WL-TR 95-3070

INTERACTIVE FLYING QUALITIES TOOLBOX FOR MATLAB® USER'S GUIDE

VOL 1, SHORT TERM PITCH RESPONSE
CRITERIA AND MODIFIED OPTIMAL CONTROL
PILOT MODEL

DAVID B. DOMAN

AUGUST 1995

FINAL REPORT FOR   08/01/94 - 08/01/95

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

19960306 083

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7562

DISCLAIMER NOTICE
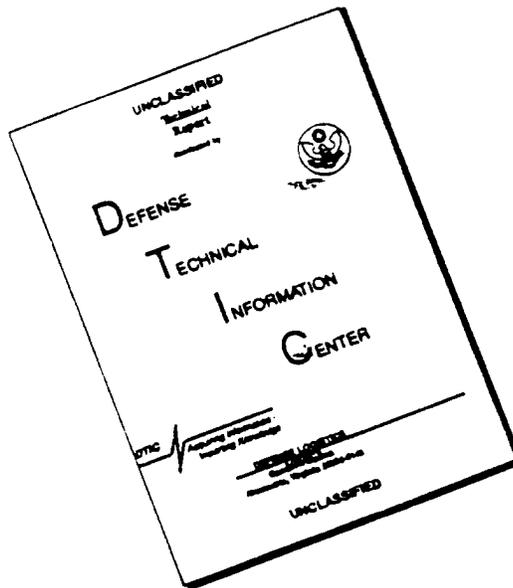
THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.


This technical report has been reviewed and is approved for publication.

DAVID B. DOMAN
Flying Qualities Engineer
Control Dynamics Branch
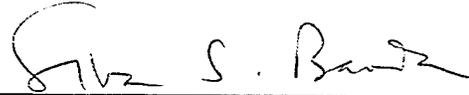
DR SIVA S. BANDA
Chief, Control Dynamics Branch
Flight Control Division

DAVID P. LEMASTER
Chief, Flight Control Division
Flight Dynamics Directorate

REPORT DOCUMENTATION PAGE

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 Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE AUG 1995	3. REPORT TYPE AND DATES COVERED FINAL 08/01/94--08/01/95

4. TITLE AND SUBTITLE INTERACTIVE FLYING QUALITIES TOOLBOX FOR MATLAB USER'S GUIDE VOL 1, SHORT TERM PITCH RESPONSE CRITERIA AND MODIFIED OPTIMAL CONTROL PILOT MODEL

6. AUTHOR(S) DAVID B. DOMAN

5. FUNDING NUMBERS

PE 62201
PR 2403
TA 05
WU 95

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

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7562

8. PERFORMING ORGANIZATION REPORT NUMBER

WL-TR-95-3070

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

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7562

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

WL-TR-95-3070

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Aircraft flying qualities are of fundamental importance to aircraft designers and flight control engineers. An integrated efficient software tool has been developed for the purpose of predicting the short term pitch response flying qualities of an aircraft. All short term pitch response criteria of MIL-STD-1797-A are included as well as some additional criteria found in the literature. A general pilot modeling tool is also provided with this package. This report covers how to install and use the software and contains reference material for developers who wish to expand its' capabilities. The software was developed as a MATLAB® Toolbox and is designed to be very easy to use. It can greatly reduce the amount of time and effort involved in a handling qualities analysis.

14. SUBJECT TERMS

HANDLING QUALITIES   FLYING QUALITIES
PILOT MODELS   MANUAL CONTROL
PILOT DYNAMICS   VEHICLE DYNAMICS

15. NUMBER OF PAGES 109

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

FOREWORD

The software described herein was developed to support an in house research effort aimed at resolving disagreements between results produced by different short term pitch response flying qualities criteria. David B. Doman is the principal author of the toolbox and this manual. Captain Brian A. Kish contributed to the development of the graphical user interface and provided valuable editorial support for this document. David Leggett was an invaluable source of information regarding flying qualities criteria and MILSTD-1797A. He also tested the software on numerous aircraft configurations and pointed out problems with the algorithms and user interface. Tom Gentry provided valuable technical advice concerning the development of the algorithms used in the Interactive Flying Qualities Toolbox. Dr. Mark Anderson of Virginia Polytechnic Institute provided a wealth of information and insight which made the development of the Modified Optimal Control Pilot Model Toolbox possible.

The flying qualities group has found that the toolbox has greatly reduced the amount of time and effort required to perform a flying qualities analysis. This manual and accompanying software is provided in that hope that a broader range of users will enjoy the benefits of our effort.

Table of Contents.

List of Figures

List of Tables

1. IFQT Introduction

Aircraft flying qualities are of fundamental importance to aircraft designers and flight control engineers. Aircraft must meet certain flying qualities requirements specified by government agencies. The short-term pitch response of an aircraft is a vital element of flying qualities. MIL-STD-1797A, *Flying Qualities of Piloted Aircraft*, offers six different methods for evaluating short term pitch response. All six methods have been maintained because the short-term pitch response characteristics are universally regarded to be extremely important - so important that controversy over both the form and the substance of requirements still exists. This has made the use of the document rather cumbersome because different criteria frequently give different estimates of a vehicles handling qualities. Use of the criteria themselves has been difficult and frequently requires consultation with an expert. Some of the older criteria are based on graphical or trial and error approaches. Members of the Flying Qualities Group developed algorithms which check the compliance of an aircraft with all of the short term pitch response criteria of MIL-STD-1797A as well as some alternative criteria found in the literature. These algorithms were written in FORTRAN and older, less efficient control analysis packages, neither of which were user friendly. This made the implementation platform specific.

An integrated, efficient, user friendly environment for checking aircraft compliance with specification requirements was desired to support an in-house research effort. The effort was aimed at resolving disagreements between flying qualities estimates produced by the different short term pitch response criteria. It was also desired to produce a portable software package for checking specification compliance which could be used by other government agencies, industry and academia. It was decided to code the algorithms in Matlab® because of its rich library of efficient numerical routines and the ability to create a user friendly environment using its graphical user interface.

The following criteria have been implemented:
- Transient Peak Ratio
- Gibson's Dropback, Phase Rate and Nichols Chart Criteria
- Neal-Smith Pilot Model analysis.
- Bandwidth Criteria
- CAP (Control Anticipation Parameter) Criteria with Equivalent Systems Matching
- $\omega_{sp} - T_{\theta_2}$ Criteria with Equivalent Systems Matching
- Modified Optimal Control Pilot Model analysis of pitch axis.
- Smith-Geddes Criteria
- Northrop's pitch rate frequency response envelope criteria.

The Matlab® implementation has greatly reduced the amount of time and effort involved in the in-house research effort. The bottom line is that anyone who is familiar with a Windows point-and-click environment can operate this toolbox without knowledge of Matlab® or an intricate knowledge of flying qualities.

1.1 Features

- Ability to analyze a pitch attitude transfer function using multiple criteria simultaneously.
- Changes in the input cause all open criteria to be updated.
- All criteria analyses can be printed individually or collectively.
- Transfer functions can be entered in short hand notation or descending powers of s.
- The software handles transfer functions with pure time delay.
- User friendly environment created using Matlab's® graphical user interface .
- Ability to put crosshairs on plots.
    - Crosshairs work with multi-trace plots via popup menu.
    - Crosshairs can be moved using the mouse, a slider or by typing in parameters.
    - Position of the crosshairs is reflected by numerical parameters displayed in a sidebar on the plot.
- Axis limits on plots can be adjusted using editable text in the plot sidebar.
- Grids on plots can be toggled from the plot sidebar.
- Greek letters and symbols are printed on hardcopy output using Dan Braithwaite's TextFcn Toolbox.
- Configuration control is provided to keep your analysis organized.

1.2 System Requirements

Matlab 4.2 or Greater.
Graphics terminal.
Mouse.
Control Systems Toolbox 3.0b or greater.
Optimization Toolbox 1.0d or greater.

Recommended:
12 MB Ram. Use virtual RAM or swap space if you must.

Note:
This toolbox was developed on a MS-Window platform. It has been successfully run on Unix based X-Windows platforms as well.

1.3 Installation

Assumptions:
- You are installing from a floppy disk in your A drive to hard disk C: on an MS-DOS based computer.
- Matlab tooboxes are installed in :   C:\MATLAB\TOOLBOX. If your configuration differs, you will have to modify the **install.bat** file provided on the floppy disk.

Insert the floppy disk into drive A:

Type install

After the installation, set the matlab path in your **matlabrc.m** file to include the following:

      ';C:\matlab\toolbox\milstd',...
      ';C:\matlab\toobox\milstd\textfcns',...
      ';C:\matlab\toolbox\milstd\mocm',...

1.4 Fundamental Operation

A detailed tutorial is provided in Chapter 2 of this manual. This section is provided for those who wish to get started immediately.

1.   Start Matlab.
2.   Type "milstd" at the prompt.
3.   Select Longitudinal analysis by clicking on the longitudinal pushbutton.
4.   Enter your pitch axis transfer function in the appropriate fields.

      Example:

$$\frac{\theta(s)}{\delta(s)} = \frac{s+1}{s(s^2 + 2s + 3)}$$

a. Option 1. Descending Powers of s is the default selection displayed in the popup menu.

      Numerator
      [1 1]

      Denominator
      [1 2 3 0]

b. Option 2. Short Hand Notation can be selected using the popup menu in the upper right hand corner of the main control panel.

      Numerator
      [1 1 1]

      Denominator
      [2 1 0 2 0.5773 1.732]

      Explanation:
      The numerator has one polynomial factor
        *
      [1 1 1]

      It is a first order factor

```
        *
[1 1 1]
```

The factor has a root at -1 (i.e. s+1)
```
      *
[1 1 1]
```

The denominator has 2 polynomial factors
```
*
[2  1 0  2 0.5773 1.732]
```

The first factor is first order and
has a root at zero (i.e. s)
```
      * *
[2  1 0  2 0.5773 1.732]
```

The second factor is second order and
has a damping of 0.577 and a frequency
of 1.732
```
          *     *     *
[2  1 0  2 0.5773 1.732]
```

Note: pure gains can be entered as zero order
factors e.g. K(s+a)

```
[2 0 K  1 a]
```

1.5 Criteria Summary

The following short term pitch response criteria and pilot models are available in this package:

Transient Peak Ratio Criteria: A time domain criteria developed for pitch rate response to a step input. In addition to the pitch attitude transfer function it needs true airspeed in ft/sec. A complete description of this criteria can be found in Reference [1], p. 217.

Gibson's Dropback Criteria: A time domain criteria developed for pitch attitude response to a boxcar input. See Reference [1], p. 244.

Bandwidth Criteria: Frequency domain critera for pitch attitude frequency response. The original criteria can be found in Reference [1], p. 225. The proposed revision can be found in Reference [13].

Neal-Smith Criteria: This criteria calculates a simple model of a human pilot subject to the rules specified in Reference [14]. The criteria in Reference [1] (MILSTD 1797A) is not used. The frequency response of the pilot/vehicle series combination is displayed on a Nichols Chart. The user must enter a target bandwidth and a pilot time delay and press the done button. A Nichols Plot of the Pilot / Vehicle series combination will be displayed. The Nichols plot should cross the -90 deg closed loop phase contour at the target frequency. Theminimum

closed loop magnitude between the specified target frequency and .1 rad/sec should be -3 dB; if not,
a warning is displayed. The most phase compensation the pilot model can generate is +/- 90 deg.
If the pilot model is incapable of meeting the requirements a warning is displayed. Try again
with a lower or higher target frequency. The flying qualities level can be determined by pushing
the "Neal-Smith Criteria" button. You can also select the pilot model representation.

Gibson's Phase Rate Criteria: A frequency domain criteria which relates PIO susceptibility
to the slope of phase curve at the -180 deg crossover frequency and the -180 crossover frequency.
A complete description of this criteria can be found in Reference [1], p. 244.

CAP Criteria: A criteria developed for the analysis of classical short period response types. It
relates the Control Anticipation parameter (CAP $= \omega_{sp}^2 g T_{\theta_2} / V_{True}$) and short period damping
ratio to flying qualities levels. If the aircraft transfer function is not given in a short period
approximation format, an equivalent systems analysis must be performed. The software
automatically takes you to an equivalent systems control panel which is described below. In
addition to the pitch attitude transfer function, true airspeed in ft/sec is also needed. A complete
description of this criteria can be found in Reference [1], p. 172.

$\omega_{sp}^2 - T_{\theta_2}$ **Criteria**: A criteria developed for the analysis of classical short period response
types. It relates ω_{sp}, T_{θ_2} and short period damping ratio to a flying qualities level. If the aircraft
transfer function is not given in a short period approximation format, an equivalent systems
analysis must be performed. The software automatically takes you to an equivalent systems
control panel which is described below. A complete description of this criteria can be found in
Reference [1], p. 190.

Equivalent Systems Analysis: Reduces a higher order pitch attitude transfer function to a lower
order equivalent system transfer function in short period form. This is only accessible through
CAP and $\omega_{sp}^2 - T_{\theta_2}$ Criteria and only when the input transfer function is not in short period
approximation format. You can supply initial guesses for the parameters or use the default
guesses. It is advisable to look at the HOS bode plot to get a rough estimate of the dominant
poles and zeros for use as initial guesses. Varying the frequency range over which the match is
performed may also improve results. A description of the procedure can be found Reference [1],
p. 681.

Northrop Criteria: A frequency domain criteria for the pitch rate response. It was developed
for up and away flying tasks for fighters. Magnitude and phase envelopes are shown, if the
frequency response is outside of those envelopes, flying qualities may not be optimal. See
Reference [8].

Gibson's Nichols Chart Criteria: A frequency domain criteria for the pitch attitude response.
It adjusts the gain such that the 0 dB crossover frequency is equal to 0.3 Hz. Flying qualities
characteristics are inferred from the comments on the Nichols chart. A complete description of
this criteria can be found in Reference [1], p. 244.

Ralph Smith Criteria: A multi-transfer function - multi domain criteria for the longitudinal response of an aircraft. Currently only pitch attitude response portions of the criteria are implemented. Background information on this criteria is available in Reference [17].

MOCM Pilot Model: The Modified Optimal Control Model is a mathematical model of a human pilot engaged in a tracking task. MOCM is supplied in a separate directory and is a complete toolbox in itself. It is capable of modeling human operator behavior in multi-axis tracking tasks. A complete manual for this toolbox is included in Part II of this report. A GUI front end has been integrated into the IFQT for pitch axis analysis. The MOCM theoretical development is available in Reference [5].

1.6   Miscellaneous

Tips: Always use the Close buttons provided. If you don't
                   you will get alot of "dead" windows lying about.

Notes: The MOCM does not handle plants with time delay. This
                   is because there is currently no known way of separating
                   plant and pilot   time delay in the MOCM framework without
                   violating some fundamental assumptions.

2. IFQT Tutorial

A Sample Session Using the Interactive Flying Qualities Toolbox.

At the Matlab prompt type:

>milstd

The graphics window shown in Figure 2-1 will appear:
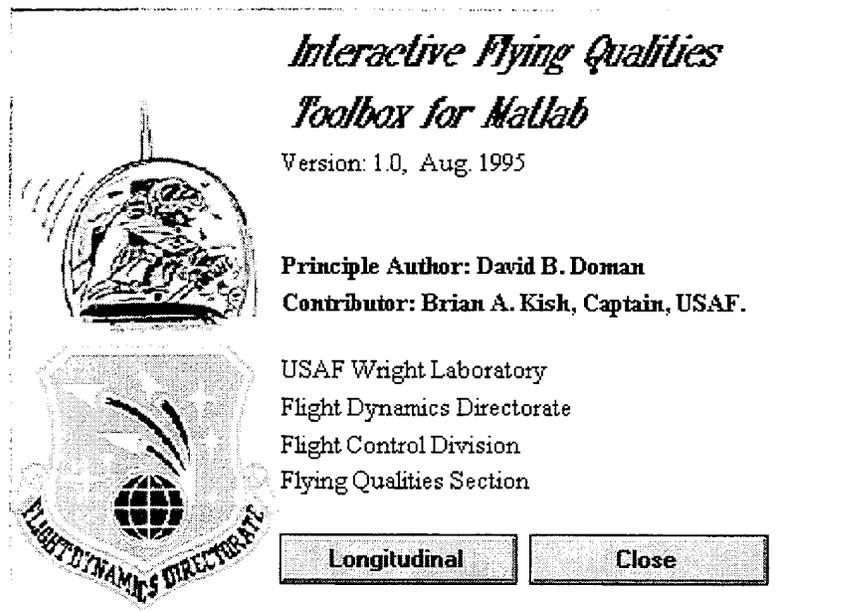
Figure 2 - 1. Introductory window.

Click on the **Longitudinal** push button.

The main menu for the longitudinal criteria will appear (Figure 2-2). At this point you must enter the transfer function using the editable text boxes. The default format for the transfer function is descending powers of s although short hand notation is available if desired. The default configuration name is Pitch-TF 0 , but you may change the name. This name will appear on each of the plot windows and printouts to keep your analyses organized. Clicking on the **Done** button will increment the configuration counter by 1 and all open criteria will be recalculated.

For illustrative purposes we have chosen to enter the following short period approximation for the pitch attitude transfer function in descending powers of s.

$$\frac{\theta(s)}{\delta(s)} = \frac{s+1}{s(s^2+2s+3)} e^{-0.1s}$$

Theta/delta Transfer Function

Pitch-TF   0   Descending Powers of s

Numerator   Delay (sec)
[1 1]   0.1
Denominator
[1 2 3 0]

Done

Criteria

○ Transient Peak Ratio   ○ CAP Criteria

○ Gibson Dropback   ○ Wsp-T_Theta_2 Criteria

○ Bandwidth Criteria   ○ Northrop Criteria

○ Neal-Smith Criteria   ○ Gibson's Nichols Chart

○ Gibson's Phase Rate   ○ MOCM Pilot Model

                                ○ Ralph Smith

Print   Close All

Figure 2 - 2. Longitudianal analysis control panel.

Once the transfer function has been entered, click on any of the criteria.

2.1 Transient Peak Ratio Criteria

The TPR criteria uses parameters extracted from a pitch rate step response to estimate the flying qualities of the pitch axis. The data window shown in Figure 2-3 appears when the **Transient Peak Ratio** radio button is selected.

t1 (sec)	0.10048	
t2 (sec)	0.433898	
t2-t1 (sec)	0.333418	
Max q time (sec)	1.21176	
q_ss (units/sec)	0.333333	
q_max / q_ss	1.46573	
delta q_1 (units/sec)	0.155244	
delta q_2 (units/sec)	0.0168366	
Transient Peak Ratio	0.108453	

Flight Phase
Nonterminal

True Airspeed (ft/sec)
100

Ok

Print   Close TPR

Figure 2 - 3. Transient peak ratio data window.

The text boxes contain parameters which are important to the analysis. Parameters which impact flying qualities have additional text boxes which indicate the predicted flying qualities levels. We will now describe the meaning of each parameter in the text boxes. The equivalent time delay **t1** is measured from the instant the step is applied to the time at the intersection of the maximum slope line with the time axis. The time measured from the instant of the step input to the time corresponding to the intersection of the maximum slope line with the steady-state pitch rate line is **t2**. The peak value of pitch rate due to a step input is **q_max** and the time at which it occurs is **Max q time.** The steady-state value pitch rate due to a step input is **qss**. The difference between the maximum value of q and the steady-state value is **delta q_1,** while **delta q_2** is the difference between the steady-state value of q and the first minimum. The **Transient Peak Ratio** is given by **delta q_1/delta q_2**.

The default **True Airspeed** is 0, but 100 ft/sec is used in this case. The user must click on the **Ok** button to update the flying qualities levels. It should be noted that the CAP criteria also uses true airspeed. If the CAP criteria is open when you change the true airspeed in the Transient Peak Ratio window, both the CAP criteria and the Transient Peak Ratio Criteria will be updated to reflect the change. The criteria requirements are also task dependent. The task can be selected by using the popup menu to select the appropriate flight phase.

In addition to the Transient Peak Ratio data window, the plot window shown in Figure 2 - 4 is also generated. Note that several lines are shown in addition to the step response. These lines are associated with the parameters displayed in the data window and are labeled here for clarity. These labels are not shown on the actual plot.
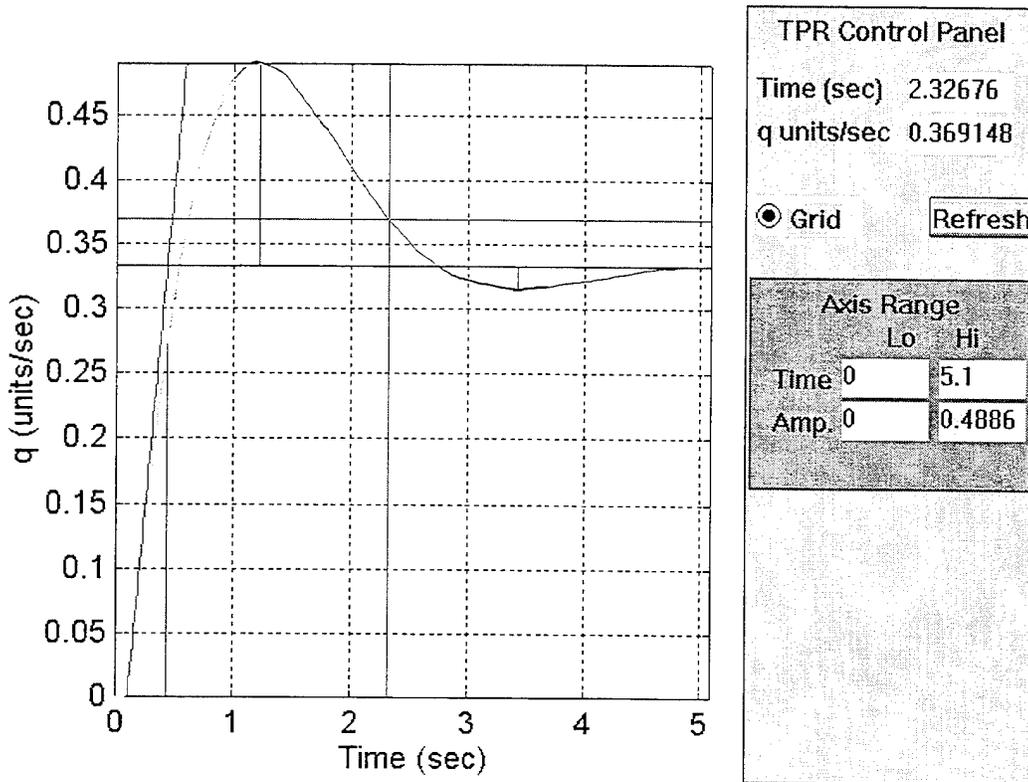
Figure 2 - 4. Plot window for the Transient Peak Ratio Criteria.

Crosshairs are useful for determining the value of the pitch rate response at a given time. They appear as soon as the user clicks on the plot area. The user can click and drag the crosshairs using the mouse. The crosshairs will automatically follow the time history of the pitch rate step response. The time and pitch rate values at the crosshair location are displayed in the sidebar and are automatically updated as the crosshairs move.

The axis scales may be changed by entering the desired values in appropriate editable text boxes. The grid lines may be toggled on and off by clicking the **Grid** radio button. The

Refresh button is provided for systems where covering the graphics window with another window disables the crosshairs. If the crosshairs do not work or the plot appears to be "messed up" just click on the **Refresh** button.

Clicking on the **Print** button will produce a hard copy of the TPR data and plot window. The data will be printed with greek letters and symbols where appropriate.

The TPR analysis may be concluded by clicking on the **Close TPR** button. It is strongly recommended that you use the **Close** buttons provided in each criteria rather than the close option from the standard pull down menu. The **Close** buttons close all windows associated with a particular criteria. Most criteria have two windows, a plot and a data window.

2.2 Gibson's Dropback Criteria

Gibson's dropback criteria estimates pitch axis flying qualities based on parameters obtained from a pitch attitude response to a pulse input. The data window shown in Figure 2-5 will appear after clicking on the **Gibson's Dropback** button on the longitudinal main menu.

t_out (sec)	7.09964
th_out (units)	2.41128
th_end (units)	2.33319
Drop Back (units/sec)	0.0780938
S.S. q (units/sec)	0.333333
Drop Back / q	0.234281
q_max / q	1.46573

◉ Theta Response to a Box-Car Input

○ Precision Tracking q ˜ Theta Trends

Print	Close

Figure 2 - 5. Data window for Gibson's dropback criteria.

The parameters in the data window will now be explained. The time at which the pulse transitions from 1 to 0 is **t_out** and the value of pitch attitude at this time is **th_out.**. The steady state pitch attitude after the pulse is removed is given by **th_end. Drop Back** is defined as **th_out - th_end.** The steady-state pitch rate due to a step input is given by **S.S. q.** The ratio of the dropback parameter and steady-state pitch rate is given by **Drop Back/q**. The ratio of maximum pitch rate to steady-state pitch rate is **q_max/q**.

The default plot window shown in Figure 2 - 6 shows the response of the pitch attitude transfer function to a pulse or boxcar input.
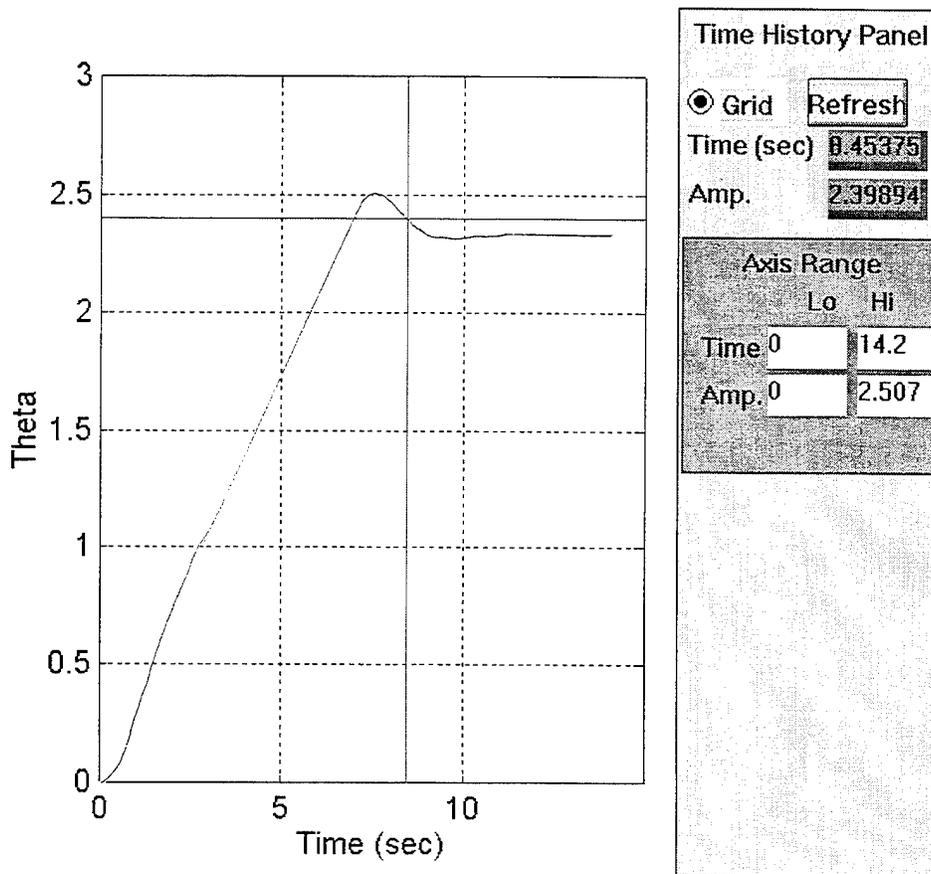
Figure 2 - 6. Default Plot window for Gibson's dropback criteria: pitch attitude response to a pulse.

Note that crosshairs and the sidebar are available for use. This plot window behaves exactly like the plot window associated with the Transient Peak Ratio criteria.

You may select what appears in the plot window by clicking the appropriate radio button in the data window. A qualitative estimate of the pitch axis flying qualities can be obtained by clicking on the **Precision Tracking ~q-Theta Trends** radio button. This generates a plot of the regions of acceptable and deficient flying qualities in terms of the criteria parameters and shows the characteristics of the aircraft under analysis by an asterisk. (See Figure 2-7).
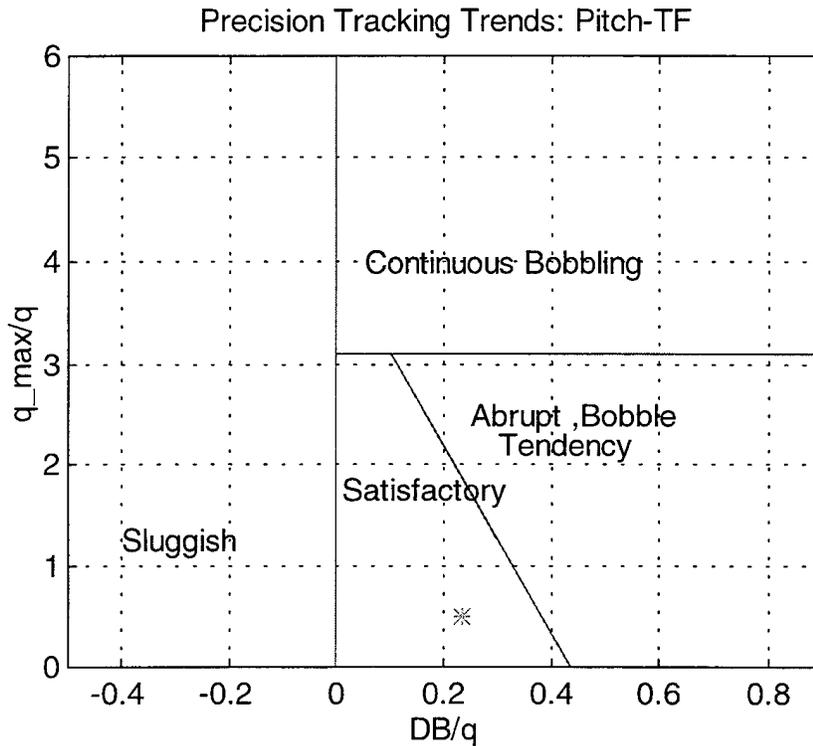
Figure 2 - 7. Flying qualities prediction based on dropback criteria.

The dropback criteria predicts that this aircraft will have satisfactory pitch axis flying qualities.

2.3 Bandwidth Criteria

The bandwidth criteria estimates an aircraft's pitch axis flying qualities from parameters extracted from a bode plot of the pitch attitude transfer function. By selecting the **Bandwidth Criteria** from the longitudinal main menu, the data window shown in Figure 2-8 will appear.

w_45 (r/s)	1.83897
w_6 dB (r/s)	2.57245
w_bw (Bandwidth) (r/s)	1.83897
tau_p (sec)	0.0790825
w_180 r/s	3.57915
phi_dot w180 (deg/(cps))	-87.1415

Analysis

◉ Bode Plot

○ Current Cat. A and B Requirement

○ HAI Cat. A and B Requirement

○ Cat. C Requirement

Print	Close
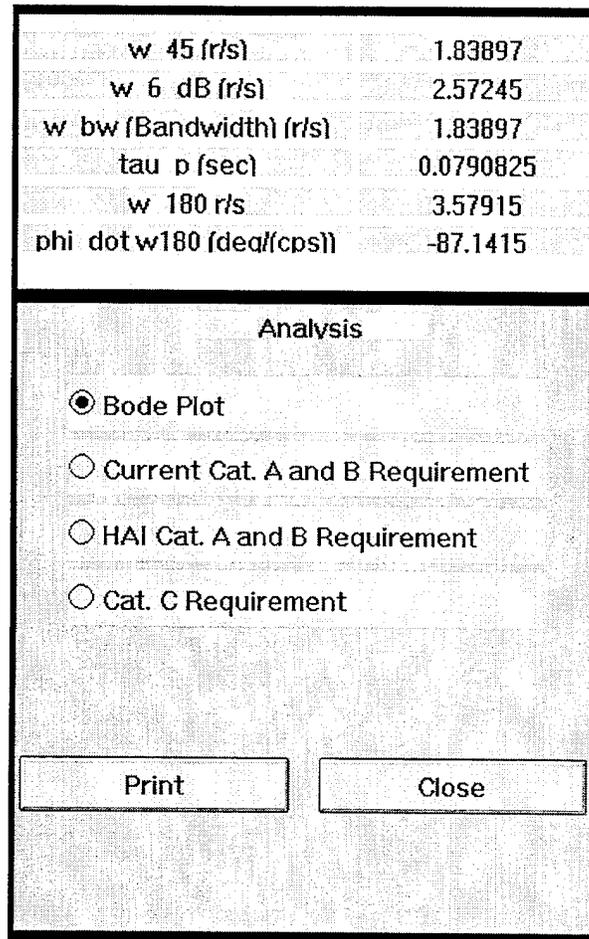
Figure 2 - 8. Bandwidth criteria data window.

Important parameters associated with the criteria are displayed in the text boxes in the data window. The -180 deg phase crossover frequency is **w_180**. The frequency at which the phase margin is 45 deg is **w_45**. The frequency at which the magnitude curve is 6 dB higher than the magnitude at the -180 deg phase crossover frequency is **w_6 dB**. The bandwidth **w_bw** is defined as the least of **w_45** and **w_6dB**. The effective time delay based on analysis of the phase curve at high frequencies is given by **tau_p**. The rate of change of phase with frequency at the -180 deg phase crossover frequency is defined by **phi_dot w_180.**

The default plot shown in Figure 2 - 9, is a bode plot of the pitch attitude transfer function. Additional lines corresponding to criteria parameters are also displayed (e.g. $\omega_{135}, \omega_{180}$ etc.). The familiar sidebar and crosshairs are also available.
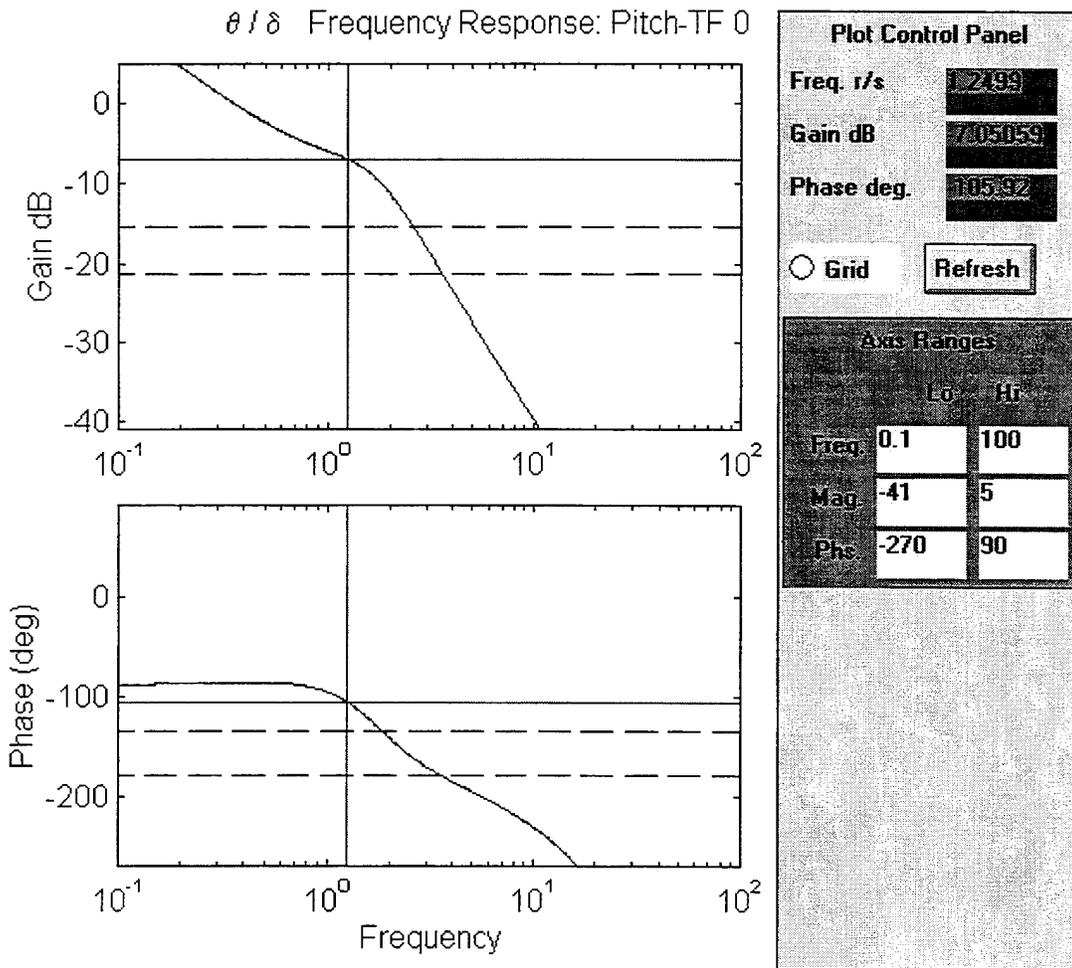
Figure 2 - 9. Default bode plot for the bandwidth criteria.

To obtain an estimate of the flying qualities level for an aircraft in a nonterminal flight phase, click on the Current Cat. A and B Requirement button in the Bandwidth Criteria data window. The current configuration is Level 2 as indicated by the asterisk on the plot shown in Figure 2-10.
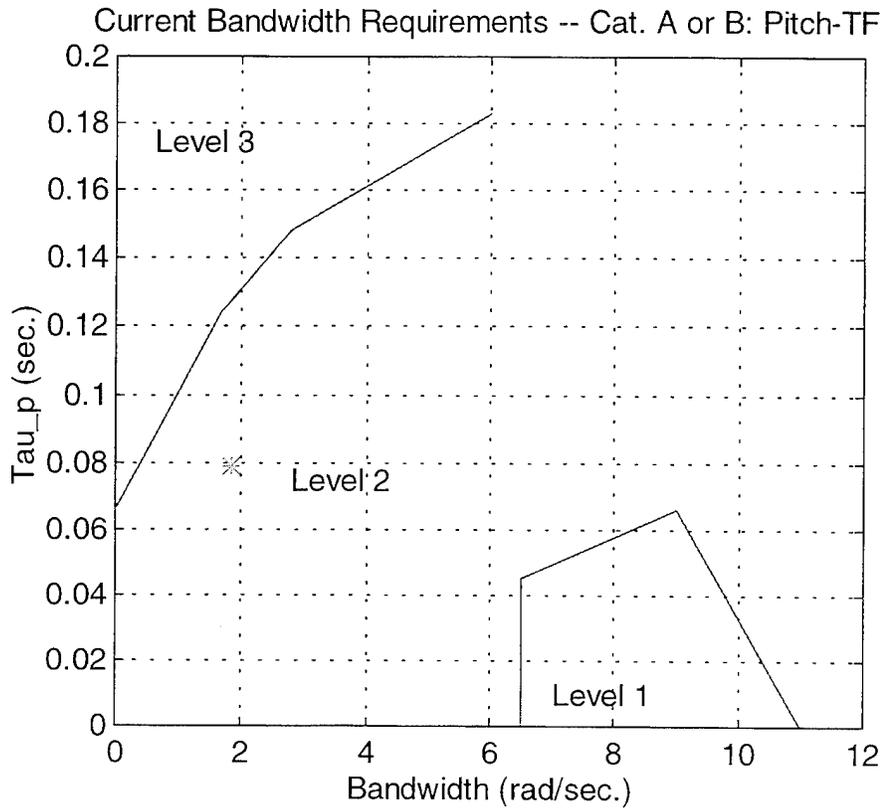
Figure 2 - 10. Bandwidth criteria for nonterminal flight phase.

Flying qualities estimates based on bandwidth criteria parameters for terminal flight phases may be obtained by clicking on the Cat. C Requirement button in the bandwidth data window. Estimates of flying qualities for a nonterminal flight phases based on a revision proposed by Hoh Aeronautics Inc. can be obtained by clicking on the HAI Cat. A and B Requirement button. This proposed revision can be found in WL-TR-94-3162.

2.4 Neal-Smith Criteria

This criteria calculates a simple model of a human pilot subject to the rules specified in AFFDL-TR-70-74. The criteria in the MIL-STD-1797A is not used. The frequency response of the pilot/vehicle series combination is displayed on a Nichols Chart. The user must enter a target bandwidth and a pilot time delay and press the **OK** button shown in Figure 2-11. A Nichols plot of the pilot/vehicle series combination will be displayed. The Nichols plot should cross the -90 deg closed loop phase contour at the target frequency. The minimum closed loop magnitude below the specified target frequency and 0.1 rad/sec should be -3 dB. If it is not, a warning is displayed. The most phase compensation the pilot model can generate is +/- 90 deg. If the pilot model is incapable of meeting the requirements, a warning is displayed. Try again with a lower or higher target bandwidth. The flying qualities level can be determined by pushing the **Neal-Smith Criteria** button in the Neal-Smith Analysis frame.

Figure 2 - 11. Neal-Smith criteria control panel.

The Neal-Smith criteria parameters are **Closed Loop Resonance** and **Pilot Phase Compensation.** The closed loop resonance is the maximum magnitude of the the closed loop pilot/vehicle over all frequencies. The pilot phase compensation is obtained by evaluating the lead or lag-lead portion of the pilot model at the target bandwidth frequency and computing the phase angle.

The default plot for the Neal-Smith criteria is a Nichols chart of the pilot/vehicle transfer function Figure 2 - 12. The sidebar for the Nichols Chart has been modified from previous sidebars. Since frequency is implicit, a slider is available to move the crosshairs along the plot. The frequency can also be entered in the editable text box causing the crosshairs to snap to a specified frequency. The ability to add or remove a superimposed closed loop magnitude and phase grid is controlled by the **Nichols Grid** radio button. In this example the closed loop information required for the Neal-Smith criteria (9 dB, 3 dB, -3 dB, and -90° phase lines) is shown without the full Nichols grid.
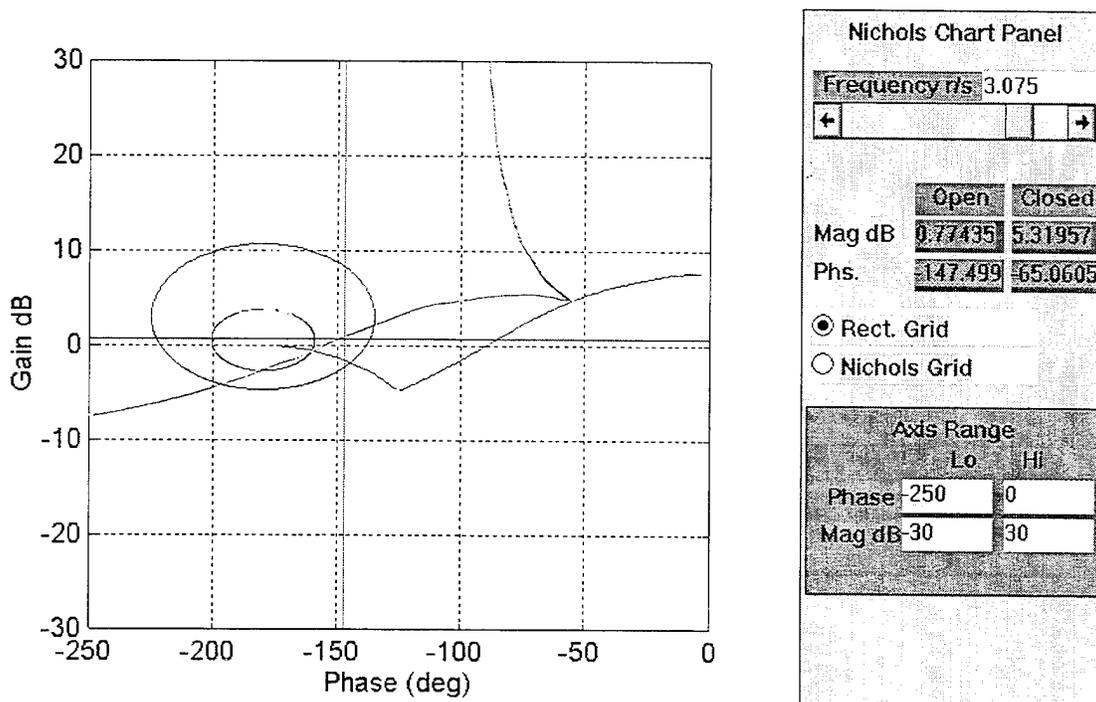
Figure 2 - 12. Default plot for the Neal-Smith Criteria.

The flying qualities prediction is obtained by clicking on the **Neal-Smith Criteria** button in the Neal-Smith Analysis frame. Pilot ratings or flying qualities levels can be predicted by observing the location of the asterisk in Figure 2 - 13. The Neal-Smith Criteria predicts that this configuration will have Level 3 (poor) flying qualities for this particular task (i.e. compensatory tracking task with 3.5 rad/sec input bandwidth).

Pilot Rating Estimate : Pitch-TF
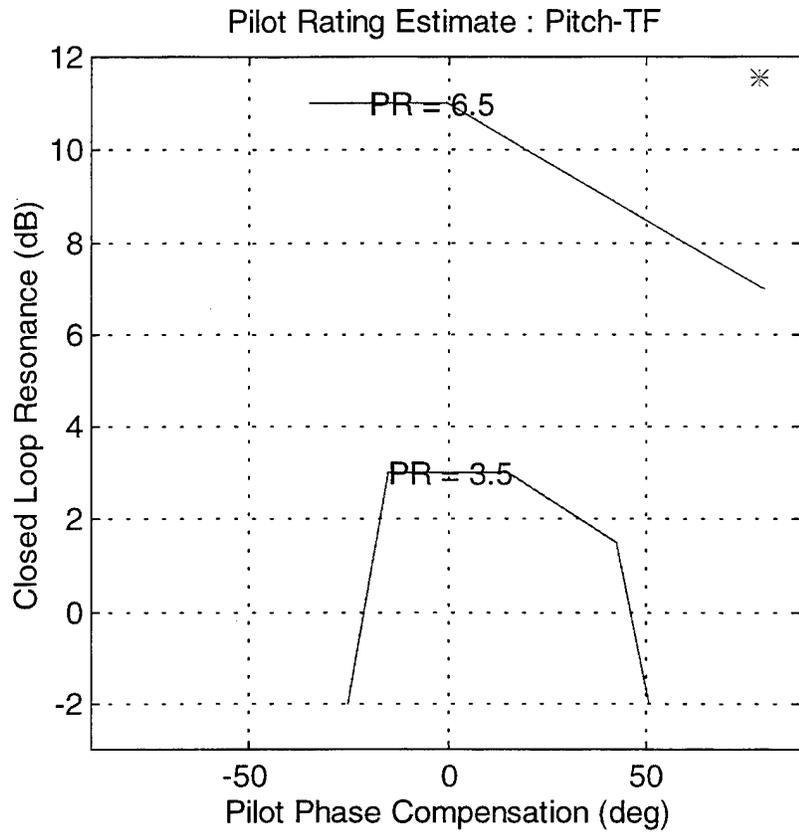
Figure 2 - 13. Estimation of pilot ratings using the Neal-Smith criteria.

2.5 Gibson's Phase Rate Criteria

A frequency domain criteria which relates PIO susceptibility to the slope of the pitch attitude transfer function phase curve at the -180 deg crossover frequency. The criteria parameters and estimates of an aircraft's PIO susceptibility can be obtained by clicking the **Gibson's Phase Rate** button on the main menu. The data and plot windows are shown in Figures 2-14 and 2-15 respectively.

Gibson's Phase

180 deg Lag Freq. (Hz)	0.569639
Phase Rate at -180 deg	-87.1415

Print   Close
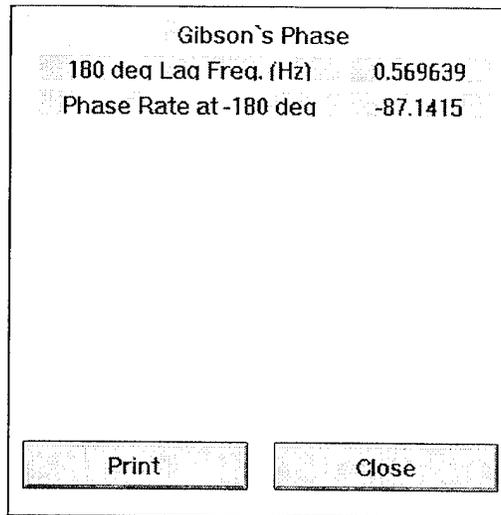
Figure 2 - 14 Data window for Gibson's phase rate criteria.

Phase Rate Trends: Pitch-TF

Severe PIO

Moderate PIO

No PIO

180 deg. Phase Lag Frequency (Hz) (f_180)
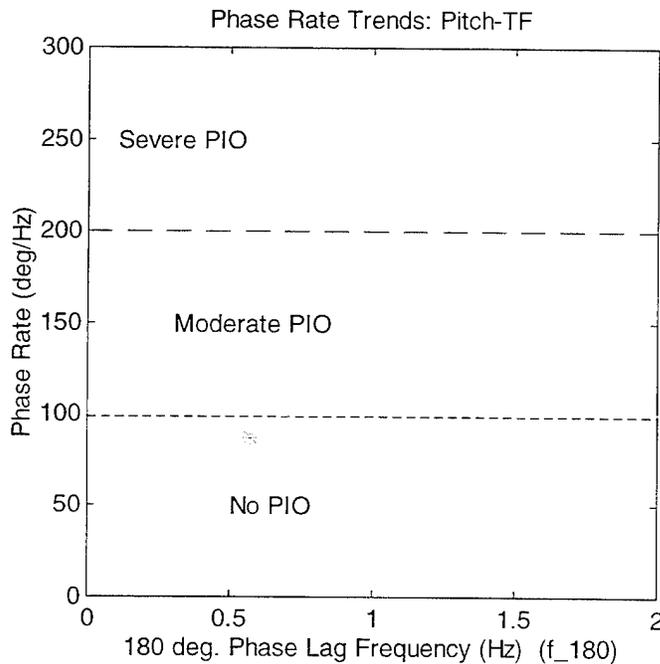
Figure 2 - 15. Plot window for Gibson's Phase rate criteria.

2.6 CAP Criteria

A criteria developed for the analysis of classical short period response types. It relates the Control Anticipation parameter (CAP $= \omega_{sp}^2 g T_{\theta 2} / V_{True}$) and short period damping ratio to flying qualities levels. If the aircraft transfer function is not given in a short period approximation format, an equivalent systems analysis must be performed. Under this condition, the software automatically switches you to an equivalent systems control panel which is discussed Section 2-11.

The parameters of low order equivalent system (short period approximation) are displayed in the text boxes of the CAP control panel (see Figure 2-16). The short period natural frequency and damping ratio are **wsp** and **zeta_sp** respectively. The equivalent time delay is **tau,** while the short period numerator time constant is **T_Theta_2**. The gain of the short period transfer function is **K**.

Short Period Parameters	
K	1
T_Theta_2	1
tau	0.1
zeta_sp	0.57735
wsp	1.7321

CAP/Airspeed	
TAS (ft/s)	100
CAP (1/(g s^2))	0.966
	OK

CAP Criteria

◉ Cat. A Requirement

○ Cat. B Requirement

○ Cat. C Requirement

Additional Requirements for

Class I, II-C, IV Aircraft ⬇

wsp	N/A
n / alpha	N/A
tau	Level 1

Print

Close
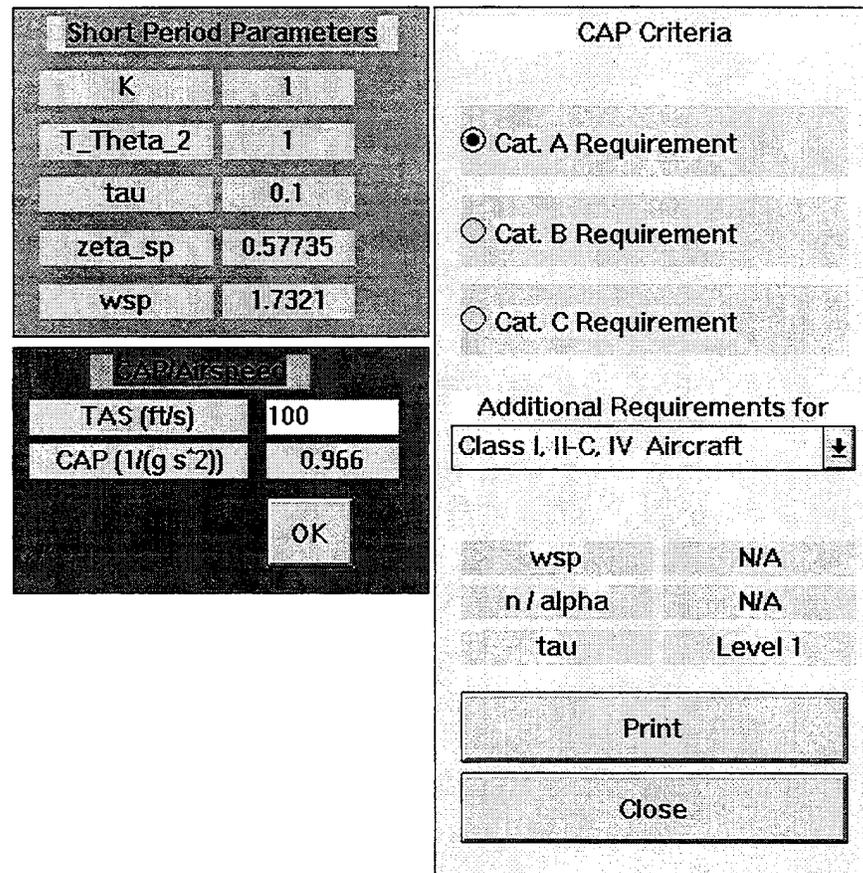
Figure 2 - 16. Control panel and data window for CAP Criteria.

To predict the flying qualities level of a particular aircraft using the CAP criteria enter the airspeed in the **CAP/Airspeed** frame and click **OK**. Note that if the **Transient Peak Ratio** criteria is open, the change in airspeed will cause that criteria to update automatically. Next click the radio button associated with the flight phase category to

obtain a plot of the appropriate level boundary for the aircraft category which you are analyzing. The flying qualities level can be determined by noting where the asterisk appears on the CAP- ζ_{so} plane and taking note of the additional requirements related to aircraft class located just above the print button in the CAP data window. Aircraft class can be selected by clicking on the pop-up menu.

An example of the CAP plot window for Category A is shown in Figure 2-17 . (Note that the * for this example is located under the second "e" in the "Level 1" text.)
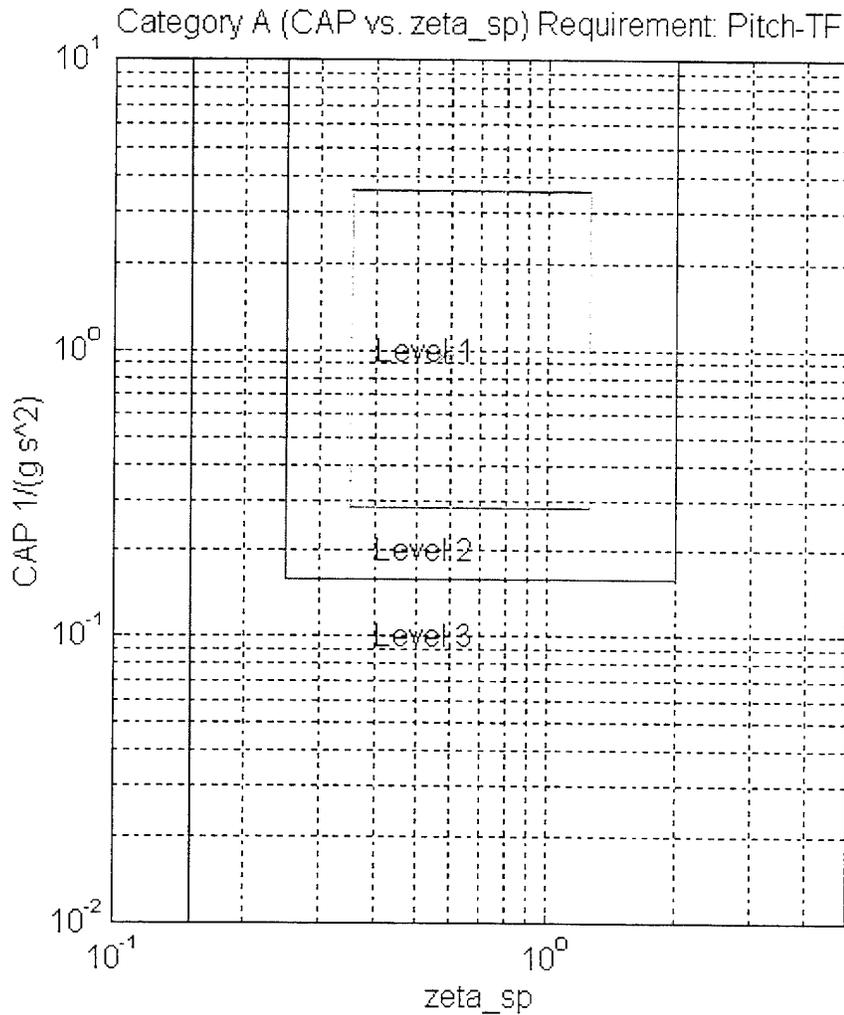
Figure 2 - 17. CAP criteria for Category A flight phase.

2.7 $\omega_{sp} - T_{\theta_2}$ Criteria

A criteria developed for the analysis of classical short period response types. It relates ω_{sp} and short period damping ratio to a flying qualities level. If the aircraft transfer function is not given in a short period approximation format, an equivalent systems analysis must be performed. The software automatically switches you to an equivalent systems control panel if necessary. See Section 2-11 for Equivalent Systems Analysis.

The parameters of low order equivalent system (short period approximation) are displayed in the text boxes of the control panel shown in Figure 2-18. The short period natural frequency and damping ratio are **wsp** and **zeta_sp** respectively. The equivalent time delay is **tau,** while the short period numerator time constant is **T_Theta_2**. The gain of the short period transfer function is **K**.

Short Period Parameters	
K	1
T_Theta_2	1
tau	0.1
zeta_sp	0.57735
wsp	1.7321

Wsp-T_Theta_2 Criteria

◉ Cat. A Requirement

○ Cat. B Requirement

○ Cat. C Requirement

Additional Requirements for

Class I, II-C, IV Aircraft	⬇

wsp	Level 1
1/T_Theta_2	N/A
tau	Level 1

Print
Close

Figure 2 - 18. Control panel for $\omega_{sp} - T_{\theta_2}$ criteria.

To predict the flying qualities level of a particular aircraft using the criteria click the radio button associated with the flight phase category to obtain a plot of the flying qualities boundaries. The flying qualities level can be determined by noting where the asterisk appears on the $(\omega_{sp} - T_{\theta_2})$- ζ_{sp} plane and taking note of the additional requirements related to aircraft class located just above the print button in the data window. Aircraft class can be selected by clicking on the pop-up menu.

Figure 2-19 shows the plot window for Category A. (Note that the * is on the Level 1 boundary.)
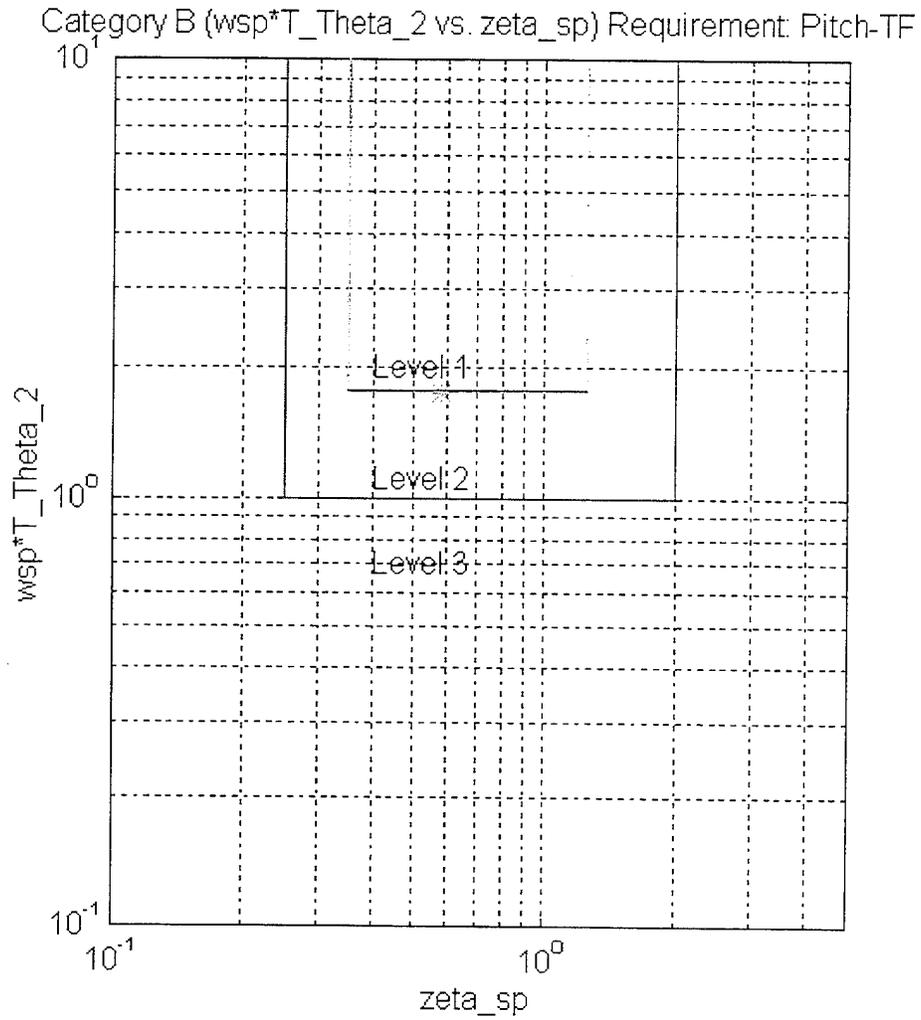
Figure 2 -19. $\omega_{sp} - T_{\theta_2}$ criteria for Category A flight phase.

2.8 Northrop Criteria

A frequency domain criteria for the pitch rate response. This criteria was developed for up and away flying tasks for fighter aircraft. Magnitude and phase envelopes are shown. If the frequency response is outside of those envelopes, flying qualities are predicted to be poor. The Northrop Plot and Control Panel is shown in Figure 2-20.
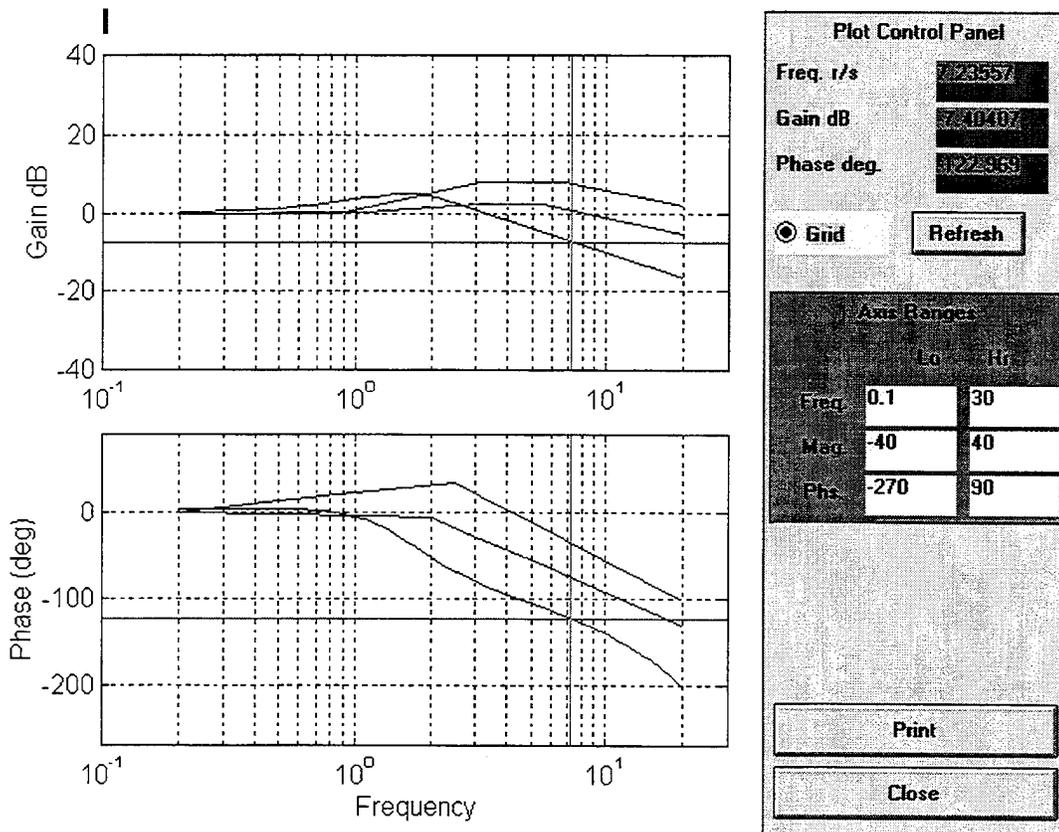
Figure 2-20. Plot and Control Panel for the Northrop criteria.

2.9 Gibson's Nichols Chart Criteria

A frequency domain criteria for the pilot/vehicle pitch attitude response. This criteria assumes that the pilot compensation can be described by a pure gain and time delay. It suggests that the pilot adjusts the gain such that the 0 dB crossover frequency is 0.3 Hz. Flying qualities characteristics are inferred from the comments on the Nichols chart.

The plot window shown in Figure 2-21 will appear after selecting the Gibson's Nichols Chart radio button from the main menu. The pilot/pitch attitude frequency response will be plotted along with the criteria boundaries and comments.
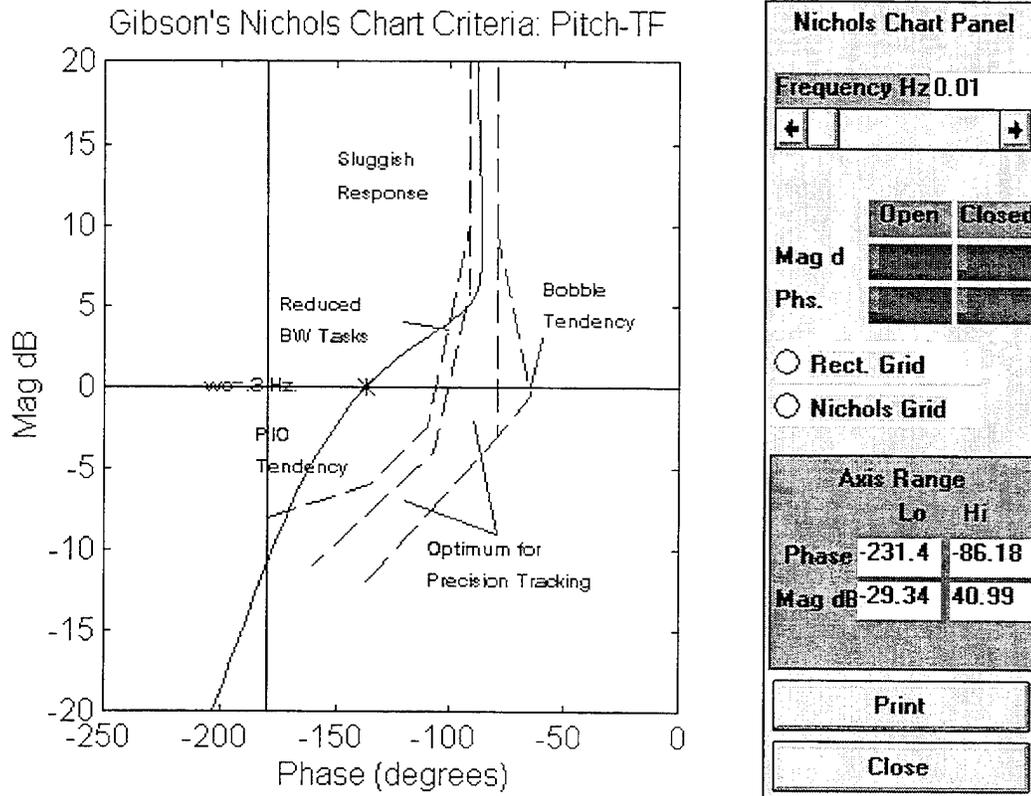
Figure 2 - 21. Gibson's Nichols chart criteria.

2.10 Ralph Smith Criteria

A multi-transfer function   time and frequency domain criteria for the longitudinal response of an aircraft. Currently only pitch attitude response portions of the criteria are implemented. The load factor portion of this criteria has not yet been implemented.

By selecting the **Ralph Smith Criteria** from the longitudinal main menu, the data window shown in Figure 2-22 will appear.

Slope (dB/oct)	-9.97717	
w_c (rad/sec)	3.60548	
phase_wc (deg)	-180.353	
t_qpeak (sec)	1.21176	
Notes: Use Transient Peak Ratio to verify t_qpeak. Does not include n_z portion of criteria.		
Print		Close

Figure 2 -22. Control Panel for the pitch portion of the Ralph Smith criteria.

Important criteria parameters are displayed in the text boxes of the control panel. The average slope of the magnitude curve between 2 and 6 rad/sec is given by **Slope.** The estimated pilot/vehicle gain crossover frequency is **w_c**, while the phase angle at this frequency is **phase_wc**. The peak value of the pitch rate step response is given by **t_qpeak**. To obtain a plot of the pitch rate step response, click on the **Transient Peak Ratio** button in the longitudinal main menu. Parameters which influence flying qualities levels have level indicators to the right of their respective text boxes. This configuration is predicted to be uncontrollable by this criteria as indicated by the **CHR 10** (Cooper-Harper Rating).

A bode plot of the pitch attitude transfer function will automatically be generated (see Figure 2-23. A line of best fit which matches the average slope of the magnitude curve between 1 and 6 rad is also displayed on the plot.
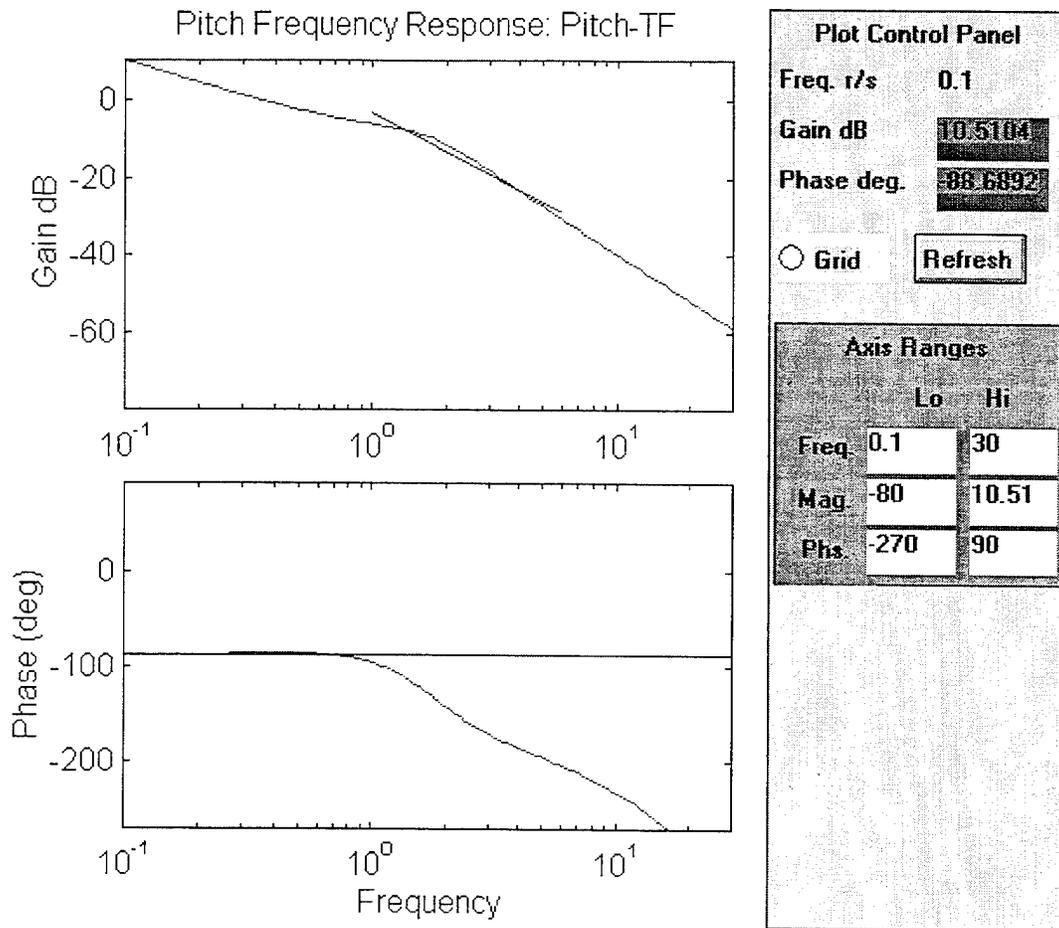
Figure 2 - 23. Bode plot of the pitch attitude transfer function.

2.11 Equivalent Systems

The ω_{sp}-$T_{\theta 2}$ and CAP criteria require classic modal parameters such as ω_{sp} and ζ_{sp}. In order to apply these criteria to aircraft configurations that are higher order, an equivalent system must be calculated. The method used in this toolbox involves a weighted sum of squares of the frequency response differences in magnitude and phase between the lower order equivalent system (LOES) and the higher order system (HOS) at n discrete frequencies. This method is described in MIL-STD-1797A, Appendix B, p. 681. The following cost function is minimized:

$$J = \sum_{i=1}^{n}\left[\left(\|HOS_i\|_{dB} - \|LOES_i\|_{dB}\right)^2 + 0.01745\left(\angle(HOS_i) - \angle(LOES_i)\right)^2\right]$$

where

- gain is in dB
- phase is in degrees
- i denotes the *ith* input frequency
- n is the number of discrete frequencies

The short period approximation is used for the LOES:

$$LOES = K\frac{T_{\theta 2}s + 1}{s(s^2 + 2\zeta_{sp}\omega_{sp}s + \omega_{sp}^2)}e^{-\tau s}$$

The parameters K, $T_{\theta 2}$, τ, ζ_{sp}, and ω_{sp} are systematically varied until the cost function J is minimized.

In this example we shall consider the following higher order system:

$$\frac{\theta(s)}{\delta(s)} = \frac{s + 1}{s(s^2 + 2 \cdot 0.5 \cdot 3s + 3^2)(s + 5)}$$

Since the modal parameters are specified, it is convienient to enter the transfer function using short hand notation. Select **Short Hand (senter)** using the popup menu in the main control panel and enter the numerator and denominator parameters in short hand notation as shown in Figure 2-24

```
┌─────────────────────────────────────────────────────────┐
│                            Theta/delta Transfer Function │
│        Pitch-TF        0    ┌──────────────────────────┐ │
│                             │ Short Hand (senter)    ±│ │
│                             └──────────────────────────┘ │
│                                                          │
│  Numerator                         Delay (sec)           │
│  [ 1   1 1 ]                       0                      │
│                                                          │
│  Denominator                                             │
│  [ 3  1 0   2 .5 3   1  5 ]                               │
│                                              ┌─────────┐ │
│                                              │  Done   │ │
│                                              └─────────┘ │
└─────────────────────────────────────────────────────────┘
┌─────────────────────────────────────────────────────────┐
│                         Criteria                         │
│                                                          │
│   ○ Transient Peak Ratio        ○ CAP Criteria           │
│                                                          │
│   ○ Gibson Dropback             ○ Wsp-T_Theta_2 Criteria │
│                                                          │
│   ○ Bandwidth Criteria          ○ Northrop Criteria      │
│                                                          │
│   ○ Neal-Smith Criteria         ○ Gibson's Nichols Chart │
│                                                          │
│   ○ Gibson's Phase Rate         ○ MOCM Pilot Model       │
│                                                          │
│                                 ○ Ralph Smith            │
│                                                          │
│   ┌──────────────┐              ┌──────────────┐         │
│   │    Print     │              │   Close All  │         │
│   └──────────────┘              └──────────────┘         │
└─────────────────────────────────────────────────────────┘
```

Figure 2 - 24. Main control panel for equivalent systems example.

The **Numerator** field appears as [1 1 1] which is interpreted as follows: There is one numerator factor, of order 1 with a zero at 1. The **Denominator** field appears as [3 1 0   2 .5 3   1 5] which is interpreted as follows: There are 3 denominator factors, the first factor is of order 1 with a pole at 0 (i.e. a free s), the second factor is of order 2 with a damping of .5 and a natural frequency of 3 and the third factor is of order 1 with a pole at 5. Spaces between the factors are transparent to the operation and are recommended for organization purposes.

If a higher order system is typed into the main menu and either the CAP or ω_{sp}-$T_{\theta2}$ criteria are selected, the Equivalent Systems control panel of Figure 2-25 will appear.

Figure 2 - 25. Equivalent Systems control panel.

You can change initial guesses or the frequency range of the match as needed. $T_{\theta2}$ can be fixed or allowed to vary with the rest of the parameters. As indicated above, click on the "LOES fit" radio button to begin the match. You do not have to enter true airspeed to get a match. However, once the match is complete, true airspeed is needed to run the CAP criteria. If the **Transient Peak Ratio** criteria is open while the airspeed is changed, both criteria will update. To run the ω_{sp}-$T_{\theta2}$ criteria, you do not have to return to the main menu. Simply select the criteria from the pull down menu above (CAP is shown active for this example). The LOES will be the same for both criteria.

When a LOES is found, the following plot window shown in Figure 2-26 will be generated.
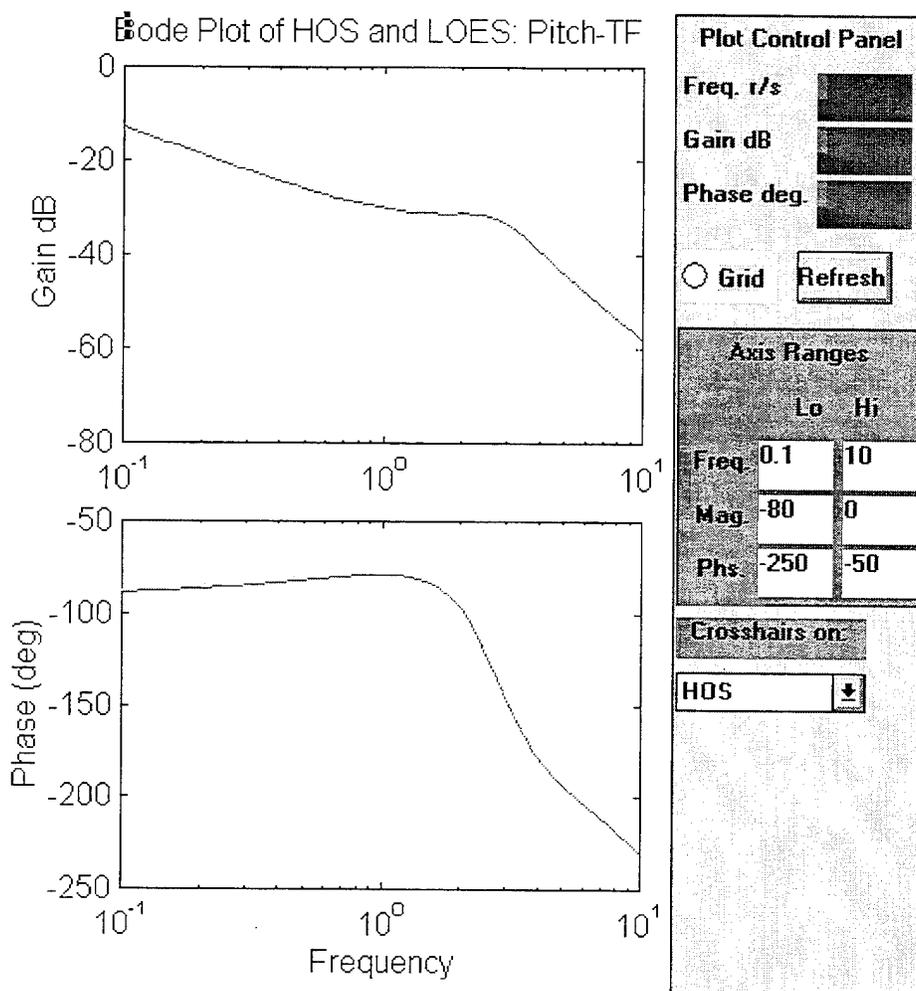
Figure 2 - 26. Comparison of HOS and LOES bode plots.

This example shows a close match between the HOS and the LOES. The next step is to determine if the match is acceptable by checking to see if the mismatch remains within the magnitude and phase envelopes defined by MIL-STD-1797A.

Clicking on the "LOES/HOS Mismatch" button will generate the plot shown in Figure 2-27.
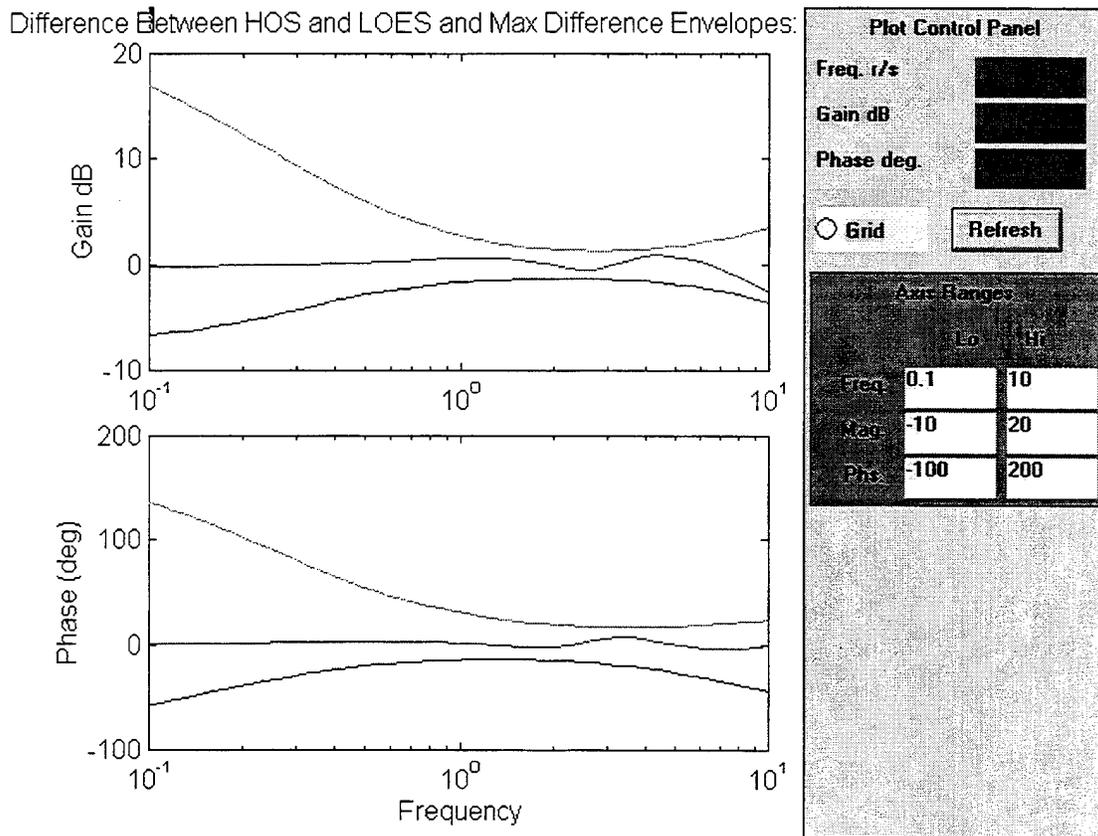
Figure 2 - 27. Regions of acceptable mismatch.

If the mismatch stays within the magnitude and phase envelopes, the match is acceptable. If not, try changing the initial guesses or varying the frequency range over which the match is performed.

2.12 Modified Optimal Control Pilot Model

The MOCM generates a mathematical model of a human pilot engaged in a tracking task. The MOCM assumes that a pilot adjusts his compensation to minimize a weighted sum of the squares of tracking error, error rate, control deflection and control rate. The compensation is subject to human limitations such as the pilots time delay and the inability of a limb to respond to thought instantaneously with infinite precision. A detailed discussion of the MOCM is provided in Chapters 4-6 of this manual. This tutorial covers the use of a GUI front end designed to facilitate a single axis analysis of a pitch tracking or stabilization task. The MOCM control panel for this example is shown in Figure 2-28.

Figure 2-28. MOCM control panel.

This example covers a pitch attitude compensatory tracking task. A compensatory task is one in which the pilot's objective is move a manipulator in such a way as to minimize tracking error given a display of the difference between a desired position and the vehicles actual position. A simplified block diagram of such a scenario is shown in Figure 2-29.
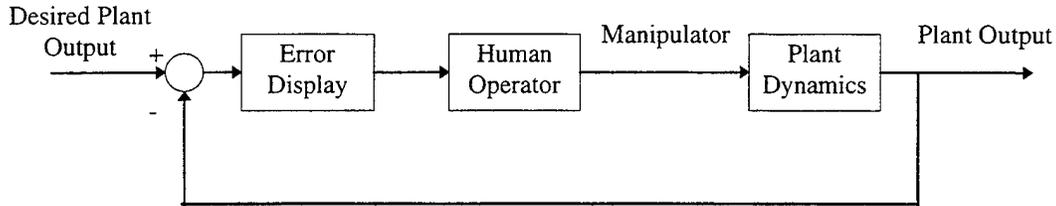
Figure 2 - 29. Simplified block diagram of a compensatory tracking task.

We will assume that the pitch attitude to stick transfer function is given by:

$$\frac{\theta(s)}{\delta(s)} = \frac{s+1}{s(s^2 + 2s + 3)}$$

It should be noted that plants with pure time delay are not handled by the MOCM because there is currently no way of distinguishing between plant and pilot delay. (Warning! Plants with time delay will be analyzed with the delay set to zero.) For this reason it is recommended that higher order systems without pure time delay be used for MOCM analysis and not their lower order equivalent systems. If you want to include plant delays in spite of this warning, you can add the pilot delay and plant delay together an enter them in the pilot delay text box.

It will also be assumed that the desired output of the plant can be represented by a forcing function. The forcing function is to have a random appearance to the pilot. Such a forcing function can be generated by filtered white noise. Forcing functions with approximately rectangular spectra and prespecified RMS values can be generated by passing Gaussian white noise through a low pass Butterworth filter. The user can select from 1st through 4th order filters using the pull down menu. The higher the filter order the sharper the break frequency and the more the input spectrum appears rectangular. Forcing function bandwidth is a nebulous term. It is only well defined for forcing functions which have a rectangular spectrum. A method of computing an effective rectangular bandwidth for a nonrectangular spectrum was developed by Blackman and Tukey. This method of defining forcing function bandwidth has been widely used in the pilot modeling literature to form a basis for comparing analytical and experimental results. For this case we will use a second order low pass Butterworth filter with a break frequency of 1 rad/sec with an output RMS of 1 deg. The filter gain is automatically selected to produce the desired ouput RMS which is diplayed in the **Filter Gain (Vw=1)** text box. The effective rectangular bandwidth of the filter output is 1.481 rad/sec. It should be noted that the break frequency of the filter is not the same as the effective rectangular bandwidth of the filter output. For lowpass Butterworth filters of finite order, the effective rectangular bandwidth will always be greater than the filter break frequency. The output of the filter is injected as a disturbance which the pilot will attempt to reject.

This distubance can also be thought of as a negative forcing function which the pilot attempts to track. Since the pilot will be attempting to reject a pitch disturbance, we will inject a disturbance at the output of the plant using the popup menu adjacent to the **Inject Disturbance at:** text box.

A block diagram of the example system is shown in Figure 2-30.
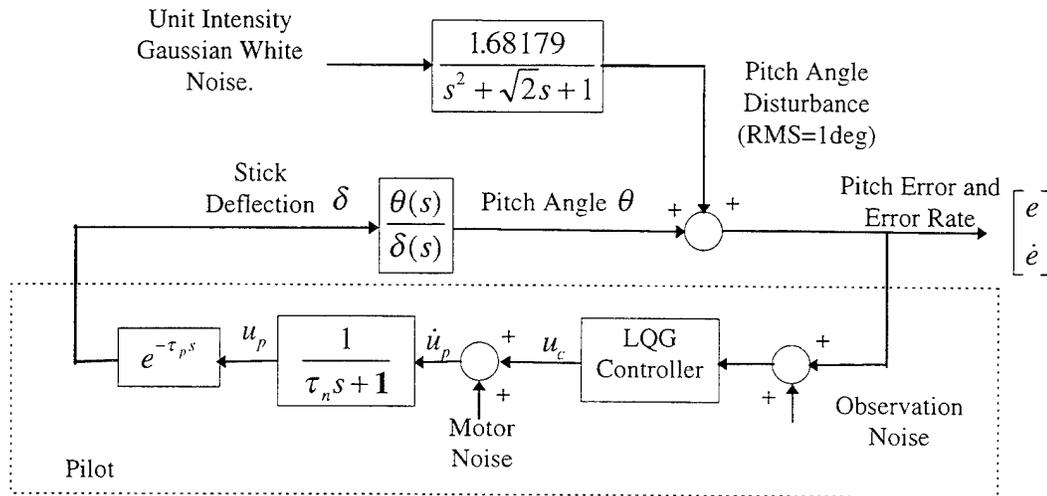
Figure 2 - 30. Block diagram of a compensatory pitch tracking task using MOCM.

In this case we will assume that the pilot will attempt to minimize RMS tracking error given a display of the error. Experimental evidence has shown that pilots can ascertain the first order derivative a displayed variable; therefore, we will assume that the pilot can obtain error rate information as well. It will be assumed that the pilot will attempt to minimize the following quadratic cost function.

$$J_p = E_\infty \left[\begin{bmatrix} e & \dot{e} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + g\dot{u}_p^2 \right]$$

The unity weight on error and null weight on error rate reflects the task: minimize RMS error. These weights are selectable under the **Cost Wght. Qy** column. The default weights are 1 for error and 0 for error rate. The weight on control rate reflects the physiological limitation that that pilot's limbs cannot respond infinitely fast to a thought command. The control rate weight g is driven by the selection of neuromotor time constant τ_n. The user does not explicitly set g, but must define τ_n which uniquely defines g. Values of τ_n range between 0.6 sec and 0.1 sec, where the latter is considered to be near the upper limit of human ability. For this example the default value of $\tau_n = 0.1$ will be used.

Experimental data have shown that pilot time delays typically range from 0.1 sec to 0.25 sec. This example will use the default pilot delay of $\tau_p = 0.2$ sec. The MOCM uses a second order Pade approximation of the pilot's time delay in order to generate a state space model or transfer function model of the pilot. Note the presence of the additive noises in Figure 2-30. The observation noise is a representation of the pilot's inability to percieve information from the error display with infinite resolution. Two independent Gaussian white noise processes are added to the error and error rate. The noise intensities are found such that the ratios of the noise intensities to error and error rate variances achieve a prespecified value. Experimental evidence suggests that noise to signal ratios of 0.01 are applicable to a wide range of foveal viewing conditions. These are used as defaults in the MOCM control panel. The motor noise models the pilot's inability to move the stick with infinite precision. The motor noise intensity is adjusted such that the ratio of the motor noise intensity to variance of the commanded control u_c, achieves a specified value. A motor noise to signal ratio of 0.00316 or -25 dB has been found to provide good matches to experimental data and is used as a default in the MOCM control panel.

The MOCM control panel provides a means to analyze more complex phenomena as well, such as indifference thresholds and divided attention. Indifference thresholds are a representation of how much error a pilot will tolerate before moving the stick to take corrective action. The default indifference thresholds are set to zero. The indifference thresholds are modeled using statistical describing functions of dead zone elements. The observation noise to signal ratios are then scaled accordingly. The effect of distractions on pilot performance can also be modeled by specifying an attentional fraction less than 1. The fraction of attention can be interpreted as the average amount of time a pilot spends looking at the error display over a given period of time. For example, if the pilot spent all of his time and effort trying to zero the error, the fractional attention would be 1. If he spent 75% of his time trying to zero the error and 25% of his time on other task (e.g. looking at another instrument), the attentional fraction would be 0.75.

It should be noted that MOCM analysis via the MOCM control panel is designed to be as simple as possible. Default values are provided for all input parameters so MOCM analysis can proceed without entering any additional data. For most analyses, it will not be necessary to change the noise to signal ratios, indifference thresholds or attentional fractions. Default values for pilot parameters represent realistic values based on experimental data. If the user has no information on pilot parameters, it is recommended that the default values be used.

After the input parameters have been selected or modified, the MOCM can be generated by clicking on the **Run MOCM** button. The following bode plot of the pilot transfer function $\delta(s)/e(s)$ is displayed in the plot window shown in Figure 2-31.

Yp (Pilot) Frequency Response - Pitch Axis: Pitch-
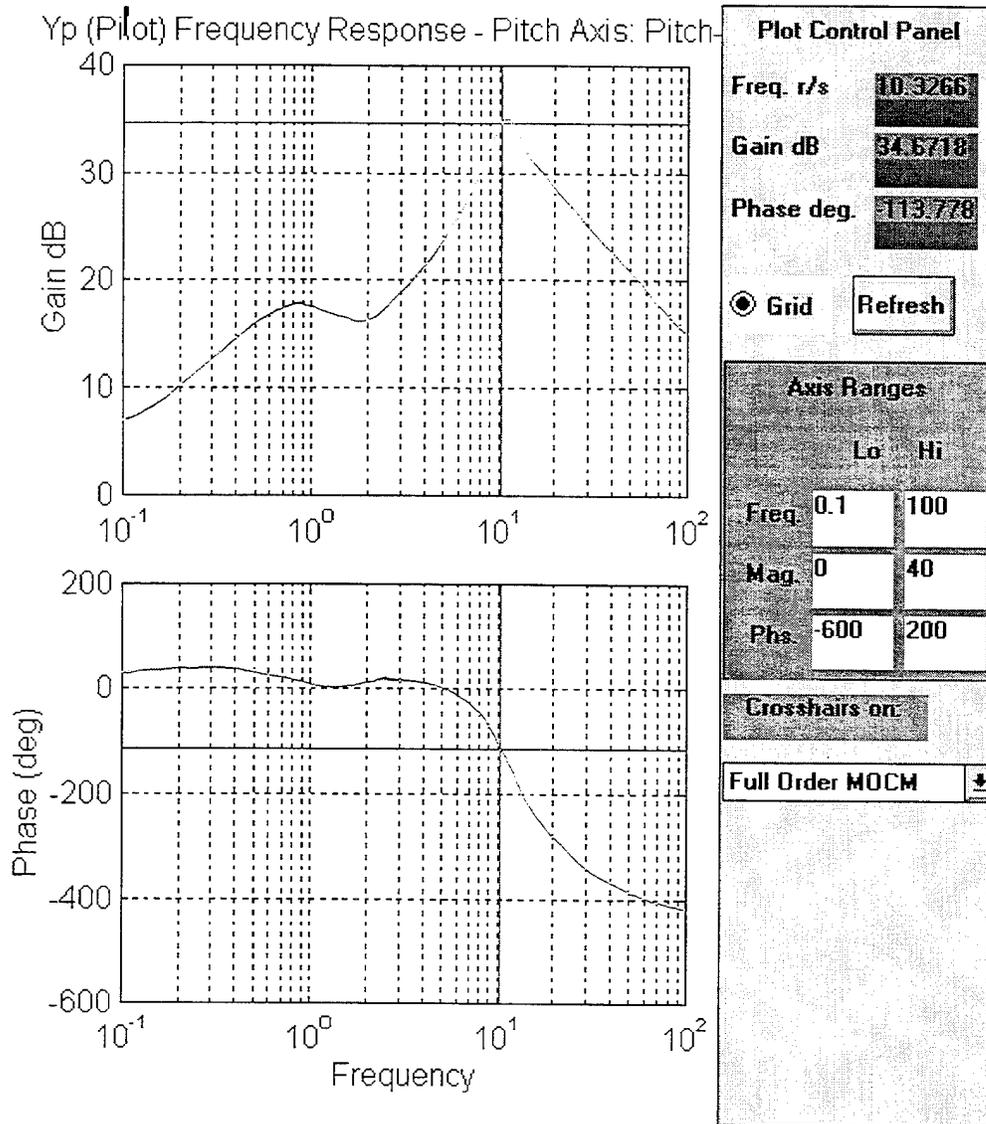
Figure 2-31. Bode plot of pilot transfer function for pitch tracking example.

Elements of the pilot compensation strategy can be inferred by observing the slope of the magnitude curve over different frequency ranges. The model predicts the pilot will generate lead compenation in the ranges $0.1 < \omega \leq 0.8$ rad/sec as indicated by the positive slope of the magnitude curve. It appears that the pilot will generate lag compensation in the range from $1 < \omega \leq 2$ rad/sec as indicated by the negative slope of the magnitude curve. More information can be obtained by investigating the characteristics of the open loop pilot/vehicle bode plot. The plot shown in Figure 2-32 was generated by clicking on the **Pilot/Vehicle Frequency Response** button and adjusting the axis scales. The open loop pilot/vehicle 0 dB crossover frequency is approximately 3.16 rad/sec. It should also be noted that over a wide frequency range centered on the crossover frequency, the slope of the magnitude curve is approximately -20 dB/dec. This is consistent with McRuer's

classical crossover model which predicts that in a compensatory tracking task, the pilot will adjust his compensation such that the open loop pilot vehicle will have the characteristics of a K/s plant with time delay in the region of gain crossover.
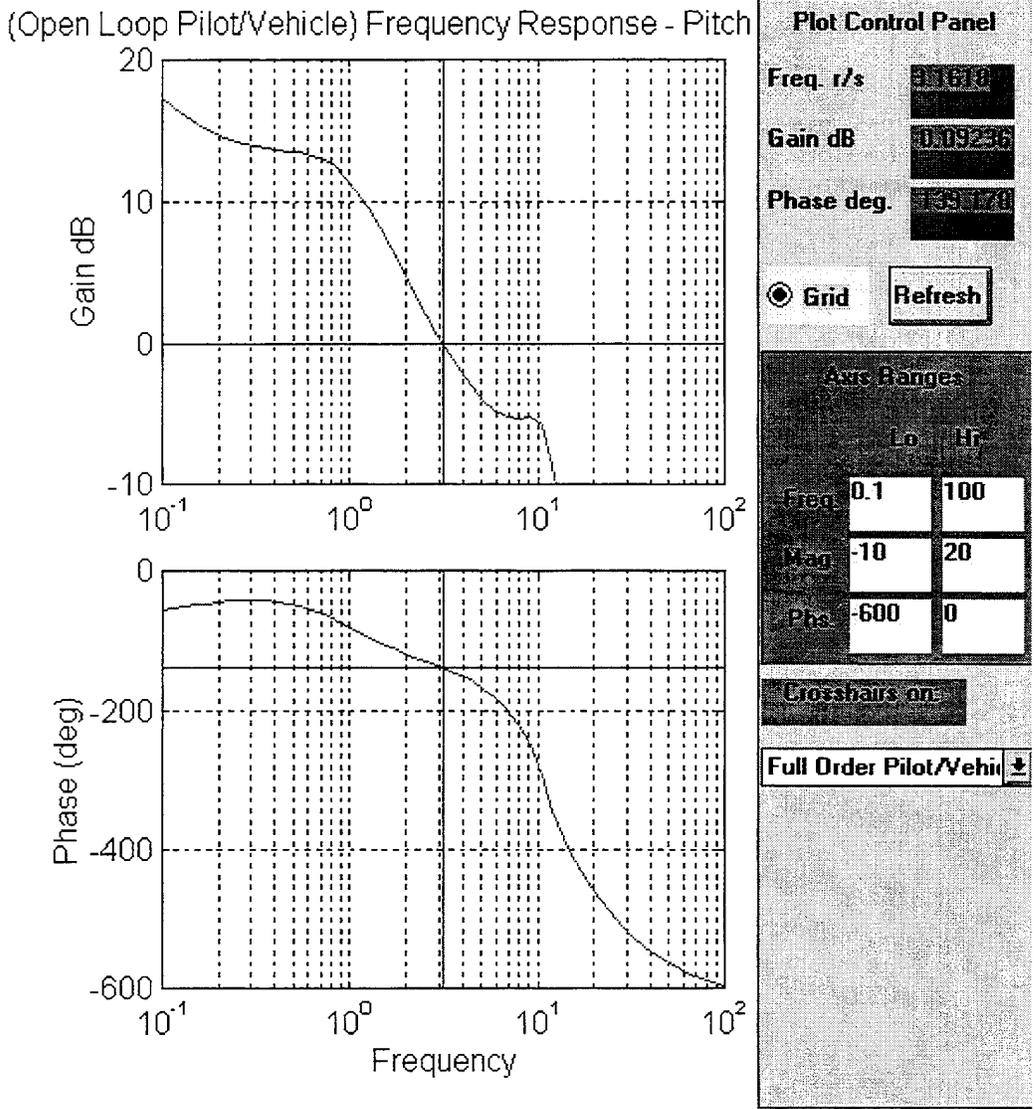
Figure 2-32. Bode plot of open loop MOCM pilot/vehicle.

The MOCM can be a useful tool for predicting the pilot/vehicle gain crossover frequency for various forcing function bandwidths. It is interesting to note that MOCM predicts the phenomenon of crossover regression. Crossover regression means the pilot/vehicle crossover frequency for a particular forcing function bandwidth is much lower than the crossover frequency for a very low forcing function bandwidth. There are circumstances when a pilot can achieve better performance (as measured by the cost function) by

crossing over at frequencies lower than the crossover frequency for a very low bandwidth forcing function.

The factors of the pilot's transfer function can be obtained by clicking on the **MOCM Factors** button. Note that the format of the factors is similar to that of the **Control System Toolbox damp** command. Physical interpretation of some factors will now be advanced.

Pilot Gain	Numerator Factors		Denominator Factors	
550.616503	Damping	Frequency	Damping	Frequency
	-0.707107	14.142136	0.707442	0.999968
	-0.707107	14.142136	0.707442	0.999968
	1.000000	0.124892	1.000000	1.001008
	0.586623	1.707859	0.242114	10.866005
	0.586623	1.707859	0.242114	10.866005
	1.000000	1.897751	1.000000	3.610548
	1.000000	3.635504	1.000000	10.000000
	0.707107	14.142136	0.707107	14.142136
	0.707107	14.142136	0.707107	14.142136
	1.000000	10.029993	0.819359	19.433505
			0.819359	19.433505

The pilot delay is represented by the ratio of second order factors: [-.707107, 14.142136] / [-.707107, 14.142136]. This is the Pade approximation of the delay given by $e^{-.2s}$. The neuromotor lag is represented by the first order denominator factor [1 , 10]. The remaining terms represent pilot's compensation strategy. Various reduction techniques are available via push button including pole-zero cancellations, balanced realizations and crossover equivalent realization. Descriptions of the reduction techniques are described in Chapter 5. Statistics and numerical information related to the MOCM analysis can be obtained by clicking on the **MOCM Stats./Info. button.**

MOCM Statistics/Information

0.1794 = J(Cost Function)
  1 = Error Weight in Cost Function
  0 = Error Rate Weight in Cost Function
0.00003295 = Control Rate Weight in Cost Function
0.3765 = RMS Error
 1.164 = RMS Error Rate
 4.455 = RMS Control Deflection
 33.82 = RMS Control Rate
0.06671 = RMS Error Observation Noise
0.2063 = RMS Error Rate Observation Noise
0.5315 = RMS Motor Noise
 -20 = Error Noise to Signal Ratio dB
 -20 = Error Rate Noise to Signal Ratio dB
 -25 = Motor Noise to Control Signal Ratio
 2.7 = Estimated CHR

The value of the cost function is displayed, as well as the weight selections on error, error rate and control rate. RMS statistics are also displayed as well as the noise to signal ratios in power dB. An estimate of the Cooper Harper rating is also given. This estimate is based upon an empirical formula given in WL-TR-89-3125 vol. II. See Chapter 5 for more details.

Changing any of the input parameters in the MOCM Control panel causes the plot window to be cleared and the **Run MOCM** radio button to be turned off. To generate another MOCM based on the updated information, click on the **Run MOCM** button.

3. IFQT Reference

Although the IFQT is designed to run using a graphical user interface, users or developers may need access to the functions which generate the numerical results. This section begins with a list of functions grouped by subject and continues with the reference entries in alphabetical order. Descriptions of functions included in the IFQT which are useful by themselves are included in this section. The MOCM reference section is in Chapter 6.

Interactive Plots for Control Analysis	
bodeplot	Creates an interactive bode plot with multiple trace capability and crosshairs
nichplot	An interactive Nichols plot facility with crosshairs
thist	Generates an interactive time history plot window with crosshairs

Model Generation	
senter	Enter a transfer function in shorthand notation

Short Term Pitch Response Criteria	
bw	Calculates bandwidth criteria parameters.
gdb	Generates criteria parameters for Gibson's Dropback Criteria.
gpr	Computes parameters for Gibson's Phase Rate Criteria.
north	Generates data for Northrop frequency response envelope.
ns	Determines a pilot model based on Neal-Smith adjustment rules
nsout	Calculates parameters for the Neal-Smith Criteria.
rs	Computes pitch response parameters for Ralph Smith's criteria
tpr	Calculates time domain parameters for Transient Peak Ratio and Ralph Smith criteria.

bodeplot

Purpose

Creates a multi trace interactive bode plotter with crosshairs and a sidebar control panel for an arbitrary set of magnitude and phase data.

Synopsis

[bode_plot,bode_axes]=bodeplot('start',bode_plot,w,mag,phase,ptitle,traces);

Description

The output arguments **bode_plot** and **bode_axes** must be specified. They keep track of the figure window and individual semilog axes which comprise the bode plot. The function makes explicit reference to these variables when the mouse buttons are used to move the crosshairs. The input arguments are:

w; The frequency vector which generated the magnitude and phase data

mag and **phase;** either column vectors or matrices whose columns contain magnitude and phase data for each trace on the bode plot e.g.
        mag=[mag1 mag2 .... magn] ;phase=[phase1 phase2 .... phasen];

ptitle; A string for the plot title e.g. 'Bode Plot'

traces; An optional argument specifying names for each plot trace. The default trace string is: 'Trace 1 | Trace 2 | ..... Trace n'. The user can specify the names using the same format e.g. 'Theta | Nz | Alpha'.

bodeplot uses magnitude, phase and frequency data generated by the control system toolbox **bode** command or other sources (e.g. flight test data, **bode** generated data modified for time delay) to create an interactive plotting environment. Crosshairs are created when the user clicks on the plot window. Crosshairs can be moved with the mouse by clicking and dragging. The position of the crosshairs is displayed in the sidebar. Traces may be selected using the popup menu. Axes may be scaled by entering the range in the appropriate text box. A grid may turned on or off by clicking on the **Grid** radio button. Occasionally the crosshairs on the phase plot may become inoperable. This is usually the result of covering the plot window with another window. The phase crosshairs may be reactivated by clicking on the **Refresh** button in the sidebar.

See Also

Control System Toolbox; bode

bw

Purpose

Generate bandwidth criteria data for a transfer function with time delay.

Synopsis

[w45pm,w6db,wbw,taup,w180,mag6db,mag180,w,mag,phase,phidot180,fr_range] = bw(num,den,tau);

Description

Data for the bandwidth criteria are generated for a transfer function with time delay. The numerator and denominator must be given in descending powers of s and the time delay is given in seconds. The outputs are as follows:

w45pm; The frequency at which the phase margin is 45 degrees.

w6db; The frequency at which the magnitude curve is 6 db higher than the magnitude curve at ω_{180}.

wbw; The "bandwidth," the lesser of **w45pm** and **w6db.**

taup; The equivalent time delay defined by $-\dfrac{\phi_{2\times180}+180^{\circ}}{57.3\omega_{2\times180}}$

w180; The frequency at which the phase curve crosses -180 deg.

mag6db; The value of the magnitude curve at **w6db.**

mag180; The value of the magnitude curve at **w180.**

w,mag,phase; The bode data for the transfer function with delay evaluated at 10,000 frequency points. This high frequency resolution was used in the in-house crossmapping effort where extreme accuracy was required.

phidot180; The slope of the phase curve at the 180 deg phase cross-over frequency expressed in deg/Hz. This parameter is used by Gibsons phase rate criteria.

freq_range; The upper and lower limits of the frequency range over which the bode parameters are computed. The appropriate frequency range is automatically determined such that all of the critical parameters can be found.

See Also

Control System Toolbox; bode.

References

[1] Anon: MILSTD-1797A "Flying Qualities of Piloted Aircraft," Jan 30, 1990 . ASD/ENES, Wright Patterson AFB OH.

[2] Hoh, Roger H,. Mitchel, David G. and Hodgkinson, John, "Proposed Military Standard and Handbook -- Flying Qualities of Air Vehicles, Vol II, Proposed Handbook," AFWAL-TR 82-3081, 1982.

[3] Mitchell, David G., Hoh, Roger H., Aponso, Bimal L., and Klyde, David., "Incorporation of Mission Oriented Flying Qualities into MIL-STD-1797A." WL-TR-94-3162, 1994.

gdb

Purpose

Generate Gibsons Dropback criteria data for a transfer function with time delay.

Synopsis

[tout,thout,thend,db,qss,dbperq,theta,t,qmax]=gdb(num,den,tau);

Description

Data for Gibsons Dropback criteria are generated for a transfer function with time delay. The numerator and denominator must be given in descending powers of s and the time delay is given in seconds. The criteria parameters are the based on an analysis of the pitch attitude response to a boxcar input (i.e. a pulse). The pulse duration is computed by finding the time required to reach a steady-state pitch rate and adding the time delay. The control system toolbox function **timevec** is used to determine the required pulse duration. The time history consists of 10,000 points between 0 and twice the pulse duration.

The outputs are as follows:

tout; The time at which the pulse transitions from 1 to 0.

thout; The pitch attitude at **tout.**

thend; The steady-state pitch attitude after the pulse is removed.

db; The dropback parameter, defined as **thout - thend.**

qss; Steady-state pitch rate due to a step input. Calculated by using the control systems toolbox command **dcgain** on the pitch rate transfer function,

dbperq; The ratio of the dropback parameter and steady-state pitch rate. (i.e. **db/qss**).

theta and **t;** The 10,000 point time history of the pitch attitude response to a boxcar input of duration **tout**.

qmax; The maximum pitch rate achieved during the pitch rate response to a boxcar input of duration **tout**.

See Also
Control Systems Toolbox; lsim, dcgain, stepfun, timevec

References
[1] Anon: MILSTD-1797A "Flying Qualities of Piloted Aircraft," Jan 30, 1990 .
ASD/ENES, Wright Patterson AFB OH.

[2] Gibson, J. C., "Handling Qualities for Unstable Combat Aircraft," ICAS-86-5.3.4,
Sept 1986.

[3] Reynolds, P.A., "Criteria for Handling Qualities in Military Aircraft; Simulation for
Predicting Flying Qualities," AGARD-CP-333, June 1982.

gpr

Purpose

Generate Gibsons Phase Rate criteria data for a transfer function with time delay.

Synopsis

[f180,phidot180]=gpr(num,den,tau);

Description

Data for Gibson's Phase Rate criteria are generated for a transfer function with time delay. The numerator and denominator must be given in descending powers of s and the time delay is given in seconds. The criteria parameters are based on an analysis of the pitch attitude frequency response plot. The slope of the phase curve at the -180 deg phase crossover frequency is defined as the phase rate. The phase rate is expressed in deg/Hz or equivalently deg/cps. Gibson uses phase rate as a measure of PIO susceptibility.

The outputs are as follows:

f180; The -180 deg phase crossover frequency in Hz or cps.

phidot180; The slope of the phase rate curve at **f180** in deg/Hz.

See Also

bw, Control Systems Toolbox; bode

References

[1] Anon: MIL-STD-1797A "Flying Qualities of Piloted Aircraft," Jan 30, 1990 . ASD/ENES, Wright Patterson AFB OH.

[2] Gibson, J. C., "Handling Qualities for Unstable Combat Aircraft", ICAS-86-5.3.4, Sept 1986.

[3] Reynolds, P.A., "Criteria for Handling Qualities in Military Aircraft; Simulation for Predicting Flying Qualities," AGARD-CP-333, June 1982.

nichplot

Purpose

Creates an interactive Nichols chart with crosshairs and a sidebar control panel for an arbitrary set of magnitude and phase data.

Synopsis

[nich_plot]=nichplot('start',nich_plot,w,mag,phase);

Description

The output arguments **nich_plot** must be specified. The function makes explicit reference to this variables when the mouse buttons are used to move the crosshairs. The input arguments are:

w; The frequency vector which generated the magnitude and phase data

mag and **phase;** either column vectors of magnitude and phase data.

nichplot uses magnitude, phase and frequency data generated by the control system toolbox **bode** command or other sources (e.g. flight test data, **bode** generated data modified for time delay) to create an interactive plotting environment. Crosshairs can be moved with the mouse by clicking the mouse button while on a point on the locus. Crosshairs may also be moved using the slider or entering the frequency in the text box above the slider. The open and closed loop magnitude and phase is displayed in the sidebar. Axes may be scaled by entering the range in the appropriate text box. Both rectangular and Nichols grids may turned on or off by clicking on the appropriate radio button.

See Also

Control System Toolbox; bode

north

Purpose

Generate bode data for the Northrop pitch rate response criteria.

Synopsis

[wri1,wri2,mag1,mag2,phz1,phz2,w,mag,phase]=north(num,den,tau);

Description

north computes the frequency response data for the pitch rate transfer function using the pitch attitude transfer function as an input. The pitch attitude transfer function is given in descending powers of s and the delay is given in seconds. The magnitude and phase envelopes which form the criteria boundaries are returned as well. The pitch rate frequency response is normalized such that the steady-state response lies on the zero db line. For systems where the steady-state pitch rate response is zero (i.e. systems with 0 DC gain), the system is normalized by the average value of the magnitude curve between 0.2 and 1 rad/sec.

The outputs are as follows:

wri1,wri2; The complex frequency response data required to generate the upper and lower criteria boundaries respectively.

mag1,mag2; Upper and lower magnitude criteria boundaries in dB.

phz1,phz2; Upper and lower phase criteria boundaries in deg.

w,mag,phase; Bode data for the normalized pitch rate transfer function evaluated at 50 frequency points logarithmically spaced between 0.2 and 20 rad/sec.

See Also

Matlab Reference; abs, angle, unwrap; Control Systems Toolbox; bode.

References

[1] Iloputaife, Obi and Ma, Lily, "Handling Qualities Design of a Northrop High-Performance Fighter," AIAA-87-2450, 1987.

ns

Purpose

Determines a lead or lag-lead pilot model based on the adjustment rules specified by the Neal-Smith criteria.

Synopsis

[p,warn]=ns(num,den,tau,wtarget);

Description

ns finds the parameters of a lead or lag-lead pilot model which makes the open loop pilot/vechicle transfer function cross the -90 deg closed loop phase contour at a specified frequency and causes the low frequency closed loop droop to touch the -3 dB contour. When using this function, the pilot delay must be added to the plant delay (i.e. **tau** $= \tau_{pilot} + \tau_{plant}$). The pilot model can take one of the following forms:

$$Y_p(s) = K_p(T_L s + 1)e^{-\tau_{pilot} s}$$

or

$$Y_p(s) = K_p \frac{(T_L s + 1)}{(T_I s + 1)} e^{-\tau_{pilot} s}$$

depending on which type of compensation is required. The **ns** algolrithm automatically determines which type of compensation is required by looking at the phase of the open loop plant at the target frequency or bandwidth and/or the minimum closed loop magnitude between .1 rad/sec and the target bandwidth. The **constr** function from the optimization toolbox is used to adjust the parameters in the appropriate pilot model to meet the objectives. The problem is posed as a one dimensional constrained optimization problem where the model parameters are adjusted such that the squared distance between the minimum closed loop pilot/vehicle magnitude between .1 rad/sec and the target bandwidth and the -3 dB closed loop magnitude contour is minimized. One of the following is used in this process, **leadcost** or **lagcost.** These two functions calculate the the pilot gain required to force the pilot/vehicle frequency response to cross the   -90 deg closed loop contour to at the target bandwith as well as the cost required for the **constr** function. In some cases, the -3 dB requirement may not be met or the pilot model may not be able to produce enough compensation to meet the objectives. Under these circumstances, a warning message is returned in the variable **warn**. The default **warn** message is "None." The output variable **p** is a vector containing the pilot parameters which meet the objectives stated above. The elements of **p** are as follows $\mathbf{p} = \begin{bmatrix} K_p & T_L & T_I \end{bmatrix}$.

See Also.

nsout; Optimization Tool box; constr;

References

[1] Neal, T. P. and Smith, R.E., "An In-Flight Investigation to Develop Control System Design Criteria for Fighter Airplanes", AFFDL-TR-70-74, Vol I, Dec 1970

nsout

Purpose

Compute Neal-Smith criteria parameters.

Synopsis

[Kp,T1,T2,Clres,phasep,phase_ypyc,phase_ypyc_wtarget,mag_ypyc,mag_ypyc_wtarget,
w_ypyc]=nsout(p,num,den,tau,wtarget);

Description

nsout generates the data required to estimate an aircraft's flying qualities using
the Neal-Smith criteria. The input parameters include **p** which is obtained after
executing the **ns** function (see **ns**). The pitch attitude transfer function must be
entered in descending powers of s and the pilot delay must be added to the plant
delay (i.e. **tau** $= \tau_{pilot} + \tau_{plant}$). The output data are:

Kp,T1,T2; Pilot gain, lead and lag time constants (see **ns**).

Clres; Closed loop resonance of the pilot/vehicle combination in dB.

phasep; Phase compensation produced by the pilot at the target bandwidth.

phase_ypyc, mag_ypyc, w_ypyc; Frequency response data for the open loop
pilot/vehicle system evaluated at 3000 points between .01 and 10 rad/sec. The
data is suitable for plotting on a Nichols plot (i.e. magnitude in dB and phase in
deg).

phase_ypyc_wtarget, mag_ypyc_wtarget; The magnitude and phase at the
target bandwidth **wtarget.**

See Also.

ns;

References

[1] Neal, T. P. and Smith, R.E., "An In-Flight Investigation to Develop Control
System Design Criteria for Fighter Airplanes," AFFDL-TR-70-74, Vol I, Dec
1970

rs

___

Purpose

Determine short term pitch response parameters for the Ralph Smith criteria.

Synopsis

[slope,wc,phasewc,line_bf,w,mag,phase]=rs(num,den,tau);

Description

The Ralph Smith criteria consists of three components: a frequency domain pitch, and normal acceleration response criteria, and a time domain pitch response criteria. The normal acceleration portion of the criteria has not been implemented at this time. The time at which the peak value of the pitch rate response to a step input occurs is defined as t_q. This is only time domain parameter required for the analysis. For aircraft which do not exhibit pitch rate overshoot due to a step input, t_q is taken as the time required to reach 90% of the steady state pitch rate. This parameter is calculated using the **tpr** function for convenience. The **rs** function computes the frequency domain parameters associated with the pitch attitude response. The pitch attitude transfer function must be given in descending powers of s and the time delay in sec. The output data are:

slope; A least squares fit of a line to the pitch attitude magnitude curve is performed. The fit is based on 10 magnitude points (in dB) between 2 and 6 rad/sec using logarithmic frequency as the domain. The slope of the line is returned in dB/octave as specified by the criteria.

wc; An estimate of the pilot/vehicle gain crossover frequency.

phasewc; The phase angle of the pitch attitude transfer function at the estimated pilot/vehicle gain crossover frequency.

line_bf; A vector which contains the parameters of the line of best fit to the magnitude curve. The first element of the vector is the slope in db/dec. The second element is the point at which the line intersects the log(1 rad/sec) axis in dB.

w,mag,phase; The bode data for the transfer function with delay evaluated at 200 frequency points.

See Also

tpr; Control System Toolbox; step, bode

References

[1] Smith, R.H. and Geddes, N.D., "Handling Qualities Requirements for Advanced Aircraft Design: Longitudinal Mode", AFFDL-TR-78-154, Aug , 1979

[2] Strang, Gilbert, "Linear Algebra and its Applications, 3rd ed.", pp 160-162, Harcourt Brace, Jovanovich, SanDiego 1988.

senter

Purpose

Enter a transfer function in shorthand notation

Synopsis

[num,den]=senter([nfn,ofn1,fn1,.....,ofnn,fnn],[nfd,ofd1,fd1,.....,ofdn,fdn])

Description

Parameters:

nfx = number of factors in the numerator or denominator
ofxx = order of the n'th factor
fxx = the n'th factor
= constant (if ofn = 0 (gain) or 1 (pole)
= zeta, omega (if ofn = 2)

Example:

$$G(s) = \frac{2(s+3)(s+2\zeta_n\omega_n + \omega_n^2)}{(s+5)(s+2\zeta_d\omega_d + \omega_d^2)}$$

[num,den]=senter([3,0,2,1,3,2,ζ_n,ω_n],[2,1,5,2,ζ_d,ω_d])

There are 3 factors in the numerator, a gain, zero and second order pair.
There are 2 factors in the denominator, a pole and second order pair.

Senter will handle an arbitrary number of factors in the numerator and denominator. It returns the numerator and denominator in standard Matlab transfer function format (i.e. a vector of polynomial coefficients in descending powers of s).

See Also

Control System Toolbox User's Manual. zp2tf, damp, tf2zp

thist

Purpose

Creates an interactive time history plot with crosshairs.

Synopsis

[t_hist,time_axis]=thist('start',t_hist,t,y);

Description

The output arguments **t_hist** and **time_axis** must be specified. They keep track of the figure window and axis. The function makes explicit reference to these variables when the mouse buttons are used to move the crosshairs. The input arguments <u>must</u> use the names specified in the synopsis because they are referenced when the mouse is used to move the crosshairs. The input arguments are:

t; The time vector.

y; The system response vector.

See Also

bodeplot.

tpr

___

Purpose

Compute the parameters for the time domain transient peak ratio (TPR) criteria.

Synopsis

[t1,t2,tqmax,qss,qmax,deltaq1,deltaq2,pr,tsmall,qsmall,tconts,qmaxslopeline,tming2,minq2,qt1,qt2]=tpr(num,den,tau);

Description

The TPR criteria uses parameters extracted from a pitch rate step response to estimate the pitch axis flying qualities. The transfer function must be entered in descending powers of s and the time delay in sec. The output parameters are:

t1; Equivalent time delay. Measured from the instant the step is applied to the time at the intersection of the maximum slope line with the time axis.

t2; Time measured from the instant of the step input to the time corresponding to the intersection of the maximum slope line with the steady-state pitch rate line.

tqmax; The time at which the maximum pitch rate is reached. (Maximum q). For systems with no overshoot this is taken as the time it takes to reach 90% of the steady-state pitch rate. (Consistent with Ralph Smith, see **rs**) .

qss; The steady-state value pitch rate due to a step input. (Computed using the final value theorm).

qmax; The peak value of pitch rate due to a step input. For systems with no overshoot this is taken as 90% of the steady state pitch rate (see **rs**).

deltaq1; The distance between the maximum value of q and the steady-state q.

deltaq2; The steady-state value of q minus the first minimum.

pr; The transient peak ratio. (i.e. **deltaq1/deltaq2**).

tsmall,qsmall; Two 500 point vectors containing the pitch rate response to a step input.

tconst; The length of the time history.

qmaxslopeline; A 500 point vector which defines a line tangent to the point at which the maximum slope of the pitch rate step response occurs.

minq2,tminq2; The minimum value of the pitch rate response and the time that it occurs respectively.

qt1,qt2; The pitch rate at **t1** and **t2** respectively.

See Also

rs; Control Systems Toolbox; stepfun, lsim, timevec

References

[1] Anon: MILSTD-1797A "Flying Qualities of Piloted Aircraft," Jan 30, 1990 . ASD/ENES, Wright Patterson AFB OH.

[2] Chalk, C.R., " Calspan Recommendations for SCR Flying Qualities Design Criteria," NASA-CR-159236, April 1980.

PART II:

Modified Optimal Control Pilot Model Toolbox

4. MOCM Introduction

The interaction between man and machines has been a subject of interest for over 50 years. Several control theoretic models of a human operator engaged in a compensatory tracking task have been proposed. A compensatory task is one in which the operator is asked to minimize tracking errors given a manipulator with which to control the machine and a display of the error which he/she is trying to minimize. A block diagram of such a situation is shown in Figure 4-1.
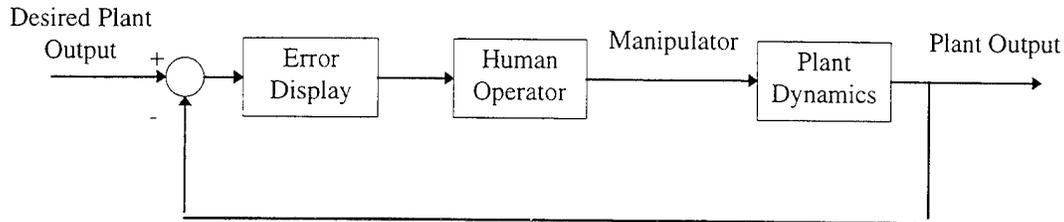
Figure 4-1. Block diagram of single axis compensatory tracking task.

Classical models of human operator behavior use specific rules to determine dynamic elements which comprise the human operator block. These models include those developed in References [11],[12],[14]. Much of this work has been done in the aerospace field, i.e. developing models of human pilot behavior; therefore, this manual will use the term pilot and human operator interchangeably. The classical pilot models commonly take the form of a lead-lag filter with time delay and adjustable gain.

$$Y_p(s) = K_p \frac{T_1 s + 1}{T_2 s + 1} e^{-\tau_p}$$

The classical models do not allow one to conveniently analyze multi-axis tracking tasks such as pitch and roll attitude tracking. This gave rise to the Optimal Control Model developed in Reference [9] which applied LQG optimal control theory to the problem of estimating human pilot behavior. The OCM models the pilot as a state estimator (Kalman filter) with a state feedback control law and accounts for pilot delay and neuromuscular effects. The OCM keeps track of the time delay explicitly and treats it as an observation delay. The output of the OCM is in the form of a frequency response describing function which makes analysis somewhat inconvenient. Recent developments have given rise to a Modified Optimal Control Model, Reference [5]. MOCM is input and output compatible with the OCM but models the time delay with a second order Pade approximation placed at the output of the neuromotor lag filter. A block diagram of the MOCM structure is shown in Figure 4-2. We will consider a single axis tracking task as an example; however, the MOCM does allow multi-axis tasks to be modeled. The pilot controls the plant through a manipulator. The scalar manipulator output is labeled δ. The state vector $\mathbf{x}$ contains the plant and disturbance states. The output vector $\mathbf{y}$ contains the variable that the pilot is trying to regulate as well as its time derivative and other perceived variables. These outputs are assumed to be corrupted by observation noise, v_y, a zero mean Gaussian white noise process with intensity V_y. The state feedback gains

and state estimator generate the scalar u_c which is essentially the pilot's thought command i.e. what he/she would if do the machine responded to thought and not to muscular forces acting on
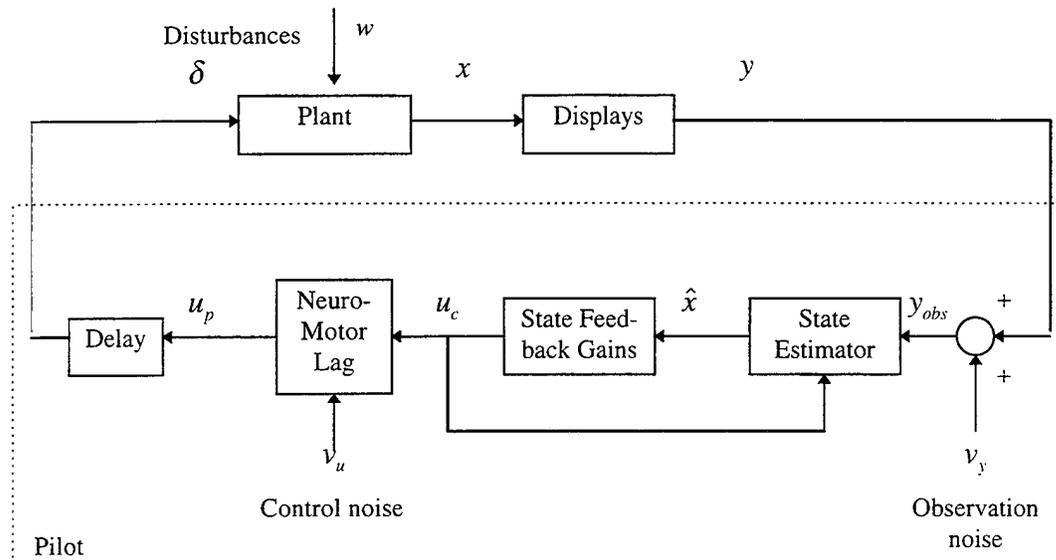
Figure 4-2. Conceptual block diagram of MOCM (Reference [5]).

a manipulator. A pilot's muscles and limbs cannot respond infinitely fast to a thought command. This characteristic is modeled by placing a neuromotor lag in series with the gains and estimator. Mathematically, the neuromotor lag is represented by a first order low pass filter with time constant τ_n. Control or motor noise is added to the pilot's "thought command" to account for uncertainty in the pilot's input. This models the pilot's inability to move the manipulator with infinite precision. The output of the neuromotor lag can be mathematically expressed as:

$$u_p = \frac{1}{\tau_n s + 1}(u_c + v_u)$$

The pilot's performance is also limited by reaction time delay due to the pilot's inability to instantly act on a stimulus. This delay is modeled using the second order Pade approximation.

The basic assumption of the MOCM is the pilot chooses the control input in such a way as to minimize the following quadratic performance index:

$$J = E_\infty \left\{ y^T Q_y y + u_p^2 r + \dot{u}_p^2 f \right\}$$

The output vector y in a single axis compensatory tracking task is comprised of the error and error rate. The relative importance of these variables is defined by selecting the diagonal elements of the Q_y matrix. The contribution of the pilot's manipulator input to the performance index is governed by the weighting r. The importance of control rate is controlled by the weighting f which is automatically selected by the MOCM algorithm when one selects the neuromotor time constant. Selecting the neuromotor time constant

MOCM Introduction

fixes the weighting *f* and drives the pilot and pilot/vehicle bandwidth. The MOCM can also model the effects of divided attention and indifference thresholds which are common behaviors exhibited by pilots. For a more detailed discussion of the theoretical development of the MOCM, the reader is referred to Reference [5].

4.1 Installation

- Create a subdirectory called MOCM under your toolbox directory.
- Copy the files from the floppy disk to the MOCM subdirectory.
- Add the MOCM subdirectory to your Matlab path, on MS Windows systems this is found in your **matlabrc.m** master startup file.
- Run Matlab and start your analysis.
- In order to use the **analyze.m** Graphical User Interface you must be using Matlab® Version 4.2 or greater.

4.2 Variable Definitions

Since MOCM and MOCMMULT are Matlab® scripts, the variables they use are available after an analysis. The **analyze.m** Graphical User Interface is available for single and multi-axis MOCM analysis. It is expected that most if not all required information is available through this interface. Nevertheless there may be times when one wishes to obtain information not directly available from the GUI; therefore, a list of user accessible variables is given below:

Input Variables:

a,b,c,d,e	Augmented disturbance filter and plant dynamics (Single axis)
Edotweight	Weight on the error rate in quadratic performance index (scalar).
Eweight	Weight on the error in quadratic performance index (scalar).
mdisturbance	String vector whose rows consist of 'out' or 'inp' describing where the disturbance is to be added, e.g. ['out' ; 'inp'; 'out'] would describe the structure of a three axis task where the disturbance is added to the output of axis 1, the input of axis 2 and the output of axis 3.
mEdotweight	Weights on the error rates in quadratic performance index (vector, multi-axis).
mEweight	Weights on the errors in quadratic performance index (vector, multi-axis)
(m)thresh	Observation indifference threshold(s) (scalar,vector)
mycden	Multi-axis transfer function matrix of denominator coefficients.
mycnum	Multi-axis transfer function matrix of numerator coefficients.

mywden	Multi-axis transfer function matrix of disturbance filter denominator coefficients.
mywnum	Multi-axis transfer function matrix of disturbance filter numerator coefficients.
naxes	Number of axes to be controlled.
Qy	Weighting matrix for y in performance index.
r,mr	Weight(s) on control u_p in quadratic performance index.
rhoy,mrhoy	Observation noise to signal ratio(s) (scalar,vector)
tau,mtau	Pilot reaction time delay(s) (scalar,vector).
taun,mtaun	Neuromotor time constant(s) (scalar,vector).
Vw, mVw	White noise disturbance intensity (scalar,vector).
ycden	Plant transfer function denominator coefficients in descending powers of s.
ycnum	Plant transfer function numerator coefficients in descending powers of s.
ywden	Disturbance filter denominator coefficients in descending powers of s.
ywnum	Disturbance filter numerator coefficients in descending powers of s.

Intermediate MOCM Results:

a0,b0,c0	Control rate formulation of system matrices.
as,bs,cs,ds,es	State space model of disturbance filter, plant and Pade approximation. (System Matrices)
f	Control rate weight in quadratic performance index. (Tied to taun and bandwidth, computed from an optimization routine).
J	Value of quadratic performance index.
K0	Steady-state solution the algebraic Riccati equation for the control rate formulation (see Davidson and Schmidt 1992 p. 6.)
L	State feedback gain matrix.
q0	State weighting in a quadratic performance index for the control rate formulation.
Udot	Variance of manipulator rate.
Vu	Control noise intensity required to meet the desired noise to signal ratio.
Vy	Observation noise intensity required to meet the desired noise to signal ratio.
X	State covariance matrix of the closed loop pilot vehicle system.
Y	Output and control covariance matrix of the closed loop pilot vehicle system. Y(1,1)=variance of controlled variable, Y(2,2)=variance of time derivative of controlled variable, Y(3,3)= variance of manipulator output.
ypden	denominator of pilot transfer function.

ypnum   numerator of pilot transfer function.

Note: Most variables associated with the state estimator are computed in a function file **mocmkbf.m** therefore not all of these variables are available to the user unless you declare them global. Similarly the pilot matrices are computed in the function file mocmpilot.m and are not available without a global declaration. The function files are well documented and the variable names are consistent with the notation used in Reference [5].

Intermediate MOCMMULT Results:

acoeff,bcoeff,ccoeff   Results of a least squares fit of a parabola to Jopt(ifrac)
                            data $\hat{J}_{opt} = a/f^2 + b/f + c$

attfracopt   Optimal attentional allocation vector e.g. [.2 .3 .5] means
                            20% of the pilots attention is devoted to axis 1, 30% to
                            axis 2 and 50% to axis 3 for a three axis tracking task.

frac   Vector of attentional fractions [0 .1 .2 ..... 1]

ifrac   Reciprocal of elements of frac.

Jopt   Table of optimal costs.

sigmac   Vector of disturbance noise variances.

warn   A string matrix which indicate computed optimal attention
                            fractions are out of range of accurate parabolic fit.

Note that **mocmmult.m** only computes optimal attention fractions for decoupled multi-axis cases. One must run the individual axes through MOCM using the appropriate element of **attfracopt** for each axis. The GUI **analyze.m** does this automatically so it is transparent to the user.

4.3 MOCM and MOCMMULT Program Structure

Flow charts of the Matlab implementation of the MOCM and MOCMMULT scripts are shown in Figure 4-3 and
Figure 4-4 respectively. The tutorial explains in detail how to use these two scripts.

Single Axis MOCM Program Structure
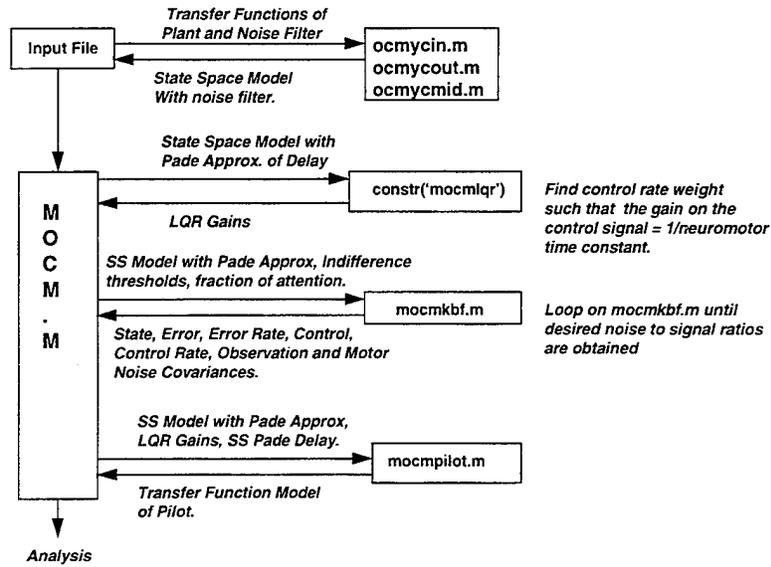
Figure 4-3. Flowchart for single axis MOCM.

Decoupled Multi-Axis MOCM Program Structure
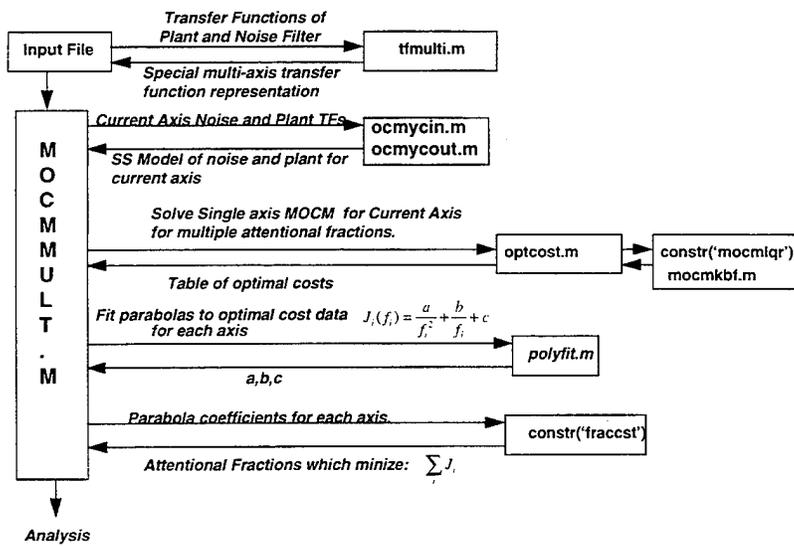
Figure 4-4. Flowchart for multi-axis MOCM.

5. Tutorial

Single Axis Analysis using MOCM

The Modified Optimal Control Model of human pilot behavior is well suited to the analysis of multi-axis as well as single axis compensatory tracking tasks. A wealth of techniques have been developed for describing human operator behavior in a single axis tracking task (References [11],[12],[14]). The MOCM (Reference [5]) is based on the original OCM developed by Kleinman, Baron and Levison (Reference [9]). MOCM has been shown to produce results which are very similar to the original OCM, but has the distinct advantage of returning a pilot-model in transfer function form. The original OCM produced a pilot describing function since it kept explicit track of the pilot's delay. The MOCM replaces the pure delay term by a second order Pade approximation. This has been shown to be sufficiently accurate for typical values of time delay over the frequency range at which man/machine systems typically operate. This section discusses how to use the MOCM toolbox to analyze a single axis man/machine system.

5.1 Simple Integrator Controlled Element

First, one must develop a model of the system to be controlled. Here we will consider the problem of driving the output of a k/s plant to zero in the presence of an input disturbance. This is analogous to a pilot using an attitude rate command system to maintain a desired attitude (zero attitude in this example) in the presence of an attitude rate disturbance or forcing function. This example is designed to model some experimental data analyzed in Reference [9] where the operator was instructed to keep the output of a k/s plant at zero in the presence of an input disturbance consisting of a sum of sine waves. The sum of sines forcing function has the most frequency content below 2 rad/sec with mean squared value of 2.2. The bandwidth of 2 rad/sec and variance of $\sigma_v^2 = 2.2$ match the important properties of the experiment. The MOCM is an LQG formulation of the pilot in the loop tracking problem; therefore we must introduce an approximation of the sum of sines forcing function using filtered white noise. A block diagram representation of the forcing function is shown in Figure 5-1.

$$
\begin{array}{ccc}
w(t) & Y_w(s) = \dfrac{1}{s+2} & v(t) \\
V_w = 8.8 & & \sigma_v^2 = 2.2
\end{array}
$$
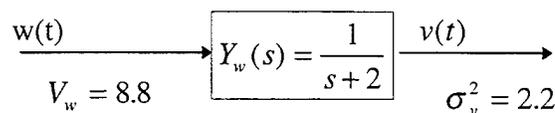
Figure 5-1. Forcing function approximated by filtered white noise.

Where w(t) is a zero mean Gaussian white noise process with intensity V_w and v(t) is the colored noise which has the same RMS and bandwidth as the original forcing function. In general, the calculation of the required intensity V_w to yield a desired input disturbance σ_v^2 can be determined by finding the expression for the power spectral density and

Tutorial

evaluating residues. A simpler method of selecting V_w for low pass filters is outlined below.   Since most experiments use a sum of sines to approximate a rectangular input spectrum, we will show how to select the noise intensity V_w to match a sum of sines RMS.

1. Calculate the sum of sines RMS

$$f(t) = A_0 + \sum_{k=1}^{n} A_k \cos(\omega_k t + \phi_k)$$

$$\sigma_f^2 = A_0^2 + \sum_{k=1}^{n} \frac{A_k^2}{2}$$

Sum of sines RMS = $\sqrt{\sigma_f^2} = \sigma_f$

2. Select filter order and break frequency based on the power distribution of the sum of sines and put it into the following form:

$$Y_w(s) = \frac{1}{a_0 s^n + a_1 s^{n-1} + \dots a_n}$$

3. Calculate the required value V_w using the Table 5-1.

Table 5-1. Require white noise intensity to acheive desired RMS filter output.

Filter Order	Required Value of V_w
1	$2 a_0 a_1 \sigma_f^2$
2	$2 a_1 a_2 \sigma_f^2$
3	$\dfrac{2 a_0 a_3 (a_1 a_2 - a_0 a_3) \sigma_f^2}{a_0 a_1}$
4	$\dfrac{2 a_0 a_4 (a_1 a_2 a_3 - a_0 a_3^2 - a_1^2 a_4) \sigma_f^2}{a_0 (a_1 a_2 - a_0 a_3)}$

Alternately Butterworth filters with prespecified break frequencies or equivalent rectangular bandwidths and RMS responses to unit intensity white noise inputs can be generated using the **noisefilt** and **effbw** functions respectively. The next step is to generate a state space model of the tracking task which is consistent with the MOCM framework. A set of Matlab script files are included in this package to make this a simple task. In this case the pilot is attempting to zero the attitude in the presence of an attitude rate disturbance for a k/s plant. A block diagram of this situation is shown in Figure 5-2. Note that the error is equal to the difference between the actual and desired vehicle attitude (i.e. $e(t) = y_a - (y_d = 0)$ or $e(t) = y_a$).

Tutorial
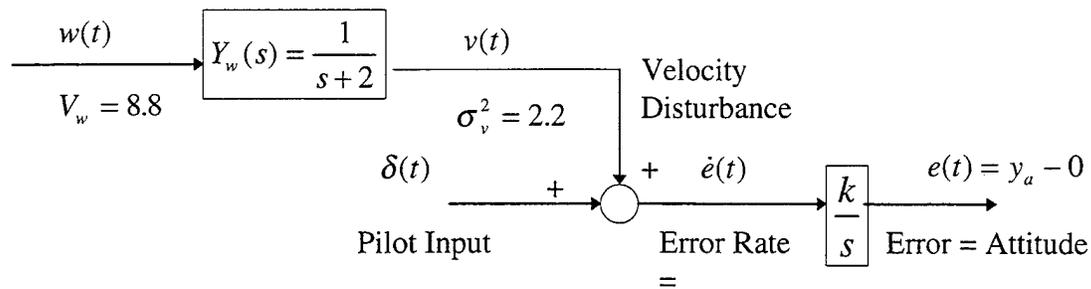
Figure 5-2. Block Diagram of disturbance rejection task.

The Matlab® script **ocmycin.m** can be used to generate the state space representation of this situation which is required by the MOCM main module. The script **ocmycin.m** can be used with any system where input to the plant is the sum of the disturbance and the pilot input. Similar scripts called **ocmycout.m** and **ocmycmid.m** allow you create models of systems where the disturbances are added at the output of the plant and at some intermediate state of the plant respectively. To form the state space model of the plant for use with MOCM, invoke the following command:

[a,b,c,d,e]=ocmycin(1,[1 0],1,[1 2]);

For more information on **ocmycin.m** see the MOCM reference section. Next a few more input parameters must be defined. First, the pilot time delay is set to τ_p=0.15 sec. This parameter arises from the pilot's inability to react instantaneously to a given observation. MOCM also accounts for the pilot's inability to process observations over an infinite frequency range and the inability to move the stick with infinite precision. These characteristic are modeled by low pass filtering the pilot's thought command u_c and motor noise v_u.

$$u(s) = \frac{1}{\tau_N s + 1}(u_c(s) + v_u(s))$$

where τ_N is called the neuromotor time constant. Generally, the neuromotor time constant has been found to vary over the range $0.1 \le \tau_N \le 0.6$ sec. A neuromotor time constant of 0.1 sec is generally regarded as the upper limit of human ability and is a good starting point for many analyses. In this case a τ_N of 0.08 is used to be consistent with the example of Reference [9].

Now we must define the observation and motor noise to signal ratios. Observation noise arises from the pilot's inability to perfectly observe the vehicle output and output rate. For compensatory tracking tasks it has been found that a pilot can obtain first order derivative (rate) information from the displayed variable; therefore, observation noise to signal ratios for both the displayed variable and the time derivative of the displayed variable must be defined. Researchers have found that over a wide range of foveal viewing conditions the observation noise to signal ratios ρ_y are 1:100 or -20 dB. Other

variables that can influence these ratios include display dynamics, size or delays. The motor noise to signal ratio ρ_u is typically set to 0.00316 or -25 dB, a value which has produced good results when compared to experimental data. (Note that these noise ratios are expressed in power dB.) Another parameter which must be set is the attentional fraction f. This parameter describes how much of the pilot's attention is focused on the task. For example if the pilot is devoting total concentration to the task at hand, $f = 1$. If the pilot spends an average of half his time concentrating on the display and the other half on another operation (i.e. looking at another instrument, looking for other aircraft, etc.), then $f = 0.5$. For single axis tracking tasks like this example, we typically pick $f=1$.

Normally, pilots do not react to <u>every</u> change in the variable they are trying to control. This is called an indifference threshold (i.e. how much deviation will I tolerate until I move the stick to compensate). For compensatory displays the controlled variable is the difference between the desired and actual plant output. Therefore, indifference thresholds must be placed on the error and error rate. For this example the thresholds are set to zero which (unrealistically) corresponds to the situation where the pilot reacts to every change in $e(t)$ and $\dot{e}(t)$ no matter how small.

The entire premise of MOCM is that the pilot will behave in such a manner as to minimize a weighted sum of squares of vehicle states and control (stick) deflections and rates. For this case the quadratic cost function takes the form:

$$J = \int_0^\infty [e \quad \dot{e}]Q_y[e \quad \dot{e}]^T + r\,u^2 + f\,\dot{u}^2 \; dt$$

You must pick values for Q_y and r but not f. The diagonal elements of Q_y represent how much importance the pilot places on the error and error rate. The value of r represents the importance of control deflection to the pilot. MOCM finds the value of the control rate weight g which yields the specified neuromotor time constant. A full explanation of this relationship can be found in References [5] and [9]. For this case diag(Q_y) = [1 0] and $r=0$.

The pilot model can now be computed using **mocm.m**. Below is a listing of an input file which integrates all of the above discussion into a Matlab® script file called **mocmex1.m**. It is included as part of the MOCM toolbox

```
clc
%Velocity Command System with
%Velocity disturbance;

%Define Number of Axes
naxes=1;

%Define the controlled element transfer function.
ycnum=1;ycden=[1 0];
```

```
%Define the disturbance filter.
ywnum=[1];ywden=[1 2];

%Use ocmycin to inject the disturbance at the
%input to create a velocity disturbance.
help ocmycin
[a,b,c,d,e]=ocmycin(ycnum,ycden,ywnum,ywden);


Vw=8.8 ;          %Variance of Disturbance Noise
tau=.15;          %Reaction time delay
taun=.08;         %set the neuromuscular time constant
                  %typically .1 sec but ranges .1 <=taun<=.6
                  %taun drives the bandwidth of the
                  %pilot/vehicle system.
rhoy=[.01 .01];   %Define observation noise variance to signal
                  %variance ratio.
frac=1 ;          %Fraction of attention devoted to the error
                  %and error rate respectively.
thresh=[ 0  0];   %Indifference thresholds on the error and
                  %error rate respectively i.e. Define the
                  %dead zone over which a pilot will not
                  %respond to an error. Value must be in error
                  %or error rate units. Note that 0 means the
                  %pilot will respond to any error or error
                  %rate no matter how small.
rhoua=.00316;     %Define observation noise variance to signal
                  %variance ratio.
Eweight=1;        %Error Weight in the Performance Index.
Edotweight=0;     %Error rate weight in the performance index.
r=0;              %Control Weight in performance index
%
%           |^inf
%    J = |          [e edot] Qy [e edot]' + r u^2  + f udot^2
%           v 0
%
Qy = diag([Eweight Edotweight]);
%Call main module mocm
mocm
```

After running this script, one can analyze the results using a graphical user interface by invoking the analyze.m script (Note: Requires Matlab Version 4.2 or greater). This package gives the user access to the most frequently used information that can be obtained from MOCM results. The following features are currently available:

Tutorial

a) Full order MOCM frequency response.
b) Open loop pilot/vehicle frequency response.
c) Full order MOCM gain, poles and zeros.
d) Reduced order MOCM by pole-zero cancellations.
e) Crossover Equivalent Realization of MOCM.
    I) Preserves gain, phase, slope of gain curve at open loop pilot vehicle crossover
      frequency.
    II) Produces a standard form pilot model, compatible with the classical models of
      References [11],[12] and [14].
f) MOCM reduction by Balanced Realization.
g) MOCM Statistics

We will now show the results for the case study of the 1/s plant with an input disturbance.

a) Full order MOCM frequency response (Figure 5-3).
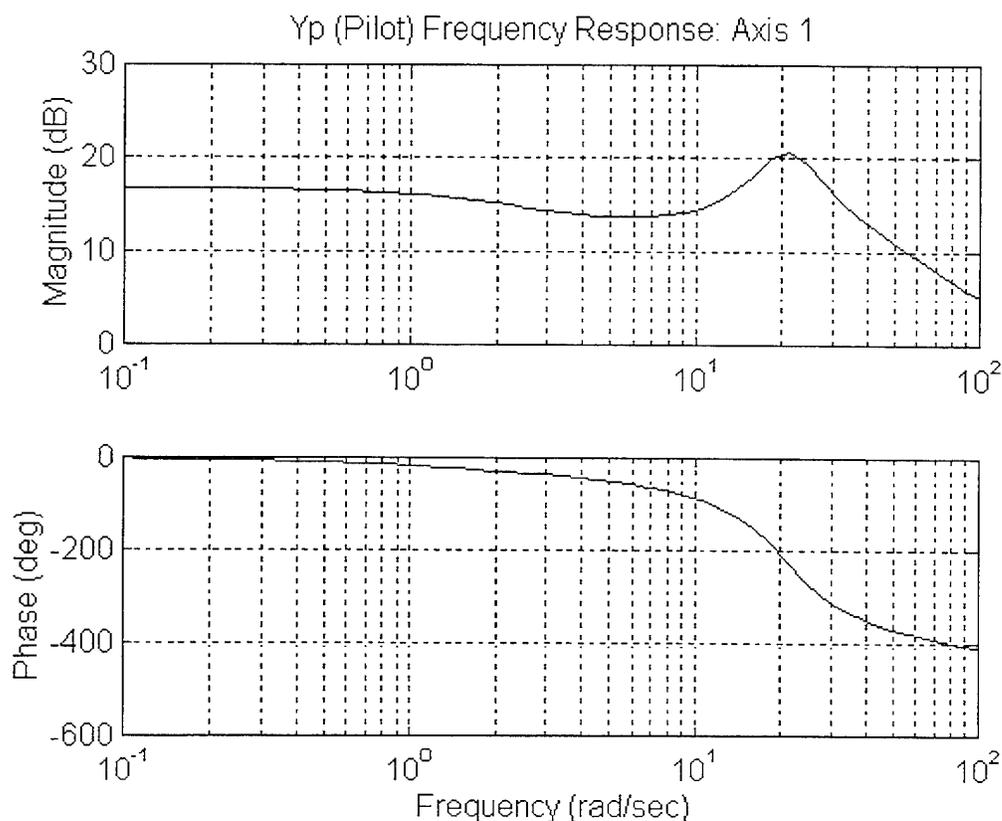
Figure 5-3. Bode plot of MOCM pilot transfer function.

Tutorial

b) Open loop pilot/vehicle frequency response (Figure 5-4).
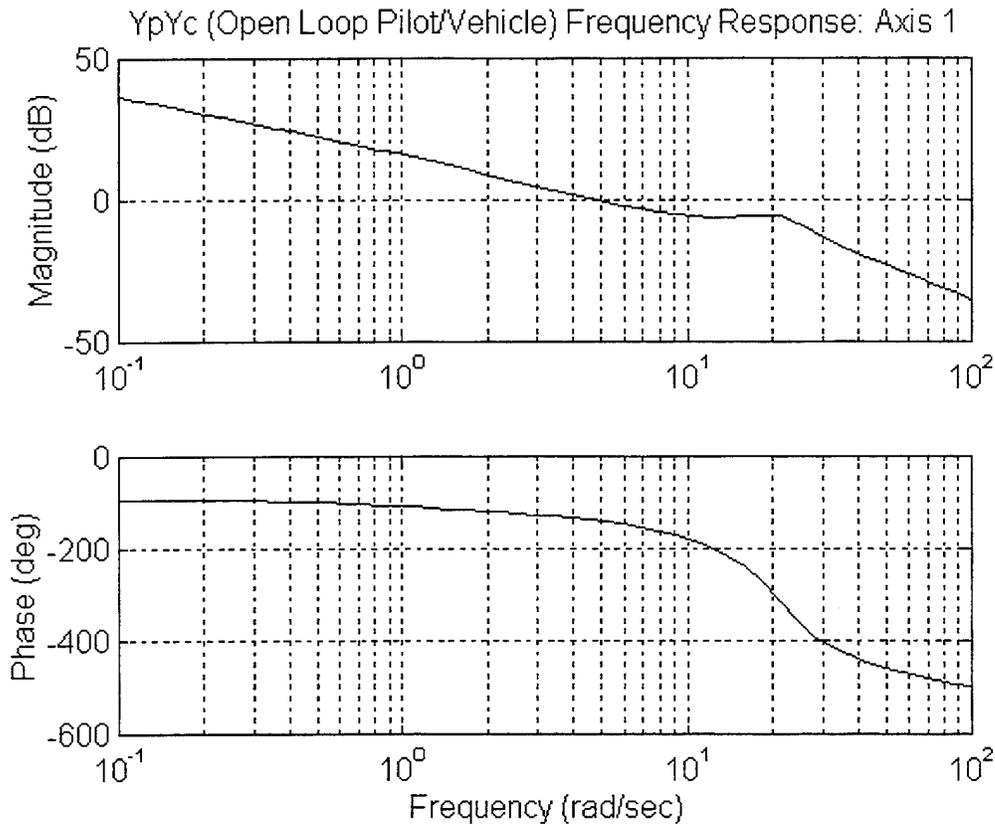
Figure 5-4. Bode plot of open loop pilot/vehicle transfer function.

c)Full order MOCM gain, poles, zeros.

```
Full Order Pilot Model Parameters
Pilot Gain:

kp =

   181.4674

Pilot Zeros:
   Eigenvalue          Damping          Freq. (rad/sec)
   13.3333 +13.3333i   -0.7071             18.8562
   13.3333 -13.3333i   -0.7071             18.8562
   -3.2580              1.0000              3.2580
   -6.3734              1.0000              6.3734
  -12.7510              1.0000             12.7510
  -13.3333 +13.3333i    0.7071             18.8562
  -13.3333 -13.3333i    0.7071             18.8562
```

Tutorial

```
Pilot Poles:
   Eigenvalue            Damping           Freq. (rad/sec)
   -1.9912               1.0000               1.9912
   -5.5373  +20.2544i    0.2637              20.9977
   -5.5373  -20.2544i    0.2637              20.9977
   -6.4478               1.0000               6.4478
  -12.5000               1.0000              12.5000
  -13.3333  +13.3333i    0.7071              18.8562
  -13.3333  -13.3333i    0.7071              18.8562
  -35.3484               1.0000              35.3484
```

d) Reduced order MOCM by pole-zero cancellations.

```
2 pole-zeros cancelled
Pilot Gain

kp =

   181.4674

Reduced Order Zeros
   Eigenvalue            Damping           Freq. (rad/sec)
   13.3333  +13.3333i    -0.7071             18.8562
   13.3333  -13.3333i    -0.7071             18.8562
  -12.7510               1.0000              12.7510
   -6.3734               1.0000               6.3734
   -3.2580               1.0000               3.2580

Reduced Order Poles
   Eigenvalue            Damping           Freq. (rad/sec)
  -35.3484               1.0000              35.3484
   -5.5373  +20.2544i    0.2637              20.9977
   -5.5373  -20.2544i    0.2637              20.9977
  -12.5000               1.0000              12.5000
   -6.4478               1.0000               6.4478
   -1.9912               1.0000               1.9912
```

e) Crossover Equivalent Realization (Figure 5-5).

```
The Crossover Equivalent Realization is:

               (0.189 s + 1)          -0.156 s
Yp(s) = 5.263  -------------       e
               (0.223 s + 1)


   Model Information for Axis 1
MOCM Pilot/Vehicle Gain Crossover freq wx = 4.864(rad/sec)
CER Gain Curve Slope at Crossover        = -1.646  (dB/dec)
CER Phase Angle Contribution (No Delay)   = -4.721 (deg)
```
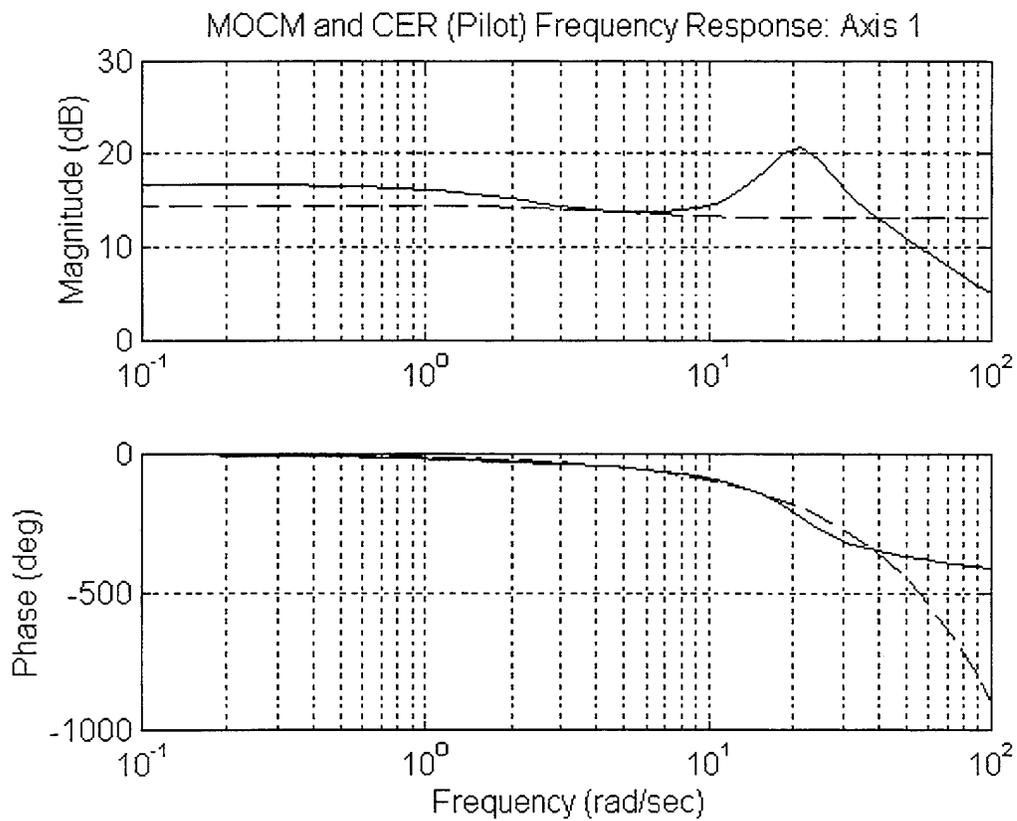

Figure 5-5. Bode plot of full order MOCM and Crossover Equivalent Realization.

Tutorial

f) MOCM Reduction by Balanced Realization (Figure 5-6).
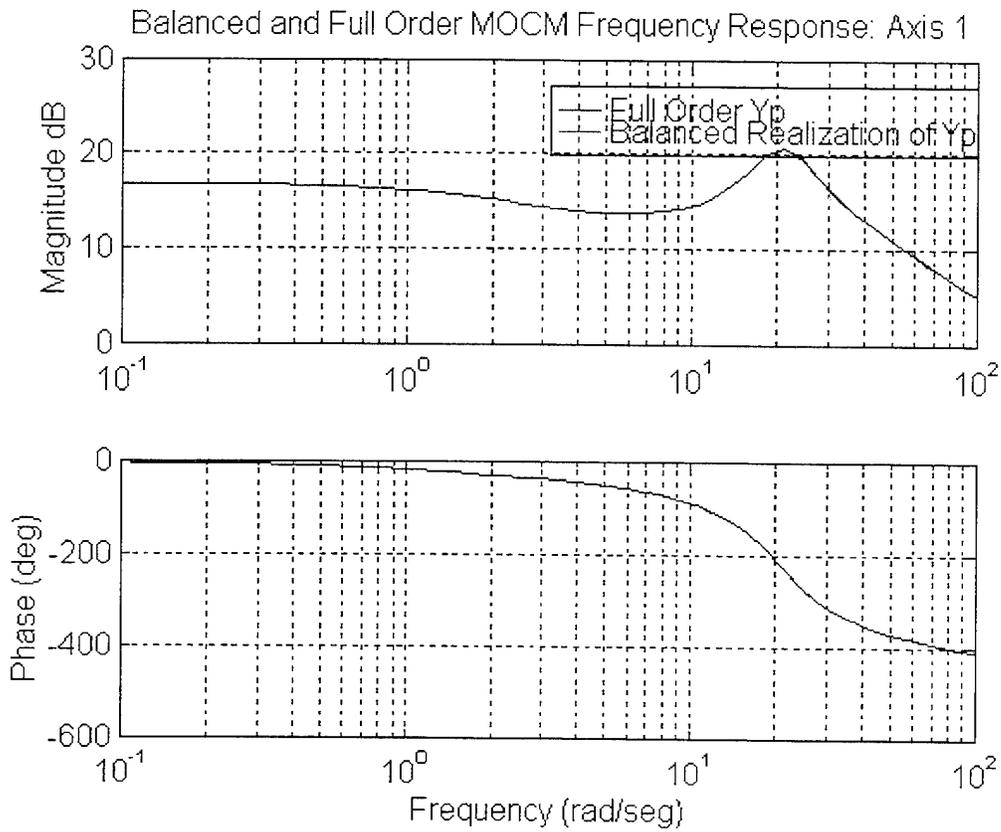
Figure 5-6. Bode plot of full order MOCM and reduced order MOCM by balancing.

```
Balance Realization: Axis 1
Pilot Gain

kr =

   0.0083



Reduced Order Zeros
   Eigenvalue          Damping          Freq. (rad/sec)
   1.0e+004 *

   -2.1840             0.0001              2.1840
   -0.0003             0.0001              0.0003
    0.0013 + 0.0013i  -0.0001              0.0019
    0.0013 - 0.0013i  -0.0001              0.0019
```

```
Reduced Order Poles
     Eigenvalue         Damping        Freq. (rad/sec)
     -5.5540 +20.2546i   0.2644           21.0022
     -5.5540 -20.2546i   0.2644           21.0022
    -34.7427             1.0000           34.7427
     -1.9900             1.0000            1.9900
```

g) MOCM Statistics.

```
MOCM Statistics
      0.1592  = J (Cost Function)
           1  = Error Weight in Cost Function
           0  = Error Rate Weight in Cost Function
   0.0001638  = Control Rate Weight in Cost Function
      0.344   = RMS Error
      1.755   = RMS Error Rate
      1.966   = RMS Control Deflection
     15.79    = RMS Control Rate
    0.06095   = RMS Error Observation Noise
      0.3109  = RMS Error Rate Observation Noise
      0.2196  = RMS Motor Noise
       -20    = Error Noise to Signal Ratio dB
       -20    = Error Rate Noise to Signal Ratio dB
       -25    = Motor Noise to Control Signal Ratio
       -0.95  =    Estimated CHR
```

5.2 Simple Tracker Controlled Element

We will now use another example taken from Reference [9]. This case is representative of a typical flying qualities experiment where the pilot is asked to make vehicle output match a time varying forcing function. For attitude tracking the pilot is instructed to minimize error, or difference between the actual and desired attitude. Forcing functions in compensatory tracking experiments are commonly composed of a sum of sine waves. Using sine waves allows the analyst to easily compute Fourier transforms to generate frequency response data. For MOCM analysis, the forcing function is approximated by filtered white noise which has the same bandwidth and variance as the sum of sines forcing function. In this case a simple tracker (attitude command system) is approximated by a low pass filter with a gain of 40 and -3 dB cutoff frequency of 40 rad/sec. The forcing function is approximated by zero mean, Gaussian white noise with intensity $V_w=10$ passed through a second order low pass filter with a double pole at -2.

This yields a disturbance with a bandwidth of 2 rad/sec and variance $\sigma_v^2 = 0.3125$ which is added to the output of the tracker. A block diagram of this situation is shown in **Figure 5-7**. Note that the disturbance or forcing function is added at the output of the plant; therefore, the Matlab script **ocmycout.m** should be used to form the MOCM state space model of the plant with error and error rate outputs (pilot observations).
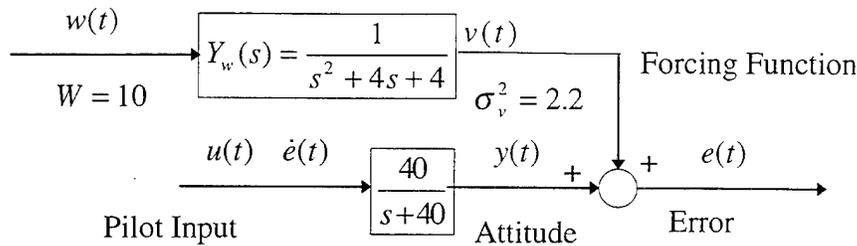
Figure 5-7. Block diagram of compensatory tracking task with position command controlled element.

The following listing of **mocmex3.m** shows how to set up the remaining parameters for this system. It should be noted that all single axis systems must define and use the same variable names as those listed in the two example input files of this section.

```
clc
%Position Command System with
% Position disturbance;


%Define Number of Axes
naxes=1;


%Define the controlled element transfer functions.
ycnum=40;ycden=[1 40];


%Define the Disturbance filter
ywnum=[1];ywden=[1 4 4];


%Use omcycout to inject the disturbance at the
%controlled element output. i.e. position is the
%output of Yc so we must add the disturbance to
%the output.


help ocmycout
[a,b,c,d,e]=ocmycout(ycnum,ycden,ywnum,ywden);


Vw=10   ;           %Variance of Disturbance Noise
tau=.15;            %Reaction time delay
taun=.11;           %set the neuromuscular time constant
                    %typically .1 sec but ranges
                    %.1 <= taun <= .6taun drives the bandwidth
                    %of the pilot/vehicle system.
rhoy=[.01 .01];     %Define full attention observation noise
                    %variance to signal variance ratio.
frac=1;             %Fraction of attention devoted to the error
                    %and error rate respectively
```

```
thresh=[ 0  0];   %Indifference thresholds on the error and
                  %error rate respectively;
                  % i.e. Define the dead zone over which a
                  %pilot will not respond to an
                  % error. Value must be in error or error
                  %rate units. Note that 0 means the pilot
                  % will respond to any error or error rate
                  %no matter how small.
rhoua=.00316;     %Define motor noise variance to input
                  %signal variance ratio.


Eweight=1;        %Error Weight in the Performance Index.
Edotweight=0;     %Error rate weight in the performance
                  % index.
r=0;              %Control Weight in performance index



%
%           |^inf
%    J =    |          [e edot] Qy [e edot]' + r u^2  + g udot^2
%           v 0
%


Qy = diag([Eweight Edotweight]);

%Call main module mocm.m

mocm
```

The resulting pilot model can be analyzed using the script analyze.m.

5.3 Multi Axis Analysis Using MOCMMULT

The ability to model the human operator in multi-axis tracking tasks is a major advantage that the MOCM/OCM has over conventional models which only handle one axis at a time. The MOCM implementation developed for this toolbox allows one to analyze multiple decoupled axes. A more general implementation developed by Kleinman et al. was called PIREP (AFFDL-TR-76-124), a FORTRAN implementation of the original OCM which could handle multiple axes but only gave a frequency response as a pilot describing function, not a transfer function. Nevertheless the majority of individual tasks that a pilot performs are effectively decoupled (i.e. lateral directional, longitudinal dynamics). The utility of this implementation is not expected to be significantly hampered by its inability to handle coupled axes.

Tutorial

The features of the MOCMMULT package will be illustrated by means of an example based on one of Dander's experiments (Reference[4]). This case was in Reference [19]; therefore, this example was chosen as a benchmark case.

This example is based on a three axis Dander experiment. The pilot was assigned the task of making the output of three uncoupled plants follow three independent sum of sines forcing functions. The pilot was instructed to minimize tracking error in <u>all</u> axes i.e. no axis was to given preference. The three controlled elements are given below:

$$Y_{c_1} = \frac{4(s+0.04)(s+0.9)}{s(s+5)(s^2+2\ (0.7)(.25)s\ +\ (.25)^2)}$$

$$Y_{c_2} = \frac{0.5(s+0.1)}{(s+1.5)(s^2+\ 2\ (-0.84)\ (0.5)\ s+(0.5)^2)}$$

$$Y_{c_3} = \frac{10(s+0.1)}{(s+3)(s^2+2\ (0.5)\ (0.5)\ +(0.5)^2)}$$

We will construct an input file for this multiaxis case as we go along. Now that the controlled elements have been defined, we must put them into a structure usable by **mocmmult.m**. First define the number of axes and the individual controlled elements. Note that since **mocmmult.m** is a script and not a function, the variable names given in this example must be used when you develop your own input files with the exceptions of the definitions of the individual controlled elements and noise filters (if the variable name starts with an "n" or an "m" it must be used when using **mocmmult.m**).

```
naxes=3;   %Number of axes to be controlled.

%Define the controlled element transfer functions.
[ycnum1,ycden1]=senter([3,0,4,1,.04,1,.9],[3,1,0,2,.7,.25,1,
5]);
[ycnum2,ycden2]=senter([2,0,.5,1,.1],[2,1,1.5,2,-.84,.5]);
[ycnum3,ycden3]=senter([2,0,10,1,.1],[2,1,3,2,.5,.5]);
```

Next you must use tfmulti.m to form a matrix of the controlled elements which can be used by **mocmmult.m**. The script **tfmulti.m** will currently handle up to five axes but can easily be extended to handle more if necessary.

```
[mycnum,mycden]=tfmulti(ycnum1,ycden1,ycnum2,ycden2,ycnum3,y
cden3);
```

Remember that you must store the multi-axis transfer function in the variable names given above (the variables "m"ycnum and "m"ycden)

5-14

Tutorial

The characteristics of the forcing functions used in the Dander experiment are given in Table 5-2.

Table 5-2. Composition of Dander's forcing functions.

Axis	Frequency (rad/sec)	Unscaled Amplitude
1 (Amplitude = 0.75 in.)	0.367	4.80
	0.483	5.50
	0.816	1.44
	1.32	1.64
2 (Amplitude = 45 deg)	0.367	4.80
	0.483	2.20
	0.816	1.15
	1.32	1.64
3 (Amplitude=1.8 units)	0.483	0.92
	0.816	0.33
	1.32	0.33

The white noise filters used to simulate these forcing functions are given below:

$$Y_{w_1} = \frac{0.2219}{s^2 + 2\,(0.7)\,(0.5)\,s + (0.5)^2}$$

$$Y_{w_2} = \frac{13.3}{s^2 + 2\,(0.7)\,(0.5)\,s + (0.5)^2}$$

$$Y_{w_3} = \frac{0.53}{s^2 + 2\,(0.7)\,(0.5)\,s + (0.5)^2}$$

Note that the break frequency was selected to reflect that most of the signal content was present at frequencies less than 0.5 rad/sec. The numerator gains were selected such that the amplitudes listed in the above table were equal to twice the RMS of the filter outputs; for example, $2\sigma_2 = 45$ deg ,(Refererence [10]). This was apparently done because RMS of the true sum of sines was never calculated in Dander's experiment and cannot be calculated exactly since he only lists unscaled amplitudes and no information about the scale factor is given. The $2\sigma = \max(Amplitude)$ is just a way of making the RMS value of the filtered white noise equal the RMS value of the sum of sines. In general, there is a much better way to select the gain of the noise filter when the noise intensity is set to unity i.e. $V_w = 1$. This will be explained below. Since most experiments use a sum of sines to approximate a rectangular input spectrum, we will show how to select the filter gain to match a sum of sines RMS.

Tutorial

1. Calculate the sum of sines RMS

$$f(t) = A_0 + \sum_{k=1}^{n} A_k \cos(\omega_k t + \phi_k)$$

$$\sigma_f^2 = A_0^2 + \sum_{k=1}^{n} \frac{A_k^2}{2}$$

Sum of sines RMS = $\sqrt{\sigma_f^2} = \sigma_f$

2. Select filter order and break frequency based on the power distribution of the sum of sines and put it into the following form:

$$Y_w(s) = \frac{K}{a_0 s^n + a_1 s^{n-1} + \ldots a_n}$$ where K is the filter gain to be determined.

3. Calculate the required value K using the Table 5-3.

Table 5-3. Filter gain required to achieve desired RMS output for unit intensity white noise input.

Filter Order	Required Value of K
1	$\sqrt{2 a_0 a_1 \sigma_f^2}$
2	$\sqrt{2 a_1 a_2 \sigma_f^2}$
3	$\sqrt{\dfrac{2 a_0 a_3 (a_1 a_2 - a_0 a_3) \sigma_f^2}{a_0 a_1}}$
4	$\sqrt{\dfrac{2 a_0 a_4 (a_1 a_2 a_3 - a_0 a_3^2 - a_1^2 a_4) \sigma_f^2}{a_0 (a_1 a_2 - a_0 a_3)}}$

4. Set the white noise intensity to unity $V_w = 1$.

Alternately, Butterworth filters with prespecified break frequencies or equivalent rectangular bandwidths and RMS responses to unit intensity white noise inputs can be generated using the **noisefilt** and **effbw** functions respectively.

We can now define the forcing functions in our input file.

```
%Define the Noise Filter
[ywnum1,ywden1]=senter([1,0,.2219],[1,2,.7,.5]);
[ywnum2,ywden2]=senter([1,0,13.3],[1,2,.7,.5]);
[ywnum3,ywden3]=senter([1,0,.53],[1,2,.7,.5]);
[mywnum,mywden]=tfmulti(ywnum1,ywden1,ywnum2,ywden2,ywnum3,y
wden3);
```

Tutorial

Note that **tfmulti.m** was used to form a matrix of white noise filters which must be defined for the **mocmmult.m** script.

Recall that the pilot was instructed to make the output of each plant track a forcing function. This means that one will have to use the **ocmycout.m** script since the disturbance or forcing function is added to the vehicle output. The **mocmmult.m** script allows for input and output disturbances but not for intermediate state disturbances. Unlike the single axis case **ocmycout.m** will be automatically called, you just have to tell **mocmmult.m** where the disturbances are to be added i.e. input or output. The variable **mdisturbance** is a Matlab string vector whose rows hold the string 'out' or 'inp'. For this case:

```
%Define where to add the disturbances. ['out';'inp']
mdisturbance=['out'; 'out'; 'out'];
```

A block diagram of the multi-axis system is shown in Figure 5-8.
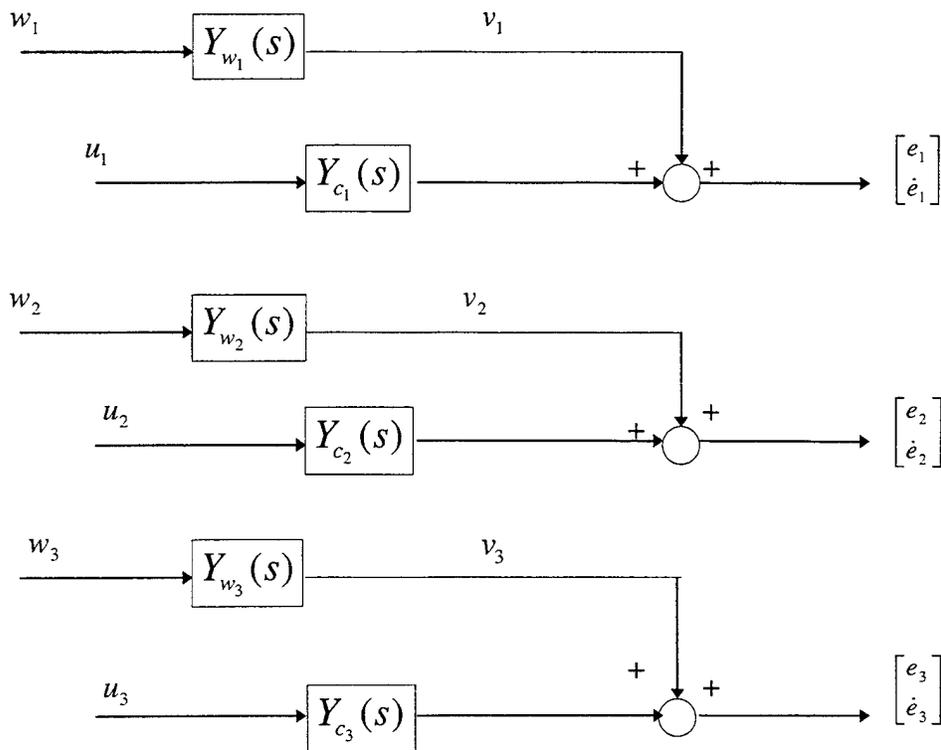
Figure 5-8. Block diagram of Dander's experiment.

Next define the white noise intensity that drives the filters. The noise filter gain was calculated to achieve the desired forcing function RMS in the presence of a Gaussian white noise input with unit intensity; therefore, we must set the elements of the noise intensity vector mVw equal to 1.

```
mVw=[1 1 1];            %Variance of Disturbance Noise
```

Next we must define the pilot observation/reaction delay and neuromotor time constants. Pilot delay is typically in the range of 0.1 to 0.2 sec and the neuromotor time constant that is generally believed to be the upper limit of human ability is 0.1 sec. In this example we use the following:

```
mtau=[.2 .2 .2];        %Reaction time delay
mtaun=[.1 .1 .1];       %set the neuromuscular time constant
                        % typically .1 sec but ranges
                        % between .1 <= taun <= .6
                        %taun drives the bandwidth of the
                        %pilot/vehicle
```

Next we must define the full attention observation noise to signal ratios. Researchers have found that over a wide range of foveal viewing conditions the observation noise to signal ratios ρ_y are 1:100 or -20 dB. (Note that these noise ratios are expressed in power dB.) Recall that the observations in this case are error and error rate in each axis, therefore we must define the observation noise to signal ratios for each observation.
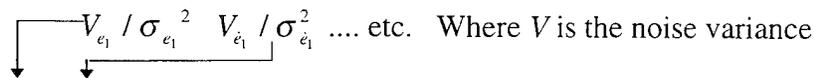
$$V_{e_1}/\sigma_{e_1}^2 \quad V_{\dot{e}_1}/\sigma_{\dot{e}_1}^2 \;\;\; \text{.... etc. Where } V \text{ is the noise variance}$$

```
mrhoy=[.01 .01 .01 .01 .01 .01];   %Define full attention
                                   %observation noise
                                   %variance to signal
                                   %variance ratios.
```

The motor noise to signal ratio ρ_u is set to 0.01 or -20 dB for this analysis but is typically set to -25 dB. Since we have three pilot inputs we must define a three element vector of motor noise to signal ratios.

```
mrhoua=[.01 .01 .01];              %Define motor noise variance to
                                   %input signal variance ratio.
```

In the single axis case one must define the fraction of attention for the axis of interest. **mocmmult.m** automatically calculates the optimal attention fractions to minimize the total normalized cost.

$$J_{tot} = \sum_{i=1}^{3} \frac{J_i}{\sigma_{v_i}}$$

The determination of the optimal attention fractions for multi-axis cases is computationally intensive so be patient. This implementation forms a table of normalized costs over a selected range of attentional fractions (f = {0.1, 0.2, ...,0.9, 1.0}) for each

Tutorial

axis. In other words the individual axes must be solved as a single axis problem 10 times. Table 5-4 shows the contents of the variable Jopt for this case.

Table 5-4. Normalized cost as a function of attentional fraction

f Axis	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.2151	0.1183	0.0879	0.0730	0.0641	0.0581	0.0538	0.0506	0.0480	0.0459
2	Inf	Inf	135.776	4.3312	1.8198	1.1982	0.8985	0.7258	0.6143	0.5370
3	0.2169	0.1408	0.1117	0.0959	0.0860	0.0791	0.0739	0.0700	0.0668	0.0642

Note that axis 2 has infinite cost at low fractions of attention. This simply means that the closed loop pilot vehicle system is unstable for that fraction of attention. Next we fit parabolas to the optimal cost data for each axis.

$$\hat{J}_i = \frac{a}{f_i^2} + \frac{b}{f_i} + c$$

If there is more than 10% difference between the cost from the table and the estimated cost then the fit is redone using a subset of table data. Most problems with the fits occur for low values of attentional fractions so if we have a fit problem, we kick out the cost associated with the lowest attention fraction and perform the fit on the remaining set of data. This is process is continued until all estimates are within 10% of the values in the cost table. This determines the range of validity for the subsequent optimization. The following optimization problem is solved using **constr.m,** the constrained optimization routine from the Matlab Optimization Toolbox.

$$\min_{f_i} \hat{J}_{TOT} = \sum_{j=1}^{3} \frac{a_j}{f_i^2} + \frac{b_j}{f_i} + c_j$$

Subject to:

$$\sum_{i=1}^{3} f_i = 1$$

It is possible that some attentional fractions will fall outside of the validity range. A common cause of this is the case where one axis requires a large amount of attention and the other require very little i.e. f < 0.1. Under these conditions it is recommended that you compare the estimated cost from the parabolic fit to the cost generated by a single axis MOCM analysis to determine if the estimate is valid.

Normally pilots do not react to every change in the variable they are trying to control. This is called an indifference threshold (i.e. how much deviation will I tolerate until I move the stick to compensate). For compensatory displays the controlled variable is the difference between the desired and actual plant output. Therefore indifference thresholds must be placed on the error and error rate. Reference [10] used the indifference thresholds given in Table 5-5 for the analyis of the Dander experiments based on visual

Tutorial

perception thresholds of 0.05 deg and 0.1 deg/s of visual arc and arc rate at the pilot's eye.

Table 5-5. Indifference thresholds for Dander's experiment.

Axis	Error Threshold	Error Rate Threshold
1	.015 in	.025 in/sec
2	.75 deg	1.5 deg/sec
3	.07 unit	.14 unit/sec

The variable **mthresh** in the input file should read as follows:

```
mthresh = [.015 .025   .75   1.5   .07   .14];
```

Note that the indifference thresholds are half of the dead-zone widths.

In Dander's experiments, the subject was instructed not to try to get better performance in one axis than another, i.e. try to maintain equal performance in all axes. This dictates the weights on the errors in the cost function for each axis. Unity weights are set on the errors and zero weights are set on the error rates and the manipulator. The control rate weight is automatically selected to yield the desired neuromotor time constant. The form of the cost function is given below. Note that the individual axis cost functions are normalized with respect to the variance of the driving noise which deals with the different units and command signal strengths that are used in the forcing functions.

$$J_{TOT} = \sum_{i=1}^{3} \frac{1}{\sigma_{v_i}^2} \left[\sigma_{e_i}^2 + f_i \sigma_{\delta_i}^2 \right]$$

The variables mEweight, mEdotweight, and mr should be set as follows:

```
mEweight =[ 1 1 1];
mEdotweight=[0 0 0];
mr=[0 0 0];
```

All input parameters have now been defined and discussed. The Matlab script file used analyze the above example is given below.

```
%Input File mocmex5.m

clc
%Multi Axis Dander Example with output disturbances
%
naxes=3;   %Number of axes to be controlled.
```

```
%Define the controlled element transfer functions.
[ycnum1,ycden1]=senter([3,0,4,1,.04,1,.9],[3,1,0,2,.7,.25,1,
5]);
[ycnum2,ycden2]=senter([2,0,.5,1,.1],[2,1,1.5,2,-.84,.5]);
[ycnum3,ycden3]=senter([2,0,10,1,.1],[2,1,3,2,.5,.5]);
[mycnum,mycden]=tfmulti(ycnum1,ycden1,ycnum2,ycden2,ycnum3,y
cden3);


%Define the Noise Filter
[ywnum1,ywden1]=senter([1,0,.2219],[1,2,.7,.5]);
[ywnum2,ywden2]=senter([1,0,13.3],[1,2,.7,.5]);
[ywnum3,ywden3]=senter([1,0,.53],[1,2,.7,.5]);
[mywnum,mywden]=tfmulti(ywnum1,ywden1,ywnum2,ywden2,ywnum3,y
wden3);


%Define where to add the disturbances. ['out';'inp']
mdisturbance=['out'; 'out'; 'out'];


mVw=[1 1 1];              %Variance of Disturbance Noise
mtau=[.2 .2 .2];         %Reaction time delay
mtaun=[.1 .1 .1];        %Set the neuromuscular time constant
                         %typically .1 sec but ranges between
                         %.1 <= taun <= .6.
                         %taun drives the bandwidth of the
                         %pilot/vehicle system.

mrhoy=[.01 .01 .01 .01 .01 .01]; %Define full attention
                                 %observation noise variance
                                 %to signal variance ratio.
mthresh=[.015 .025 .75 1.5 .07 .14];   %Indifference
                                 %thresholds on the
                                 %error and error rate

mrhoua=[.01 .01 .01];            %Define motor noise variance to
                                 %input signal variance ratio.

mEweight=[1 1 1];        %Error Weight in the performance
                         %Index.
mEdotweight=[0 0 0];     %Error rate weight in the
                         %performance index.
mr=[0 0 0];              %Control Weight in performance index
```

Tutorial

Once the above script is executed, execute the **mocmmult.m** script which forms the table of optimal costs and finds the set of attentional fractions which minimize the performance index J_{TOT}. The results can be analyzed using the **analyze.m** script which allows one to use a GUI to view the results for the individual axes. For example, suppose you want to look at the frequency response of the full order MOCM for axis 1. Execute the analyze script. A graphics window will open and three menu headings will be shown at the top of the graphics window. Use the mouse to select Axis 1 from the Axis pulldown menu. Then select MOCM frequency response from the MOCM Analysis pulldown menu. The MOCM frequency response for axis 1 will be displayed as a Bode plot. Axes 2 and 3 can be analyzed in a similar manner. The Multi-Axis Info pull down menu can show you the optimal fractions of attention as well as Multi-axis statistics which include Cooper Harper Rating estimates. Below are examples of the Multi-Axis Info menu selections for this case.

- Optimal Attention Fractions (Figure 5-9)
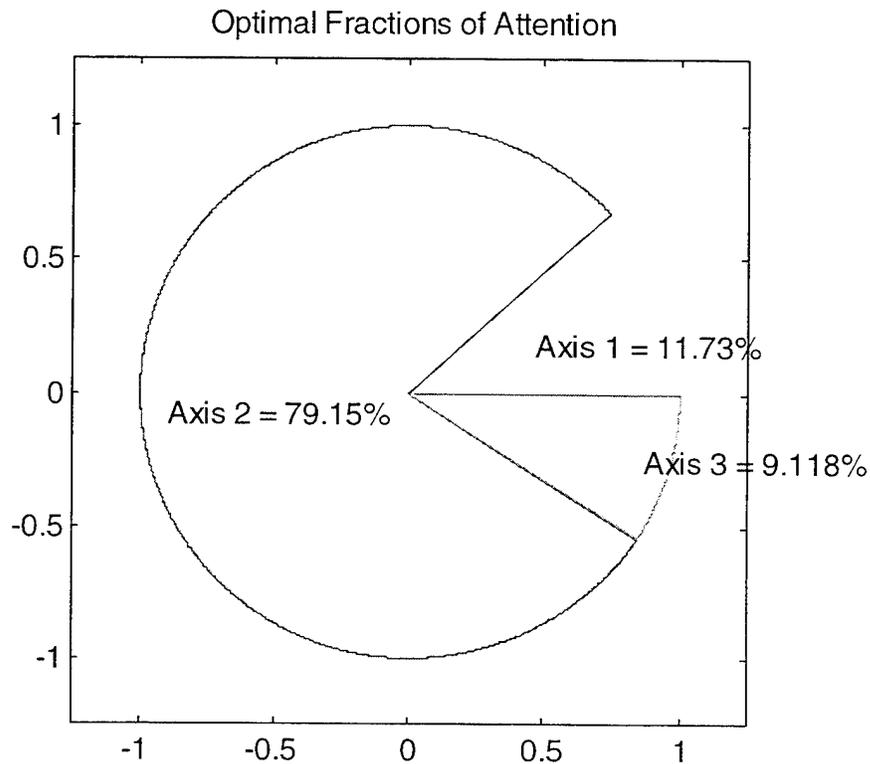
Figure 5-9. Predicted pilot attention allocation.

- Multi-Axis Statistics

» Multi-Axis Statistics

Estimated Cooper Harper Ratings

Axis 1   2.8
Axis 2   6.7
Axis 3   3.3
All Axes 8

Normalized Cost Functions (J/sigmac^2)
Axis 1   0.18581
Axis 2   0.72607
Axis 3   0.22862
All Axes 1.141

Input Noise (Forcing Function or Task)
	Bandwidth	RMS
Axis 1	0.5 rad/sec	0.3751
Axis 2	0.5 rad/sec	22.48
Axis 3	0.5 rad/sec	0.8959

Optimal Fractions of Attention
Axis 1   0.1173   Accuracy Range 1 >= fraction of attention >= 0.1
Axis 2   0.7915   Accuracy Range 1 >= fraction of attention >= 0.5
Axis 3   0.09118   Accuracy Range 1 >= fraction of attention >= 0.1

Note that Axis 2 requires about 80% of the pilots attention while the Axes 1 and 3 require roughly 12 % and 9% respectively. Recall that Axis 2 had the worst set of dynamics from the outset since it had negative damping i.e. unstable.

Another limitation of these estimates must be noted and that is the limited accuracy of the Cooper Harper Rating Estimates. An expression relating CHRs and the normalized cost function J_{TOT} and the input noise bandwidth ω_v has been given in Reference [19].

$$CHR = 5.5 + 3.7 \log_{10}\left[\frac{J_{TOT}}{\omega_v^2}\right]$$

This empirical expression was based on data from the Dander database where the pilot engaged in a compensatory tracking task with the objective of minimizing errors in all axes. Therefore, one must remember that in order for the CHR estimates to be accurate it is neccesary that the cost function error weights be equal, the error rate weights be zero and the task be compensatory tracking. Furthermore, it is possible for the CHR estimate to be off of the CHR scale since the expression does not recognize the scale limits.

6. MOCM Reference

This section contains detailed descriptions of MOCM toolbox functions. It begins with a list of functions grouped by subject and continues with the reference entries in alphabetical order. Information is also available through the online help facility.

Controlled Element Definition	
senter	Define controlled element transfer function in short hand notation
tfmulti	Create a matrix of decoupled transfer functions.

Controlled Element and Disturbance Augmentation	
ocmycin	Inject disturbance at the controlled element input.
ocmycmid	Inject disturbance into an intermediate controlled element state.
ocmycout	Inject disturbance at controlled element output. (i.e. Forcing Function)

Forcing Functions and Disturbances	
effbw	Low pass Butterworth filter with an equivalent rectangular bandwidth.
noisfilt	Low pass Butterworth filter.

Pilot Model Generation	
mocm	Single axis pilot model generator
mocmmult	Decoupled multi-axis pilot model generator

effbw

Purpose

Generate a low pass Butterworth filter of a specified order, effective rectangular bandwidth and RMS output response to unit intensity Gaussian white noise.

Synopsis

[numw,denw]=effbw(wie,rms,order);

Description

This function is provided to assist the user in creating forcing functions for tracking tasks. Forcing function bandwidth is a nebulous term for nonrectangular spectral shapes. The bandwidth of a spectrum can be defined in terms of an equivalent rectangular bandwidth. This provides a basis for comparing the bandwidth of different spectral shapes. One application where this can prove useful is in comparing results of experiments where the forcing function was a sum of sine waves to MOCM results where the input disturbance or forcing function is filtered white noise. The shapes of the input spectra are obviously different; however, if the equivalent bandwith of the sum of sines forcing function is obtained, the **effbw** function can find a low pass Butterworth filter whose equivalent rectangular bandwidth and RMS output matches those of the sum of sines. **effbw** makes use of **noisfilt** to generate Butterworth filters up to 4th order.

See Also

noisfilt

References

[1] McRuer, D.T. et al., "Human Pilot Dynamics in Compensatory Systems," AFFDL-TR-65-15.

[2] Blackman, R. B. and Tukey, J.W., "The Measurement of Power Spectra; From the Point of View of Communications Engineering," Dover Publications Inc., New York, N.Y., 1958.

mocm

Purpose

Compute a MOCM human operator model for a single axis.

Synopsis

mocm

Description

A matlab script file which calls the needed optimization routines and Matlab functions which are required to compute a MOCM of the human operator. It is best described in flow chart form (See Figure 4-3). The theoretical basis of this program can be found in the following reference:

References

[1] Davidson, John, B. and Schmidt, David, K., "Modified Optimal Control Pilot Model for Computer-Aided Design and Analysis," NASA Technical Memorandum 4382, October 1992.

mocmmult

Purpose

Compute a MOCM human operator model for multiaxis tracking tasks where the individual axes are decoupled.

Synopsis

mocm

Description

A matlab script file which calls the needed optimization routines and Matlab functions which are required to compute a multi-axis MOCM of the human operator. It solves a MOCM for each single axis for different fractions of attention to form a table of optimal costs. Parabolas are fit to the cost vs. 1/f data in the table. A constrained optimization routine then seeks to minimize the total normalized cost for all axes over the fractions of attention subject to the constraint that the sum of the attention fractions = 1. The optimal fractions of attention are then used by **analyze.m** to generate analysis data of the individual axes as well as some multi-axis information. A flow chart of the **mocmmult.m** script is shown in Figure 4-4.

References

[1] Thompson, Peter, M., "Minimum Flying Qualities, Vol. III: Program CC's Implementation of the Human Optimal Control Model," WRDC-TR-89-3125, Jan. 1990.

noisfilt

Purpose

Generate a low pass Butterworth filter with a specified order, break frequency and RMS output response to unit intensity Gaussian white noise.

Synopsis

[numw,denw]=noisfilt(bw,rms,order);

Description

This function is provided to assist the user in creating forcing functions for tracking tasks or disturbances for stabilization tasks. The only way one can specify forcing functions or disturbances in a MOCM is through filtered Gaussian white noise. An important class of forcing functions or disturbances can be defined using low pass filtered white noise. This function generates low pass filters up to 4th order with a maximally flat magnitude curve in the pass band. The filter is designed such that when a unit intensity Gaussian white noise is applied to the input, a forcing function or disturbance is generated whose RMS output is equal to that specified in the **rms** argument. The filter **order** controls how rapidly the magnitude curve rolls off at the frequency defined by **bw**. As the filter order increases, the disturbance or forcing function spectrum becomes more rectangular. To generate low pass Butterworth filters whose spectrum has an equivalent rectangular bandwith use **effbw**.

See Also

effbw

References

[1] Skelton, R.E., "Dynamic Systems Control," John Wiley and Sons, 1988, p.355.

ocmycin

Purpose

Create MOCM state space model with driving noise at the input.

Synopsis

[a,b,c,d,e]=ocmycin(ycnum,ycden,ywnum,ywden);

Description

ocmycin.m augments the noise filter and plant dynamics into a state space model of a situation where the disturbance enters at the input of the plant dynamics and the pilot observes the error and error rate on a display.
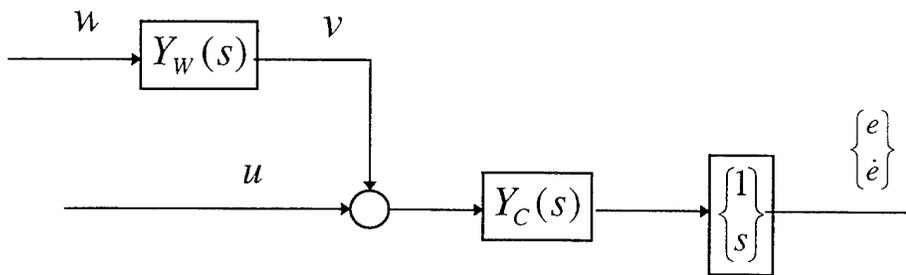
omcycin.m forms a state space system consisting of the augmented noise filter and plant. The resulting system is:

$$\begin{bmatrix} \dot{x}_w \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A_w & 0 \\ B_c C_w & A_c \end{bmatrix} \begin{bmatrix} x_w \\ x_c \end{bmatrix} + \begin{bmatrix} 0 \\ B_c \end{bmatrix} u + \begin{bmatrix} B_w \\ 0 \end{bmatrix} w$$

$$\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} D_c C_w & C_c \\ D_c C_w A_w + C_c B_c C_w & C_c A_c \end{bmatrix} \begin{bmatrix} x_w \\ x_c \end{bmatrix} + \begin{bmatrix} D_c \\ C_c B_c \end{bmatrix} u$$

See Also

ocmycout, ocmycmid

ocmycout

Purpose

Create MOCM state space model with driving noise at the output.

Synopsis

[a,b,c,d,e]=ocmycout(ycnum,ycden,ywnum,ywden);

Description

ocmycout.m augments the noise filter and plant dynamics into a state space model of a situation where the disturbance enters at the output of the plant dynamics and the pilot observes the error and error rate on a display.
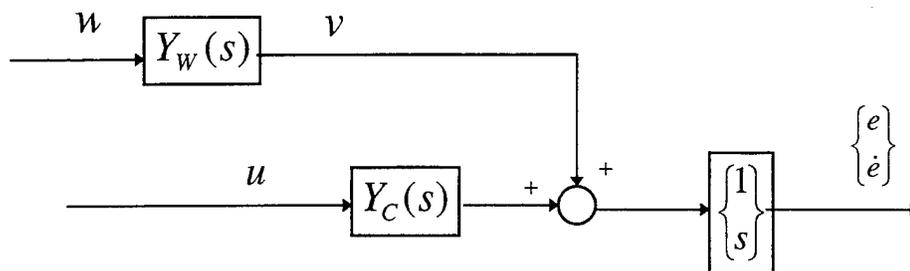
omcycout.m forms a state space system consisting of the augmented noise filter and plant. The resulting system is:

$$\begin{bmatrix} \dot{x}_w \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A_w & 0 \\ 0 & A_c \end{bmatrix} \begin{bmatrix} x_w \\ x_c \end{bmatrix} + \begin{bmatrix} 0 \\ B_c \end{bmatrix} u + \begin{bmatrix} B_w \\ 0 \end{bmatrix} w$$

$$\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} C_w & C_c \\ C_w A_w & C_c A_c \end{bmatrix} \begin{bmatrix} x_w \\ x_c \end{bmatrix} + \begin{bmatrix} D_c \\ C_c B_c \end{bmatrix} u$$

See Also

ocmycin, ocmycmid

ocmycmid

Purpose

Create MOCM state space model with driving noise at an intermediate state.

Synopsis

[a,b,c,d,e]=ocmycmid(ycnum1,ycden1,ycnum2,ycden2,ywnum,ywden);

Description

ocmycmid.m augments the noise filter and plant dynamics into a state space model of a situation where the disturbance enters in the middle of the plant dynamics and the pilot observes the error and error rate on a display.
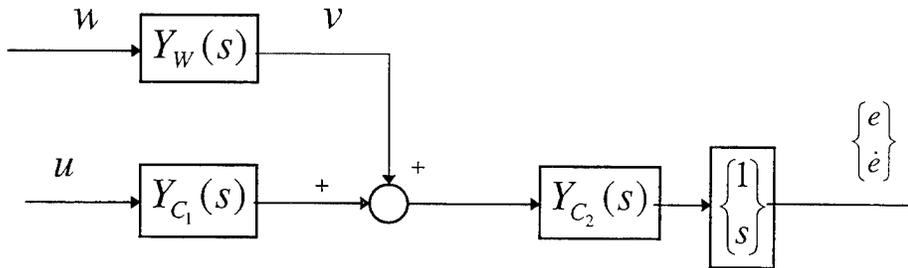
omcymid forms a state space system consisting of the augmented noise filter and plant. The resulting system is:

$$
\begin{bmatrix} \dot{x}_w \\ \dot{x}_{c_1} \\ \dot{x}_{c_2} \end{bmatrix} = \begin{bmatrix} A_w & 0 & 0 \\ 0 & A_{c_1} & 0 \\ B_{c_2}C_w & B_{c_2}C_{c_1} & A_{c_2} \end{bmatrix} \begin{bmatrix} x_w \\ x_{c_1} \\ x_{c_2} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{c_1} \\ B_{c_2}D_{c_1} \end{bmatrix} u + \begin{bmatrix} B_w \\ 0 \\ B_{c_2}D_w \end{bmatrix} w
$$

$$
\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} D_{c_2}C_w & D_{c_2}C_{c_1} & C_{c_2} \\ C_{c_2}B_{c_2}C_w + D_{c_2}C_wA_w & C_{c_2}B_{c_2}C_{c_1} + D_{c_2}C_{c_1}A_{c_1} & C_{c_2}A_{c_2} \end{bmatrix} \begin{bmatrix} x_w \\ x_{c_1} \\ x_{c_2} \end{bmatrix} + \begin{bmatrix} D_{c_2}D_{c_1} \\ C_{c_2}B_{c_2}D_{c_1} + D_{c_2}C_{c_1}B_{c_1} \end{bmatrix} u
$$

See Also

ocmycin, ocmycout

Reference

senter

Purpose
Enter a transfer function in shorthand notation

Synopsis
[num,den]=senter([nfn,ofn1,fn1,.....,ofnn,fnn],[nfd,ofd1,fd1,.....,ofdn,fdn])

Description

Parameters:
nfx = number of factors in the numerator or denominator
ofxx = order of the n'th factor
fxx = the n'th factor
= constant (if ofn = 0 (gain) or 1 (pole)
= zeta, omega (if ofn = 2)

Example:

$$G(s) = \frac{2(s+3)(s+2\zeta_n\omega_n + \omega_n^2)}{(s+5)(s+2\zeta_d\omega_d + \omega_d^2)}$$

[num,den]=senter([3,0,2,1,3,2,ζ_n,ω_n],[2,1,5,2,ζ_d,ω_d])

There are 3 factors in the numerator, a gain, zero and second order pair.
There are 2 factors in the denominator, a pole and second order pair.

Senter will handle an arbitrary number of factors in the numerator and denominator. It returns the numerator and denominator in standard Matlab transfer function format (i.e. a vector of polynomial coefficients in descending powers of s).

See Also
Control System Toolbox User's Manual. zp2tf, damp, tf2zp

tfmulti

Purpose

Form a decoupled multi-axis plant for analysis by **mocmmult.m**.

Synopsis

[mnum,mden]=tfmulti(num1,den1,num2,den2,..... , numn,denn)

Description

mocmmult.m requires a matrix of plants and noise filters to perform a multi-axis analyis. **tfmulti.m** provides a simple means of generating these transfer function matrices for both the plant and the input noise filters. The following block diagram illustrates a 2 axis problem:
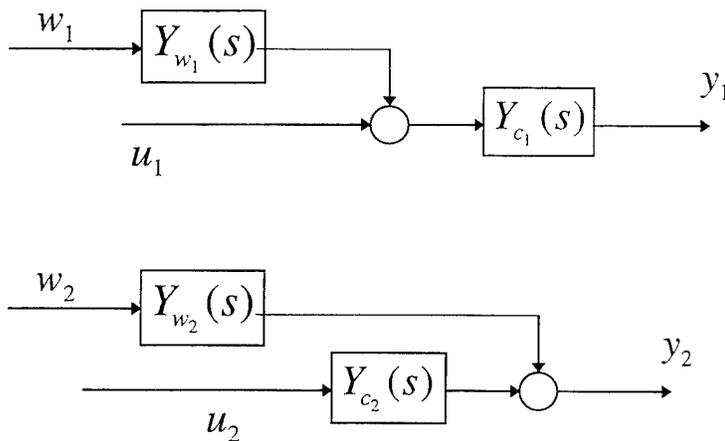
To get the system into a form that **mocmmult.m** can process use to following procedure to form a matrix of transfer functions:

Step 1: Form a matrix of noise filters:

[mnumw,mdenw]=tfmulti(numw1,denw1,numw2,denw2)

Step 2: Form a matrix of controlled elements:

[mnumc,mdenc]=tfmulti(numc1,denc1,numc2,denc2)

Step 3: Define where the disturbance enters the system (Currently only input and output disturbances are available.)

mdisturbance=['inp'; 'out']

Limitations
No more than 5 axes can be handled. This can be changed if necessary by modifying **tfmulti.m**.

Only input and output disturbances are handled.

Algolrithm
Determines the highest order transfer function in the list and pads lower order transfer functions with leading zeros. The function returns a matrix of numerators (mnum) and denominators (mden) whose column dimension is equal to the order of the highest order transfer fucntion and whose row dimension is equal to the number of axes to be controlled.

See Also
mocmmult

Bibiography

[1]   Anon: MIL-STD-1797A "Flying Qualities of Piloted Aircraft," Jan 30, 1990 ASD/ENES, Wright-Patterson AFB, OH.

[2]   Blackman, R. B. and Tukey, J.W., "The Measurement of Power Spectra; From the Point of View of Communications Engineering," Dover Publications Inc., New York, N.Y., 1958.

[3]   Chalk, C.R., "Calspan Recommendations for SCR Flying Qualities Design Criteria," NASA-CR-159236, April, 1980.

[4]   Dander,V., 1962: An Evaluation of Four Methods for Converting Single Axis Pilot Ratings Using Fixed Base Simulation Data : MS Thesis, Air Force Inst. of Technology, Wright-Patterson AFB, OH, GE/EE/62-4.

[5]   Davidson, John B. and Schmidt, David K.: "Modified Optimal Control Pilot Model for Computer-Aided Design and Analysis," NASA -TM-4384.

[6]   Gibson, J.C., "Handling Qualities for Unstable Combat Aircraft," ICAS-86-5.3.4, Sept, 1986.

[7]   Hoh, Roger H., Mitchel, David G. and Hodgkinson, John., " Proposed Military Standard and Handbook -- Flying Qualities of Air Vehicles, Vol II, Proposed Handbook," AFWAL-TR-82-3081, 1982.

[8]   Iloputaife, Obi and Ma, Lily.: "Handling Qualities Design of a Northrop High-Performance Fighter," AIAA-87-2450, 1987.

[9]   Kleinman, D.L.; Baron, S.; and Levison, W. H. 1970: An Optimal Control Model of Human Response -- Part 1: Theory and Validation. Automatica, vol 16, no.3, pp. 237-253.

[10]   McRuer, D. T. and Schmidt, D. K., 1989: Pilot-Vehicle Analysis of Multiaxis Tasks: AIAA Journal of Guidance, Control and Dynamics., vol. 13, no. 2, pp. 348-355.

[11]   McRuer, D.T. and Krendel, E.S., 1974: Mathematical Models of Human Pilot Behavior: AGARD-AG-188.

[12]   McRuer, D.T. et al., "Human Pilot Dynamics in Compensatory Systems," AFFDL-TR-65-15, WPAFB, OH, 1965.

[13]   Mitchell, David G., Hoh, Roger H., Aponso, Bimal L., and Klyde, David., "Incorporation of Mission Oriented Flying Qualities into MIL-STD-1797A," WL-TR-94-3162, 1994.

[14]   Neal, T. Peter and Smith, Rogers E.: "An In-Flight Investigation to Develop Control System Design Criteria for Fighter Airplanes," AFFDL-TR-70-74, Vol I, Dec. 1970.

[15]   Reynolds, P.A., "Criteria for Handling Qualities in Military Aircraft; Simulation for Predicting Flying Qualities," AGARD-CP-333, June, 1982.

[16]   Skelton, R.E., "Dynamic Systems Control," John Wiley and Sons, 1988.

[17]   Smith, Ralph H. and Geddes, Norman D.: "Handling Quality Requirements for Advanced Aircraft Design: Longitudinal Mode," AFFDL-TR-78-154, Aug. 1979.

[18]   Strang, Gilbert., "Linear Algebra and its Applications, 3rd ed," Harcourt, Brace, Jovanovich, SanDiego, 1988.

[19]   Thompson, Peter, M., "Minimum Flying Qualities, Vol. III: Program CC's Implementation of the Human Optimal Control Model," WRDC-TR-89-3125, Jan. 1990.