TR-85-16

DESIGN OF A FLIGHT CONTROLLER FOR AN F14 AIRCRAFT USING THE DELIGHT.MaryLin OPTIMIZATION-BASED CACSD SYSTEM

by

M.K.H. Fan
A.L. Tits
W.S. Levine
C.D. Walrath
M. Rimer
C. Lee
R. Grant

TR-85-16

Design of a Flight Controller for an F14 Aircraft Using the DELIGHT.MaryLin Optimization-Based CACSD System

by

M.K.H. Fan[†], *C.D. Walrath*[‡], *C. Lee*[†], *A.L. Tits*[†], *M. Rimer*[§], *R. Grant*[§], *W.S. Levine*[†]

[†] Dept. of Electrical Engineering, Univ. of Maryland, College Park

[‡] Westinghouse Defense and Electronics Center, Baltimore, Md.

[§] Grumman Aerospace Corporation, Bethpage, N.Y.

This example problem was submitted to us by Grumman Aerospace to assess the differences between our design methodology—see attached report for a complete description—as implemented in DELIGHT.MaryLin, and the Linear Quadratic Regulator approach used in their CASCADE program.

We start from a model of the entire system including: an implicit model for the pitch angle response; models for wind gust and sensor error; and a specified structure of the controller. The 'best' values of some controller parameters remain to be determined. The model has several artificial features that were introduced by Grumman to allow its handling by CASCADE—for instance, the step input is modeled as slowly decreasing to zero. Even though such peculiarities are not needed by DELIGHT.MaryLin, the model was left unaltered to permit a better assessment of the differences between the two approaches. A block diagram of the model is attached.

Our design methodology allows the designer to consider diverse specifications simultaneously and to explore resulting trade-offs interactively. For the F14 design problems, our specifications (as suggested to us by Grumman), consist of: bounds on the frequency response of the pitch rate; stability criteria (gain margin, phase margin); bounds on the covariance of the tail rate, of the normal acceleration at the pilot station, and of the angle of attack, due to both wind gust and sensor error; bounds (for all times) on the magnitude of the tail rate, on the error in angle of attack as compared to that generated by the implicit model, and on the jerk (derivative of the normal acceleration) at the pilot station. Several of these specifications (stochastic specifications, bound on jerk) are concerned with 'flying qualities' (see [1]).

In order to be able to compare the performance of a given design relative to such diverse specifications, we use the concepts of *good value* (or target value) and *bad value*, to be given by the designer for each of the specifications. The 'badness' (or scaled value) of a design in relation to a given specification f is then given by

$$\frac{f - f_{\text{good}}}{f_{\text{bad}} - f_{\text{good}}}\,.$$

The goal of is then to perform a 'minimax' optimization of all the scaled values.

For specifications on a time or frequency response (functional specifications), good and bad values may be functions of time or frequency.

DELIGHT.MaryLin formulation of the design problem

A DELIGHT.MaryLin design problem formulation generally consists of 5 computer files. A printout of these files for the F14 problem is attached.

The 'S' (setup) file contains a description of the system to be controlled as well as a list of design parameters with their initial values (these should be the best ones the designer can think of).

The 'M' (multicost) file contains a description of the ordinary (non-functional) objective functions, i.e., of functions of the design parameters that the designer wishes to minimize (or maximize). (There is none in our F14 example.)

The 'FM' (functional multicost) file contains a description of the functional objective functions. These are functions of the design parameters and also of an independent parameter (time, frequency, ...). The designer wishes to minimize (or maximize) the worst value (over the independent parameter) of such functions.

The 'I' (inequality constraint) and 'FI' (functional inequality constraint) files are analogous to 'M' and 'FM' except that here, instead of trying to minimize, the designer wishes to reach (or at least approach) target values (good values).

As long as all the objectives and soft constraints have not reached their good values (as is the case in this example), *all specifications compete equally.*

Pcomb graphical display

This display is meant to convey to the designer information about the overall performance of the current design—as well as about the previously displayed design, in light color, for the sake of comparison—and help him explore tradeoffs. A complete description of this display is given on pages 29-32 of [2]. Hard copies (monochrome, unfortunately) of the *Pcomb* display corresponding to iterations 0 (initial guess) versus 12 and 12 versus 24 (near final design) are attached.

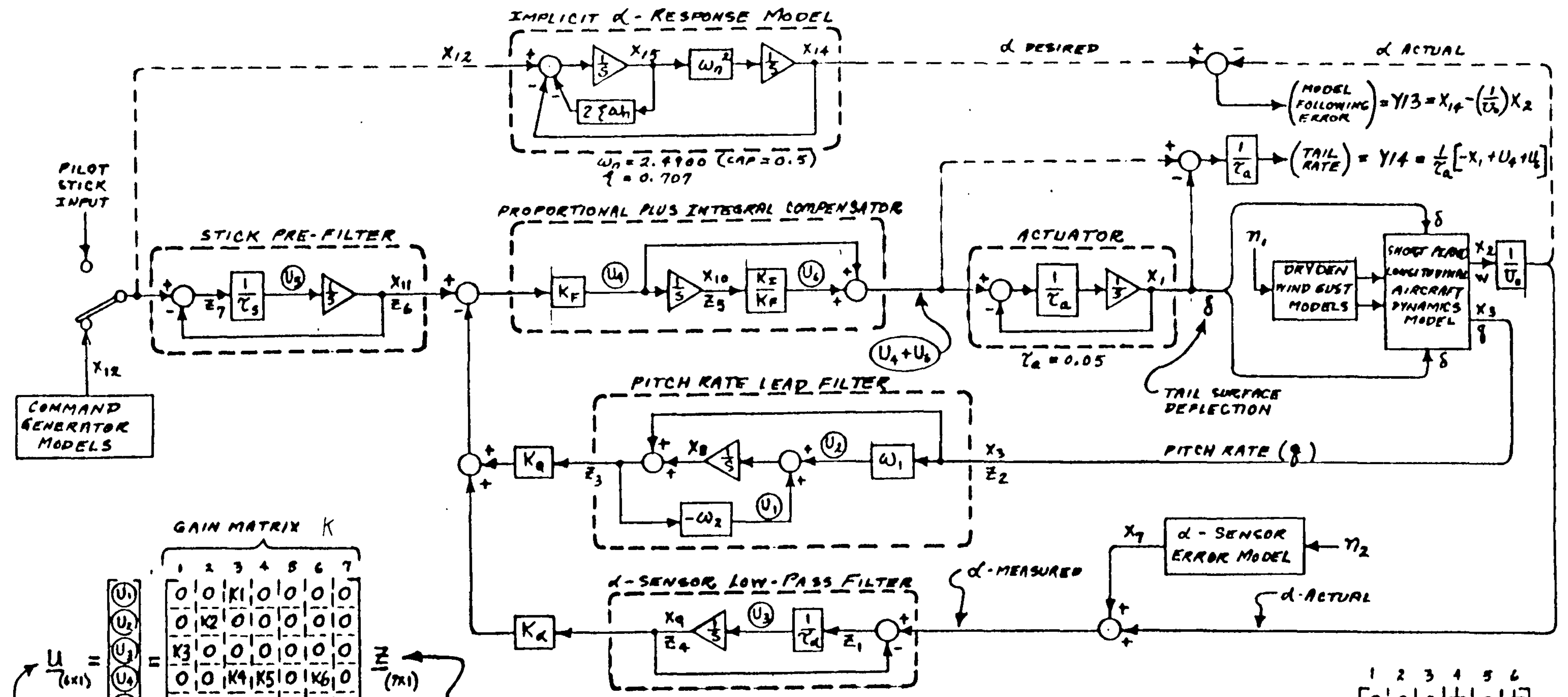
References

[1] "Military Specification. Flying Qualities of Piloted Airplanes," MIL-F-8785C, ASD/ENESS, Wright-Patterson AFB (November 1980).

[2] W.T. Nye and A.L. Tits, "An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems," to appear in International Journal of Control (1985).

FIGURE 1 ~ STATE VARIABLE REPRESENTATION OF CONTROLLER, ACTUATOR, IMPLICIT MODEL, MODEL FOLLOWING ERROR, & TAIL RATE

(BASELINE FLIGHT CONTROL SYSTEM DESIGN EXAMPLE)



GAIN MATRIX K

$$\underset{(6\times1)}{\underline{U}} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix} = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & K1 & 0 & 0 & 0 & 0 \\ 0 & K2 & 0 & 0 & 0 & 0 & 0 \\ K3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K4 & K5 & 0 & K6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K7 \\ 0 & 0 & 0 & 0 & K8 & 0 & 0 \end{array}\right]_{(6\times7)} \end{array} \underset{(7\times1)}{\underline{Z}}$$

CONTROL VECTOR

$K1 = -\omega_2$ $\quad K5 = -K_F K_\alpha$

$K2 = \omega_1$ $\quad K6 = K_F$

$K3 = 1/\tau_\alpha$ $\quad K7 = 1/\tau_s$

$K4 = -K_F K_q$ $\quad K8 = K_I/K_F$

OBSERVATION VECTOR

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \end{bmatrix} = \begin{array}{c} \begin{array}{ccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{array} \\ \left[\begin{array}{ccccccccccccccc}   & \frac{1}{U_0} &   &   &   &   & 1 &   & -1 &   &   &   &   &   &   \\   &   & 1 &   &   &   &   &   &   &   &   &   &   &   &   \\   &   & 1 &   &   &   &   & 1 &   &   &   &   &   &   &   \\   &   &   &   &   &   &   &   & 1 &   &   &   &   &   &   \\   &   &   &   &   &   &   &   &   & 1 &   &   &   &   &   \\   &   &   &   &   &   &   &   &   &   & 1 &   &   &   &   \\   &   &   &   &   &   &   &   &   &   & -1 & 1 &   &   &   \end{array}\right] \end{array} \underset{(15\times1)}{\underline{X}}$$

AUGMENTED SYSTEM STATE VECTOR

CONTROL INPUT COUPLING MATRIX $B =$

$$\begin{array}{c|c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ \vdots \\ 15 \end{array} & \left[\begin{array}{cccccc} 0 & 0 & 0 & \frac{1}{\tau_a} & 0 & \frac{1}{\tau_a} \\ \\ & & O_{(6\times6)} & & & \\ \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \\ & & O_{(3\times6)} & & & \end{array}\right] \end{array}$$

$$\left(\begin{array}{c} \dot{\underline{X}} = A\underline{X} + B\underline{U} \\ \text{or} \\ \dot{\underline{X}} = (A + BKC_m)\underline{X} \end{array}\right)$$

p 4

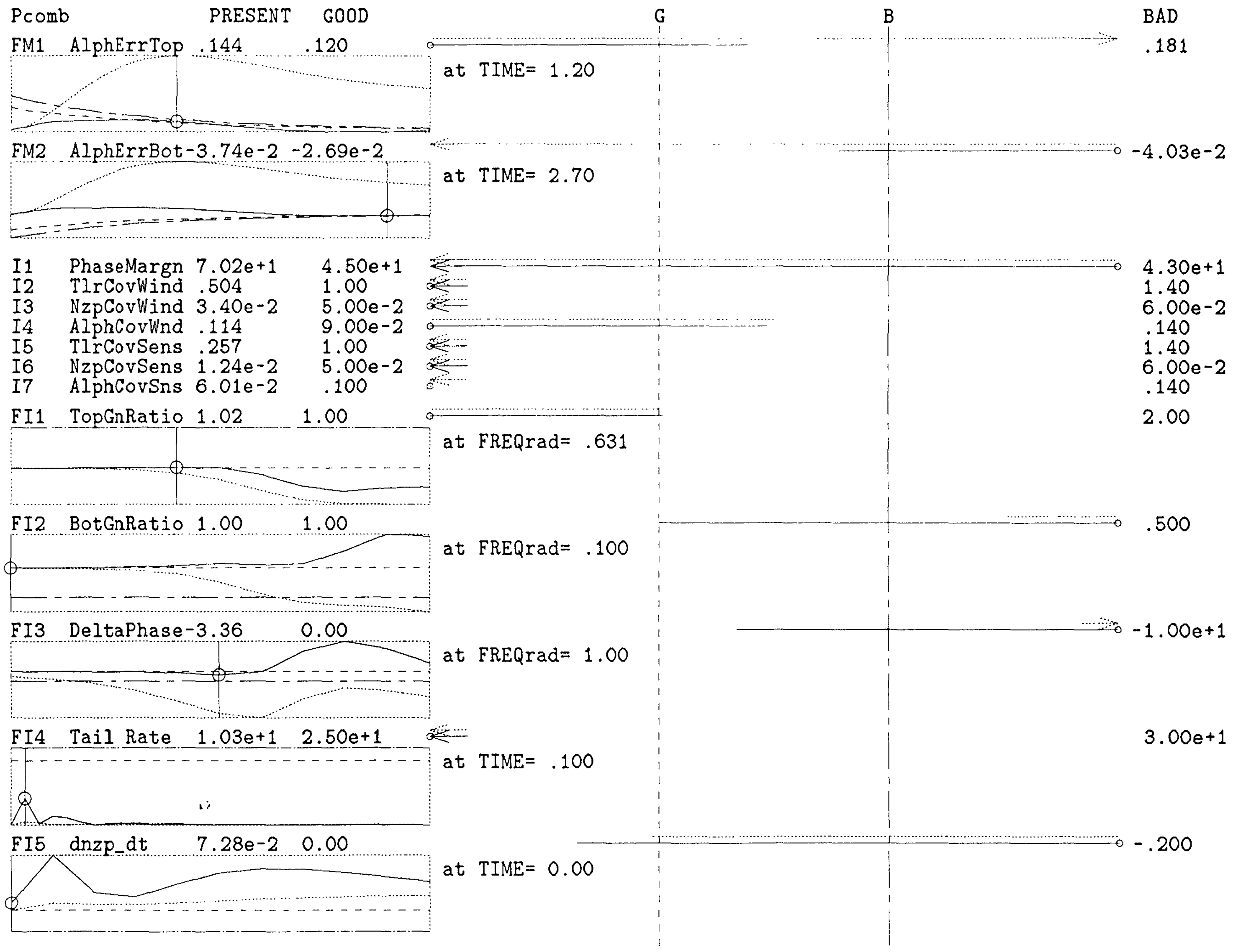


Figure 2 : Pcomb for F14 design at iterations 0 and 12.

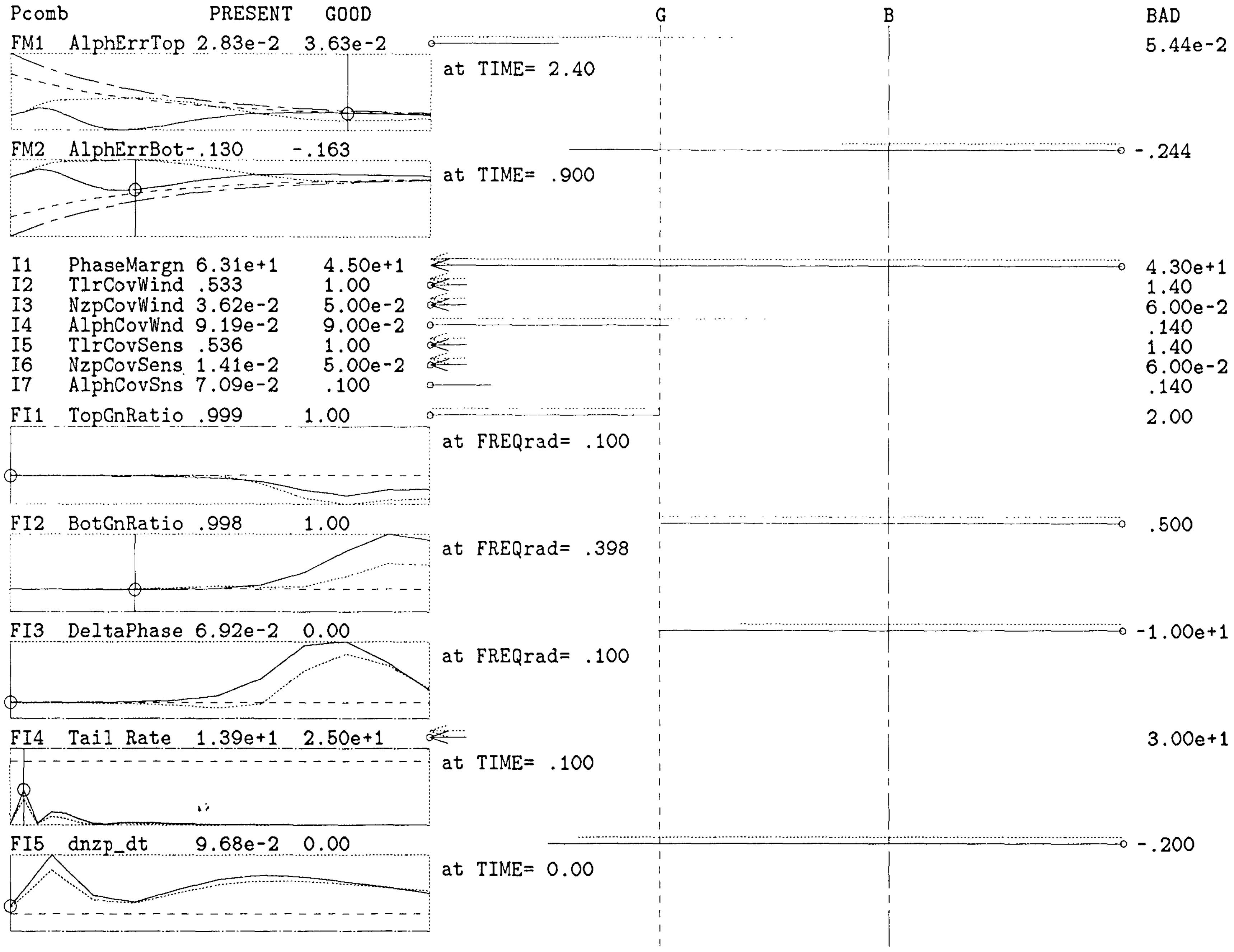


Figure 3 : Pcomb for F14 design at iterations 12 and 24.

```
#=======
# F14S - ("S" file) Setup file for Grumman F14 pitch control problem
#=======

 define (TO_DEGREES,(360/TWOPI))
 define (TO_RADIANS,(TWOPI/360))

# defines used to distinguish 2 types of noise analysis.
 define (WIND_GUST,   1)
 define (SENSOR_NOISE,2)

# We make use of a state space representation of the system that includes
# the various sensors and controllers (this was provided by Grumman).
# A complete description of the model is available on demand.

# System model :
#
#    Xdot  =  A * X + B * U + G * N
#    Y     =  C * X + D * U
#    U     = -K * Cm * X + UFORCE
#
#  where A,B,C,D,Cm,G are given,
#
#    A:  Nstates   * Nstates
#    B:  Nstates   * Ninputs
#    C:  Noutputs  * Nstates
#    D:  Noutputs  * Ninputs
#    Cm: Nmoutputs * Nstates
#    G:  Noutputs  * Nninputs
#
#  and the structure of matrix K (Ninputs*Nmoutputs) is given as
#
#      | 0  0  K1 0  0  0  0  |
#      | 0  K2 0  0  0  0  0  |
#    - | K3 0  0  0  0  0  0  |
#      | 0  0  K4 K5 0  K6 0  |
#      | 0  0  0  0  0  0  K7 |
#      | 0  0  0  0  K8 0  0  |
#
#  where (-K1) through (-K8) are our design parameters.
#

# system dimensions
 Ninputs   =  6
 Nninputs  =  2          # Noise inputs
 Nstates   = 15
 Noutputs  = 16          # Output of design related interest
 Nmoutputs =  7          # Measured outputs (used in feedback)
 matrix_sizes


#==================================#
# PARAMETERS USED IN THE SYSTEM MODEL #
#==================================#
```

```
# Universal constant
 g = 32.2          # gravitational acceleration (feet/sec**2)

# Geometric parameter
 L = 22.88         # distance from CG to pilot station (ft)

# Parameters for actuator
 Ta_inv = 20             # bandwidth (1/sec)

# Parameters for aircraft dynamics model: F14 at Mach 0.71,
# 35 Kft, total air speed=690.4 ft/sec, alpha trim=3.03 degrees.
 Zdelta  = -63.9979  # force in vertical direction due to displacement
                     #    of control surface (actuator motion): ft/(rad sec**2)
 Mdelta  = -6.8847   # pitching moment due to displacement of control
                     #    surface (actuator motion): 1/(rad sec**2)
 Uo      = 689.4     # longitudinal velocity (feet/second)
 Zw      = -0.6385   # force in vertical direction due to vertical
                     #    velocity (1/sec)
 Mw      = -5.92e-3  # pitching moment due to vertical velocity (1/(ft sec))
 Mq      = -0.6571   # pitching moment due to pitch rate (1/sec)

# Parameters for command generator
 alpha = 0.01             # slow inverse time constant (1/sec)
 beta  = 426.4352         # fast inverse time constant (1/sec)

# Parameters for angle-of-attack (alpha) implicit model
 Wn  = 2.49          # control anticipation parameter (CAP) = 0.5
 ksi = 0.707         # critical damping

# Parameters and bounds for second order model for pitch rate
 tautheta2  = 1.7                          # numerator time constant
 taueff_max =  .1                          # maximum delay allowed
 wsp_min    = 2.122  ; wsp_max    = 3.5214 # bounds on equivalent nat. freq
 ksisp_min  =  .5    ; ksisp_max =  .707   # bounds on eq. dumping coeff.

# Artificial open loop poles (needed for Grumman's LQ approach)
 P1 = -1e-5
 P2 = -2e-5
 P3 = -3e-5
 P4 = -4e-5

# Parameters for wind gust model (Grumman data sheet No. 4)
 a         = 2.5348              # gust correlation time (seconds)
 sigma_wg  = 3.0                 # rms of gust velocity (ft/sec)
 b         = 64.13               # reference span (feet)
 Vto       = 690.4               # total air speed (feet/second)

# Parameters for alpha sensor noise
 sigma_alpha  = TO_RADIANS*.3degrees # rms of alpha sensor noise (rad)
 Walpha   = 10                       # half variance density bandwidth (rad/sec)

# Parameters for G matrix
 G51          = sigma_wg/sqrt(a**3)
 G72          = sqrt(2*Walpha)*sigma_alpha

# System matrices
```

```
readmatrix ~expressions A

 -Ta_inv  0   0    0               0    0   0   0   0   0   0   0   0    0   0
 Zdelta  Zw  Uo  -Zw   -Zw*sqrt(3)*a  0   0   0   0   0   0   0   0    0   0
 Mdelta  Mw  Mq  -Mw-Mq*PI/(4*b)  -Mw*sqrt(3)*a-Mq*PI*sqrt(3)*a/(4*b)
                   Mq*PI*Vto/(4*b)      0   0   0   0   0   0   0    0   0
  0       0   0    0               1    0   0   0   0   0   0   0   0    0   0
  0       0   0 -1/(a**2)        -2/a   0   0   0   0   0   0   0   0    0   0
  0       0   0  PI/(4*b)  PI*sqrt(3)*a/(4*b)  -PI*Vto/(4*b)
                                            0   0   0   0   0   0   0    0   0
  0       0   0    0               0   0 -Walpha 0  0   0   0   0   0    0   0
  0       0   0    0               0    0   0  P1   0   0   0   0   0    0   0
  0       0   0    0               0    0   0   0  P2   0   0   0   0    0   0
  0       0   0    0               0    0   0   0   0  P3   0   0   0    0   0
  0       0   0    0               0    0   0   0   0   0  P4   0   0    0   0
  0       0   0    0               0    0   0   0   0   0   0 -beta beta 0   0
  0       0   0    0               0    0   0   0   0   0   0   0 -alpha 0   0
  0       0   0    0               0    0   0   0   0   0   0   0   0    0 Wn**2
  0       0   0    0               0    0   0   0   0   0   0   1   0 -1 -2*ksi*Wn

readmatrix  ~expressions B

 0 0 0  Ta_inv  0  Ta_inv
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 1 1 0  0       0  0
 0 0 1  0       0  0
 0 0 0  1       0  0
 0 0 0  0       1  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0
 0 0 0  0       0  0

readmatrix ~expressions C

 0    0    0    0    0    0    0    0    0    0    0    0 0 TO_DEGREES  0
 0    0    0    0    0    0    0    0    0    0    0   TO_DEGREES 0 0   0
 -TO_DEGREES*Ta_inv 0 0 0 0 0  0    0    0    0    0    0    0    0    0
 (-A(2,1)+L*A(3,1))/g  (-A(2,2)+L*A(3,2))/g  (-A(2,3)+Uo+L*A(3,3))/g
             (-A(2,4)+L*A(3,4))/g  (-A(2,5)+L*A(3,5))/g L*A(3,6)/g
                                 0    0    0    0    0    0    0    0    0
 0 1/Uo*TO_DEGREES 0 0 0    0    0    0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0    0    0 TO_DEGREES 0 0     0    0
 A(3,1)*TO_DEGREES     A(3,2)*TO_DEGREES     A(3,3)*TO_DEGREES
                       A(3,4)*TO_DEGREES     A(3,5)*TO_DEGREES
                       A(3,6)*TO_DEGREES 0 0 0 0    0    0    0    0    0
 -A(2,1)/g    -A(2,2)/g      (-A(2,3)+Uo)/g     -A(2,4)/g     -A(2,5)/g
                            0    0    0    0    0    0    0    0    0
 0  -TO_DEGREES/Uo   0     0    0     0 0   0 0 0    0    0 0 TO_DEGREES   0
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 0     0 TO_DEGREES 0 0     0     0     0     0     0     0     0     0     0     0
 0     0 TO_DEGREES 0 0     0     0 TO_DEGREES 0 0     0     0     0     0     0
 0  -1/Uo  0     0     0     0     0     0     0     0     0     0     0     1     0
 -Ta_inv 0 0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     1     0     0     0     0     0     0
 0     0     1     0     0     0     0     1     0     0     0     0     0     0     0

readmatrix ~expressions D

 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     TO_DEGREES*Ta_inv    0      TO_DEGREES*Ta_inv
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     TO_DEGREES           0      TO_DEGREES
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0
 0     0     0     0                    0      0

readmatrix ~expressions Cm

0    1/Uo    0 0 0 0 1 0 -1 0   0 0 0 0 0
0    0       1 0 0 0 0 0  0 0   0 0 0 0 0
0    0       1 0 0 0 0 1  0 0   0 0 0 0 0
0    0       0 0 0 0 0 0  1 0   0 0 0 0 0
0    0       0 0 0 0 0 0  0 1   0 0 0 0 0
0    0       0 0 0 0 0 0  0 0   1 0 0 0 0
0    0       0 0 0 0 0 0  0 0  -1 1 0 0 0


readmatrix ~expressions Xtr0          # For 2 degree alpha command

0 0 0 0 0 0 0 0 0 0 0 0   TO_RADIANS*((beta-alpha)/beta)*2degrees    0 0

# Design parameters

 sim_design_parameter K1_3  variation=4.144      # Grumman -K1
 sim_design_parameter K2_2  variation=2.971      # Grumman -K2
 sim_design_parameter K3_1  variation=2.526      # Grumman -K3
 sim_design_parameter K4_3  variation=1.361      # Grumman -K4
 sim_design_parameter K4_4  variation=1.182      # Grumman -K5
 sim_design_parameter K4_6  variation=1.746      # Grumman -K6
 sim_design_parameter K5_7  variation=10.00      # Grumman -K7
 sim_design_parameter K6_5  variation=2.213      # Grumman -K8

#   # Initial guesses (Grumman's solution).
#
#     set K1_3  =  4.144
#     set K2_2  = -2.971
#     set K3_1  = -2.526
#     set K4_3  = -1.361
#     set K4_4  = -1.182
#     set K4_6  =  1.746
#     set K5_7  = -10.00
#     set K6_5  = -2.213

## Set initial guesses. (Grumman's initial guess)

 set K1_3 =  1                  # K1 = -1
 set K2_2 = -1                  # K2 =  1
 set K3_1 = -1                  # K3 =  1
 set K4_3 = -1                  # K4 =  1
 set K4_4 = -1                  # K5 =  1
 set K4_6 =  1                  # K6 = -1
 set K5_7 = -1                  # K7 =  1
 set K6_5 = -1                  # K8 =  1

## Set initial guesses. (from Grumman's initial guess and run 20 iterations)

#  set K1_3 =   1.041
#  set K2_2 =  -1.138
#  set K3_1 =  -1.687
#  set K4_3 =  -1.133
#  set K4_4 =  -1.200
#  set K4_6 =   1.879
#  set K5_7 =  -1.038e+1
#  set K6_5 =  -1.471

# Simulation outputs    # by these declarations, the simulation routines
                        # (time and frequency response computation) are
                        # notified that the specified responses should
                        # always be saved (they are used in the specifications)

 simulation_output Ytr(3)      # tail rate (degrees/sec)
 simulation_output Ytr(4)      # normal acceleration at pilot station (g's)
 simulation_output Ytr(9)      # pitch position (alpha) error (degrees)
 simulation_output Ymag(11)    # magnitude of freq. resp. of pitch rate (deg/sec)
 simulation_output Yphase(11)  # phase     of freq. resp. of pitch rate (degrees)
 simulation_output Yimag(15)   # imag. part of freq. resp. of alpha sensor output
 simulation_output Yreal(15)   # real  pt of freq. resp. alpha sensor output
 simulation_output Yimag(16)   # imag. pt of freq. resp. pitch rate filter output
 simulation_output Yreal(16)   # real  pt of freq. resp. pitch rate filter output

# Parameters used by algorithm

 Nparam      = 8               # number of parameters
 Nfmulticost = 2               # number of functional multicosts
 Nineq       = 7               # number of inequalities
 Nfineq      = 5               # number of functional inequalities

# Procedure to compute covariance matrix

procedure F14_outputCOV(analysis_type)  {

   import G, G51, G72
```

```
    if FIRSTCALL saved_analysis_type = NULL

    if (analysis_type != saved_analysis_type) {
       zero_out G
       saved_analysis_type = analysis_type

       case1 (analysis_type == WIND_GUST)
          G(5,1)=G51

       case2 (analysis_type == SENSOR_NOISE)
          G(7,2)=G72

       touch G
       }

    OutputCOV
    }
```

```
#========
# F14FM - ("FM" file) Specifies functional objective for Grumman
#======== F14 pitch control problem



#============
prob_function  fmulticost
#============


#==============
# objective 1 -  Minimize alpha error (in degrees)
#==============


#  objective 1 'AlphaError' minimize Ytr(9) good=0.2 bad=0.4
   objective 1 'AlphErrTop' minimize Ytr(9) good=0.4*exp(-TIME) \
      bad=0.6*exp(-TIME)
   for_every TIME from 0 to 3 initially by .1



#==============
# objective 2 -  Maximize alpha error (in degrees)
#==============


#  objective 2 'AlphaError' maximize Ytr(9) good=-.01 bad=-.015
   objective 2 'AlphErrBot' maximize Ytr(9) good=-0.4*exp(-TIME) \
      bad=-0.6*exp(-TIME)
   for_every TIME from 0 to 3 initially by .1




end_multicost
```

```
#=======
# F14I - ("I" file) Specifies inequality constraints for Grumman
#======= F14 pitch control problem


#============
prob_function  ineq
#============


   define(LGi, saved_K44*Yimag(15)+saved_K43*Yimag(16))
   define(LGr, saved_K44*Yreal(15)+saved_K43*Yreal(16))
   define(LOOP_GAIN,  sqrt((LGr)**2+(LGi)**2))
   define(LOOP_PHASE, (TO_DEGREES*atan2(LGi,LGr)))


   import UFORCE, COV

#=====================================================#
# No constraint on gain margin since always = infinity #
#=====================================================#


#===============
# constraint 1 -  Lower bound on phase margin
#===============


   constraint 1 'PhaseMargn' margin >= good=45 bad=43 soft using{
      UFORCE(4) = 1

      saved_K43 = K4_3 ; set K4_3 = 0        # to open ...
      saved_K44 = K4_4 ; set K4_4 = 0        # ... the loop

      findroot LOOP_GAIN-1 vs FREQrad from 0 to 100

      margin =  LOOP_PHASE+180

      set K4_3 = saved_K43                   # closing back ...
      set K4_4 = saved_K44                   # ... the loop

      UFORCE(4) = 0
      }

#============================================================#
# Constraints 2 to 7 express bounds on the covariance of       #
# the tail rate, normal acceleration at the pilot station,      #
# and angle of attack, due respectively to rms wind gust        #
# of 3 ft/sec (2 to 4) and rms sensor error of 0.3 deg (5 to 7)#
#==============================================================#


#===============
# constraint 2 -  Tail rate (degree/sec)
#===============


   constraint 2 'TlrCovWind' sigma <= good=1 bad=1.4 soft using {
      F14_outputCOV (WIND_GUST)
      sigma = sqrt(COV(3,3))
      }



#===============
# constraint 3 -  Normal acceleration at pilot station (g's)
#===============


   constraint 3 'NzpCovWind' sigma <= good=.05 bad=.06 soft using {
      F14_outputCOV (WIND_GUST)
      sigma = sqrt(COV(4,4))
      }



#===============
# constraint 4 -  Angle of attack (degrees)
#===============


#  constraint 4 'AlphCovWnd' sigma <= good=0.1 bad=0.14 soft using {
   constraint 4 'AlphCovWnd' sigma <= good=0.09 bad=0.14 soft using {
      F14_outputCOV (WIND_GUST)
      sigma = sqrt(COV(9,9))
      }



#===============
# constraint 5 -  Tail rate (degree/sec)
#===============


   constraint 5 'TlrCovSens' sigma <= good=1 bad=1.4 soft using {
      F14_outputCOV (SENSOR_NOISE)
      sigma = sqrt(COV(3,3))
      }



#===============
# constraint 6 -  Normal acceleration at pilot station (g's)
#===============


   constraint 6 'NzpCovSens' sigma <= good=.05 bad=.06 soft using {
      F14_outputCOV (SENSOR_NOISE)
      sigma = sqrt(COV(4,4))
      }



#===============
# constraint 7 -  Angle of attack (degrees)
#===============


   constraint 7 'AlphCovSns' sigma <= good=0.1 bad=0.14 soft using {
      F14_outputCOV (SENSOR_NOISE)
      sigma = sqrt(COV(9,9))
      }




end_ineq
```

```
#========
# F14FI - ("FI" file) Specifies functional inequality constraints
#======== for Grumman F14 pitch control problem



#============
prob_function  fineq
#============

   import UFORCE, K
   import tautheta2, taueff_max, wsp_min, wsp_max, ksisp_min, ksisp_max


#==================================================================#
# Constraints 1 to 3 aim at fitting between 2 given second order #
# models the transfer function from output of command generator  #
# to pitch rate.                                                 #
#==================================================================#



#================
#  constraint 1 -  Upper bound on the magnitude ratio to the
#================  top given model


   constraint 1 'TopGnRatio' top_ratio <= good=1.0 bad=2.0 soft using {
      UFORCE(5) = 1
      w = FREQrad
      FREQrad = 0
      dc_gain = Ymag(11)
      FREQrad = w
      top_ratio = Ymag(11) / (dc_gain*
         sqrt((1+(FREQrad*tautheta2)**2)/
              ((1-(FREQrad/wsp_max)**2)**2 + (2*ksisp_min*FREQrad/wsp_max)**2)))
      UFORCE(5) = 0
      }
   for_every FREQrad from .1 to 10 initially dec 5



#================
#  constraint 2 -  Lower bound on the magnitude ratio to the bottom
#================  given model


   constraint 2 'BotGnRatio' bot_ratio >= good=1.0 bad=0.5 soft using {
      UFORCE(5) = 1
      w = FREQrad
      FREQrad = 0
      dc_gain = Ymag(11)
      FREQrad = w
      bot_ratio = Ymag(11) / (dc_gain*
         sqrt((1+(FREQrad*tautheta2)**2)/
              ((1-(FREQrad/wsp_min)**2)**2 + (2*ksisp_max*FREQrad/wsp_min)**2)))
      UFORCE(5) = 0
      }
   for_every FREQrad from .1 to 10 initially dec 5



#================
#  constraint 3 -  Lower bound on the phase difference with the
#================  most-negative-phase model


   constraint 3 'DeltaPhase' delta_phase >= good=0 bad=-10 soft using {
      UFORCE(5) = 1
      denom_real = 1-(FREQrad/wsp_min)**2
      denom_imag = 2*ksisp_min*FREQrad/wsp_min
      delta_phase = Yphase(11) -
                    TO_DEGREES *  (atan2(FREQrad*tautheta2,1) -
                                   atan2(denom_imag,denom_real) -
                                   taueff_max*FREQrad)
      UFORCE(5) = 0
      }
   for_every FREQrad from .1 to 10 initially dec 5



#================
#  constraint 4 -  Upper bound on the magnitude of the tail rate
#================  (in deg/sec)


   constraint 4 'Tail Rate' abs(Ytr(3)) <= good=25 bad=30 soft
   for_every TIME from 0 to 3 initially by .1



#================
#  constraint 5 -  Lower bound on the pilot jerk
#================  (derivative of acceleration: in g/sec)


   constraint 5 'dnzp_dt' dnzp_dt >= good=0 bad=-.2 soft using {
      nzp  = Ytr(4)
      t    = TIME
      TIME = TIME + .01
      dnzp_dt = (Ytr(4)-nzp)/.01    # finite difference: derivative of Ytr(4)
      TIME = t
      }
   for_every TIME from 0 to 1 initially by .1


end_fineq
```