

NASA/CR-2007-215096



# Advanced Transport Delay Compensation Algorithms: Results of Delay Measurement and Piloted Performance Tests

*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](mailto: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-2007-215096



# Advanced Transport Delay Compensation Algorithms: Results of Delay Measurement and Piloted Performance Tests

*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**

This report summarizes the results of delay measurement and piloted performance tests that were conducted to assess the effectiveness of the adaptive compensator and the state space compensator for alleviating the phase distortion of transport delay in the visual system in the Visual Motion Simulator (VMS) at the NASA Langley Research Center.

In order to determine the delay, a series of measurements were made on the VMS using a device called the SIMulator Evaluation System, and the baseline transport delay in the visual cueing path was determined to be 90 ms when the frame length of the dynamics computer was 16 ms.

Piloted simulation tests were conducted to assess the effectiveness of two novel compensators in comparison to the McFarland predictor and the baseline system with no compensation. Thirteen pilots with heterogeneous flight experience executed straight-in and offset approaches, at various delay configurations, on a flight simulator where different predictors were applied to compensate for transport delay. Four metrics—the glideslope and touchdown errors, power spectral density of the pilot control inputs, NASA Task Load Index, and Cooper-Harper rating of the handling qualities were employed for the analyses. The overall analyses show that the adaptive predictor results in slightly poorer compensation for short added delay (up to 48 ms) and better compensation for long added delay (up to 192 ms) than the McFarland compensator. The analyses also show that the state space predictor is fairly superior for short delay and significantly superior for long delay compared to the McFarland compensator. The state space predictor also achieves better compensation than the adaptive predictor. The results

of the evaluation of the effectiveness of these predictors, in the piloted tests, agree with those obtained in the theoretical offline tests conducted with the recorded simulation aircraft states.

## **Preface**

This report is the second of two NASA Contractor Reports documenting the research on the 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 is presented in three parts: transport delay measurement in the 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 this report draws conclusions on the delay measurement and piloted simulation tests, and includes suggestions for future research. The appendices of this 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.

The first report, NASA CR 2007-215095<sup>1</sup> 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. That 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 introduced by the McFarland predictor, it presents the main body of research, i.e., development of two novel compensators. That 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 of a reference aircraft model and its integral. Several aircraft models were tested and the large commercial transport landing model 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.



## Table of Contents

<b>ABSTRACT</b> .....	<b>iii</b>
<b>PREFACE</b> .....	<b>v</b>
<b>TABLE OF CONTENTS</b> .....	<b>vii</b>
<b>LIST OF FIGURES</b> .....	<b>ix</b>
<b>LIST OF TABLES</b> .....	<b>xi</b>
<b>NOMENCLATURE</b> .....	<b>xii</b>
<b>ACRONYMS</b> .....	<b>xiv</b>
<b>1. INTRODUCTION</b> .....	<b>1</b>
<b>2. TIME DELAY MEASUREMENT</b> .....	<b>3</b>
2.1. LITERATURE STUDY ON TRANSPORT DELAY MEASUREMENT .....	3
2.2. TIME DELAY MEASUREMENT IN THE VMS .....	7
2.2.1. <i>Scenario I</i> .....	11
2.2.2. <i>Scenario II and III</i> .....	14
2.2.3. <i>Summary</i> .....	14
<b>3. EXPERIMENTAL DESIGN</b> .....	<b>17</b>
3.1. THE MAN-MACHINE SYSTEM—THE SIMULATOR AND THE PILOTS .....	18
3.2. SIMULATION MANEUVERS AND CONDITIONS .....	20
3.3. OTHER SIMULATION CONDITIONS: TIME DELAY AND COMPENSATIONS .....	22
3.4. DATA COLLECTION AND ANALYSIS METHODS .....	23
<b>4. RESULTS OF EXPERIMENTS AND DATA ANALYSES</b> .....	<b>25</b>
4.1. ANALYSIS OF PILOTS' PERFORMANCE .....	25
4.1.1. <i>Glideslope Error</i> .....	26
4.1.2. <i>Touchdown Error</i> .....	29
4.2. ANALYSIS OF PILOTS' HANDLING QUALITY RATING (CHR).....	35
4.2.1. <i>CHR of Maintaining the Glideslope</i> .....	35
4.2.2. <i>CHR on the Touchdown</i> .....	39
4.3. ANALYSIS OF THE TLX .....	42
4.3.1. <i>Analysis on the Overall Weighted TLX</i> .....	43
4.3.2. <i>Analysis on the Six Factors of the TLX</i> .....	46
4.4. ANALYSIS OF THE PSD OF PILOT CONTROL .....	49
4.4.1. <i>About the PSD</i> .....	50
4.4.2. <i>The PSD of the Aircraft Euler Angles</i> .....	52
4.4.3. <i>Analysis of the Integrated PSD of Pilot Control</i> .....	55
4.4.4. <i>Analysis of the Highest PSD Peak Migration</i> .....	60
4.4.5. <i>Analyses on Selected Pilots</i> .....	64
4.4.6. <i>Analysis of PSD in Certain Frequency Intervals</i> .....	70
4.5. SUMMARY OF RESULTS .....	73
<b>5. CONCLUSIONS AND FUTURE RESEARCH</b> .....	<b>81</b>
5.1. CONCLUSIONS .....	81
5.2. SUGGESTED FUTURE RESEARCH .....	86
<b>APPENDIX A. RUN MATRICES AND PRINT VARIABLES</b> .....	<b>87</b>
<b>APPENDIX B. MORE RESULTS OF THE PILOTED TESTS</b> .....	<b>92</b>
B.1. GLIDESLOPE ERROR .....	92
B.2. GLIDESLOPE ERROR .....	100

B.3. COOPER-HARPER RATING .....	107
B.4. TASK LOAD INDEX .....	114
B.5. POWER SPECTRAL DENSITY .....	121
B.5. FREQUENCY OF THE HIGHEST PSD PEAK .....	128
<b>APPENDIX C. MATLAB SOURCE CODES.....</b>	<b>135</b>
C.1. CONVERTING ASCII DATA TO MATLAB DATA STRUCTURE .....	135
C.2. CALCULATE SMOOTHED PSD (PERIODOGRAM).....	137
C.3. FUNCTION (SUBROUTINE) CALLED BY SMPSD.M (APPENDIX C.2).....	138
C.4. ADAPTIVE PREDICTOR: STOCHASTIC APPROXIMATION ALGORITHM.....	139
C.5. ADAPTIVE PREDICTOR: KALMAN FILTER ALGORITHM.....	141
C.6. ADAPTIVE PREDICTOR: KARZMARZ ALGORITHM .....	143
C.7. ADAPTIVE PREDICTOR: LEAST MEAN SQUARE ALGORITHM .....	145
C.8. CALCULATE THE MCFARLAND PREDICTOR COEFFICIENTS.....	147
C.9. STATE SPACE PREDICTOR: 4 <sup>TH</sup> -ORDER REFERENCE MODEL .....	148
C.10. DISCRETE STATE SPACE PREDICTOR: 4 <sup>TH</sup> -ORDER REFERENCE MODEL.....	150
C.11. CALCULATE THE ACCELERATIONS IN 6 DOF OF THE TOPODETTIC FRAME SYSTEM .....	151
C.12. CALCULATE STATE TRANSITION MATRIX USING THE CAYLEY-HAMILTON THEOREM.....	153
C.13. COMPREHENSIVE ANALYSIS OF SOBISKI/CARDULLO FILTER.....	155
C.14. A POLE PLACEMENT FUNCTION CALLED BY THE PROGRAM IN APPENDIX C.12.	159
C.15. DESIGN A 3-VELOCITY PREDICTOR USING FREQUENCY LEAST SQUARES METHOD .....	160
C.16. FUNCTION TO PLOT MULTI-NORMALIZED DATA FOR THE DELAY MEASUREMENT .....	164
C.17. CALCULATE THE PLATFORM ACCELERATIONS FROM ACCELEROMETERS READINGS .....	166
C.18. REALIZATION OF INVERSE NEWTON-RAPHSON TRANSFORMATION .....	167
C.19. DETERMINE THE PHASE ERROR BETWEEN TWO SIGNALS BY CURVE FIT .....	172
C.20. CALCULATE THE PLATFORM ACCELERATIONS FROM ACCELEROMETERS READINGS (C++).....	174
<b>REFERENCES.....</b>	<b>178</b>

## List of Figures

FIG. 2.1. SIGNAL FLOW DIAGRAM OF THE DELAY MEASUREMENT IN THE VMS .....	8
FIG. 2.2. GEOMETRY OF A SIX-DEGREE-OF-FREEDOM MOTION SYSTEM (ADOPTED FROM WU) .....	10
FIG. 2.3. POSITIONS AND DIRECTIONS OF THE SIX ACCELEROMETERS.....	10
FIG. 2.4. PITCH EYE POINT (OD) AND EVADS Y2 (NORMALIZED) .....	11
FIG. 2.5. PITCH ATTITUDE INDICATOR (OD) AND PHOTO DIODE (NORMALIZED).....	12
FIG. 2.6. MOTION COMMAND AND SELECTED MOTION SYSTEM RESPONSES (NORMALIZED) .....	13
FIG. 3.1. VMS AT THE NASA LANGLEY RESEARCH CENTER .....	18
FIG. 3.2. COCKPIT OF THE VMS SIMULATOR.....	19
FIG. 3.3. SCHEMATIC DIAGRAM OF THE OFFSET APPROACH MANEUVER FLIGHT ROUTE .....	21
FIG. 4.1. EXAMPLE OF THE AIRCRAFT ALTITUDE AND GLIDE ERROR.....	26
FIG. 4.2. MEAN AND STANDARD DEVIATION OF THE GLIDESLOPE ERRORS AS A FUNCTION OF DELAY .....	27
FIG. 4.3. MEAN AND STD OF THE INTEGRATED PSD OF THE GLIDESLOPE ERROR WITH DIFFERENT COMPENSATION ALGORITHMS .....	28
FIG. 4.4. ILLUSTRATION OF THE TOUCHDOWN BOX .....	29
FIG. 4.5. ILLUSTRATION OF CALCULATING THE HEIGHT OF THE LANDING GEARS .....	30
FIG. 4.6 AIRCRAFT FORCE, TOUCHDOWN POINT AND TOUCHDOWN ERRORS .....	31
FIG. 4.7. MEAN AND STANDARD DEVIATION OF THE TOUCHDOWN ERRORS CHANGING WITH DELAY .....	32
FIG. 4.8. MEAN AND STD OF THE X-TOUCHDOWN ERRORS CHANGING WITH THE COMPENSATION.....	33
FIG. 4.9. MEAN AND STD OF THE Y-TOUCHDOWN ERRORS CHANGING WITH THE COMPENSATION.....	34
FIG. 4.10. MEAN & STD OF CHR AND GSE CHANGING WITH THE TIME DELAY .....	36
FIG. 4.11. MEAN AND STD OF CHR & GSE WITH COMPENSATIONS: STRAIGHT-IN APPROACH.....	37
FIG. 4.12. MEAN AND STD OF CHR & GSE WITH COMPENSATIONS: OFFSET APPROACH... 38	
FIG. 4.13. MEAN & STD OF CHR AND TDE CHANGING WITH THE TIME DELAY .....	39
FIG. 4.14. MEAN AND STD OF CHR & TDE WITH COMPENSATIONS: STRAIGHT-IN APPROACH.....	41
FIG. 4.15. MEAN AND STD OF CHR & TDE WITH COMPENSATIONS: OFFSET APPROACH... 42	
FIG. 4.16. MEAN VALUE AND STD OF THE TLX AS A FUNCTION OF THE TIME DELAY .....	44
FIG. 4.17. MEAN AND STD OF THE TLX CHANGING WITH COMPENSATIONS .....	46
FIG. 4.18. MEAN AND STD OF MENTAL AND PHYSICAL DEMANDS WITH COMPENSATIONS. 47	
FIG. 4.19. MEAN AND STD OF THE TEMPORAL DEMAND & PERFORMANCE .....	48
FIG. 4.20. MEAN & STD OF THE EFFORT & FRUSTRATION WITH COMPENSATIONS.....	49
FIG. 4.21. AN EXAMPLE OF THE NORMALIZED PSD OF THE ROLL AND PITCH ANGLES .....	53
FIG. 4.22. MEAN VALUE & STD OF THE INTEGRATED PSD CHANGING WITH THE TIME DELAY .....	56
FIG. 4.23. MEAN & STD OF THE INTEGRATED ROLL INPUT PSD WITH COMPENSATION .....	57
FIG. 4.24. MEAN & STD OF THE INTEGRATED PITCH STICK PSD WITH COMPENSATIONS ...	58

FIG. 4.25. MEAN & STD OF THE INTEGRATED RUDDER PEDAL PSD WITH COMPENSATIONS ..... 59

FIG. 4.26. HIGHEST PEAK OF PSD MOVEMENT WITH THE DELAY AND COMPENSATIONS..... 60

FIG. 4.27. MEAN & STD OF THE FREQUENCY OF THE HIGHEST PSD PEAK WITH DELAY..... 61

FIG. 4.28. MIGRATION OF THE HIGHEST ROLL STICK PSD PEAK WITH COMPENSATION..... 62

FIG. 4.29. MIGRATION OF THE HIGHEST PITCH STICK PSD PEAK WITH COMPENSATION..... 63

FIG. 4.30. MEAN & STD OF THE GLIDESLOPE ERROR FOR PILOTS #1,2,5,11,12 ..... 66

FIG. 4.31. MEAN & STD OF THE TOUCHDOWN ERROR FOR PILOTS #1,2,5,11,12..... 67

FIG. 4.32. MEAN & STD OF INTEGRATED ROLL INPUT PSD FOR PILOTS #1,2,5,11,12 ..... 68

FIG. 4.33. MEAN & STD OF INTEGRATED PITCH INPUT PSD FOR PILOTS #1,2,5,11,12 ..... 69

## LIST OF TABLES

TABLE 2.1. VARIABLES RECORDED BY THE SIMES FOR THE DELAY MEASUREMENT .....	9
TABLE 2.2. TRANSPORT DELAYS IN THE THREE BASIC CUEING CHANNELS .....	12
TABLE 2.3. TRANSPORT DELAYS IN THE THREE BASIC CUEING CHANNELS .....	14
TABLE 2.4. TIME DELAYS OF DIFFERENT SOURCES AND THE TOTAL DELAYS .....	15
TABLE 3.1. STRAIGHT-IN APPROACH CONDITIONS.....	20
TABLE 3.2. OFFSET APPROACH CONDITIONS .....	22
TABLE 4.1. MEAN & STD OF THE FREQUENCY OF THE HIGHEST PEAK OF PSD OF EULER ANGLES .....	54
TABLE 4.2. FREQUENCY INTERVALS IN WHICH THE PSD INCREASES BY TIME DELAY .....	71
TABLE 4.3. FREQUENCY INTERVALS IN WHICH THE PSD DECREASES BY DIFFERENT PREDICTORS .....	72
TABLE 4.4. FREQUENCY INTERVALS OF REDUCED PSD BY COMPENSATION OF DIFFERENT DELAYS .....	73
TABLE 4.5. SUMMARY OF COMPENSATION RESULTS .....	75
TABLE 4.6. COMPENSATION RESULTS FOR THE STATE SPACE PREDICTOR WITH SHORT DELAY .....	76
TABLE 4.7. SUMMARY OF SIGNIFICANT COMPENSATIONS .....	79
TABLE 4.8. SUMMARY OF SIGNIFICANT DIFFERENCE AMONG THE FOUR PREDICTORS .....	80
TABLE A.1. STRAIGHT-IN APPROACH RUN MATRIX .....	87
TABLE A.2. OFFSET APPROACH RUN MATRIX.....	88
TABLE A.3. PILOTED TEST EVALUATION PRINTED VARIABLES .....	89
TABLE A.4. NASA TLX SUBSCALE RATING DEFINITIONS (FROM TELBAN’S THESIS) .....	90
TABLE A.5. RESUMES OF THE PILOT SUBJECTS.....	91

## Nomenclature

$A_c$	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
$\hat{k}_x$	Autocorrelation sequence of signal $x$
$l$	Aircraft longitude (arc length)
$l_\theta$	Aircraft longitude (angle)
$P_x$	Exact power spectral of signal $x$ from discrete Fourier Transform
$\hat{P}_x^f(\theta)$	Estimated power spectral of signal $x$
$p, q, r$	Aircraft angular velocities in the body frame
$t_d$	Time delay
$T$	Sampling period
$u$	Pilot control input

$v$	Velocity of $y$ ; reference input
$w$	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
$y$	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
$\phi$	Aircraft roll angle
$\Phi_d$	State transition matrix for discrete system
$\lambda$	Aircraft latitude (arc length); forgetting factor
$\lambda_\theta$	Aircraft latitude (angle)
$\theta$	Aircraft pitch angle
$\tilde{\theta}$	Estimate of $[b_0 \ b_1 \ b_2]^T$ (output of the Kalman filter algorithm)
$\psi$	Aircraft yaw angle
$\omega$	Angular frequency

## Acronyms

AP	Adaptive Predictor
CHR	Cooper-Harper Rating
DFT	Discrete Fourier Transform
DOF	Degree Of Freedom
EOM	Equations Of Motion
EVDAS	Electronic Visual Data Acquisition System
FRA	Frequency Response Analyzer
GS	Glide Slope
GSE	Glide Slope Error
HQR	Handling Quality Rating
LSF	Least Squares Fitting
MF	McFarland predictor
MFR	McFarland predictor with spike Reduction
NC	No Compensation
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
SS	State Space predictor
STD	STandard Deviation
TD	TouchDown
TDE	TouchDown Error
TLX	Task Load Index
VA	Visual to Analog
VMS	Visual Motion Simulator

## **1. Introduction**

Two novel predictors for compensating the transport delay in a flight simulator, the adaptive compensator and the state space compensator, as well as the McFarland compensator were examined in NASA CR 2007-215095. That report also presents the results of the offline tests (i.e., tests without a pilot in the loop), which compare gain and phase errors of the predictions of the three compensators. In order to further assess the effectiveness of these predictors, they were implemented in the Visual Motion Simulator (VMS) at the NASA Langley Research Center, and simulation tests were conducted with 13 pilots who executed straight-in and offset approaches. This report is a summary of the piloted test results.

To compensate for the transport delay in a simulator, the actual delay in the system must be known prior to designing and implementing a compensator. Therefore, delay measurement tests were conducted in the VMS using a comprehensive device called SIMES. The measurement is covered in Chapter 2. The first section of Chapter 2 summarizes a study of the literature on past transport delay measurements, concentrating on measurement domains and methods, devices and data collection. The second section describes, in detail, the delay measurement in the VMS, including the devices used, the test topology, and the data collection and analysis. The average total visual transport delay has been determined to be 90 ms when the simulation host computer's update cycle is 16 ms. This forms a solid basis for the final experiment design and delay compensation.

Chapter 3 describes the design of the final piloted tests. It includes information on the flight experience of the pilot subjects, a description of the simulation maneuvers and conditions, a matrix of the artificial delays inserted into the simulator, a discussion of the

compensation algorithms implemented, and a brief summary of the experimental data collected.

Chapter 4 assesses the results of the final piloted tests. These assessments are based upon four metrics: 1) the glideslope error during the approach phase, and the longitudinal and lateral error at touchdown; 2) the pilot's perception of the handling qualities (Cooper-Harper ratings); 3) the NASA Task Load Index (TLX); and 4) the power spectral density (PSD) of the pilot's control input. With the PSD, both integrated PSD and the migration of the highest peak are considered.

Some intermediate results of the piloted tests are illustrated with bar graphs in Appendix B. Appendix C contains source code and MATLAB scripts used during the piloted tests and the data analysis contained in both contractor reports.

## 2. Time Delay Measurement

Measuring the transport delay in a flight simulator is necessary and important because it provides the information necessary to investigate various relevant issues. For example: Which parts of the simulator contribute substantially to the transport delay? Delay reduction efforts should be focused on the major contributors. Are there any cueing mismatches among the primary cueing channels? This information is helpful when studying the effects of cueing mismatches in a flight simulation system. Do different simulation conditions, such as the maneuver itself, or the aerodynamic model, change the system transport delay? What is the baseline system delay? This information is necessary when deciding whether to apply compensation, and when designing the parameters of the compensator. What is the quantitative relationship between the delay and the performance of the operator, and what is the maximum tolerable delay in a flight simulator?

This chapter will present a brief summary of the literature study of transport delay measurement, and then introduce the delay measurements conducted in the NASA Visual Motion Simulator (VMS), and finally give the measurement results.

### 2.1. Literature Study on Transport Delay Measurement

Transport delay may be measured with either time domain or frequency domain methods. These two domains are related by the dual transform pair of the time delay as given by  $f(t-t_d) \Leftrightarrow e^{-j\omega t_d} F(\omega)$ . In the time-domain measurement, a step input or other signals with sharp onset are used to excite the selected control-cue loop, and the initial system response is measured. The delay is determined as the difference between the onset of the change in the output and the corresponding change in the input. Because of the

aerodynamic model, the change in output, corresponding to the onset of the pilot input may not be noticeable, and this is the primary disadvantage of the time domain methods. For this reason, in some literature, 63 percent of the commanded final value is defined as the onset of the simulator response<sup>2</sup>. Another drawback of the time-domain methods is that the delay values are subjected to sampling uncertainty.

In the frequency-domain measurement, sinusoid signals of different frequencies are employed to excite the particular control-cue loop, and the system's responses are recorded. Simply stated, the delay is obtained by determining the phase difference  $\Phi_d$  between the input and output, and then calculating the time delay using:

$$t_d = \frac{\Phi_d(\omega)}{\omega} \quad (2.1)$$

where  $\omega$  is the frequency of the sinusoidal input. But, because of the aerodynamic model, the output signal is usually not sinusoidal, and in such cases, the delay is approximated by an equivalent delay for a specific frequency range, where the gain of the system frequency characteristics is relatively constant and the phase angle increases proportionally with the input frequency. A sweep input signal is usually used to replace the pure sinusoidal signal. A sweep signal is a frequency-varied sinusoid, which starts at a very low frequency, and the frequency increases linearly up to several hertz.

The disadvantages of the time domain methods are the advantages of the frequency domain methods. In the frequency domain it is not necessary to determine the onset of the output, and the measurement does not depend on the timing of the sampling. In addition, frequency domain methods can be used to characterize the system response over the whole bandwidth of the man-machine simulator system. The main drawback of the frequency domain methods is that the spectral analysis is significantly more

complicated than those in the time domain. A second weakness is that the constant gain and linear phase range do not always exist in the system frequency characteristics.

Many researchers have conducted experiments to measure the transport delays or cueing mismatches in many types of flight simulators using many different methods. Here is a brief summary of some measurements, and a representative example, with a focus on the simulator type, the analysis method, the cueing channels studied and the devices used.

Glen Niemeyer and Barry Dougherty<sup>3</sup> measured the transport delay in a fixed-base (without motion system) flight simulator with a visual scene. The aerodynamic model was bypassed and replaced with a constant gain in order to make the onset of the output easy to detect. The operator control was monitored with a potentiometer, and the visual response was sensed with a photometer; both devices output signals to a chart recorder, from which the transport delay was determined in the time domain. It is worth mentioning that some extrapolation methods were employed to offset part of the transport delay.

Scott Horowitz<sup>4</sup> made transport delay measurements on a high performance fighter-type aircraft simulator, also without a motion system. He measured the existing delays in different components of the simulator, such as the aircraft model, and the visual system. Since the control input was created by a square wave generator, it was unnecessary to sense the input signal. The visual scene was sensed with a special device called a visual-to-analog (VA) converter, which transforms the image on a CRT into an analog signal. The aerodynamic model was either in use, or bypassed, so that its delay could be identified. The signals from the square wave generator, the simulation computer

and VA converter were all input to a strip chart recorder. The delays were determined in the time domain. This study showed that the dynamic model not only increased the delay, but also increased the measuring variance.

Marshall Smith<sup>5</sup> conducted experiments to measure the transport delays in the Flight Simulation Facilities at the NASA Langley Research Center (with instrumentation, without motion system), using both frequency and time domain methods. In the time domain approach, a strip chart recorder was used to collect data on everything except the visual image, which was sensed by a photo-electric diode or a video level detector. All these were input to a logic analyzer. In the frequency domain method, a frequency response analyzer (FRA) generated the input and collected the output signals, and calculated the phase difference. The delays were calculated using Eq. (2.1). His approach was unique in that the control input could be inserted into the normal control interface, or, it could be inserted in many different nodes. In this manner, the output of each component was easy to analyze. Another feature of his approach is that the data transfer delay was measured. And, although most of these measurements were conducted in the time domain, Smith's study yielded results in the time domain and the frequency domain that match closely.

Thomas Galloway<sup>6</sup> used a unique piloted frequency sweep technique to measure the cue-synchronization in a simulator with visual, motion and instrument cueing channels. The data were digitally sampled using a commercially available data acquisition board installed in a PC. Software spectral analyses were employed to obtain the time delays in the individual channels to yield the cue mismatches. It is worth

mentioning that the visual and motion delays measured were quite large: 940 ms and 770 ms, respectively.

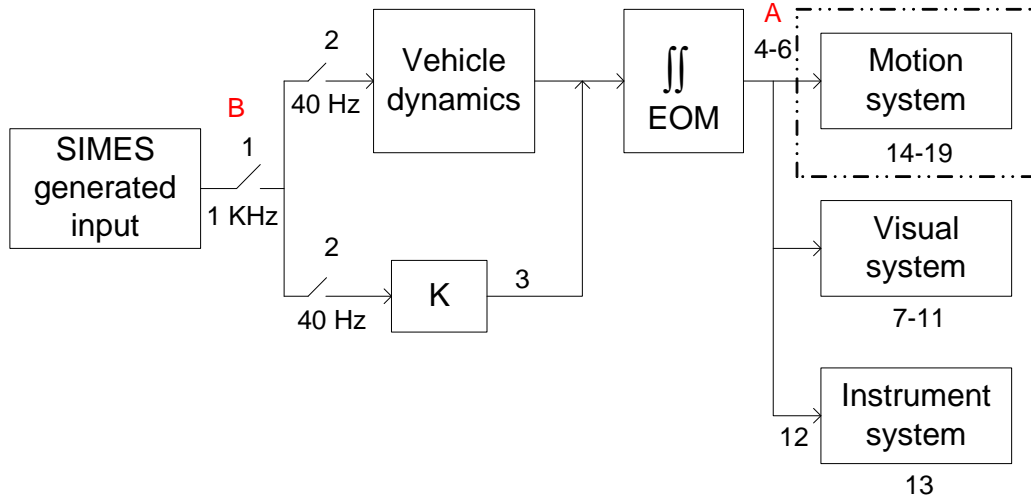
Therefore, Galloway is the only author known to have measured the delay in the motion system, and he used a software package to do the frequency analysis instead of a FRA.

## **2.2. Time Delay Measurement in the VMS**

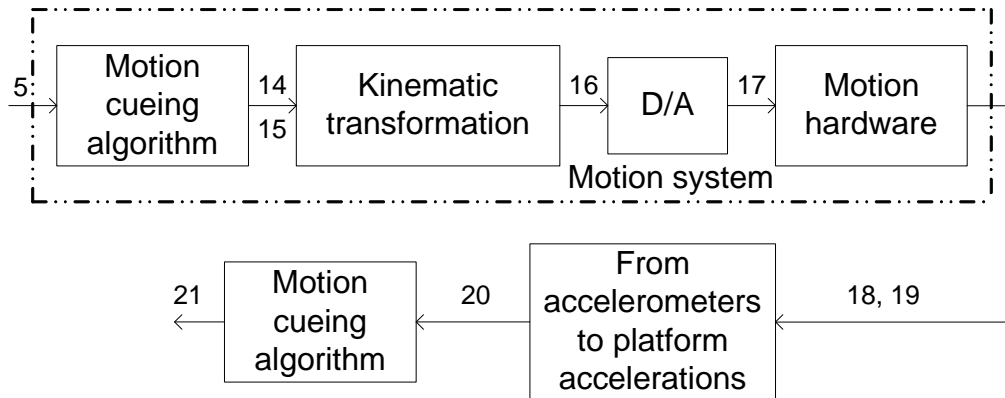
A series of tests were conducted to measure the transport delays in the Visual Motion Simulator (VMS) at the NASA Langley Research Center, using a device called SIMES<sup>7</sup> (the SIMulator Evaluation System). SIMES was developed by SAIC for the Naval Air Warfare Center Training Systems Division, in cooperation with the Wright-Patterson Air Force Base. It is a comprehensive, non-intrusive, accurate and reliable system to test, debug, and document the cueing systems and dynamic models in various kinds of simulators. SIMES can also generate control signals in place of an operator for unmanned simulations. Thus the SIMES provides a combination of the functions of a control loader, a logic analyzer and a video level detector. It can inject sinusoidal signals and sweep signals into the simulator, and collect responses from it. However, the SIMES unit cannot process frequency response analysis (it is not a FRA). Fig. 2.1 shows where the SIMES generated input was added as well as where various data were collected on the VMS. The input signal was inserted at either point A, so that the aircraft states were input to the cueing systems directly from the SIMES (Scenario I), or at point B, so that the aircraft states were provided by the EOM. In the latter case the aircraft model was either bypassed, by replacing it with a constant gain  $K$  (Scenario II), or in use (Scenario III). The SIMES generated the input and collected the data (except the EVADS visual



signals) at the rate of 1000 Hz. The mainframe computer update frequency was 40 Hz, and the visual computers ran at a rate of 60 Hz. Therefore, some asynchrony exists in the communications. The EVADS signals were recorded at 60 Hz.



(a)



(b)

**Fig. 2.1. Signal flow diagram of the delay measurement in the VMS**

The numbers in Fig.2.1 identify the data items collected from the VMS by the SIMES, and they are tabulated in Table 2.1. Specifically, the five EVDAS signals were obtained in the visual system and were used to measure the visual system transport delay. A photodiode was used to measure the instrument system transport delay. Signals 14-19

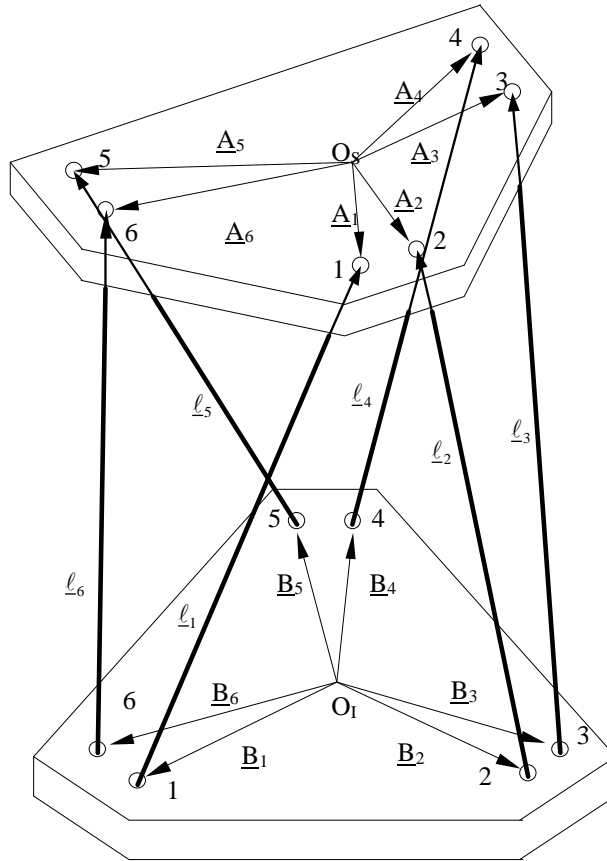
were used to measure the motion system delay. Data labeled 14-19 are located sequentially in the motion system, as depicted in the lower figure in Fig. 2.1.

**Table 2.1. Variables recorded by the SIMES for the delay measurement**

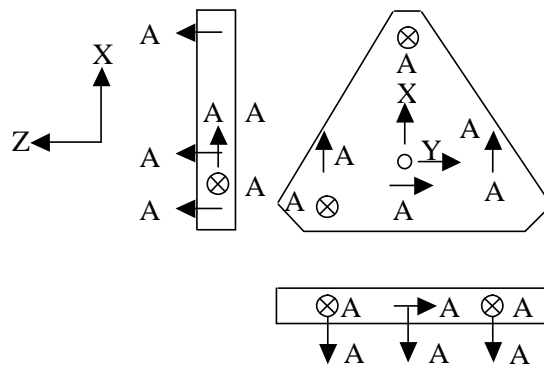
No.	Name
1	Control Device Position Over Drive
2	Control Device Position
3	Accelerations
4,5,6	Rates, Attitudes, Pilot Eye Point
7-11	EVDAS Signals
12	Attitude Indicator
13	Photo Diode
14,15	Attitude& Rate Motion Commands
16	Leg Drive Commands
17	Simulator Leg Commands
18	Accelerometers
19	Leg Positions
20	Accelerations from the Accelerometers
21	Filtered Values of 20

Fig. 2.2 shows the geometry of a six-degree-of-freedom synergistic motion system. The leg positions (labeled 19) were measured with potentiometers, and were then transformed into the position and orientation of the payload platform, using a recursive inverse Newton-Raphson transformation. The position and orientation were then used to determine the delay of the motion system. In addition, six accelerometers, which were mounted on the payload platform as shown in Fig. 2.3, sensed accelerations. These accelerations were then transformed to the payload platform accelerations in all six DOF's (labeled 20) using a kinematic transformation. The motion system transport delay was also measured from the leg positions sensed by the potentiometers. But, because the potentiometers caused larger errors than the accelerometers, the final results of the

motion system delay were taken from the accelerometer data. A 2<sup>nd</sup>-order filter was applied to the accelerometer data to reduce the signal noise inherent in the accelerometers (labeled 21).



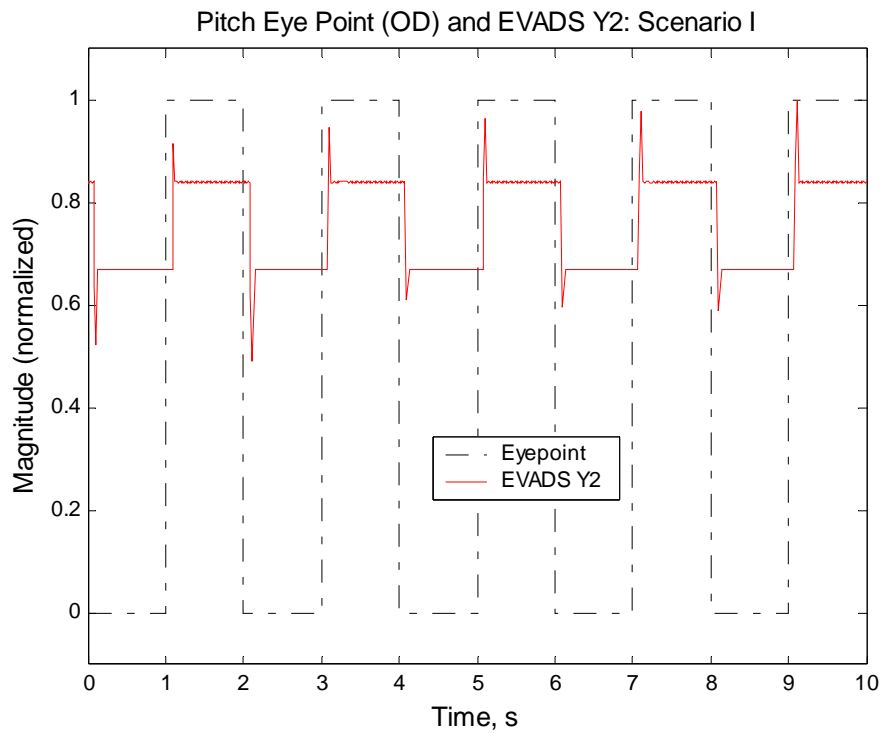
**Fig. 2.2. Geometry of a six-degree-of-freedom motion system (Adopted from Wu)**



**Fig. 2.3. Positions and directions of the six accelerometers**

### 2.2.1. Scenario I

For scenario I, the SIMES input was inserted at point A in Fig. 2.1 so that both the aircraft model and the EOM were skipped. Therefore, the output signals of the visual system and the instrument system closely resemble the input. Fig. 2.4 and Fig. 2.5 show the normalized input (doublet train) and output signals of these two cue channels (EVADS Y2 and photodiode readings). The transport delay in each of the two cue channels was obtained as the average of the delays determined from the 10 transition points, and was further averaged across the five test runs conducted. The results are given in Table 2.2.



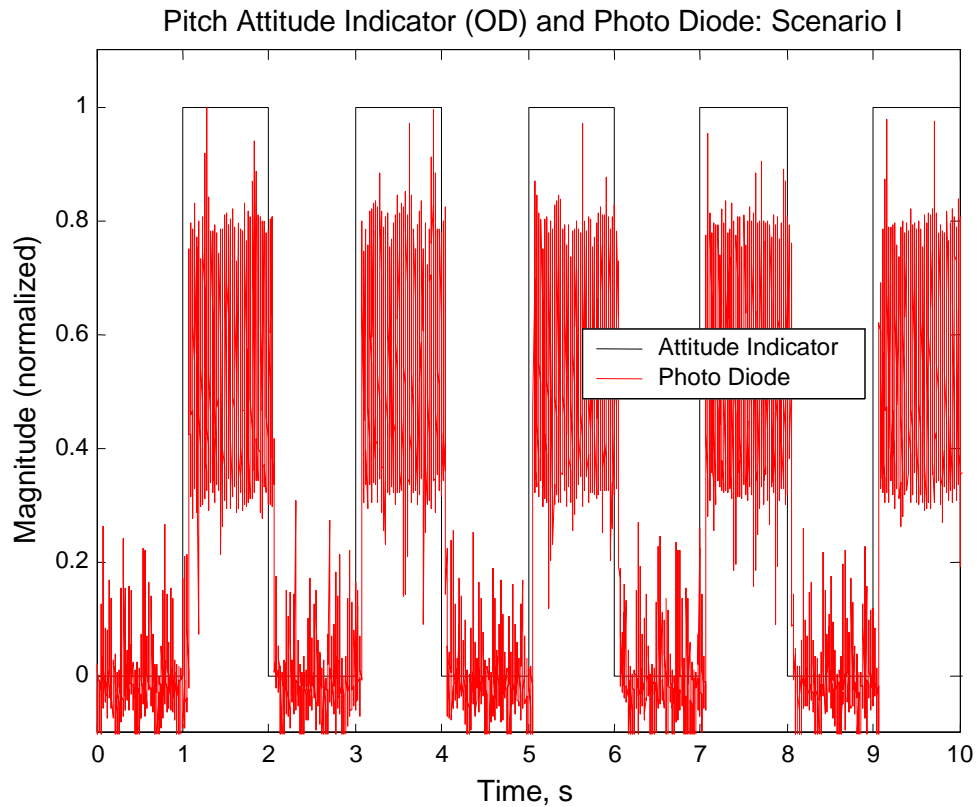
**Fig. 2.4. Pitch eye point (OD) and EVADS Y2 (Normalized)**

The delay was 58 ms in the visual channel and 40 ms in the instrument channel, with a cueing mismatch of 18 ms. There were 14 and 12 ms sampling latencies in the two channels in the test, both of which are within one frame (16.7 ms). They were not

included in the results because the sampling latency behaves as communication asynchrony, which will be addressed later. The theoretical average time delay in the visual was 3.5 frames, that is, 58.4 ms if the update cycle of the visual system was 16.7 ms. Therefore, the measuring error for the visual delay was small.

**Table 2.2. Transport delays in the three basic cueing channels**

Cue Channels	Visual		
	Visual	Motion	Instrument
Delay (ms)	58	56	40



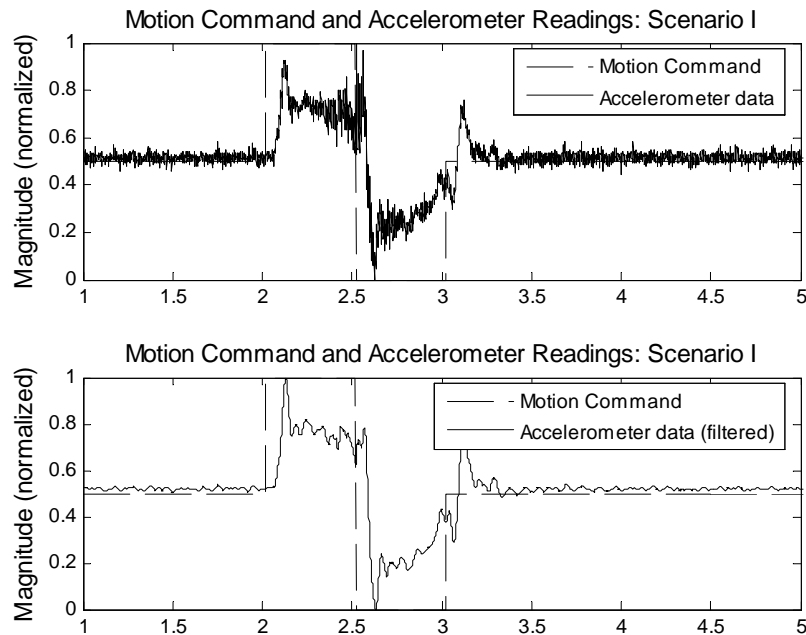
**Fig. 2.5. Pitch attitude indicator (OD) and Photo diode (Normalized)**

The input signal used to measure the delay in the motion system was either a step or a doublet. The response of the motion system was sensed with accelerometers (for the platform accelerations) and potentiometers (for leg positions). Fig. 2.6 shows the readings

from these two devices (for the doublet input only), along with some intermediate responses in the motion system (refer to Fig. 2.1 and Table 2.1). The accelerometer readings were used, rather than the position feedbacks (potentiometers), because of the large potentiometer latency. But, because of the noise inherent in the accelerometer signals, the signals were filtered before the delays could be determined from them. The 2<sup>nd</sup>-order digital filter used was

$$\frac{Y(z)}{X(z)} = \frac{0.0117}{z^2 - 1.8318z + 0.8546} \quad (2.2)$$

which corresponds to a damping ratio of 0.5 and natural frequency of 25 Hz. The acceleration curve in the upper plot of Fig. 2.6 is the raw data, and the acceleration curve below is the filtered signal. Both the raw and filtered data were employed to determine the delay. When averaged, the transport delay in the motion system alone for both step and doublet inputs, was 56 ms. Again, the sampling latency of 10 ms is not included.



**Fig. 2.6. Motion command and selected motion system responses (Normalized)**

### 2.2.2. Scenario II and III

In both scenarios, the SIMES input signal was inserted at point B in Fig. 2.1, the only difference being whether the aircraft model was skipped (scenario II) or in use. For either case, the onset of the response was difficult to identify. Therefore, a least squares curve fit was employed to fit different time segments of the responses. In this manner the intersection of the fitting curves was treated as the starting point, corresponding to the onset of the input. Doublet signals with varied duration were used as input signals. The total transport delays from the aircraft control input to the motion base and visual display are given in Table 3.2, where the sampling latency and communication asynchrony are included. All dynamic model processing, EOM integrations and various coordinate transformations were completed in a single frame, and the delay of most of the intermediate data could be determined with small errors. Processing of the aerodynamic model consumed only 7 ms of a frame.

**Table 2.3. Transport delays in the three basic cueing channels**

<b>Aerodynamics Skipped?</b>	<b>Delay (ms)</b>	
	Motion	Visual
Yes (Scenario II)	88	106
No (Scenario III)	101	132

### 2.2.3. Summary

Notice that the values in Table 2.3 are averages with considerably large variances due to the difficulty associated with determining the onset of the response and the signal noise. This suggests that measuring the total transport delay as described in scenarios II and III is not the best approach. This assertion is confirmed by the fact that the frame

length of the simulation mainframe computer must be varied as a function of the aircraft model and simulation task. This means that the total transport delay measured in one situation does not apply to another. Fortunately, this problem may be solved given that: 1) the visual system computer's frame is fixed to 16.7 ms, and the delays in the individual cueing channels can be measured accurately; 2) the delays associated with the aircraft states from the aerodynamics and the EOM can be determined with high accuracies. Therefore, a good measure of the total transport delay can be obtained using two steps: first, measure the individual cueing subsystem delay as in scenario I, and second, calculate the total delay with

$$t_d = 1.5T_m + \bar{t}_a + t_c \quad (2.3)$$

where  $T_m$  is the frame time of the mainframe computer,  $\bar{t}_a$  is the average asynchronous delay, and  $t_c$  the transport delay in a cueing channel. On average it takes a half of a frame for a control input change to be sensed by the mainframe computer (this is known as sampling latency), and the simulation requires one frame to process the aerodynamic model and the EOM, therefore the total average time required to update the aircraft states is  $1.5T_m + \bar{t}_a$ , which can be calculated with Eq. (2.13), NASA CR 2007-215095.

**Table 2.4. Time delays of different sources and the total delays**

Main frame (ms)	Cue	Delay (ms)									
		$t_m$			$t_a$			$t_c$	Total $t_d$		
		Max	Ave.	Min	Max	Ave.	Min		Max	Ave.	Min
25	Visual							58	124.7	103.8	93
	Motion	50	37.5	25	16.7	8.3	0	56	122.7	101.8	91
	Instrument							40	106.7	85.8	75
16	Visual							58	105.3	<b>89.7</b>	84
	Motion	32	24	16	15.3	7.7	0	56	103.3	87.7	82
	Instrument							40	87.3	71.7	66



For this measurement,  $T_m$  is 25 ms, and for the final piloted tests, it was set to 16 ms. The values of all three terms in Eq. (2.3), the maximums and minimums, if any, and the total delays in the three basic cueing channels for these two mainframe computer frame lengths are summarized in Table 2.4.

Therefore, the average total delay (baseline delay) in the visual channel in the final tests will be 90 ms (approximation of 89.7 ms highlighted in bold in Table 2.4).

### **3. Experimental Design**

The theoretical analyses presented in Chapter 5, NASA CR 2007-215095, are based on offline tests of the predictors, which were conducted using aircraft state data recorded during previous simulations. This is an open loop process because there is no visual display, and there is no operator at all. Instead, the purpose of the offline tests is to compare the predicted aircraft states to the undelayed ones in order to evaluate the prediction error. But, these open loop, offline tests are not sufficient to demonstrate effectiveness due to the lack of interaction with the operator.

The piloted test is illustrated in Fig. 1.4, NASA CR 2007-215095. The compensators were implemented where the block “Prediction” is located, which generated predicted (future) aircraft states from the current aircraft states (outputs of the EOM). Artificial delay could be added to the predicted aircraft states by using a FIFO buffer. In this manner, the visual system would retrieve the aircraft states from the buffer, after the specified delay, and use this data to generate the visual image. The compensators were designed to give the predicted states a phase lead equivalent to the baseline delay plus the added delay. Thus the aircraft states, after the added delay, were still ahead of the EOM-generated states by the amount of the baseline delay. This amount of prediction was designed to offset the transport delay in the visual system. If the compensator were ideal, the pilot would sense no delay in the visual image.

This chapter presents the experimental design for the final tests, including a description of the pilots involved, the simulation tasks and maneuvers, the amount of transport delay, compensation techniques, data collection, and evaluation methods.

### 3.1. The Man-Machine System—the Simulator and the Pilots

All simulation tests were conducted using the Visual Motion Simulator (VMS), shown in Fig. 3.1, at the NASA Langley Research Center, in Hampton, Virginia. The VMS is a general-purpose simulator consisting of a two-crewmember cockpit mounted on a 60-inch stroke six-degree-of-freedom synergistic motion base<sup>8, 9</sup>. The relative extension and retraction of the six hydraulic actuators of the motion base provide the motion cues. The latest NASA/SUNY<sup>10</sup> nonlinear motion cueing algorithm was used because it has been proven to have better performance than other prominent ones.



**Fig. 3.1. VMS at the NASA Langley Research Center**

The instrument cues are provided by gauges on the panel in the cockpit of the VMS, as shown in Fig. 3.2, and by three heads-down CRT displays—a generic electronic primary flight display, an electronic horizontal situation indicator display, and a generic electronic engine display. The visual cues are provided by a wide angle, collimated, out-the-window, display driven by an Evans and Sutherland ESIG 3000/GT computer generated image system.

There are three primary controls available in the cockpit: a control loaded two-axis side stick (roll and pitch), a control loaded rudder pedal system, and a four-lever throttle quadrant. Other control inputs include a flap handle, a speed brake handle, a slats handle, etc.

During the simulation scenarios, a high fidelity, highly nonlinear mathematical model of a large commercial transport aircraft was used. This simulation included a landing gear dynamics model, gust and wind models, flight management systems, and flight control computer systems. For this study, the test subjects flew the aircraft in the manual control mode (no autopilot) and with manual throttle control (no autothrottle).



**Fig. 3.2. Cockpit of the VMS simulator**

In order to perform statistical analyses and avoid the possibility of general conclusions based on one special case, 13 subjects with varying aircraft and flight experience flew in the piloted study. A brief resume for each pilot is available in Table A.5. All 13 test subjects were volunteers, and therefore their training and flight experiences were not as homogenous as desired. For most of the pilots, the tests were

conducted over a 6-hour period, with approximately three hours of familiarization and training, and then three hours of the actual test scenarios.

### 3.2. Simulation Maneuvers and Conditions

The final piloted simulation tests consisted of two flight scenarios: (1) a straight-in approach, and (2) an offset approach.

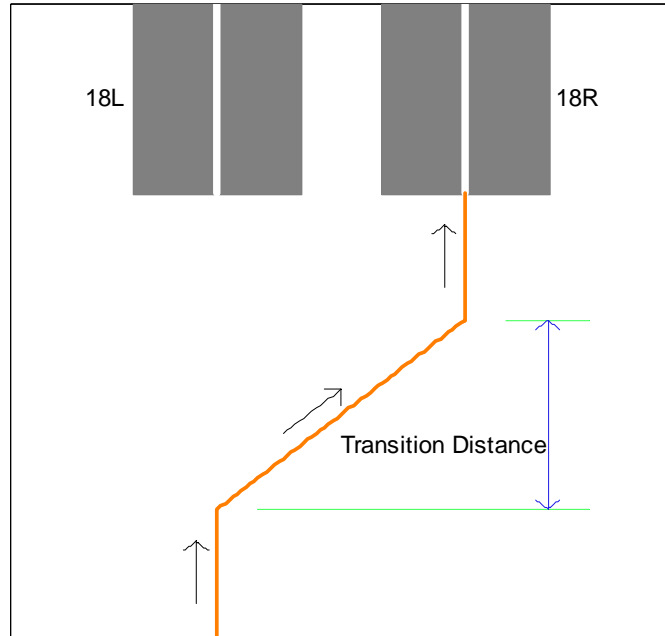
In the straight-in approach, the airplane landed directly on a runway (Runway 18R) from the initial conditions listed in table 3.1. The straight-in approach included a 10 knots wind, which began as a head wind, swung around to a 90 deg wind from the left mid-way through the approach, and continued to swing around to a tail wind as the aircraft crossed the threshold.

**Table 3.1. Straight-in approach conditions**

Altitude	1300 ft BARO, 697 ft AGL
Airspeed	135 kts
Heading angle	180 deg
Distance to Runway	2 nm
Flaps, Gear	Full, Gear-down
EPR	1.19
Autopilot	OFF
Autothrottle	OFF
Glideslope	ON Glideslope
Localizer	ON Localizer

In the offset approach, the aircraft was initially aligned with the left runway (Runway 18L); and at a certain point the pilot was instructed to roll to the right, and line up with the adjacent runway (Runway 18R), within some transition distance. The aircraft

then landed on Runway 18R. The initial conditions for the offset approach are included in Fig. 3.3.



**Fig. 3.3. Schematic diagram of the offset approach maneuver flight route**

The offset approach included a lateral gust from the left, 90 deg to runway centerline, which came on at 3000 ft from the runway threshold, and turned off as the aircraft crossed the threshold, and severe turbulence. A complete list of the initial conditions for the offset approach is included in Table 3.2.

**Table 3.2. Offset approach conditions**

Altitude	1300 ft BARO, 697 ft AGL
Airspeed	135 kts
Heading angle	180 deg aligned with Runway 18L
Distance to Runway	2 nm
Flaps, Gear	Full, Gear-down
EPR	1.19
Autopilot	OFF
Autothrottle	OFF
Glideslope	ON Glideslope

### **3.3. Other Simulation Conditions: Time Delay and Compensations**

In Chapter 2, the average time delay in the VMS was determined to be 90 ms. In order to test the compensation capabilities of the McFarland predictor and the two novel predictors, artificial delay was inserted into the simulation in addition to the 90 ms baseline transport delay. In the preliminary tests, the Pade approximation was used to add 0, 50, 100 and 200 ms of artificial delays to the system, none of which was an integer multiple of the simulator computer frame. In the final tests, the added delay amounts were changed to 0, 48, 96 and 192 ms, which are integer multiples of the update period (16 ms). The artificial delay was added by storing the predicted aircraft states in a FIFO buffer, where they were held for 0, 3, 6 and 12 frames before being output to the visual system. The total delay in the simulation was thus 90, 138, 186 or 282 ms for the four cases.

There were five compensation options: no compensation, the original McFarland filter, the McFarland predictor with spike reduction, the adaptive predictor, or the state

space predictor. Each subject completed 40 test runs: 20 for the straight-in approach, and 20 for the offset approach. The 20 tests runs for each maneuver came from the enumeration of the 4 time delay options and 5 compensation options. The order of the runs, in each maneuver group, was randomized, e.g., the zero delay case was not necessarily the first run, and an uncompensated case was not necessarily followed by a compensated case. The run matrix of the two maneuvers is included in Appendix A.

### **3.4. Data Collection and Analysis Methods**

All test runs were recorded using a DVD and videotape. The recorded images include a camera pointed at the motion system from the rear, a camera pointed at the motion system from the side, the pilot's out-the-window display, and the pilot's primary flight display (EADI).

Over 60 data items were recorded from the simulation. These include:

- 1) The four pilot control input signals: roll stick, pitch stick, rudder pedal and the throttle;
- 2) The aircraft accelerations, velocities and displacements in each of the 6-DOF, in the geodetic frame, which contributes 36 items;
- 3) Glideslope and localizer error and X and Y position of the aircraft.
- 4) Other items, including the motion system position and acceleration, the wind velocity, and data that is displayed on the pilot's instrument system.

The print parameters are included in Appendix A. These variables were collected every fourth frame for the duration of each simulation test run. Therefore, the time gap between two consecutive data points is 64 (16×4) ms.



The power spectral density analysis (PSD) was applied to the four pilot control inputs to evaluate the operator performance and workload, which were expected to be affected by the time delay and compensation. The PSD was also used to analyze the frequency characteristics of some aircraft states.

As stated in Chapter 2, NASA CR 2007-215095, the pilot workload, in terms of the PSD, is an objective metric. In addition to the pilot input PSD, two other objective metrics—the aircraft glideslope error and touchdown error were employed for analysis as well.

The NASA TLX (Task Load Index) data and the Cooper-Harper rating data were also collected and analyzed. The TLX and CHR are two quasi-objective metrics assessing the pilots' workload and aircraft handling quality, respectively.

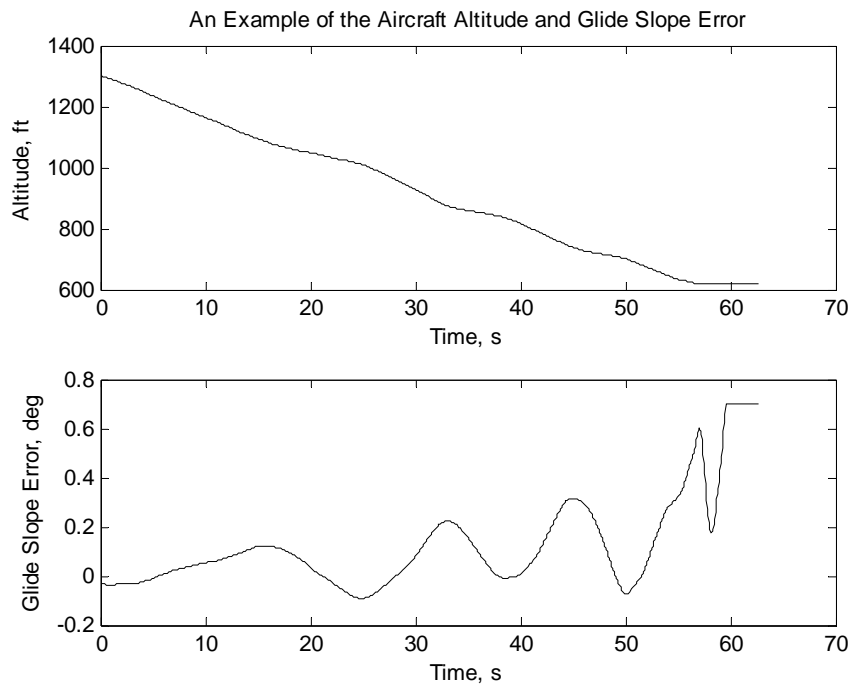
## **4. Results of Experiments and Data Analyses**

The final piloted simulation tests were conducted using the NASA Langley Visual Motion Simulator to assess the relative effectiveness of the two novel compensation algorithms along with the McFarland compensator. Thirteen pilots executed 520 tests from mid October 2004 to late January 2005. The details of the simulation maneuvers and other conditions are available in the previous chapter. Four types of analyses of the collected data have been completed to evaluate the results, they are: 1) the pilots' performance in minimizing the glideslope error and touchdown point error; 2) the pilots' handling qualities rating from the Cooper-Harper Ratings (CHR) on the glideslope and touchdown errors; 3) the pilots' workload and psychological evaluation analysis from the NASA Task Load Index (TLX); and 4) the pilots' control workload based on the power spectral density (PSD) of the simulator control inputs. The first and last assessments are objective evaluations and the middle two are quasi-objective evaluations. This chapter describes the analyses and the results of the final tests. In all analyses that follow, the mean values and standard deviations are calculated across all 13 pilots. In addition, an analysis was completed with the pilots grouped together based on their experience. It is important to recognize that pilot performance measures alone are not a sufficient metric for determining total system performance. This is because pilots are very good at maintaining high performance levels under adverse circumstances. They just work harder. Therefore, metrics sensitive to pilot workload are necessary.

### **4.1. Analysis of Pilots' Performance**

In both the straight-in approach and the offset approach, the aircraft, started from an initial altitude of 1300 ft, descended gradually and landed on the designated runway.

The angle that the aircraft deviates from the ideal glideslope is called the glideslope error. The distances in the  $x$  and  $y$  directions between the actual touchdown point and the desired touch down point are referred to as touchdown errors. The glideslope error and the touchdown errors are two objective measures directly reflecting the pilots' maneuvering performance.



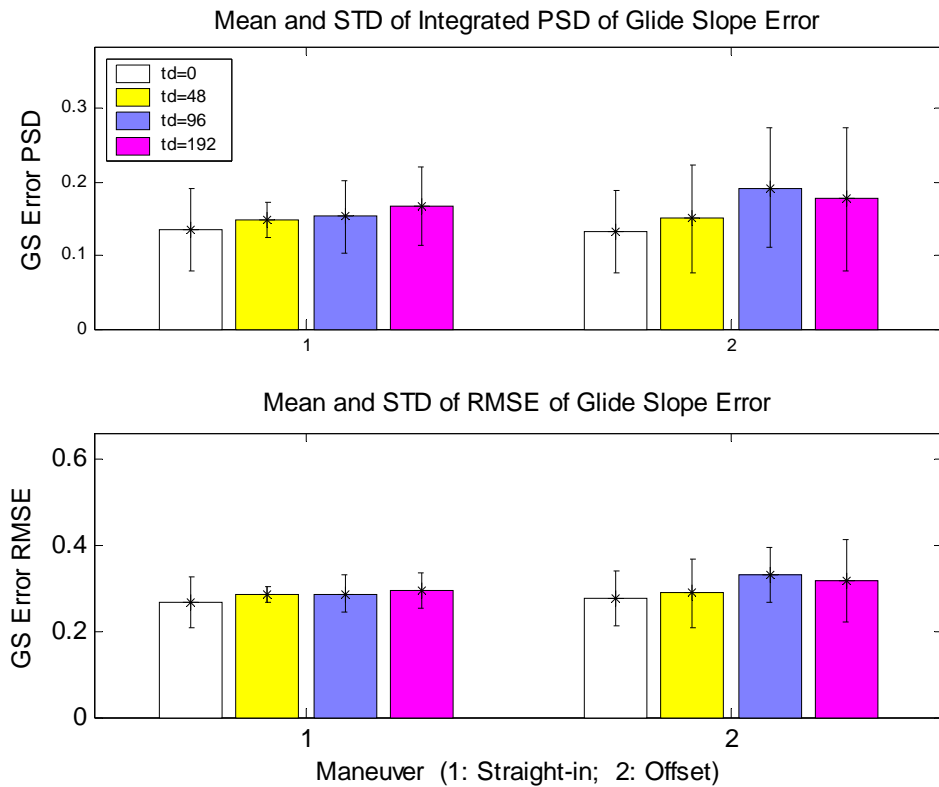
**Fig. 4.1. Example of the aircraft altitude and glideslope error**

#### ***4.1.1. Glideslope Error***

Fig. 4.1 shows an example of the aircraft altitude and the corresponding glideslope error. As the aircraft approaches the runway, the glideslope error usually increases in magnitude, and it saturates at  $\pm 0.7^\circ$  as the aircraft actually touches the ground. Because saturation of the glideslope error indicates that the aircraft touched down, the saturated value was removed from the data, and the analysis was terminated at this point. The integrated PSD and root mean squared error (RMSE) were also calculated, but

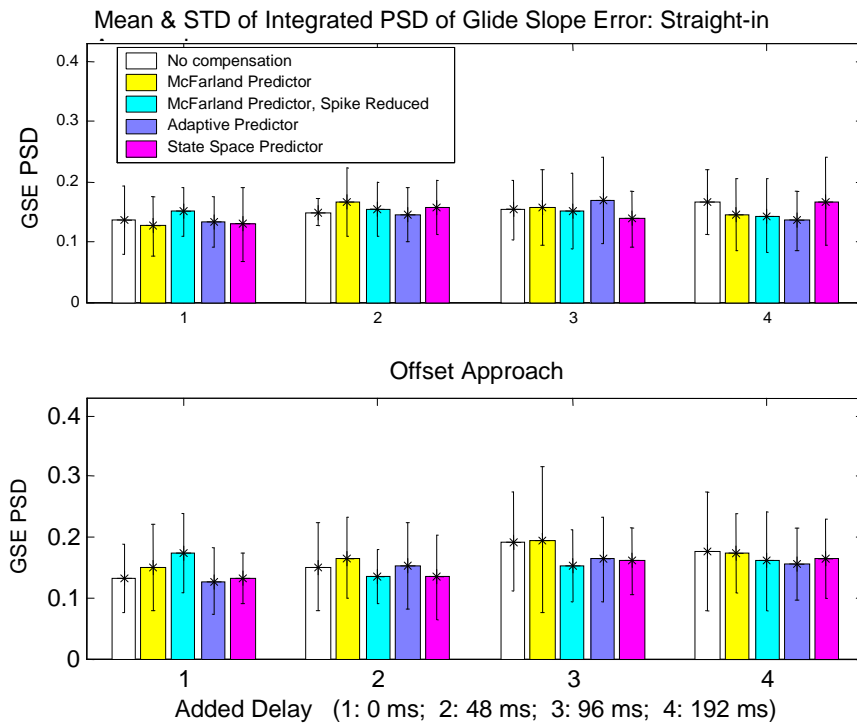
because the PSD showed more distinction between cases, the analysis was based primarily on the PSD. The integrated PSD is the integration of the PSD over the frequency range in which it is distributed.

Fig. 4.2 shows the impact of the time delay added to the simulator in terms of the mean value and standard deviation of the integrated PSD of the glideslope error. In general, the glideslope error increases with an increase in the time delay, except that in the offset approach, the glideslope error for 192 ms of delay was smaller than that of the 96 ms delay case. But, because of the relatively large deviations, the difference was insignificant. With the exception of the zero added delay case, the glideslope errors for the offset approach maneuver were slightly larger than those for the straight-in approach, for each time delay case.



**Fig. 4.2. Mean and standard deviation of the glideslope errors as a function of delay**

Fig. 4.3 illustrates the integrated PSD of the glideslope errors for the different compensation algorithms, namely, no compensation, McFarland compensation, McFarland compensation with spike reduction, the adaptive compensation and the state space compensation. Note that for all cases, the term “no compensation” means no compensation whatsoever. However, when any of the compensators was in use, and the case calls for zero added delay, the compensators were set up to compensate for the 90 ms of baseline delay. The adaptive filter and state space filter both showed a slight reduction in the glideslope error in compensating the baseline delay for either approach. While the McFarland compensator resulted in a larger decrease in the glideslope error in the straight-in approach than the two novel compensators, it increased the error in the offset approach. Finally, in the zero added delay cases, the spike reduction algorithm even increased the glideslope error over that of the basic McFarland filter.

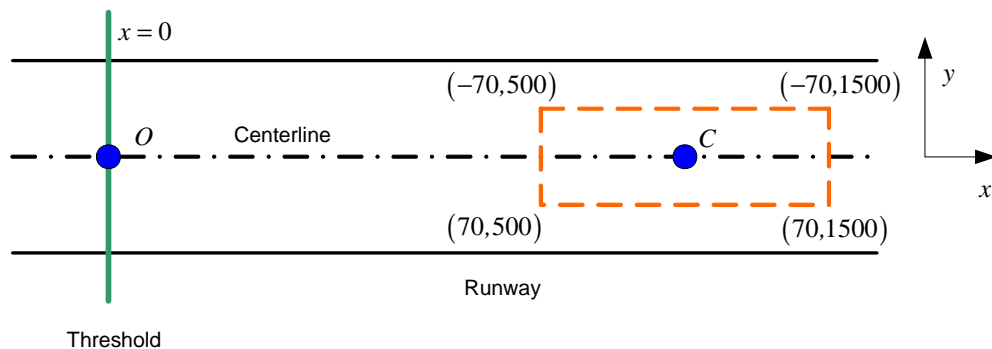


**Fig. 4.3. Mean and STD of the integrated PSD of the GSE with compensation**

For the added delay cases, the state space filter revealed a glideslope error reduction except for the 192 ms added delay in the straight-in approach. In general, the adaptive compensator showed decreased error, except for the 96 ms delay case in the straight-in approach and 48 ms delay case in the offset approach. The McFarland compensator revealed a reduction in the glideslope error only for the 192 ms additional delay cases. Aside from the McFarland filter, the compensation was obvious when using the remaining three filters for 96 ms added delay in the offset approach. Smaller errors were observed with the spike reduction algorithm than with the plain McFarland compensator for all added delay cases.

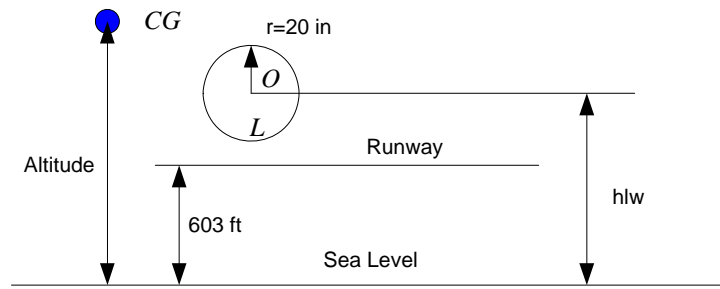
#### 4.1.2. Touchdown Error

The subjects were instructed to touchdown within a “touchdown box” as illustrated in Fig. 4.4, which is 1000ft  $\times$  140ft, centered at (1000,0). If the touchdown error was considered zero when touching down anywhere inside the box, there would be many runs without any touchdown error. Therefore, the  $x$  touchdown error was defined as the absolute value of the difference between the  $x$  coordinate of the aircraft CG at the touchdown point and the center  $C$ , and the  $y$  touchdown error was the distance between the  $y$  position of the CG at the touchdown point and the runway centerline.



**Fig. 4.4. Illustration of the touchdown box**

However, determining the exact touchdown point became a tricky problem because the landing gear forces were not recorded. One way to solve this problem is to use the available aircraft vertical force that includes spikes that occurred when the landing gear touched the runway. The beginning of the first spike was used to signal the touch down. While this method worked well for almost all straight-in runs and most of the offset runs, it proved to be defective for many offset runs because the high accelerations resulting from the vertical forces caused by the turbulence, masked the spikes caused by the touchdown. Fortunately, there is an alternative way to decide the touchdown point, which is illustrated in Fig. 4.5.

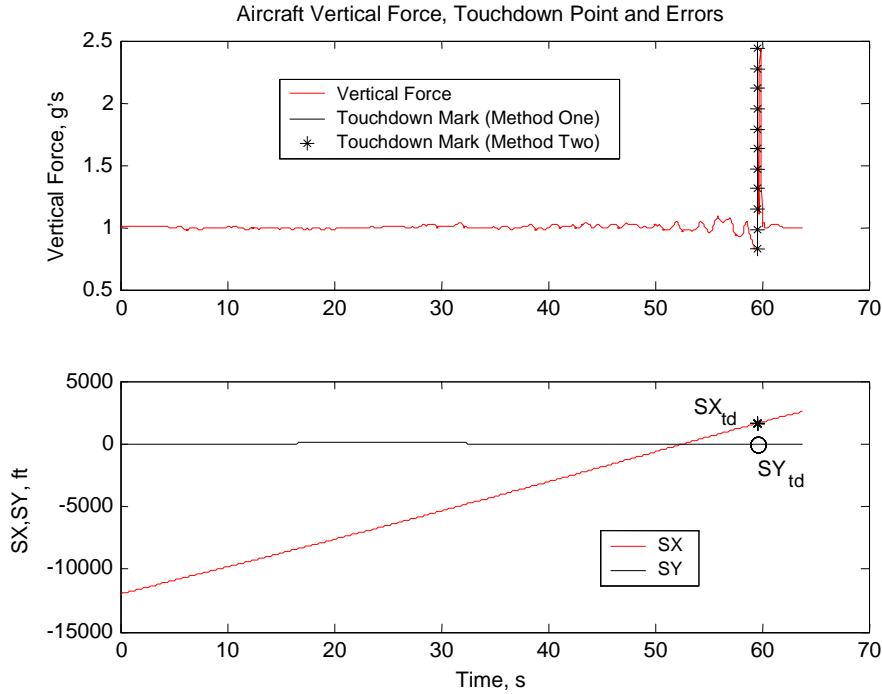


**Fig. 4.5. Illustration of calculating the height of the landing gear**

The height of the centers of the landing gear are given by

$$hlw = h - r_2 \cos \phi \cos \theta - r_1 \sin \phi + r_0 \sin \theta \quad (4.1)$$

where  $h$ ,  $\phi$  and  $\theta$  are the aircraft altitude, roll angle and pitch angle (all available), and  $r_0 = 7\text{ft}$ ,  $r_1 = \pm 12\text{ft}$  (positive for the right gear and negative for the left gear) and  $r_2 = 13.7\text{ft}$ . Then, the height of the lowermost point on a landing gear is  $hlw - r$ . Once this is less than 603ft, the first touchdown occurs. The touchdown point was determined with these two methods. An example of determining the touchdown errors is given in Fig. 4.6, where  $SX$  and  $SY$  are the aircraft positional coordinates in the Earth frame, relative to the runway. It shows the two methods yield identical results.



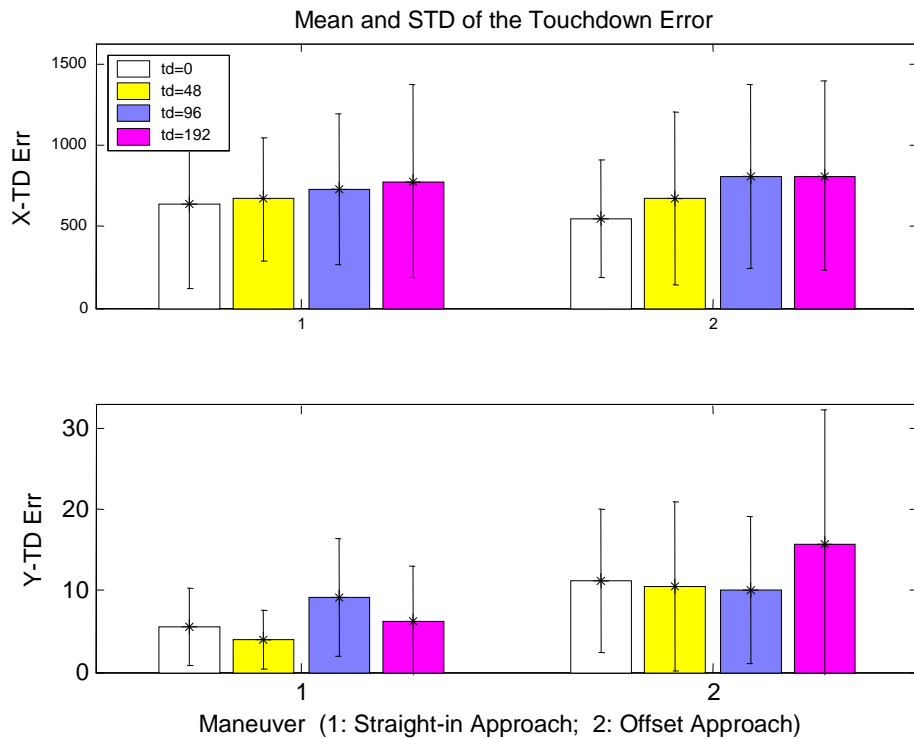
**Fig. 4.6 Aircraft force, touchdown point and touchdown errors**

Fig. 4.7 illustrates the change in the touchdown error in both X and Y directions induced by the addition of time delay. The X-touchdown error showed consistent increase with increasing artificial time delay, though in the offset approach, the X-touchdown errors for the 96 ms and 192 ms delay cases were about the same (the latter was slightly higher). In general, the standard deviation also increases with the amount of inserted delay. With the exception of the zero added delay case, the offset approach reveals a slightly larger X-touchdown error than the straight-in approach.

The Y-touchdown error, however, did not show consistent change with the changing delay. For the straight-in maneuver, the 48 ms delay case resulted in a considerable decrease in the Y-touchdown error compared to the zero delay case, and the 192 ms delay case resulted in a much smaller error than the 96 ms delay case. For the offset approach, adding 48 ms or 96 ms delay resulted in smaller Y-touchdown error as



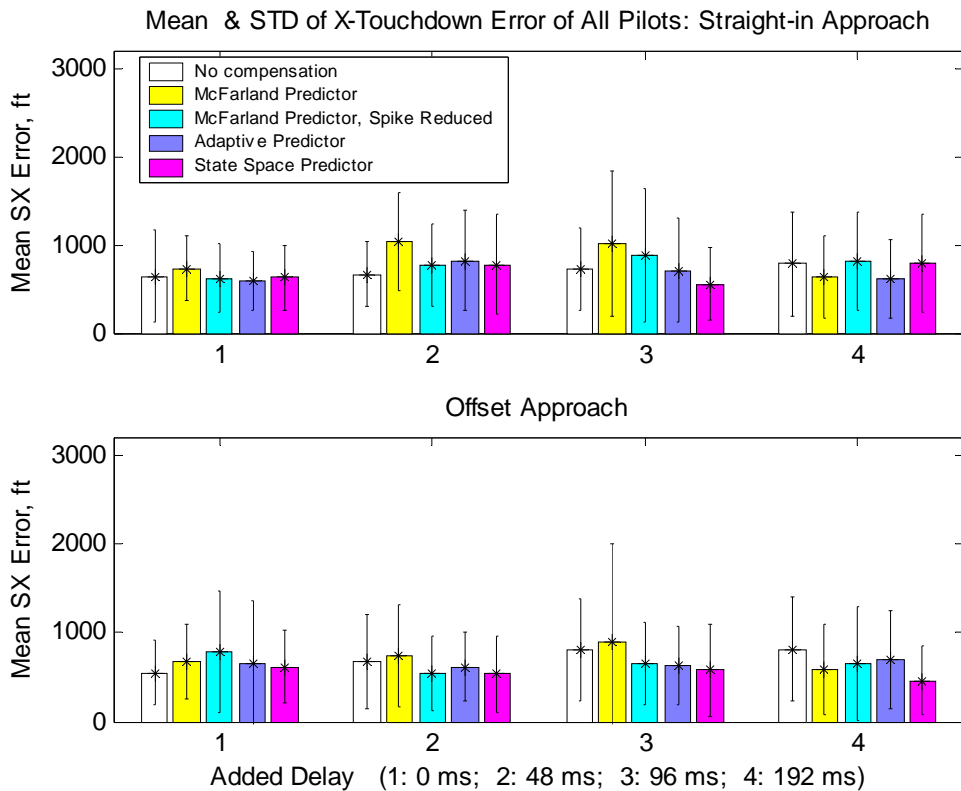
compared to the zero delay case. The inconsistency in the Y-touchdown error might be associated with the fact that the subjects were instructed to pay more attention to the X-touchdown accuracy. Therefore, it is reasonable to weight the X-touchdown error more than the Y-touchdown error in the remaining analyses.



**Fig. 4.7. Mean and standard deviation of the touchdown errors changing with delay**

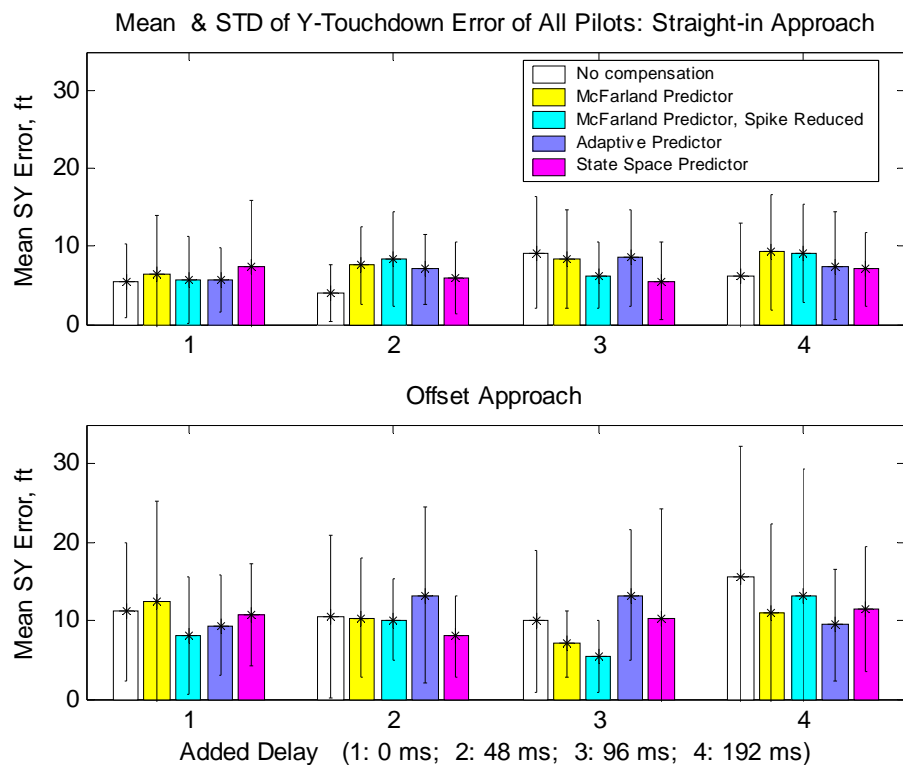
The differences imposed by the four different compensation techniques on the X-touchdown error are given in Fig. 4.8. For the zero added delay straight-in case, all but the McFarland compensator revealed a reduction in the mean X-touchdown error, with the smallest reduction occurring with the state space filter. For the offset approach, all compensators yielded increased X-touchdown error, which might be attributed to the relatively small mean X-touchdown error with the zero added delay case (it was even smaller than in the straight-in approach). The increased X-touchdown error for the 48 ms

delay in the straight-in approach was unexpected. As a comparison, all but the McFarland compensators showed a reduction in the X-touchdown error for the 48 ms delay in offset maneuver. With the exception of the 192 ms delay in the straight-in approach, the two novel compensators revealed a noticeable decrease in the X-touchdown error, with the state space filter making a significant difference in compensating 192 ms delay in the offset approach. The state space predictor showed a significantly smaller X-touchdown error than the McFarland predictor for 96 ms delay during the straight-in approach. The McFarland filter only reduced the X-touchdown error in the 192 ms delay case. For all but the 192 ms delay straight-in case and the zero added delay offset-approach case, the spike reduction algorithm produced decreased X-touchdown error. Finally, notice that the offset approach showed a smaller X-touchdown error than the straight-in approach.



**Fig. 4.8. Mean and STD of the X-touchdown errors changing with the compensation**

Fig. 4.9 shows the counterpart of Fig. 4.8 for the Y-touchdown error. For the straight-in approach, all compensations resulted in a reduction of the Y-touchdown error only for the 96 ms added delay case. In other words, compensation with all predictors increased the Y-touchdown error for the zero, 48 ms and 192 ms added delay cases, for the straight-in approach. The results improved for the offset approach. With the exception of a few cases (the McFarland compensation for the zero added delay, and the adaptive filter for the 48 ms and 96 ms), all compensations revealed a reduction of the Y-touchdown error. And, unlike the X-touchdown errors, the Y-touchdown errors during the offset approaches were substantially greater than those of the straight-in approach.



**Fig. 4.9. Mean and STD of the Y-touchdown errors changing with the compensation**

Comparing Fig. 4.2 with Fig. 4.7, and Fig. 4.3 with Fig. 4.8 and Fig. 4.9 shows that the added time delay and all the compensators produced a very similar influence on

the glideslope error and X-touchdown error. Two noticeable differences are: 1) the mean X-touchdown error does not increase with the maneuver complexity (i.e., it should be larger for the offset approach) as does the glideslope error; and 2) the compensation for the 48 ms delay case with the novel predictors did not result in an improvement of the pilots' performance in terms of the X-touchdown error, as it did in terms of the glideslope error. As was previously stated, this might be attributed to the relatively small X-touchdown error for the zero added delay case. The Y-touchdown error displayed more randomness as compared to the X-touchdown error. Therefore, the glideslope error showed more positive results for the effects of time delay and compensation than the touchdown error as a whole.

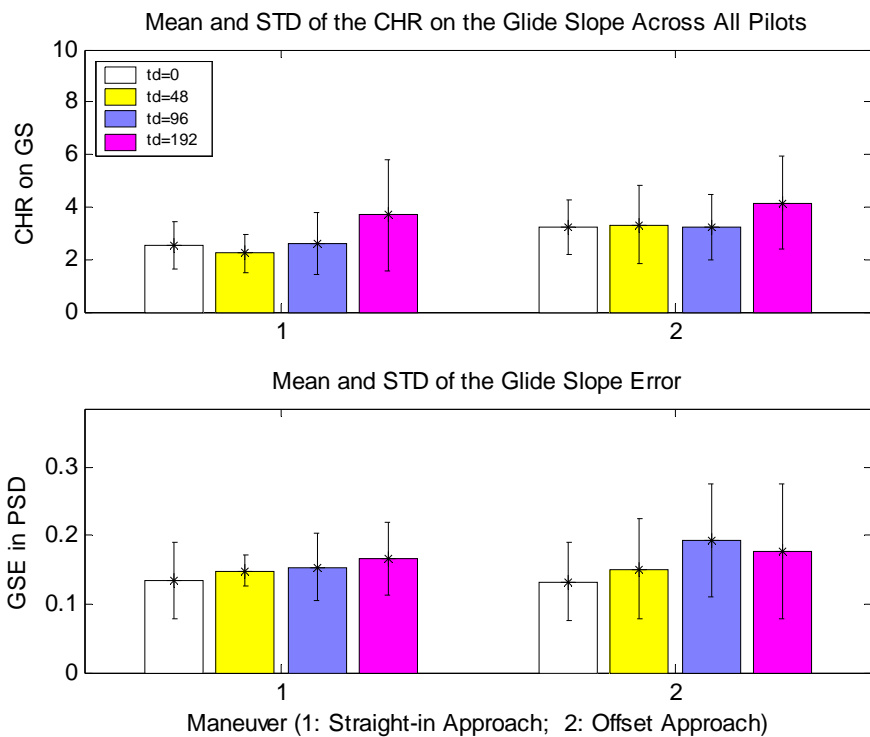
#### **4.2. Analysis of Pilots' Handling Quality Rating (CHR)**

Cooper Harper Rating (CHR) data were recorded to assess the pilots' handling quality. Unlike the CHR collection in the preliminary tests, two sets of CHR were collected in the final piloted tests: one CHR on the pilot's ability to maintain the glideslope and the other on his ability to land in the touchdown box. In other words, the pilots were instructed to evaluate the handling qualities based upon the glideslope and the touchdown results separately, after each simulation run. The advantage of this approach is that it becomes possible to compare the two CHRs with the two types of errors individually. The presentation of the CHR analyses in this section is given in conjunction with the glideslope error and touchdown error (in the X-direction) for convenience.

##### ***4.2.1. CHR of Maintaining the Glideslope***

Fig. 4.10 shows how the mean values and standard deviations of the glideslope error (GSE) and the CHR change with the addition of time delay. In general, longer

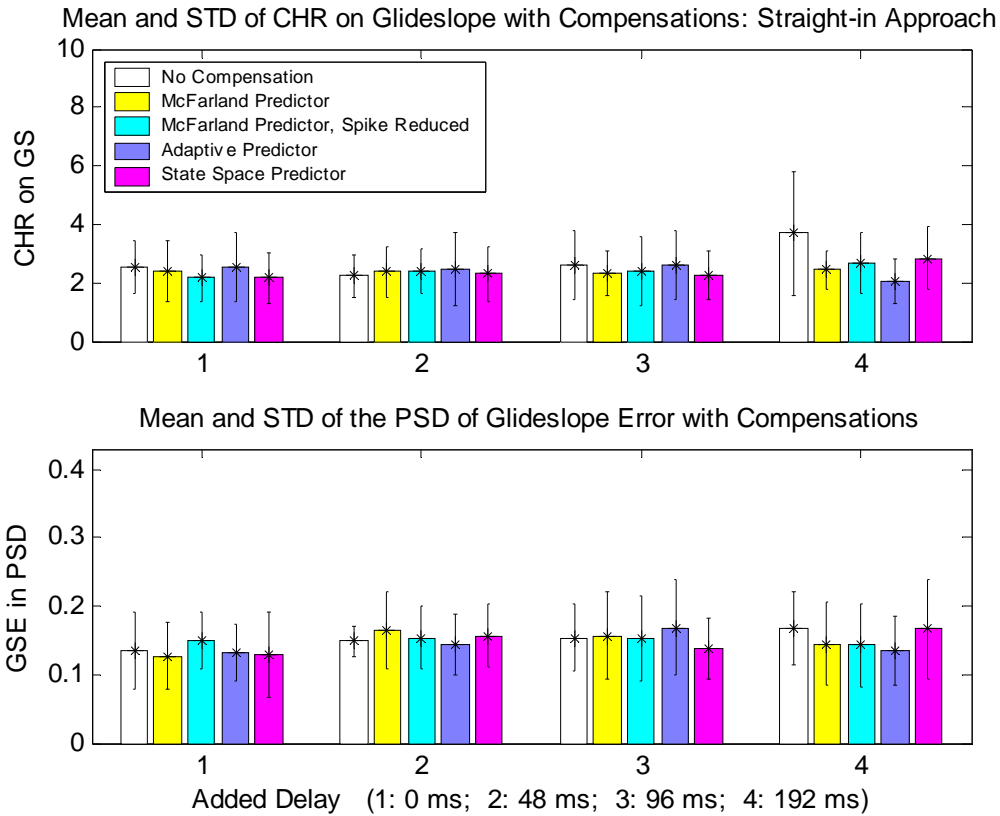
added time delay corresponds with a higher mean CHR (i.e., poorer handling quality) except the CHR decreases from zero delay to 48 ms delay for the straight-in approach, and from 48 ms delay to 96 ms delay for the offset approach. Because the time delay is expected to diminish the pilot’s assessment of the handling quality (higher CHR) and increase the GSE, a change in the opposite direction may be considered an exception. Analysis of the CHR revealed two exceptions, whereas analysis of the GSE only revealed one, and they were in different situations. This indicates some discrepancy between CHR and the GSE. The GSE yielded “better” results than the CHR, which is not surprising because the GSE is an objective metric and the CHR is a quasi-objective metric.



**Fig. 4.10. Mean & STD of CHR and GSE changing with the time delay**

Pilot #6 gave a CHR of 9 for the 192 ms added delay, straight in approach, with the comment “Roll PIO (pilot induced oscillation) during most of the approach; (the

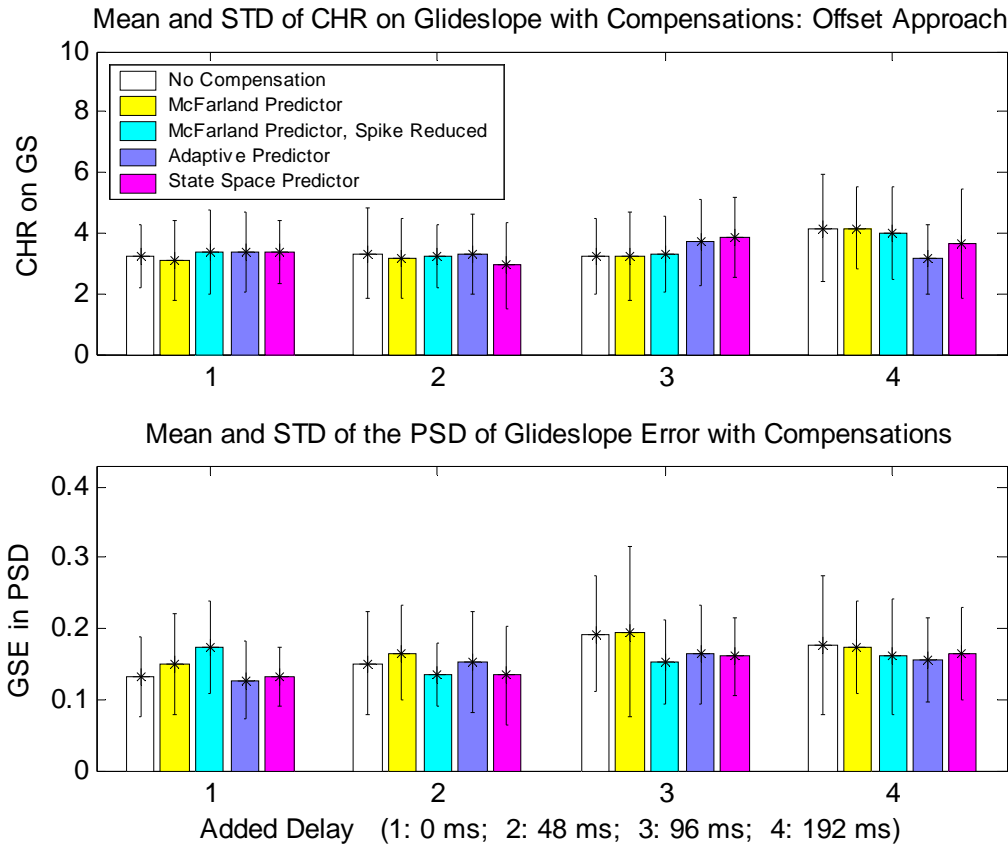
motion base kicked off about the time the aircraft crossed the threshold”. Refer to Fig B.3.10 in Appendix B.



**Fig. 4.11. Mean and STD of CHR & GSE with compensations: straight-in approach**

Fig. 4.11 shows the mean values and standard deviations of the GSE and the CHR for different compensations in the straight-in approach. For the 0 ms, 96 ms and 192 ms delay cases, the CHR revealed noticeably better handling qualities with all compensators, except the adaptive predictor, which on average, made no difference when the added delay was 0 ms or 96 ms. For the 48 ms delay case, all compensation algorithms produced an increased CHR. The spike reduction algorithm made an inconsistent difference in the McFarland compensator. The discrepancy between the CHR and GSE was not substantial. The largest discrepancy between the CHR and the GSE was for the

zero delay case, when using the McFarland compensation with the spike reduction. This case had the lowest CHR but highest GSE among all compensators.



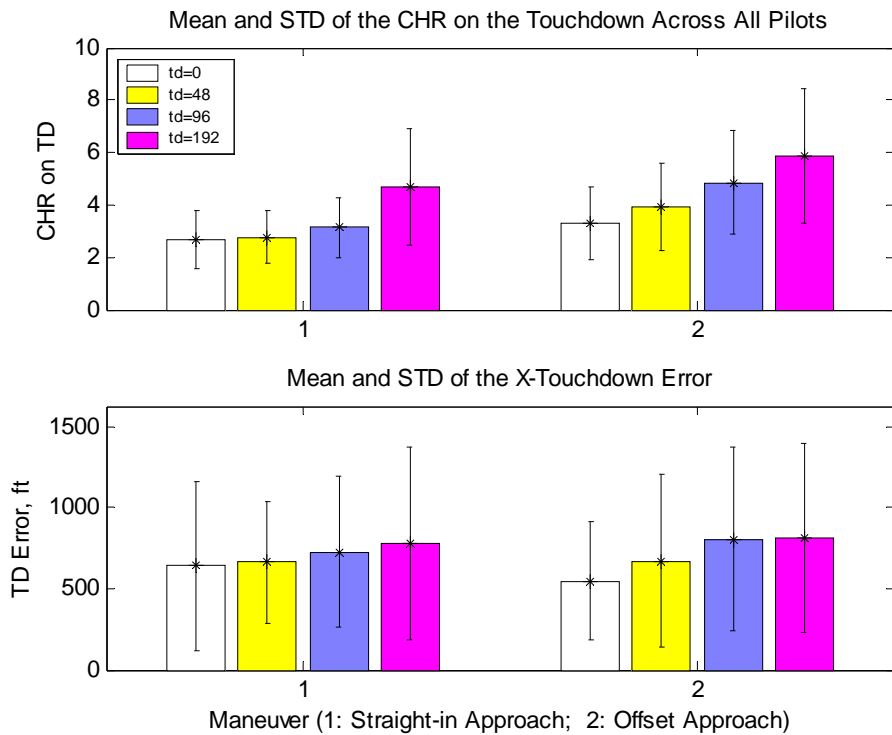
**Fig. 4.12. Mean and STD of CHR & GSE with compensations: offset approach**

The counterpart of Fig.4.11 for the offset approach is given in Fig.4.12. In terms of the CHR, obvious, desirable compensation results were observed only for the McFarland predictor for the zero delay case, the state space predictor for the 48 ms and 192 ms delay cases, and the adaptive predictor for the 192 ms delay case. Notice that the McFarland predictor with spike reduction, the adaptive predictor and the state space filter produced an even higher CHR compared to the uncompensated cases for both zero and 96 ms artificial delay, and for these cases (except for McFarland with spike reduction in zero delay), the CHR shows a sharp discrepancy with the GSE. The objective evaluation

(GSE) clearly revealed more desirable compensation results than the quasi-objective evaluation.

#### 4.2.2. CHR on the Touchdown

Fig. 4.13 shows the change in the mean values and standard deviations of the touchdown error (TDE) and the CHR as a function of the added time delay. With the exception of the 48 ms added delay straight-in approach case, in which the CHR was approximately the same as the zero added delay case, all other cases showed a distinct increase in the CHR. The increase in the CHR in the 192 ms delay case was significant. Although the CHR revealed a more apparent increase as a function of the delay than the TDE, the difference between them was not substantial. The offset approach significantly degraded the handling quality compared to the straight-in approach.

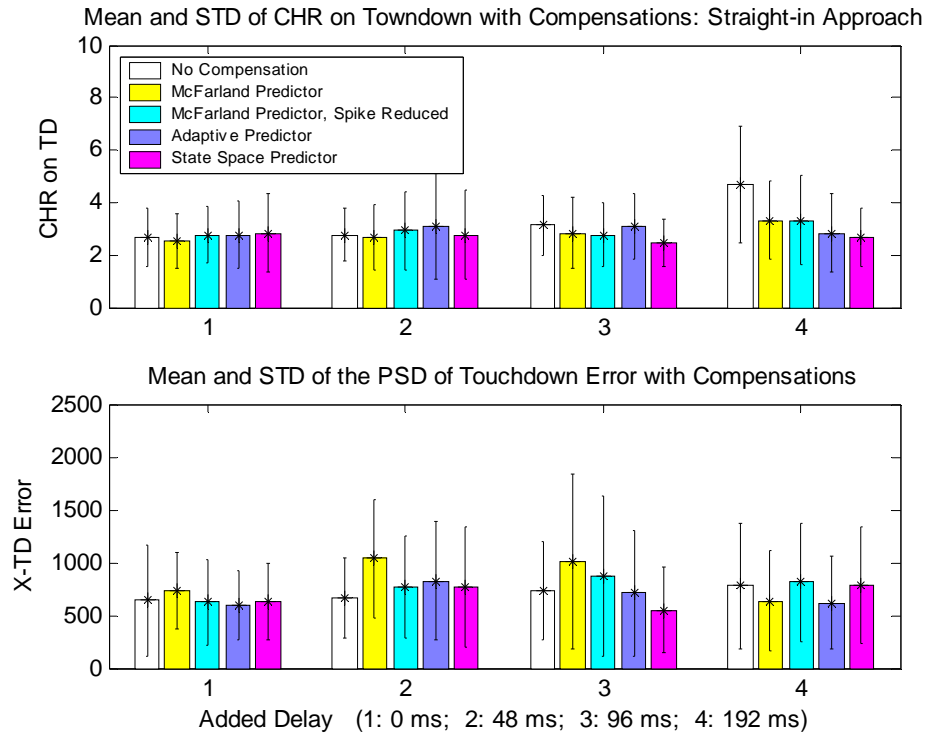


**Fig. 4.13. Mean & STD of CHR and TDE changing with the time delay**



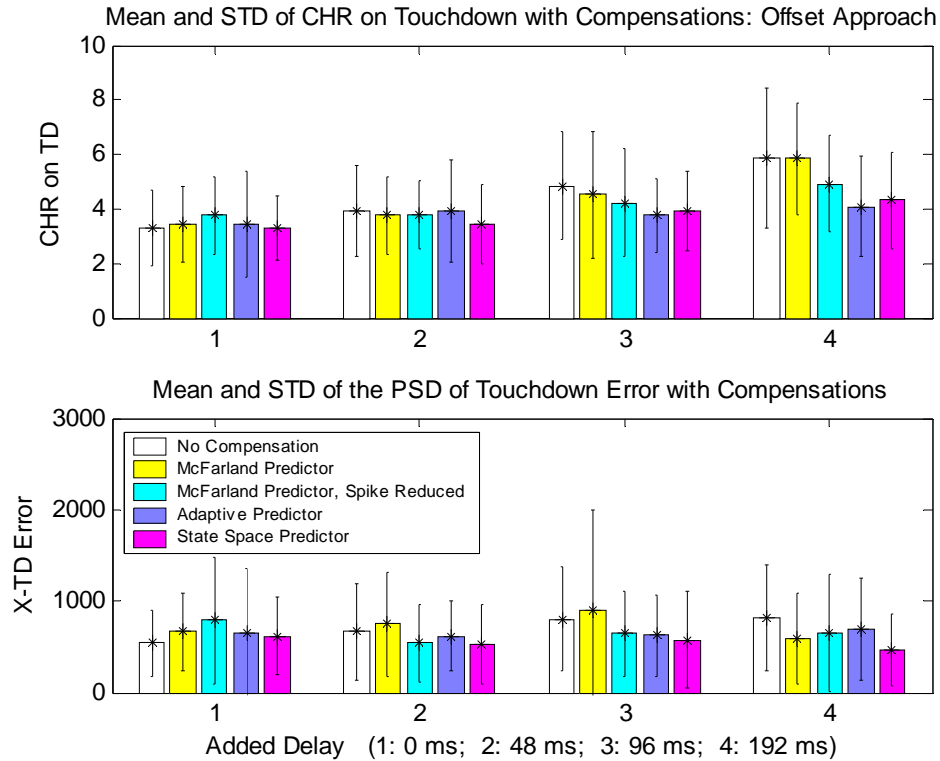
During this study, three full CHR (CHR=10, indicating the system is totally uncontrollable) occurred for long artificial delays: The first was from pilot #5 for the 192 ms delay case, and the second and third were from pilot #13 for the 96 ms and 192 ms delay cases. In all three cases, the pilot reported PIO, which significantly affected the touchdown.

A comparison of different compensation algorithms in terms of the TDE and the CHR for the straight-in approach is shown in Fig.4.14. Obvious CHR decreases by all compensations were found only for the 96 ms and 192 ms delay cases. The two novel predictors produced significantly decreased CHR in compensating 192 ms delay during the straight-in approach. On the other hand, with the exception of the McFarland compensator, which resulted in a decreased CHR for the zero and 48 ms delay cases, all the other predictors revealed an increased CHR. The most substantial discrepancy between the TDE and the CHR occurred with the McFarland compensation for an added delay shorter than 192 ms: When using the McFarland compensation, the CHR decreased but the TDE increased.



**Fig. 4.14. Mean and STD of CHR & TDE with compensations: straight-in approach**

Fig. 4.15 gives the counterpart of Fig. 4.14 for the offset approach. Almost all compensators improved the pilots' handling qualities for non-zero added delay cases in terms of the mean CHR. With the exception of the McFarland compensator with spike reduction, all the compensators made little difference in the mean CHR for the zero added delay case, although the state space predictor had considerably smaller standard deviation. With the exception of the McFarland predictor in the non-zero added delay cases, the CHR and TDE agreed with each other fairly well. The two novel predictors made a significant improvement over the McFarland compensator in terms of the CHR and the TDE for the 192 ms delay.



**Fig. 4.15. Mean and STD of CHR & TDE with compensations: offset approach**

There were two occurrences of a full CHR (worst handling quality) with compensation while the corresponding uncompensated runs had a CHR of 4. Both were from pilot #4 while using the McFarland compensation. The first occurred for the 96 ms delay and the second was for 192 ms delay. These indicate that serious handling problems may result when using the McFarland compensator for long time delay. No full CHRs were associated with the compensation algorithms designed by the author.

Finally, it is worth mentioning that the Cooper-Harper Ratings on the glideslope and touchdown did not show substantial difference.

### 4.3. Analysis of the TLX

The NASA Task Load Index (Hart and Staveland, 1988<sup>11</sup>) is a multi-dimensional quasi-objective metric to rate the overall workload of the operator in a man-machine

interaction. The TLX is a weighted average of six ratings of six subscales of workload<sup>12</sup>: mental demand, physical demand, temporal demand, performance, effort and frustration. The definitions of these six subscales are given in Appendix A (Table A.4). Each scale is presented as a line divided into 20 equal intervals (i.e., graduations) anchored by bipolar descriptors (e.g., High/Low). Obtaining a NASA TLX is a three-step procedure: rating, weighting and averaging. First, the pilot gives a rating by marking each scale at the desired location either during a task, after a task segment, or following an entire task. Then, the weighting is achieved by making choices for 15 pairs of comparisons resulting from the combination of the six factors. The subject has to select the member of each pair that contributes more to the workload of that task. Finally, sum the weighted ratings and divide the sum by 15 to get the overall weighted TLX.

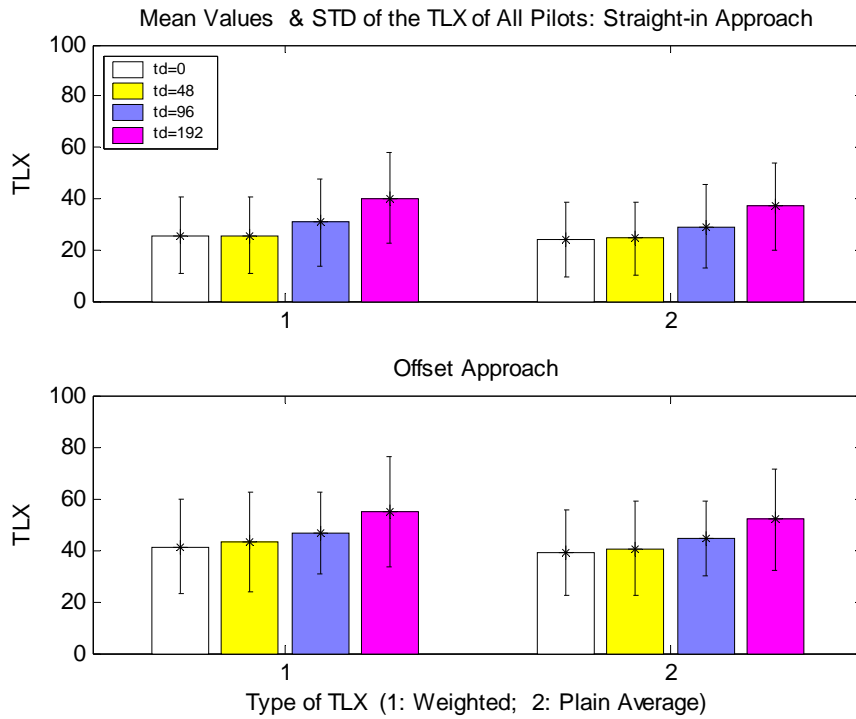
For the final piloted tests, in addition to the weighted TLX, another type of TLX was also calculated by simply averaging the six ratings, giving an even weight to each subscale, and this method will be referred to as an Evenly Averaged TLX in this report.

High TLX (and high EA TLX) indicates a high pilot workload.

#### ***4.3.1. Analysis on the Overall Weighted TLX***

Fig. 4.16 shows the effects of added time delay on the mean value and standard deviation (across the 13 pilots) of the overall weighted TLX and the Evenly Averaged TLX. For the straight-in approach, from zero delay to 48 ms delay, the mean overall weighted TLX remained unchanged, the evenly averaged TLX showed only a very slight increase. However, both types of TLX increased for the 96 ms and 192 ms delay cases. The standard deviation of both types of TLX did not reveal any considerable difference among the four cases of artificial delay. The changes in the overall weighted TLX, and

the Evenly Averaged TLX for the offset approach were similar to those of the straight-in approach. The offset approach substantially increased the TLX compared to the straight-in approach.



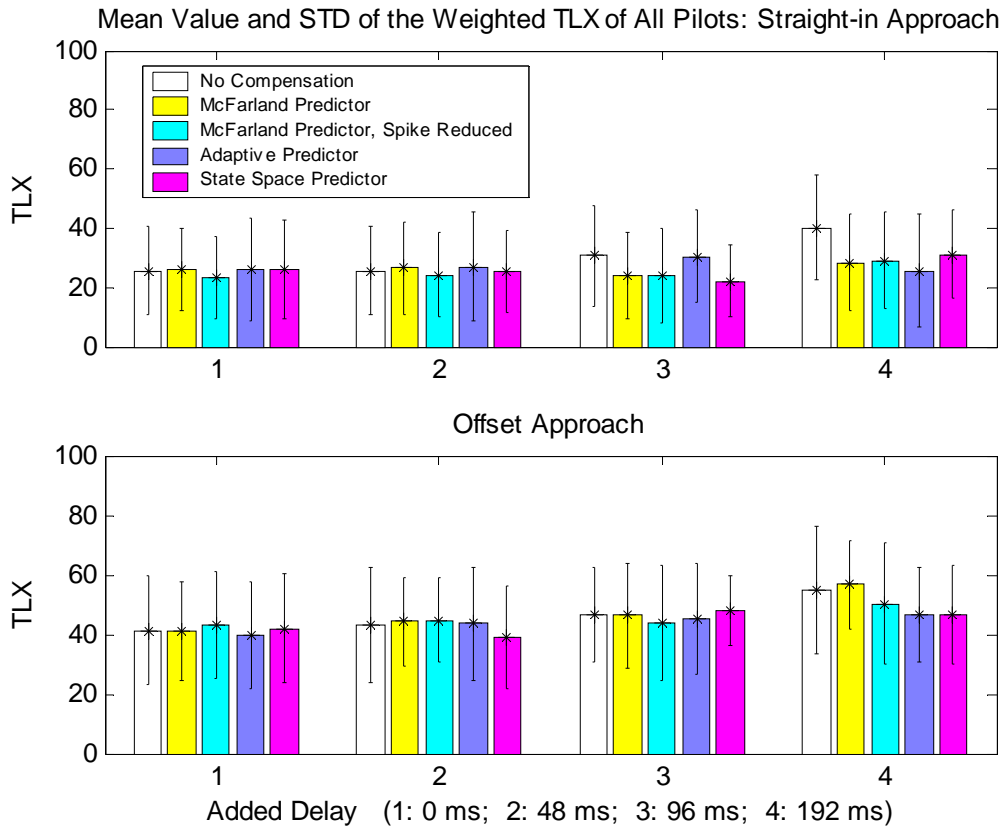
**Fig. 4.16. Mean value and STD of the TLX as a function of the time delay**

The counterintuitive increase in the two types of TLX for the 48 ms added delay case was similar to that of the other quasi-objective metric, the CHR, which also increased with the same artificial delay. This shows that a small delay of up to 48 ms made only a slight difference in the quasi-objective metrics. Because adding 48 ms delay did not cause a noticeable increase in TLX, adding compensation for the 48 ms added delay was not expected to cause a noticeable decrease in TLX, and the TLX of the compensated system was not expected to be lower than the TLX of the undelayed system. This will be proved in the following analysis of TLX for the compensations.

In this study the Evenly Averaged TLX did not show significant differences between cases when compared to the overall weighted TLX. Therefore, results based upon the evenly averaged TLX will not be presented further, and the overall weighted TLX will be referred to as TLX.

Fig. 4.17 shows the difference in the mean value and standard deviation of the TLX from the four types of compensators. For the straight-in approach, all compensators caused an obvious decrease in the TLX for 96 ms or more added delay cases. The state space compensator caused a slight decrease in the TLX for 48 ms added delay case, but a slight increase for the zero added delay case. Both the McFarland predictor and the adaptive predictor revealed increased TLX for zero and 48 ms added delay cases, while the reduced-spike McFarland predictor showed decreased TLX for these two delay cases.

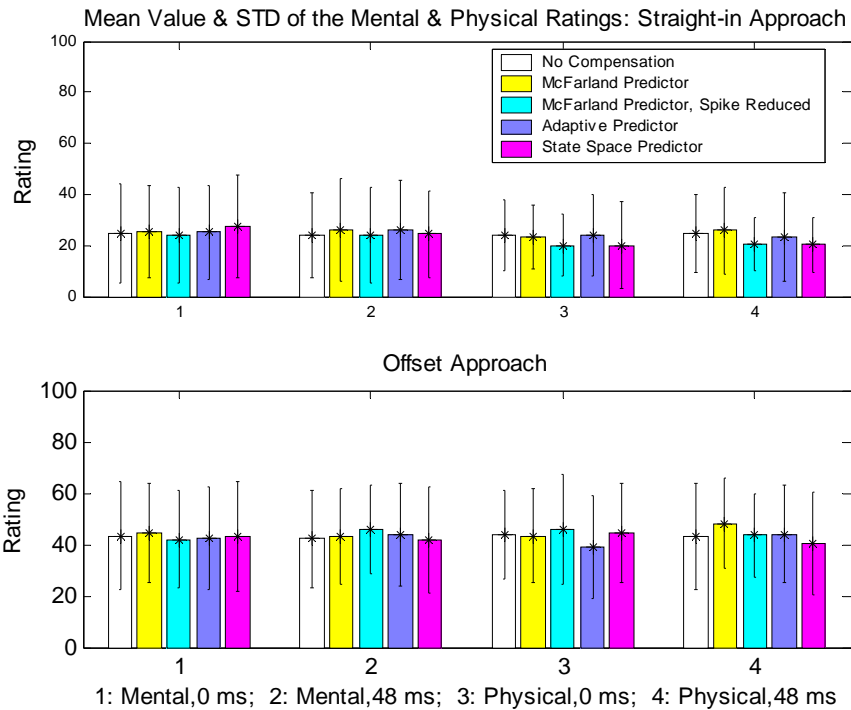
For the offset approach, all predictors, except the McFarland compensator and the state space compensator, showed decreased TLX when compensating for 96 ms or more added delay. In the 48 ms delay case, only the state space compensator showed a decreased TLX, with all the other compensators causing the TLX to increase. Only the adaptive predictor showed a noticeable reduction of the TLX in the zero added delay case. Inconsistent differences were observed when using the spike reduction algorithm compared to the (plain) McFarland compensator.



**Fig. 4.17. Mean and STD of the TLX changing with compensations**

### 4.3.2. Analysis on the Six Factors of the TLX

As was previously stated, the TLX is an overall weighted average of the six ratings on the six subscales. But, because the influence of the compensation algorithms on the overall weighted TLX is unclear, it will be necessary to investigate the contribution of each individual component. Because the compensation yielded an obviously decreased TLX for the 96 ms and longer added delay cases, the examination of the individual items in the following section will only be concerned with the zero and 48 ms added delay cases. The results are illustrated in Fig. 4.18 (for the mental and physical demands), Fig. 4.19 (for the temporal demand and performance) and Fig. 4.20 (for the effort and frustration).



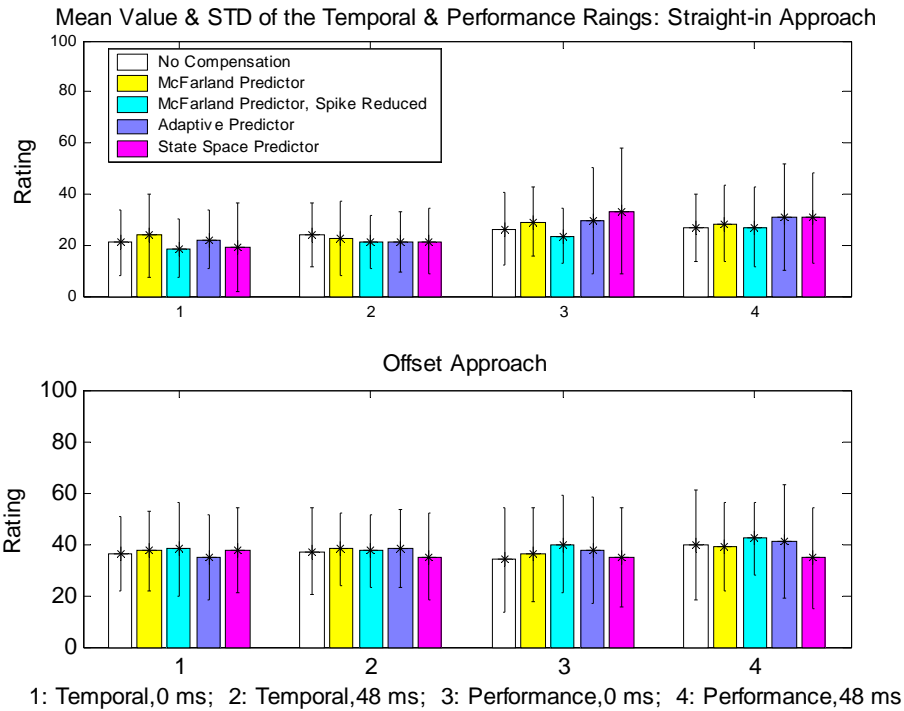
**Fig. 4.18. Mean and STD of mental and physical demands with compensations**

Fig. 4.18 shows that, in general, compensation tends to increase the mental demand, and decrease the physical demand, during the straight-in approach. But for the offset approach, the contradiction between the mental and physical demands vanished, and the change in the demands was neither significant nor consistent. The two novel compensators revealed more occurrences of decreased demand than the other two, though the decrease was very slight.

As illustrated in Fig. 4.19, a similar contradiction exists between the temporal demand and the performance for the straight-in approach. Compensation (except for the McFarland predictor for zero added delay case) tended to decrease the average temporal demand, while it tended to increase the average performance rating. The two novel



compensators revealed a slightly larger decrease in the temporal demand. For the offset approach, the compensation appeared to increase the two ratings slightly.

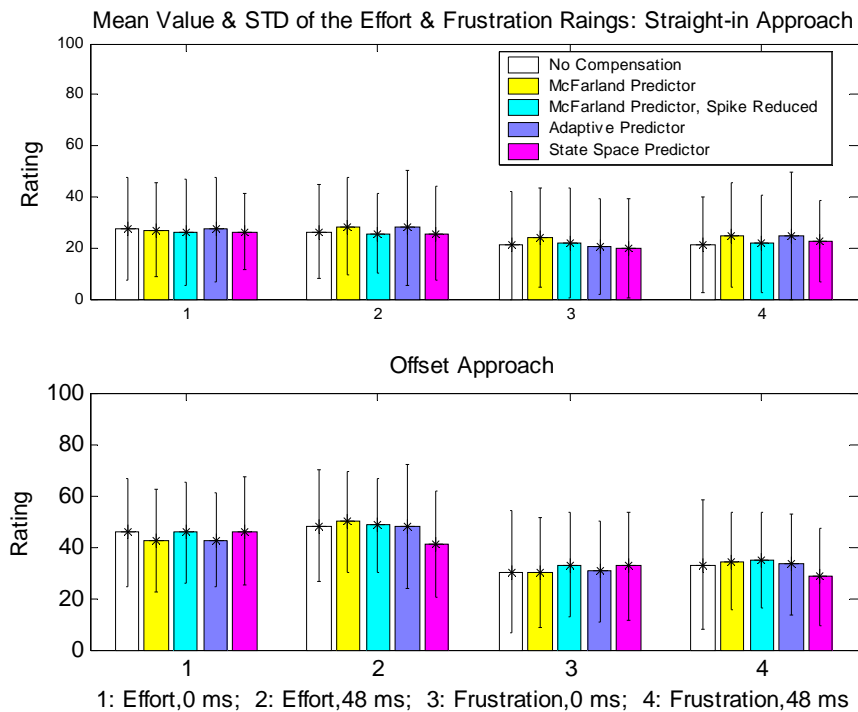


**Fig. 4.19. Mean and STD of the temporal demand & performance**

The compensators resulted in inconsistent changes to the effort and frustration ratings, as can be seen in Fig. 4.20. The change in the two ratings was not significant, except for the state space predictor, which substantially reduced the effort rating for the zero added delay straight-in approach case and for the 48 ms added delay offset approach case. The state space predictor also reduced the frustration rating for the 48 ms added delay offset approach case, and increased the frustration rating for the zero added delay straight-in approach case.

In summary, the compensation tended to reduce the ratings of the physical and temporal demands, increase the ratings of the mental demand and performance, and produced inconsistent changes in the ratings of the effort and frustration for the zero and

48 ms added delay cases. Because the TLX is a weighted average of these six ratings, the overall change in the final TLX caused by the compensation was not obvious for the above conditions. For the offset approach with an added delay of 48 ms or less, the compensation produced inconsistent changes in almost every individual rating. There were more occurrences of reduced NASA TLX when using the two novel compensators.



**Fig. 4.20. Mean & STD of the effort & frustration with compensations**

#### 4.4. Analysis of the PSD of Pilot Control

The power spectral density is a frequency domain analysis tool. The time history alone does not necessarily display differences among the various simulator conditions. Therefore, in order to study the effects of time delay and delay compensation on pilot control of a simulator, frequency domain techniques were employed. The different conditions may appear similar because of the existence of noise. The results of

multivariate analysis of variance of control force and displacements indicate that clear differences exist in the variances of the various control parameters, but not the nature of what those differences might be. On the other hand, the time delay and compensation impose greater impact on the frequencies, such as the peak positions of the primary components, and the total PSD (energy) over a certain frequency range.

#### **4.4.1. About the PSD**

The PSD of a discrete process is the Discrete time Fourier Transform (DFT) of its autocorrelation sequence<sup>13</sup> (Eq. (4.2)). 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 the signal as a function of the frequency<sup>14</sup>. Therefore, the total PSD, or the integral of the PSD over a certain range of frequencies of the control variable reveals the energy the human operator exerts on the control, thereby providing some insight into the pilot's control performance. Usually higher PSD means poorer handling qualities and therefore higher pilot workload. One drawback of the PSD analysis is the loss of phase information.

$$P_x(e^{-j\omega}) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \{x[n]x[n+m]\}e^{-j\omega m}, (m = n_2 - n_1) \quad (4.2)$$

There are over 10 different methods to estimate the PSD of a signal. The two used for PSD calculations in this paper are the direct method and the indirect method. The direct method, called the periodogram, is equivalent to the square of the magnitude of the DFT. The periodogram shows all details of the PSD. In contrast, the indirect method, or the smoothed periodogram, is the DFT of the autocorrelation. To use this method the autocorrelation must be calculated first. The smoothed periodogram suppresses some random details of the PSD so as to show the macroscopic characteristics of it. For smooth signals, the periodogram will be used for examining the peak details, while for signals

with more noise, such as the control inputs, the smoothed periodogram will be used to eliminate the artifacts of noise. The mathematical expressions of the direct and indirect methods of calculating the PSD are given by

$$\hat{K}_x^f(\theta) = \frac{1}{N} \left| \sum_{i=0}^{N-1} x[i] w[i] e^{-j\theta i} \right|^2 \quad (4.3)$$

$$\hat{K}_x^f(\theta) = \sum_{m=-M}^M \hat{k}_x[m] w[m] e^{-j\theta m}, (M \leq N-1) \quad (4.4)$$

respectively, with

$$\hat{k}_x[m] = \frac{1}{N} \sum_{i=0}^{N-1-|m|} x[i] x[i+|m|], (|m| \leq N-1) \quad (4.5)$$

In these equations,  $x$  is the signal,  $w$  is the window used to calculate the PSD, and  $N$  is the number of data points of signal  $x$ .

From the definition in Eq. (4.2), calculating the PSD of a random signal requires an infinite number of data points. For a signal of finite length, its PSD can only be estimated. Therefore, an estimate of the PSD of a finite-length signal can be obtained by truncating the signal, or by multiplying it by a rectangular window of the same length. Because the frequency characteristics of a rectangular window show a wide main lobe and high side lobes, the PSD estimation with a rectangular window tends to have relatively high power leakage to the adjacent frequencies<sup>15</sup>. Thus, other types of windows have been developed. This is why the two estimation algorithms given in Eqs. (4.3) and (4.4) involve the window function. No matter which window function is used, power leakage always exists. Therefore, applying the window in calculating the PSD is also a smoothing process due to the power leakage. The smoothed algorithm given in Eq. (4.4) actually double smooths. After investigation, the Hamming window was adopted for the

calculation of PSD for this study, because it yields a relatively good balance between suppressing the noise and showing the details of the PSD.

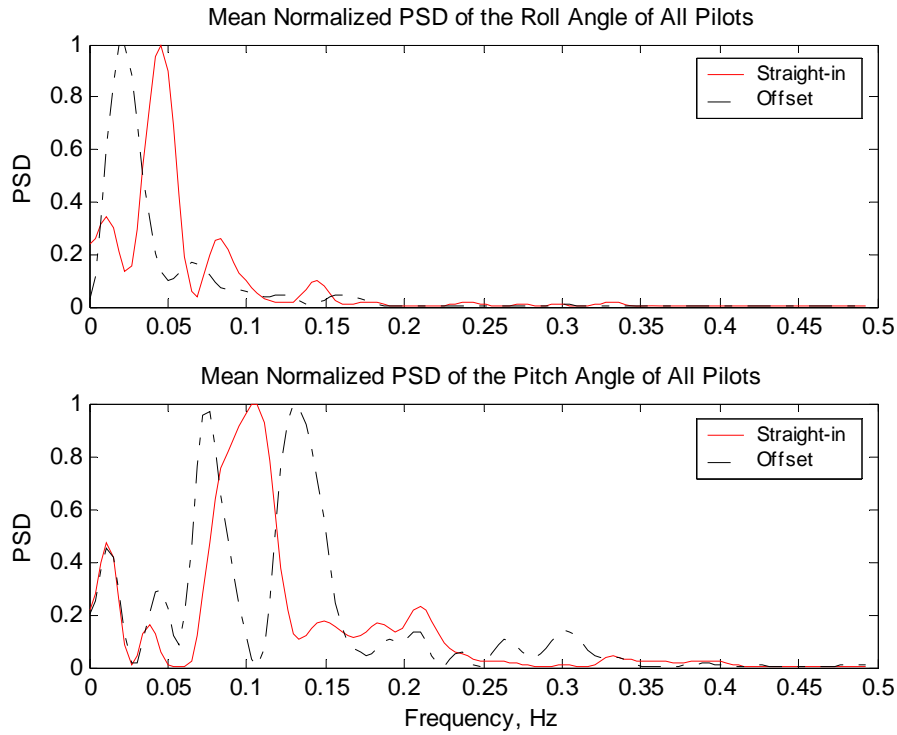
The average of a signal is deducted from the signal before calculating the PSD to avoid an artificial peak at zero Hz, which may dominate the PSD and make other meaningful peaks invisible. Zero padding is also applied to increase the resolution of the adjacent peaks.

For most of the 520 runs ( $13 \times 40$ ), the PSD of the control input, outside the interval  $[0 \ 1]$  Hz contributed less than 5% to the total PSD. Therefore, the PSD over the range  $[0 \ 1]$  Hz of the control input is a reasonable measure of the total energy the pilot uses for control. In this study, the cutoff frequency of the control inputs may be taken as 1 Hz, except for the pedal input which has a narrower band, and a cutoff frequency 0.5 Hz. Note that the processes of removing the mean value of a signal before calculating its PSD and zero padding affect the integrated PSD. Investigation shows that the smoothed algorithm (Eq. (4.4)) is more robust than the direct algorithms in counteracting these two processes. This is another reason why the smoothed algorithm is used for analyzing the integrated PSD of the pilot's control inputs.

#### ***4.4.2. The PSD of the Aircraft Euler Angles***

The aircraft Euler angles are the output of the simulator equations of motion (EOM) in the topodetic (earth surface) reference frame. Fig. 4.21 gives a typical example of the normalized PSD of the roll and pitch angles for the straight-in and offset approaches. The PSD of the aircraft angles revealed some interesting findings. First, the pitch angle PSD was distributed in a wider frequency range than the roll angle PSD. Secondly, the pitch angle PSD contained more high frequency components for the offset

approach than for the straight-in approach, while the roll angle PSD contained fewer high frequency components for the offset approach than for the straight-in approach. The movement of the roll angle PSD components was surprising, and the explanation will be given as follows.



**Fig. 4.21. An example of the normalized PSD of the roll and pitch angles**

Table 4.1 gives the mean values and standard deviations of the frequencies of the highest peak of the PSD of the roll and pitch angle. Note that the roll angle PSD (Row 3) showed a significantly smaller standard deviation in the offset approach than the other three cases. This indicates that the highest roll angle PSD peak was at 0.0288 Hz for all offset runs, varying little from one run to another. This peak is directly related to the offset maneuver because its inverse—34.48 sec is very close to the offset approach duration. This explains why the roll angle PSD for the offset approach contains fewer high frequency components than the straight-in approach. The peak in the roll angle PSD,

related to the 34.48 sec duration, was the outstanding characteristic of the offset approach. The PSD of the roll angle for most offset runs had a second peak at about 0.06 Hz. Unlike the dominant peak, this peak was not present for all offset approach runs. The two peaks of the roll angle PSD in the offset approach agreed with the preliminary piloted tests, and were also reported by Middendorf<sup>12</sup>.

**Table 4.1. Mean & STD of the frequency of the highest peak of PSD of Euler Angles**

Maneuver	Euler Angle	Mean (Hz)	STD (Hz)
Straight-in Approach	Roll	0.0471	0.0051
	Pitch	0.0330	0.0184
Offset Approach	Roll	<b>0.0288</b>	<b>5.87E-4</b>
	Pitch	0.0821	0.0170

Notice that the PSD shown in Fig. 4.21 is not zero at zero frequency even though the mean value was deducted before calculating the PSD. This is an artifact of the smoothing process, which uses windowing, and it can be seen by setting theta to zero in the mathematical expressions given in Eqs. (4.3) and (4.4) . For the direct method

$$\hat{K}_x^f(0) = \frac{1}{N} \left| \sum_{i=0}^{N-1} x[i]w[i] \right|^2 \quad (4.6)$$

If a rectangular window is used, set  $w[i]=1$ , and then

$$\hat{K}_x^f(0) = \frac{1}{N} \left| \sum_{i=0}^{N-1} x[i] \right|^2 = 0$$

The summation is zero because the mean value was subtracted from the signal. For the smoothed algorithm, the PSD at zero frequency is

$$\hat{K}_x^f(0) = \frac{1}{N} \sum_{i=0}^{N-1} x[i] \sum_{l=0}^{N-1} x[N-1-l]w[l+i] \quad (4.7)$$

Similarly, if the window is a rectangle, set  $w[l+i]=1$ , and then Eq. (4.7) becomes

$$\hat{K}_x^f(0) = \frac{1}{N} \left\{ \sum_{i=0}^{N-1} x[i] \right\}^2 = 0.$$

But for a non-rectangular window, Eq. (4.7) is not necessarily zero, because it cannot guarantee the weighted mean of zero-mean signal will be zero. Because the Hamming window was used for all PSD analysis, the PSD in zero frequency was not zero.

#### ***4.4.3. Analysis of the Integrated PSD of Pilot Control***

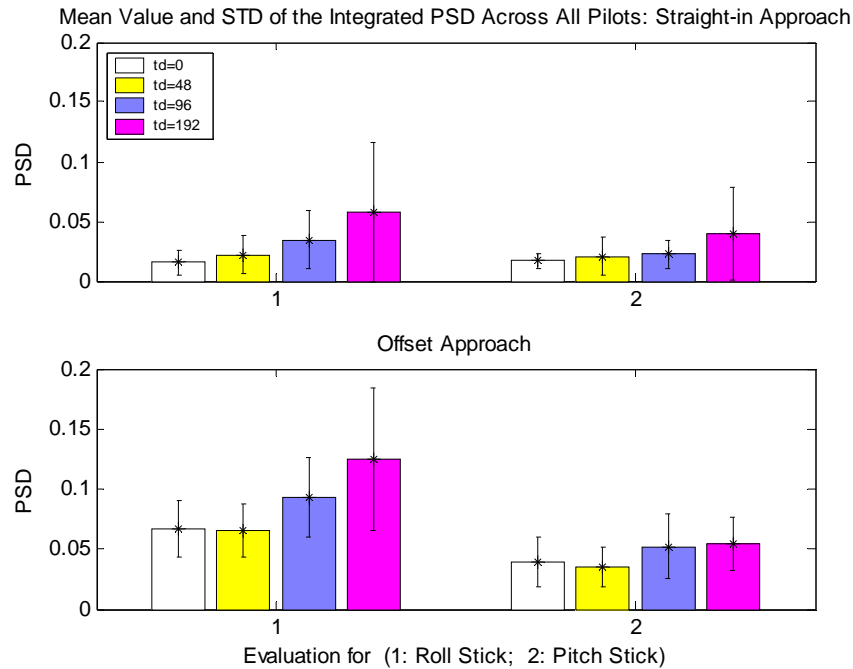
Fig. 4.22 shows the effects of added time delay on the integrated PSD of the pilot's roll and pitch stick inputs. For the straight-in approach, the longer the added time delay, the greater the mean integrated PSD of both control inputs, and the standard deviation tended to be larger also. The increase in the integrated PSD with increasing time delay was obvious for all cases. For the offset approach, the mean integrated PSD of both control inputs tended to increase with the amount of the artificial delay except for the 48 ms delay case. The roll input showed a higher mean integrated PSD than the pitch input, especially for the offset approach.

Though not shown in the figure, the PSDs of the rudder pedal and the throttle are worth mentioning. The 13 pilots revealed the largest standard deviation in the pedal input among all four control inputs. As a result, the addition of time delay generally increased the integrated PSD of the pedal, but there were many exceptions.

The PSD of the throttle inputs did not reveal significant or consistent changes with the time delay or compensation. A similar observation was included in the report on the preliminary piloted tests. For this reason, the throttle input PSD will not appear in the



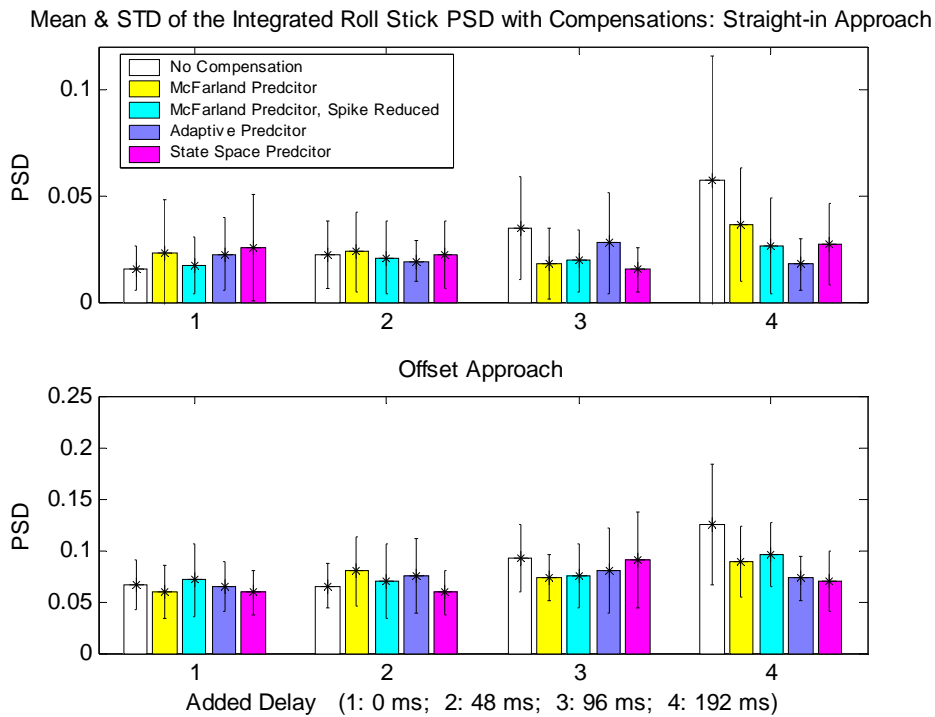
remaining analyses. The analysis will concentrate on the roll and pitch inputs, and the rudder pedal will be included when necessary.



**Fig. 4.22. Mean value & STD of the integrated PSD changing with the time delay**

Fig. 4.23 shows the effects of compensation on the mean values and standard deviations of the integrated PSD of the roll input. For the straight-in approach, increased mean PSD was observed for all four compensation algorithms for the zero added delay case, and for the McFarland predictor and the state space predictor for the 48 ms delay case. For all other cases, all the predictors resulted in decreased integrated PSD. For the offset approach, all except the McFarland predictor with spikes reduced showed a decrease in power for the zero added delay case, and only the state space predictor showed a decrease in power when compensating the 48 ms added delay case. For all other cases, all predictors produced decreased integrated PSD. For both approaches, the two novel predictors had fewer exceptions in reducing the pilot’s roll workload. Significant reduction of the integrated roll input PSD was observed for: 1) all but the

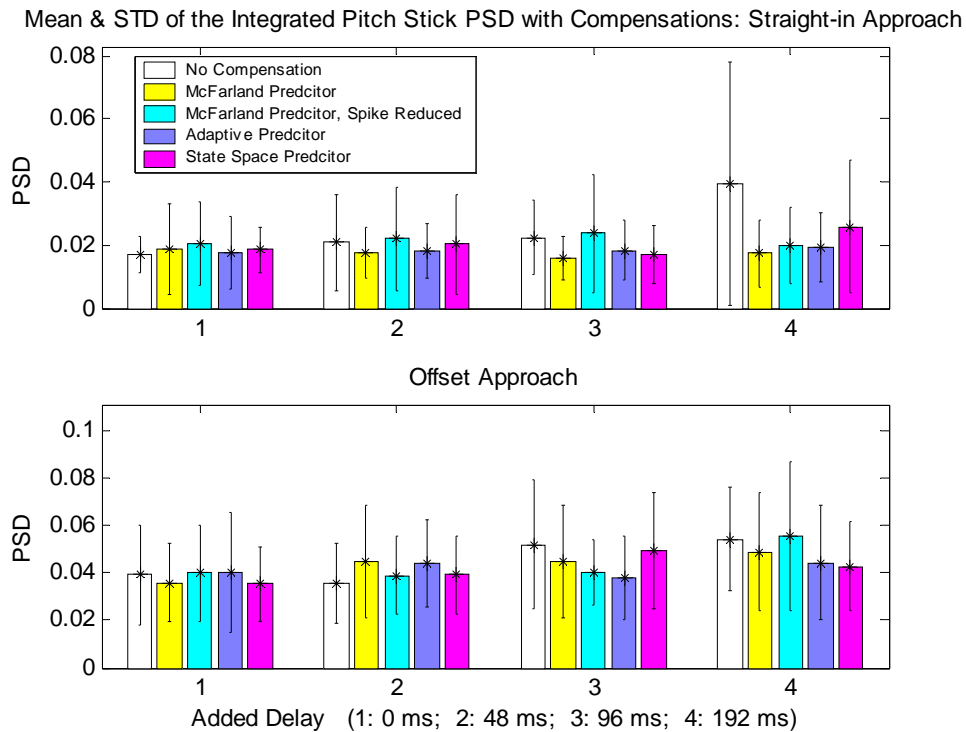
adaptive predictor for the 96 ms delay straight-in approach, 2) all but the McFarland predictor for the 192 ms straight-in approach, and 3) all but the McFarland predictor with spike reduction for the 192 ms delay offset approach. The adaptive predictor showed significant improvement over the McFarland predictor for the 192 ms delay straight-in approach, and the state space filter showed a marginally significant improvement over the McFarland predictor for the 48 ms delay offset approach.



**Fig. 4.23. Mean & STD of the integrated roll input PSD with compensation**

Fig. 4.24 shows the effects of compensation on the mean values and standard deviations of the integrated PSD of the pitch input. The integrated roll input PSD revealed that there were two cases in which the compensations increased the control workload: the zero added delay case for the straight-in approach and the 48 ms delay case for the offset approach. In addition to these cases, the adaptive predictor showed an increased integrated PSD for the zero added delay case in the offset approach. The spike

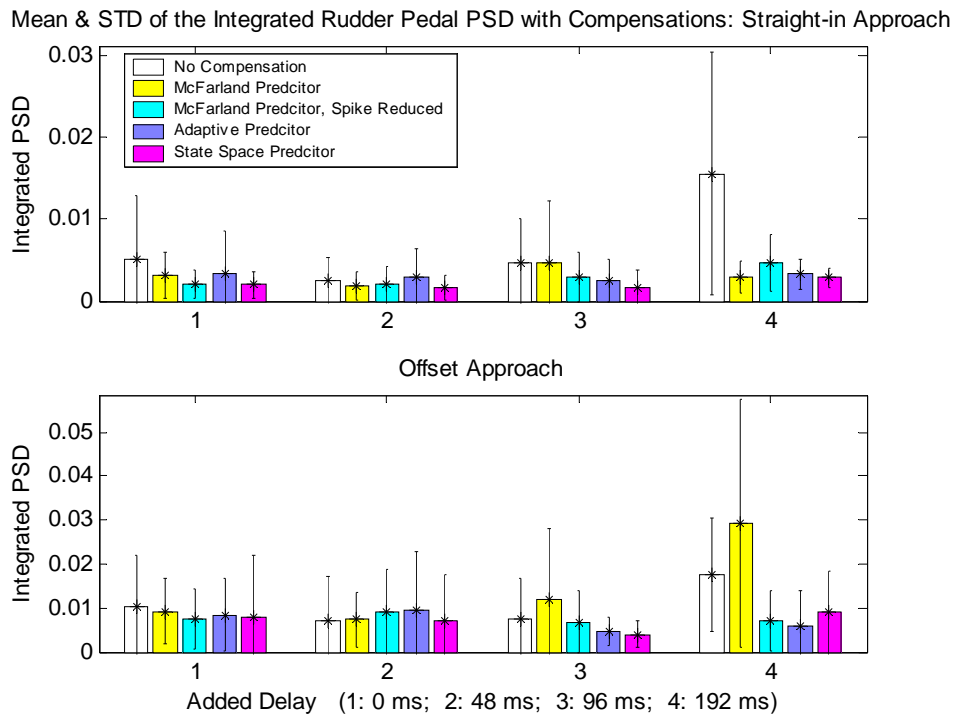
reduction algorithm resulted in poorer compensation than the McFarland predictor for most cases in terms of the pitch input PSD. A significant reduction in PSD was revealed when compensating for the 192 ms delay, for the straight-in approach, with all but the state space predictor.



**Fig. 4.24. Mean & STD of the integrated pitch stick PSD with compensations**

Fig. 4.25 shows the effects of compensation on the mean value and standard deviation of the integrated PSD of the rudder input. The predictors produced surprisingly good compensation results in the rudder input PSD. For the straight-in approach, all compensators revealed a decrease in the integrated PSD except for the McFarland predictor in the 96 ms added delay case. For the offset approach, the two novel predictors revealed a decrease in the integrated PSD except for the adaptive predictor for the 48 ms delay case. The McFarland predictor increased the PSD when compensating for the 48, 96 and 192 ms delay cases for the offset approach. In general the spike reduction

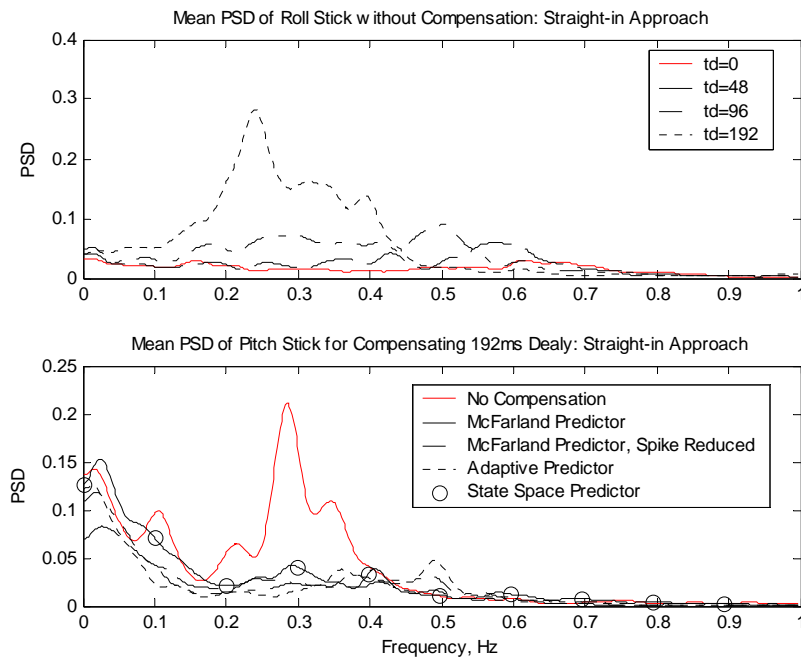
algorithm improved the compensation of the McFarland compensator. The state space predictor produced the lowest integrated rudder input PSD for almost all cases, and it resulted in a significant reduction of the PSD for all delay cases, except the 48ms case, for the offset approach. Other significant pedal input PSD reductions observed include: the McFarland compensation for the 192 ms delay in the straight-in approach; the McFarland predictor with spike reduction for the zero, and the 192 ms delay in the straight-in approach, and for the 192 ms delay in the offset approach; and the adaptive compensation for the 192 ms delay in both approaches. A significant improvement over the McFarland compensation was observed in the adaptive predictor for the 96 and the 192 ms delay cases in the offset approach, and the state space predictor for the 96 ms delay straight-in approach case, and for the 96 and the 192 ms delay cases in the offset approach.



**Fig. 4.25. Mean & STD of the integrated rudder pedal PSD with compensations**

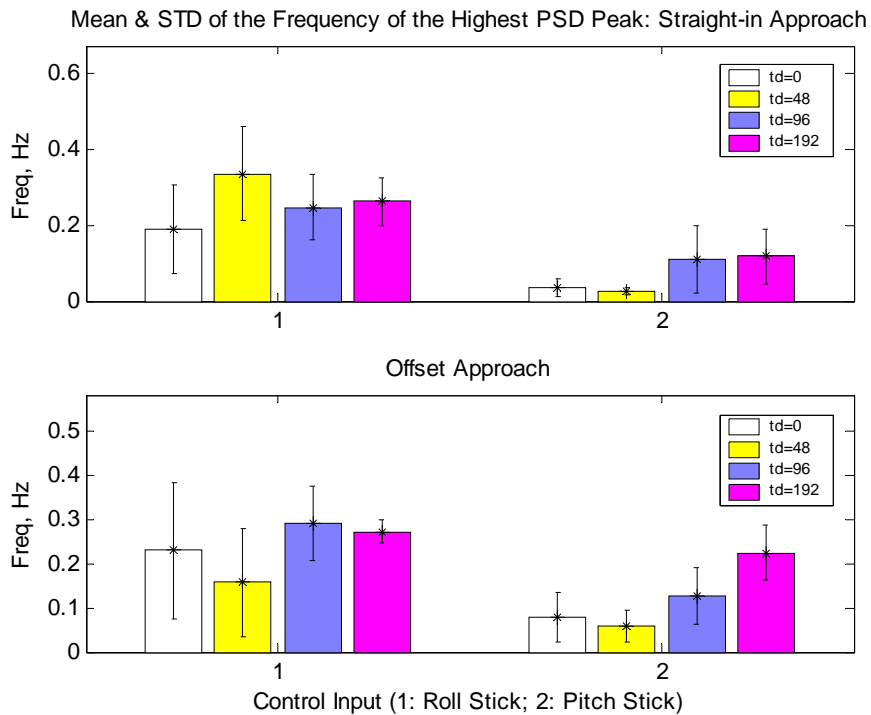
#### 4.4.4. Analysis of the Highest PSD Peak Migration

Time delay and delay compensation not only affected the total PSD of the pilot's control inputs, they also affected the location of the highest peak of the PSD. Fig. 4.26 illustrates some examples of the highest peak migration resulting from the delay and compensation. For both the straight-in and offset approaches, when there was no time delay, the highest peak of the PSD of the roll input or the pitch input usually was either the first peak, or near the first peak. When time delay is added, the highest peak tended to move to the right, or higher frequency area, but the compensation moved it back to lower frequencies. In other words, time delay increased the operator's workload in the high frequencies while compensation decreased it. This result agreed with the results of the preliminary piloted tests. Time delay and compensation made little difference in the highest peak of the PSD of the rudder input and the throttle: the highest peak of their PSD was always the first peak.



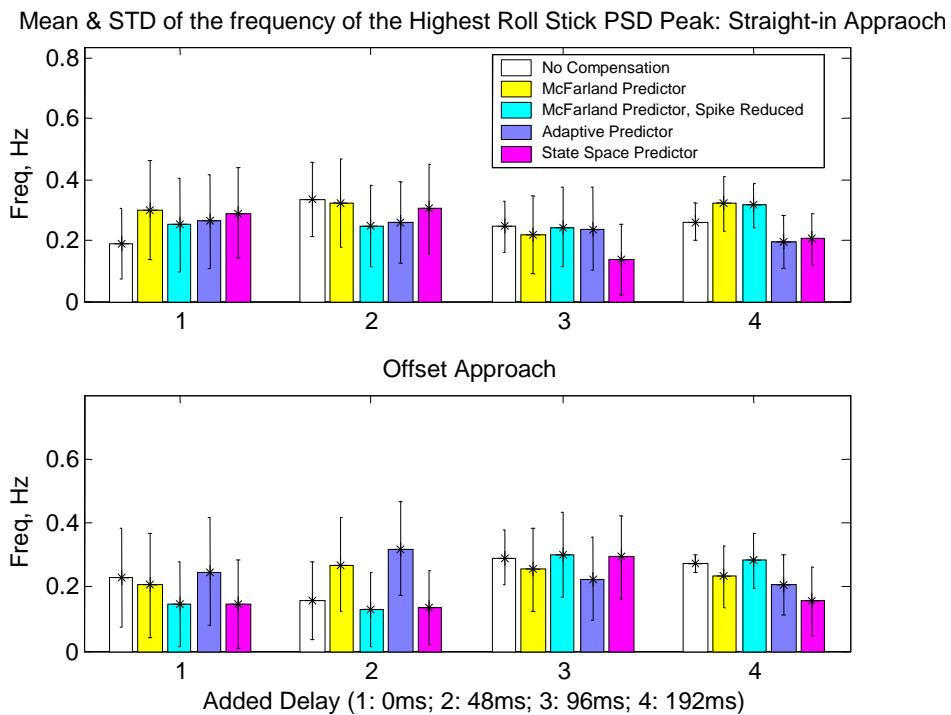
**Fig. 4.26. Highest peak of PSD movement with the delay and compensations**

Fig. 4.27 shows the mean value and standard deviation of the frequency of the highest PSD peak of the roll and pitch inputs for different delay cases. Generally, with longer time delay, the highest peak tended to have a higher frequency. Most of the exceptions were associated with the 48 ms added delay case. In the 48 ms added delay, straight-in approach, the highest peak of the roll input PSD had a higher frequency than that of the 192 ms added delay case. For all other cases, the frequency of the highest peak of the 48 ms delay was even lower than that for the zero added delay. Another exception was the highest roll input PSD in the offset approach decreased from the 96 ms delay case to the 192 ms delay case, with the standard deviation reduced substantially. Conversely, increasing the delay from 96 ms to 192 ms significantly increased the frequency of the highest pitch input PSD for the offset approach. The highest PSD peak of the roll input was clearly higher than that of the pitch input.



**Fig. 4.27. Mean & STD of the frequency of the highest PSD peak with delay**

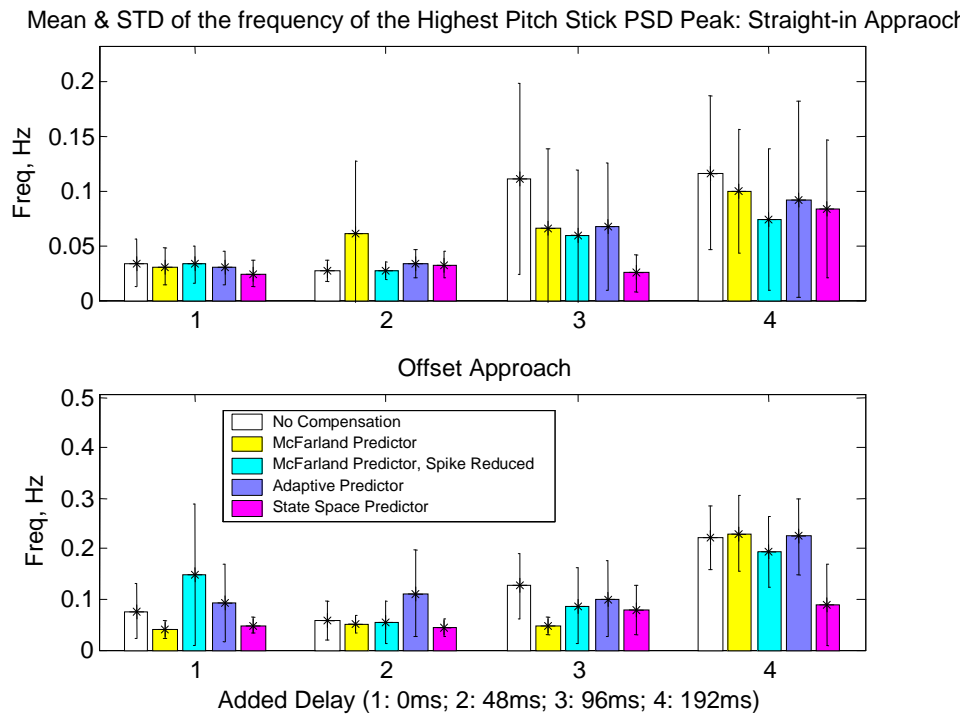
Fig. 4.28 illustrates the mean value and standard deviation of the frequency of the highest PSD peak of the roll inputs for different compensation algorithms. For the straight-in approach, all compensators increased the highest PSD peak frequency for the zero added delay case. The McFarland compensator (with or without spike reduction) showed an increased highest PSD peak frequency for 192 ms delay case. For all other cases, the compensation reduced the frequency of the highest PSD peak. The state space predictor for the 96 ms delay, on the straight-in approach showed the lowest frequency of the highest PSD peak among all predictors for all cases. The two novel predictors demonstrated significantly better compensation than the McFarland predictor in terms of the highest PSD peak movement for the 192 ms delay case.



**Fig. 4.28. Migration of the highest roll stick PSD peak with compensation**

For the offset approach, the adaptive predictor produced an increased mean value of the highest PSD peak for short added time delay (48 ms or less), and the McFarland

predictor with spike reduction resulted in increased mean highest PSD peak for long added time delay cases (96 and 192 ms). Other cases of increased frequency of the highest PSD peak were also observed for the McFarland predictor when compensating 48 ms delay, and the state space predictor when compensating 96 ms delay. For all other cases in the offset approach, the compensation reduced the frequency of the highest PSD peak. The state space predictor showed significant reduction in the highest PSD peak frequency for the 192 ms delay case, and it also showed a significantly lower highest PSD peak than the McFarland predictor for the 48 ms delay compensation case.



**Fig. 4.29. Migration of the highest pitch stick PSD peak with compensation**

Fig. 4.29 is the counterpart of Fig. 4.28 and shows the highest pitch input PSD peak migration. For the straight-in approach, decreased frequency of the highest PSD peak can be observed for all cases except three: compensating the 48 ms delay with the McFarland predictor, the adaptive predictor and the state space filter. Significant



reduction of the highest PSD peak was observed with the state space predictor for the 96 ms delay, which also showed significantly lower frequency of the highest PSD peak than all other compensators.

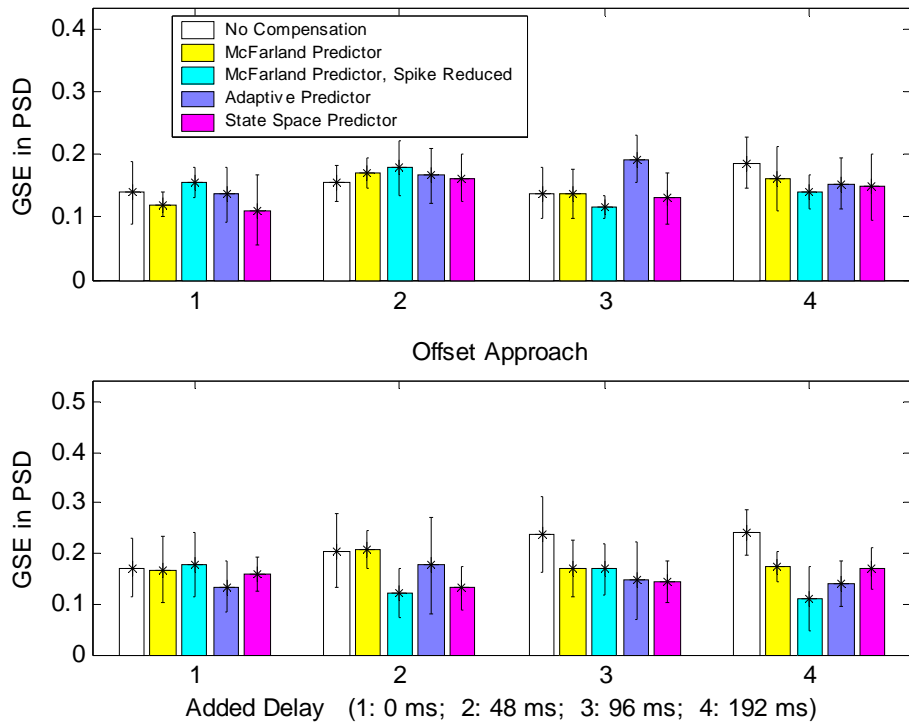
For the offset approach, the adaptive predictor resulted in an increase in the highest PSD peak frequency for all but the 96 ms added delay case. Other highest PSD peak increase cases included the McFarland predictor for 192 ms delay, and the McFarland predictor with spike reduction for zero added delay. For all other cases, the compensation reduced the mean frequency of the highest PSD peak. A significant reduction in the highest PSD peak frequency was shown with the McFarland compensation for zero and 96 ms added delay, and the state space compensation for the zero and 192 ms added delay. In addition, the state space predictor also resulted in significantly lower frequency of the highest PSD peak than the McFarland compensator for the 192 ms delay.

#### ***4.4.5. Analyses on Selected Pilots***

The error bars on the plots discussed in the previous sections indicate a lack of significance among the various cases, while at the same time the trends are in the correct direction. This variation is attributed to the widely differing control strategies employed by the pilots. Brief resumes of all pilots are available in Table A.5. After careful examination of the individual results of all pilots, it is evident that pilots #6, 7, 8 and 9 produced significantly higher measures in most of the metrics. On the other hand, pilots #11, 12 and 13 (all experienced commercial pilots) seemed to employ similar control strategies. The aggregation of the pilots' performance leads to analyses of a few selected pilots so that the standard deviations of some metrics are smaller than for the 13-pilot

group. The selection was made as follows: for all nine sub-metrics (glideslope error, touchdown error, CHR on glideslope and on touchdown, TLX, integrated PSD of roll input, pitch input and rudder input, frequencies of the highest peaks of the roll input and pitch input), there were eight test cases (0, 48, 96 and 192 ms added delays for both approaches), giving 72 total cases. By summing the number of cases for each pilot in which he/she showed improved performance with compensation, five pilots were selected who produced more positive cases than the others. They were pilots #1, 2, 5, 11, and 12. Further investigation shows that this group demonstrated an obvious reduction in standard deviations of the glideslope error, touchdown error, and the integrated PSD of the two primary control inputs, but for the other sub-metrics, the change in standard deviation was neither obvious nor consistent. Therefore, only the results of these four sub-metrics for this five-pilot group are included in Fig. 4.30 – Fig. 4.33.

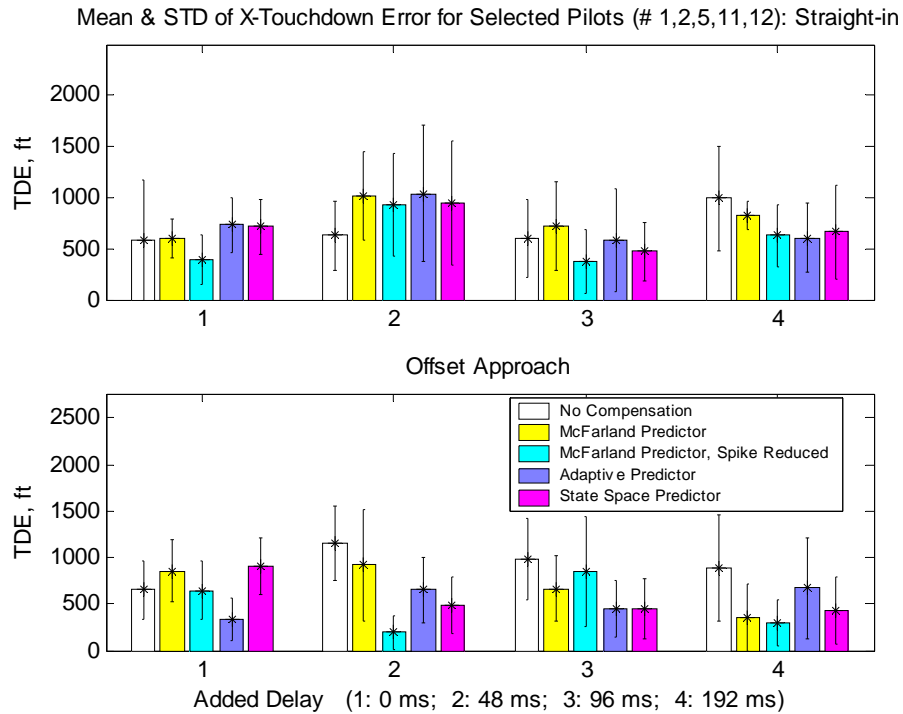
Mean & STD of Glide Slope Error (in PSD) for Selected Pilots (# 1,2,5,11,12): Straight-in



**Fig. 4.30. Mean & STD of the Glideslope Error for Pilots #1,2,5,11,12**

Fig. 4.30 shows the mean values and standard deviations of the glideslope error for the selected pilots with various compensation algorithms. As a comparison to Fig. 4.3 of the 13-pilot group, this chart showed better compensation (lower mean error) for most cases. For the straight-in approach, this group showed decreased standard deviation for all cases except the no-compensation for the 48 ms added delay, the McFarland compensation with spike reduction for the 48 ms added delay, and the adaptive prediction for the zero added delay. For the offset approach, this group revealed reduced standard deviation for all cases except the no-compensation for zero added delay, the McFarland compensation with spike reduction for 48 ms added delay, and the adaptive compensation for the 48 and the 96 ms delays. Significant compensation was found using the

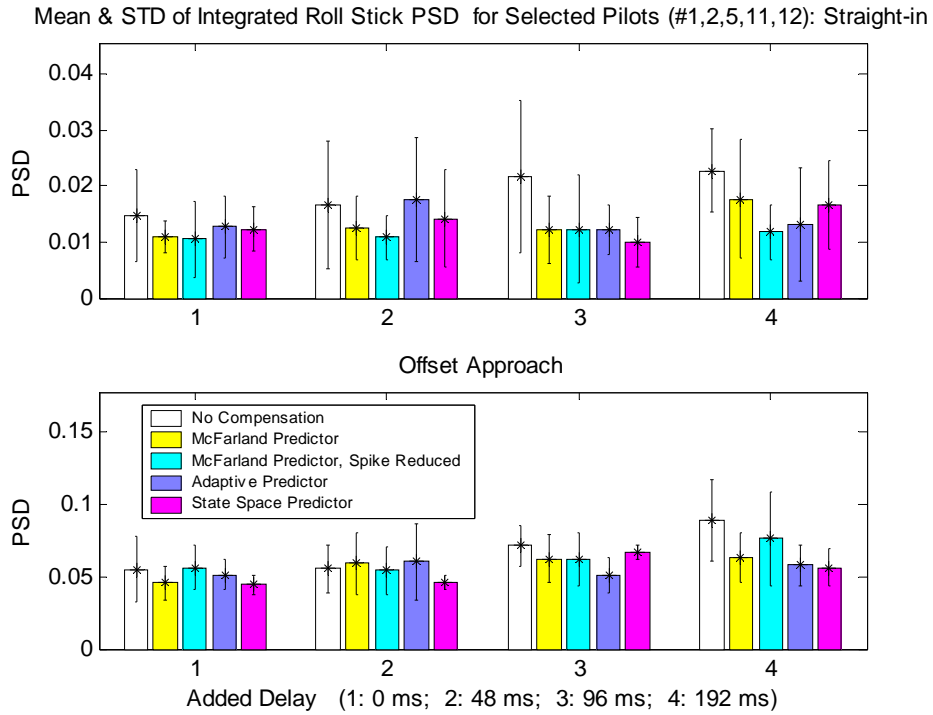
McFarland compensator with spike reduction and adaptive compensator for the 192 ms delay in the offset approach.



**Fig. 4.31. Mean & STD of the Touchdown Error for Pilots #1,2,5,11,12**

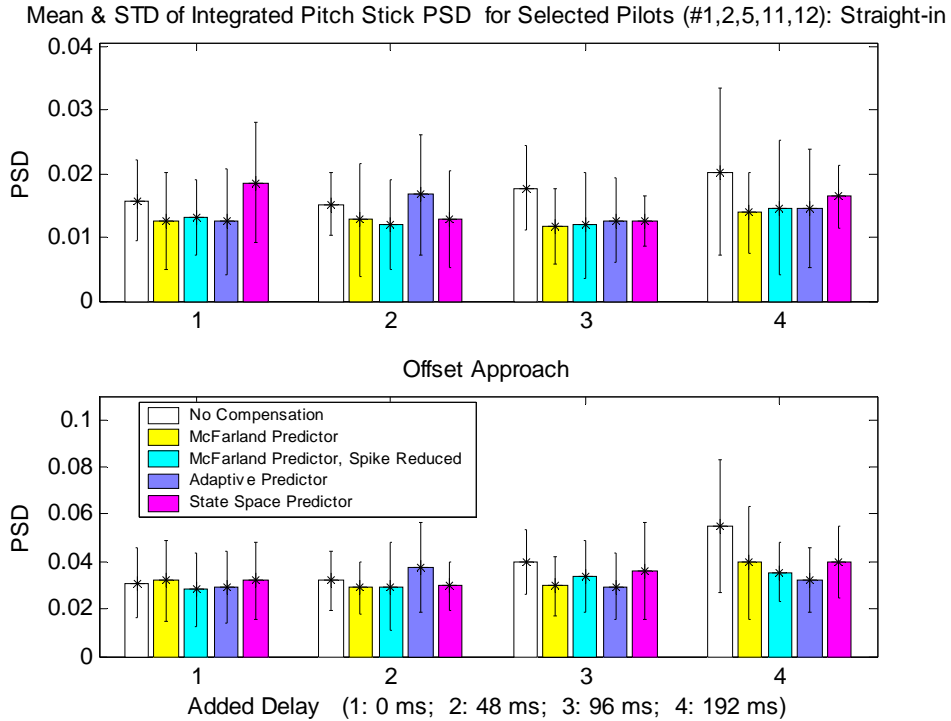
Fig. 4.31 shows the mean values and standard deviations of the touchdown error of the selected pilots with all the compensation algorithms. As a comparison to Fig. 4.8, which includes data from the 13-pilot group, this plot demonstrated generally better compensation in terms of the mean touchdown error. It also showed shorter error bars than for the group of 13 pilots for all cases except the no-compensation with zero added delay straight-in approach case, the McFarland compensation with 48 ms added delay offset approach case, and the reduced-spike McFarland compensation, the adaptive compensation and the state space compensation for the 48 ms added delay straight-in

approach cases. Significant compensation was found with the reduced-spike McFarland predictor for the 48 ms added delay offset approach case.



**Fig. 4.32. Mean & STD of Integrated Roll Input PSD for Pilots #1,2,5,11,12**

Fig. 4.32 shows the mean values and standard deviations of the integrated roll input PSD of the selected pilots with all compensation algorithms. As a comparison to Fig. 4.23, which includes data from the 13-pilot group, this plot group revealed better compensation results for most cases in terms of the mean integrated PSD. This is especially true for the compensation of the zero added delay in the straight-in approach. A reduced standard deviation was found for all cases except for the adaptive compensation for the 48 ms delay straight-in approach. The error bars are considerably shorter for most of the offset approach cases.



**Fig. 4.33. Mean & STD of Integrated Pitch Input PSD for Pilots #1,2,5,11,12**

Fig. 4.33 shows the mean values and standard deviations of the integrated pitch input PSD of the selected pilots with all compensation algorithms. As a comparison to Fig. 4.24, which includes data from the 13-pilot group, this plot group demonstrated better compensation for most cases in terms of the mean values of PSD. The standard deviation was reduced for two cases out of eight for all four compensation algorithms, and it was considerably decreased for the offset approach. Significant compensation was found when using the McFarland compensator for the 96 ms delay offset approach case, the spike-reduced McFarland compensator and the adaptive compensator for the 192 ms delay offset approach case, and the state space compensator for the 96 ms delay straight-in approach case.

#### ***4.4.6. Analysis of PSD in Certain Frequency Intervals***

This analysis was undertaken to determine if there is any relationship between the intervals in which the PSD increases with the delay and the intervals in which the PSD decreases with the compensation.

Fig. 4.26 also shows that the movement of the highest PSD peak as the result of time delay and compensation was accompanied by an increase or decrease in the PSD in the frequency intervals in the neighborhood of the highest peak. These frequency intervals are summarized in Tables 4.2, 4.3 and 4.4.

Table 4.2 shows the frequency intervals in which the PSD of the four pilot control inputs decreased for different delay cases. The interval for a specific delay amount (48, 96 or 192 ms) was obtained by comparing the PSD for that amount of delay to the PSD of the zero added delay case. No compensation was applied. As stated in section 4.4.1, 95% of the power of the roll and pitch inputs fell within the interval [0 1] Hz. The frequency interval in which the time delay raised the PSD of the two control inputs was narrower than [0 1] Hz. The lower limit of the interval was not necessarily zero but was close to zero, and the upper limit was around 0.5 – 0.6 Hz, seldom exceeding 0.7 Hz. For both primary control inputs, in both approaches, one interesting trend was that the PSD frequency-increase interval tended to shrink with increasing delay: the upper bound decreased as the added delay got longer, while the lower limit revealed no consistent change. For both control inputs, the upper limit of the PSD frequency-increase interval was considerably lower for the offset approach than for the straight-in approach.

**Table 4.2. Frequency intervals in which the PSD increases by time delay**

Maneuver	Delay (ms)	Straight-in Approach (Hz)	Offset Approach (Hz)
Roll Stick	48	[0 0.656]	[0.063 0.605]
	96	[0 0.602]	[0.062 0.573]
	192	[0 0.531]	[0.056 0.471]
	<b>Mean</b>	<b>[0 0.596]</b>	<b>[0.060 0.550]</b>
Pitch Stick	48	[0.017 0.68]	[0.188 0.632]
	96	[0.114 0.62]	[0 0.47]
	192	[0 0.505]	[0.108 0.416]
	<b>Mean</b>	<b>[0.044 0.632]</b>	<b>[0.099 0.506]</b>
Pedal	48	/	/
	96	[0.039 0.16]	[0.114 0.408]
	192	[0 0.5]	[0.063 .5]
Throttle	48	[0 0.325]	/
	96	[0 0.065]	[0.082 .5]
	192	/	/

For the pedal and throttle, there was no consistent PSD frequency-increase interval. It might be due to the fact that the PSDs of both had one major peak near 0.05 Hz that did not move much for the two conditions. Therefore, in the analysis, regarding the effects of the compensation on the frequency interval, only the PSD of the two primary controls are described.

Table 4.3 shows the frequency intervals in which the PSD of both control inputs was reduced by different compensation algorithms. Each interval was the result of averaging four intervals corresponding to the four amounts of delay. In general, the four



predictors did not show substantial or consistent difference in the PSD-reduction interval of frequency. However, for most cases, the adaptive predictor resulted in the narrowest PSD-reduction intervals of frequency, while the state space predictor resulted in the widest intervals.

**Table 4.3. Frequency intervals in which the PSD decreases by different predictors**

Maneuver	Delay (ms)	Straight-in Approach (Hz)	Offset Approach (Hz)
Roll Stick	McFarland	[0.105 0.492]	[0.039 0.537]
	Spike Reduced	[0.056 0.572]	[0.037 0.560]
	Adaptive	[0.078 0.541]	[0.053 0.514]
	State Space	[0.000 0.581]	[0.025 0.581]
Pitch Stick	McFarland	[0.102 0.608]	[0.136 0.494]
	Spike Reduced	[0.100 0.525]	[0.089 0.510]
	Adaptive	[0.038 0.549]	[0.113 0.452]
	State Space	[0.062 0.586]	[0.129 0.540]

Table 4.4 gives the PSD frequency-reduction intervals caused by the compensation grouped by the amounts of delay. With the exception of the transition from zero added delay to 48 ms delay for the roll input PSD on the straight-in approach, the PSD-frequency-reduction interval with compensation always shrank with increasing added delay. This resembled the PSD-increase interval caused by the time delay. However, the mean PSD-reduction interval, caused by the addition of compensation, for each of the four cases (a combination of the two primary inputs and the two approaches), was narrower than the corresponding mean PSD-increase interval resulting from the time delay. This can be observed by comparing each interval in bold in Table 4.4 with the

corresponding item in bold in Table 4.2. The reduced upper limit of the PSD-reduction interval implies that the PSD between the upper limit and 1 Hz was also increased by the compensation. This proves that compensators introduced gain distortion at high frequencies. This means that, the current compensators cannot restore the system performance to one hundred percent. In other words, the compensated system cannot be exactly the same as the undelayed system.

**Table 4.4. Frequency intervals of reduced PSD by compensation of different delays**

Maneuver	Delay (ms)	Straight-in Approach (Hz)	Offset Approach (Hz)
Roll Stick	0	[0.094 0.529]	[0.037 0.735]
	48	[0.116 0.626]	[0.000 0.529]
	96	[0.032 0.567]	[0.056 0.526]
	192	[0.000 0.448]	[0.059 0.403]
	<b>Mean</b>	<b>[0.060 0.54]</b>	<b>[0.038 0.548]</b>
Pitch Stick	0	[0.109 0.686]	[0.094 0.622]
	48	[0.070 0.638]	[0.199 0.527]
	96	[0.101 0.549]	[0.062 0.469]
	192	[0.022 0.436]	[0.112 0.378]
	<b>Mean</b>	<b>[0.076 0.577]</b>	<b>[0.117 0.499]</b>

#### 4.5. Summary of Results

So far the analyses results have been presented individually in terms of four metrics: glideslope error (GSE), touchdown error (TDE), Cooper-Harper Rating (CHR) (on glideslope and on touchdown) and power spectral density (integrated PSD and the frequency of the highest PSD peak). Cross-metric comparisons have been made only between the CHR and the performance errors (GSE and TDE). These are inadequate

because there was no overall comparison. An overall comparison would have to cover all four metrics (10 sub-metrics) and eight conditions (four amounts of delay times two different approaches) among the four predictors—McFarland predictor (MF), McFarland predictor with spike reduction (MFR), adaptive predictor (AP) and state space predictor (SS). The 10 sub-metrics are: GSE, TDE (in the X direction), CHR on GS, CHR on TD, TLX, integrated PSD (IPSD) of the roll control input (RS), pitch control input (PS) and rudder input (RP), and the frequencies of the highest PSD peak (FPSD) of the roll input and pitch input. This section presents an overall comparison.

Table 4.5 is a summary of the compensation results, in which a “↓” means the mean value of a certain sub-metric is decreased by the compensation compared to that of the uncompensated case, and a “↑” means the opposite. The last column gives the total occurrences of “↓” for that condition. This table shows that for all four predictors, the compensation was more obvious for a long added delay (96 ms or more) than for a short one (48 ms or less). For the short delay compensation, these 10 sub-metrics showed discrepancies: about one half increased while the other half decreased. As was previously stated, the discrepancy might be the result of an obscure and inconsistent increase of each sub-metric due to the short amount of delay. A second possible cause might come from factors in the simulation other than the delay and compensation, such as the wind vector or the turbulence, which might become dominant over the compensation for a short delay. Finally, the pilots themselves might produce anomalies.

**Table 4.5. Summary of compensation results**

Predictor	Approach	Delay (ms)	Metrics										
			GSE	TDE	CHR (GS)	CHR (TD)	TLX	IPSD (RS)	IPSD (PS)	IPSD (RP)	FPSD (RS)	FPSD (PS)	# of ↓
MF	Straight-in	0	↓	↑	↓	↓	↑	↑	↑	↓	↑	↓	5
		48	↑	↑	↑	↓	↑	↑	↓	↓	↓	↑	5
		96	↑	↑	↓	↓	↓	↓	↓	↑	↓	↓	7
		192	↓	↓	↓	↓	↓	↓	↓	↓	↑	↓	9
	Offset	0	↑	↑	↓	↑	↓	↓	↓	↓	↓	↓	7
		48	↑	↑	↓	↓	↑	↑	↑	↑	↑	↓	3
		96	↑	↑	↓	↓	↓	↓	↓	↑	↓	↓	7
		192	↓	↓	↓	↓	↑	↓	↓	↑	↓	↑	7
MFR	Straight-in	0	↑	↓	↓	↑	↓	↑	↑	↓	↑	↓	5
		48	↑	↑	↑	↑	↓	↓	↑	↓	↓	↓	5
		96	↓	↑	↓	↓	↓	↓	↑	↓	↓	↓	8
		192	↓	↑	↓	↓	↓	↓	↓	↓	↑	↓	8
	Offset	0	↑	↑	↑	↑	↑	↑	↑	↓	↑	↑	1
		48	↓	↓	↓	↓	↓	↑	↑	↑	↓	↓	7
		96	↓	↓	↑	↓	↓	↓	↓	↓	↑	↓	8
		192	↓	↓	↓	↓	↓	↓	↓	↓	↑	↓	9
AP	Straight-in	0	↓	↓	↓	↑	↑	↑	↑	↓	↑	↓	5
		48	↓	↑	↑	↑	↑	↓	↓	↑	↓	↑	4
		96	↑	↓	↓	↓	↓	↓	↓	↓	↓	↓	9
		192	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	10
	Offset	0	↓	↑	↑	↑	↓	↓	↑	↓	↑	↑	4
		48	↑	↓	↓	↓	↑	↑	↑	↑	↓	↑	4
		96	↓	↓	↑	↓	↓	↓	↓	↓	↓	↓	9
		192	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	10
SS	Straight-in	0	↓	↓	↓	↑	↑	↑	↑	↓	↑	↓	5
		48	↑	↑	↑	↓	↓	↑	↓	↓	↓	↑	5
		96	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	10
		192	↓	↑	↓	↓	↓	↓	↓	↓	↓	↓	9
	Offset	0	↓	↑	↑	↓	↑	↓	↓	↓	↓	↓	7
		48	↓	↓	↓	↓	↓	↓	↑	↓	↓	↓	9
		96	↓	↓	↑	↓	↑	↓	↓	↓	↓	↓	8
		192	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	10

The spike reduction algorithm seemed to make very little difference to the McFarland predictor. In the offset approaches, it produced significantly more increased sub-metrics for zero added delay case, somewhat fewer increases for the 48 ms delay case, and slightly fewer increases for the longer delay cases.

The adaptive predictor showed slightly more increased sub-metrics than the McFarland predictor when compensating for the 48 ms delay in the straight-in approach, and twice as many increased sub-metrics for the zero delay offset approach. However, it revealed an obvious improvement over the McFarland predictor when compensating long delays for both approaches. This is reminiscent of the theoretical analysis results given in Table 4.5, NASA CR 2007-215095, which also shows that the adaptive predictor produces larger gain distortion for short delays, but smaller gain distortion for long delays when compared to the McFarland predictor. The results from the theoretical data and from piloted experimental data agree with each other.

**Table 4.6. Compensation results for the state space predictor with short delay**

Predictor	Approach	Delay (ms)	Metrics										Total ↓ out of 10
			GSE	TDE	CHR (GS)	CHR (TD)	TLX	IPSD (RS)	IPSD (PS)	IPSD (RP)	FPSD (RS)	FPSD (PS)	
SS	Straight-in	0	↑	↓	↓	↑	↓	↑	↓	↓	↓	↓	7
		48	↓	↓	↓	↑	↓	↓	↑	↓	↓	↓	8
	Offset	0	↓	↓	↑	↓	↓	↓	↓	↓	↓	↑	8
		48	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	10
Total ↓ out of 4			3	4	3	2	4	3	3	4	4	3	33 of 40

The state space predictor resulted in exactly the same number of reduced sub-metrics as the McFarland predictor for short delays in the straight-in approach, and in the 0ms added delay case for the offset approach. The state space predictor also produced significantly more reduced submetrics in the 48 ms added delay offset approach case, while it showed substantially fewer increased sub-metrics for the long delays when compared to the McFarland predictor. Notice that for short delays, there were fewer increased objective sub-metrics, especially the direct measures of the system performance (GSE and TDE), associated with the state space predictor. Another comparison between

the two for only short delays is shown in Table 4.6, where a “↓” means the state space predictor produces a lower mean value of a specific sub-metric than the McFarland predictor, and a “↑” means the opposite. The state space predictor showed better compensation than the McFarland predictor with a ratio of 31:9. Obvious improvement has been demonstrated with the state space predictor.

Different compensation results among the four predictors can also be seen by comparing the number of significant compensations, as summarized in Table 4.7. While the McFarland predictor (with or without spike reduction) and the adaptive predictor produced almost the same number of significant compensation cases, the state space predictor achieved considerably more significant compensation cases. Note that most of the significant compensation cases were associated with the long delays especially the 192 ms delay, indicating that compensation was more beneficial for a long time delay. There were also more significant compensation cases associated with the straight-in approach than the offset one, however this might not indicate that compensation was more beneficial for the straight-in approach, but instead might be a result of the differences between the winds and turbulence between the two approaches.

Significant difference among the four compensators can also be seen by directly comparing their 10 sub-metrics, with the results summarized in Table 4.8. In this table, the first number in the second column is the total number of cases in which the compensator in the first column showed significantly better performance than the compensators in the last column, at the conditions listed in column 4. Conversely, the second number in the second column (in parentheses) is the total number of cases in which the compensator in the first column showed significantly worse performance than

the compensators in the last column, at the conditions listed in column 4. For example, in the second row the numbers “6 (17)” mean that the McFarland predictor showed significantly better performance than the other three compensators for 6 cases, but showed significantly worse performance for 17 cases. This table reveals three obvious results. First, the reduced-spike McFarland compensator and the adaptive compensator showed approximately the same number of significantly better cases and fewer significantly worse cases than the McFarland compensator. Second, the reduced-spike McFarland compensator resulted in almost the same number of significantly better cases and significantly worse cases as the adaptive predictor. Third, the state space filter showed significantly more “better cases” and fewer “worse cases” than the other three compensators.

**Table 4.7. Summary of significant compensations**

Predictor	Total	Metric	Conditions
McFarland	7	CHR (GS)	192 ms, straight-in
		IPSD (RS)	96 ms, straight-in
		IPSD (RS)	192 ms, offset
		IPSD (PS)	192 ms, straight-in
		IPSD (RP)	192 ms, straight-in
		FPSD (PS)	0 ms, offset
		FPSD (PS)	96 ms, offset
Spike Reduced	6	IPSD (RS)	96 ms, straight-in
		IPSD (RS)	192 ms, straight-in
		IPSD (PS)	192 ms, straight-in
		IPSD (RP)	0 ms, straight-in
		IPSD (RP)	192 ms, straight-in
		IPSD (RP)	192 ms, offset
Adaptive	7	CHR (GS)	192 ms, straight-in
		CHR (TDE)	192 ms, straight-in
		IPSD (RS)	192 ms, straight-in
		IPSD (RS)	192 ms, offset
		IPSD (PS)	192 ms, straight-in
		IPSD (RP)	192 ms, straight-in
		IPSD (RP)	192 ms, offset
State Space	11	CHR (TD)	192ms, straight-in
		IPSD (RS)	96 ms, straight-in
		IPSD (RS)	192 ms, straight-in
		IPSD (RS)	192 ms, offset
		IPSD (RP)	0 ms, straight-in
		IPSD (RP)	96 ms, straight-in
		IPSD (RP)	192 ms, straight-in
		FPSD (RS)	192 ms, offset
		FPSD (PS)	96 ms, straight-in
		FPSD (PS)	0 ms, offset
		FPSD (PS)	192 ms, offset

Almost all of the significant compensation cases listed in Table 4.7 were associated with PSD metrics (except the first case which is associated with CHR) . This means that the PSD metric showed clearer difference between the compensators. The two novel predictors showed more substantial improvement in moving the highest PSD peak back to lower frequencies than did the McFarland predictor.



**Table 4.8. Summary of significant difference among the four predictors**

Predictor (Better)	Total Better (Worse)	Metric	Conditions	Predictor (Worse)
McFarland (MF)	6 (17)	FPSD (PS)	0 ms, offset	MFR
		FPSD (PS)	0 ms, offset	AP
		FPSD (PS)	48 ms, offset	AP
		FPSD (PS)	96 ms, offset	MFR
		FPSD (PS)	96 ms, offset	AP
		FPSD (PS)	96 ms, offset	SS
McFarland Spike Reduced (MFR)	4 (6)	IPSD (RP)	192 ms, offset	MF
		FPSD (RS)	48 ms, offset	MF
		FPSD (PS)	48 ms, straight-in	MF
		FPSD (PS)	48 ms, offset	AP
Adaptive Predictor (AP)	5 (8)	IPSD (RS)	192 ms, straight-in	MF
		IPSD (RP)	96 ms, offset	MF
		IPSD (RP)	192 ms, offset	MF
		FPSD (RS)	192 ms, straight-in	MF
		FPSD (PS)	48 ms, straight-in	MF
State Space (SS)	17 (1)	TDEX	96 ms, straight-in	MF
		IPSD (RP)	96 ms, straight-in	MF
		IPSD (RP)	96 ms, offset	MF
		IPSD (RP)	192 ms, offset	MF
		FPSD (RS)	192 ms, straight-in	MF
		FPSD (RS)	48 ms, offset	MF
		FPSD (RS)	192 ms, offset	MF
		FPSD (PS)	48 ms, straight-in	MF
		FPSD (PS)	96 ms, straight-in	MF
		FPSD (PS)	96 ms, straight-in	MFR
		FPSD (PS)	96 ms, straight-in	AP
		FPSD (PS)	0 ms, offset	MFR
		FPSD (PS)	0 ms, offset	AP
		FPSD (PS)	48 ms, offset	AP
		FPSD (PS)	192 ms, offset	MF
		FPSD (PS)	192 ms, offset	MFR
FPSD (PS)	192 ms, offset	AP		

Finally, only the McFarland predictor resulted in runs with a full CHR, in each of which the pilot reported pilot induced oscillation.

Learning among the pilots may have been a problem. But, the randomization of the runs removes any ability to track this effect. Given more time, we would have “trained” the pilots to minimize learning in the tests.

## 5. Conclusions and Future Research

### 5.1. Conclusions

A device called SIMES was used to measure the transport delay in the Visual Motion Simulator (VMS) at the NASA Langley Research Center. The transport delays in the visual system, motion system and 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 outputs 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 16.7 ms. Therefore, the average baseline transport delay of the visual cue was 90 ms (rounded from 89.7 ms). This formed a basis for the time delay compensator designs for the final piloted simulation tests.

Two objective metrics, the power spectral density (PSD) of the pilot control inputs and the errors in the glideslope and the touchdown position during the approaches, were used in conjunction with two quasi-objective metrics, the Cooper-Harper rating (CHR) and NASA Task Load index (TLX), to assess the effectiveness of the three compensators in piloted simulation tests. Ten sub-metrics were derived from these four evaluations: glideslope error (GSE), touchdown error (TDE), CHR on GS, CHR on TD, TLX, integrated PSD (IPSD) of the roll input (RS), the pitch input (PS) and the rudder input (RP), and the frequency of the highest PSD peak (FPSD) of the RS and PS. These showed considerably more discrepancies for the short delay than for the long delays when evaluating the effectiveness of the compensators. Some obvious discrepancies existed among these sub-metrics between various cases. There were more discrepancies between the GSE and the CHR on GS than between the TDE and CHR on TD. The CHR

and TLX agreed with each other fairly well. These discrepancy cases indicate that not all of the sub-metrics are sensitive enough, and none of them is perfect. Nevertheless, combining them as a whole should yield a generally correct evaluation.

About 95% of the PSD of the roll and pitch inputs was distributed within the frequency interval [0 1] Hz. The PSD interval of the rudder input and throttle was [0 0.5] Hz and [0 0.2] Hz, respectively. The PSD of the throttle did not show obvious or consistent changes with the addition of time delay and compensation. The PSDs of the two primary inputs had many peaks, whereas the PSDs of the rudder and throttle usually had only a dominant peak at around 0.035 Hz (for the pedal) or 0.02 Hz (for the throttle), which did not move much with the delay or compensation. The PSD of the roll angle showed a dominant peak at 0.0288 Hz, which appeared to be the inverse of the duration of the offset approach.

All ten sub-metrics revealed a fairly consistent, but not substantial, increase for the 48 ms delay, and showed a substantial increase for the 96 and 192 ms added delays with few exceptions. Adding 192 ms delay resulted in several full-scale CHRs, in which pilot induced oscillation were reported. With the exception of the longitudinal touchdown error, all the metrics showed larger mean values for the offset approach than for the straight-in approach.

Approximately five sub-metrics showed a decreased mean value by all four of the predictors, while the remaining five resulted in an increased mean value. The discrepancy might be the result of an obscure and inconsistent increase of each sub-metric by the short time delay, and the compensation effects for short delays might be dominated by other simulation conditions, such as the wind vector or turbulence, combined with pilot

anomalies. All predictors showed obvious compensation for long delays. The two novel predictors resulted in more cases of reduced objective evaluations in compensating short delays than the other two predictors (Table 4.5).

None of the compensators showed obvious or consistent changes to the TLX for short delays. However, they tended to reduce the ratings of the physical and temporal demands, increase the ratings of the mental demand and performance, and produce inconsistent changes in the ratings of the effort and frustration for the zero and 48 ms added delay.

Only the McFarland compensator resulted in multiple full-scale CHR cases. These were cases in which the uncompensated CHR did not exceed 4. In all full CHR cases, pilot-induced oscillations (PIO) were cited. The lack of any full scale CHRs by the other three predictors, indicates some improvement in their ability to avoid PIO.

The spike reduction algorithm made an inconsistent impact on the McFarland compensator. With the spike-reduction algorithm, the McFarland compensator revealed slightly worse compensation for short delays, but slightly better compensation for long delays. The McFarland compensator produced only one fewer superior compensation case, and many fewer inferior compensation cases, when compared to the other compensators, then did the spike reduction algorithm (Table 4.7). The spike reduction algorithm did reduce the jitter in the visual image and made the compensator more tolerable for the pilot.

The adaptive predictor seemed to be slightly inferior to the McFarland predictor in compensating for short delays, but obviously superior to it in compensating for the long delays. This agreed with the theoretical analysis. While these two predictors resulted

in about the same number of significant compensation cases, the adaptive predictor showed more cases of significant superiority, and fewer cases of significant inferiority, when compared to other predictors, than did the McFarland predictor (Table 4.7).

The state space predictor resulted in exactly the same number of reduced sub-metrics as the McFarland predictor for the short delays in both approaches, while it showed substantially fewer increased sub-metrics for long delays as compared to the latter (Table 4.5). For short delays, the state space predictor resulted in 31 better compensation cases than the McFarland predictor, and only 9 worse compensation cases (Table 4.6). The state space predictor also revealed more significant compensation cases than the McFarland compensator (Table 4.7). The McFarland predictor resulted in 6 cases of significant superiority and 17 cases of significant inferiority to other predictors, while the corresponding numbers of cases for the state space predictor were 17 and 1. All these results showed that the state space predictor made a substantial improvement over the McFarland predictor for all delays (Table 4.8).

For both the straight-in and offset approaches, when there was no time delay, the highest peak of the PSD of the roll input or the pitch input was either the first peak, or near the first peak. When time delay was presented, the highest peak tended to move to the right, or into a higher frequency area, but the compensation moved it back to lower frequencies. In other words, time delay increased the operator's workload in the high frequencies but the compensation decreased it.

The transport delay and compensation did not affect the PSD of the two control inputs in the whole interval [0 1] Hz, instead, they changed the PSD in certain sub-intervals. The averaged sub-intervals in which the time delay increased the PSD of the

roll input were [0 0.596] and [0.060 0.550] Hz for the two approaches, respectively, and for the pitch input they were [0.044 0.632] and [0.099 0.506] Hz. These frequency intervals tended to shrink with the amount of delay; that is the longer the delay, the narrower the intervals were.

Similar to the PSD-increase intervals associated with the time delay, there were also PSD-reduction intervals associated with the compensation. The mean PSD-reduction intervals for the roll input are [0.060 0.54] and [0.038 0.548] Hz for the two approaches, respectively, and for the pitch input they were [0.076 0.577] and [0.117 0.499] Hz. The PSD-reduction intervals of the compensation were narrower than the corresponding PSD-increase intervals. The PSD beyond the PSD-reduction intervals even increased with the compensation because of the gain distortion of the predictors at high frequencies. The PSD-reduction interval also shrank with the amount of delay, indicating the gain distortion is likely to occur at a lower frequency for longer delay compensation. The width of the interval did not vary considerably between the predictors. For most cases, the adaptive predictor resulted in the narrowest PSD frequency reduction intervals, while the state space predictor resulted in the widest intervals.

The two novel predictors did not demonstrate the superiority to the McFarland predictor in the piloted tests that was indicated in the theoretical offline tests, primarily because of the large standard deviations, which resulted from two factors. First, the subject population could not be optimally chosen because the subjects were volunteers. Secondly, the 13-pilot population was not large enough to generate statistically significant results. Therefore, the theoretical analyses better reveal the improved performance of the two novel predictors over the McFarland predictor.

## 5.2. Suggested Future Research

It would be worth while to implement a frequency domain method to measure the simulator transport delay and phase lag, and compare the results with those determined 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, this method did not result in satisfactory results because aliasing was present in the output aircraft states of the EOM. 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 can be avoided by choosing a lower rate of frequency increase in the sweep signal.

One additional problem could be learning among the pilots. The randomization of the cases removes any ability to track the effect. Given more time, the pilots could be “trained” to minimize learning during the tests.

## Appendix A. Run Matrices and Print Variables

Nomenclature:

Compensation: AP Adaptive Predictor

MF McFarland Predictor

MFR McFarland Predictor with Spike Reduction

SS State Space Predictor

NC No Compensation

**Table A.1. Straight-in approach run matrix**

Run	Delay (ms)	Compensation
1	0	SS
2	48	SS
3	0	MF
4	192	SS
5	96	AP
6	48	MF
7	96	NC
8	48	NC
9	48	MFR
10	0	AP
11	96	MFR
12	192	MF
13	192	NC
14	0	MFR
15	192	MFR
16	48	AP
17	96	MF
18	192	AP
19	96	SS
20	0	NC



Nomenclature:

Compensation: AP Adaptive Predictor

MF McFarland Predictor

MFR McFarland Predictor with Spike Reduction

SS State Space Predictor

NC No compensation

**Table A.2. Offset approach run matrix**

Run	Delay (ms)	Compensation
21	100	SS
22	50	MF
23	100	NC
24	0	MFR
25	100	MFR
26	200	MFR
27	200	NC
28	100	AP
29	50	AP
30	0	MF
31	0	AP
32	50	MFR
33	0	NC
34	0	SS
35	200	AP
36	200	MF
37	50	NC
38	200	SS
39	50	SS
40	100	MF

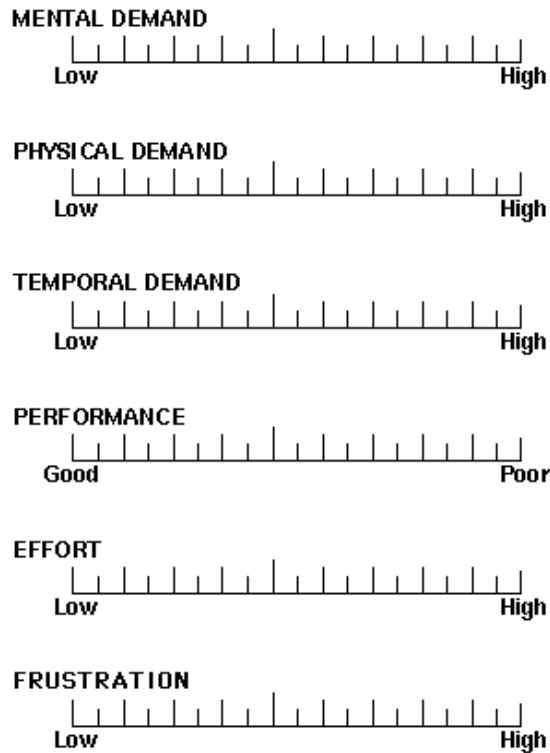
**Table A.3. Piloted test evaluation printed variables**

Variable	No.	Units	Description
time	1	Sec	time
pitch_stic , roll_stick , rudder_ped , throttle	2-5	%	Pilot control inputs
alt, lat, long	6, 72-73	ft, rad, rad	Altitude, latitude and longitude
alt_dot, lat_dot, long_dot	42,36, 39	ft/s	rates of altitude, latitude & long
alt_d_dot, lat_d_dot, long_d_dot	43,37, 40	ft/s/s	acc of altitude, latitude & long
Ias	7	ft/s	indicated air speed
p, q, r	10-12	rad/s	a/c angular velocity
pdot, qdot, rdot	13-15	rad/s/s	a/c angular acceleration
u, v, w	16-18	m/s	a/c linear velocity
udot, vdot, wdot	19-21	m/s/s	a/c linear acceleration
phi, theta, psi	24,23, 22	rad	a/c Euler angles
phi_dot, theta_dot, psi_dot	50,47, 44	rad/s	Euler angle rates
phi_d_dot, theta_d_dot, psi_d_dot	51,48, 45	rad/s/s	Euler angle accelerations
nx_ps, ny_ps, nz_ps	25-27	G	a/c normal force at pilot station
sx, sy, sz	69-71	M	Earth-frame positions (x,y,z) with respect to runway threshold
gs_error	8	deg	Glideslope error
loc_error	9	deg	localizer error
volt_leg_[1:6]	52-57	volts	commanded leg lengths
lin_accel[1:6]	58-63	G	linear accelerometers (six)
pitchincl, rollincl	64,65	deg	inclinometer angles (roll and pitch)
u_gust, v_gust, w_gust	66-68		Gust – magnitude & direction
phic, thec, psic, altc, latc, longc	49,46,30, 41,35,38	rad	Compensated aircraft state
Tdelay	29	sec	added delay
Icom	28		predictor type

Sampling Rate: 62.5 Hz Data Sampling Rate: 15.6 Hz (Every 4th Frame)

**Table A.4. NASA TLX Subscale rating definitions (from Telban’s thesis)**

Title	Endpoints	Descriptions
<b>MENTAL DEMAND</b>	Low/High	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
<b>PHYSICAL DEMAND</b>	Low/High	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
<b>TEMPORAL DEMAND</b>	Low/High	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
<b>EFFORT</b>	Low/High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
<b>PERFORMANCE</b>	Good/Poor	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
<b>FRUSTRATION LEVEL</b>	Low/High	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?



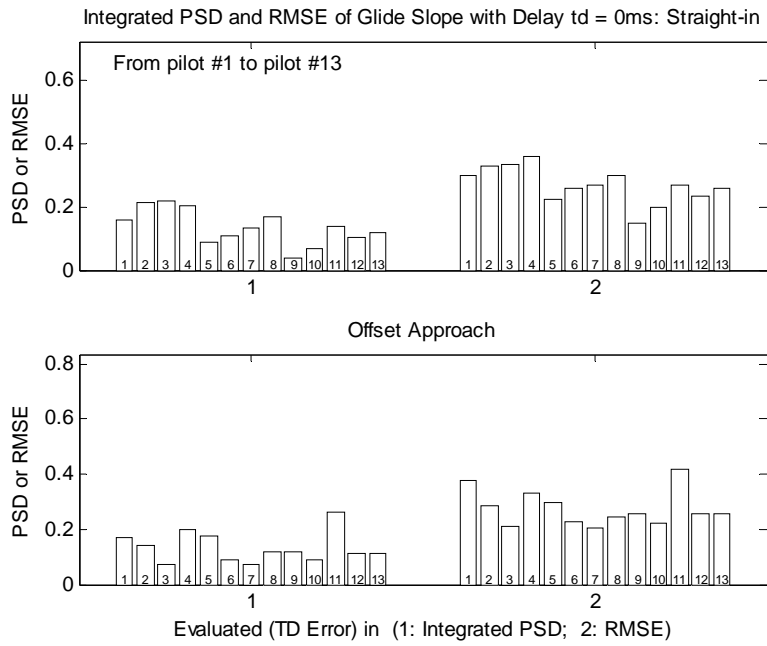
**Fig. A.1. NASA TLX Subscale data sheet (from Telban’s thesis)**

**Table A.5. Resumes of the pilot subjects**

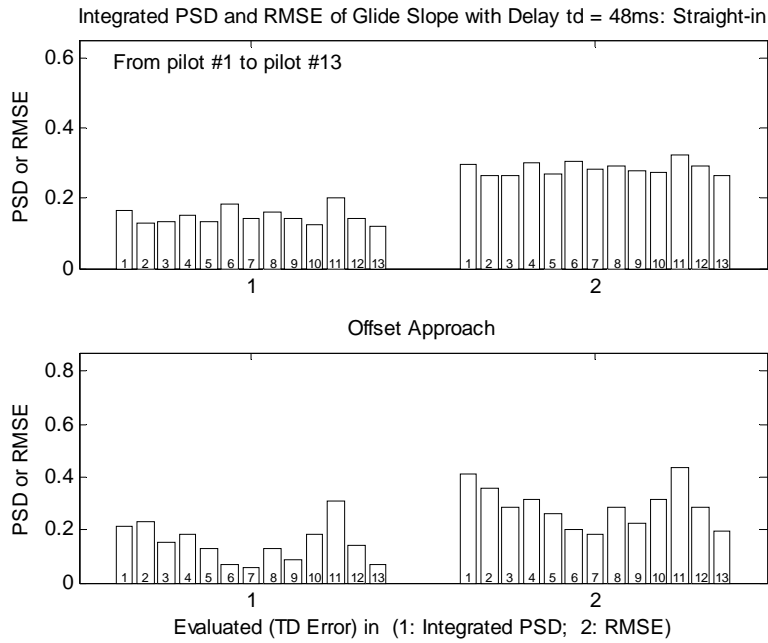
<b>Pilot</b>	<b>Category</b>	<b>Description</b>	<b>Experience</b>
#1	Simulator Engineer/Pilot	Various simulators including B-757, B-737, DC-9, L1011, various rotorcraft, general aviation, and fighters	35 years
#4	Simulator Engineer/Pilot	Private Pilot, various simulators including B-757, B-737, DC-9, general aviation, etc.	30 years
#5	Simulator Engineer/Pilot	Private Pilot, various simulators including B-757, B-737, DC-9, general aviation, space shuttle, HL-20, etc.	35 years
#2	Research Pilot	Military Aircraft (200h), Transport Aircraft (750h), Corporate Aircraft (1200h), General Aviation (2700h), Helicopter (600h)	6,000 hours
#3	Research Pilot	Military Aircraft (1275h), Transport Aircraft (5350h), Corporate Aircraft (3650h)	10,275 hours
#7	Research Pilot	Military Aircraft (3300h), Transport Aircraft (350h), Corporate Aircraft (3800h), Other Aircraft (600h)	8,000 hour
#8	Corporate Pilot	Military Aircraft (5800h), Helicopter (2800h)	9,000 hours
#10	Corporate Pilot	Transport Aircraft (3400h), Corporate Aircraft (5800h), General Aviation (1200h)	10,000 hours
#6	FAA Pilot	Military Aircraft (1950h), Transport Aircraft (14500h), Corporate Aircraft (360h)	16,000 hours
#9	Commercial Pilot	Military Aircraft (2320h), Transport Aircraft (16425h), Various Aircraft (400h)	19,100 hours
#11	Commercial Pilot	Military Aircraft (3950h), Transport Aircraft (26h)	4,000 hours
#12	Commercial Pilot	Military Aircraft (3000h), Transport Aircraft (2500h)	5,500 hours
#13	Commercial Pilot	Military Aircraft (3500h), Transport Aircraft (11400h)	14,900 hours

## Appendix B. More Results of the Piloted Tests

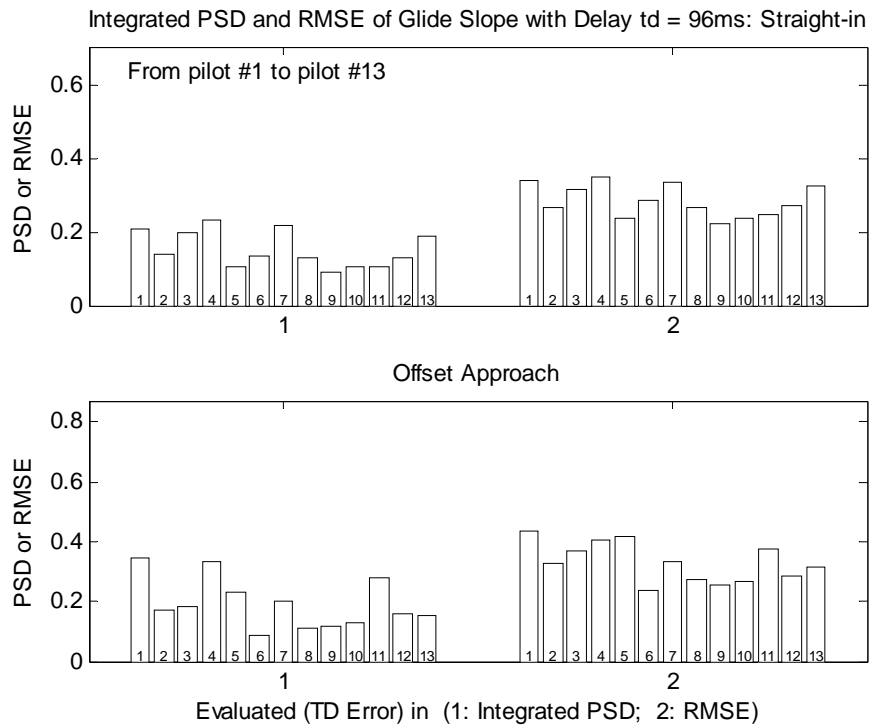
### B.1. Glideslope Error



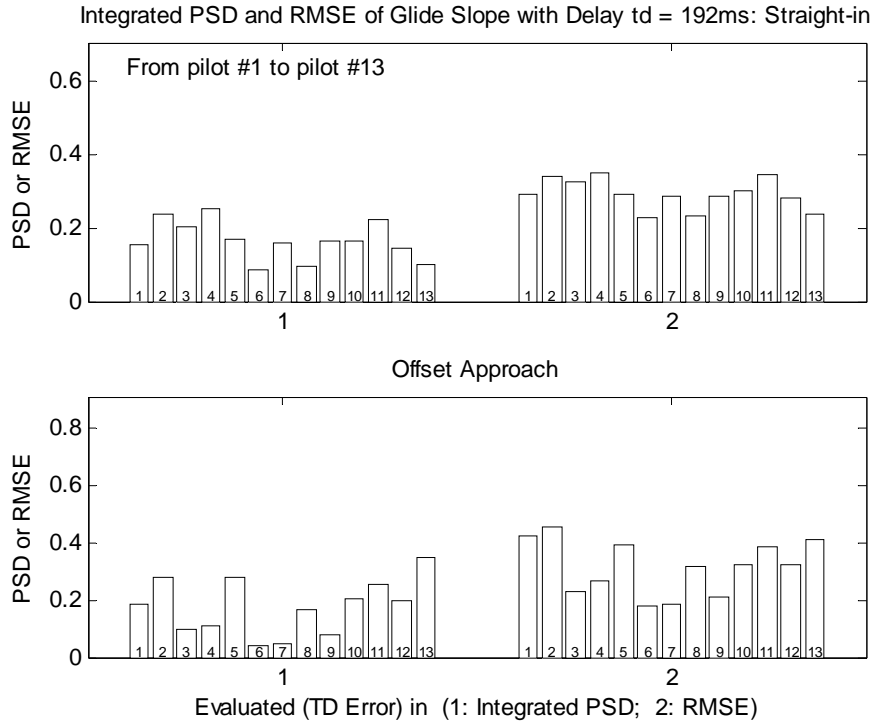
**Fig. B.1.1. Integrated PSD & RMSE of GSE with  $t_d=0$  ms (No compensation)**



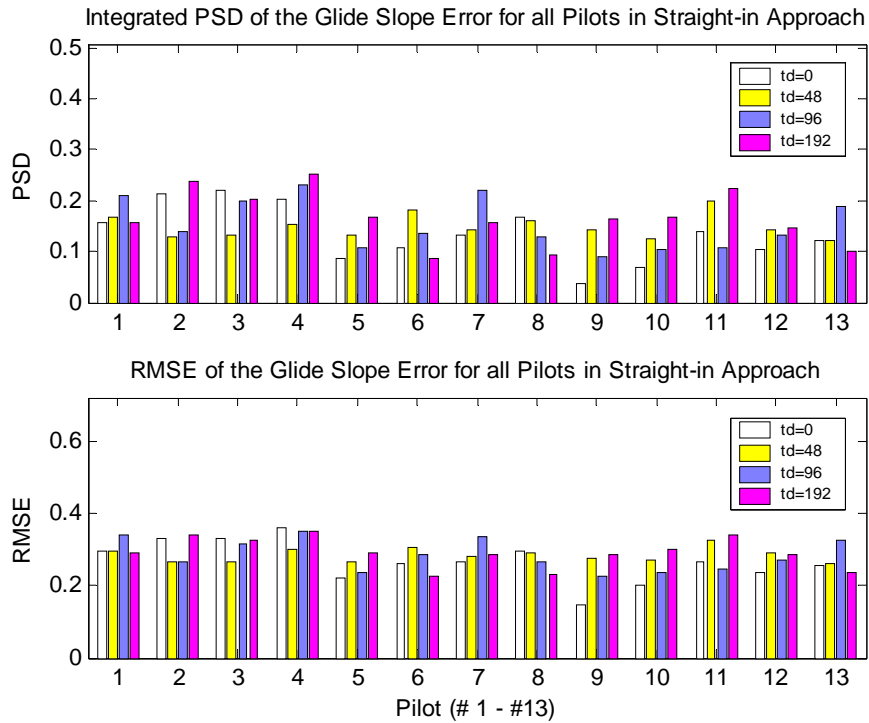
**Fig. B.1.2. Integrated PSD & RMSE of GSE with  $t_d=48$  ms (No compensation)**



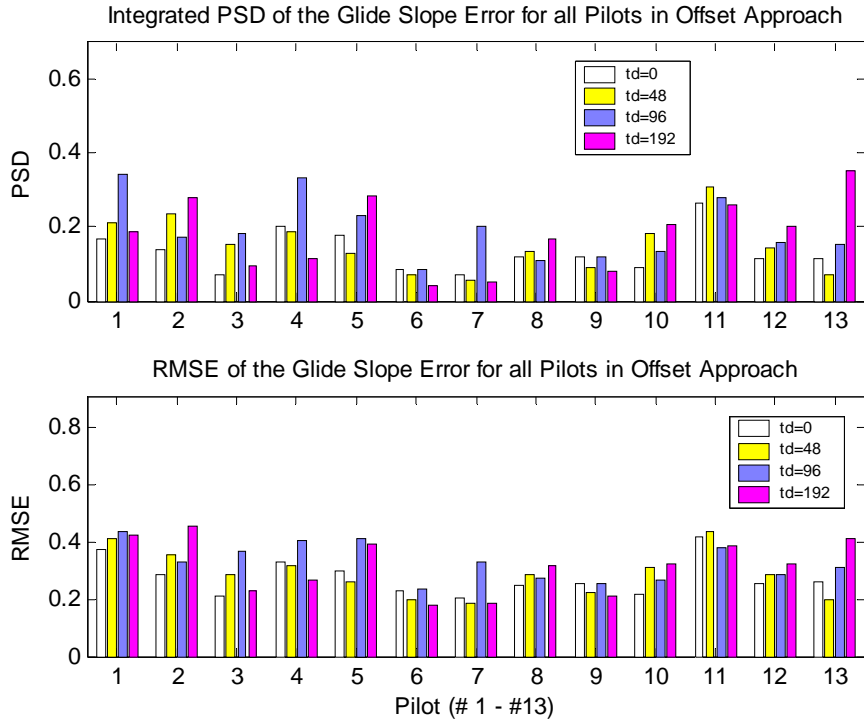
**Fig. B.1.3. Integrated PSD & RMSE of GSE with  $t_d=96$  ms (No compensation)**



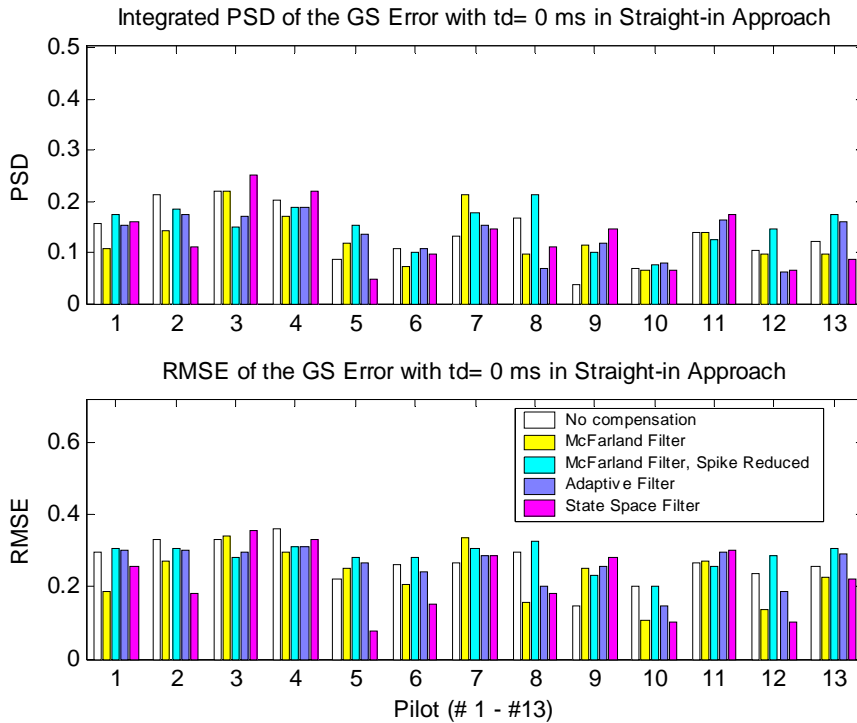
**Fig. B.1.4. Integrated PSD & RMSE of GSE with  $t_d=192$  ms (No compensation)**



**Fig. B.1.5. GSE of Different Time Delay for 13 Pilots: Straight-in Approach**

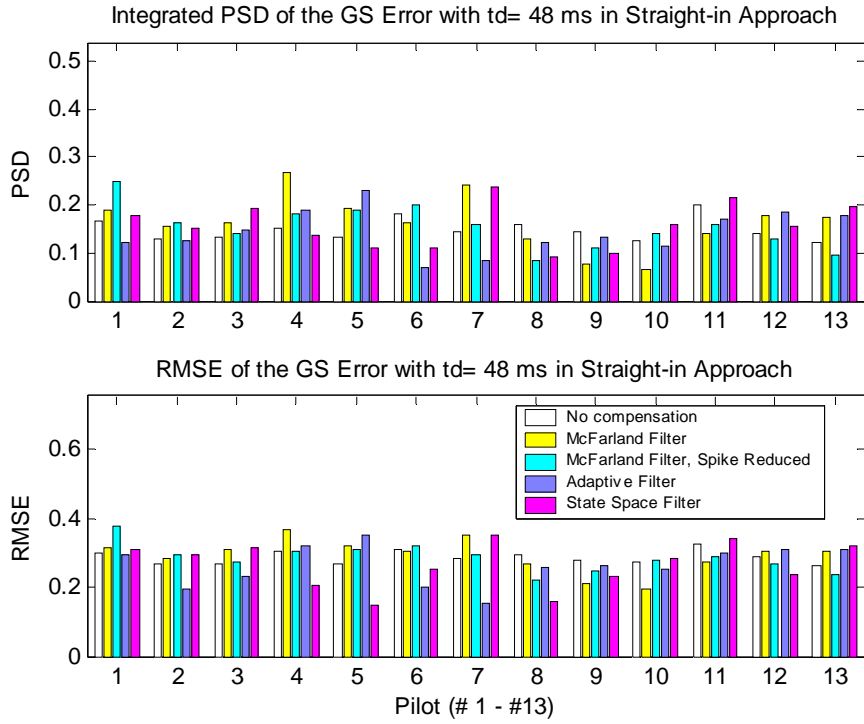


**Fig. B.1.6. GSE of Different Time Delay for 13 Pilots: Offset Approach**

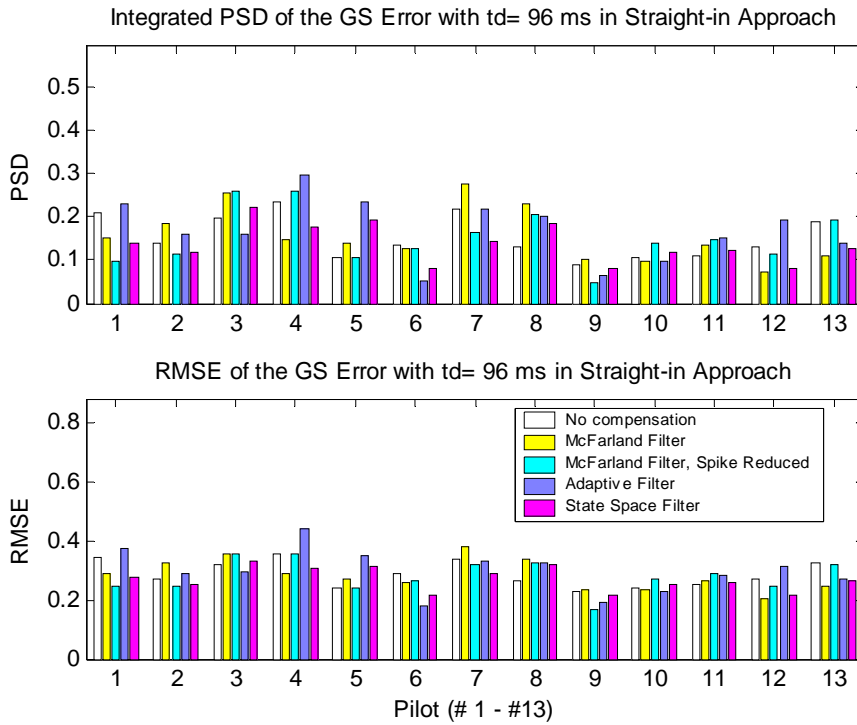


**Fig. B.1.7. Compensation Effects to GSE by Individual Pilot:  $t_d = 0$  ms, Straight-in**

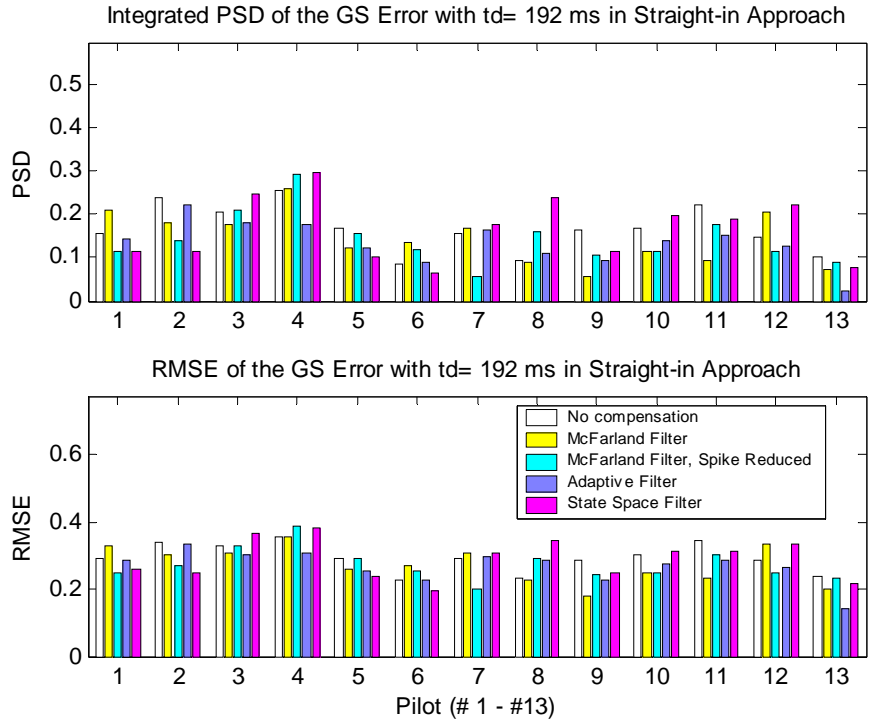




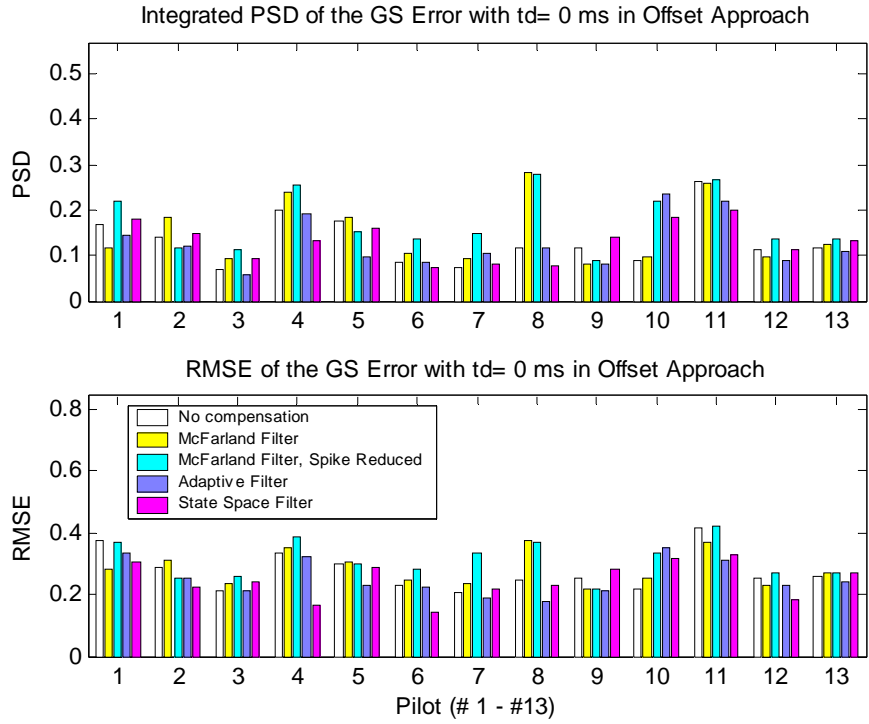
**Fig. B.1.8. Compensation Effects to GSE by Individual Pilot:  $t_d=48$  ms, Straight-in**



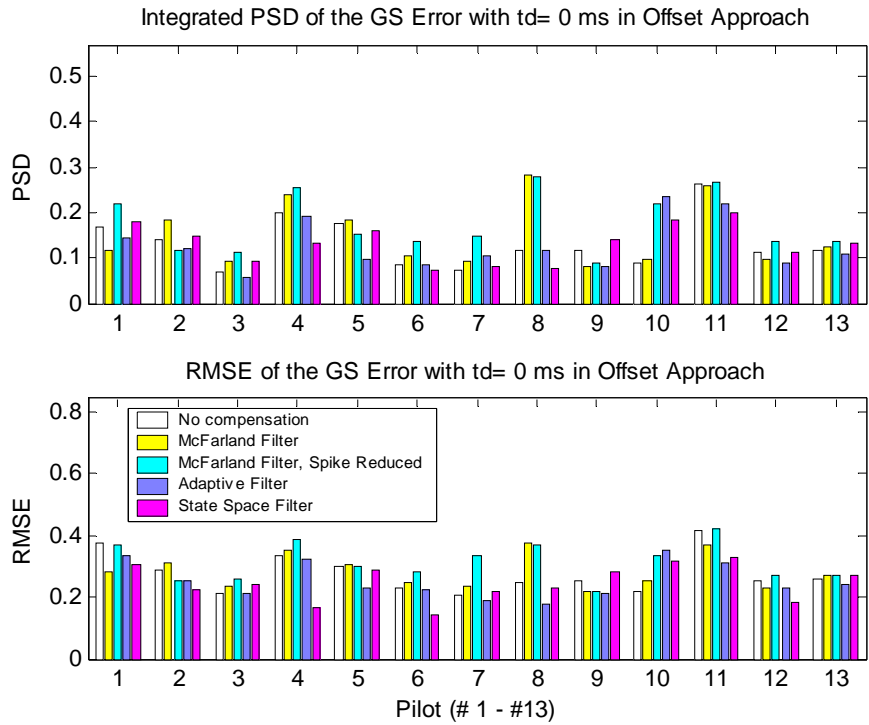
**Fig. B.1.9. Compensation Effects to GSE by Individual Pilot:  $t_d=96$  ms, Straight-in**



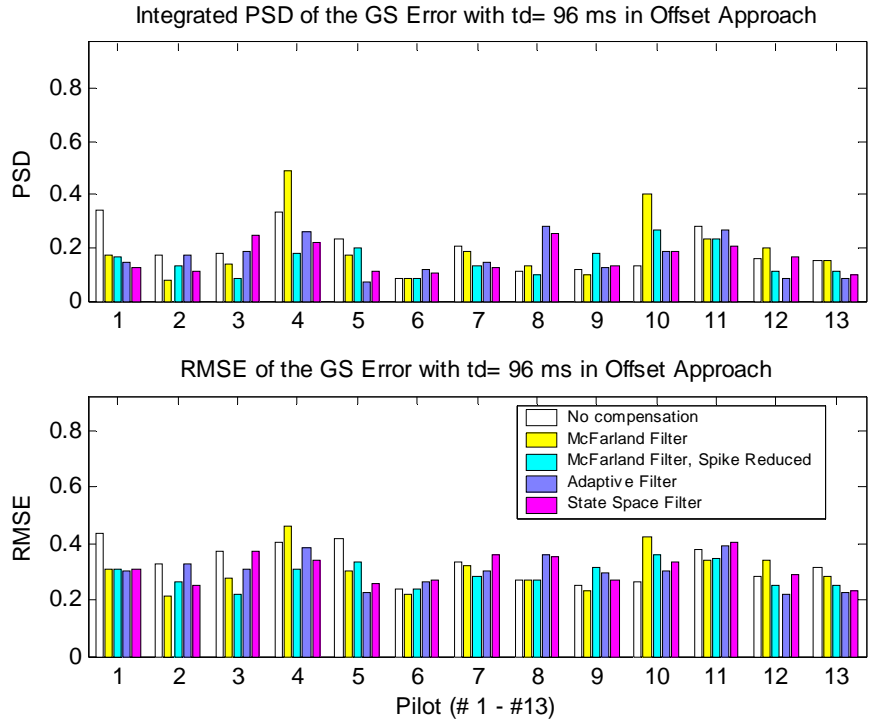
**Fig. B.1.10. Compensation Effects to GSE by Individual Pilot:  $t_d=192$ ms, Straight-in**



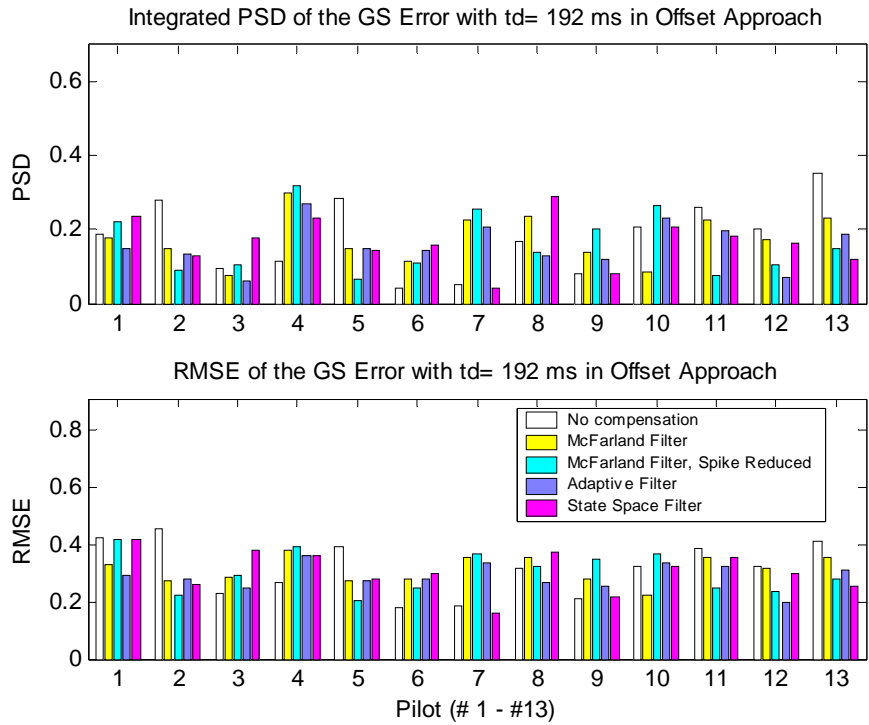
**Fig. B.1.11. Compensation Effects to GSE by Individual Pilot:  $t_d=0$  ms, Offset**



**Fig. B.1.12. Compensation Effects to GSE by Individual Pilot:  $t_d=48$  ms, Offset**

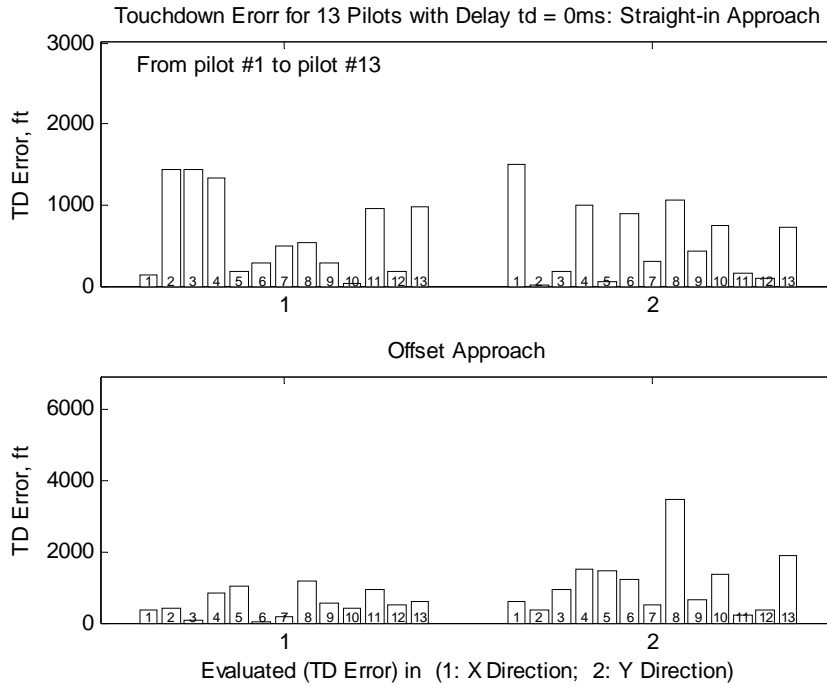


**Fig. B.1.13. Compensation Effects to GSE by Individual Pilot:  $t_d=96$  ms, Offset**

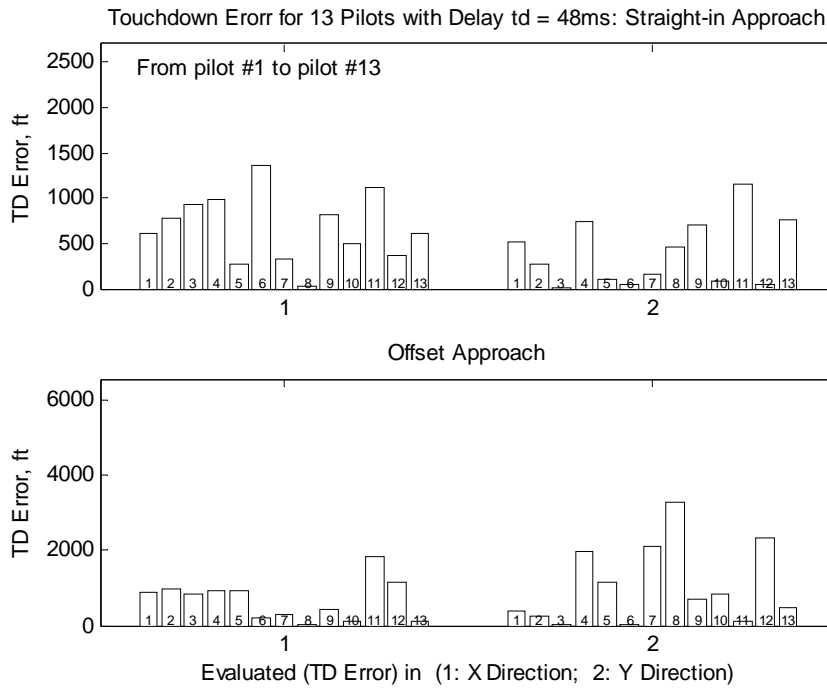


**Fig. B.1.14. Compensation Effects to GSE by Individual Pilot:  $t_d=192$  ms, Offset**

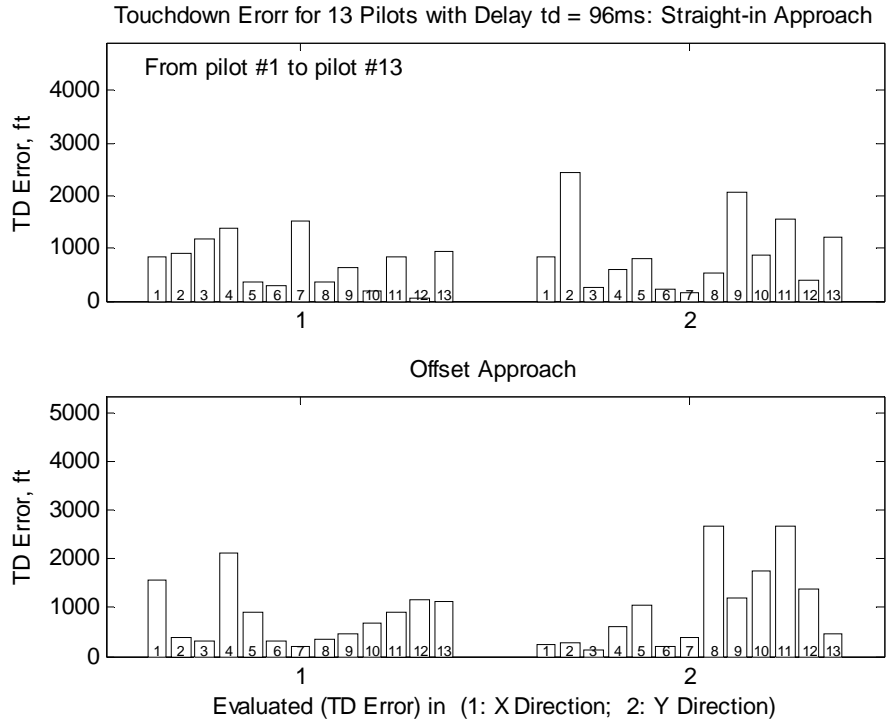
## B.2. Glideslope Error



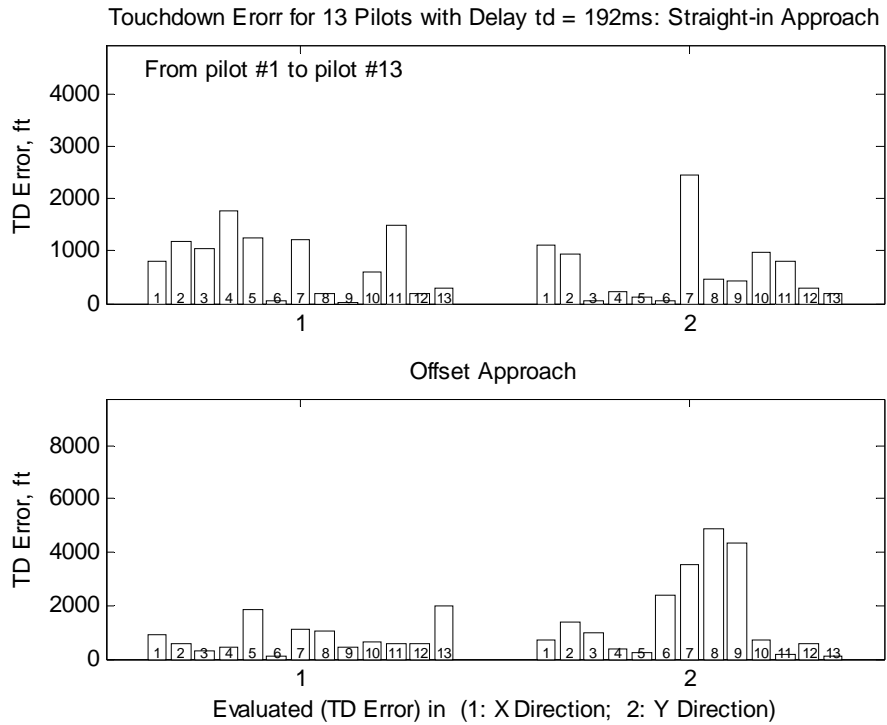
**Fig. B.2.1. TDE for 13 Pilots with  $t_d=0\text{ ms}$  (No compensation)**



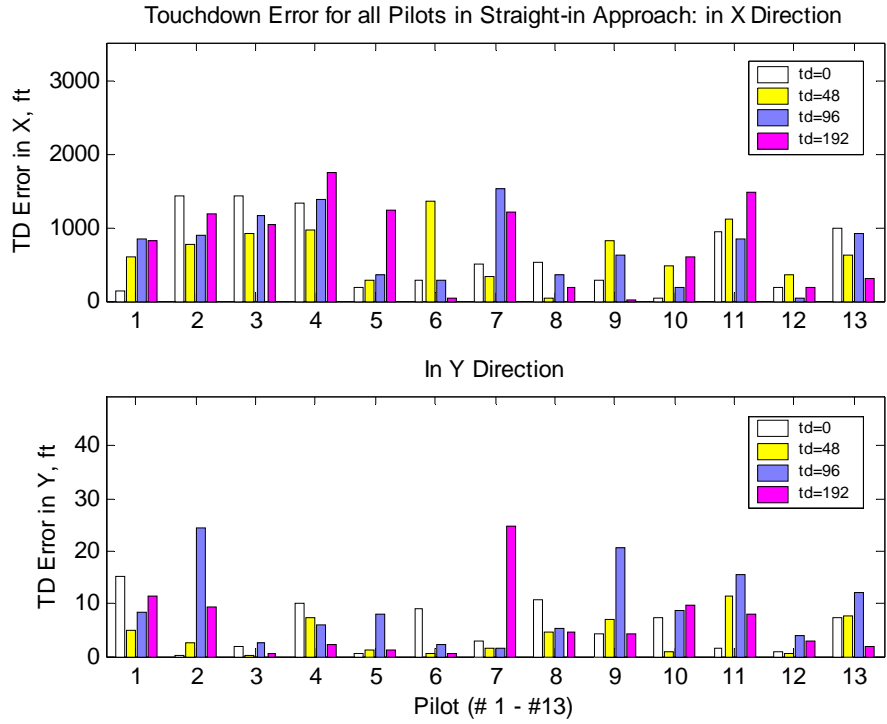
**Fig. B.2.2. TDE for 13 Pilots with  $t_d=48\text{ ms}$  (No compensation)**



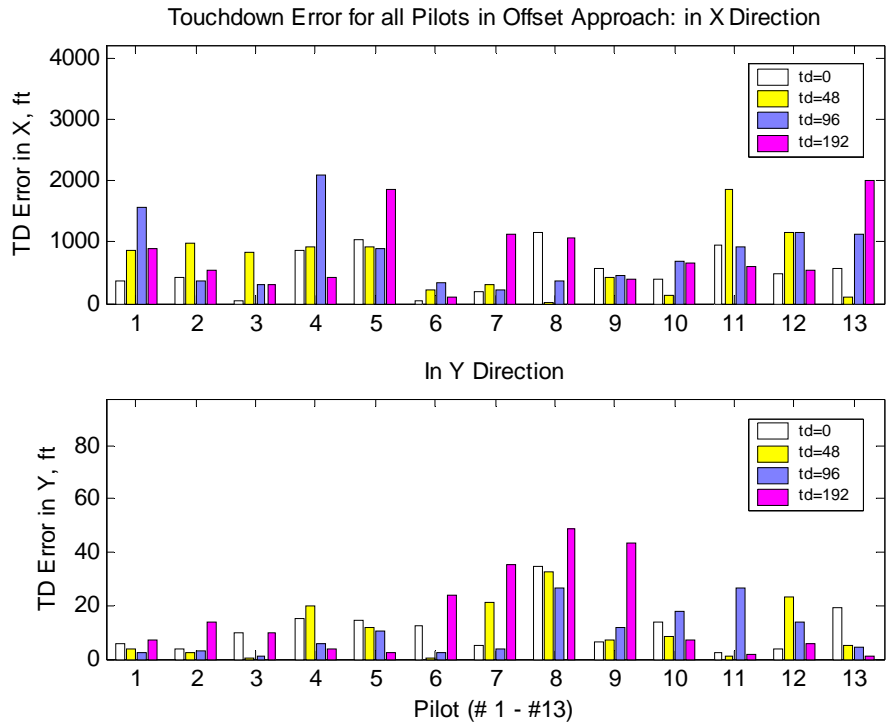
**Fig. B.2.3. TDE for 13 Pilots with  $t_d=96\text{ ms}$  (No compensation)**



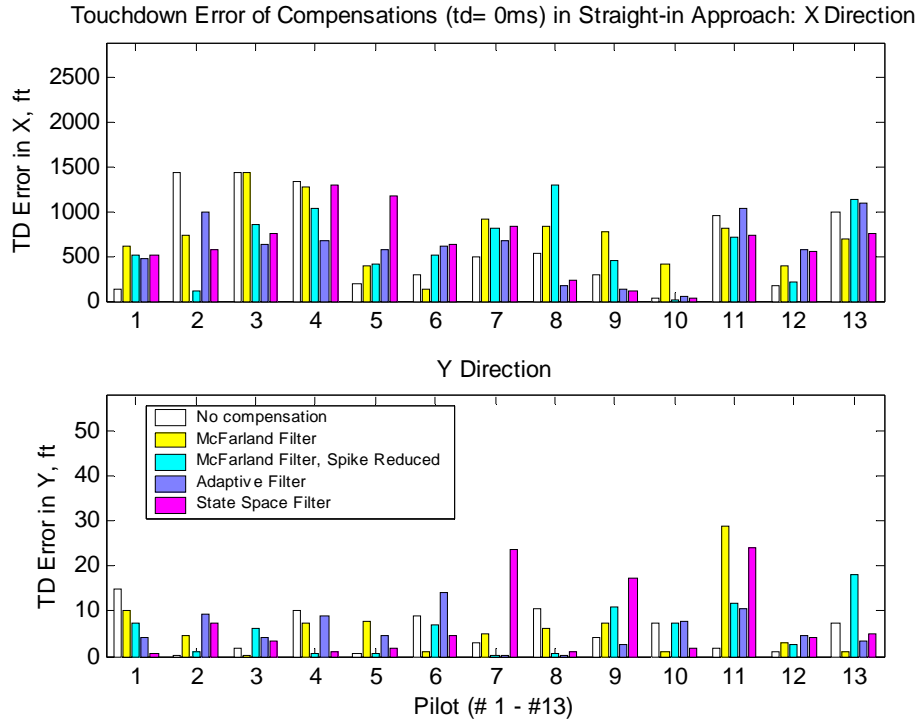
**Fig. B.2.4. TDE for 13 Pilots with  $t_d=192\text{ ms}$  (No compensation)**



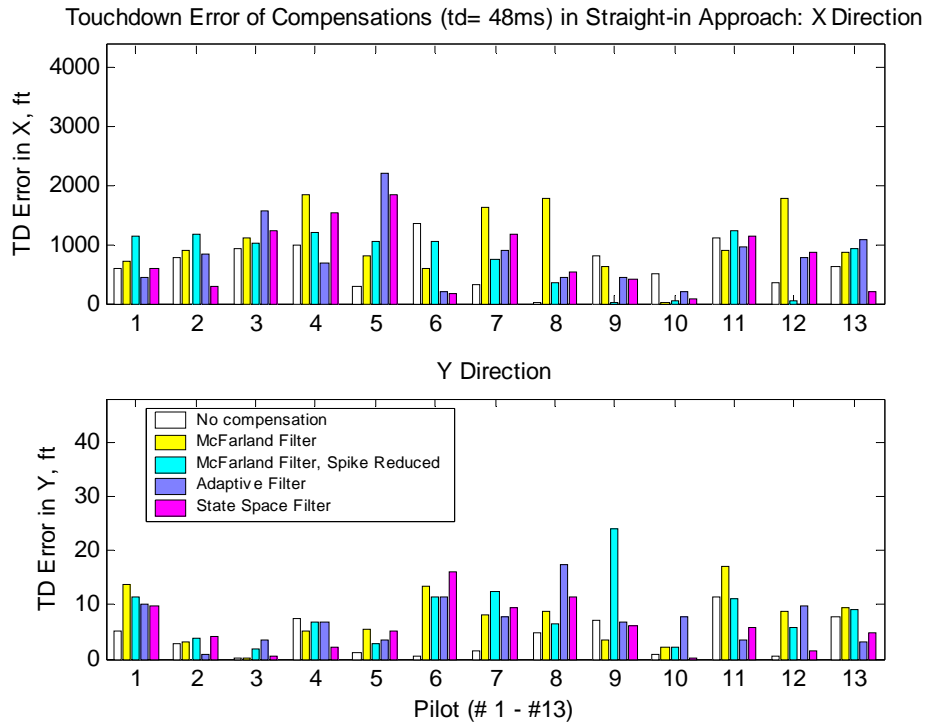
**Fig. B.2.5. TDE of Different Time Delays for 13 Pilots: Straight-in Approach**



**Fig. B.2.6. TDE of Different Time Delays for 13 Pilots: Offset Approach**

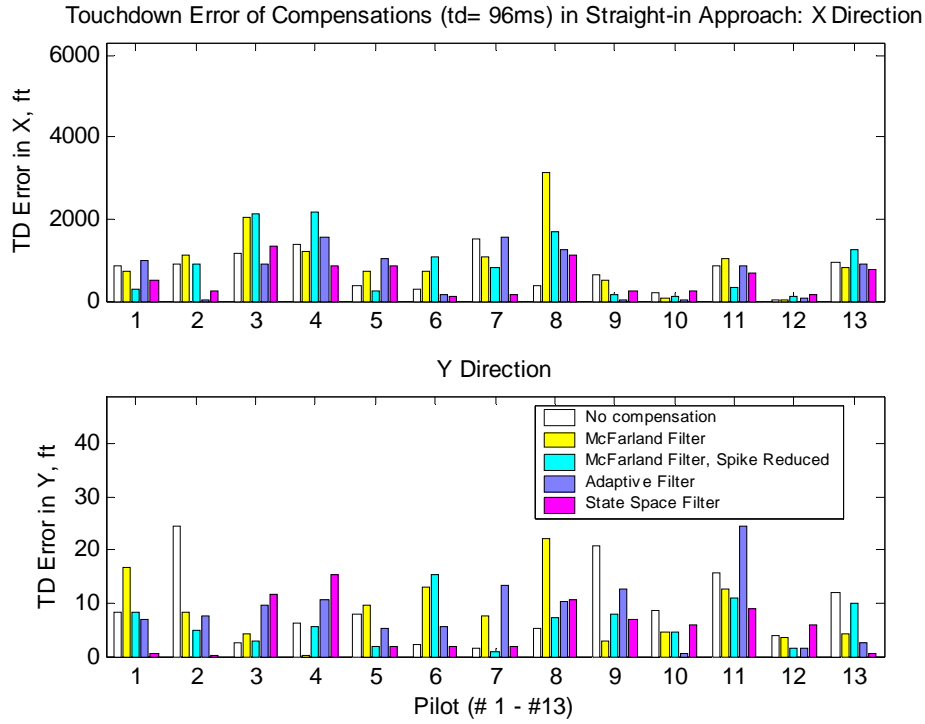


**Fig. B.2.7. Compensation Effects to TDE by Individual Pilot:  $t_d=0$  ms, Straight-in**

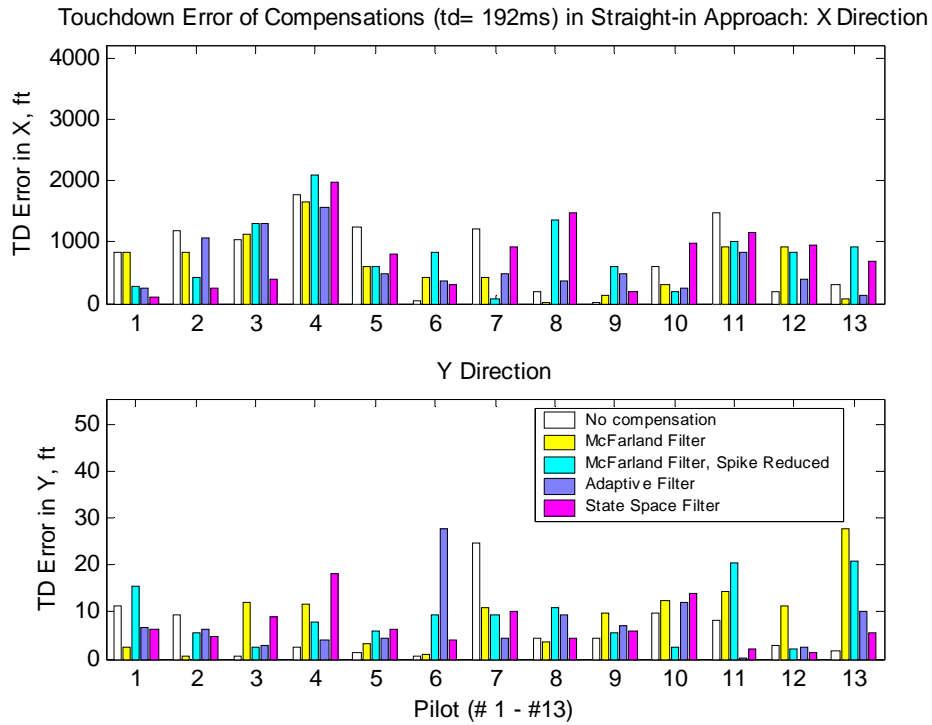


**Fig. B.2.8. Compensation Effects to TDE by Individual Pilot:  $t_d=48$  ms, Straight-in**

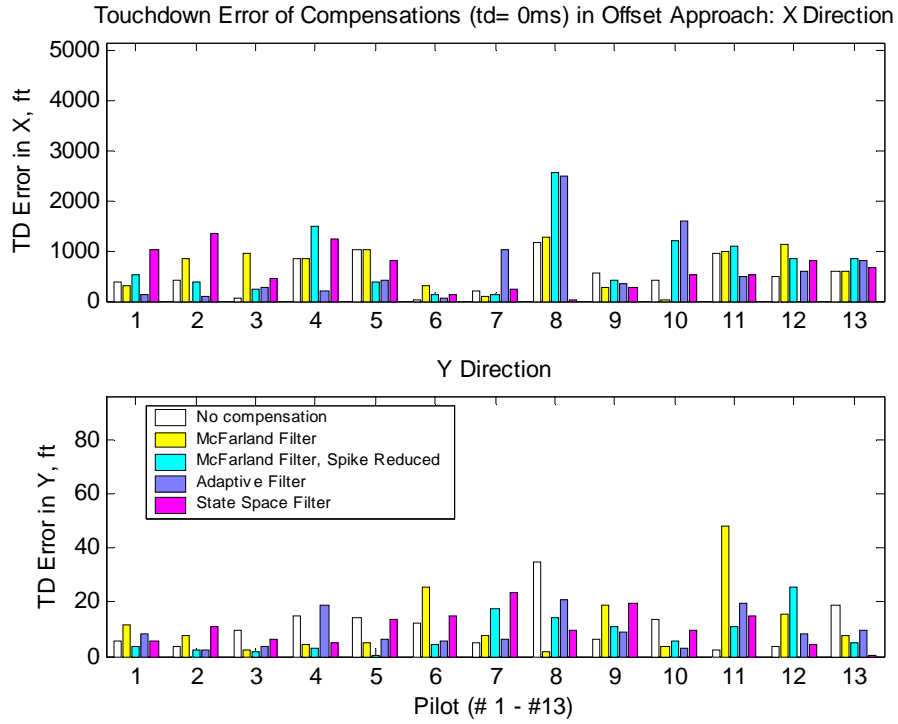




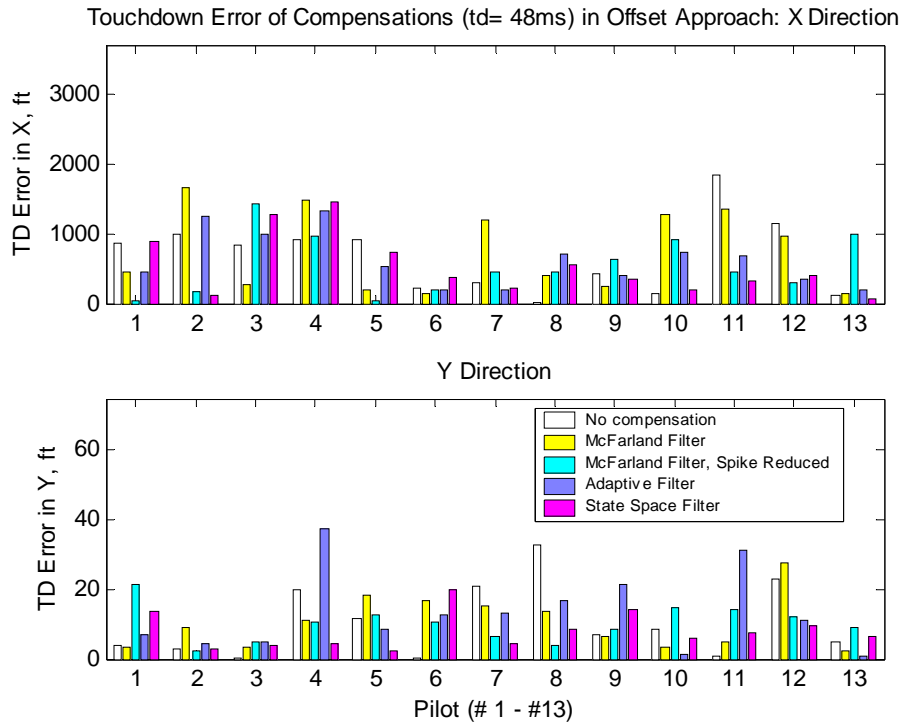
**Fig. B.2.9. Compensation Effects to GSE by Individual Pilot:  $t_d=96$  ms, Straight-in**



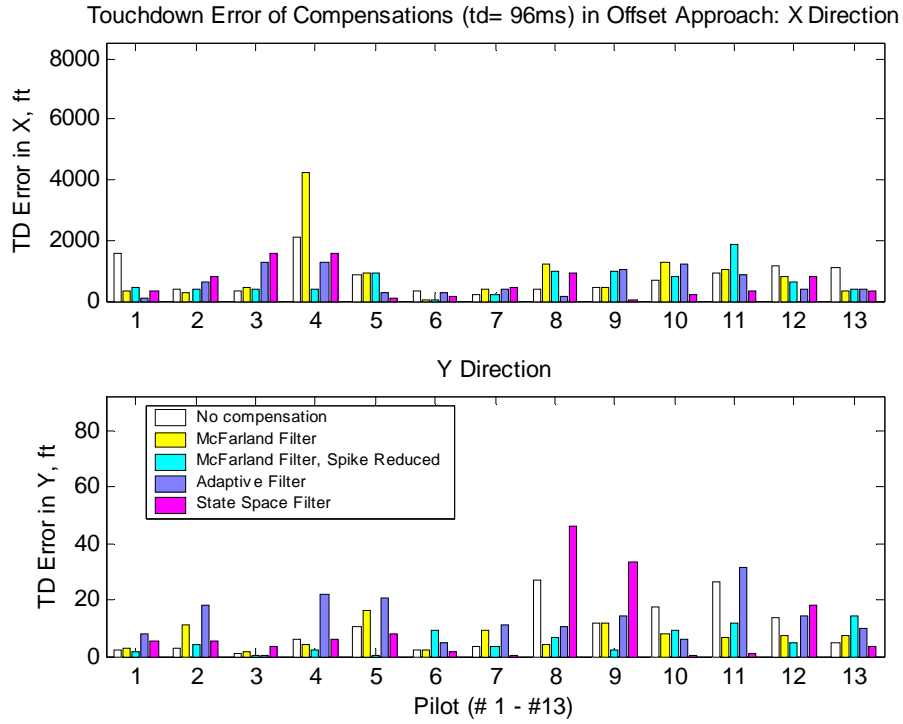
**Fig. B.2.10. Compensation Effects to TDE by Individual Pilot:  $t_d=192$ ms, Straight-in**



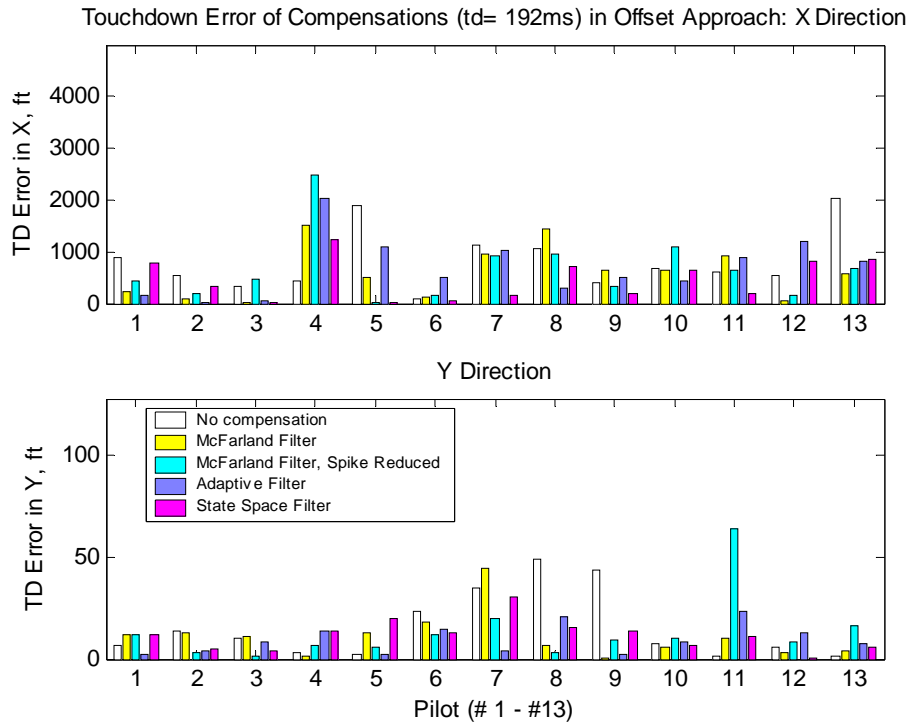
**Fig. B.2.11. Compensation Effects to TDE by Individual Pilot:  $t_d=0$  ms, Offset**



**Fig. B.2.12. Compensation Effects to TDE by Individual Pilot:  $t_d=48$  ms, Offset**

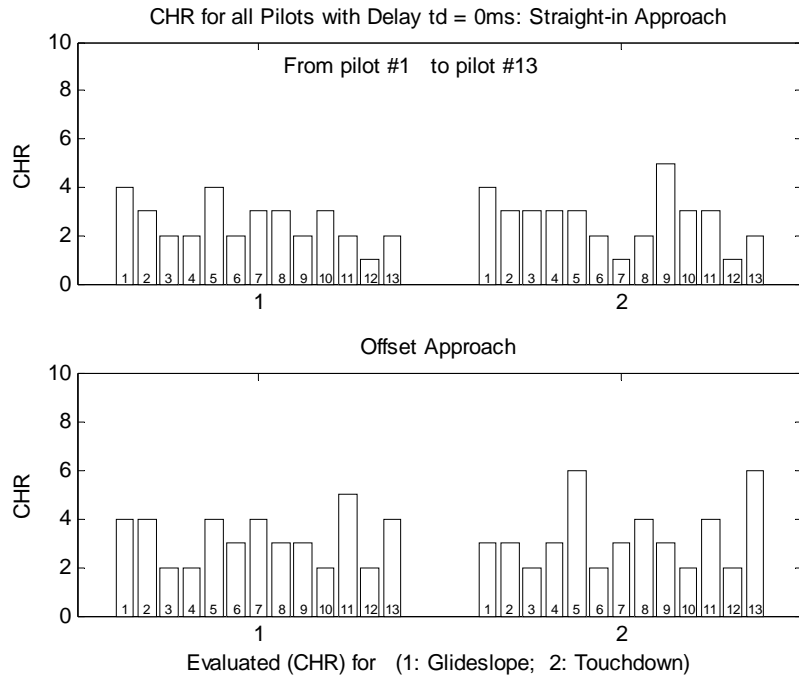


**Fig. B.2.13. Compensation Effects to TDE by Individual Pilot:  $t_d=96$  ms, Offset**

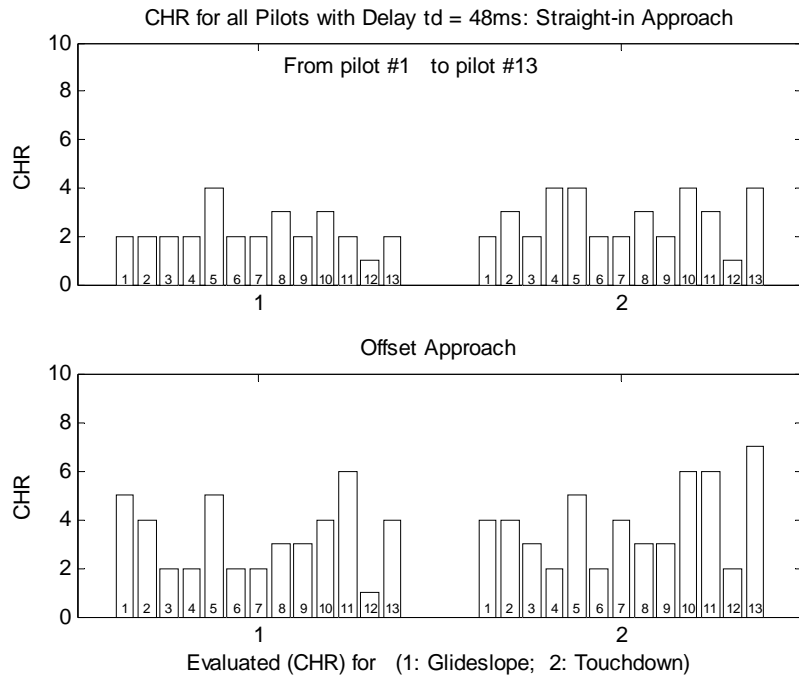


**Fig. B.2.14. Compensation Effects to TDE by Individual Pilot:  $t_d=192$ ms, Offset**

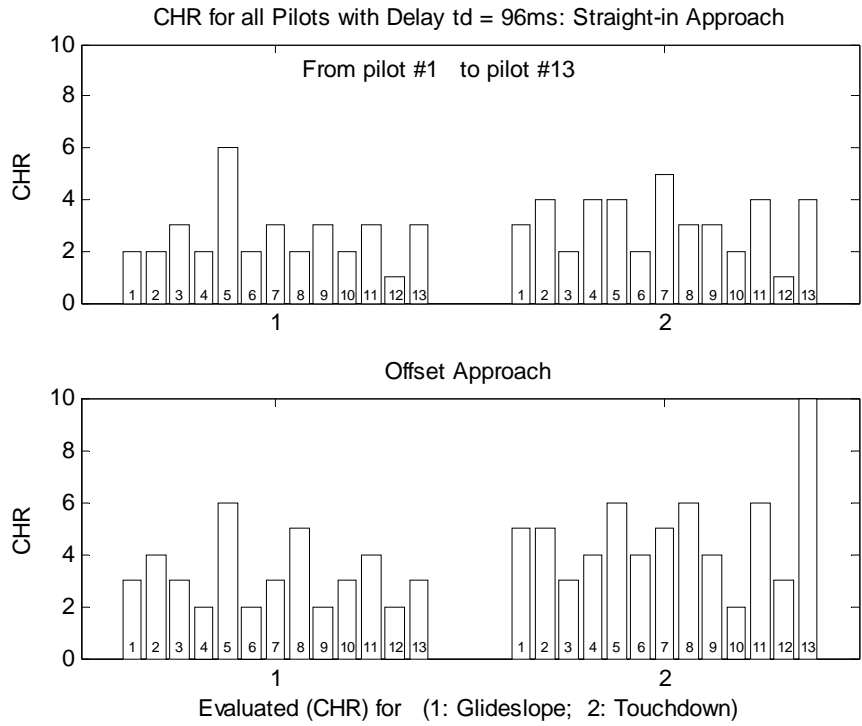
### B.3. Cooper-Harper Rating



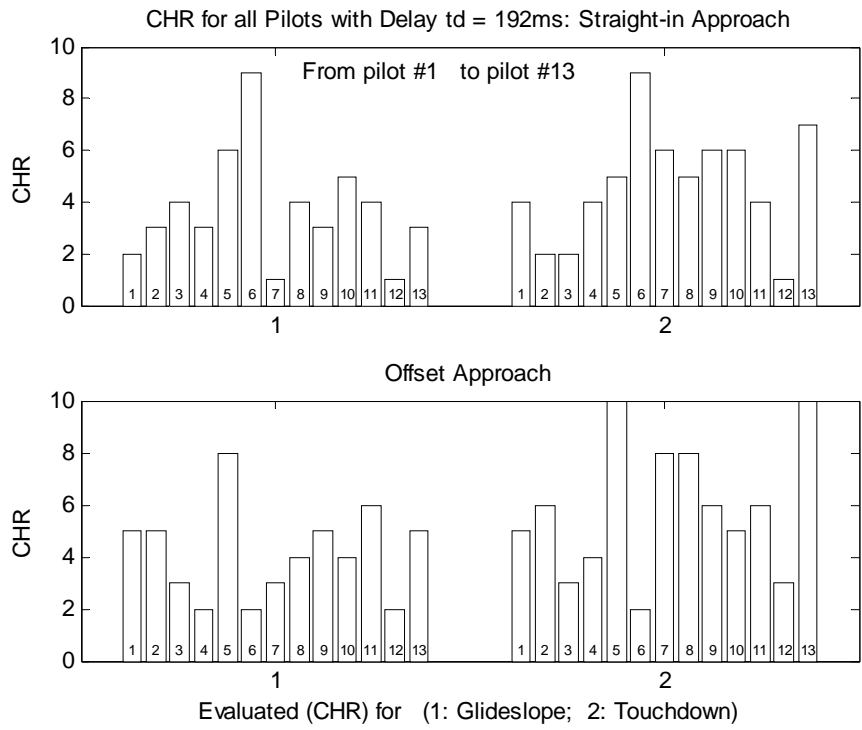
**Fig. B.3.1. CHR for 13 pilots with  $t_d=0$  ms (no compensation)**



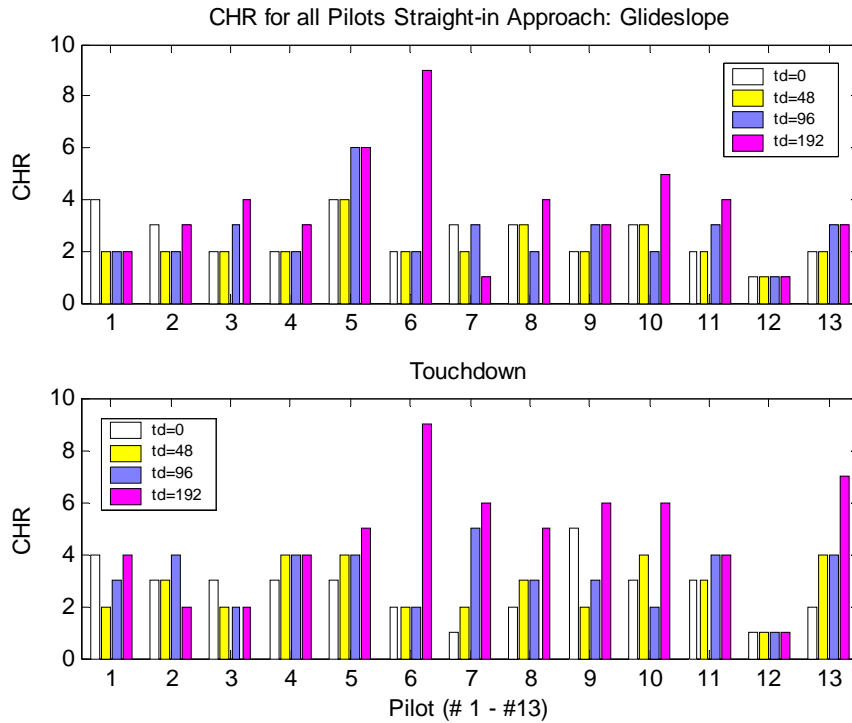
**Fig. B.3.2. CHR for 13 pilots with  $t_d=48$  ms (no compensation)**



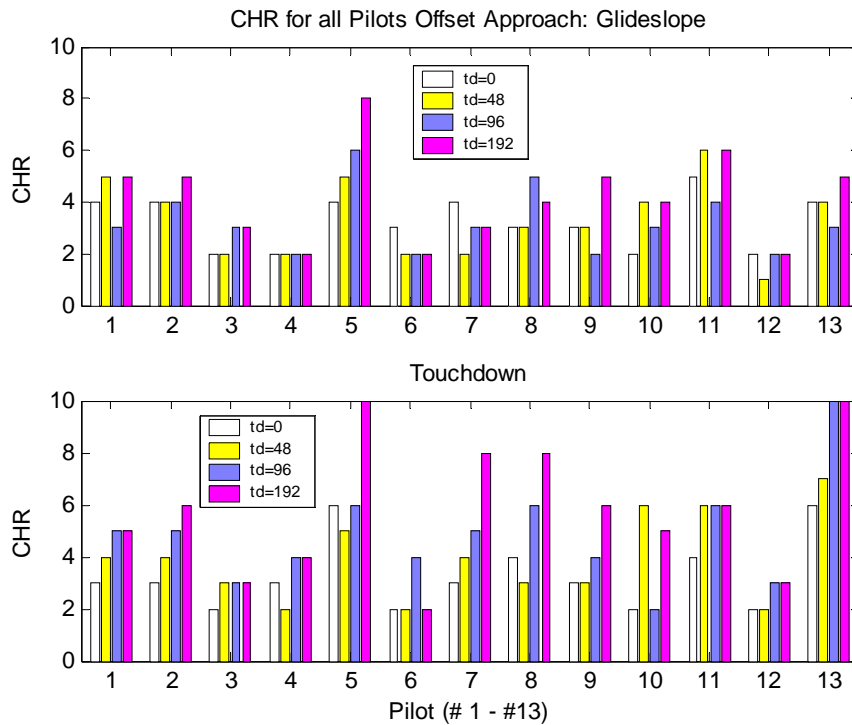
**Fig. B.3.3. CHR for 13 pilots with  $t_d=96\text{ ms}$  (no compensation)**



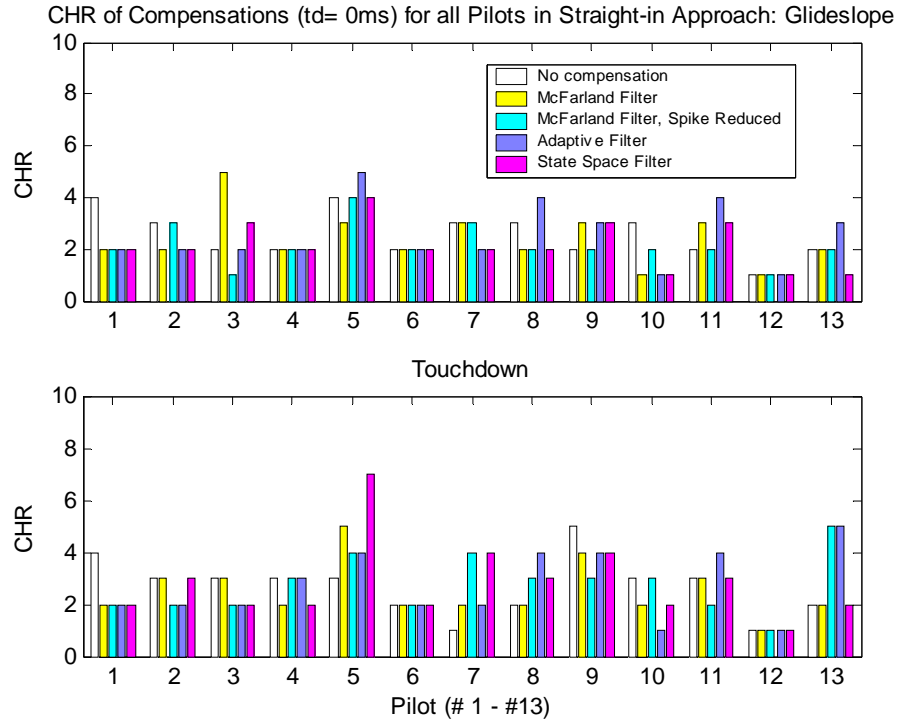
**Fig. B.3.4. CHR for 13 pilots with  $t_d=192\text{ ms}$  (no compensation)**



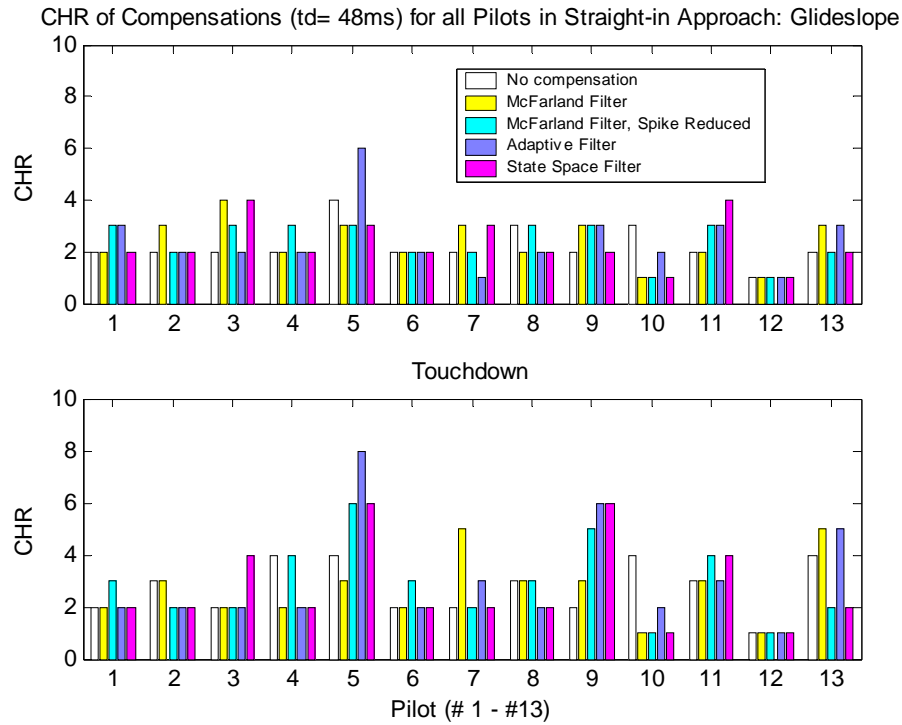
**Fig. B.3.5. CHR of different time delays for 13 pilots: straight-in approach**



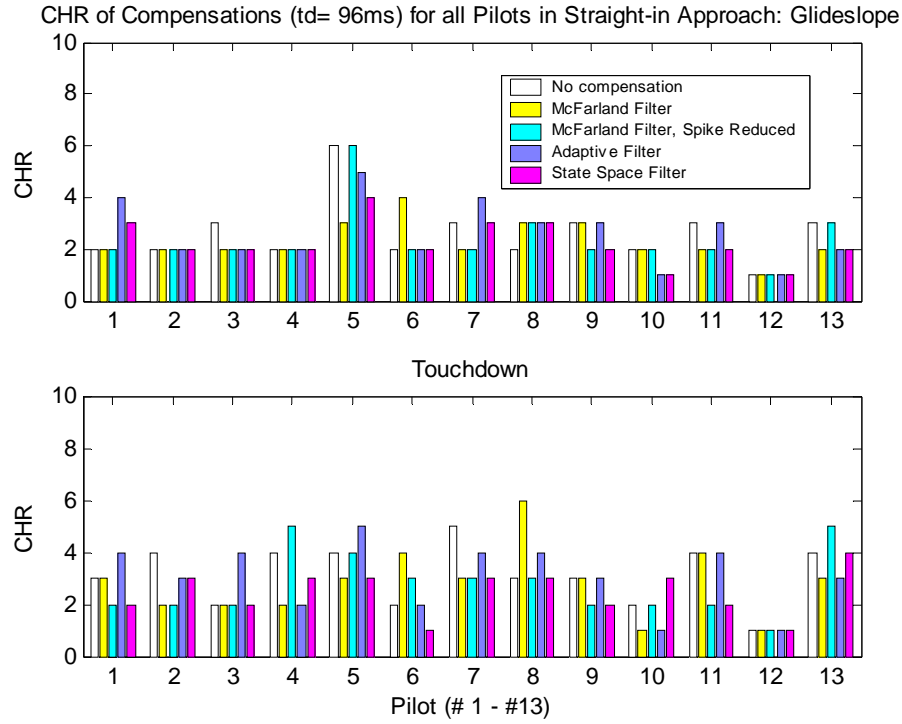
**Fig. B.1.6. CHR of different time delays for 13 pilots: offset approach**



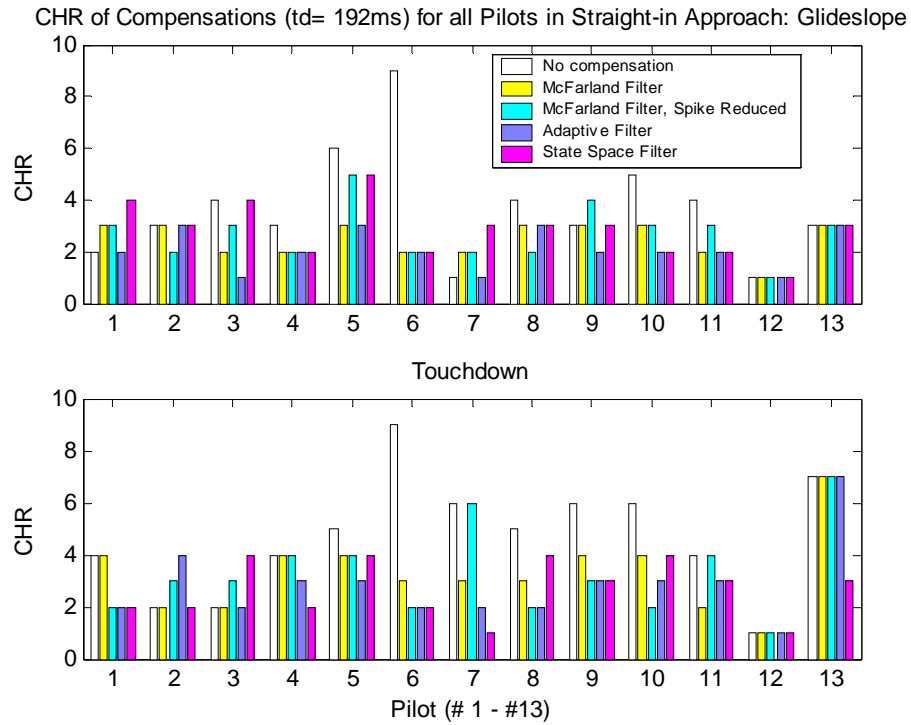
**Fig. B.3.7. Compensation effects to CHR by individual pilot:  $t_d=0$  ms, straight-in**



**Fig. B.3.8. Compensation effects to CHR by individual pilot:  $t_d=48$  ms, straight-in**



**Fig. B.3.9. Compensation effects to CHR by individual pilot:  $t_d=96$  ms, straight-in**



**Fig. B.3.10. Compensation effect to CHR by individual pilot:  $t_d=192$ ms, straight-in**



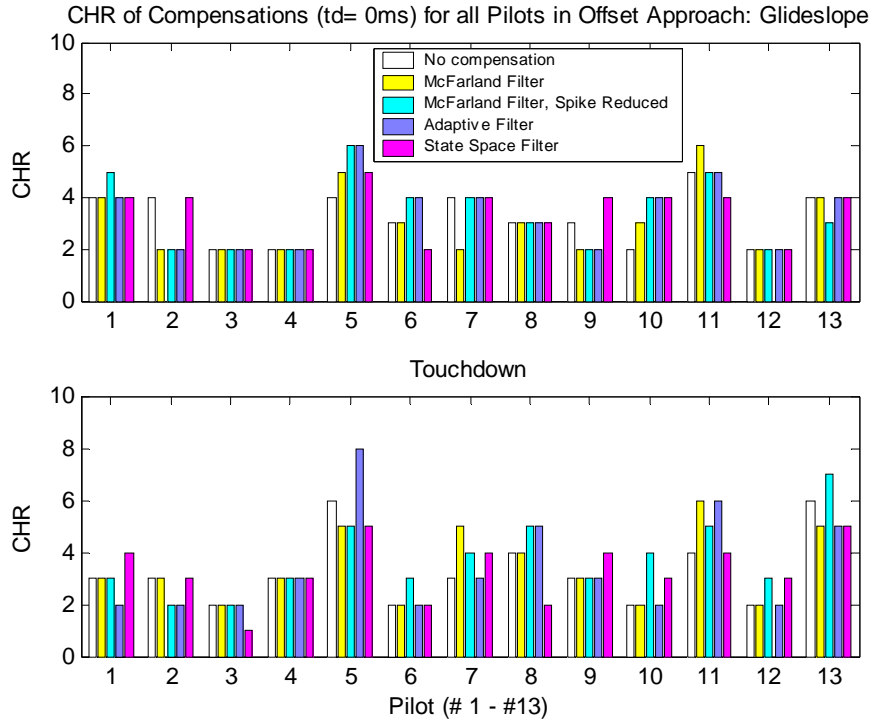


Fig. B.3.11. Compensation effect to CHR by individual pilot:  $t_d=0$  ms, offset

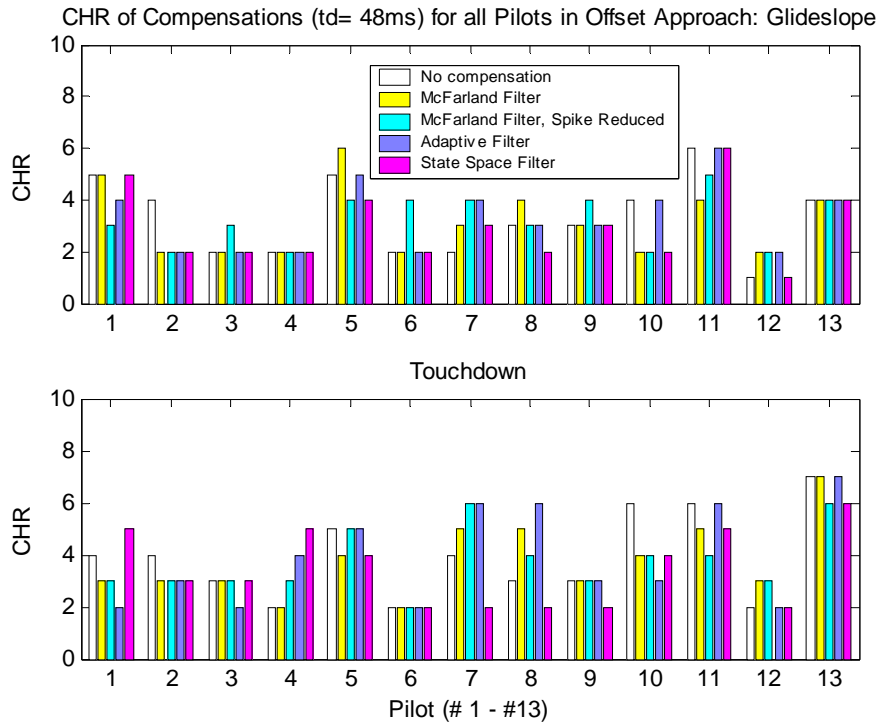
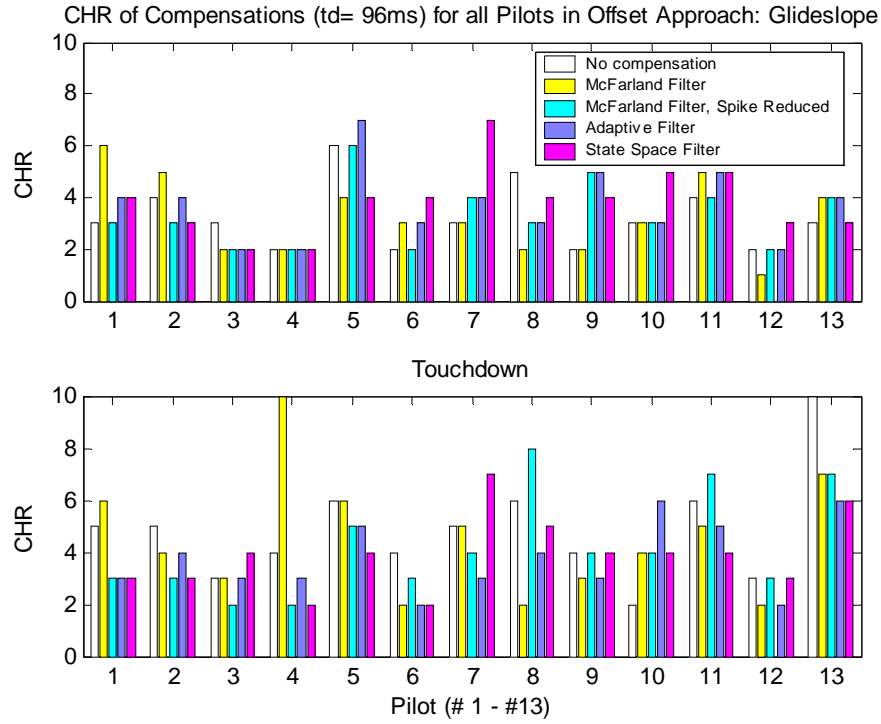
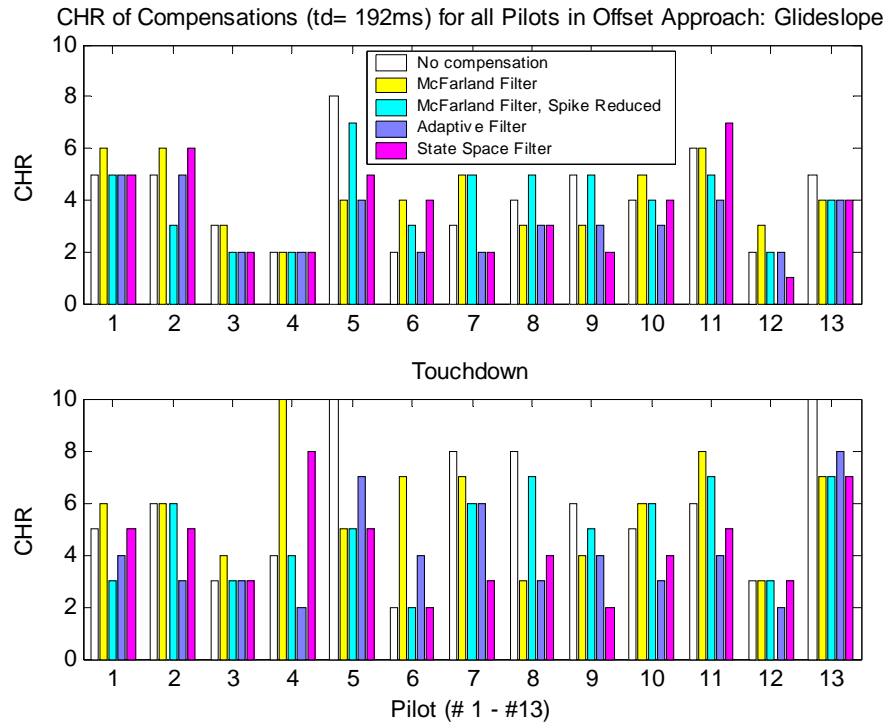


Fig. B.3.12. Compensation effect to CHR by individual pilot:  $t_d=48$  ms, offset

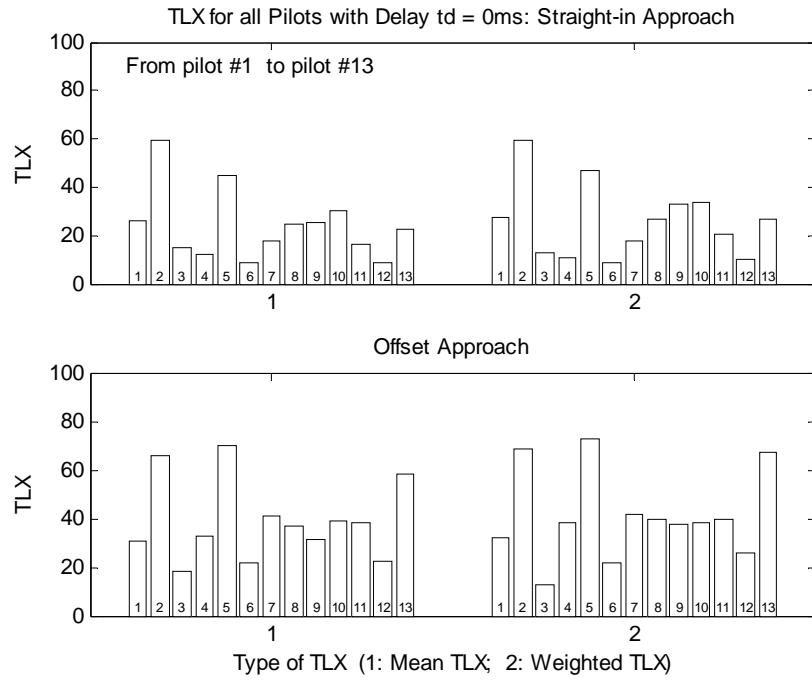


**Fig. B.3.13. Compensation effect to CHR by individual pilot:  $t_d=96$  ms, offset**

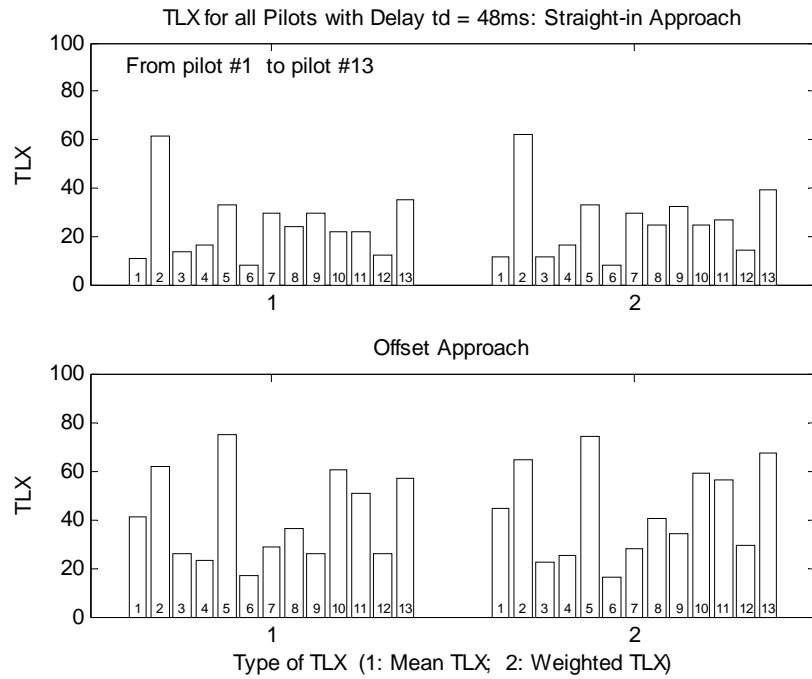


**Fig. B.3.14. Compensation effect to CHR by individual pilot:  $t_d=192$  ms, offset**

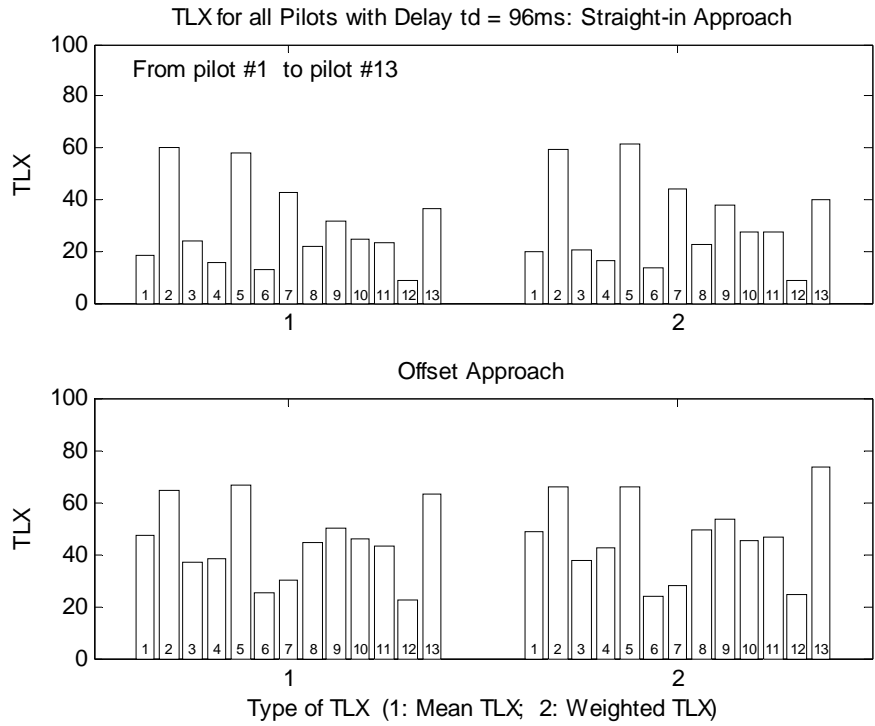
## B.4. Task Load Index



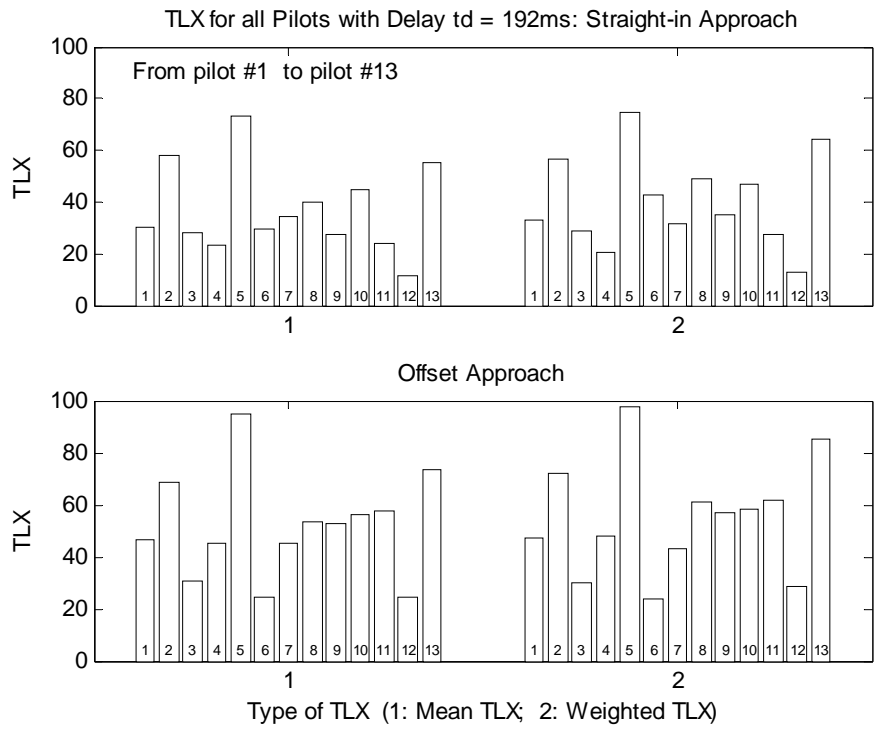
**Fig. B.4.1. TLX for 13 pilots with  $t_d=0$  ms (no compensation)**



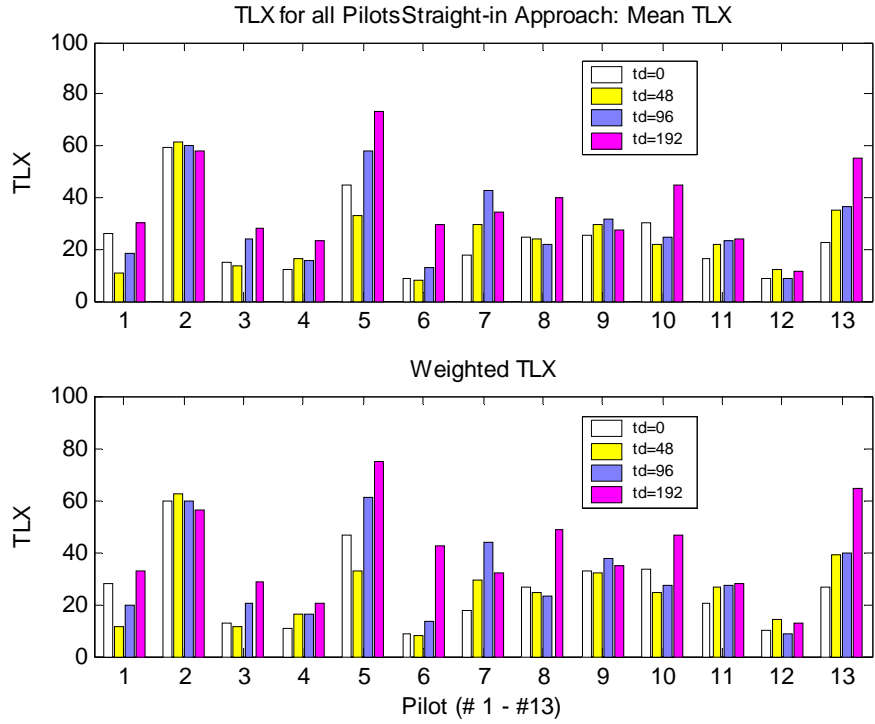
**Fig. B.4.2. TLX for 13 pilots with  $t_d=48$  ms (no compensation)**



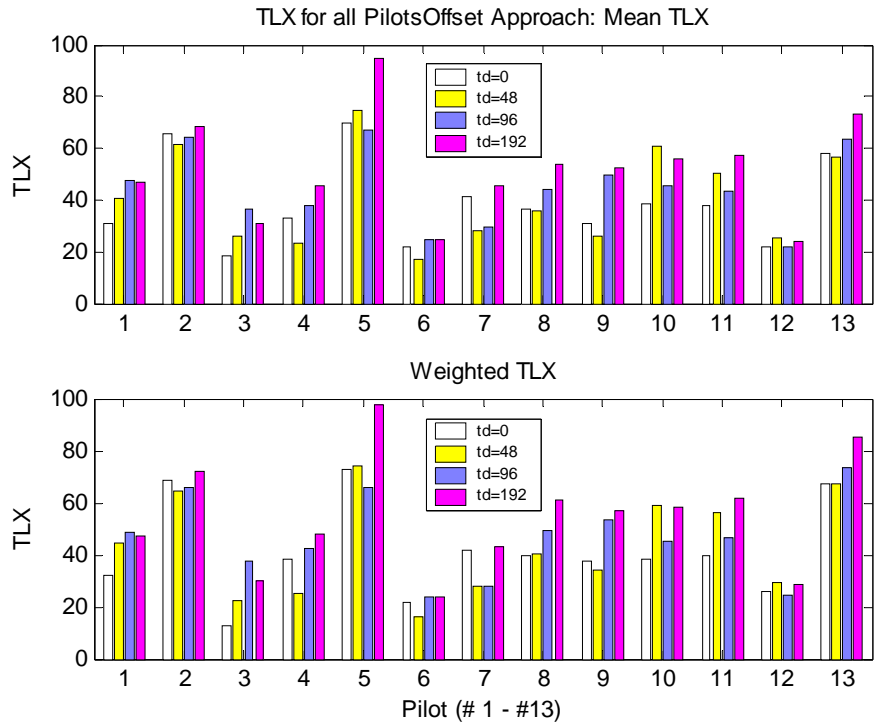
**Fig. B.4.3. TLX for 13 pilots with  $t_d=96$  ms (no compensation)**



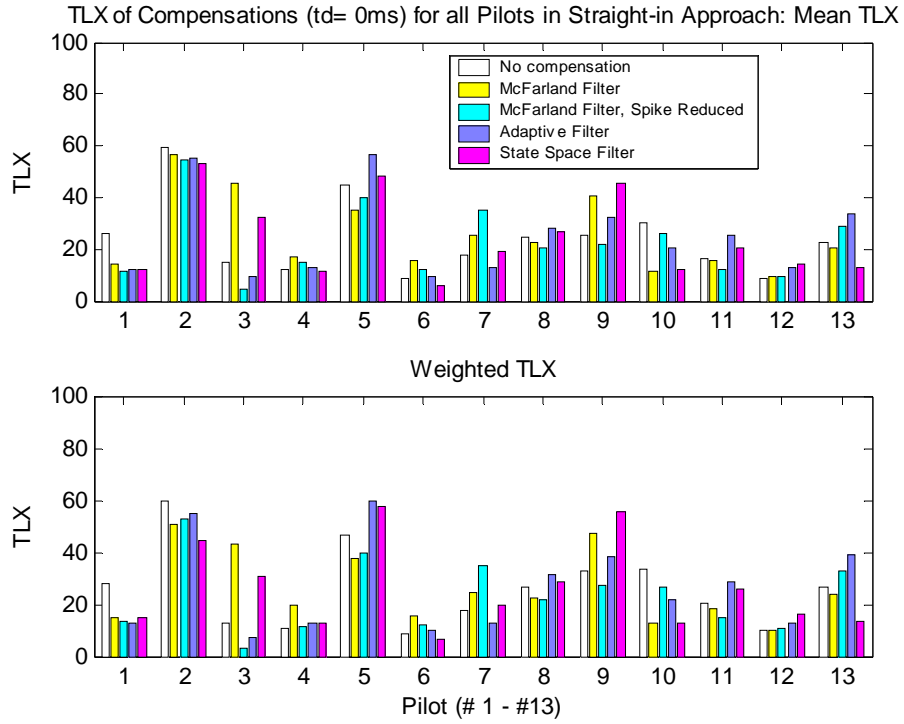
**Fig. B.4.4. TLX for 13 pilots with  $t_d=192$  ms (no compensation)**



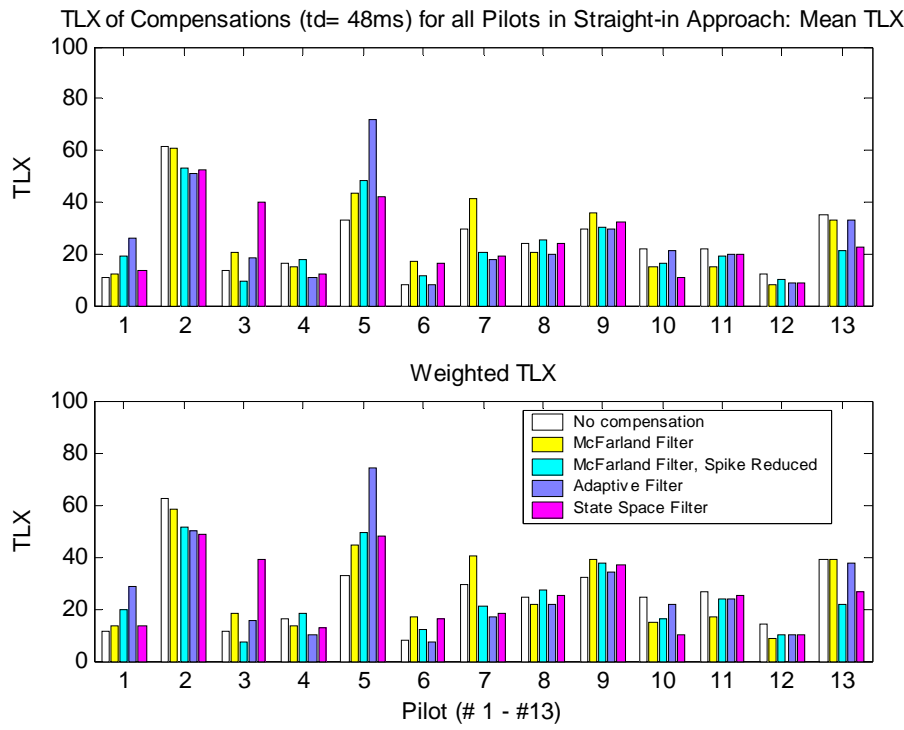
**Fig. B.3.5. TLX of different time delays for 13 Pilots: straight-in approach**



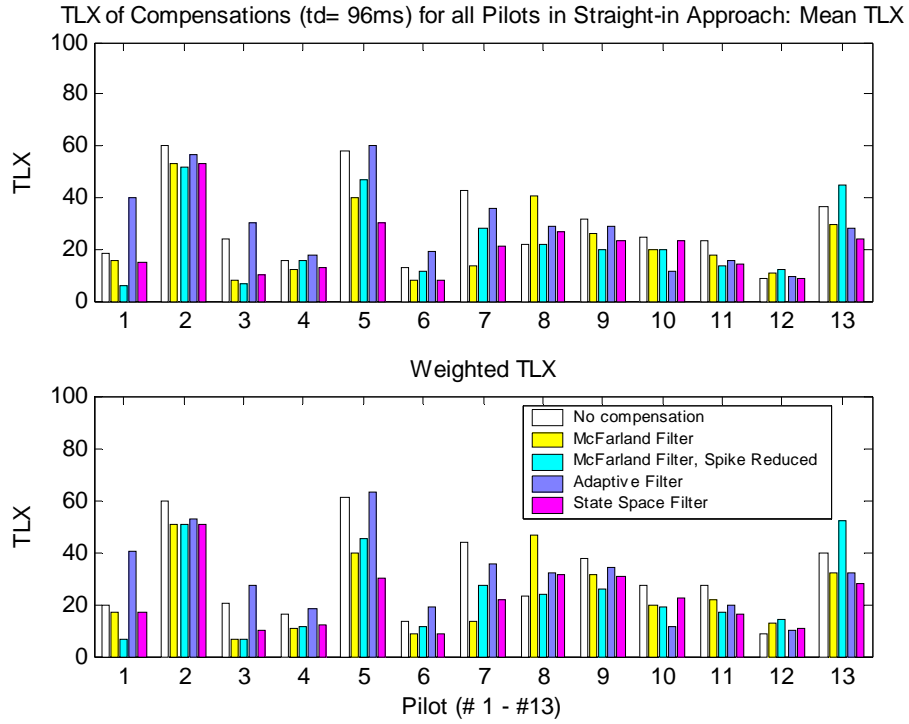
**Fig. B.4.6. TLX of different time delays for 13 Pilots: offset approach**



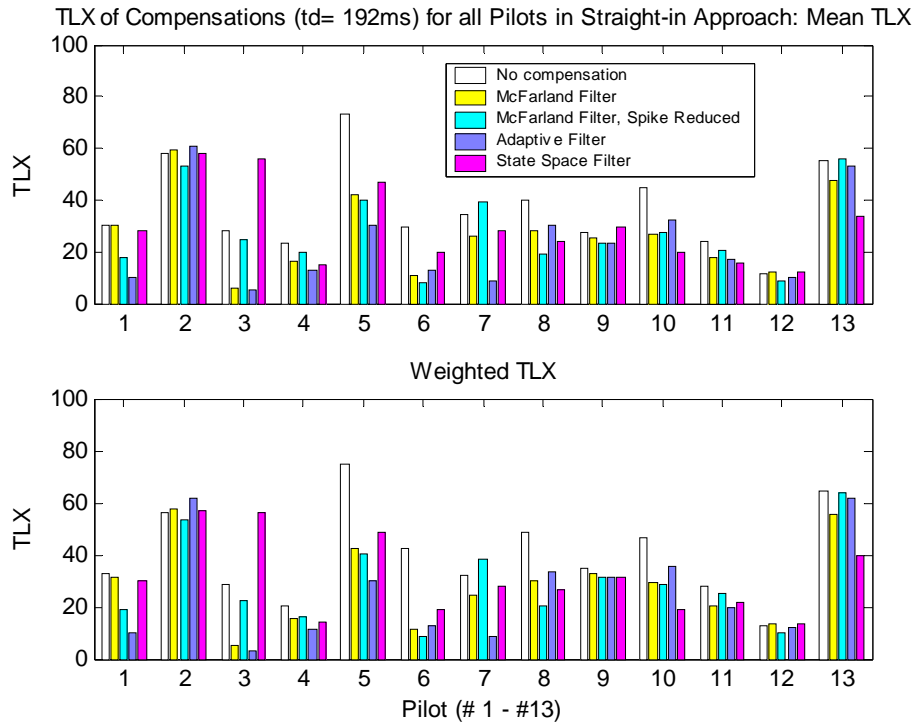
**Fig. B.4.7. Compensation effect to TLX by individual pilot:  $t_d=0\text{ ms}$ , straight-in**



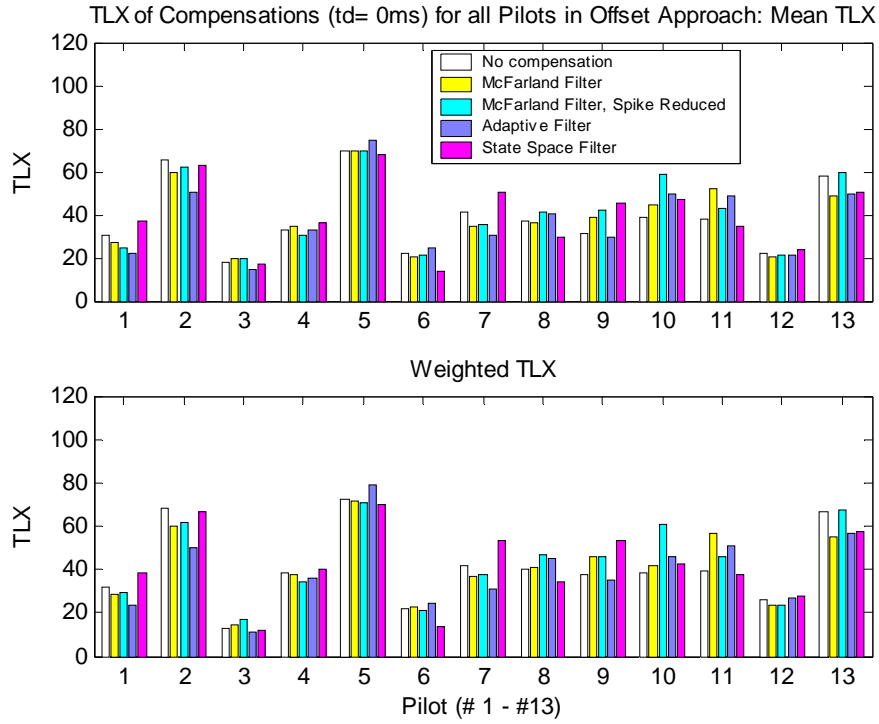
**Fig. B.4.8. Compensation effect to TLX by individual pilot:  $t_d=48\text{ ms}$ , straight-in**



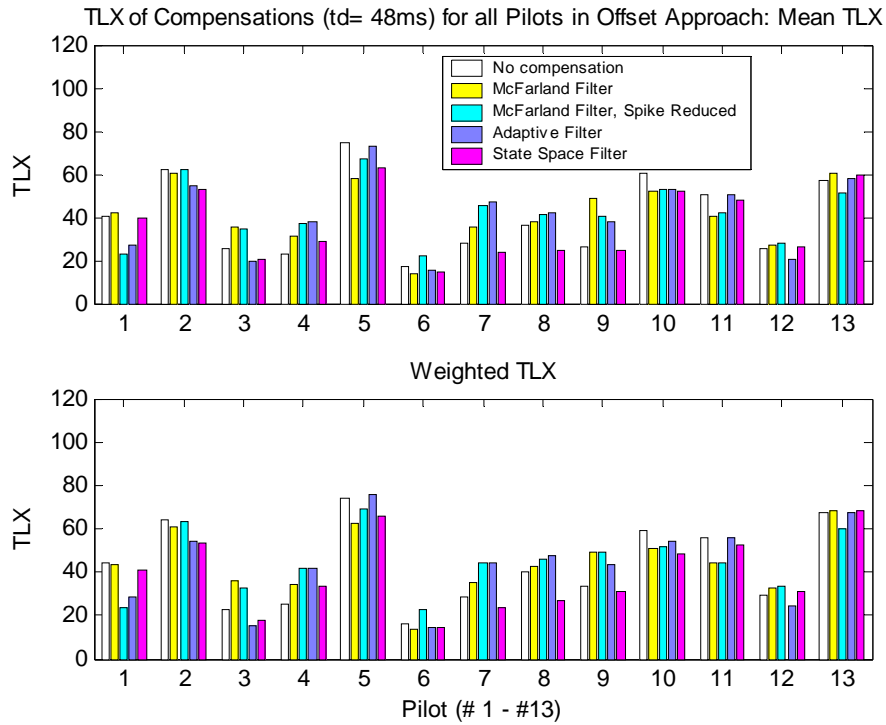
**Fig. B.4.9. Compensation effect to TLX by individual pilot:  $t_d=96$  ms, straight-in**



**Fig. B.4.10. Compensation effect to TLX by individual pilot:  $t_d=192$ ms, straight-in**

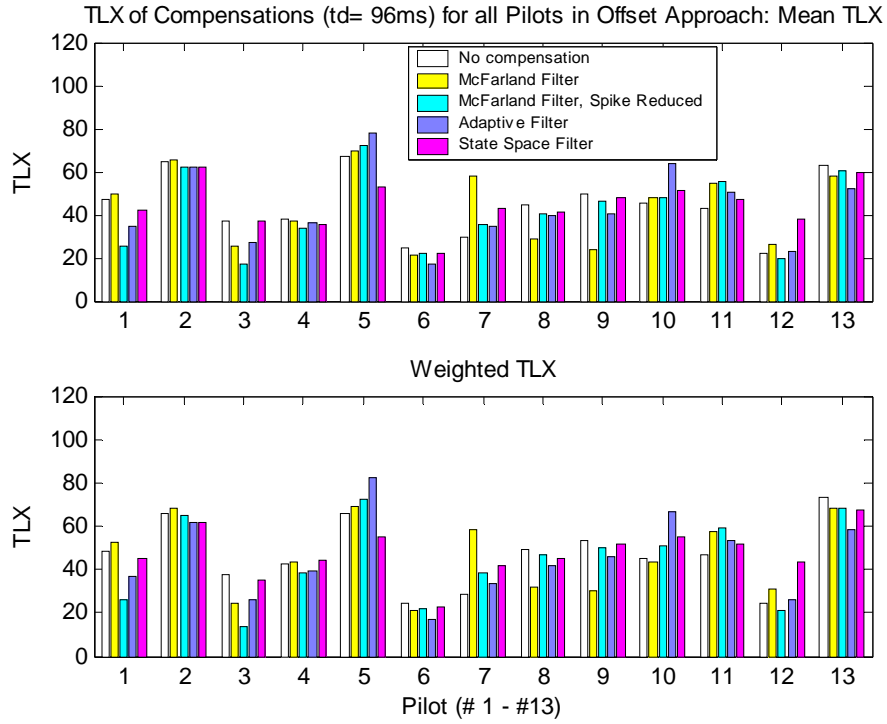


**Fig. B.4.11. Compensation effects to TLX by individual pilot:  $t_d=0\text{ms}$ , offset**

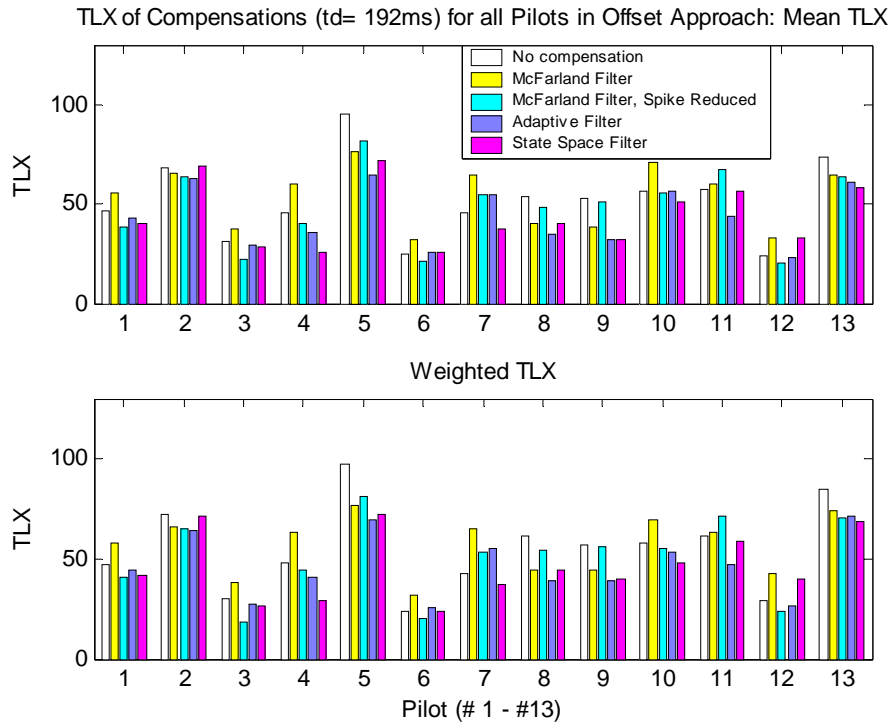


**Fig. B.4.12. Compensation effects to TLX by individual pilot:  $t_d=48\text{ms}$ , offset**



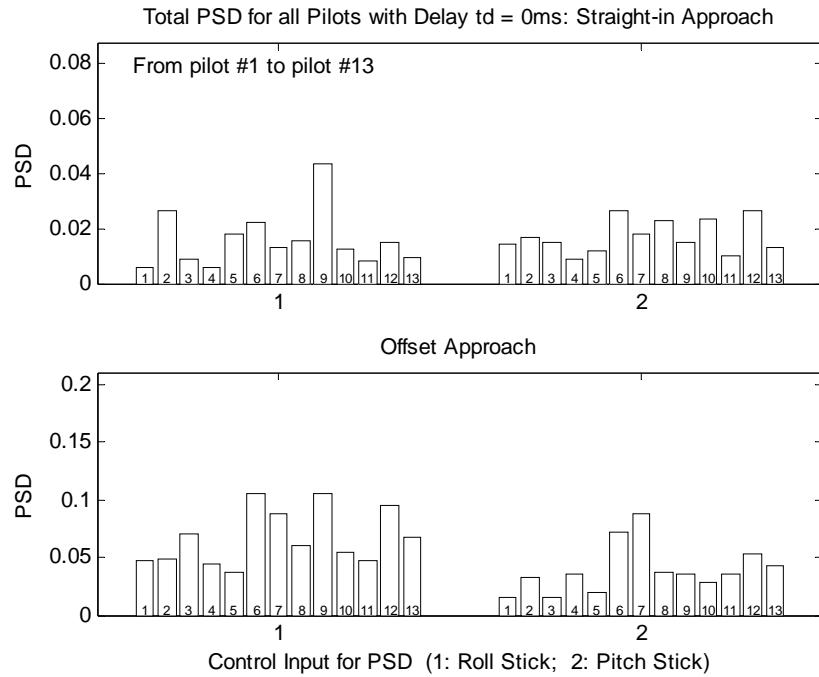


**Fig. B.4.13. Compensation effects to TLX by individual pilot:  $t_d=96$  ms, offset**

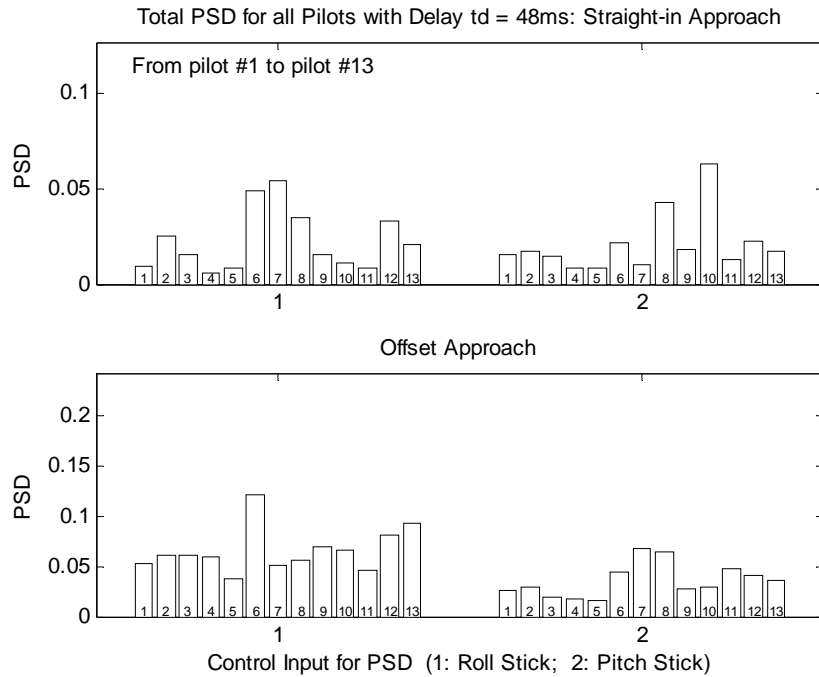


**Fig. B.4.14. Compensation effects to TLX by individual pilot:  $t_d=192$  ms, offset**

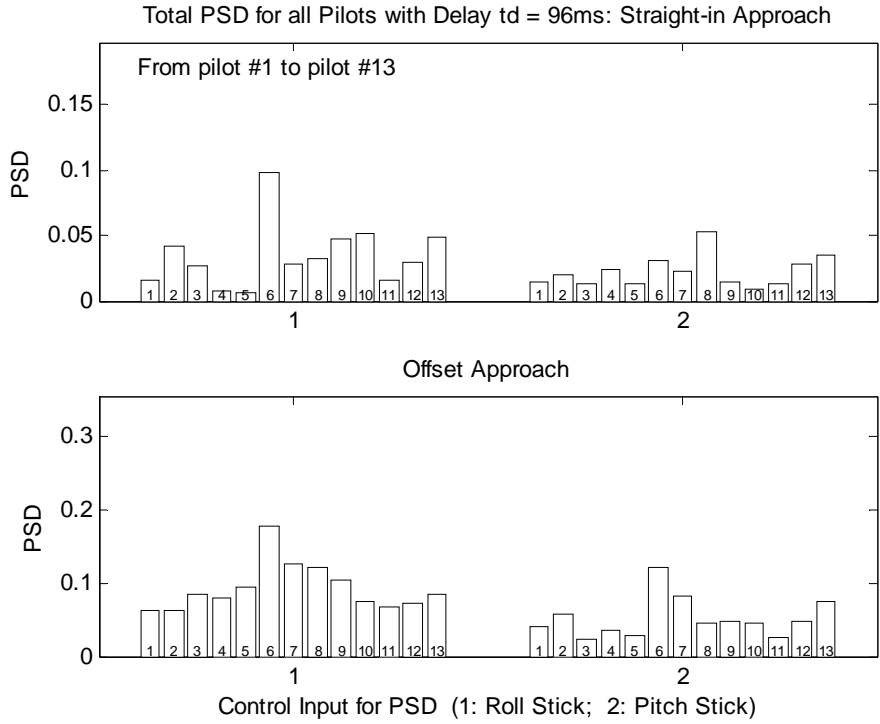
## B.5. Power Spectral Density



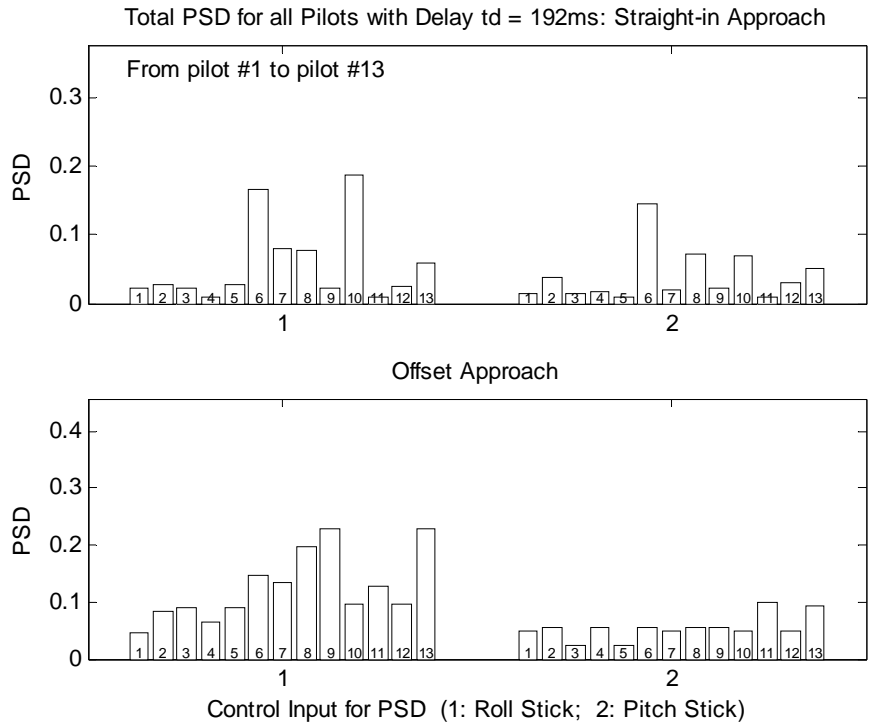
**Fig. B.5.1. PSD for 13 pilots with  $t_d=0$  ms (no compensation)**



**Fig. B.5.2. PSD for 13 pilots with  $t_d=48$  ms (no compensation)**



**Fig. B.5.3. PSD for 13 pilots with  $t_d=96$  ms (no compensation)**



**Fig. B.5.4. PSD for 13 pilots with  $t_d=192$  ms (no compensation)**

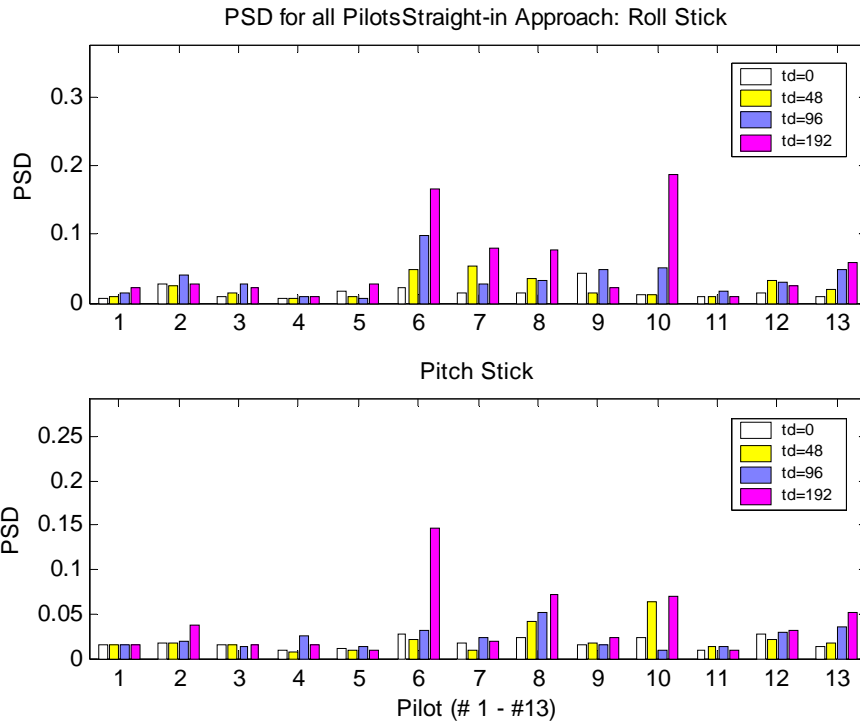


Fig. B.5.5. PSD of different time delays for 13 pilots: straight-in approach

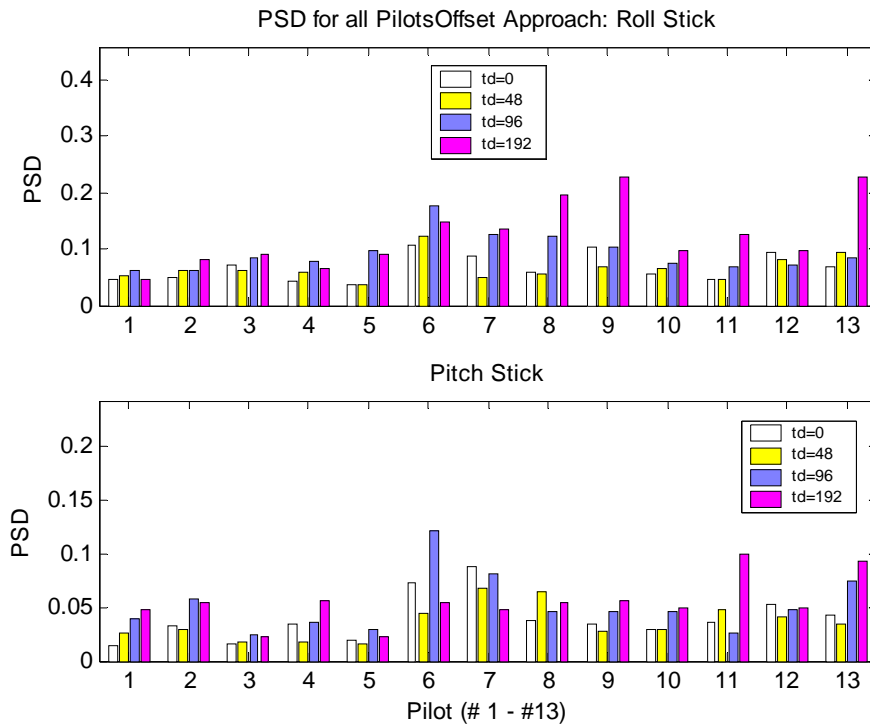
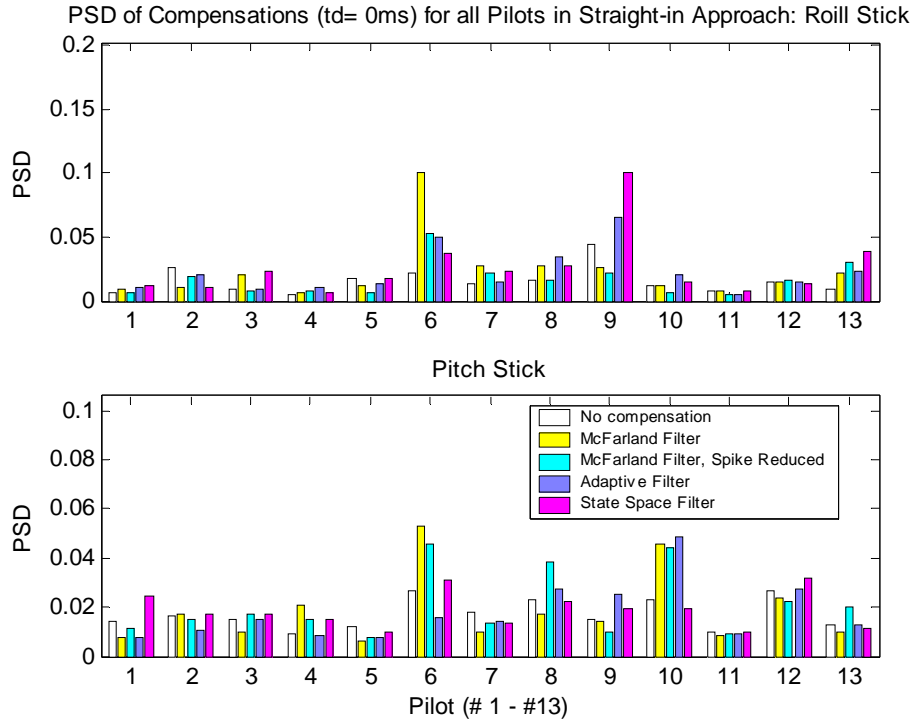
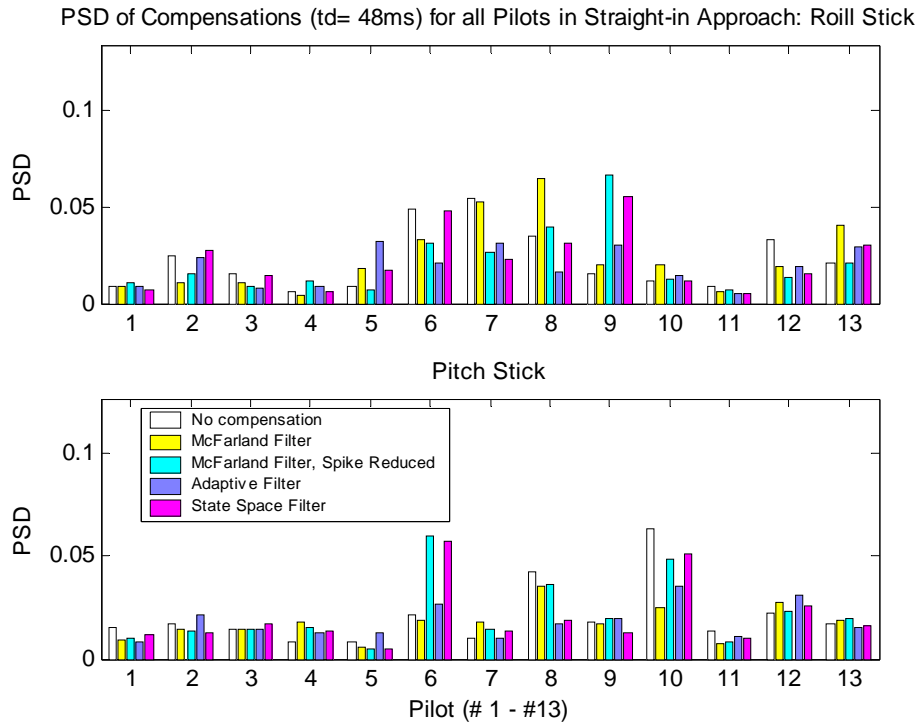


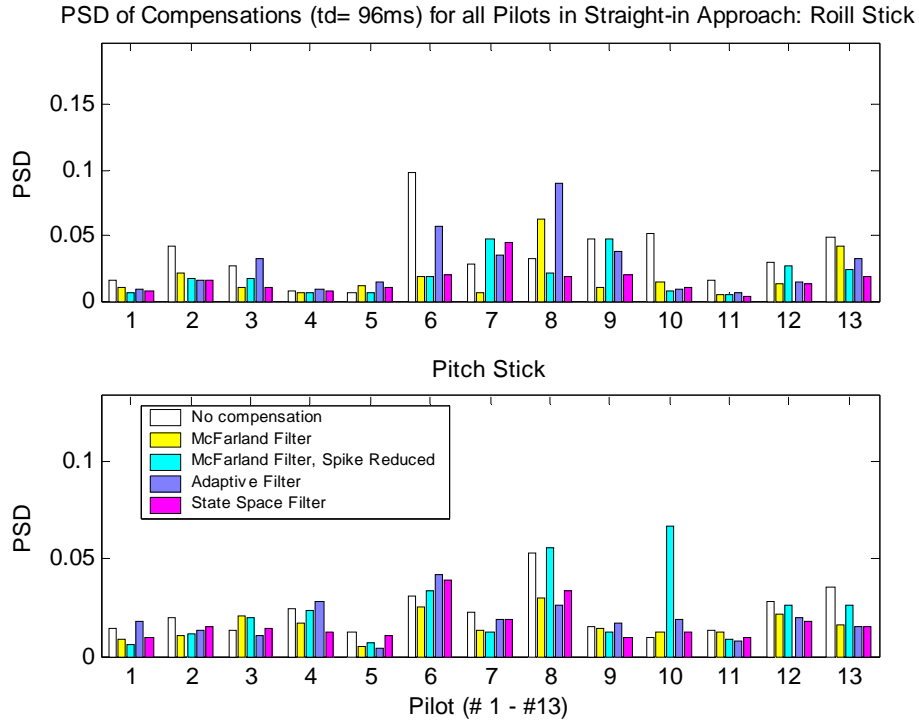
Fig. B.5.6. PSD of different time delays for 13 Pilots: offset approach



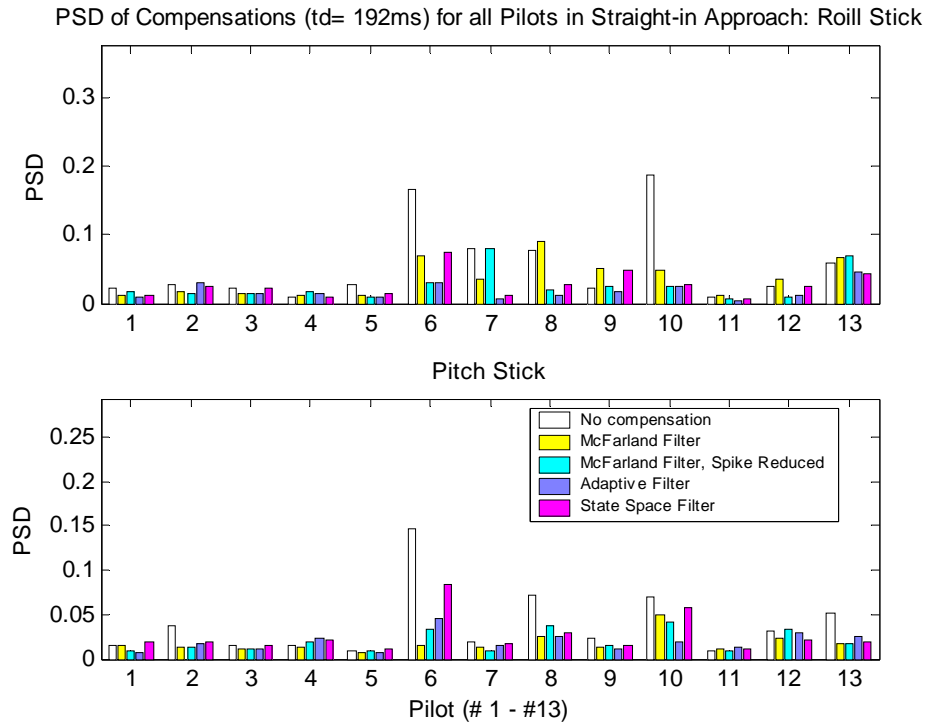
**Fig. B.5.7. Compensation effect on PSD by individual pilot:  $t_d=0$  ms, straight-in**



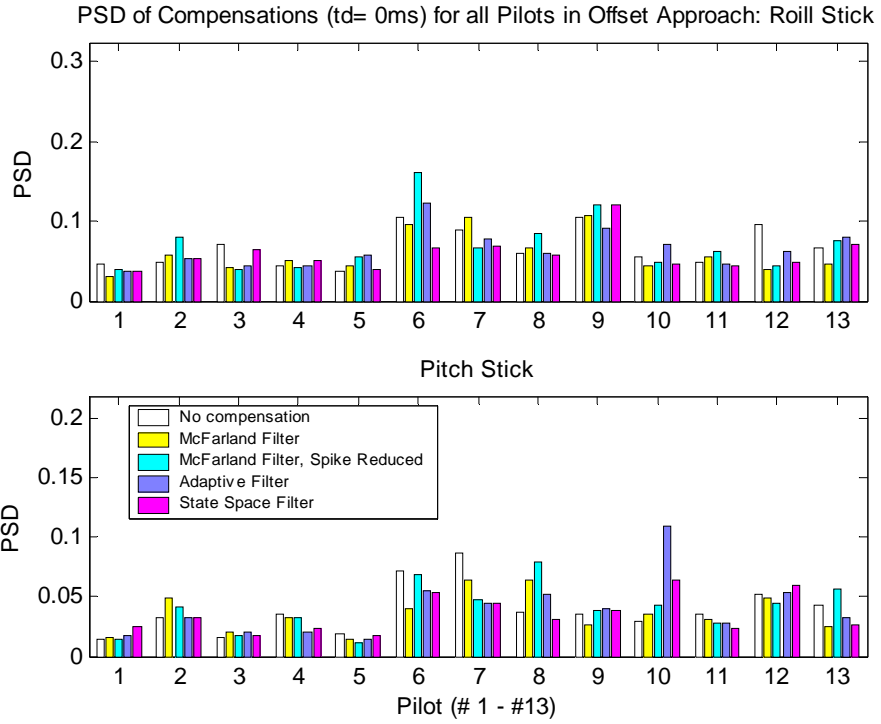
**Fig. B.4.8. Compensation effect on PSD by individual pilot:  $t_d=48$  ms, straight-in**



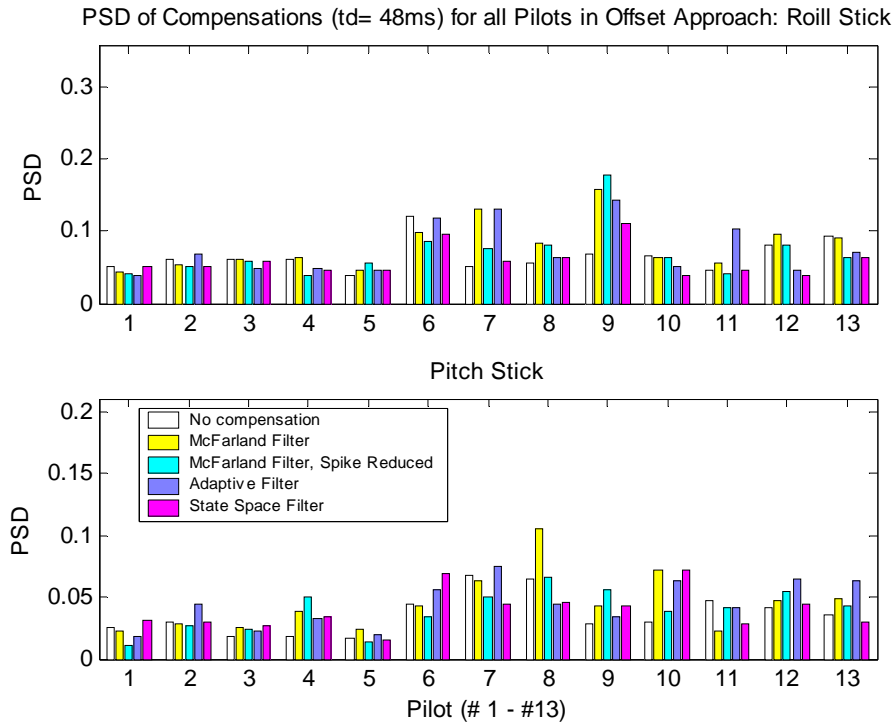
**Fig. B.5.9. Compensation effect on PSD by individual pilot:  $t_d=96\text{ ms}$ , straight-in**



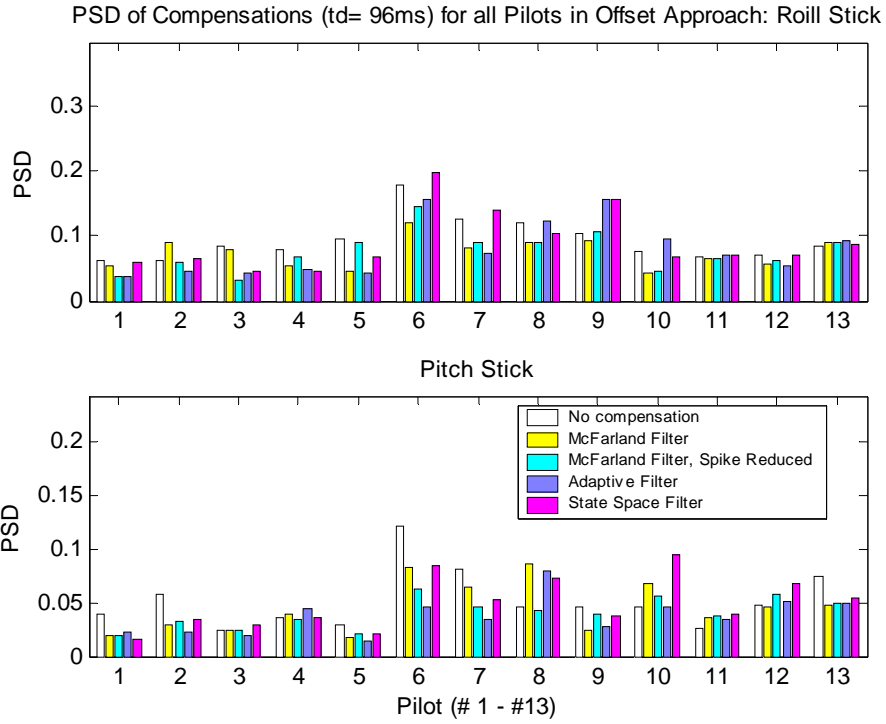
**Fig. B.5.10. Compensation effect on PSD by individual pilot:  $t_d=192\text{ms}$ , straight-in**



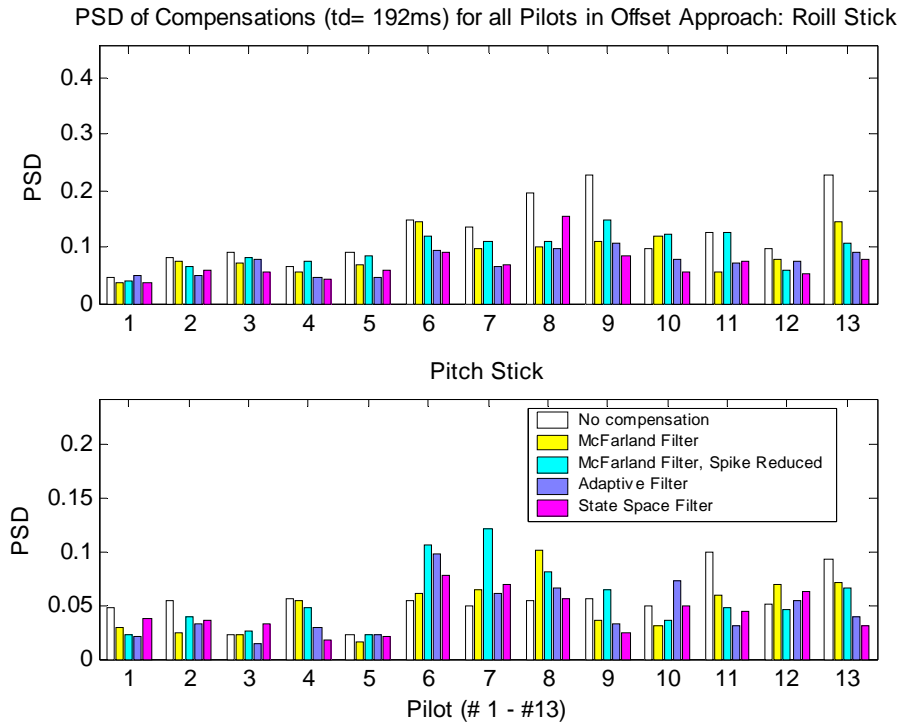
**Fig. B.5.11. Compensation effect on PSD by individual pilot:  $t_d=0$  ms, offset**



**Fig. B.5.12. Compensation effect on PSD by individual pilot:  $t_d=48$  ms, offset**



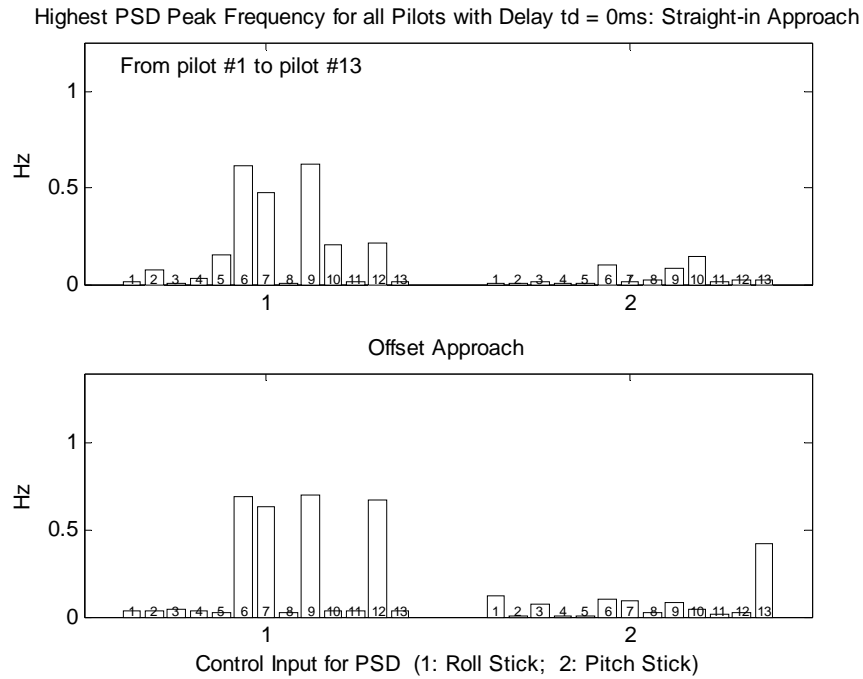
**Fig. B.5.13. Compensation effect on PSD by individual pilot:  $t_d=96$  ms, offset**



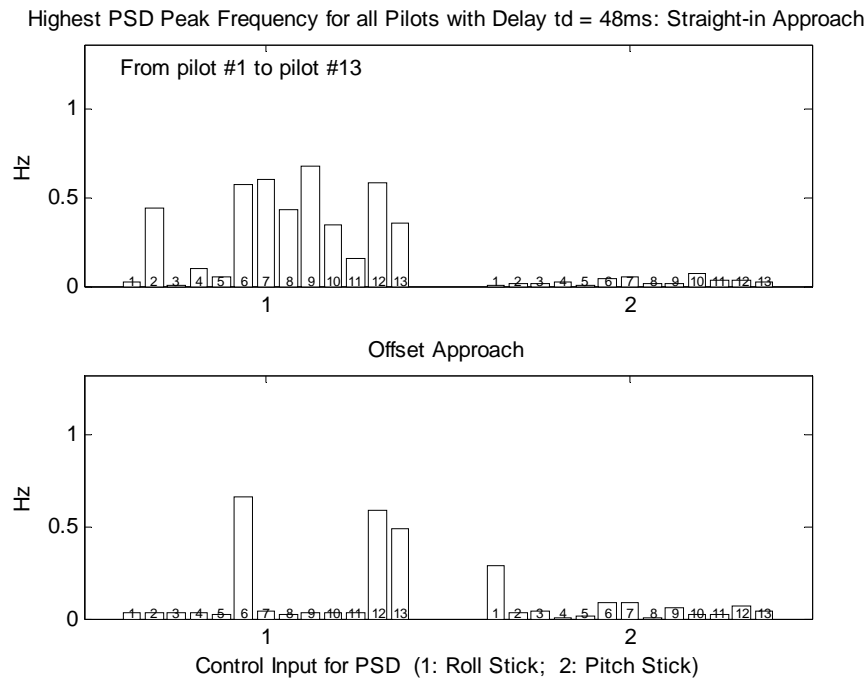
**Fig. B.5.14. Compensation effect on PSD by individual pilot:  $t_d=192$  ms, offset**



## B.5. Frequency of the Highest PSD Peak

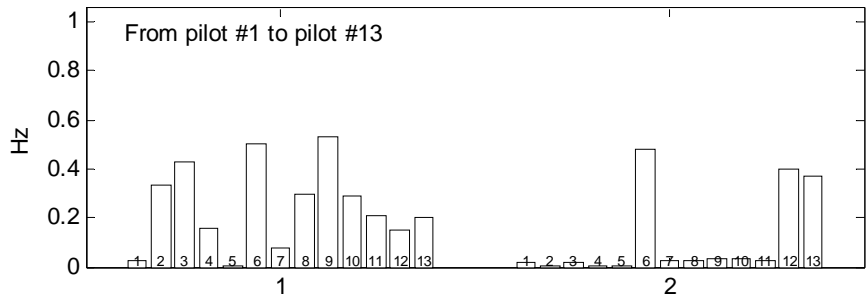


**Fig. B.6.1. Highest PSD peak frequency for 13 pilots with  $t_d=0\text{ms}$  (no compensation)**

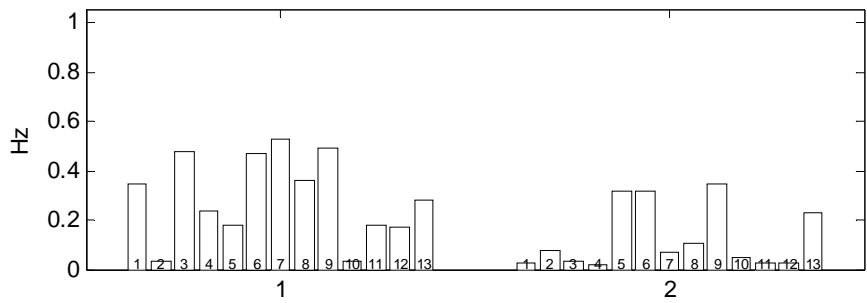


**Fig. B.6.2. Highest PSD peak frequency for 13 pilots with  $t_d=48\text{ms}$  (no compensation)**

Highest PSD Peak Frequency for all Pilots with Delay  $t_d = 96\text{ms}$ : Straight-in Approach



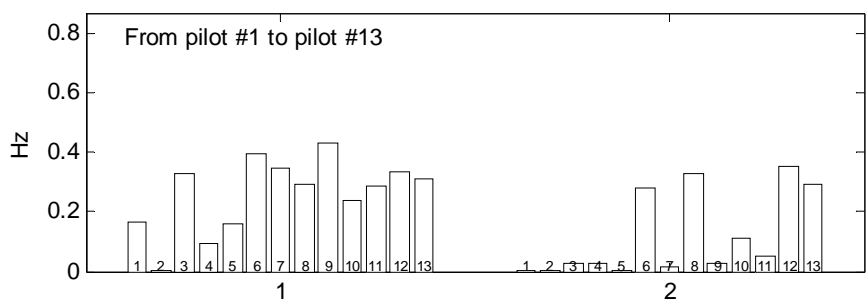
Offset Approach



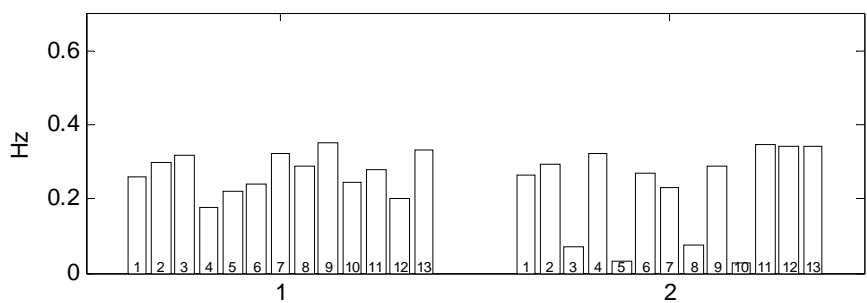
Control Input for PSD (1: Roll Stick; 2: Pitch Stick)

**Fig. B.6.3. Highest PSD peak frequency for 13 pilots with  $t_d=96\text{ms}$  (no compensation)**

Highest PSD Peak Frequency for all Pilots with Delay  $t_d = 192\text{ms}$ : Straight-in Approach

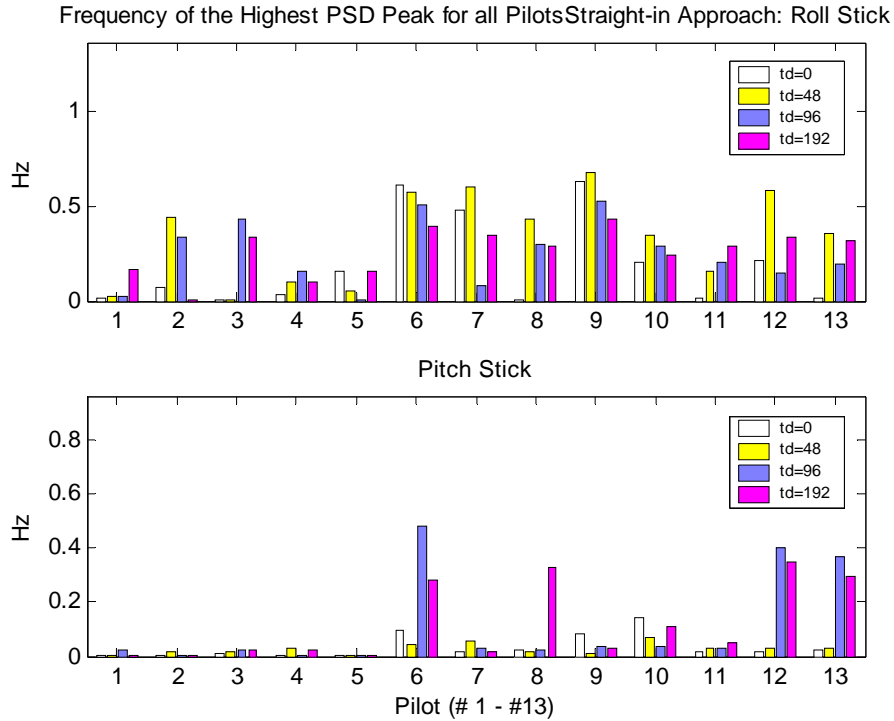


Offset Approach

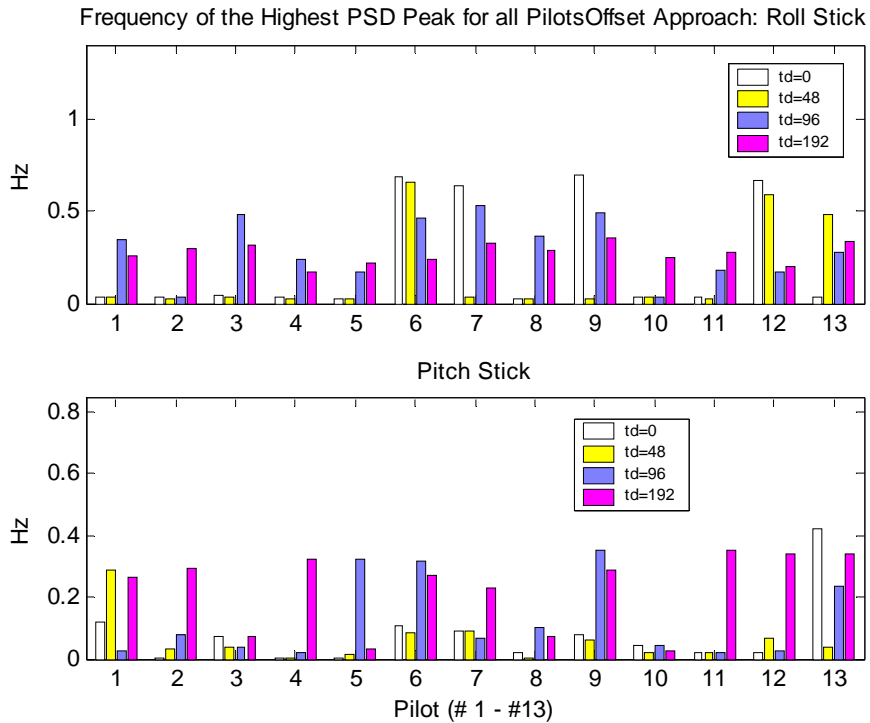


Control Input for PSD (1: Roll Stick; 2: Pitch Stick)

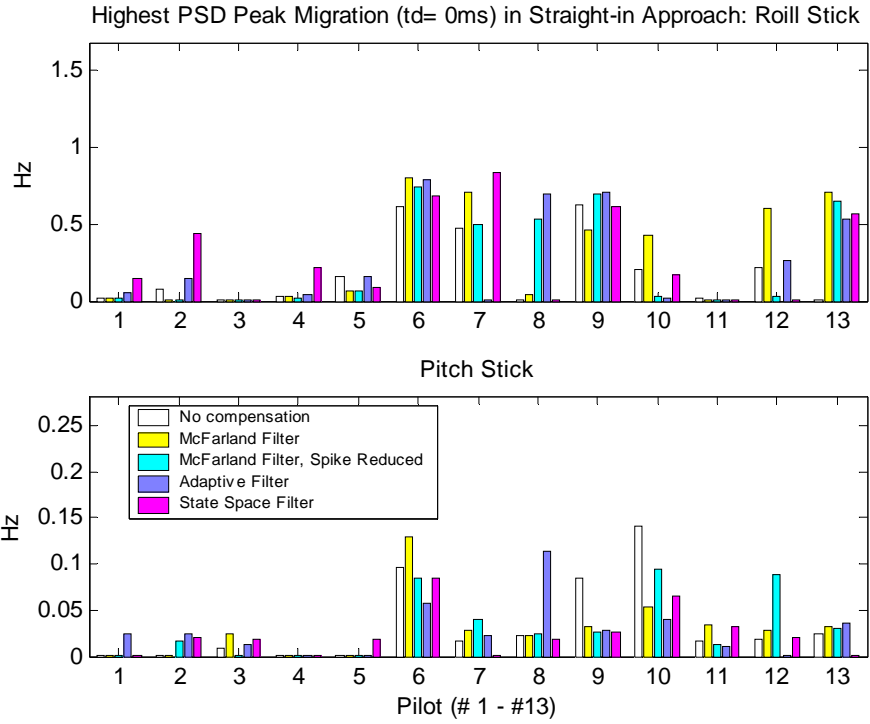
**Fig. B.6.4. Highest PSD peak frequency for 13 pilots for  $t_d=196\text{ms}$  (no compensation)**



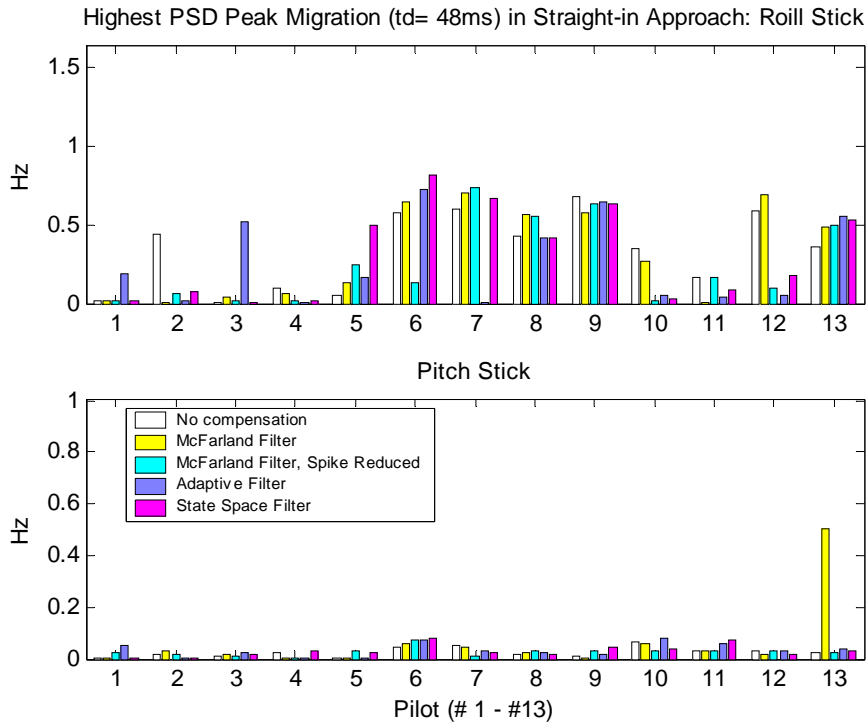
**Fig.B.6.5. Highest PSD peak frequency of different delays: straight-in approach**



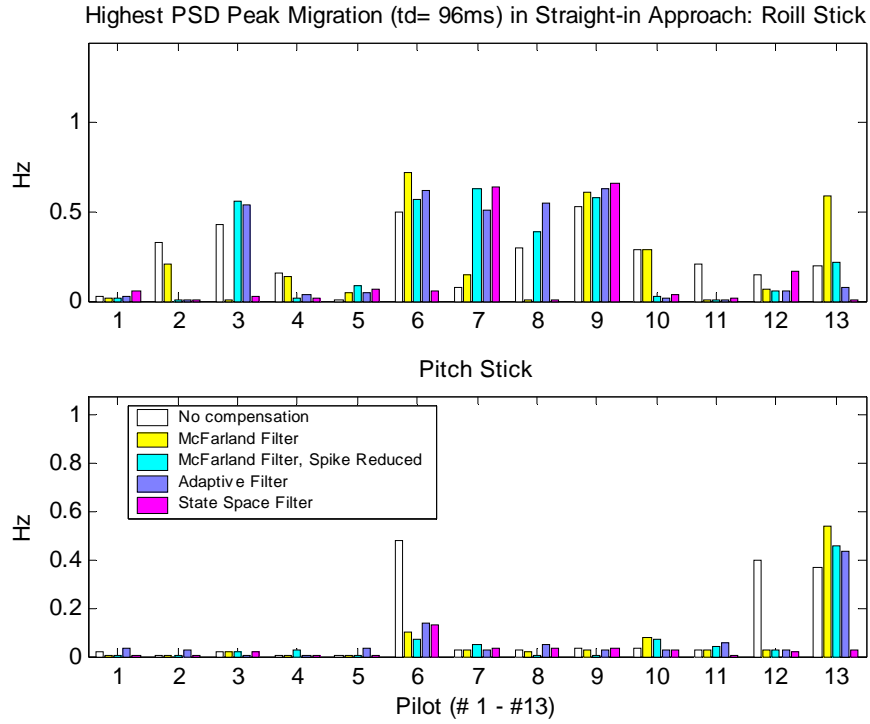
**Fig. B.6.6. Highest PSD peak frequency of different delays: offset approach**



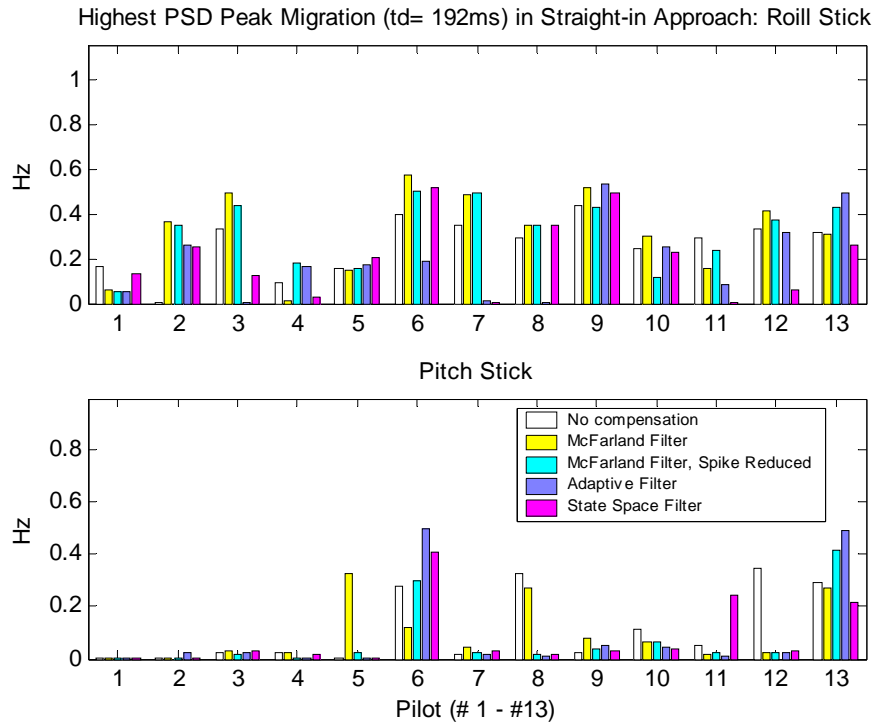
**Fig. B.6.7. Highest PSD peak migration by compensation:  $t_d = 0\text{ms}$ , straight-in**



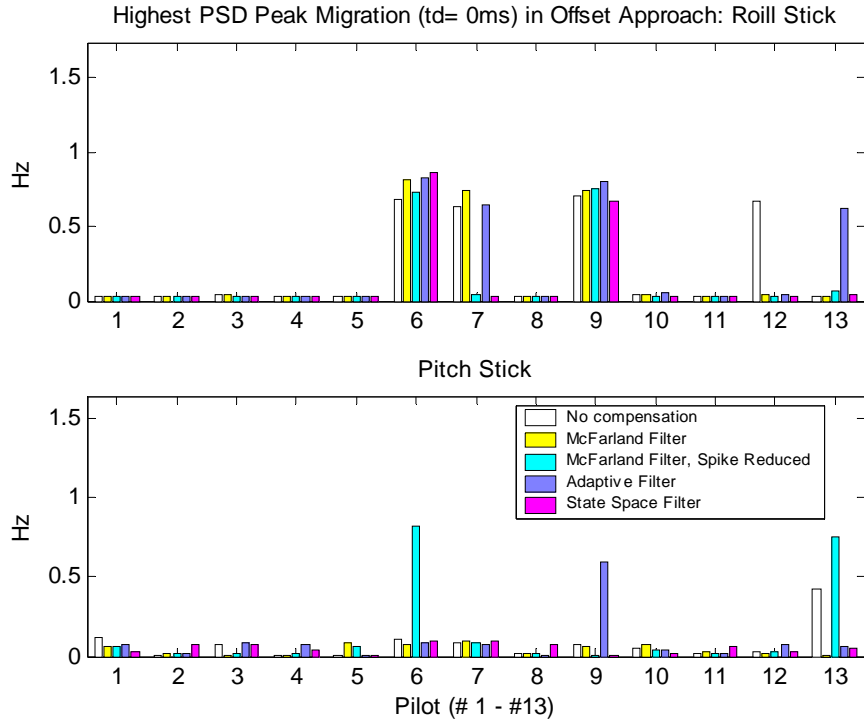
**Fig. B.6.8. Highest PSD peak migration by compensation:  $t_d = 48\text{ms}$ , straight-in**



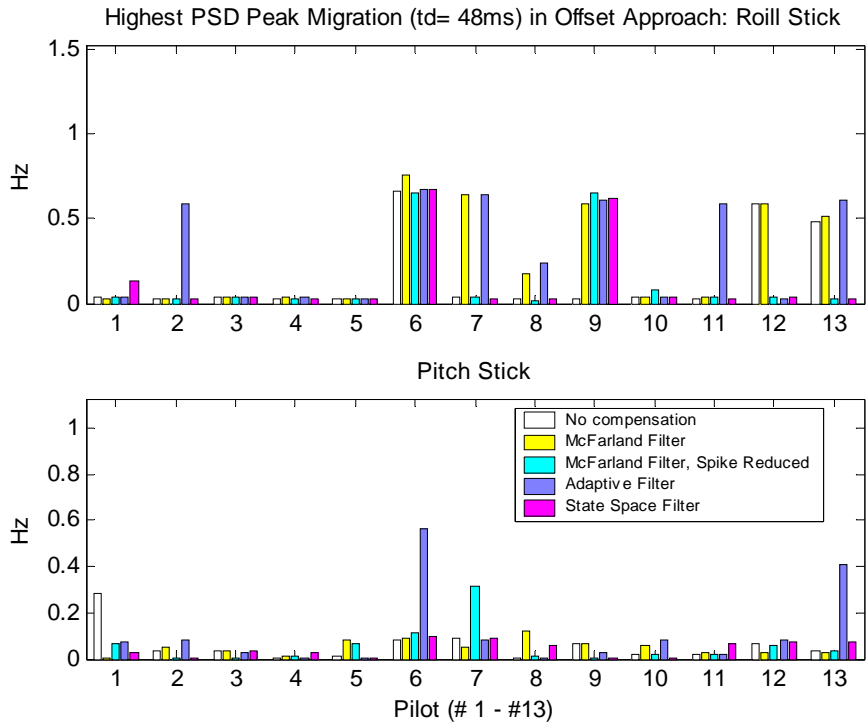
**Fig. B.6.9. Highest PSD peak migration by compensation:  $t_d=96\text{ms}$ , straight-in**



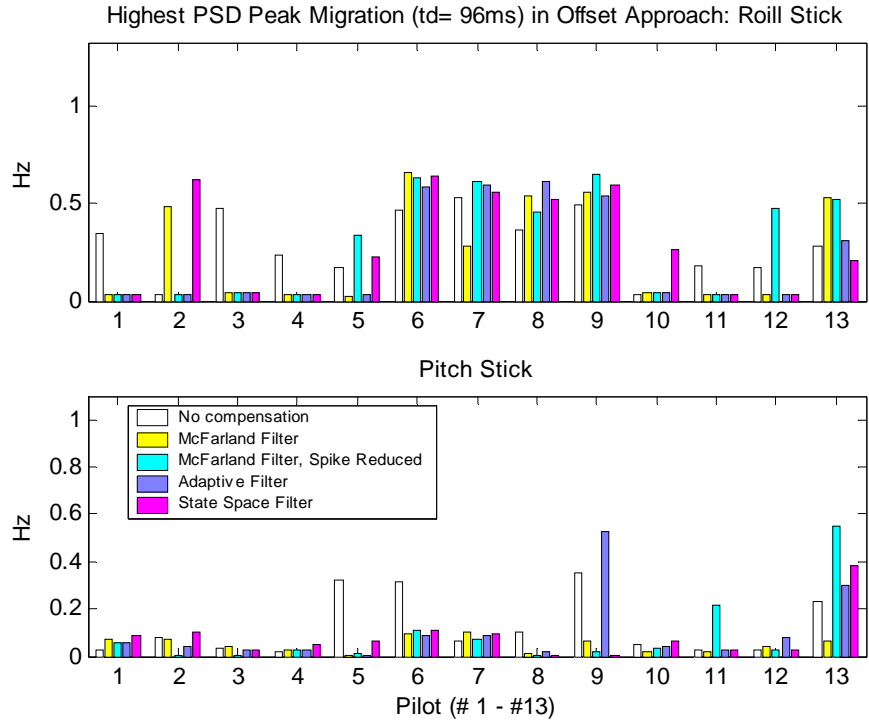
**Fig. B.6.10. Highest PSD peak migration by compensation:  $t_d=0\text{ms}$ , straight-in**



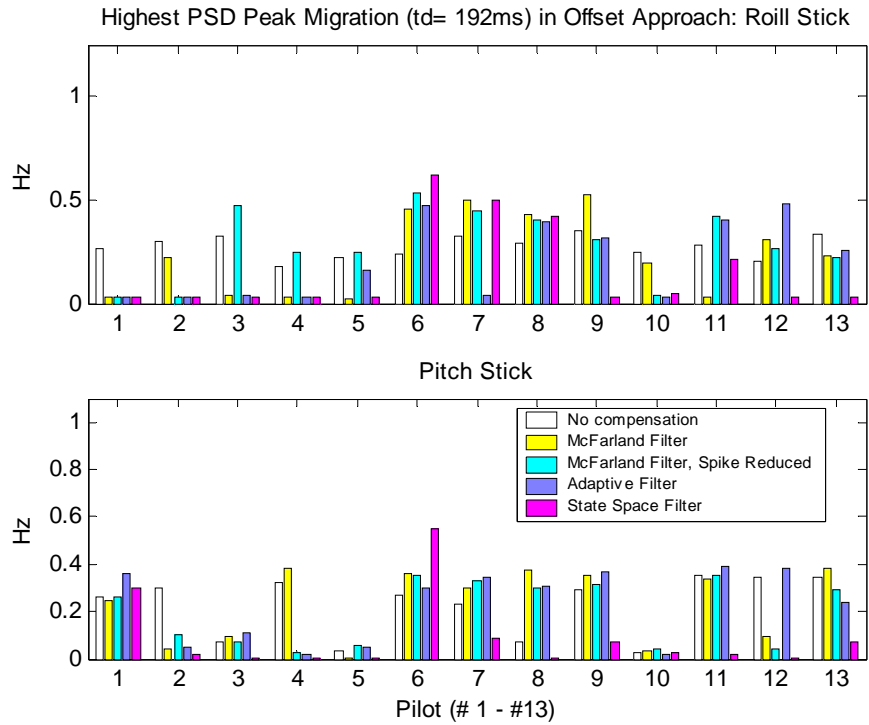
**Fig. B.5.11. Highest PSD peak migration by compensation:  $t_d=0ms$ , offset**



**Fig. B.6.12. Highest PSD peak migration by compensation:  $t_d=48ms$ , offset**



**Fig. B.6.13. Highest PSD peak migration by compensation:  $t_d=96\text{ms}$ , offset**



**Fig. B.6.14. Highest PSD peak migration by compensation:  $t_d=192\text{ms}$ , offset**

## Appendix C. MATLAB Source Codes

### C.1. Converting ASCII Data to MATLAB Data Structure

```
% Program name: tfgetlFinal.m
% This MATLAB script converts the ASCII text data files from the
% NASA LaRC real-time sim.
% Variables are saved in MATLAB structure with NAMES, UNITS and
% DATA

% Oct. 24, 2004

clear

% Read variable names and units

filename1='rt_run';
filename3='_0202.txt';
filename4='_p13';

% pilot 13. The number is different for other pilots

for z=1:40 % there are 40 runs for each pilot
    filename2=num2str(z);
    SourceFile=[filename1 filename2 filename3];
    fid=fopen(SourceFile);

    flag=0;
    while flag<75 %from line 3 to 63, list of variable names and units
        tline = fgetl(fid);
        flag=flag+1;
        len=length(tline);
        if flag>=2
            for i=8:18 % from column 8 to 18, the variable name
                if tline(i)~=' '
                    name(flag-1,i-7)=tline(i);
                else
                    break;
                end
            end
            for i=19:28 % from column 8 to 18, the variable unit
                if tline(i)~=' '
                    unit(flag-1,i-18)=tline(i);
                else
                    break;
                end
            end
        end
    end
end

% Read data

while 1
    tline = fgetl(fid);
```



```

len=length(tline);
flag=flag+1;
if ~ischar(tline), break, end
if flag>=91          % the data begin from line 79
    datan=floor((flag-91)/13)+1; % data start from line 91
    line1=91+13*(datan-1);      % 13 digits for each variable
    datam=(flag-line1)*6;      % there are 6 variables in each row
    for l=0:5
        if (datam==72)&(l==2), break, end
        lcol=l*13;
        clear temp
        for i=lcol+1:lcol+13
            % from column 8 to 18, the variable name
            if i<=len
                if tline(i)~=' '
                    temp(i-lcol)=tline(i);
                else
                    break;
                end
                ovfl=0;
            else
                ovfl=1;
            end
        end
        if ovfl==0
            data(datam+l+1,datan)=str2num(temp);
        end
    end
end
end

% Create the data structure. Save it in a file that a MATLAB
analysis script can read

fclose(fid);
disp(filename2)
run=struct('name',name,'unit',unit,'data',data);
TargetFile=[filename1 filename2 filename4];
save(TargetFile, 'run')
clear name unit data run
end

```

## C.2. Calculate Smoothed PSD (Periodogram)

```
% Program name: smPSD.m. Date: June, 2002

function [P, f] = smPSD(x, T, ifNorm)

% This function calculates the normalized PSD of a real signal and the
% frequency of the PSD
% the called function smoooper.m must be in the same directory

% input: x--the signal
% input: T--the sampling rate of the signal
% input: ifNorm--flag for normalization (1) or not (0)
% output: P--the PSD
% output: f--the frequency (Hz)

x=x-mean(x);
len=length(x);
lfft=4096;          % Total number of PSD points
xz=[x zeros(1,lfft-len)]; % zero padding

% determine the length of the window, it must be odd

if (rem(len,2) == 0)
    lwin=len-1;
else
    lwin=len;
end

% Use hamming window. Other window can also be used

w = window(@hamming,lwin);
P = smpgram(xz,w'); % smpgram is available in Appendix C.3
if ifNorm==1
    P = P/max(P); % normalization, optimal
end

% Determine the frequency sequence whose length is the same as the PSD
% sequence

Fs = 1/T; % T is the sampling period
df = Fs/length(P)/2; % Frequency step size
f = 0:df:Fs/2-df;
```

### C.3. Function (Subroutine) Called by smPSD.m (Appendix C.2)

```
% Program name: smpgram.m. Date: June, 2002
% Computes the smoothed periodogram of the data vector x with w as the
% windowing sequence.

function s = smpgram(x,w)

% Syntax: s = smpgram(x,w).
% Input:
% x: the data vector
% w: the window; must have odd length.
% Output:
% s: the smoothed periodogram of length equal to that of x.

if (rem(length(w),2) == 0)
% The function "rem" calculates the remains
error('Window in SMOOPER must have an odd length');
end
x = reshape(x,1,length(x)); % change x to a column vector

% "conv" calculate the convolution of a signal
% "fliplr" reverse the input vector
% Therefore, "conv(x,fliplr(x))" returns the autocorrelation of x

kappa = (1/length(x))*conv(x,fliplr(x));
n = 0.5*(length(kappa)-length(w));

% "fft" calculates the FFT of the input

s = fft([zeros(1,n),w,zeros(1,n)].*kappa);

% "abs" returns the absolute value of the input.
% If the input is a complex number, it returns the magnitude
% For PSD, it only needs the magnitude information

s = abs(s(1:length(x)))
```

## C.4. Adaptive Predictor: Stochastic Approximation Algorithm

```
% MATLAB function to implement the adaptive predictor using the
% stochastic approximation algorithm

% Program name: adaptiveSA.m. Date: September 2003

function yp=adaptiveSA(y,v,T,td);

% input: y-the sequence of the aircraft state to be predicted
% input: v-the velocity of y
% input: T-the update period (frame length)
% input: td-the time delay
% output: yp-the prediction of y with the stochastic approximation
% method

[numd,dend]=pade(td,2);      % Pade approximation of td
if td==0
    numd=[0 0 1];
    dend=[0 0 1];
end
[A,B,C,D]=tf2ss(numd,dend);
[Ad,Bd]=c2d(A,B,T);

[b0,b1,b2]=mcfcoefs(td,T,3); % Calculate the McFarland coefficients
%
% Calculate the velocity vectors consisting of 3 steps of velocity
%
v0=v;
for i=1:len4
    if i==1
        v1(i)=v0(i);
        v2(i)=v1(i);
    elseif i==2
        v1(i)=v0(i-1);
        v2(i)=v1(i);
    else
        v1(i)=v0(i-1);
        v2(i)=v1(i-1);
    end
end
%
% Stochastic approximation algorithm
%
x=zeros(2,1);
iP=0;          % iP is the epsilon in the SA algorithm
for i=1:length(y)
    x=Ad*x+Bd*y(i);
    yd(i)=C*x+D*y(i); % calculate the delayed y
%
% use McFaland coefficients for the 1st 4 iterations
%
    if i<5
        phi=[v0(i) v1(i) v2(i)]';
        iP=iP+phi'*phi;
%
end
```

```

% use left pseudo-inversion to realize LS regression for 5th iteration
%
elseif i==5
    Phi=[v0(1:i)' v1(1:i)' v2(1:i)'];
    phi=[v0(i) v1(i) v2(i)]';
    Y=y(1:i)'-y4(1:i)';
    P=inv(Phi'*Phi);
    theta=P*Phi'*Y;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
    iP=iP+phi'*phi;
%
% Start the stochastic approximation algorithm from 6th iteration
%
elseif i>5
    phi=[v0(i) v1(i) v2(i)]';
    iP=iP+phi'*phi;
    theta=theta+phi*(y(i)-yd(i)-phi'*theta)/iP;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
end
%
% Calculate the prediction after the coefficients are updated
%
yp(i)=y(i)+b0*v0(i)+b1*v1(i)+b2*v2(i);
end

```

## C.5. Adaptive Predictor: Kalman Filter Algorithm

```
% MATLAB function to implement the adaptive predictor using the Kalman
% filter algorithm

% Program name: adaptiveKalman.m. Date: September 2003

function yp=adaptiveKalman(y,v,T,td);

% input: y--the sequence of the aircraft state to be predicted
% input: v--the velocity of y
% input: T--the update period (frame length)
% input: td--the time delay
% output: yp--the prediction of y with the Kalman method

[numd,dend]=pade(td,2);      % Pade approximation of td
if td==0
    numd=[0 0 1];
    dend=[0 0 1];
end
%
% to state space form
%
[A,B,C,D]=tf2ss(numd,dend);
[Ad,Bd]=c2d(A,B,T);

[b0,b1,b2]=mcfcoefs(td,T,3);% Calculate the McFarland coefficients
%
% Calculate the velocity vectors consisting of 3 steps of velocity
%
v0=v;
for i=1:len4
    if i==1
        v1(i)=v0(i);
        v2(i)=v1(i);
    elseif i==2
        v1(i)=v0(i-1);
        v2(i)=v1(i);
    else
        v1(i)=v0(i-1);
        v2(i)=v1(i-1);
    end
end
end
%
% Kalman filter algorithm
%
x=zeros(2,1);
iP=0;
for i=1:length(y)
    x=Ad*x+Bd*y(i);
    yd(i)=C*x+D*y(i); % calculate the delayed y
%
% use left pseudo-inversion to realize LS regression for 5th iteration
%
    if i==5
        Phi=[v0(1:i)' v1(1:i)' v2(1:i)'];
```

```

    phi=[v0(i) v1(i) v2(i)]';
    Y=y(1:i) '-yd(1:i)';
    P=inv(Phi'*Phi);
    theta=P*Phi'*Y;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
%
% Start the Kalman filter algorithm from 6th iteration
%
elseif i>5
    phi=[v0(i) v1(i) v2(i)]';
    K=P*phi*inv(ff+phi'*P*phi);
    theta=theta+K*(y(i)-yd(i)-phi'*theta);
    P=(eye(3)-K*phi')*P;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
end
%
% Calculate the prediction after the coefficients are updated
%
yp(i)=y(i)+b0*v0(i)+b1*v1(i)+b2*v2(i);
end

```

## C.6. Adaptive Predictor: Karzmarz Algorithm

```
% MATLAB function to implement the adaptive predictor using
% the Karzmarz algorithm

% Program name: adaptiveKarzmarz.m. Date: September 2003

function yp=adaptiveKarzmarz(y,v,T,td);

% input: y--the sequence of the aircraft state to be predicted
% input: v--the velocity of y
% input: T--the update period (frame length)
% input: td--the time delay
% output: yp--the prediction of y with the Karzmarz method

[numd,dend]=pade(td,2);      % Pade approximation of td
if td==0
    numd=[0 0 1];
    dend=[0 0 1];
end
%
% to state space form
%
[A,B,C,D]=tf2ss(numd,dend);
[Ad,Bd]=c2d(A,B,T);

[b0,b1,b2]=mcfcoefs(td,T,3); % Calculate McFarland coefficients
%
% Calculate the velocity vectors consisting of 3 steps of velocity
%
v0=v;
for i=1:len4
    if i==1
        v1(i)=v0(i);
        v2(i)=v1(i);
    elseif i==2
        v1(i)=v0(i-1);
        v2(i)=v1(i);
    else
        v1(i)=v0(i-1);
        v2(i)=v1(i-1);
    end
end

% Least mean square approximation algorithm
x=zeros(2,1);
lms=50;
for i=1:length(y)
    x=Ad*x+Bd*y(i);
    yd(i)=C*x+D*y(i);      % calculate the delayed y
%
% use left pseudo-inversion to realize LS regression for 5th iteration
%
    if i==5
        Phi=[v0(1:i)' v1(1:i)' v2(1:i)'];
        phi=[v0(i) v1(i) v2(i)]';
```



```

        Y=y(1:i) '-yd(1:i) ' ;
        P=inv(Phi '*Phi) ;
        theta=P*Phi '*Y;
        b0=theta(1) ;
        b1=theta(2) ;
        b2=theta(3) ;
%
% Start the Karzmarz algorithm from 6th iteration
%
elseif i>5
    phi=[v0(i) v1(i) v2(i)] ' ;
    K=nf*phi/(df+phi '*phi) ;
    theta=theta+K*(y(i)-yd(i)-phi '*theta) ;
    b0=theta(1) ;
    b1=theta(2) ;
    b2=theta(3) ;
end
%
% Calculate the prediction after the coefficients are updated
%
    yp(i)=y(i)+b0*v0(i)+b1*v1(i)+b2*v2(i) ;
end

```

## C.7. Adaptive Predictor: Least Mean Square Algorithm

```
% MATLAB function to implement the adaptive predictor using
% the least mean square algorithm

% Program name: adaptiveLMS.m. Date: September 2003

function yp=adaptiveLMS(y,v,T,td);

% input: y--the sequence of the aircraft state to be predicted
% input: v--the velocity of y
% input: T--the update period (frame length)
% input: td--the time delay
% output: yp--the prediction of y with the least mean square method

[numd,dend]=pade(td,2);      % Pade approximation of td
if td==0
    numd=[0 0 1];
    dend=[0 0 1];
end
%
% to state space form
%
[A,B,C,D]=tf2ss(numd,dend);
[Ad,Bd]=c2d(A,B,T);

[b0,b1,b2]=mcfcoefs(td,T,3); % Calculate McFarland coefficients
%
% Calculate the velocity vectors consisting of 3 steps of velocity
%
v0=v;
for i=1:len4
    if i==1
        v1(i)=v0(i);
        v2(i)=v1(i);
    elseif i==2
        v1(i)=v0(i-1);
        v2(i)=v1(i);
    else
        v1(i)=v0(i-1);
        v2(i)=v1(i-1);
    end
end

% Least mean square approximation algorithm
x=zeros(2,1);
nf=1.2; % gamma
df=10; % alpha
for i=1:length(y)
    x=Ad*x+Bd*y(i);
    yd(i)=C*x+D*y(i);      % calculate the delayed y
%
% use left pseudo-inversion to realize LS regression for 5th iteration
%
    if i==5
        Phi=[v0(1:i)' v1(1:i)' v2(1:i)'];
```

```

    phi=[v0(i) v1(i) v2(i)]';
    Y=y(1:i) '-yd(1:i)';
    P=inv(Phi'*Phi);
    theta=P*Phi'*Y;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
%
% Start the least mean algorithm from 6th iteration
%
elseif i>5
    phi=[v0(i) v1(i) v2(i)]';
    theta=theta+phi*(y(i)-yd(i)-phi'*theta)/lms;
    b0=theta(1);
    b1=theta(2);
    b2=theta(3);
end
%
% Calculate the prediction after the coefficients are updated
%
yp(i)=y(i)+b0*v0(i)+b1*v1(i)+b2*v2(i);
end

```

## C.8. Calculate the McFarland Predictor Coefficients

```
%  
% Function to calculate the McFarland filter's coefficients  
%  
  
function [b0,b1,b2]=mcfcoefs(td,T, f)  
  
% input: td -- time delay (second)  
% input: f -- bandpass frequency (Hz)  
% input: T -- update frame cycle (second)  
% outputs: b0, b1, b2 -- McFarland filter coefficients  
  
w0=2*pi*f;  
t0=w0*T;          % theta0  
p0=w0*td;        % psi0  
  
%  
% intermediate variables  
%  
  
st=sin(t0);  
ct=cos(t0);  
sp=sin(p0);  
cp=cos(p0);  
stp=sin(t0+p0);  
ctp=cos(t0+p0);  
s2tp=sin(2*t0+p0);  
c2tp=cos(2*t0+p0);  
  
%  
% The common denominator  
%  
  
den=2*w0*st*(1-ct);  
  
%  
% The numerators  
%  
  
num0=(p0+sp*(1-2*ct))*st+(t0*st/2-cp*(1-ct))*(1+2*ct);  
num1=st*(2*stp-2*p0*ct-t0*(1+ct));  
num2=st*(p0-sp+t0/2)-cp*(1-ct);  
  
b0=num0/den;  
b1=num1/den;  
b2=num2/den;
```

## C.9. State Space Predictor: 4<sup>th</sup>-Order Reference Model

```
% MATLAB function to implement the state space predictor using
% the 4th order large commercial transport landing model in pitch

% Program name: ssPredictor4.m. Date: November 2003

function yp=ssPredictor4(y,v,a,u,T,td)

% input: y--the sequence of the aircraft state to be predicted
% input: v--the velocity of y
% input: a--the acceleration of y
% input: u--pilot control input
% input: T--the update period (frame length)
% input: td--the time delay
% output: yp--the prediction of y with the state space predictor

%
% Transfor function of a large commercial transport landing model
%
n757L=0.852549*conv([1 0.103205], [1 0.360746]);
w3=0.96671;      xi3=0.66305;
w4=0.155840;    xi4=0.07393;
d757L=conv([1 2*w3*xi3 w3^2], [1 2*w4*xi4 w4^4]);
%
% Change the mdoel into state space form (observable canonical form)
%
A7L=[[-d757L(2) -d757L(3) -d757L(4)]' eye(3);-d757L(5) zeros(1,3)];
B7L=[0 n757L]';
C7L=[1 0 0 0];
D7L=0;
%
% Calculate the state transition matrix P7L and matric Q7L=Q*B7L,
% where Q is the integral of P7L
%
[P7L,Q7L]=c2d(A7L,B7L,td);
%
% Alternative way of calculate the state transition matrix and its
integral
%
P7L=eye(4);
n=15;
f=1;
for i=1:n
    P7L=P7L+(A7L*td)^i/f;
    f=f*(i+1);
end

Q=eye(4);
n=15;
f=1;
for i=1:n
    f=f*(i+1);
    Q=Q+(A7L*td)^i/f;
end
Q=Q*td;
```

```

Q7L=Q*B7L;
%
% Calculate the five coefficients of the predictor
%
c7L0=P7L(1,1)+P7L(1,2)*d757L(2)+P7L(1,3)*d757L(3);
c7L1=P7L(1,2)+P7L(1,3)*d757L(2);
c7L2=P7L(1,3);
c7L3=Q7L(1)-P7L(1,3)*n757L(1);
c7L4=P7L(1,4)*n757L(3);
%
% Calculate the prediction
%
yp=c7L0*y+c7L1*v+c7L2*a+c7L3*u+c7L4*intAdam2(u,T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Adam Bashforth 2nd Order Integtration
%
function z=intAdam2(y,T)

% input y: to be integrated
% input T: sampling period
% output z: integration of y by Adam Bashforth 2nd order

len=length(y);
for i=1:len
    if i==1
        z(i)=y(i)*T;
    elseif i==2
        z(i)=z(i-1)+(y(i-1)+y(i))*T/2;
    else
        z(i)=z(i-1)+(3*y(i-1)-y(i-2))*T/2;
    end
end
end

```

## C.10. Discrete State Space Predictor: 4<sup>th</sup>-Order Reference Model

```
% MATLAB function to implement the discrete state space predictor using
% the 4th order large commercial transport landing model in pitch

% Program name: dssPredictor4.m. Date: December 2003

function yp=dssPredictor4(y,u,T,td)

% input: y--the sequence of the aircraft state to be predicted
% input: u--pilot control input
% input: T--the update period (frame length)
% input: td--the time delay
% output: yp--the prediction of y with the state space predictor

% Transfor function of a large commercial transport landing model
%
n757L=0.852549*conv([1 0.103205], [1 0.360746]);
w3=0.96671;      xi3=0.66305;
w4=0.155840;    xi4=0.07393;
d757L=conv([1 2*w3*xi3 w3^2], [1 2*w4*xi4 w4^4]);
%
% Change the mdoel into state space form (observable canonical form)
%
A7L=[[-d757L(2) -d757L(3) -d757L(4)]' eye(3);-d757L(5) zeros(1,3)];
B7L=[0 n757L]';
C7L=[1 0 0 0];
D7L=0;
[Ald, Bld]=c2d(A7L,B7L,T);      %discretize the model
%
% Calculate the discrete state transition matrix Pd and matrix Qd=Q*Bld,
% where Q is the integral of Pd
%
Pd=Ald^nd;      % Discrete state transition matrix
Qd=eye(4);
for i=1:nd-1
    Qd=Qd*Ald+eye(4);
end
Qd=Qd*Bld;      % Integral of the discrete state transition matrix
%
% Calculate predictor state vector
%
xd1(1)=y(1);
xd2(1)=0;
xd3(1)=0;
xd4(1)=0;
for i=1:lennngth(y)-1
    xd2(i+1)=Ald(2,1)*xd1(i)+xd3(i)+Bld(2)*u(i);
    xd3(i+1)=Ald(3,1)*xd1(i)+xd4(i)+Bld(3)*u(i);
    xd4(i+1)=Ald(4,1)*xd1(i)+Bld(4)*u(i);
    xd=[xd1(i+1);xd2(i+1);xd3(i+1);xd4(i+1)];

    xdp=Pd*xd+Qd*u(i); % Calculate the prediction
    yp(i+1)=C7L*xdp;
end
```

## C.11. Calculate the Accelerations in 6 DOF of the Topodetic Frame System

```
% MATLAB scripts to calculate the accelerations in the 6 DOF of the
% topodetic frame from the Euler angles, p, q, r, u, v, w, pd,
% qd, rd, altitude, latitude and longitude.

% Program name: topodeticAcc.m. Date: June 2004

clear all

% load data of the Euler angles, p, q, r, u, v, w, and pd, qd and rd.

load testrun13 % could be any data file containing the above data
%
% Calculate the accelerations of the Euler angles
%
phi_dd=pd+tan(the).*sin(phi).*qd+tan(the).*cos(phi).*rd;
the_dd=cos(phi).*qd-sin(phi).*rd;
psi_dd=sin(phi)./cos(the).*qd+cos(phi)./cos(the).*rd;
%
% calculate the projection on the equatorial plane of the geocentric
% radius of a point
%
a=2.092565e+7; % Earth equatorial radius
f=1/298.257; % Earth flattening parameter
b=a*(1-f); % Earth polar radius
c=sqrt(a^2-b^2);
e=c/a; % Earth eccentricity
e2=(a^2-b^2)/a^2;
C=1./sqrt(cos(lat).^2+sin(lat).^2*b^2/a^2);
rp=(a*C+alt).*cos(lat); % projected radius
rp=rp/3.2808; % convert to meters
re=alt/3.2808+6.3782e+006; % (radius of the Earth, in meter)
re=(alt+rho)/3.2808; % (radius of the Earth, in meter)
%
% calculate rho
%
drho=(b^2+(a^2-b^2)*cos(lat).^2).^1.5;
rho=a^2*b^2./drho;
%
% calculate the velocities of the altitude, latitude and longitude
%
pnd=u.*cos(the).*cos(psi)-v.*cos(phi).*sin(psi)
+v.*sin(phi).*sin(the).*cos(psi)+w.*sin(phi).*sin(psi)
+w.*cos(phi).*sin(the).*cos(psi);
ped=u.*cos(the).*sin(psi)+v.*cos(phi).*cos(psi)
+v.*sin(phi).*sin(the).*sin(psi)-w.*sin(phi).*cos(psi)
+w.*cos(phi).*sin(the).*sin(psi);
hd=u.*sin(the)-v.*sin(phi).*cos(the)-w.*cos(phi).*cos(the);
%
%Change to angular latitude and longitude velocities
%
pnd=pnd./re;
ped=ped./rp;
%
% Calculate the translational accelerations
```



```

%
% 1. part a
%
pna1=ud.*cos(the).*cos(psi)-vd.*cos(phi).*sin(psi)
      +vd.*sin(phi).*sin(the).*cos(psi)+wd.*sin(phi).*sin(psi)
      +wd.*cos(phi).*sin(the).*cos(psi);          % in m/s/s

pea1=ud.*cos(the).*sin(psi)+vd.*cos(phi).*cos(psi)
      +vd.*sin(phi).*sin(the).*sin(psi)-wd.*sin(phi).*cos(psi)
      +wd.*cos(phi).*sin(the).*sin(psi);          % in m/s/s

ha1 =ud.*sin(the)-vd.*sin(phi).*cos(the)-wd.*cos(phi).*cos(the);
%
% 2. part b
%
pna2=(v.*cos(phi).*sin(the).*cos(psi)+v.*sin(phi).*sin(psi)
      +w.*cos(phi).*sin(psi)-w.*sin(phi).*sin(the).*cos(psi)).*phid
      +(-u.*sin(the).*cos(psi)+v.*sin(phi).*cos(the).*cos(psi)
      +w.*cos(phi).*cos(the).*cos(psi)).*thed+(-u.*cos(the).*sin(psi)
      -v.*sin(phi).*sin(the).*sin(psi)-v.*cos(phi).*cos(psi)
      +w.*sin(phi).*cos(psi)-w.*cos(phi).*sin(the).*sin(psi)).*psid;

pea2=(v.*cos(phi).*sin(the).*sin(psi)-v.*sin(phi).*cos(psi)
      -w.*cos(phi).*cos(psi)-w.*sin(phi).*sin(the).*sin(psi)).*phid
      +(-u.*sin(the).*sin(psi)+v.*sin(phi).*cos(the).*sin(psi)
      +w.*cos(phi).*cos(the).*sin(psi)).*thed...
      +(u.*cos(the).*cos(psi)+v.*sin(phi).*sin(the).*cos(psi)
      -v.*cos(phi).*sin(psi)+w.*sin(phi).*sin(psi)
      +w.*cos(phi).*sin(the).*cos(psi)).*psid;

ha2=(-v.*cos(phi).*cos(the)+w.*sin(phi).*cos(the)).*phid...
      +(u.*cos(the)+v.*sin(phi).*sin(the)+w.*cos(phi).*sin(the)).*thed;

pna=pna1+pna2;
pea=pea1+pea2;
ha=ha1+ha2;
pna=pna./re;      % in rad/s/s
pea=pea./rp;      % in rad/s/s
ha=ha*3.2808;

% phi_dd is the phi acceleration
% the_dd is the theta acceleration
% psi_dd is the psi accelerationm
% pnd is the latitude velocity
% ped is the longitude velocity
% hd is the altitude velocity
%
% pna is the latitude velocity
% pea is the longitude velocity
% ha is the altitude velocity
%

```

## C.12. Calculate State Transition Matrix Using the Cayley-Hamilton Theorem

```
% MATLAB function to calculate the state transition matrix using
% the Cayley-Hamilton theorem, which is introduced in Appendix A,
% NASA CR 2007-215095.
```

```
% Program name: dssPredictor4.m. Aug 10, 2001
```

```
function z=stCH(A,td)
```

```
% input A: a square matrix
% input td: time delay
% output z: state transitionmatrix of Atd-->z=exp(A*td)
```

```
lamda=eig(A); % eigenvalues of am
sd=sort(lamda); % sort lamda in ascending order
len=length(sd); % the order of A
```

```
%
% Check if there are repeated eigenvalues.
```

```
%
nrep=ones(len,1);
np=zeros(len,1);
i=1;
while i<len
    if sd(i)==sd(i+1)
        tmp=1;
        m=i+1;
        while sd(m)==sd(i)
            np(m)=m-i;
            tmp=tmp+1;
            m=m+1;
            if m>len
                break
            end
        end
        for k=i:i+tmp-1
            nrep(k)=tmp;
        end
        i=i+tmp;
    else
        i=i+1;
    end
end
```

```
%
% Form the vector of the powers of the eigenvalues
%
```

```
for i=1:len
    fp(i)=td^np(i)*exp(sd(i)*td)/factorial(np(i));
end
```

```
%
% Calculate the powers of the matrix A
%
```

```
Pa(1)=eye(len);
for i=2:len
    Pa(i)=A*Pa(i-1);
end
```

```

%
% Quasi-Vandemonde matrix
%
for i=1:len
    if np(i)==0
        for m=1:len
            qv(i,m)=sd(i)^(m-1);
        end
    else
        for m=1:np(i)
            qv(i,m)=0;
        end
        for m=np+1:len
            c=1;
            for n=m-1:-1:m-np(i)
                c=c*n;
            end
            qv(i,m)=c*sd(i)^(m-np(i)-1);
        end
    end
end
end
%
% Calculate the inverse of qv
iqv=inv(qv);
if abs(imag(iqv))<0.0001
    iqv=real(iqv);
end
%
% Calculate the alphas.
%
alpha=iqv*fp';
if abs(imag(alpha))<0.0001
    alpha=real(alpha);
end
%
% finally , calculate the state transition matrix
%
z=zeros(len,len);
for i=1:len
    z=z+alpha(i)*Pa(i);
end
end

```

### C.13. Comprehensive Analysis of Sobiski/Cardullo Filter

```
% This program does the following designs and analyses:
% 1. Feedforward Sobiski/Cardullo predictor analysis;
% 2. Discrete state space compensation;
% 3. Five-point method to design the feedforward gain;
% 4. Least square fit method to design the feedforward gain;
% 5. Observaor design;
% 6. Analyze a S/C compensation system with an observer;

% Program name: scgAnalysis.m. Date: April 2001

clear all

% Jet aircraft model from Ricard's paper in roll
%
njr=1.1644*[1/3.46 0.48/1.86 1];
djr=conv(conv([1 0], [0.16 1]), [1/3.53 0.48/1.88 1]);
djr=djr/djr(1);
%
% Pilot model from Sobiski's thesis
%
nump=18*[1 1];
denp=[1 9 18];
x=-.3; % pilot lumped delay
numt=[x*x 6*x 12];
dent=[x*x -6*x 12];
[nump,denp]=series(nump,denp, numt,dent); % whole pilot model
%
% Artificial time delay
%
td=0.4;
[numt,dent]=pade(td,2);
%
% Change models into state space format
%
[Aa Ba Ca Da]=tf2ss(njr,djr);
[Ap Bp Cp Dp]=tf2ss(nump,denp);
[At Bt Ct Dt]=tf2ss(numt,dent);
%
% Cascade models together
%
[An Bn Cn Dn]=series(Aa,Ba,Ca,Da,Ap,Bp,Cp,Dp); % without added delay
[Ad Bd Cd Dd]=series(An,Bn,Cn,Dn,At,Bt,Ct,Dt); % with added delay
sysn=ss(An,Bn,Cn,Dn);
sysd=ss(Ad,Bd,Cd,Dd);
%
% Feed forward compensation with state transition matrix
%
[P,Q]=c2d(Ad,Bd,td); % P=exp(Ad*td), Q=int(P)*Bd;
sysc=ss(Ad,Bd,Cd*P,Cd*Q); % with compensation
%
% Frequency analysis
%
w1 = 0.1:0.1:1;
w2 = 2:1:20;
```

```

w = [w1 w2]; % frequency interval (rad/s) for bode: open loop
[mn,pn] = bode(sysn, 1, w);
[md,pd] = bode(sysd, 1, w);
[mc,pc] = bode(sysc, 1, w);

%
% plot Bode diagrams
%
figure(1)
subplot(211);
semilogx(w, 20*log10(mn), '-', w, 20*log10(md), '--', w, 20*log10(mc), '.');
xlabel('Frequency, rad/s');
ylabel('Magnitude, dB');
ttl=['Bode Diagrams of Systems without Delay, ' num2str(td1)...
     's Delay, Sobiski/Cardullo Compensation'];
title(ttl);
grid;
axis([0.1 max(w) -60 40])
lgd2=['With ' num2str(td1) 's Delay'];
legend('With no delay', lgd2, 'Sobiski/Cardullo Compensation',3)
subplot(212);
semilogx(w, pn, '-', w, pd, '--', w, pc, '.'), grid;
xlabel('Frequency, rad/s');
ylabel('Phase, deg');
axis([0.1 max(w) -1200 0])
legend('With no delay', lgd2, 'Sobiski/Cardullo Compensation',3)
%
% Step responses: closed loop
%
T=1/60; % frame length
tall=10; % total time for step response
t=0:T:tall;
lent=length(t);
u=ones(1,lent); % step input
%
sysnc=ss(An-Bn*Cn,Bn,Cn,Dn); % closed-loop undelayed system
sysdc=ss(Ad-Bd*Cd,Bd,Cd,Dd); % closed-loop delayed system
G=inv(1+Cd*Q); % feedback gain with compensation
syscc=ss(Ad-Bd*G*Cd*P,Bd*G, Cd*P,Cd*Q); % closed-loop compensated
system
%
yn=lsim(sysnc,u,t); % step response of undelayed system
yd=lsim(sysdc,u,t); % step response of delayed system
yc=lsim(syscc,u,t); % step response of compensated system

%
% Plot step responses
%
figure(2)
plot(t,yn,'r', t,yd,'g-.', t,yc,'b--')
title('Step Responses of Undelayed, Delayed and Compensated (S/C)
Systems')
xlabel('t, s')
ylabel('Roll Angle, rad')
set(gca,'FontSize',8);
lgd=['Delayed by ' num2str(td1) 's'];
legend('Undelayed', lgd, 'Compensated', 3)

```

```

%
% Discrete Sobiski/Cardullo filter implementation
%
[Az,Bz]=c2d(Ad,Bd,T); % Discretize the system
nd=floor(td/T);
tau=td-nd*T; % td=nd*T+tau;
Pz=Az^nd; % Discrete state transition matrix
I=eye(length(Az));
Qz=I; % eye creates identity matrix
for i=1:nd-1
    Qz=Qz*Az+I;
end
Qz=Qz*Bz; % Integral of the discrete state transition matrix
%
sysz=ss(Az,Bz,Cd*Pz,Cd*Qz); % Discretely compensated system
if tau~=0
    [Ptau, Qtau]=c2d(Ad,Bd,tau);
    sysz=ss(Az,Bz,Cd*Ptau*Pz,Cd*(Ptau*Qz+Qtau));
end
[mz,pz] = bode(sysz, 1, w); % Bode diagram with discrete S/C filter
syszc=feedback(sysz,-1); % Closed loop of sysz
yz=lsim(syszc,u,t); % step response of discretely compensated system

%
% 5-point method to design the feedback gain
%
s = [0.1 1 6 10 17]*j; % 5 frequency points
I = eye(length(An));
for m=1:length(s)
    hn(m) = Cn*inv(s(m)*I-An)*Bn + Dn;
end
hnr = real(hn);
hni = imag(hn);
b = [hnr' ; hni'];
%
I = eye(length(Ad));
for m=1:length(s)
    ina(:, :, m) = P*inv(s(m)*I-Ad);
    inab(:, m) = ina(:, :, m)*Bd;
    inabd(:, m) = inab(:, m) + Q;
end
inabdr = real(inabd);
inabdi = imag(inabd);
a = [inabdr' ; inabdi'];
K5 = (a\b)'; % K5 is the feed forward gain by the 5-P method
%
sysc5=ss(Ad,Bd,K5*P,K5*Q); % compensated system with feedback gain K5
[mc5,pc5] = bode(sysc5, 1, w); % frequency response
G5=inv(1+K5*Q); % feedback gain with compensation
syscc5=ss(Ad-Bd*G5*K5*P,Bd*G, K5*P,K5*Q); % closed-loop compensated
system
yc5=lsim(syscc5,u,t); % step response of compensated system

%
% Least squares method to design the feedback gain
%

```

```

s=w*j;
I = eye(length(Ad));
for m=1:length(s)
    p1(:, :, m) = P*inv(s(m)*I-Ad);
    p2(:, m) = p1(:, :, m)*Bd;
    p(:, m) = p2(:, m) + Q;
end
Pr = real(p);
Pi = imag(p);
%
for k=1:length(Ad)
    ar(:, k) = Pr*(Pr(k, :)' );
    ai(:, k) = Pi*(Pi(k, :)' );
    br(k) = hnr*(Pr(k, :)' );
    bi(k) = hni*(Pi(k, :)' );
end
a = ar+ai;
b = br' + bi';
Kf = (a\b)'; % Kf is the feed forward gain by the lsf method
%
syscf=ss(Ad,Bd,Kf*P,Kf*Q); % compensated system with feedback gain Kf
[mcf,pcf] = bode(syscf, 1, w); % frequency response
Gf=inv(1+Kf*Q); % feedback gain with compensation
sysccf=ss(Ad-Bd*Gf*Kf*P,Bd*G, Kf*P,Kf*Q); % closed-loop compensated
system
ycf=lsim(sysccf,u,t); % step response of compensated system

%
% Observer design
%
pnd = eig(As - Bs*Cs)';
pndr = real(pnd)*1.2;
pndl = pndr+j*imag(pnd);
pobsv = [pndl real(delaypol')];
L = plfun2(Ad', Cd', pobsv); % plfun2 is available in the next
section
L = real(L);
L = L'; % L is the observer gain

%
% Observer analysis
%
Ar=Ad-L*Cd; % Ar is the Matrix Ac in Eq. (B.10), NASA CR-2005
Ao=[Ad, -Bd*G*Cd*P; L*Cd, Ar-Bd*G*Cd*P];
Bo=[Bd*G; Bd*G];
Co=[Cd*P-Cd*Q*G*(Cd*P), zeros(size(Cd*P))];
Do=Cd*Q*Bd*G;
syso=ss(Ao,Bo,Co,Do); % the observer system
yco=lsim(syso,u,t); % step response of compensated system with an
observer
syse=ss(Ar,Bd*(1-G),Cd, 0); % Error system with the Observer
e=lsim(syse,u,t); % error response

```

## C.14. A Pole Placement Function Called by the Program in Appendix C.12

```
% this function implements the pole placement
% for the case: 1. SISO, 2. repeated poles:
% one pole repeats once.
% It is more robust than the MATLAB place function, which does not
% accept repeated Poles

% Program name: plfun2.m. Date: May 2001

function y=plfun2(a,b,L)

% input: matrices a and b, pole vector L
% the repeated poles must be in the end of L
% output: state feedback control matrix k

ord=length(b);
for i=1:ord-1
    temp = [L(i)*eye(ord)-a];
    if rank(temp)==ord
        ai=temp;
        evt = ai\b;
        ka(:,i)=evt;
        kb(i)=-1;
        %kb(i)=last;
    else
        for m=1:ord-1
            ai(:,m)=temp(:,m+1);
        end
        ai(:,ord)=b;
        b1=temp(:,1);
        evt = ai\b1;
        for m=1:ord-1
            ka(m+1,i)=evt(m);
        end
        ka(1,i)=-1;
        kb(i)=evt(ord);
    end
end

t1 = [L(ord-1)*eye(ord)-a b];
% t1 is the [lamda*I-a b] matrix, where lamda is the repeated pole.
[m,n] = size(t1);
for i=2:n
    t2(:,i-1)=t1(:,i);
end
t3=ka(:,ord-1)-t1(:,1);
t=t2\t3;
ka(1,ord)=1;
for i=2:ord ka(i,ord)=t(i-1); end
kb(ord)=t(ord);
y=kb*inv(ka);
```



### C.15. Design a 3-Velocity Predictor Using Frequency Least Squares Method

```
% MATLAB program to design a 3-velocity predictor using
% frequency domain least squares method

% Program name: flsMcf.m. Date: July 2002

clear

% Jet aircraft model from Ricard's paper->output: roll velocity (P)
%
nac=1.1644*[1/3.46 0.48/1.86 1];
dac=conv([0.16 1], [1/3.53 0.48/1.88 1]);
dac=dac/dac(1);
T=0.016; % sampling period
[nad,dad]=c2dm(nac,dac,T); % discretize the A/C model
%
% EOM
%
nv2u=T*[1 1]/2; % trapezoid integration from velocity to acceleration
dv2u=[1 -1];
%
% Pilot model from Sobiski's thesis
%
nump=18*[1 1];
denp=[1 9 18];
x=-.3; % pilot lumped delay
numt=[x*x 6*x 12];
dent=[x*x -6*x 12];
[nopc,dopc]=series(nump,denp, numt,dent); % whole pilot model
%
% Artificial time delay
%
td=0.2;
[ndlyc,ddlyc]=pade(td,2);
[ndlyd,ddlyd]=c2dm(ndlyc,ddlyc,T,'zoh');
%
% cascade
%
[nvoc,dvoc]=series(nopc,dopc,nac,dac); % continuous A/C (P) & operator
[nvod,dvod]=c2dm(nvoc,dvoc,T); % discrete A/C (P) & operator
[nuoc,duoc]=series(nvoc,dvoc, 1,[1 0]); % continuous A/C & operator
[nuod,duod]=series(nvod,dvod, nv2u,dv2u); % discrete A/C & operator
[nallc, dallc]=series(nuoc,duoc, ndlyc,ddlyc); % continuous: all
[nalld, dalld]=series(nuod,duod, ndlyd,ddlyd); % discrete: all
[A0,B0,C0,D0]=tf2ss(nuoc,duoc); % State space: continuous A/C & Op
%
% Frequency analysis
%
w1 = 0.1:0.1:1;
w2 = 2:1:20;
w = [w1 w2]; % frequency interval (rad/s) for bode: open loop
s=w*j;
len=length(w);
%
% evaluate z the z-transform operator
```

```

z=exp(j*w*T);
zi=1./z;
%
[m0,p0]=bode(nuoc,duoc,w);% frequency response (undelayed)
[md,pd]=bode(nuod,duod,w); % frequency response (delayed)
%
% Add the McFarland compensator
%
[b0,b1,b2]=mcfcoefs(td,T,3); % Calculate McFarland coefficients
%
nmcfb=[b0 b1 b2];
dmcfb=[1 0 0];
npara=nv2u;
dpara=dv2u;
nt1=conv(nmcfb,dpara);
nt2=conv(dmcfb,npara);
len1=length(nt1);
len2=length(nt2);
if len1==len2
    npara=nt1+nt2;
else if len1>len2
    npara=nt1+[zeros(len1-len2) nt2];
    else
    npara=nt2+[zeros(len2-len1) nt1];
end
end
dpara=conv(dmcfb,dpara);
[nf, df]=series(nvod,dvod, npara,dpara);
[nf, df]=series(nf,df, ndlyd,ddlyd);
lenn=length(nf);
lend=length(df);
for m=1:len
    sumn=0;
    sumd=0;
    for k=1:lenn
        sumn=sumn+nf(k)*z(m)^(lenn-k);
    end
    for k=1:lend
        sumd=sumd+df(k)*z(m)^(lend-k);
    end
    hz(m)=sumn/sumd;
end
mf=abs(hz);
pf=unwrap(angle(hz))*180/pi;
%
% design a K=[b0 b1 b2], using the least square method
%
I = eye(length(A0));
for m=1:len
    h0(m) = C0*inv(s(m)*I-A0)*B0 + D0;
end
% h0 is the frequency characteristic function of undelayed system
%
lenn=length(nvod);
lend=length(dvod);
for m=1:len
    sumn=0;

```

```

    sumd=0;
    for k=1:lenn
        sumn=sumn+nvod(k)*z(m)^(lenn-k);
    end
    for k=1:lend
        sumd=sumd+dvod(k)*z(m)^(lend-k);
    end
    hs(m)=sumn/sumd;
end
% hs is the frequency characteristic function of A/C + operator
%
% least square design
%
tpa(1,:)=hs;
tpa(2,:)=hs.*zi;
tpa(3,:)=hs.*zi.*zi;
tpb=h0-T*hs.*(z+1)./(z-1)/2;
tpar = real(tpa);
tpai = imag(tpa);
tpbr = real(tpb);
tpbi = imag(tpb);
ar=tpar*tpar';
br=tpar*tpbr';
ai=tpai*tpai';
bi=tpai*tpbi';
a=ar+ai;
b=br+bi;
K=a\b; % K contains the new set of predictor coefficients
nmcf=K';
dmcf=[1 0 0];
%
npara=nv2u;
dpara=dv2u;
nt1=conv(nmcf,dpara);
nt2=conv(dmcf,npara);
len1=length(nt1);
len2=length(nt2);
if len1==len2
    npara=nt1+nt2;
else if len1>len2
    npara=nt1+[zeros(len1-len2) nt2];
else
    npara=nt2+[zeros(len2-len1) nt1];
end
end
dpara=conv(dmcf,dpara);
[nlsf, dlsf]=series(nvod,dvod, npara,dpara);
[nlsf, dlsf]=series(nlsf,dlsf, ndlyd,ddlyd);
%
lenn=length(nlsf);
lend=length(dlsf);
for m=1:len
    sumn=0;
    sumd=0;
    for k=1:lenn
        sumn=sumn+nlsf(k)*z(m)^(lenn-k);
    end
end

```

```

    for k=1:lend
        sumd=sumd+dlsf(k)*z(m)^(lend-k);
    end
    hz(m)=sumn/sumd;
end
mlsf=abs(hz);
plsf=unwrap(angle(hz))*180/pi;
%
% Bode plot
%
figure(1)
subplot(2, 1, 1);
p=semilogx(w, 20*log10(m0),'r',w, 20*log10(mf),'k--',
w,20*log10(mlsf),'k. ');
set(p,'LineWidth',1)
xlabel('Frequency - rad');
ylabel('Magnitude - dB');
axis([0.1 20 -60 40]);
ttl=['Bode Diagrams of Systems with no Delay, with McFarland
Compensations'];
title(ttl);
grid;
%lgd2=['With ' num2str(Td) ' Delay'];
legend('Undelayed','McFarland','3-V: Least Squares',0)
subplot(2, 1, 2);
semilogx(w, p0,'r',w, pf,'k--', w,plsf,'k. ');
xlabel('Frequency - rad');
ylabel('Phase - deg');
grid;
legend('Undelayed','McFarland','3-V: Least Squares',0)
axis([0.1 20 -600 0]);

```

## C.16. Function to Plot Multi-Normalized Data for the Delay Measurement

```
% This program primarily was written by Tom Wolters, and Liwen Guo made
% some chnages.
```

```
% Program name: plfun2.m. Date: May 2000
```

```
function nplot_data(var,nvar,tstart,tend,tstring)
```

```
% NPLOT_DATA Plots a normalized version of the given variable/s with
% respect to time. In the legend, the range of a variable
% with unit follows the variable name. The optional tstart
% and tend limit the time of the plot. Tstring adds a
% title.
```

```
% Other input:
```

```
% var—name of the data structure referencing the data
```

```
% nvar—sequence holding the indices of the data items in the structure
```

```
style={'b-', 'r-', 'g-', 'c-', 'm-', 'y-', 'b--', 'r--', 'g--', ...
      'c--', 'm--', 'y--'};
```

```
for i=1:length(nvar)
```

```
% usually = 4, nvar is seti, i=1~5, i!=4,sometimes, =3, ie,set4
```

```
if(nargin<4) % usually = 5
```

```
    t2 = length(var(nvar(i)).time);
```

```
else
```

```
    t2 = max(find(var(nvar(i)).time<tend));
```

```
end
```

```
%
```

```
if(nargin<3)
```

```
    t1 = 1;
```

```
else
```

```
    t1 = min(find(var(nvar(i)).time>tstart));
```

```
end
```

```
%
```

```
t = [t1:t2]; % time interval is formed
```

```
%
```

```
bias(i) = min(var(nvar(i)).data(t));% for normalizing.
```

```
scale(i) = max(var(nvar(i)).data(t)) - bias(i);
```

```
if(scale(i)==0)
```

```
    scale(i) = 1; % avoiding dividing by 0
```

```
end
```

```
%
```

```
nstyle = mod(i-1,6)+1;
```

```
plot(var(nvar(i)).time(t), (var(nvar(i)).data(t)-...
```

```
    bias(i))/scale(i), style{nstyle})
```

```
hold on
```

```
end
```

```
%
```

```
% Add labels, title, grid and legend
```

```
%
```

```
set(gca,'FontSize',8); % font of the 2 axes
```

```
grid on
```

```
xlabel('Time (sec)');
```

```
if ( length(nvar) == 1 )
```

```

        ylabel([var(nvar(i)).name, ' (', var(nvar(i)).unit, ') ']);
else
    ylabel('Data')
    for i=1:length(nvar)
        mintext = num2str(bias(i));
        maxtext = num2str(bias(i)+scale(i));
%
        legendtext{i}=[var(nvar(i)).name, ' (',mintext,' to ',...
            maxtext, ' ', var(nvar(i)).unit, ') '];
    end
    [hnd,ohnd] = legend(legendtext,0); % forming the legend parameters
    len=length(ohnd);
    for j=1:len
        if ( get(ohnd(j),'Type') == 'text' )
            set(ohnd(j),'FontSize',10);
        end
    end
    legend(hnd) % create the legend

end
%
if(nargin==5)
    title(tstring);
end
hold off

```

## C.17. Calculate the Platform Accelerations from Accelerometers Readings

```
% MATLAB program to calculate the accelerations of the motion
% platform in 6-DOF from six accelerometers mounted on the
% platform as shown in Fig. 2.3

% Program name: accPlatform.m. Date: Feb. 2000

clear all

fid=fopen('12_17Rollt.txt','r');
% 12_17Rollt.txt is a text data file with 12 columns, the last 6
% of which store the readings from the 6 accelerometers
%
a = fscanf(fid,'%f',[12,2401]);
fclose(fid);
%
% positions of the accelerometers
%
x1=-14.5;   y1=91.375;   z1=-13.375;
x2=-5.5;    y2=0;       z2=-9.75;
x3=-95;     y3=-93.125;  z3=-12.8125;
x4=111.875; y4=0;       z4=-12.25;
x5=0;       y5=91.375;  z5=-13.5;
x6=-14.5;   y6=-91.375; z6=-13.5;
%
A=[1 0 0 0  z1  -y1;
   0 1 0 -z2  0  x2;
   0 0 1 y3 -x3  0;
   0 0 1 y4 -x4  0;
   0 0 1 y5 -x5  0;
   1 0 0 0  z6  -y6;];
%
dt=0.025;
tmax=6*0.628-dt;
t=0:dt:tmax;
n=length(t);
%
for i=1:n
    for j=1:6
        bt(j)=a(j+6,i)*192.9175;
    end
    B=bt';
    C=A\B;
    roll(i)=C(4)*57.296;
end
%
plot(t, roll),grid
ylabel('Roll Acc., deg/s^2');
xlabel('Time t ,s');
title('Roll Acceleration vs Time for 6 Periods');
```

## C.18. Realization of Inverse Newton-Raphson Transformation

```
% This m-file is for Newton-Raphson inverse
% transform, i.e., calculate simulator 6 DOF's
% from the actuators' extension.
```

```
% Program name: inverseNR.m. Date: March 2000
```

```
clear all
```

```
% Initialize Parameters
```

```
leneut=3.2649;
```

```
aaais(1,1)=2.1117179;
aaais(2,1)=0.0762;
aaais(3,1)=0;
aaais(1,2)=2.1117179;
aaais(2,2)=-0.0762;
aaais(3,2)=0;
aaais(1,3)=-0.98986594;
aaais(2,3)=-1.8669;
aaais(3,3)=0;
aaais(1,4)=-1.12184942;
aaais(2,4)=-1.7907;
aaais(3,4)=0;
aaais(1,5)=-1.12184942;
aaais(2,5)=1.7907;
aaais(3,5)=0;
aaais(1,6)=-0.98986594;
aaais(2,6)=1.8669;
aaais(3,6)=0;
```

```
bbii(1,1)=1.5021179;
bbii(2,1)=1.9812;
bbii(3,1)=2.58064;
bbii(1,2)=1.5021179;
bbii(2,2)=-1.9812;
bbii(3,2)=2.58064;
bbii(1,3)=0.96471232;
bbii(2,3)=-2.29147116;
bbii(3,3)=2.58064;
bbii(1,4)=-2.46682768;
bbii(2,4)=-0.31027116;
bbii(3,4)=2.58064;
bbii(1,5)=-2.46682768;
bbii(2,5)=0.31027116;
bbii(3,5)=2.58064;
bbii(1,6)=0.96471232;
bbii(2,6)=2.29147116;
bbii(3,6)=2.58064;
```

```
for i = 1 : 6
    xm(i) = aaais(1,i);
    ym(i) = aaais(2,i);
    zm(i) = aaais(3,i);
```



```

    xs(i) = bbii(1,i);
    ys(i) = bbii(2,i);
    zs(i) = bbii(3,i);
%   raml(i) = rli(i)+leneut;
end
%
% Initialize Simulator Position.
%
t=0;
s=0;
p=0;
x=0;
Y=0;
z=0;
%
% Set up the 'while' loop criteria
%
% it=0;                                % # of steps
zlim1=0.01;                             % step difference criterion for
translational DOFs.
zlim2=0.1/57.296;                         % step difference criterion for angular
DOFs.
%
% Set up initial to some values greater than the criteria
%
fid=fopen('rmod.txt','r');
aa = fscanf(fid,'%f',[6,2401]);
fclose(fid);
%
dt=0.025;
tmax=6*0.628;
time=0:dt:(tmax-dt);
n=length(time);
%
for ii=1:n
    for j=1:6
        rli(j)=aa(j,ii)*0.0254;
        raml(j) = rli(j)+leneut;
    end
%
    max1=100;
    max2=100;
%
% start the loop of computation
%
nstep=0;
while ((max1 > zlim1) | (max2 > zlim2))
    a(1,1) = cos(s)*cos(t);
    a(1,2) = sin(s)*cos(t);
    a(1,3) = -sin(t);
    a(2,1) = cos(s)*sin(t)*sin(p) - sin(s)*cos(p);
    a(2,2) = sin(s)*sin(t)*sin(p) + cos(s)*cos(p);
    a(2,3) = cos(t)*sin(p);
    a(3,1) = cos(s)*sin(t)*cos(p) + sin(s)*sin(p);
    a(3,2) = sin(s)*sin(t)*cos(p) - cos(s)*sin(p);
    a(3,3) = cos(t)*cos(p);
%

```

```

for i = 1 : 6
    f(i)=xm(i)^2 + ym(i)^2 + zm(i)^2 + xs(i)^2 + ys(i)^2...
        +zs(i)^2 + x^2 + y^2 + z^2 - raml(i)^2...
        +2*(x-xs(i))*(xm(i)*a(1,1)+ym(i)*a(2,1)+zm(i)*a(3,1))...
        +2*(y-ys(i))*(xm(i)*a(1,2)+ym(i)*a(2,2)+zm(i)*a(3,2))...
        +2*(z-zs(i))*(xm(i)*a(1,3)+ym(i)*a(2,3)+zm(i)*a(3,3))...
        -2*(x*xs(i)+y*ys(i)+z*zs(i));
%

    pfx(i)=2*(x+xm(i)*a(1,1)+ym(i)*a(2,1)+zm(i)*a(3,1)-xs(i));
    pfy(i)=2*(y+xm(i)*a(1,2)+ym(i)*a(2,2)+zm(i)*a(3,2)-ys(i));
    pfz(i)=2*(x+xm(i)*a(1,3)+ym(i)*a(2,3)+zm(i)*a(3,3)-zs(i));
    pfs(i)=-2*(x-xs(i))*(xm(i)*a(1,2)+ym(i)*a(2,2)+zm(i)...
        *a(3,2)+2*(y-ys(i))*(xm(i)*a(1,1)+ym(i)*a(2,1)+zm(i)...
        *a(3,1));
%

    pft(i) =2*(x-xs(i))*(-xm(i)*sin(t)*cos(s)+ym(i)*sin(p)...
        *cos(t)*cos(s)+zm(i)*cos(p)*cos(t)*cos(s))+2*(y-...
        ys(i))*(-xm(i)*sin(t)*sin(s)+ym(i)*sin(p)*cos(t)...
        *sin(s)+zm(i)*cos(p)*cos(t)*sin(s))-2*(z-zs(i))...
        *(xm(i)*cos(t)+ ym(i)*sin(p)*sin(t)+zm(i)*cos(p)...
        *sin(t));
%

    pfp(i)=2*(x-xs(i))*(ym(i)*a(3,1)-zm(i)*a(2,1))+2*(y-...
        ys(i))*(ym(i)*a(3,2)-zm(i)*a(2,2))+2*(z-zs(i))...
        *(ym(i)*a(3,3)-zm(i)*a(2,3));
end
%

a = [pfx pfy pfz pfs pft pfp];
n=6;
tol=0.;
ks=0;
jj=-n;
j=1;
condition=0;
freturn=0;
while ((j<=n) & (condition == 0))
    jy=j+1;
    jj=jj+n+1;
    biga=0.;
    it=jj-j;
    for i=j:n
        ij=it+i;
        if abs(biga)-abs(a(ij))<0
            biga=a(ij);
            imax=i;
        end
    end
    if abs(biga)-tol<=0
        condition=1;
        freturn=1;
    else
        i1=j+n*(j-2);
        it=imax-j;
        for k=j:n
            i1=i1+n;
            i2=i1+it;
            save=a(i1);

```

```

        a(i1)=a(i2);
        a(i2)=save;
        a(i1)=a(i1)/biga;
    end
    save=f(imax);
    f(imax)=f(j);
    f(j)=save/biga;
    if (j~=n)
        iqs=n*(j-1);
        for ix=jy:n
            ixj=iqs+ix;
            it=j-ix;
            for jx=jy:n
                ixjx=n*(jx-1)+ix;
                jjx=ixjx+it;
                a(ixjx)=a(ixjx)-a(ixj)*a(jjx);
            end
            f(ix)=f(ix)-f(j)*a(ixj);
        end
    else
        condition=1;
    end
end
j=j+1;
end
if freturn==0
    ny=n-1;
    it=n*n;
    for j=1:ny
        ia=it-j;
        ib=n-j;
        ic=n;
        for k=1:j
            f(ib)=f(ib)-a(ia)*f(ic);
            ia=ia-n;
            ic=ic-1;
        end
    end
else
    ks=1;
end
%
if ks == 1
    disp('Matrix is singular');
    break;
else
    x = x - f(1);
    y = y - f(2);
    z = z - f(3);
    s = s - f(4);
    t = t - f(5);
    p = p - f(6);
    nstep = nstep+1;
    if nstep == 51
        break;
    else
        abs1=[abs(f(1)), abs(f(2)), abs(f(3))];
    end
end

```

```

                                abs2=[abs(f(4)), abs(f(5)), abs(f(6))];
                                max1=max(abs1);
                                max2=max(abs2);
                                end
                                end
                                end
%
    xp(ii)=x;
    yp(ii)=y;
    zp(ii)=z;
    tp(ii)=t;
    sp(ii)=s;
    pp(ii)=p;
%
    rec_step(ii)=nstep;
end
dof=[x y z p t s]';
pp=pp*57.296;
k=0.315*0.0254*57.296; %k=0.008
phi=3.305;
s=k*sin(2*pi/0.628*time+phi);
delta=s-pp;
sqd=0;
for i=1:n
    sqd=sqd+delta(i)^2;
end
%
plot(time,pp,'r', time,s,'g');
grid
ylabel('Roll Displacement (deg)');
xlabel('Time t (sec)');
title('Roll Displacement with LSESF by Newton Raphson vs Time');

```

## C.19. Determine the Phase Error Between Two Signals by Curve Fit

```
% fit y and yc at the range of [a b] with a curve,
% and calculate the phase difference between signals
% y and yc

% Program name: phaseError.m Nov., 2004

% y--input: sequence 1 to be fit
% yc--input: sequence 2 to be fit
% ta--input: starting time (s)
% tb--input: ending time (s)
% T--input: sampling period
% d--output:  $y(x)=\text{mean}(y)$ ,  $yc(x)=\text{mean}(yc)$ ,  $d=xc-x$ 
% e--output: fitting covariance for sequence 1
% ec--output: fitting covariance for sequence 2

function [d,e,ec]=phaseError(y, yc, ta, tb, T, td)
leny=length(y);
t=0:T:(leny-1)*T;
a=floor(ta/T);
b=floor(tb/T);
tp=t(a:b)'-t(a);
lp=length(tp);
%
% frequencies of the sinusoidal functions to fit the curve
%
wi=[0.18850, 0.31416, 0.50265, 0.87965, 1.44513, 2.13628, 3.07876...
    4.20973, 5.78053, 8.23097, 11.24690, 15.77079, 23.93894];
A=[ones(lp,1) tp tp.^2];
for i=1:length(wi)
    A(:,i+3)=sin(wi(i)*tp+i*pi/7);
end
B=y(a:b)';
C=inv(A'*A)*A'*B;
%
% calculate the covariance
%
yf=C(1)+C(2)*tp+C(3)*tp.^2;
for i=1:length(wi)
    yf=yf+C(i+3)*sin(wi(i)*tp+i*pi/7);
end
dy=B-yf;
e=sum(dy.^2);
%
f=[num2str(C(1)) '+' num2str(C(2)) '*x+' num2str(C(3)) '*x^2'];
for i=1:length(wi)
    f=[f '+' num2str(C(i+3)) '*sin(' num2str(wi(i)) '*x+'
num2str(i*pi/7) ')']];
end
f=[f '-' num2str(ym)];
ym=mean(B);
dtp=max(tp)/10;
ti=0:dtp:max(tp);
z=-1;
i=1;
```

```

while z<td
    z=fzero(inline(f),ti(i));
    i=i+1;
end
% for ym
Bm=yc(a:b)';
C=inv(A'*A)*A'*Bm;
%
% calculate the covariance
%
yfm=C(1)+C(2)*tp+C(3)*tp.^2;
for i=1:length(wi)
    yfm=yfm+C(i+3)*sin(wi(i)*tp+i*pi/7);
end
dy=Bm-yfm;
ec=sum(dy.^2);
%
f=[num2str(C(1)) '+' num2str(C(2)) '*x+' num2str(C(3)) '*x^2'];
for i=1:length(wi)
    f=[f '+' num2str(C(i+3)) '*sin(' num2str(wi(i)) '*x+'
num2str(i*pi/7) ')']];
end
f=[f '-' num2str(ym)];
zc=-1;
i=1;
while zc<0
    zc=fzero(inline(f),ti(i));
    i=i+1;
end
d=z-zc;
d=floor(d*1000+0.5)/1000;Time');

```

## C.20. Calculate the Platform Accelerations from Accelerometers Readings (C++)

```
/* C++ program to calculate the accelerations of the motion
   platform in 6-DOF from six accelerometers mounted on the
   platform as shown in Fig. 2.3
*/

// Program name: main.cpp,ReadDataFile.h & ReadDataFile.cpp, Nov., 2000

/* main.cpp */
//
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include "ReadDataFile.h"

int main()
{
    ReadDataFile Solution;

    cout << "Start Reading Data\n\n";

    Solution.get_data();

    return 1;
}

/* ReadDataFile.h */
//
#ifndef __POPULATION_H
#define __POPULATION_H

#include <iostream.h>
#include <stdlib.h>
#include <math.h>

class ReadDataFile
{
    double xx[3], yy[3], zz[3];
    double crame(double *xx, double *yy, double *zz);
public:
    void get_data();
    ReadDataFile();
    double A[6],A1, A2, A3, A4, A5, A6, AA[2400][12];
    double x[6], y[6], z[6], den, num, a1, a2, a3, t1, t2, t3;
};

#endif

/* ReadDataFile.h */
```

```

//
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include "ReadDataFile.h"

ReadDataFile :: ReadDataFile()
{
}

// function to calculate the determinant for using Crame rule.

double ReadDataFile :: crame(double *c1, double *yy, double *zz)
{
    double sum;

    sum=c1[0]*yy[1]*zz[2]+yy[0]*zz[1]*c1[2]+zz[0]*c1[1]*yy[2]
        -c1[2]*yy[1]*zz[0]-yy[2]*zz[1]*c1[0]-zz[2]*c1[1]*yy[0];

    return sum;
}

void ReadDataFile :: get_data()
{
    int i,j;
    float col[12];

    // Set the coordinates of the 6 accelerometers

    x[0]=-14.5; y[0]=91.375;
    x[1]=-5.5; y[1]=0;
    x[2]=-95; y[2]=-93.125;
    x[3]=111.875; y[3]=0;
    x[4]=0; y[4]=91.375;
    x[5]=-14.5; y[5]=-91.375;

    z[0]=-13.375;
    z[1]=-9.75;
    z[2]=-12.8125;
    z[3]=-12.25;
    z[4]=-13.5;
    z[5]=-13.5;

    for (i=0; i<3; i++)
    {
        xx[i]=1;
        yy[i]=y[i+2];
        zz[i]=-x[i+2];
    }

    den=crame(xx, yy, zz);
    ifstream fin("12_17Xt.txt");
    FILE *stream;
    stream = fopen( "xoutpt.txt", "w" );
}

```



```

// The total number of steps is 2400
for (j = 0; j <= 2400; j++)
{
    // The original data file has 12 columns

    for (i=0; i<12; i++)
    {
        fin >> col[i];
    }

    // the last 6 columns are sensed accelerations

    for (i=0; i<6; i++)
    {
        A[i]=col[i+6]*192.9175;
    }

    // Calculate vertical acceleration t3

    for (i=0; i<3; i++)
    {
        xx[i]=A[i+2];
        yy[i]=y[i+2];
        zz[i]=-x[i+2];
    }
    num=crame(xx, yy, zz);
    t3=num/den;
    t3 = t3*57.296; // Converting to degree/s^2
    cout << "Translational acceleration t3: " << t3 << "\n";

    // Calculate roll acceleration a1

    for (i=0; i<3; i++)
    {
        yy[i]=xx[i];
        xx[i]=1;
    }
    num=crame(xx, yy, zz);
    a1=num/den;
    a1=a1/385.835; // Converting to value in terms of g

    // Calculate pitch acceleration a2

    for (i=0; i<3; i++)
    {
        zz[i]=yy[i];
        yy[i]=y[i+2];
    }
    num=crame(xx, yy, zz);
    a2=num/den;
    a2=a2/385.835; // Converting to value in terms of g

    // Calculate yaw acceleration a3

    a3 = (A[0]-A[5]-(z[0]-z[5])*a2)/(y[5]-y[0]);
    a3=a3/385.835; // Converting to value in terms of g

```

```

// Calculate longitudinal acceleration t1

t1 = A[0]+(A[0]-A[5]-(z[0]-z[5])*a2)/(y[5]-y[0])*y[0]-
      z[0]*a2;
t1 = t1*57.296; // Converting to degree/s^2

// Calculate the lateral acceleration t2
t2 = A[1]-x[1]*a3+z[1]*a1;
t2 = t2*57.296; // Converting to degree/s^2

// Save the 6 accelerations into a file

fprintf( stream, "%f %f %f %f %f %f\n",t1,t2,t3,a1,a2,a3);
}
fclose( stream );
}

```

## References

- <sup>1</sup> Guo L., Cardullo, F.M., and Kelly, L.C., “Advanced Control Algorithms for Compensating the Phase Distortion Due to Transport Delay in Human-Machine Systems”, NASA CR 2007-215095, 2007
- <sup>2</sup> Horowitz, S. C., “Measurements and Effects of Transport Delays in a State-of-the-Art F-16C Flight simulator”, (AFHRL-TP-87-11), Brooks Air Force Base, TX: Air Force Human Resources Laboratory
- <sup>3</sup> Niemeyer, G. E., and Dougherty, B., “Measurement of Flight Simulator Time Delays”,
- <sup>4</sup> Horowitz, S. C., “Measurements and Effects of Transport Delays in a State-of-the-Art F-16C Flight simulator”, (AFHRL-TP-87-11), Brooks Air Force Base, TX: Air Force Human Resources Laboratory
- <sup>5</sup> Smith, R.M., “A Method for Determining Transport Delays in the Flight Simulation Environment”, AIAA Flight Simulation Technologies Conference, 1991, New Orleans
- <sup>6</sup> Galloway, R.T., and Smith R.B., “Cue Synchronization Measurement Using the Piloted Frequency Sweep Technique”, Proceedings of the 17<sup>th</sup> I/ITSEC, Nov 13-16, pp786-196
- <sup>7</sup> Wolters, T.E., “Research Flight Deck Transport Delay Study”, Simulation Systems Branch, Simulation and Research Aircraft division
- <sup>8</sup> Parrish, R.V., Dieudonne, J.E., Martin, D.J., Jr., and Copeland, J.L., “Compensation Based on Linearized Analysis for a Six-Degree-of-Freedom Motion Simulator,” NASA TND-7349, 1973.
- <sup>9</sup> Parrish, R.V., Dieudonne, J.E., and Martin, D.J., Jr., “Motion Software for a Synergistic Six-Degree-of-Freedom Motion Base,” NASA TND-7350, 1973.
- <sup>10</sup> Telban, R.J., "New Human-Centered Linear and Nonlinear Motion Cueing Algorithms for Simulator Motion System", State University of New York at Binghamton. Binghamton, NY, 2003.
- <sup>11</sup> Hart, S.G., and Staveland, L.E. (1988). “Development of a Multi-dimensional Workload Rating Scale: Results of Empirical and Theoretical Research”, In P. A. Hancock, and Meshkati, N. (Ed.), *Human Mental Workload*. Amsterdam, The Netherlands: Elsevier.
- <sup>12</sup> Middendorf, M.M., Fiorita, A.L., and McMillan G.R., “The Effects of Simulator Transport Delay on Performance, Workload, and Control Activity”, AIAA-91-2965, 1991
- <sup>13</sup> Porat B., “A Course in Digital Signal Processing”, John Wiley & Sons, Inc, 1996
- <sup>14</sup> Hu, G., “Digital Signal Processing”, Press of Tsinghua University, 2000
- <sup>15</sup> Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., “Numerical Recipes in C++”, Cambridge University Press, 2002

**REPORT DOCUMENTATION PAGE**

*Form Approved  
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.  
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 01-12-2007		<b>2. REPORT TYPE</b> Contractor Report		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b> Advanced Transport Delay Compensation Algorithms: Results of Delay Measurement and Piloted Performance Tests				<b>5a. CONTRACT NUMBER</b> NNL06AA74T	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Guo, Liwen; Cardullo, Frank M.; and Kelly, Lon C.				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b> 160961.01.01.01	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> NASA Langley Research Center Hampton, VA 23681-2199				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration Washington, DC 20546-0001				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> NASA	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> NASA/CR-2007-215096	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Unclassified - Unlimited Subject Category 54 Availability: NASA CASI (301) 621-0390					
<b>13. SUPPLEMENTARY NOTES</b> Langley Technical Monitor: Jacob A. Houck An electronic version can be found at <a href="http://ntrs.nasa.gov">http://ntrs.nasa.gov</a>					
<b>14. ABSTRACT</b> This report summarizes the results of delay measurement and piloted performance tests that were conducted to assess the effectiveness of the adaptive compensator and the state space compensator for alleviating the phase distortion of transport delay in the visual system in the VMS at the NASA Langley Research Center. Piloted simulation tests were conducted to assess the effectiveness of two novel compensators in comparison to the McFarland predictor and the baseline system with no compensation. Thirteen pilots with heterogeneous flight experience executed straight-in and offset approaches, at various delay configurations, on a flight simulator where different predictors were applied to compensate for transport delay. The glideslope and touchdown errors, power spectral density of the pilot control inputs, NASA Task Load Index, and Cooper-Harper rating of the handling qualities were employed for the analyses. The overall analyses show that the adaptive predictor results in slightly poorer compensation for short added delay (up to 48 ms) and better compensation for long added delay (up to 192 ms) than the McFarland compensator. The analyses also show that the state space predictor is fairly superior for short delay and significantly superior for long delay than the McFarland compensator.					
<b>15. SUBJECT TERMS</b> Delay Compensation Algorithm; Flight Simulation; Simulator; Transport Delay; Visual System					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			STI Help Desk (email: <a href="mailto:help@sti.nasa.gov">help@sti.nasa.gov</a> )
U	U	U	UU	196	<b>19b. TELEPHONE NUMBER (Include area code)</b> (301) 621-0390