in the offline tests, in terms of the two error metrics defined in the previous section and the magnitudes of spikes on the compensations, this section will present a comparison of compensation qualities among:

1) The McFarland predictor and four adaptive predictors.
2) Five state space predictors using five different reference models (four 4^{th}-order models and one 3^{rd}-order model).

3) The McFarland predictor, the adaptive predictor using the stochastic approximation algorithm, and the state space predictor using the large commercial transport landing model in pitch.

5.2.1. Comparison of the McFarland Predictor and the Adaptive Predictors

These five predictors were applied to the aircraft roll angle data recorded from 16 straight-in approach tests. The predictors were designed to compensate for 48, 96, 192 and 288 ms (all integer multiples of the update period 16 ms) of transport delay, and in each time delay category, the amount of prediction (in ms) was averaged across the 16 test runs. Table 5.1 gives the average predictions and standard deviations for each case.

Among the four adaptive predictors, the least mean square algorithm demonstrates larger phase prediction error and a larger standard deviation than the other three. The basic Kalman filter, Kaczmarz and stochastic approximation algorithms show insignificant differences in phase prediction, whereas the stochastic approximation algorithm tends to have smaller standard deviation than the previous two when the transport delay is long (equal or greater than 192 ms). All four adaptive algorithms generate larger phase prediction error than the McFarland compensator, but apart from the least mean square algorithm, the differences are not significant. The standard deviations of the stochastic approximation algorithm and the McFarland predictor are very close, while the mean phase prediction error of the former is about 5 ms larger than that of the latter.

Table 5.1. Mean values & STD of the predictions with five 3-velocity predictors

t_d (ms)	Mean Prediction (LEFT) and Standard Deviation (RIGHT) (ms)									
	McFarland Predictor		Basic Kalman Predictor		Kaczmarz Algorithm		Stochastic Approximation		Least Mean Square	
48	48	1	43	1	43	1	43	1	41	5
96	96	2	93	3	93	2	93	2	90	8
192	190	5	185	8	191	15	187	6	179	18
288	283	9	271	14	275	14	278	10	266	24

Table 5.2 lists the compensation gain errors defined in Eq. (5.1) of these five predictors averaged across the same 16 test runs. Among the four adaptive algorithms, the basic Kalman filter algorithm causes the greatest gain error, and the stochastic approximation algorithm causes the least. Apart from the stochastic approximation algorithm, all the other adaptive algorithms show larger gain distortion than the McFarland predictor. As mentioned in Chapter 4, the McFarland predictor generates noticeably larger spikes than all the adaptive predictors, and this indicates the error measure in Eq. (5.1) is a more useful metric of gain error than the magnitude of spikes. The former is defined for a whole test, whereas the spikes only exist locally. In other words, greater total gain error does not mean larger spikes. If the transport delay is short, the gain error of compensation by the stochastic approximation algorithm is even greater than the McFarland predictor, but as the time delay becomes longer, the former displays smaller gain distortion.

Table 5.2. Gain error index of the McFarland and four adaptive predictors

t_d (ms)	Gain Error Index				
	McFarland Predictor	Basic Kalman Predictor	Kaczmarz Algorithm	Stochastic Approximation	Least Mean Square
48	0.0416	0.3818	0.3731	0.2705	0.4763
96	0.7735	2.5586	2.3050	1.0363	1.9108
192	17.6686	37.2153	33.1492	13.9038	22.8639
288	74.5940	171.2643	152.5877	59.4277	95.3306

The motive for developing the adaptive predictors is to reduce the large spikes and gain distortion of the McFarland predictor. The analyses given above of Table 5.1 and Table 5.2 demonstrate that the stochastic approximation algorithm is superior to the other three adaptive algorithms. Therefore, an adaptive predictor using the stochastic algorithm is the best among the five in this group.

5.2.2. Comparison of Five State Space Predictors

In Chapter 4, five aircraft reference models for the state space predictor were introduced, i.e., four 4^{th}-order models: the pitch model (Model I) and the roll model (Model II) of a fixed wing jet flying at an altitude of 30,000ft and an airspeed of 430 knots, the cruise model of the large commercial transport in pitch (Model III) and the landing model of the large commercial transport in pitch (model IV), and a 3^{rd}-order model. The gain errors, phase predictions and standard deviations of compensation by these five predictors for the four transport delay cases are listed in Table 5.3 and Table 5.4, respectively. While Model III introduces the least gain distortion, the difference in terms of gain error between Models III and IV (the two large commercial transport models) are negligible; the two fixed-wing jet models introduce very similar gain error

(the pitch model brings slightly smaller gain error than the roll model), but the gain distortion is obviously larger than that of the two large commercial transport models; and the 3rd-order model introduces the largest gain distortion.

Table 5.3. Gain error index of the state space predictors with five reference models

t_d (ms)	Gain Error Index				
	4th-order Model I	4th-order Model II	4th-order Model III	4th-order Model IV	3rd-order Model A
48	0.0071	0.0080	0.0062	0.0062	0.0542
96	0.1378	0.1573	0.1063	0.1079	0.7374
192	4.2312	4.5971	2.8558	3.0095	9.9674
288	28.5543	30.4164	16.1325	17.5617	44.4455

Table 5.4. Mean values & STD of state space prediction with five reference models

t_d (ms)	Mean Prediction (LEFT) and Standard Deviation (RIGHT) (ms)									
	4th-order Model I		4th-order Model II		4th-order Model III		4th-order Model IV		3rd-order Model A	
48	48	1	48	1	48	1	48	1	48	1
96	94	2	95	2	95	2	96	2	95	3
192	175	5	184	10	188	4	192	4	186	8
288	236	10	266	31	275	8	287	5	273	16

Model IV yields the least mean phase prediction error and the least standard deviation (from 16 test runs). For delay up to 192 ms, the mean phase is zero, and for a delay of 288 ms, the phase error is only 1 ms. Models II, III and the 3rd-order model introduce very similar mean phase errors (about 1 frame for a delay of 288 ms), though the standard deviation of Model II is significantly greater than the other two when time delay is 288 ms; Model I introduces the greatest mean phase error among these five models.

Therefore, in both gain distortion and phase error, the large commercial transport landing model in pitch (Model IV) works best compared with the other four reference models. In general, the 3rd-order model is inferior to the 4th-order models.

5.2.3. Comparison of the McFarland, the Adaptive and the State Space Predictors

From the comparisons in the previous two sections, the stochastic approximation algorithm is the best adaptive predictor, and the large commercial transport landing model in pitch is the best candidate as the reference model for the state space predictor among the five models. Thus, these two novel predictors were chosen, to be implemented along with the McFarland compensator in the final piloted tests, in order to compare them in a human-in-the-loop simulation. In this section, the performance of these three compensators will be compared with offline tests. Though the data are available in Tables 5.1 – 5.4, the mean phase prediction and gain error specifically for these three predictors were extracted and are listed in Table 5.5. As an example, Table 5.3 illustrates the phase compensations of these three.

Table 5.5. Mean values & STD of the predictions of three predictors

t_d (ms)	Mean Prediction (LEFT, ms) and Gain Error (RIGHT)					
	McFarland Predictor		Adaptive Predictor		State Space Predictor	
48	48	0.0416	43	0.2705	48	0.0062
96	96	0.7736	93	1.0363	96	0.1079
192	190	17.6686	188	13.9038	192	3.0085
288	283	74.5940	278	59.4277	287	17.56175

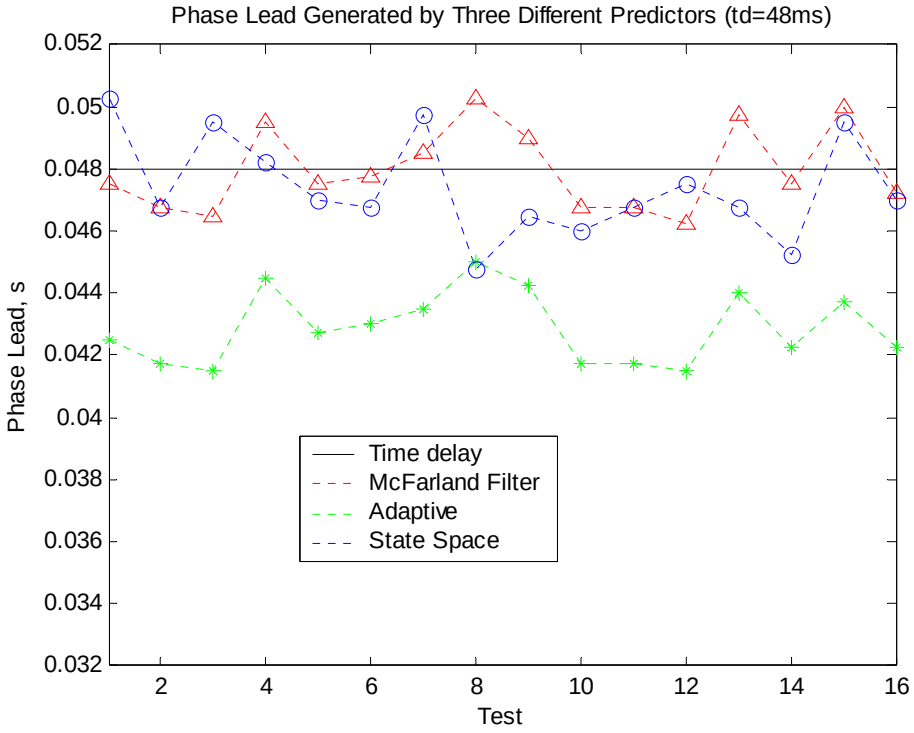


Fig. 5.3. Phase lead generated by three types of predictors (t_d =192 ms)

The state space predictor tends to introduce significantly less gain distortion than the other two. It causes less phase prediction error than the adaptive predictor, and achieves slightly better phase compensation than the McFarland predictor for long time delays. It seems to be superior to the adaptive predictor, but it introduces moderately larger spikes than the latter. The phase prediction error of the adaptive predictor is larger than that of the McFarland predictor, but not by much. When the delay is longer than 192

ms, the adaptive predictor significantly reduces the high frequency artifacts, especially the spikes generated by the McFarland compensator, resulting in the reduction of gain error. Although the McFarland predictor shows the least phase error, which is determined using a smooth fitting curve, the actual phase error in areas of large gain distortion may be even larger than that of the other two predictors.

5.3. Sensitivity Analysis

As the transport delay to be compensated gets longer, there is no doubt that both the gain distortion and phase error increase. But how fast the errors increase for different compensators, or in other words, how sensitive they are to changes in time delay, is still an interesting problem. This section is an investigation of the sensitivity of three compensators: the McFarland predictor, the adaptive predictor and the state space predictor.

In order to complete this analysis, the group of compensators was implemented, and tested on the 16 sets of aircraft roll angle data recorded from piloted simulations. For this test, the transport delay was varied from 48 ms to 288 ms, and the frame time was 16 ms. The phase prediction and gain error were calculated, and the results are summarized in Table 5.6, which is actually an expansion of Table 5.5.

The phase prediction of the state space predictor is insensitive to changes in time delay, whereas the McFarland predictor and the adaptive predictor are sensitive to changes in time delay to a similar degree (from 0 to 5 ms for the former, and from 5 to 10 ms for the latter, as the delay increases from 48 to 288 ms). In terms of absolute increase of the gain error, the state space predictor is the least sensitive to the time delay, and the McFarland is the most sensitive (the absolute gain error increases of these three

predictors are, in descending order, 74.4524, 59.1572 and 17.5556). But in terms of relative gain error increase, the adaptive predictor is least sensitive to the change of time delay, while the state space predictor is most sensitive. The relative gain errors increase by 1791, 220 and 2833 times respectively for the McFarland predictor, the adaptive predictor and the state space predictor. But, the most important metric is the absolute gain error increase, and the state space predictor is the least sensitive to the time delay, whereas the McFarland predictor is the most sensitive in both the phase and gain errors of prediction.

Table 5.6. Mean predictions and gain error index of the McFarland predictor, an adaptive predictor and a state space predictor ($t_d \in$ [48, 288] ms)

t_d (ms)	Mean Prediction (LEFT, ms) and Gain Error (RIGHT)					
	McFarland Predictor		Adaptive Predictor		State Space Predictor	
48	48	0.0416	43	0.2705	48	0.0062
64	64	0.1127	93	0.2786	64	0.0099
80	80	0.3202	77	0.5211	80	0.0402
96	96	0.7736	93	1.0363	96	0.1079
112	112	1.6210	109	1.8812	112	0.2332
128	128	3.0330	125	3.1277	128	0.4477
144	144	5.1968	141	4.8614	144	0.7962
160	160	8.2767	156	7.1684	160	1.3126
176	175	12.4131	171	10.1465	176	2.0359
192	191	17.6686	187	13.9038	192	3.0095
208	206	24.1493	202	18.5569	208	4.2808
224	222	30.1740	217	20.4700	224	8.6872
240	237	40.8535	232	31.0103	240	7.9571
256	253	50.9008	248	39.0578	255	10.5185
272	268	62.1538	262	48.4900	272	13.6921
288	283	74.4940	278	59.4277	287	17.5618

In addition, as the time delay becomes longer, all three predictors tend to be more sensitive to the time delay. For instance, for the McFarland predictor, the gain error index increases only 0.0711 as the delay increases 1 frame from 48 ms, but it increases 12.3402 as the delay increases 1 frame from 272 ms. In other words, the prediction error does not increase linearly with the transport delay.

6. Conclusions and Future Research

6.1. Conclusions

Theoretical analyses in both time domain and frequency domain show that the effects of the transport delay depend on the simulation dynamics. Specifically, the higher the system bandwidth, the larger the loss of the phase margin, or stability. Meanwhile, the system suffers with the same amount of transport delay, and therefore, the system has less ability to tolerate the time delay. In a flight simulator system, higher bandwidth means the system is more vulnerable to pilot-induced oscillation and instability.

Measurements were made with a device called SIMES to measure the transport delay in the Visual Motion Simulator (VMS) at the NASA Langley Research Center. The transport delays in the visual system, the motion system and the instrument system were measured to be 58, 56 and 40 ms, respectively, with the frame cycle being 16.7 ms. The average total transport delays from the pilot control input to the ends of these three sub-systems were determined to be 89.7, 87.7 and 71.7 ms, with the frame length of the simulation computer being 16 ms. Therefore, the average baseline transport delay of the visual cue is 90ms (rounded from 89.7ms) . This forms the basis for the time delay compensator designs for the final piloted simulation tests.

Three basic criteria (phase error, gain distortion and computation complexity) were adopted to assess and compare the three prominent, previously developed time delay compensators—the lead/lag filter, McFarland predictor and the Sobiski/Cardullo predictor. This was done using theoretical simulations with the same aerodynamic model and pilot model as used for the analysis of time delay in Chapter 2. For the lead/lag filter,

the formula proposed by Crane to design the filter pole was revised, resulting in better compensation. By applying the McFarland compensator to the recorded simulation Euler angles, large spikes in the compensated output resulting from its gain distortion have been found and determined to be the primary cause of the poor compensation observed in the preliminary piloted tests. Investigation shows that the excessive gain distortion comes from the alternative sign changes of the three coefficients of the McFarland predictor, and from the constant coefficients, which do not change with the simulation process. For the Sobiski/Cardullo predictor, a discrete version was formulated which yields the same compensation results as the former in theoretical simulation. Some limitations in Sobiski's implementation were found.

A spike-reduction algorithm was proposed as an expedient solution to the annoying large spikes in the McFarland compensation. Both time and frequency domain least squares methods were formulated to design the coefficients of a three-velocity predictor. A novel adaptive predictor was developed using the Kalman filter algorithm that updates the predictor coefficients during the course of a simulation. Five versions of the adaptive algorithms were tested and compared, the stochastic approximation algorithm was chosen for the final piloted tests because it produces the smallest compensation errors. A stochastic approximation algorithm with a forgetting factor was also formulated. The analysis in Chapter 4 mathematically proves the viability of using the asymptotic ODE, and demonstrates why the stochastic approximation algorithm achieves the best compensation among all the five adaptive predictors.

The first practical state space predictor to compensate the transport delay was developed by forming a group of predictor states and making use of a linear aerodynamic

reference model. This makes it possible to apply a state space compensator to a flight simulator in which the aircraft dynamics are nonlinear and time-variant. The state space predictor was significantly simplified from the original Sobiski/Cardullo implementation, and reduced to a general PID filter. Several 3rd and 4th-order dynamic reference models were tested and compared, and the large commercial transport landing model in pitch was chosen for the final piloted tests. A mathematical explanation of why this model is the best reference model for use in the state space compensation was discussed. While exploring the relationships between the state space compensation effectiveness and the reference model, the following important points were found:

1) The state space compensation may be viewed as a Taylor series extrapolation in the state space form;
2) The bandwidth of the reference model must be close to that of the simulation dynamics;
3) The damping ratio of the reference model does not make a significant difference in the state space compensation;
4) The 4th-order reference model is the best choice in this case;
5) The effects of the gain of a reference model are much less significant than its bandwidth.

Theoretical analyses were conducted by applying the McFarland predictor and the two novel predictors to recorded Euler angles from past simulation tests, and evaluating their phase and gain errors. The adaptive predictor with the stochastic approximation algorithm produces slightly higher phase error than the McFarland predictor, which reveals about the same phase error as the state space predictor with the large commercial

transport landing dynamics as the reference model. The difference in phase errors of the three predictors is not significant, and therefore the gain errors play a more important role in deciding their relative effectiveness. The adaptive predictor yields a greater gain error than the McFarland predictor for the zero and 48 ms added delays (short delays), but shows a smaller error for the 96 and 192 ms added delays (long delays). The advantages of the adaptive predictor become more obvious for longer time delay. Conversely, the state space predictor results in substantially smaller gain error than the other two predictors for all the four delay cases.

6.2. Suggested Future Research

It would be worthwhile to study a frequency domain method to measure the simulator transport delay and phase lag, and compare the results with those obtained with the time domain method. Although some frequency domain data were collected with the SIMES while it generated sweep signals as control inputs to the VMS, they did not result in satisfactory results because aliasing was included in the aircraft EOM outputs. The frequency of the sweep signal was increased too quickly, and the lower frequency input did not last long enough. A high percentage of high frequency input signals caused the aliasing. This could be avoided by choosing a lower rate of increasing the frequency of the sweep signal.

The effectiveness of the three-velocity adaptive predictor for compensating a short time delay may be improved by updating the coefficients in a more “intelligent” manner, which is more adaptable to the velocity changes, especially sudden and abrupt changes, such as the initiation of a maneuver. An exponential forgetting factor can only weigh the past data continuously and gradually, and therefore cannot solve this problem.

It would be worthwhile to develop some “intelligent” algorithms that can detect significant changes in the flight dynamics. Fuzzy logic and neural network techniques may be helpful to achieve the detection of these changes. Maneuver phase detection would be of great significance, and may be useful in applications other than transport delay compensation.

The compensation quality of a state space predictor depends on the choice of the reference aircraft model. As is stated in Chapter 4 (4.5.4), system identification may be a good way to obtain an optimal reference model. To avoid adding extra delay resulting from a highly complex aircraft model, techniques to minimize the computational burden should be investigated. Because different phases of a maneuver may involve varying-order dynamics, system identification may also require maneuver phase detection.

Appendix A. Calculation of State Transition Matrix and Its Convolution Integral

A.1. State Transition Matrix

Computation of the state transition matrix $\mathbf{\Phi}$ and the convolution integral matrix $\mathbf{\Psi}$ of the state matrix $\mathbf{A}$ plays an important role in implementing the Sobiski/Cardullo filter and the aerodynamic reference model state space predictor. These matrices can be calculated directly or indirectly. The direct method makes use of the definition of the state transition matrix, and the indirect method makes use of the eigenvalues of the matrix $\mathbf{A}$, or the inverse Laplace transform.

A.1.1. Direct Method

By definition, Eq. (4.49), the state transition matrix $\mathbf{\Phi}$ is an infinite Taylor series

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} = \sum_{i=0}^{\infty} \frac{\left(\mathbf{A}t_d\right)^i}{i!} \tag{A.1}$$

Because this series is always convergent, the infinite series can be approximated with a finite number of terms. The number of terms required depends on the criterion set for the accuracy. If the power of the matrix $\mathbf{A}$ is equal to or higher than its order n, this power can be calculated in terms of those lower-order powers by making use of the Cayley-Hamilton theorem

$$\phi\left(\mathbf{A}\right) = \sum_{i=0}^{n} a_i \mathbf{A}^i = \mathbf{0} \tag{A.2}$$

where $a_0 - a_n$ are the coefficients of the characteristic equation of $\mathbf{A}$. This shows that the $\mathbf{A}^n$ can be calculated in terms of $\mathbf{A}, \mathbf{A}^2, ..., \mathbf{A}^{n-1}$, i.e.,

$$\mathbf{A}^n = -\left(a_0\mathbf{I} + a_1\mathbf{A} + ... + a_{n-1}\mathbf{A}^{n-1}\right) \tag{A.3}$$

If the matrix $\mathbf{A}$ is obtained from a transfer function, its characteristic equation is known from the denominator of the transfer function. If it is obtained from another approach, say system identification for example, the coefficients $a_0 - a_n$ of the characteristic equation may be given by

$$a_{n-1} = -tr\mathbf{A}, a_{n-2} = -\frac{1}{2}tr\mathbf{A}\mathbf{T}_{n-2}, ..., a_0 = -\frac{1}{n}tr\mathbf{A}\mathbf{T}_0 \tag{A.4}$$

where the matrices $\mathbf{T}_1$ - $\mathbf{T}_{n-1}$ can be given as follows

$$\mathbf{T}_{n-2} = \mathbf{A} + a_{n-1}\mathbf{I}, \mathbf{T}_{n-3} = \mathbf{A}\mathbf{T}_{n-2} + a_{n-2}\mathbf{I}, ..., \mathbf{T}_0 = \mathbf{A}\mathbf{T}_1 + a_1\mathbf{I} \tag{A.5}$$

High matrix power can also be avoided by a recursive algorithm, which may be obtained by rewriting Eq. (A.1) in the form of

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{A}t_d\left\{\mathbf{I} + \mathbf{A}t_d/2\left[\mathbf{I} + \mathbf{A}T/3\left(\mathbf{I} + \mathbf{A}t_d/4\left(\mathbf{I} + ... + \left(\mathbf{I} + \mathbf{A}t_d/n\right)\right)\right)\right]\right\} \tag{A.6}$$

The recursion must be carried out in an inverse order, e.g., from $\mathbf{I} + \mathbf{A}t_d/N$, to $\mathbf{I} + \mathbf{A}t_d/N-1$, through to $\mathbf{I} + \mathbf{A}t_d$. Thus the number of retained terms needs to be determined before hand, and the criterion for truncation may be that $\left(\mathbf{A}t_d\right)^2 / N\left(N+1\right)$ is acceptably small when compared to $\mathbf{A}t_d / N$.

A.1.2. Indirect Method

The Cayley-Hamilton theorem, Eq. (A.2), says that every square matrix satisfies its own characteristic equation. This leads to the indirect calculation of the state transition matrix in that the infinite series in Eq. (A.1) can be rewritten as a product of $\phi(\mathbf{A})$ and a polynomial in $\mathbf{A}$ plus a remainder, that is

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} = P(\mathbf{A})\phi(\mathbf{A}) + R(\mathbf{A}) \tag{A.7}$$

where the remainder $R(\mathbf{A})$ is of lower order than $\phi(\mathbf{A})$. The counterpart of Eq. (A.7) in terms of a scalar variable is given by

$$e^{x} = P(x)\phi(x) + R(x) \tag{A.8}$$

From the Cayley-Hamilton theorem, $\phi(\mathbf{A}) = \mathbf{0}$, then

$$\begin{aligned} \mathbf{\Phi} = e^{\mathbf{A}t_d} &= R(\mathbf{A}) \\ &= \alpha_0\mathbf{I} + \alpha_1\mathbf{A} + ... + \alpha_{n-1}\mathbf{A}^{n-1} \end{aligned} \tag{A.9}$$

The coefficients $\alpha_0, \alpha_1, ..., \alpha_{n-1}$ can be calculated by noting that every eigenvalue of $\mathbf{A}$ satisfies $\phi(\lambda_i) = 0$, substituting this into Eq. (A.8) gives

$$e^{\lambda_i t_d} = R(\lambda_i) = \alpha_0 + \alpha_1\lambda_i + ... + \alpha_{n-1}\lambda_i^{n-1} \tag{A.10}$$

If the matrix $\mathbf{A}$ has all identical eigenvalues, the n equations in (A.10) can be solved for the coefficients $\alpha_0, \alpha_1, ..., \alpha_{n-1}$. Solving the coupled equation group involves inverting a Vandermonte matrix

$$\mathbf{V} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{n-1} \end{bmatrix} \tag{A.11}$$

The solution is

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} = \mathbf{V}^{-1} \begin{bmatrix} e^{\lambda_1 t_d} \\ e^{\lambda_2 t_d} \\ \vdots \\ e^{\lambda_n t_d} \end{bmatrix} \tag{A.12}$$

If there are repeated eigenvalues, with a multiplicity of m ($m \le n$), then $(m-1)$ more equations are needed, and they are given by

$$\left.\frac{d^j}{d\lambda^j}\left(e^{\lambda t_d}\right)\right|_{\lambda=\lambda_0} = \left.\frac{d^j}{d\lambda^j}\left(\alpha_0+\alpha_1\lambda_i+\ldots+\alpha_{n-1}\lambda_i^{n-1}\right)\right|_{\lambda=\lambda_0}, j=1,2,\ldots,m-1 \tag{A.13}$$

If a pair of eigenvalues of complex conjugates exists, the resulting state transition matrix is real because the two complex equations can be replaced with two real equations obtained from the real and imaginary equalities. For example, for $\lambda = p \pm jq$, the two complex equations $e^{\lambda t_d} = R(\lambda) = \alpha_0 + \alpha_1\lambda + \ldots + \alpha_{n-1}\lambda^{n-1}$ can be separated into two real equations

$$\begin{cases} e^{pt_d}\cos(qt_d) = real\left[\alpha_0+\alpha_1(p+jq)+\ldots+\alpha_{n-1}(p+jq)^{n-1}\right] \\ e^{pt_d}\sin(qt_d) = imag\left[\alpha_0+\alpha_1(p+jq)+\ldots+\alpha_{n-1}(p+jq)^{n-1}\right] \end{cases} \tag{A.14}$$

Another indirect approach for calculating the state transition matrix is to use the inverse Laplace transform,. which is given by

$$\mathbf{\Phi} = e^{\mathbf{A}t_d} = L^{-1}\left\{\left[s\mathbf{I}-\mathbf{A}\right]^{-1}\right\}\Big|_{t=t_d} \tag{A.15}$$

Inversion of $\left[s\mathbf{I}-\mathbf{A}\right]$ can be computed using the expansion of its adjoint as follows

$$\left[s\mathbf{I}-\mathbf{A}\right]^{-1} = \frac{adj\left(s\mathbf{I}-\mathbf{A}\right)}{\left|s\mathbf{I}-\mathbf{A}\right|} \tag{A.16}$$

where $\left|s\mathbf{I}-\mathbf{A}\right|$ is the characteristic polynomial:

$$\left|s\mathbf{I}-\mathbf{A}\right| = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s + a_0 \tag{A.17}$$

The coefficients $a_0 - a_n$ can be calculated with Eq. (A.4). The adjoint matrix may be given as

$$adj\left(s\mathbf{I}-\mathbf{A}\right) = \mathbf{I}s^{n-1} + \mathbf{T}_{n-2}s^{n-1} + \ldots + \mathbf{T}_1 s^1 + \mathbf{T}_0 \tag{A.18}$$

where the matrices $\mathbf{T}_1 - \mathbf{T}_{n-1}$ are given by Eq. (A.5)

A.2. Calculation of the State Transition Matrix Integral

By definition, the integral of the state transition matrix is given by

$$\boldsymbol{\Psi} = \int_0^{t_d} e^{\mathbf{A}(t_d-\tau)} d\tau \tag{A.19}$$

If the matrix $\mathbf{A}$ is non-singular, it can be calculated by direct integration

$$\boldsymbol{\Psi} = \mathbf{A}^{-1}\left(e^{\mathbf{A}t_d} - \mathbf{I}\right) = \mathbf{A}^{-1}\left(\boldsymbol{\Phi} - \mathbf{I}\right) \tag{A.20}$$

But if $\mathbf{A}$ is singular, this method cannot be used because its inverse matrix does not exist. Matrix inversion can be avoided by substituting the definition of $\boldsymbol{\Phi}$ into Eq. (A.20), which then becomes

$$\begin{aligned}\boldsymbol{\Psi} &= \mathbf{A}^{-1}\left(\mathbf{I} + \mathbf{A}t_d + \frac{1}{2!}\left(\mathbf{A}t_d\right)^2 + \ldots + \frac{1}{n!}\left(\mathbf{A}t_d\right)^n + \ldots - \mathbf{I}\right) \\ &= t_d \sum_{i=0}^{\infty} \frac{\left(\mathbf{A}t_d\right)^i}{(i+1)!}\end{aligned} \tag{A.21}$$

The repetition of completing the factorial may be avoided by using a recursion algorithm

$$\boldsymbol{\Psi} = t_d\left\{\mathbf{I} + \mathbf{A}t_d/2\left[\mathbf{I} + \mathbf{A}T/3\left(\mathbf{I} + \mathbf{A}t_d/4\left(\mathbf{I} + \ldots + \left(\mathbf{I} + \mathbf{A}t_d/n\right)\right)\right)\right]\right\} \tag{A.22}$$

Using the recursion in a manner similar to the implementation of Eq. (A.6). The calculation of the matrix $\boldsymbol{\Psi}$ is very similar to calculation of $\boldsymbol{\Phi}$. The power functions of the matrix $\mathbf{A}$ used when forming $\boldsymbol{\Phi}$ with the direct method may be saved for later calculation of $\boldsymbol{\Psi}$.

If the order of $\mathbf{A}$ is not high, the method given by Eq. (A.20) is simpler than the recursive algorithm (A.22) when the matrix $\boldsymbol{\Phi}$ is already available. However, if $\mathbf{A}$ is of high order, algorithm (A.20) becomes complicated, because inversion of a high order matrix is time consuming. In this case, use of the recursive algorithm is recommended.

A.3. Calculation of the Discrete State Transition Matrix

The calculation of the discrete state transition matrix $\mathbf{\Phi}_d = \mathbf{G}^l$ is straightforward, however, it can also be obtained using the inverse Z-transform, as follows

$$\mathbf{\Phi}_d = \mathbf{G}^l = Z^{-1}\left\{\left[z\mathbf{I} - \mathbf{G}\right]^{-1} z\right\}\Big|_{k=l} \tag{A.23}$$

Appendix B. Complements to Sobiski/Cardullo Filter

Sobiski and Cardullo first introduced the idea of using the state transition matrix to compensate the time delay in flight simulation. The state space predictive filter named after them bears many unique features for designing and implementing state space compensation. This section presents a brief description of the Sobiski/Cardullo predictive filter and some additional work done by the author of this report.

B.1. State Space Compensation: Output Feedback and State Feedback

Figure 2.3 is a block diagram of the flight simulation system, which includes the Sobiski/Cardullo compensator as it was implemented for this study. The transfer functions of the operator model, the aircraft dynamics and the Pade approximation of the time delay were first transformed to state space equations and output equations, which were then concatenated together, resulting in a single-input and single-output (SISO) linear time-invariant system, represented by

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} \end{cases} \quad (D = 0).$$

Because the Pade approximation of the time delay is included, the output y has been delayed, and the state space prediction is to be employed to compensate for the delay. The matrices $\mathbf{A}$, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ all contain information about the time delay, and are of 10^{th}-order. The state space compensation may be implemented as depicted in Fig. B.1. The compensated system is described by the equations

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{\Phi}\mathbf{x} + \mathbf{C}\mathbf{\Psi}\mathbf{B}u \end{cases} \tag{B.1}$$

The matrix $D = \mathbf{C}\mathbf{\Psi}\mathbf{B}$, a scalar now (a one-by-one matrix), is not necessarily zero.

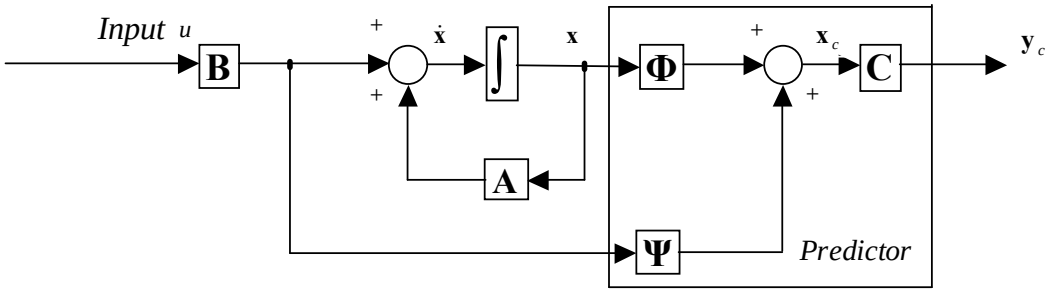


Fig. B.1. State space compensation in an open loop system

The unit output feedback closed-loop system with state compensation is shown in Fig. B.2, and the state equation and output equation of the closed loop compensated system are

$$\begin{cases} \dot{\mathbf{x}} = \left(\mathbf{A} - \mathbf{B}G\mathbf{C}\Phi\right)\mathbf{x} + \mathbf{B}Gv \\ y = \mathbf{C}\Phi G\mathbf{x} + \mathbf{C}\Psi\mathbf{B}Gv \end{cases} \tag{B.2}$$

where $G = \left(1 + \mathbf{C}\Psi\mathbf{B}\right)^{-1}$ is, as Sobiski and Cardullo defined, the feed forward gain of the closed loop compensated system.

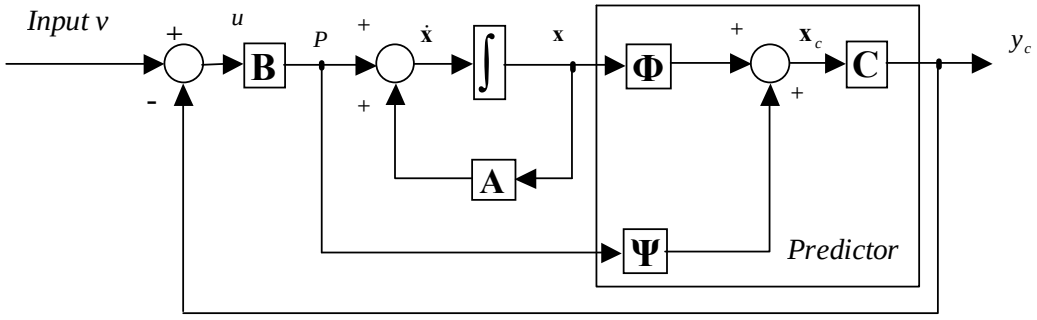


Fig. B.2. Output feedback closed loop system with state space compensation

The state feedback can also be used in a state transition matrix-based compensation. Actually the Sobiski/Cardullo predictive filter uses the state feedback, as illustrated in Fig. B.3. If the state feedback is used, the feed forward gain matrix becomes $G = \left(1 + \mathbf{K}\Psi\mathbf{B}\right)^{-1}$. The feedback matrix $\mathbf{K}$ is designed using a pole placement technique so that the closed loop poles of the compensated system consist of the closed loop poles

of the undelayed system and the real parts of the Pade approximation of the time delay. The first step is to obtain $\mathbf{K}_{fdbk}$ using ordinary pole placement so that the homogeneous equation $\dot{\mathbf{x}} = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_{fdbk}\right)\mathbf{x}$ has the desired poles. The second step is to calculate the feedback gain $\mathbf{K}$ as follows: comparing $\dot{\mathbf{x}} = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_{fdbk}\right)\mathbf{x}$ with the first equation of (B.2) gives $\mathbf{K}_{fdbk} = G\mathbf{K}\boldsymbol{\Phi}$, and the equations $G = \left(1 + \mathbf{K}\boldsymbol{\Psi}\mathbf{B}\right)^{-1}$ and $\mathbf{K}_{fdbk} = G\mathbf{K}\boldsymbol{\Phi}$ together give

$$\mathbf{K} = \mathbf{K}_{fdbk}\left(\boldsymbol{\Phi} - \boldsymbol{\Psi}\mathbf{B}\mathbf{K}_{fdbk}\right)^{-1} \tag{B.3}$$

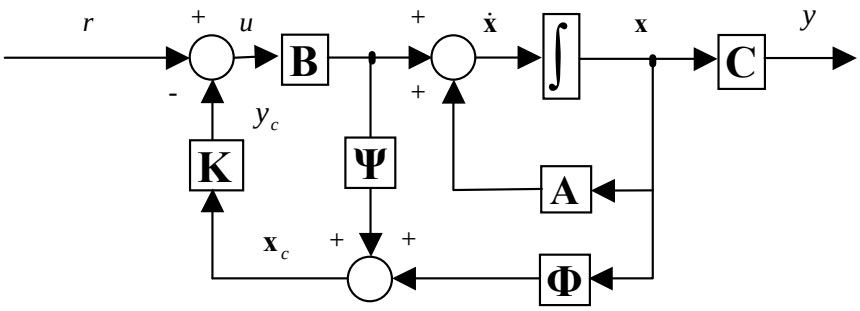


Fig. B.3. State feedback closed loop system with state space compensation

Because the real parts of the 2nd-order Pade approximation are identical, the desired closed loop poles for the pole placement have repeated elements, and this prevents the MATLAB "place" function from working. A new MATLAB function has been written to achieve pole placement with repeated poles (Available in Appendix C.14, NASA CR—2007-215096)

The final Sobiski/Cardullo filter is a little different from Fig. B.3, and is actually an equivalent system obtained using the feed forward gain G . The equivalent system is shown in Fig. B.4. It can be proved that the systems in Fig. B.3 and Fig. B.4 share the same closed loop transfer function

$$\frac{Y_c(s)}{R(s)} = \frac{\mathbf{K\Phi}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}G}{1+\mathbf{K\Phi}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}G} \tag{B.4}$$

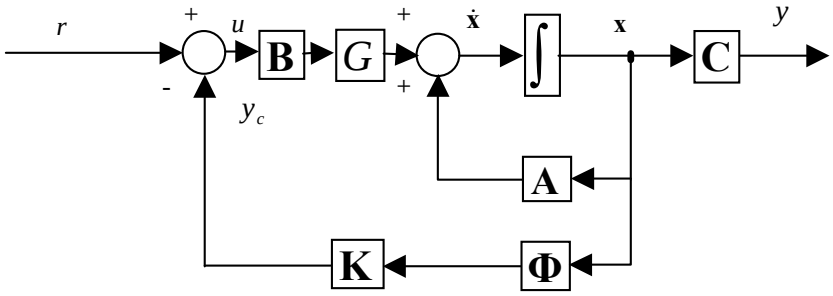


Fig. B.4. The Sobiski/Cardullo filter

One advantage of the equivalent system shown is that the compensated system may be easily simplified to canonical form because the matrix **Ψ** in the initial state feedback system (Fig. B.4) is changed to the feed forward scalar gain G. However, these two systems are not exactly equivalent although the feedback y_c is the same, the system state **x** is different, and hence the system output y is also different. Comparing the two systems shows that they are identical only if the feed forward gain G is unity. Because G is not unity when using the pole placement method, the system states are changed. In an aircraft simulation, the aircraft states may be used by different cueing systems. With the equivalent system, although the operator is fed back with the correct, compensated visual cue, the instrument cue and motion cue (if any) are not correct. This is a limitation of Sobiski/Cardullo compensation.

When the system is of high order, e.g., above the 10^{th}-order, pole placement may sometimes cause problems. The state feedback matrix **K** can be designed using other approaches. Two frequency domain methods have been used: the five-point method and the least squares method. Both methods assume the operator working frequency does not

exceed an upper limit ω_0. In the five-point approach, five frequency points are chosen evenly from the pass band $[0\ \omega_0]$, and the frequency characteristic function of the compensated system is forced to equal that of the undelayed system at these points. It is given by

$$\mathbf{K}\left\{\mathbf{\Phi}\left[j\omega_i\mathbf{I}-\mathbf{A}\right]^{-1}\mathbf{B}+\mathbf{\Psi}\mathbf{B}\right\}=H_0\left(j\omega_i\right),\ \left(i=1,...,5\right) \tag{B.5}$$

where H_0 is the Laplace transfer function of the undelayed system. The five real part equalities and five imaginary part equalities can be used to solve the 10 elements of the matrix $\mathbf{K}$.

In the least squares approach, more than five frequency points are chosen, and the sum of the Euclidean norm of the difference between the frequency characteristic functions of the compensated system and the undelayed system at these points is minimized. In mathematical terms, minimize the equation

$$S=\sum_{i=1}^{n}\left\|\mathbf{K}*\left\{\mathbf{\Phi}\left[j\omega_i\mathbf{I}-\mathbf{A}\right]^{-1}\mathbf{B}+\mathbf{\Psi}\mathbf{B}\right\}-H_0\left(j\omega_i\right)\right\|^2 \tag{B.6}$$

Then, setting the derivatives of S with respect to each element of the matrix $\mathbf{K}$ to zero results in 10 coupled linear algebraic equations that can be used to solve $\mathbf{K}$. Analysis shows that the state feedback matrix designed with the least squares methods can achieve better compensation than with the five-point method and much better than with the pole placement method. Figs. B.5 and B.6 show the Bode diagrams and step responses of the state space compensation with the feedback matrix $\mathbf{K}$ designed by the least squares fitting (LSF) method for delays of 200, 400 and 800 ms. The compensation is perfect and the responses of the compensated system line up exactly on top of the uncompensated response.

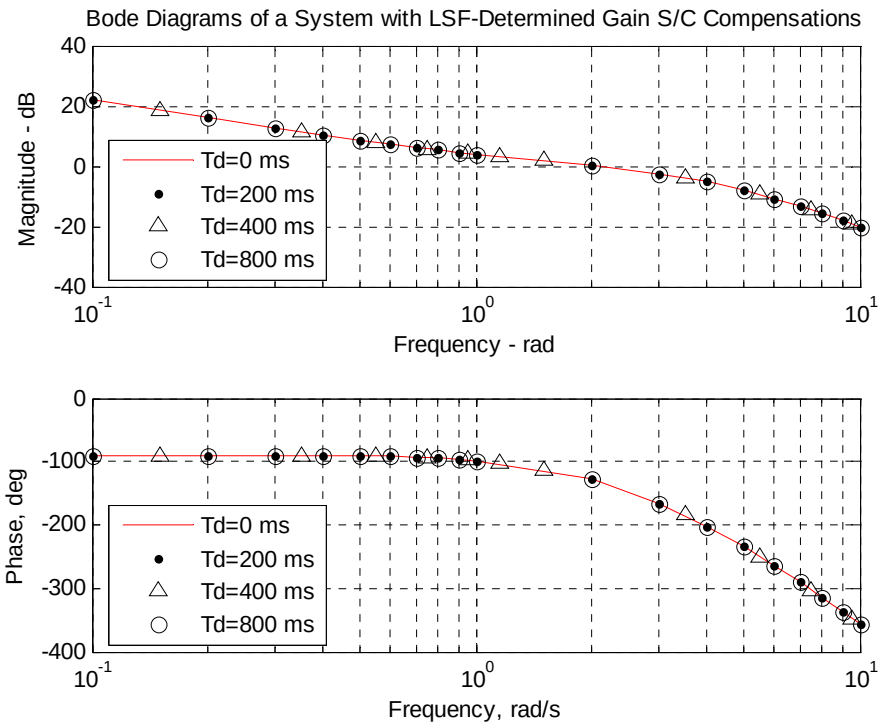


Fig. B.5. Bode diagrams with state space compensation with K by LFS

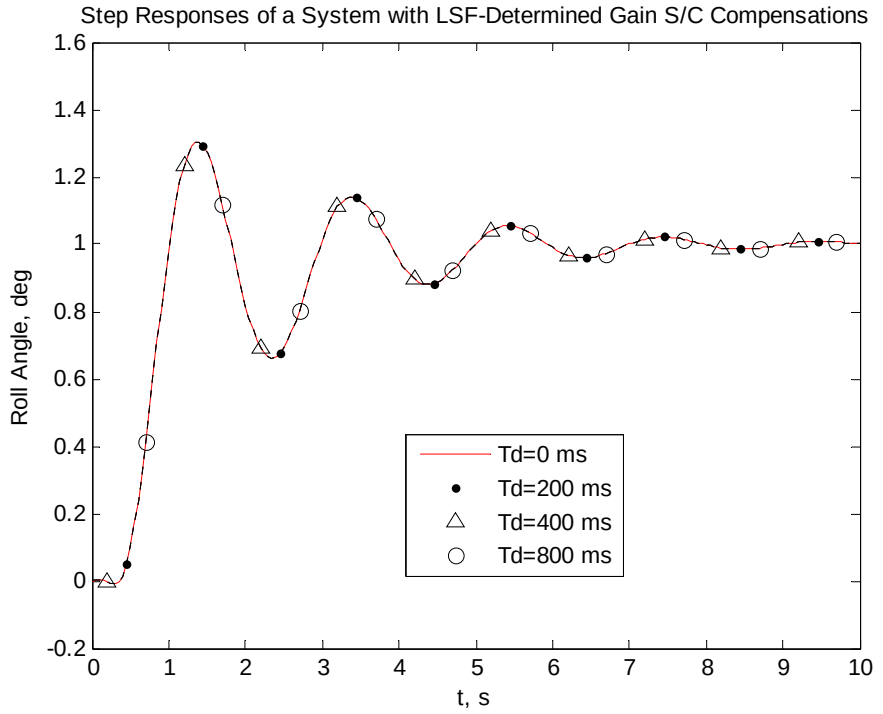


Fig. B.6. Step responses with state space compensation with K by LFS

Another advantage of the five-point method and the LSF method over the pole placement method is that the gain matrix $\mathbf{K}$, designed with them, makes the forward gain $G = (1 + \mathbf{K\Psi B})^{-1}$ very close to unity, and because of this, the equivalent system is almost identical to the original system, without a noticeable change of system states. In the five-point method and the LSF method, the frequency characteristic function of the compensated system is forced to be close to that of the undelayed system. Hence the matrix $\mathbf{D}$ of the open loop compensated system ($D = \mathbf{C\Psi B}$) is also forced to be close to that of the undelayed system, which is zero. Therefore, $D = \mathbf{K\Psi B} \approx 0$, and $G = (1 + \mathbf{K\Psi B})^{-1} \approx 1$. This explains why the feed forward gain is unity from the two frequency design methods. This is significant in the simplification of the compensation.

B.2. State Observer for the State Space Compensation

If the state space filter is used in another control system to compensate for time delay, and some of the states to be fed back are not measurable, then a state observer can be designed to make available those immeasurable states. Fig. B.7 illustrates a state space compensation system making use of a full order state observer. The state equations and the output equations of the control system and the predictor with the observer are, respectively

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} \\ u = v - \tilde{y}_c \end{cases} \tag{B.7}$$

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = \mathbf{A}_c\tilde{\mathbf{x}} + \mathbf{B}u + \mathbf{L}y \\ \tilde{y} = \mathbf{C\Phi}\tilde{\mathbf{x}} + \mathbf{C\Psi B}u \end{cases} \tag{B.8}$$

where $\mathbf{A}_c = \mathbf{A} - \mathbf{LC}$. Define $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$ then it follows from Eqs. (B.7) and (B.8) that

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}} \\ &= \mathbf{A}_c\mathbf{e}\end{aligned} \tag{B.9}$$

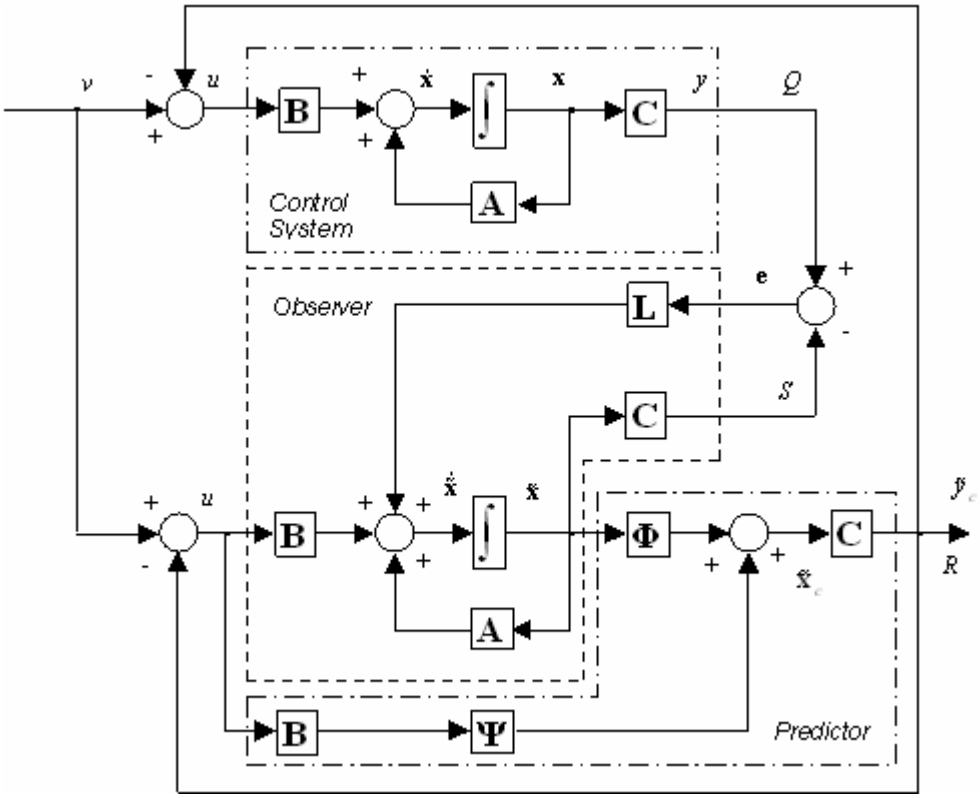


Fig. B.7. State space compensation with a full order state observer

If all eigenvalues of $\mathbf{A}_c$ are selected to be negative, Eq. (B.9) is an asymptotically stable error equation. The eigenvalues of $\mathbf{A}_c$ control the speed of convergence. Experience indicates that a good design usually results if the continuous-time observer poles are selected to be a little farther to the left in the s-place than the desired closed loop state feed back poles (Brogan 1974). This technique is used here. The fact that the control system is now completely observable makes it possible to choose any eigenvalues for $\mathbf{A}_c$ as desired. By eliminating the variable u ($u = G\left(v - \mathbf{C}\Phi\tilde{\mathbf{x}}\right)$), the equations that characterize the entire system are obtained as

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}G\mathbf{C}\boldsymbol{\Phi} \\ \mathbf{LC} & \mathbf{A}_c - \mathbf{B}G\mathbf{C}\boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}G \\ \mathbf{B}G \end{bmatrix} v \\ \tilde{y} = \begin{bmatrix} \mathbf{C}\boldsymbol{\Phi} - \mathbf{C}\boldsymbol{\Psi}\mathbf{B}G(\mathbf{C}\boldsymbol{\Phi}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{bmatrix} + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}Gv \end{cases} \tag{B.10}$$

where $G = (\mathbf{I} + \mathbf{C}\boldsymbol{\Psi}\mathbf{B})^{-1}$. The observed states can be fed back with the gain matrix $\mathbf{K}$ designed with the two frequency methods introduced above in place of the matrix $\mathbf{C}$. The poles of the observer and the poles of the feed back controller can be chosen independently because the poles of Eq. (B.10) can be separated as follows

$$\begin{aligned} \Delta_T(\lambda) &= \begin{vmatrix} \mathbf{I}_n\lambda - \mathbf{A} & \mathbf{B}G\mathbf{K}\boldsymbol{\Phi} \\ -\mathbf{LC} & \mathbf{I}_n\lambda - \mathbf{A}_c + \mathbf{B}G\mathbf{K}\boldsymbol{\Phi} \end{vmatrix} \\ &= \begin{vmatrix} \mathbf{I}_n\lambda - \mathbf{A} + \mathbf{B}G\mathbf{K}\boldsymbol{\Phi} & \mathbf{B}G\mathbf{K}\boldsymbol{\Phi} \\ 0 & \mathbf{I}_n\lambda - \mathbf{A}_c \end{vmatrix} \\ &= \left|\mathbf{I}_n\lambda - \mathbf{A} + \mathbf{B}G\mathbf{K}\boldsymbol{\Phi}\right| \left|\mathbf{I}_n\lambda - \mathbf{A}_c\right| \\ &= \Delta_c(\lambda)\Delta'(\lambda) \end{aligned} \tag{B.11}$$

As previously stated, using the feedback matrix $\mathbf{K}$, which was designed with the two frequency methods, makes the feed forward gain G be close to unity and the equivalent compensation system identical to the original compensation system. This also makes it possible to adopt the equivalent system to the observer system, as illustrated in Fig. B.8. The state equation of the observer is changed from Eq. (B.8) to

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}Gu + \mathbf{LC}(\mathbf{x} - \tilde{\mathbf{x}}) \tag{B.12}$$

where the control law is also changed to

$$u = v - \tilde{y} = v - \mathbf{K}\boldsymbol{\Phi}\tilde{\mathbf{x}} \tag{B.13}$$

From these, the governing differential equation of the error for the new observer becomes

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC})\mathbf{e} + \mathbf{B}(1 - G)u \tag{B.14}$$

Since $G \approx 1$, this equation reduces to Eq. (B.9). Therefore, the design of the observer gain matrix $\mathbf{L}$ may be undertaken the same way as for the observer in Fig. B.7.

If some state variables can be measured, then a minimal order state observer may be designed to implement the state space compensation.

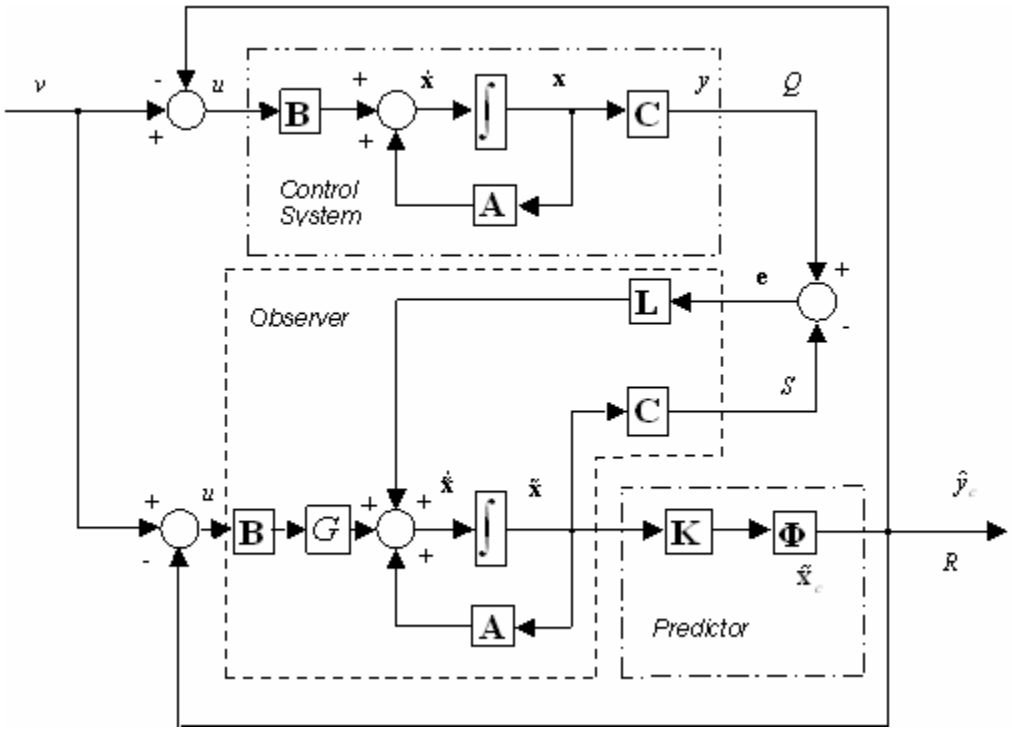


Fig. B.8. State space compensation with a full order equivalent state observer

Appendix C. State Space Compensation in LTI Systems

In the state space compensation, introduced in Appendix B, the Pade approximation of the time delay is added first so that the matrices $\mathbf{A}$, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ all contain the information about the time delay. The filter generates the compensated states from the delayed states. In this manner the prediction is not visible. In this section, the prediction y_p is produced before the time delay is added, and then it is delayed to get the compensated output to use as feedback. This approach is illustrated in Fig. C.1, where the matrices $\mathbf{A}_a$, $\mathbf{\Phi}_a$ and $\mathbf{\Psi}_a$ do not carry the information about the time delay. The transfer function of the 2nd-order Pade approximation of the time delay is given in Eq. (2.4), and the $\mathbf{A}_t$, $\mathbf{B}_t$, $\mathbf{C}_t$ and $\mathbf{D}_t$ are given by

$$\mathbf{A}_t = \begin{bmatrix} -\frac{6}{t_d} & 1 \\ \frac{12}{t_d^2} & 0 \end{bmatrix}, \mathbf{B}_t = \begin{bmatrix} -\frac{12}{t_d} \\ 0 \end{bmatrix}, \mathbf{C}_t = \begin{bmatrix} 1 & 0 \end{bmatrix}, \mathbf{D}_t = 1 \tag{C.1}$$

In whichever form is chosen for the state space equations, the matrix $\mathbf{D}_t$ is always unity because the Pade approximation transfer function is normal and its gain is unity. Then the delayed output can be calculated using

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{A}_a & \mathbf{0} \\ \mathbf{B}_t\mathbf{C}_a & \mathbf{A}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{0} \end{bmatrix} u \\ y_d = \begin{bmatrix} \mathbf{C}_a & \mathbf{C}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} \end{cases} \tag{C.2}$$

The prediction is given by

$$\begin{cases} \dot{\mathbf{x}}_a = \mathbf{A}_a\mathbf{x}_a + \mathbf{B}_a u \\ y_p = \mathbf{C}_a\mathbf{\Phi}_a\mathbf{x}_a + \mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a u \end{cases} \tag{C.3}$$

The predicted state is output to the cueing system to compensate for the time delay. The compensated output may be calculated in a manner similar to that used for Eq. (C.2) as follows

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{A}_a & \mathbf{0} \\ \mathbf{B}_t\mathbf{C}_a\mathbf{\Phi}_a & \mathbf{A}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_t\mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a \end{bmatrix} u \\ y_c = \begin{bmatrix} \mathbf{C}_a\mathbf{\Phi}_a & \mathbf{C}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} + \mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a u \end{cases} \tag{C.4}$$

Finally, the output feedback closed loop state equation and output equation are given by Eq. (C.5), where $G = \left(1 + \mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a\right)^{-1}$ is the feed forward gain. The compensation results shown in Figs. 4.14 and 4.15 were obtained with this algorithm.

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{A}_a - \mathbf{B}_a\mathbf{C}_a\mathbf{\Phi}_a & -\mathbf{B}_a\mathbf{C}_t \\ \mathbf{B}_t\mathbf{C}_a\left(\mathbf{I} - \mathbf{\Psi}_a\mathbf{B}_a\mathbf{C}_a\right)\mathbf{\Phi}_a & \mathbf{A}_t - \mathbf{B}_t\mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a\mathbf{C}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_t\mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a \end{bmatrix} v \\ y_c = G\begin{bmatrix} \mathbf{C}_a\mathbf{\Phi}_a & \mathbf{C}_t \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_t \end{bmatrix} + G\mathbf{C}_a\mathbf{\Psi}_a\mathbf{B}_a v \end{cases} \tag{C.5}$$

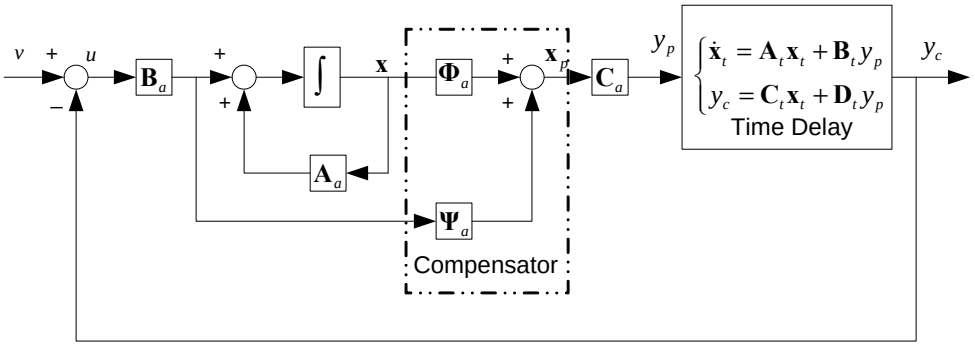


Fig. C.1. State space predictor in a linear time-variant system

The matrix $\mathbf{C}_a$ can be replaced with a matrix $\mathbf{K}$ designed with similar frequency methods described in Appendix B. With the gain matrix $\mathbf{K}$, the feed forward gain becomes $G = \left(1 + \mathbf{K}\mathbf{\Psi}_a\mathbf{B}_a\right)^{-1}$.

As shown in Figs. 3.13 and 3.14, this compensation based on prediction cannot compete with the Sobiski/Cardullo filter when compensating for long time delay. For

example, the compensation errors, which come from compensation based on prediction, are considerably larger then the errors arising from the Sobiski/Cardullo approach as the time delay approaches 800ms. In addition, it has been demonstrated that the longer the delay is, the farther the feed forward gain G deviates from unity. Remember that, as the time delay gets larger, the greater compensation errors mean the compensated system diverges from the undelayed system, and hence the matrix $\mathbf{D}_c = \mathbf{C}_a \mathbf{\Psi}_a \mathbf{B}_a$ of the compensated system moves away from $\mathbf{D}_a$ of the undelayed system, which is zero. This also means that $G = \left(1 + \mathbf{C}_a \mathbf{\Psi}_a \mathbf{B}_a\right)^{-1}$ deviates further from unity. Nonetheless, the prediction is significant. The state space predictor introduced in this section is the first step in the evolution from the Sobiski/Cardullo filter to the aircraft reference model state space filter applicable to a flight simulator.

Appendix D. Discrete State Space Filter and Time-Variant State Space Filters

D.1. Discrete State Space Filter

The discrete state space control system of the corresponding continuous control system $\begin{cases}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x}\end{cases}$ may be expressed as $\begin{cases}\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}u(k) \\ y(k) = \mathbf{C}\mathbf{x}(k)\end{cases}$, where $\mathbf{G} = e^{\mathbf{A}T}$ and $\mathbf{H} = \left(\int_0^T e^{\mathbf{A}\tau} d\tau\right)\mathbf{B}$. The discrete state space predictor comes from the recursive formula for calculating the succeeding states in terms of the previously available ones

$$\begin{aligned}
&\mathbf{x}(k+2) = \mathbf{G}\mathbf{x}(k+1) + \mathbf{H}u(k+1) = \mathbf{G}^2\mathbf{x}(k) + \mathbf{G}\mathbf{H}u(k) + \mathbf{H}u(k+1) \\
&\mathbf{x}(k+3) = \mathbf{G}\mathbf{x}(k+2) + \mathbf{H}u(k+2) = \mathbf{G}^3\mathbf{x}(k) + \mathbf{G}^2\mathbf{H}u(k) + \mathbf{G}\mathbf{H}u(k+1) + \mathbf{H}u(k+2) \\
&\vdots \\
&\mathbf{x}(k+l) = \mathbf{G}\mathbf{x}(k+l-1) + \mathbf{H}u(k+l-1) = \mathbf{G}^l\mathbf{x}(k) + \mathbf{G}^2\mathbf{H}u(k) + \sum_{j=0}^{k-1}\mathbf{G}^{k-j-1}\mathbf{H}u(j)
\end{aligned} \tag{D.1}$$

By making the same assumptions on the input u as for the continuous state space filter, the future inputs can be approximated by the current input $u(k)$, and the discrete state space predictor can be obtained from the last equation of (D.1), as given by

$$\mathbf{x}(k+l) = \mathbf{\Phi}_d\mathbf{x}(k) + \mathbf{\Psi}_d u(k) \tag{D.2}$$

where $\mathbf{\Phi}_d(l) = \mathbf{G}^l$ and $\mathbf{\Psi}_d(l) = \sum_{j=0}^{l-1}\mathbf{\Phi}_d(j)\mathbf{H}$ are the discrete state transition matrix and its integration matrix. The predicted output is

$$y(k+l) = \mathbf{C}\mathbf{\Phi}_d\mathbf{x}(k) + \mathbf{C}\mathbf{\Psi}_d u(k) \tag{D.3}$$

The unit output feedback discrete state space compensation may be implemented as shown in Fig. D.1.

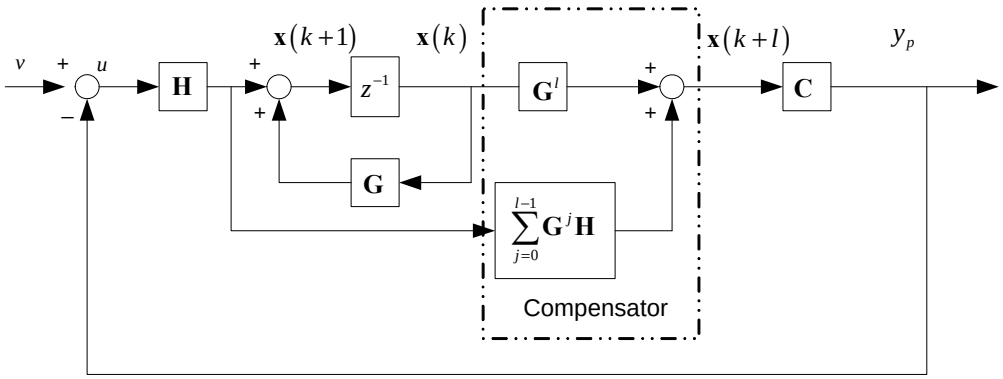


Fig. D.1. Discrete state space compensation

D.2. Continuous Time-variant State Space Filter

The time varying continuous state space filter can be derived from the solution of the time varying continuous state equation $\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)u(t)$, which is given by

$$\mathbf{x}(t) = \mathbf{\Phi}(t,t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau \tag{D.4}$$

where $\mathbf{\Phi}(t,\tau)$ is the general state transition matrix defined as

$$\begin{aligned} &\mathbf{\Phi}(\tau,\tau) = \mathbf{I}. \\ &\frac{d\mathbf{\Phi}(t,\tau)}{dt} = \mathbf{A}(t)\mathbf{\Phi}(t,\tau) \end{aligned} \tag{D.5}$$

By making some assumptions about the control input u and the system matrix $\mathbf{A}$, the continuous time-variant state space predictor is given by

$$\mathbf{x}(t+t_d) = \left[e^{\mathbf{A}(t)t_d}\right]\mathbf{x}(t) + \left[\int_0^{t_d} e^{\mathbf{A}(t)(t_d-\tau)}d\tau\right]\mathbf{B}(t)u(t) \tag{D.6}$$

D.3. Discrete Time-variant State Space Filter

The time-variant discrete state space filter comes from the solution of the time-variant state equation $\mathbf{x}(k+1)=\mathbf{G}(k)\mathbf{x}(k)+\mathbf{H}(k)u(k)$, and is given as

$$\mathbf{x}(k)=\mathbf{\Phi}(k,k_0)\mathbf{x}(k_0)+\sum_{j=k_0}^{k-1}\mathbf{\Phi}(k,j+1)\mathbf{H}(j)u(j) \tag{D.7}$$

with $\mathbf{\Phi}(k,k_0)$ being the time-variant state transition matrix, which can be calculated by

$$\begin{aligned}&\mathbf{\Phi}(k_0,k_0)=\mathbf{I}.\\&\mathbf{\Phi}(k,k_0)=\mathbf{G}(k-1)\mathbf{G}(k-2)\cdots\mathbf{G}(k_0)\end{aligned} \tag{D.8}$$

By making some assumptions about the control input u and the system matrix $\mathbf{G}$, the discrete time-variant state space predictor is given by

$$\mathbf{x}(k+l)=\left[\mathbf{G}(k)^l\right]\mathbf{x}(k)+\left[\sum_{j=0}^{l-1}\mathbf{G}(k)^j\,\mathbf{H}(k)\right]u(k) \tag{D.9}$$

The state transition matrix $\mathbf{G}(k)^l$ and its numerical integral $\sum_{j=0}^{l-1}\mathbf{G}(k)^j\,\mathbf{H}(k)$ have to be calculated for each iteration.

Appendix E. Miscellanea on the Novel State Space Compensator

E.1. An Example to Show the Compensation Principle of a State Space Filter

Suppose there is a 2nd-order linear time-invariant system, with a transfer function given by

$$\frac{Y(s)}{U(s)} = \frac{a_0}{s^2 + a_1 s + a_0} \tag{E.1}$$

Its state equation and output equation are given by $\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} + Du \end{cases}$, and the observable matrices are

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 \\ -a_0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ a_0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, D = 0 \tag{E.2}$$

Let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, then the undelayed output is

$$y = x_1 \tag{E.3}$$

With the state space compensation $y_p = \mathbf{C}\boldsymbol{\Phi}\mathbf{x} + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}u$, the predicted output can be calculated by

$$y_p = \phi_{11} x_1 + \phi_{12} x_2 + \psi_{12} a_0 u \tag{E.4}$$

where ϕ_{11} and ψ_{12} are elements of the matrix $\boldsymbol{\Phi}$ and its integration $\boldsymbol{\Psi}$. Comparing Eqs. (E.3) and (E.4) gives $y_p = \phi_{11} y + \phi_{12} x_2 + \psi_{12} a_0 u$. It indicates that the continuous state space filter uses the system states and the control input to predict. The weighting factors of the relevant terms are determined by the state transition matrix and its integral. Because the state variable x_2 carries the velocity information, the prediction makes use of the velocity. If the system is of 3rd-order, the acceleration will be used to predict, and for higher order systems, more system information is required to make a prediction.

On the other hand, for a discrete system, say $\frac{Y(z)}{U(z)} = \frac{n_1 z + n_0}{z^2 + m_1 z + m_0}$, which is the discrete form of the system given by (E.1), the predicted output becomes

$$y(k+l) = \phi_{11}^z y(k) - \phi_{12}^z m_0 x_1(k-1) + \left(n_1 \psi_{11}^z + n_0 \psi_{12}^z\right) u(k) + n_0 u(k-1) \tag{E.5}$$

Therefore, the discrete state space filter uses the current and past system information (states and input) to predict.

E.2. The Filter Coefficients in Terms of the Eigenvalues

A 2nd-order system can be characterized by its natural frequency ω_n and damping ratio ζ. In terms of ω_n and ζ, the parameters in the transfer function in (E.1) can be written as

$$a_1 = 2\omega_n \zeta, a_0 = \omega_n^2 \tag{E.6}$$

Let the two eigenvalues of this system be denoted by λ_1 and λ_2, and the following relationships become straightforward

$$a_1 = -(\lambda_1 + \lambda_2), a_2 = \lambda_1 \lambda_2 \tag{E.7}$$

As stated in Appendix A, the state transition matrix and its integral can be obtained in terms of the system eigenvalues. The results are different for three cases with regard to the two eigenvalues, i.e., two different real eigenvalues, two repeated real eigenvalues, or a pair of complex conjugates.

Two unequal real eigenvalues

This case corresponds to an over-damped system, with $\zeta > 1$. The two eigenvalues are given by

$$\lambda_{1,2} = -\omega_n \zeta \pm \omega_n \sqrt{\zeta^2 - 1} \tag{E.8}$$

Eq. (A.9) becomes $\mathbf{\Phi} = \alpha_0 + \alpha_1 \mathbf{A}$. Without the details of derivation, α_0 and α_1 are given as follows

$$\begin{cases} \alpha_0 = \dfrac{\lambda_1 e^{\lambda_2 t_d} - \lambda_2 e^{\lambda_1 t_d}}{\lambda_1 - \lambda_2} \\ \alpha_1 = \dfrac{e^{\lambda_1 t_d} - e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} \end{cases} \tag{E.9}$$

where t_d is the time delay. The state transition matrix and its integral are given, respectively, by

$$\begin{aligned} \mathbf{\Phi} = e^{\mathbf{A}t_d} &= \frac{1}{\lambda_1 - \lambda_2}\begin{bmatrix} \lambda_1 e^{\lambda_1 t_d} - \lambda_2 e^{\lambda_2 t_d} & -\lambda_1\lambda_2\left(e^{\lambda_1 t_d} - e^{\lambda_2 t_d}\right) \\ e^{\lambda_1 t_d} - e^{\lambda_2 t_d} & \lambda_1 e^{\lambda_2 t_d} - \lambda_2 e^{\lambda_1 t_d} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_0 - a_1\alpha_1 & \alpha_1 \\ -a_0\alpha_1 & \alpha_0 \end{bmatrix} \end{aligned} \tag{E.10}$$

$$\begin{aligned} \mathbf{\Psi} &= \begin{bmatrix} \dfrac{e^{\lambda_1 t_d} - e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} & \dfrac{\lambda_2 e^{\lambda_1 t_d} + (\lambda_1 - 2\lambda_2)e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} - 1 \\ \dfrac{\lambda_2 e^{\lambda_1 t_d} - \lambda_1 e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} + 1 & \dfrac{-\lambda_2 e^{\lambda_1 t_d}}{\lambda_1(\lambda_1 - \lambda_2)} - \dfrac{\lambda_1 + 2\lambda_2 e^{\lambda_1 t_d}}{\lambda_2(\lambda_1 - \lambda_2)} - \dfrac{1}{\lambda_1} - \dfrac{1}{\lambda_2} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1 & \dfrac{1-\alpha_0}{a_0} \\ \alpha_0 - 1 & \dfrac{a_1 - a_1\alpha_0 + a_2\alpha_1}{a_0} \end{bmatrix} \end{aligned} \tag{E.11}$$

The D matrix of the compensated system can be calculated by

$$\begin{aligned} D = \mathbf{C\Psi B} &= \lambda_1\lambda_2\left(\frac{\lambda_2 e^{\lambda_1 t_d} + (\lambda_1 - 2\lambda_2)e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} - 1\right) \\ &= 1 - \alpha_0 \end{aligned} \tag{E.12}$$

From (E.4), the prediction is given by

$$\begin{aligned} y_p &= \frac{\lambda_1 e^{\lambda_1 t_d} - \lambda_2 e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} y - \frac{\lambda_1\lambda_2\left(e^{\lambda_1 t_d} - e^{\lambda_2 t_d}\right)}{\lambda_1 - \lambda_2} x_2 + \left[\frac{\lambda_2 e^{\lambda_1 t_d} + (\lambda_1 - 2\lambda_2)e^{\lambda_2 t_d}}{\lambda_1 - \lambda_2} - 1\right]\lambda_1\lambda_2 u \\ &= (\alpha_0 - a_1\alpha) y + \alpha_1 x_2 + (1 - \alpha_0) u \end{aligned} \tag{E.13}$$

For formulas (E.10)-(E.13), the last equations are in terms of a_0 and a_1, the parameters of the transfer function, and α_0 and α_1, which are given in (E.9). The remaining cases use the same convention.

Two repeated real eigenvalues

This case corresponds to a marginally damped system, with $\zeta = 1$. The equationswhich correspond to Eqs. (E.8) through (E.13) for this case are given as follows

$$\lambda_{1,2} = \lambda = -\omega_n \zeta \tag{E.14}$$

$$\begin{cases} \alpha_0 = e^{\lambda t_d} - t_d e^{\lambda t_d} \\ \alpha_1 = t_d e^{\lambda t_d} \end{cases} \tag{E.15}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1+\lambda t_d & t_d \\ -\lambda^2 t_d & 1-\lambda t_d \end{bmatrix} e^{\lambda t_d} \tag{E.16}$$

$$\mathbf{\Psi} = \begin{bmatrix} t_d e^{\lambda t_d} & \dfrac{-e^{\lambda t_d} + \lambda t_d e^{\lambda t_d} + 1}{\lambda^2} \\ e^{\lambda t_d} - \lambda t_d e^{\lambda t_d} - 1 & \dfrac{2e^{\lambda t_d} - \lambda t_d e^{\lambda t_d} - 2}{\lambda} \end{bmatrix} \tag{E.17}$$

$$D = \mathbf{C\Psi B} = 1 - e^{\lambda t_d}\left(1 - \lambda t_d\right) \tag{E.18}$$

$$y_p = \left(1+\lambda t_d\right) e^{\lambda t_d} y + t_d e^{\lambda t_d} x_2 + \left[1 - e^{\lambda t_d}\left(1 - \lambda t_d\right)\right] u \tag{E.19}$$

Two complex conjugates

This case corresponds to an under-damped system, with $\zeta < 1$. The equationswhich correspond to Eqs. (E.8) through (E.13) for this case are given as follows

$$\lambda_{1,2} = -\omega_n \zeta \pm j\omega_n \sqrt{1-\zeta^2} = \alpha \pm j\beta \tag{E.20}$$

The relations given in (E.7) becomes

$$a_1 = -2\alpha, a_0 = \alpha^2 + \beta^2 \tag{E.21}$$

$$\begin{cases} \alpha_0 = \dfrac{e^{\alpha t_d}}{\beta}\left(\beta \sin \beta t_d - \alpha \sin \beta t_d\right) \\ \alpha_1 = \dfrac{e^{\alpha t_d}}{\beta} \sin \beta t_d \end{cases} \tag{E.22}$$

$$\mathbf{\Phi} = \begin{bmatrix} \beta \cos \beta t_d + \alpha \sin \beta t_d & \sin \beta t_d \\ -\left(\alpha^2 + \beta^2\right) \sin \beta t_d & \beta \cos \beta t_d - \alpha \sin \beta t_d \end{bmatrix} \frac{e^{\alpha t_d}}{\beta} \tag{E.23}$$

$$\mathbf{\Psi} = \begin{bmatrix} \dfrac{e^{\alpha t_d} \sin \beta t_d}{\beta} & \dfrac{\beta + \left(\alpha \sin \beta t_d - \beta \cos \beta t_d\right) e^{\alpha t_d}}{\left(\alpha^2 + \beta^2\right)\beta} \\ \dfrac{\left(\beta \cos \beta t_d - \alpha \sin \beta t_d\right) e^{\alpha t_d}}{\beta} - 1 & \dfrac{\left(\beta^2 - \alpha^2\right) \sin \beta t_d + 2\alpha\beta \cos \beta t_d}{\left(\alpha^2 + \beta^2\right)\beta} e^{\alpha t_d} - 2\alpha \end{bmatrix} \tag{E.24}$$

$$D = \mathbf{C\Psi B} = 1 - \left(\cos \beta t_d - \frac{\alpha}{\beta} \sin \beta t_d\right) e^{\alpha t_d} \tag{E.25}$$

$$y_p = \left(\cos \beta t_d + \frac{\alpha}{\beta} \sin \beta t_d\right) e^{\alpha t_d} y - \frac{e^{\alpha t_d}}{\beta} \sin \beta t_d \, x_2 + \left[1 - \left(\cos \beta t_d - \frac{\alpha}{\beta} \sin \beta t_d\right) e^{\alpha t_d}\right] u \tag{E.26}$$

For a 3rd-order system $\dfrac{Y(s)}{U(s)} = \dfrac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$, its state transition matrix and its integral can be calculated by

$$\begin{aligned} \mathbf{\Phi} &= \alpha_0 \mathbf{I} + \alpha_1 \mathbf{A} + \alpha_2 \mathbf{A}^2 \\ &= \begin{bmatrix} \alpha_0 - a_2\alpha_1 + \left(a_2^2 - a_1\right)\alpha_2 & \alpha_1 - a_2\alpha_2 & \alpha_2 \\ -a_1\alpha_1 + \left(a_1 a_2 - a_0\right)\alpha_2 & \alpha_0 - a_1\alpha_2 & \alpha_1 \\ -a_0\alpha_1 + a_1 a_2 \alpha_2 & -a_0\alpha_2 & \alpha_0 \end{bmatrix} \end{aligned} \tag{E.27}$$

$$\mathbf{\Psi} = \begin{bmatrix} \alpha_1 - a_2\alpha_2 & \alpha_2 & \dfrac{1-\alpha_0}{a_0} \\ \alpha_0 - 1 - a_1\alpha_2 & \alpha_1 & \dfrac{a_2}{a_0}(1-\alpha_0) + \alpha_2 \\ -a_0\alpha_2 & \alpha_0 - 1 & \alpha_0 \end{bmatrix} \tag{E.28}$$

Depending on the eigenvalues of the matrix $\mathbf{A}$, the expressions of α_0, α_1 and α_2 may be different. Usually, a 3rd-order aircraft model has one real and two complex conjugate eigenvalues. Denote them by $\lambda_1 = r, \lambda_{2,3} = m \pm n$, then the following relations hold

$$a_0 = -r\left(m^2 + n^2\right), a_2 = 2rm + \left(m^2 + n^2\right), a_3 = -\left(r + 2m\right) \tag{E.29}$$

And the expressions of α_0, α_1 and α_2 are given by

$$\begin{cases} \alpha_0 = \dfrac{1}{\Delta}\left[\dfrac{-r\left(r\alpha - \alpha^2 + \beta^2\right)}{\beta} e^{\alpha t_d} \sin\beta t_d - \dfrac{r\left(2\alpha - r\right)}{\beta} e^{\alpha t_d} \cos\beta t_d + \left(\Delta - r^2 + 2r\alpha\right) e^{rt_d}\right] \\ \alpha_0 = \dfrac{1}{\Delta}\left[\dfrac{r^2 - \alpha^2 + \beta^2}{\beta} e^{\alpha t_d} \sin\beta t_d - 2\alpha e^{\alpha t_d} \cos\beta t_d + 2\alpha e^{rt_d}\right] \\ \alpha_0 = \dfrac{1}{\Delta}\left[\dfrac{\alpha - r}{\beta} e^{\alpha t_d} \sin\beta t_d - e^{\alpha t_d} \cos\beta t_d + e^{rt_d}\right] \end{cases} \tag{E.30}$$

where $\Delta = r^2 + \alpha^2 + \beta^2 - 2r\alpha$. The matrix D of the compensated system is

$$D = \mathbf{C\Psi B} = b_2\left(\alpha_1 - a_2\alpha_2\right) + b_1\alpha_2 + b_0\frac{1 - \alpha_0}{a_0} \tag{E.31}$$

Appendix F. Calculation of the Accelerations in the Topodetic Frame

The topodetic frame is also called the north-east-up (NEU) frame. This is the frame in which most visual images in a flight simulator are generated. The six degrees of freedom (DOF) are: roll, pitch, yaw (i.e., the three Euler angles), altitude, longitude and latitude. The displacements, velocities and accelerations in the 6-DOF of the topodetic frame are usually calculated from those in the 6-DOF of the body frame, which are available from the flight dynamics and equations of motion (EOM). Because the accelerations in the topodetic frame are not used in the visual image generation, they are not usually available from the simulation computer. As mentioned in Chapter 4 (4.5.1), they are necessary for applying the state space predictor in a flight simulator to compensate the transport delay in the visual system. The formulas to calculate the accelerations in the 6-DOF of the topodetic frame are derived as follows. These six accelerations can be divided into two groups—three angular accelerations (2^{nd} derivatives of the Euler angles) and three translational accelerations.

F.1. Angular Accelerations

The transformation of the angular velocities from the body frame to the earth frame is given by the kinematic equations

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{F.1}$$

Denote $\mathbf{v}_K^E = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ (velocity vector in the earth or topodetic frame for the kinematic equations), $\mathbf{v}_K^B = \begin{bmatrix} p & q & r \end{bmatrix}^T$ (velocity vector in the body frame for the kinematic equations), and

$$\mathbf{P}_{B2E}^{A} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}, \tag{F.2}$$

then Eq. (F.1) can be rewritten in a compact form

$$\mathbf{v}_K^E = \mathbf{P}_{B2E}^{A}\mathbf{v}_K^B. \tag{F.3}$$

The accelerations may be calculated by taking the derivatives on both sides of Eq. (F.3), and the final result is

$$\dot{\mathbf{v}}_K^E = \mathbf{P}_{B2E}^{A}\dot{\mathbf{v}}_K^B + \mathbf{Q}\mathbf{P}_{B2E}^{A}\mathbf{v}_K^B, \tag{F.4}$$

where the matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} \tan\theta\left(q\cos\phi - r\sin\phi\right) & \sec^2\theta\left(q\sin\phi + r\cos\phi\right) & 0 \\ -q\sin\phi - r\cos\phi & 0 & 0 \\ q\dfrac{\cos\phi}{\cos\theta} - r\dfrac{\sin\phi}{\cos\theta} & \dfrac{\sin\theta}{\cos^2\theta}\left(q\sin\phi + r\cos\phi\right) & 0 \end{bmatrix}. \tag{F.5}$$

Note that though the matrix $\mathbf{P}_{B2E}^{A}$ is dimensionless, the matrix $\mathbf{Q}$ is not, and its units are 1/sec. This also makes the second term of Eq. (F.4) an acceleration term (the unit is $1/s^2$). The final result of product $\mathbf{P}_{B2E}^{A}\mathbf{Q}$ is given by

$$\mathbf{P}_{B2E}^{A}\mathbf{Q} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{31} \\ \alpha_{21} & \alpha_{22} & \alpha_{32} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \tag{F.6}$$

where the nine elements are given in Table F.1.

Investigation shows that the second term of Eq. (F.4) is much smaller than the first term (the former is about one thousandth of the latter), and hence Eq. (F.4) may be approximated by discarding the second term, that is

$$\dot{\mathbf{v}}_K^E \approx \mathbf{P}_{B2E}^{A}\dot{\mathbf{v}}_K^B, \tag{F.7}$$

or

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}. \tag{F.8}$$

This is exactly Eq. (4.34).

Table F.1. Expressions of the nine elements of matrix $\mathbf{P}_{B2E}^{A}\mathbf{Q}$

Element	Expression
α_{11}	$v(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)+w(\cos\phi\sin\psi-\sin\phi\sin\theta\cos\psi)$
α_{12}	$-u\sin\theta\cos\psi+v\sin\phi\cos\theta\cos\psi+w\cos\phi\cos\theta\cos\psi$
α_{13}	$-u\cos\theta\sin\psi-v(\sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi)+w(\sin\phi\cos\psi-\cos\phi\sin\theta\sin\psi)$
α_{21}	$v(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)-w(\cos\phi\cos\psi+\sin\phi\sin\theta\sin\psi)$
α_{22}	$-u\sin\theta\sin\psi+v\sin\phi\cos\theta\sin\psi+w\cos\phi\cos\theta\sin\psi$
α_{23}	$u\cos\theta\cos\psi+v(\sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi)+w(\sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi)$
α_{31}	$-v\cos\phi\cos\theta+w\sin\phi\cos\theta$
α_{32}	$u\cos\theta+v\sin\phi\sin\theta+w\cos\phi\sin\theta$
α_{33}	0

F.2. Translational Accelerations

The transformation of the angular velocities from the body frame to the earth frame is given by the navigation equations

$$\begin{bmatrix} \dot{\lambda} \\ \dot{l} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi \\ \sin\theta & -\sin\phi\cos\theta & -\cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{F.9}$$

Denote $\mathbf{v}_N^E = \begin{bmatrix} \dot{\lambda} & \dot{l} & \dot{h} \end{bmatrix}^T$ (velocity vector in the earth or topodetic frame for the navigation equations) and $\mathbf{v}_N^B = \begin{bmatrix} u & v & w \end{bmatrix}^T$ (velocity vector in the body frame for the

navigation equations). Denote the transformation matrix for the translational velocities from the body frame to the topodetic frame as

$$\mathbf{P}_{B2E}^{T}=\begin{bmatrix} cos\theta cos\psi & sin\phi sin\theta cos\psi - cos\phi sin\psi & cos\phi sin\theta cos\psi + sin\phi sin\psi \\ cos\theta sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi \\ sin\theta & -sin\phi cos\theta & -cos\phi cos\theta \end{bmatrix}, \quad \text{(F.10)}$$

then Eq. (F.9) can be rewritten in a compact form

$$\mathbf{v}_{N}^{E}=\mathbf{P}_{B2E}^{T}\mathbf{v}_{N}^{B}. \quad \text{(F.11)}$$

The translational accelerations may be calculated by taking the derivatives on both sides of Eq. (F.11), and the final result is

$$\begin{bmatrix} \ddot{\lambda} \\ \ddot{l} \\ \ddot{h} \end{bmatrix}=\mathbf{P}_{B2E}^{T}\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}+\mathbf{T}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{(F.12)}$$

This is the same as Eq. (4.35), and the matrix $\mathbf{T}$ is available in Table 4.2.

In the VMS, the longitude and latitude are expressed in angles, therefore, the terms $\ddot{\lambda}$ and $\ddot{l}$ have to be divided by the corresponding radii to change to angles. The results are

$$\ddot{\lambda}_{\theta}=\frac{\ddot{\lambda}}{h+\dfrac{a^2b^2}{\sqrt{\left[b^2+\left(a^2-b^2\right)cos^2\lambda\right]^3}}} \quad \text{(F.13)}$$

$$\ddot{l}_{\theta}=\frac{\ddot{l}\,cos\lambda}{\dfrac{a}{\sqrt{cos^2\lambda+sin^2\lambda\dfrac{b^2}{a^2}}}+h} \quad \text{(F.14)}$$

with a=2.092565e+7 ft, the earth equatorial radius and $b=a(1-f)$, the earth polar radius, where f=1/298.257 the earth flattening parameter.

Bibliography

[1] Baron, S., Kleinman, D.L., and Levison, W.H., "An Optimal Control Model of Human Response, Part I: Theory and Validation", Automatica, Vol. 6, pp. 371-383, Pergamon Press, 1970

[2] Bitmead, R.R., Gevers, M., amd Wertz, V., "Adaptive Optimal Control: The Thinking Man's GPC", Prentice Hall, 1990

[3] Cardullo, F.M., and Brown Y.J., "Visual System Lags: the Problem, the Cause, the Cure", Presented at the IMAGE V conference, Phoenix, Arizona, 1990

[4] Cardullo, F.M., and George, G., "Transport Delay Compensation: An Inexpensive Alternative to Increasing Image Generator Update Rate, "Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1993

[5] Chung, W.W., and Schroeder, A.J., "Visual and Roll-Lateral Motion Cueing Synchronization Requirements for Motion-Based Flight Simulator", Presented at the American Helicopter Society 53rd Annual Forum, Virginia Beach, Virginia, 1997

[6] Cooper, F.R., Harris, W.T., and Sharkey, V.J., "The Effect of Delay in the Presentation of Visual Information on Pilot Performance", NAVTRAEQUIPCEN IH-250), Orlando, FL: Naval Training Equipment Center, 1975

[7] Duva J.S., Harvey J.F., "Delay of Visual Feedback in Aircraft Simulation", NAVTRAEQUIPCEN TN-56, 1977

[8] Feng, G., and Lozano, R., "Adaptive Control Systems", Newnes, 1999

[9] Frank, L.H., Casali, J.G., and Wierwille, W.W., "Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator", Human Factors, *30,* 201-217, 1988

[10] Gorecki, H., Fuksa, S., Grabowski P., and Korytowski A., "Analysis and Synthesis of Time Delay Systems", Prentice Hall, 1989

[11] Gum, D.R., and Albery, W.B., "Time Delay Problems Encountered In Integrating the Advanced Simulator for Undergraduate Pilot Training", Journal of Aircraft, 14, 327-332, 1977

[12] Gum, D.R., and Martin, E.A., "The Flight Simulator Time Delay Problem", Paper 87-2369-CP, AIAA Flight Simulation Technologies Conference, Washington, DC, 1987

[13] Hayes, M.H., "Statistical Digital Signal Processing and Modeling", John Wiley & Sons, Ins., 1996

[14] Howe R.M., "A New Method for On-line Calculation of Dynamic errors in Real-time Simulation", Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1997

[15] Isermann, R., Baur, U., Bamberger, W., Kneppo, P, and Siebert, H., "Comparison of Six On-line Identification and Parameter Estimation Methods", Automatica, Vol. 10, pp. 81-103, Pergamon Press, 1974

[16] Jewell, W.F., Clement, W.F., and Hogue, J.R., "Frequency Response Identification of a Computer-Generated Image Visual Simulator With and Without a Delay Compensation Scheme", Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1987

[17] Johns, L., “A Study of the Effects of Delay Time in a Dome-to-Dome Simulation Link”, Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1988
[18] Kleinman, D.L., Baron, S., and Levison, W.H., “An Optimal Control Model of Human Response, Part I: Theory and Validation”, Automatica, Vol. 6, pp. 357-369, Pergamon Press, 1970
[19] Kwatny, H.G., “A Note on Stochastic Approximation Algorithm in System Identification”, IEEE Transactions on Automatic Control, August, 1972
[20] Landau, Y.D., “Adaptive Control: the Model Reference Approach ”, Press of Marcel Dekker, 1979
[21] Levison, W.H., Baron, S., and Kleinman, D.L., “A Model for Human Controller Remnant”, *IEEE Transactions on Man-Machine Systems, 10*(4), 101-108, 1969
[22] Ljung, L., “System Identification: Theory for the Users”, Second Edition, Upper Saddle River, New Jersey, Prentice Hall, 1999
[23] Ljung, L., Pflug, G., and Walk, H., “Stochastic Approximation and Optimization of Random Systems”, Basel, Boston, Birkhäuser Verlag, 1992
[24] Lusk G.L., Martin C.D., Whiteley J.D., and Johnson, W. V., “Time Delay Compensation Using Peripheral Visual Cues in an Aircraft Simulator”, 1990
[25] Malone, H.L., Horowitz, S., Brunderman, J.A., and Eulenbach, H., “The Impact of Network Delay on a Two-Ship Air-to-Air Combat Simulation”, Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1987
[26] McRuer, D.T., and Krendel, E.S., “Mathematical Models of Human Pilot Behavior”, *AGARDograph AG-188*, Paris: Advisory Group for Aerospace Research and Development, 1974
[27] McFarland R.E., and Bunndell J.W., “Analyzing Time Delays in a Flight Simulation Environment”, ”, Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1990
[28] Miller, G.K., and Riley, D.R., “The Effect of Visual-Motion Time Delays on Pilot Performance in a Pursuit Tracking Task”, Proceedings of the AIAA Visual and Motion Simulation Conference, Washington, DC, 1976
[29] Muckle F.A., and Obermayer R.W., “Control System Lags and Man-Machine System Performance”, NASA Contractor Report-83, Martin company-1964
[30] Nelles, O., “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”, Springer, 2001
[31] Norton, J.P., “An Introduction to Identification”, Academic Press, 1986
[32] Queijo, M.J., and Riley, D.R., “Fixed-Base Simulator Study of the Effect of Time Delays in Visual Cues on Pilot Tracking Performance”, NASA TN D-8001, Hampton, VA: NASA Langley Research Center, 1975
[33] Ricard, G.L., Norman, D.A., and Collyer, S.C., “Compensating for Flight Simulator CGI System Delays”, Proceedings of the 9th NTEC/Industry Conference, NAVTRAEQUIPCEN IH-276), Orlando, FL: Naval Training Equipment Center, 1976
[34] Ricard, G.L., and Puig, J.A., “Delay of Visual Feedback in Aircraft Simulators”, (NAVTRAEQUIPCEN TN-56), Orlando, FL: Naval Training Equipment Center, 1977

[35] Saridis, G.N., “Comparison of Six On-line Identification Algorithms”, Automatica, Vol. 10, pp. 69-79, Pergamon Press, 1974
[36] Saridis, G.N., and Stein, G., “Stochastic Approximation Algorithms for Linear Discrete-Time System Identification”, IEEE Transactions on Automatic Control, Vol. AC-13, No. 5, Oct., 1968
[37] Sen, A., and Sinha, N.K., “On-line System Identification Algorithm Combining Stochastic Approximation and Pseudoinverse”, Automatica, Vol. 11, pp. 425-429, Pergamon Press, 1975
[38] Sinha, N.K., and Griscik M.P., “A Stochastic Approximation Method”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-1, No. 4, Oct., 1971
[39] Sevier, J.A., Minturn, D.B., Bernard, D.W., and Pollard, T.J., “The Effect of Computational Time-Delays on Pilot Performance in Real-Time Flight Simulation”, Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1984
[40] So, R.H.Y., and Griffin, M.J., “Effects of Time Delays on Head Tracking Performance and the Benefits of Lag Compensation by Image Deflection”, Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1991
[41] Uliano, K.C., and Kearns, J.D., “The Effects of Intersimulator Delay on Pilot Performance in Low-Cost Aviation Simulators: A Preliminary Investigation”, Proceedings of the Interservice/Industry Training Systems and Education Conference, National Security Industrial Association, 1992
[42] Whiteley, J.D., and Lusk, S.L. “The Effects of Simulator Time Delays on a Sidestep Landing Maneuver: A Preliminary Investigation”, Proceedings of the Human Factors Society - 31st Annual Meeting, Santa Monica, CA: Human Factors Society, 1990

References

[1] Guo, L., Cardullo, F.M., and Kelly, L.C., "Advanced Transport Delay Compensation Algorithms: Results of Delay Measurement and Piloted Performance Tests", NASA CR 2007-215096, 2007

[2] Smith, R.M., "A Method for Determining Transport Delays in the Flight Simulation Environment", in AIAA Flight Simulation Technologies Conference, 1991, New Orleans

[3] Howe, R.M, "Some Methods for Reducing Time Delays in Flight Simulation", Proceedings of the AIAA Flight Simulation Technologies Conference Washington, DC, 1990

[4] Cardullo, F.M., Kazmarek, M., and Woycechowsky, B, "A Comparison of Several Numerical Integrators for Real Time Flight Simulation: Especially Including Their Impact on Effective Delay and Simulation Accuracy", AIAA Flight Simulation Technologies Conference, 1991, Washington, DC

[5] Galloway, R.T., and Smith R.B., "Cue Synchronization Measurement Using the Piloted Frequency Sweep Technique", Proceedings of the 17th I/ITSEC, Nov 13-16, pp786-196

[6] McFarland, R.E., "CGI Delay Compensation", NASA Ames Research Center, NASA Technical Memorandum, 1986

[7] Cardullo, F.M., and George, G, "Transport Delay Compensation: An Inexpensive Alternative to Increase Image Generator Update Rate", Proceedings of the AIAA Flight Simulation Technologies Conference, 1993

[8] Telban, R J., Prospectus, State University of New York at Binghamton, Binghamton, NY, 2001

[9] Cooper, G.E., and Harper, R.P. Jr., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities", NASA TN D-5153, April, 1969

[10] Crane, D.F., "Compensation for Time Delay in Flight Simulator Visual-Display Systems", Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1983

[11] Bailey, R.E., Knotts, L.E., Horowitz, S.J., and Malone, H.L., "Effect of Time Delay on Manual Flight Control and Flying Qualities During In-Flight and Ground-Based Simulation", Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1987

[12] Riccio, G.E., Cress, J.D., and Johnson, W.V., "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, Santa Monica, CA, 1987

[13] Middendorf, M.S., Lusk, S.L., "Power Spectral Analysis to Investigate the Effects of Simulator Time Delay on Flight Control Activity", Armstrong Aerospace Medical Research Laboratory, 1990 (AIAA-90-3127-CP)

[14] Guo, L., Cardullo, F.M., Telban, R.J., Houck, J.A., and Kelly, L.C., "The Results of a Simulator Study to Determine the Effects on Pilot Performance of Two Different Motion

Cueing Algorithms and Various Delays, Compensated and Uncompensated", AIAA Flight Simulation Technologies Conference, AIAA-5676, 2003
[15] Zaychik, K.B., and Cardullo, F.M., "Simulator Sickness: The Problem Remains", AIAA Flight Simulation Technologies Conference, AIAA-5526, 2003
[16] Uliano, K.C., Kennedy, R.S., and Lambert, E.Y., "Asynchronous Visual Delays and the Development of Simulator Sickness", Proceedings of the Human Factors Society - 31st Annual Meeting, Santa Monica, CA, 1987
[17] Wood, J.R., and Hodgkinson, J., "Definition of Acceptable Levels of Mismatch for Equivalent Systems of Augmented Aircraft", MDC Report A6792, 1980
[18] Ricard, G.L., and Harris, W.T., "Lead/lag Dynamics to Compensate for Display Delays", Journal of Aircraft, 17, 212-217, 1980
[19] Sobiski, D.J., Master Thesis, State University of New York at Binghamton, Binghamton, NY, 1988
[20] McFarland R.E., "Transport Delay Compensation for Computer-Generated Imagery Systems", NASA Ames Research Center, NASA JM-100084, 1988
[21] Sobiski, D.J., and Cardullo, F.M., "Predictive Compensation of Visual System Time Delays", Proceedings of the AIAA Flight Simulation Technologies Conference, Washington, DC, 1987
[22] Sinha, N. K., and Kuszta, B., "Modeling and Identification of Dynamic Systems", Van Nostrand Reinhold Company, 1983
[23] Kushner, G., and Yin, G., "Stochastic Approximation Algorithms and Applications", New York, Springer, 1997
[24] Stevens, B.L., and Lewis F.L., "Aircraft Control and Simulation", John Wiley & Sons, Inc, 1991
[25] Ogata, K., "Discrete-Time Control Systems", Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1994
[26] Brogan, W.L., "Modern Control Theory", Third Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1991

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 12 - 2007	Contractor Report	

4. TITLE AND SUBTITLE

Advanced Control Algorithms for Compensating the Phase Distortion Due to Transport Delay in Human-Machine Systems

5a. CONTRACT NUMBER

NNL06AA74T

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Guo, Liwen; Cardullo, Frank M.; and Kelly, Lon C.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

160961.01.01.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/CR-2007-215095

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 54
Availability: NASA CASI (301) 621-0390

13. SUPPLEMENTARY NOTES

Langley Technical Monitor: Jacob A. Houck
An electronic version can be found at http://ntrs.nasa.gov

14. ABSTRACT

The desire to create more complex visual scenes in flight simulators outpaces recent increases in processor speed. As a result, simulation transport delay remains a problem. Two novel new approaches for compensating the transport delay in a simulator have been developed and are reported in this thesis. These new approaches have been tested and compared to the lead/lag filter, the McFarland compensator and the Sobiski/Cardullo state space filter in terms of phase lead, gain distortion, and complexity. The first novel compensator is an adaptive predictor that uses the Kalman filter algorithm in a unique manner. The second novel approach employed a reference aircraft dynamics model to implement a state space predictor on a simulator. Analyses of data from simulations show that both novel predictors effectively suppress the spikes caused by the McFarland compensator. The phase errors of the predictors are not significant. The adaptive predictor yields greater gain errors than the McFarland predictor for short delays, but shows smaller errors for long delays. The advantage of the adaptive predictor becomes more obvious for a longer time delay. The state space predictor results in smaller gain error than the other predictors for all delay cases.

15. SUBJECT TERMS

Delay Compensation Algorithm; Flight Simulation; Simulator; Transport Delay; Visual System

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	180	(301) 621-0390