Multivariable Control Design for Twin-Lift Helicopter System

By

Varadaraya Ganesh Shenoy
(1217423766)

Arizona State University

Electrical and Computer Engineering
Ira.A.Fulton School of Engineering

DECEMBER 2020

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Abstract

This project aims at developing a centralized multivariable automatic flight control system (AFCS) for a twin-lift helicopter system(TLHS). The performance and stability robustness is described by singular value principles. An assessment of Linear-Quadratic-Gaussian with Loop Transfer Recovery design is carried out after obtaining its design.

Chapter 2

Introduction

2.1 Background

In recent years, the development of heavy lift helicopters has plateaued. The high costs that come with such developments are unfavourable, specially when the options available are economically viable. Twin-Lift is one such option where existing helicopters can be utilized to undertake missions which would otherwise be unfeasible.

2.2 Project Outline

Twin-Lift Helicopter System contains inherent coupling which prevents single input-single output (SISO) design methods to develop a centralized AFCS.
First of all, the TLHS model description needs to be studied.The output variables (which is commanded by the pilot) are chosen. Secondly, the control problem is designed by elaborately listing out the AFCS structure and specifying performance and robustness criteria. Next step involves the description of the LQG/LTR algorithm. The resulting AFCS is evaluated by discussing the behaviour of final loop, sensitivity, and closed loop transfer function matrix singular values. In addition, the transient response characteristics of the system to typical reference commands of the pilot are also examined.

Chapter 3

Modelling of System

3.1 System Description

For this project due to time constraints, only longitudinal configurations are considered.

This configuration consists of two helicopters, two tethers, a spreader bar, two load cables. and a payload. It is assumed that the system motion is restricted to the vertical plane and hence only vertical translation, horizontal translation, and pitching is possible.

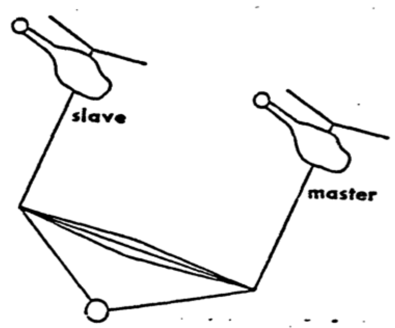


Figure 3.1: TLHS Longitudinal Configurations

This project follows the concept in [1,2] The helicopters being modeled are Sikorsky UH-60A which are taken to be identical. The lead helicopter is referred to as the master helicopter, and the trail helicopter as the slave. Each slave tether is assumed to have a fixed length, H. The helicopter-tether attachment points are assumed to lie at a fixed distance, h, below the center of gravities of the helicopters. Each tether is connected to one end of the spreader bar which is assumed to be rigid and have a fixed length, L. The payload is suspended a fixed distance, Z, below the spreader bar center of gravity via two fixed length cables. For simplicity the tethers, as well as the load cabies, are assumed to have no compliance and to always be in tension.

3.2 Variable Definition and State Space Model

A seven degree of freedom linear model $[1, 3]$ is used to characterize the rigid body dynamics of the system. [4]The key degrees of freedom are described as follows: three degrees of freedom per helicopter (vertical translation, horizontal translation, and pitching), and one degree of freedom for the load which is modeled as a point mass (constrained pendular motion), Since this study restricts the system motion to the longitudinal plane, only four controls (two per helicopter) are relevant. These are the collective controls and the cyclic controls. The collectives control the up/down motion of the helicopters. The cyclic control their fore/aft motion as well as their pitching motion. To fully exploit the symmetry of the 'equal tether configuration' being addressed, the following seven degrees of freedom [3] and four controls are defined (Fig. 3.2):

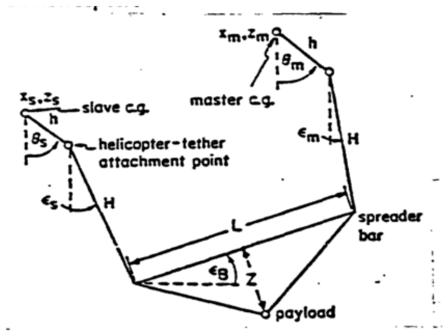


Figure 3.2: TLHS Model for Longitudinal Dynamics

Variables	Description	Significance
Σx	$(\mathrm{x}_m + x_s)/2$	Average horizontal helicopter coordinates
Σz	$(\mathrm{z}_m + z_s)/2$	Average vertical helicopter coordinates
$\Sigma\theta$	$(\theta_m + \theta_s)/2$	Average helicopter pitch attitude
$\Sigma\beta_{lc}$	$(\beta_{lcm} + \beta_{lcs})/2$	Average cyclic control
$\Sigma\theta_c$	$(\theta_{cm} + \theta_{cs})/2$	Average collective control

Table 3.1: Average Variables

Variables	Description	Significance
Δx	$\mathrm{x}_m - x_s$	Horizontal Separation
Δz	$\mathrm{z}_m - z_s)$	Vertical Separation
$\Delta\theta$	$\theta_m - \theta_s)$	Differential pitch attitude
$\Delta\beta_{lc}$	$\beta_{lcm} - \beta_{lcs}$	Differential cyclic control
$\Delta\theta_c$	$\theta_{cm} - \theta_{cs}$	Differential collective control

Table 3.2: Difference Variables

where $|\beta_{lc}| < 15$ degrees and $|\theta_c| < 10$ degrees for each helicopter. The variables defined above are used to derive the equations which govern the planar motion of the TLHS. [5] In doing so it is assumed that the time condition is one in which the helicopters are near hover with the tethers-vertical and the spreader bar horizontal. It is also assumed·that the helicopter c.g's lie the main rotor driveshafts. This assumption forces the helicopters to have no nominal pitch for the trim condition described above. With these assumptions a linear state space model is constructed having the following form:

$$\dot{x}_p = A_p x_p + B_p u_p : x_p \,\epsilon\, R^{12}, u_p \,\epsilon\, R^4 \qquad (3.1)$$

where,

$$x_p = \begin{bmatrix} \dot{\Sigma}z & & & \\ \Delta x & \Delta\dot{\theta} & \Delta\dot{x} & \Delta\theta \\ \Sigma\theta & \Delta z & \Sigma\dot{\epsilon} & \Sigma\dot{x} \\ \Sigma\dot{\theta} & \Delta\dot{z} & \Sigma\dot{\epsilon} & \end{bmatrix}^T \tag{3.2}$$

$$u_p = \begin{bmatrix} \Sigma\theta_c \\ \Delta\beta_{lc} \\ \Delta\theta_c \\ \Sigma\beta_{lc} \end{bmatrix}^T \tag{3.3}$$

$$A_p = diag(A_{p1}, A_{p2}, A_{p3}) \tag{3.4}$$

$$B_p = diag(B_{p1}, B_{p2}, B_{p3}) \tag{3.5}$$

Note that due to the symmetry of the 'equal tether configuration', the model decouples into three basic subsystems to be discussed subsequently. It should also be noted that the components of x_p represent small perturbations from nominal trim values; i.e. the values when the tethers are vertical and the spreader bar horizontal.

3.3 Three Basic Subsystems and Output Selection

The three basic subsystems of the TLHS are now discussed, and variables to be commanded by the pilot (outputs) are selected.

3.3.1 Average Vertical Motion

The pair (A_{p1}, B_{p1}) describes the system's 'average vertical motion. This motion involves the . degree of freedom, and is controlled by issuing average collective commands ($\Sigma\dot{z}$). The subsystem thus has one input which can be used to control at most one output, Σ. This SISO first order subsystem is stable and characterizes the natural damping that occurs during vertical climbs.

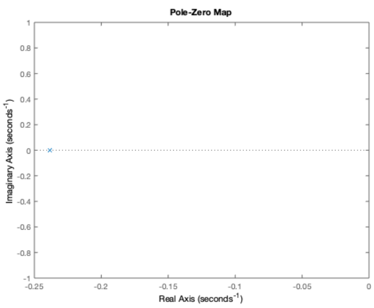


Figure 3.3: Pole-Zero map for average vertical motion

3.3.2 Symmetric Motion

The pair (A_{p2}, B_{p2}) describes the, system's they are fundamentally related to the problem of controlling master/slave workload during horizontal flight. This motion involves the Δx and $\Delta\theta$ degrees of freedom, and is controlled by issuing differential cyclic commands (β_{lc}). The subsystem thus has only one input which can be used to control at most one output. The variable Δx is selected as the output because maintaining horizontal separation is critical. This SISO fourth order subsystem is unstable and is characteristic of any helicopter tethered to a fixed point in space. The instability is referred to as the tethered helicopter mode (Fig. 3.4). This mode is a result of the tethers being attached a distance h below the helicopter cg.'s.

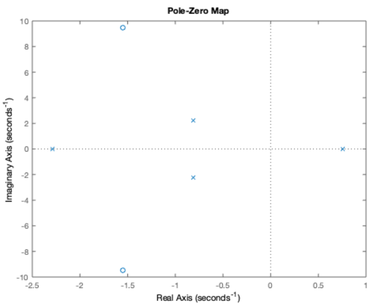


Figure 3.4: Pole-Zero map for symmetric motion

3.3.3 Anti-Symmetric Motion

Finally. the pair $(A_{p3}.B_{p3})$ describes the system's 'anti-symmetric motion' (ASM). This motion involves the $\Sigma\dot{x}$, Σz, Δz, $\Sigma\epsilon$ degrees of freedom, and is controlled by issuing differential collective commands ($\Delta\theta_c$) and average cyclic commands. The subsystem therefore has 2 inputs which can be used to control at most two outputs. The variables $x_L - \Sigma x$ and $\Sigma\dot{x}$ are selected as the outputs because they are fundamentally related to the problem of controlling master/slave workload during horizontal flight. The quantity $x_L - \Sigma x$ represents the load deviation from the center and is given by:

$$x_L - \Sigma x = h\Sigma\theta + (Z/L)\Delta z + H\Sigma\epsilon(16) \tag{3.6}$$

where h, Z. L, and H are constant geometric parameters (Fig. 2). This two-input two-output (TITO) seventh order subsystem is unstable, and if Z = 0, is characteristic of any single hovering helicopter carrying a sling load. The instability here is referred to as the 'backflapping mode' (Fig. 3.5) since it is due to the backflapping of the helicopter main rotors with forward motion [3].

The outputs described above are written in matrix form as follows:

$$y_p = C_p x_p : y_p \,\epsilon\, R^4 \tag{3.7}$$

where,

$$y_p = \begin{bmatrix} \Sigma\dot{z} & \\ \Delta x & \\ x_L - \Sigma x & \Sigma\dot{x} \end{bmatrix}^T \tag{3.8}$$

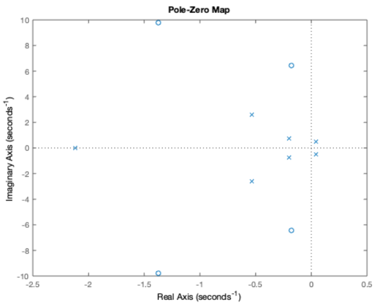


Figure 3.5: Pole-Zero map for anti-symmetric motion

$$C_p = diag(C_{p1}, C_{p2}, C_{p3}) \tag{3.9}$$

Given the above state space model (A_p, B_p, C_p). the (plant) transfer function matrix is then defined as follows:

$$G_p = C_p(sI - A_p)^{-1}B_p = diag(G_{p1}(s), G_{p2}(s), G_{p3}(s)) \tag{3.10}$$

where the $(G_{pi}(s)$ i=1,2,3 specify the input/output characteristics of the three subsystems discussed above, and are given by:

$$G_{pi} = C_{pi}(sI - A_{pi})^{-1}B_{pi} \tag{3.11}$$

It is emphasized at this point that the above decoupling into two 51SO systems and one TITO system occurs because the helicopters are assumed to be Identical and the tethers are assumed to have the same length. The ”unequal tether problem” which the SM and ASM couple to form a three-input three-output subsystem, is addressed in[4].

It is strongly emphasized that although the equations are decoupled in terms of the defined variables, they remain coupled in certain quantities such as the individual helicopter collective and cyclic controls. Consequently all three subsystems and their corresponding compensators must be evaluated simultaneously so that important quantities can be ’untangled’. This coupling. in effect, makes controlling the longitudinal twin lift ’equal tether configuration’ a four-input four-output design problem.

Chapter 4

AFCS Structure and Design Specifications

4.1 AFCS Structure and Function Definition

[2] The final control system structure is shown in Fig 4.1. In this MIMO negative

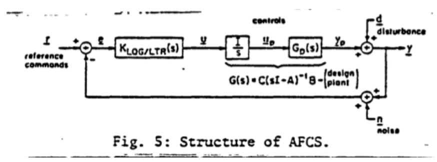


Figure 4.1: AFCS Structure

feedback structure there is the plant $G_p(s)$ (TLHS), a dynamic augmentation consisting of four integrators (one per channel), and a dynamic 16^{th} order compensator $K_{LQG/LTR}(S)$.

The compensator is designed so that the closed loop system meets the specifications-presented below. Since there are three subsystems. the LQG/LTR compensator has the form:

$$K_{LQG/LTR}(s) = diag(K_1(s), K_2(s), K_3(s)) \tag{4.1}$$

where the Ki(s) denote the individual subsystem compensators . The TLHS (plant) plus the integrators is referred to as the design plant. Its transfer function matrix is defined as follows:

$$G(s) = C(sI - A)^{-1}B = diag(G_1(s), G_2(s), G_3(s)) \tag{4.2}$$
$$= G_p(s)/s \tag{4.3}$$

where,

$$A = diag(A_1(s), A_2(s), A_3(s)) \tag{4.4}$$
$$B = diag(B_1(s), B_2(s), B_3(s)) \tag{4.5}$$
$$C = diag(C_1(s), C_2(s), C_3(s)) \tag{4.6}$$

and the i^{th} design plant input/output characteristic is given by:

$$G(s) = C_i(sI - A_i)^{-1}B_i \tag{4.7}$$
$$= G_{pi}(s)/s \tag{4.8}$$

where i = 1,2,3 denote the AVM,SM,ASM subsystems respectively.

Similarly, the final loop,sensitivity and closed loop transfer function can be represented as follows:

$$G_L(s) = diag(G_{L1}(s), G_{L2}(s), G_{L3}(s)) \tag{4.9}$$
$$S_F(s) = diag(S_{F1}(s), S_{F2}(s), S_{F3}(s)) \tag{4.10}$$
$$T_F(s) = diag(T_{F1}(s), T_{F2}(s), T_{F3}(s)) \tag{4.11}$$

4.2 Performance Specifications

To develop reasonable performance specifications, the plant singular values (Fig. 6) were analyzed and closed loop transient responses were studied to study typical maneuvers.

One performance specification is zero steady state error to pilot step reference commands, r, for $\Sigma\dot{z}$, Δx, $x_L - \Sigma x$ and $\Sigma\dot{x}$. Another is complete rejection of step disturbances, d. with respect to the above four output variables. The four integrators introduced in the AFCS (one per channel) guarantee that both of these performance specifications are met (Internal Model Principle). The integrators also help with low frequency command following, low frequency disturbance rejection, and low frequency sensitivity. To guarantee that these properties are built into the design, it is required that the final subsystem sensitivity functions satisfy the following frequency domain constraints:

$$S_{F1}(j\omega) =< -20db\ for\ all\ \omega < 0.08rad/sec \tag{4.12}$$
$$S_{F2}(j\omega) =< -20db\ for\ all\ \omega < 0.04rad/sec \tag{4.13}$$
$$\sigma_{max}S_{F3}(j\omega) =< -20db\ for\ all\ \omega < 0.06rad/sec \tag{4.14}$$

where σ_{max} denotes the maximum singular value

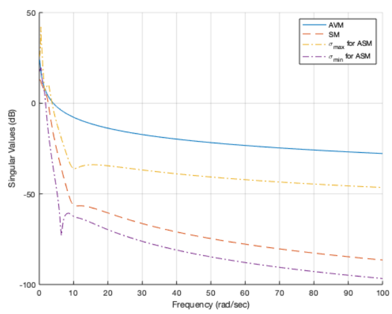


Figure 4.2: Plant Singular Values

4.3 Robustness Specifications

To be robust with respect to the high frequency unmodeled main rotor dynamics (at ω - 27 rad/sec). the final subsystem closed loops must satisfy the following frequency domain constraints:

$$T_{F1}(j\omega) =< -20db\ for\ all\ \omega > 10rad/sec \tag{4.15}$$

$$T_{F2}(j\omega) =< -40db\ for\ all\ \omega > 10rad/sec \tag{4.16}$$

$$\sigma_{max}T_{F3}(j\omega) =< -20db\ for\ all\ \omega > 10rad/sec \tag{4.17}$$

These also help in attenuating high frequency sensor noise, The above performance and stability robustness specifications are nonsensical unless the. nominal closed loop system is stable . This however, is not a problem since the LQG/LTR design methodology will be used. The methodology guarantees nominal stability and offers excellent stability margins.

Chapter 5

LQG/LTR Control Design

5.1 Obtaining the design

Next the LQG/LTR design methodology is applied to the minimum phase design plant $G(s) = C(sI - A)^{-1}B$.The design process is completed in two steps.

5.1.1 Step 1

Design a 'nice target loop', $G_{KF}(s) = C(sl - A)^{-1}H$,where H is found by solving a filter algebraic Riccati equation. The Riccati equation depends on design parameters that can be manipulated to place loop transmission zeros at appropriate locations to o give us loop shapes that satisfy the aforementioned specifications. $G_{KF}(s)$ has guaranteed closed loop robustness properties.

5.1.2 Step 2

'Recover' the nice loop properties of GKF(s) from the design plant:

$$LTR : G(s)K_{LQG/LTR}(s) \longrightarrow G_{KF}(s) \;\; as \;\; \rho_1, \rho_2, \rho_3 \longrightarrow 0 \tag{5.1}$$

[6] Eq.(5.1) shows that the as the design parameter ρ (one for each subsystem) get small the loop transfer function matrix approaches a function with good performance and guaranteed robustness properties.

5.2 Controller Design

The controller K is determined using :

$$K(s) = G(sI - A + BG + HC)^{-1}H \tag{5.2}$$

Chapter 6

Results

6.1 Average Vertical Motion(For $\Sigma\dot{z} = 5ft/sec$)

$$A_{avm} = -0.2384$$
$$B_{avm} = 4.0989$$
$$C_{avm} = 1$$
$$D_{avm} = 0$$

Using LQG/LTR design for compensator :

$$A_1 = \begin{bmatrix} 0 & 0 \\ 4.0989 & -0.2384 \end{bmatrix} \; B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \; C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix} \; D_1 = 0$$

$$G_1 = 10^6 \begin{bmatrix} 0.0051 & 3.1620 \end{bmatrix} \; H_1 = \begin{bmatrix} 0.1839 & 3.1623 \end{bmatrix}^T$$

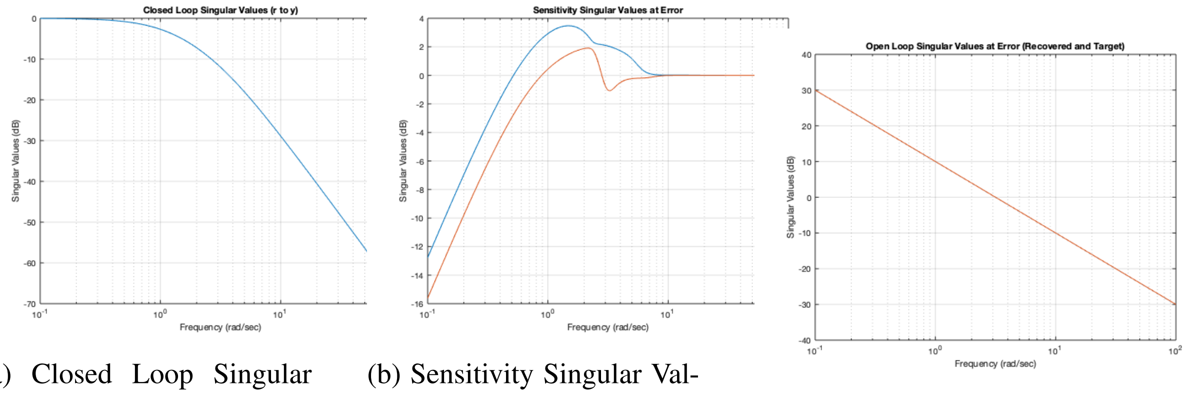
(a) Time v/s Output

(b) Time v/s Control

Figure 6.1: Average Vertical Motion of TLHS

6.2 Symmetric Motion(For Δx = 2ft)

$$A_{sm} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1.0974 & -0.8847 & -0.0600 & 0 \\ -17.2659 & -5.0777 & 2.3491 & -3.1 \end{bmatrix} \; B_{sm} = \begin{bmatrix} 0 \\ 0 \\ 0.4782 \\ -47.24 \end{bmatrix}$$



(a) Closed Loop Singular Values

(b) Sensitivity Singular Values

(c) Loop Singular Values

Figure 6.2: Singular Values of Average Vertical Motion of TLHS

$$C_{sm} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \; D_{sm} = 0$$

Using LQG/LTR design for compensator :

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.4782 & -1.0974 & -0.8847 & -0.0600 & 0 \\ -47.24 & -17.2659 & -5.0777 & 2.3491 & -3.1 \end{bmatrix} \; B_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \; D_2 = 0$$

$$G_2 = 10^6 \begin{bmatrix} 0.0002 & 3.1617 & -0.0004 & 0.0551 & 0 \end{bmatrix}$$
$$H_2 = \begin{bmatrix} 0.6939 & 4.6745 & -10.9 & 5.92 & -11.48 \end{bmatrix}^T$$

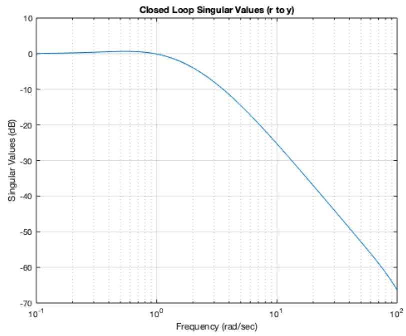
(a) Time v/s Output

(b) Time v/s Control

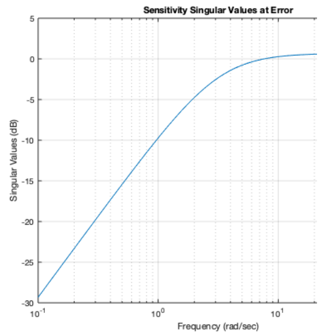
Figure 6.3: Symmetric Motion of TLHS

6.3 Anti-Symmetric Motion(x_L - Σx = -1ft $\Sigma \dot{x}$ = 5ft/sec)

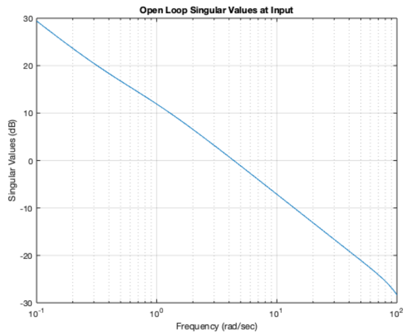
Using LQG/LTR design for compensator :



(a) Closed Loop Singular Values



(b) Sensitivity Singular Values



(c) Open Loop Singular Values

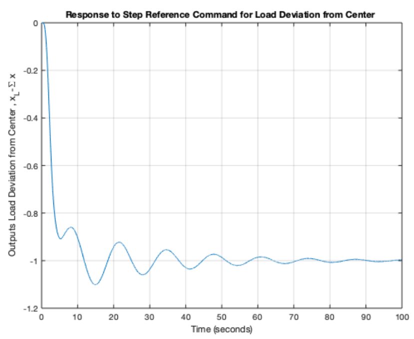
Figure 6.4: Singular Values for Symmetric Motion of TLHS

```
>> A_asm

A_asm =

         0         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000         0
         0         0         0         0         0         0    1.0000
   -0.5620         0    1.0974   -0.0600         0         0         0
         0         0   17.2659    2.3491   -3.1000         0         0
    0.4679   -0.3885    2.0233         0         0   -0.3361         0
   -0.2220    0.1844   -9.5654   -0.6309    0.9117    0.1595         0

>> B_asm

B_asm =

         0         0
         0         0
         0         0
         0    0.4782
         0  -47.2400
    5.7794         0
   -2.7425   13.4145

>> C_asm

C_asm =

    0.2941    0.5000    1.0000         0         0         0         0
         0         0         0    1.0000         0         0         0

>> D_asm

D_asm =

     0     0
     0     0
```

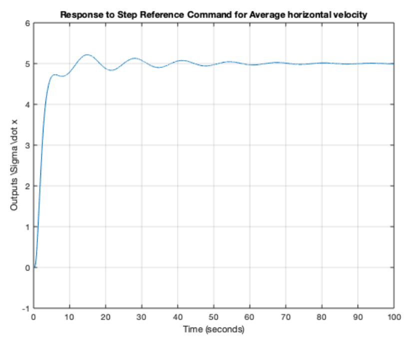
Figure 6.5: Plant State-Space

```
>> G3

G3 =

    7.7069   -0.8296    0.8620   12.8664   -0.6333   16.8506    5.3501   13.5835   17.6708
   -0.8296   12.5241  -10.0402   -8.3617   -1.2326   25.7336   -3.3712   -4.8653   -6.9172

>> H3

H3 =

    0.1095    0.0078
   -0.0030    0.0434
    0.1541   -0.1527
    1.7305   -0.0571
    0.0036    0.0021
   -0.0714    0.8797
    0.0222    0.0076
    0.1523   -0.0957
    0.0044   -0.0026
```

Figure 6.6: G and H for Anti-Symmetric Plant



(a) Time v/s Load Deviation from Center



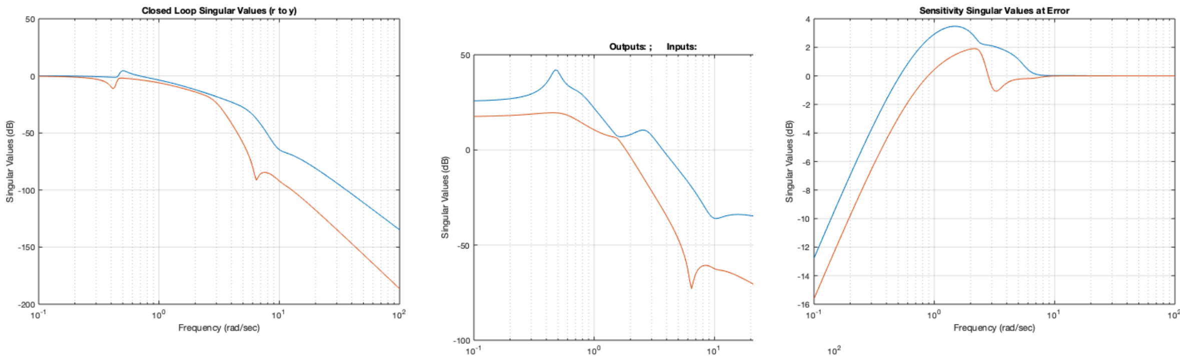
(b) Time v/s Average Horizontal velocity

Figure 6.7: Time v/s Output

(a) Time v/s Differential Cyclic Control

(b) Time v/s Average Cyclic Control

Figure 6.8: Time v/s Control



(a) Closed Loop Singular Values

(b) Loop Singular Values

(c) Sensitivity Singular Values

Figure 6.9: Singular Values

All sensitivity singular values lie below the -20 db 'Performance region' thus guaranteeing good low frequency command following, good low frequency disturbance rejection, and good low frequency sensitivity reduction. The poor SM and ASM sensitivity properties at frequencies above 0.6 rad/sec are expected since these subsystems have unstable modes which force the tradeoff of nice feedback properties at high frequencies for nice feedback properties at low frequencies.

All closed loop singular values lie below the -20 db 'robustness region' thus guaranteeing robustness to the unmodeled rotor dynamics at ω : 27 rad/sec. The design also provides a degree of high frequency sensor noise attenuation built into the design.

6.4 Conclusions

Comparing figures 6.1a 6.3a, it is seen that the vertical acceleration characteristics are slightly worse than the horizontal acceleration characteristics. This is primarily due to the fact that although Δz does appear in the expression for $x_L - \Sigma x$, it is not methodology can be used to address the TLHS control directly 'penalized'; i.e. it is not an output as is its horizontal counterpart, Δx. It should be noted that the four integrators (one per input channel) have provided the desired zero steady state error to the pilot's step reference commands. It should also be noted that because of the unstable 'tethered helicopter' and 'backflapping' modes, a minimum bandwidth is required to just stabilize the symmetric and anti-symmetric motions. This, of course. translates into minimum control rates and more importantly (from a passenger's viewpoint), larger pitch rates. These large pitch rates must be lived with, and are linear functions of the pilot reference commands as well as the system initial conditions.

The LQG/LTR design methodology, coupled with singular value ideas, can be used to formulate performance and stability robustness specifications which can then be used to develop a centralized MIMO AFCS for a twin lift helicopter system (TLHS). The methodology is extremely powerful in that it possesses enough degrees of freedom to tailor designs to prescibed pilot specifications. The methodology can be used to address the TLHS control directly with respect to the following five fundamental feedback issues:

1. Low frequency disturbance rejection
2. low frewuency command following
3. low frequency sensitivity reduction
4. robustness to high frequency modelling
5. high frequency noise attenuation

Bibliography

[1] A. A. Rodriguez and M. Athans, “Multivariable control of a twin lift helicopter system using the lqg/ltr design methodology,” 1986.

[2] A. Rodriguez, “Multivariable control of a twin lift helicopter system using the lqg/ltr design methodology,” 1986.

[3] H. Curtiss and F. Warburton., “Stability and control of the twin lift helicopter system,” *40th Annual forum of Arlington, Va.*, 1984.

[4] A. Bramwell, “Helicopter dynamics,” vol. 125, pp. 134–154, 2019.

[5] L. Kaufman and E. Schultz, “The stability and control of tethered helicopters’,” *JAHS*, vol. 7, no. 4, 1962.

[6] G. Stein and 1.Athans, “The lqg/ltr procedure for multivariable feedback control design,” *MIT LIDS-P-1384,*, 1984.

-----------------------------LoadData.m---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  loaddata.m                                                             %
%  Routine to load the matlab workspace with the appropriate parameters   %
%  (longitudinal dynamics)                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;clear;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Definition of physical parameters for the Twin Lift Helicopter System %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
% Acceleration due to gravity
g=32.2;

%%
%Helicopter Parameters
W_h=14000;
I_y=5700;
hp=3.6;

%%
%Control Derivatives
X_thc=0;
X_blc=27.4;
Z_thc=340.9;
Z_blc=0;
M_thc=0;
M_blc=-47.24;

%%
%Aerodynamic derivatives at hover
X_u=-.06;
X_w=0;
X_q=0;
Z_u=0;
Z_w=-.346;
Z_q=0;
M_u=.041;
M_w=0;
M_q=-3.1;

%%
%Tether lengths
H_m=13.25;
x=1;                     % tether length factor for slave
H_s=x*H_m;

%%
%%Spreader bar parameters
L=69;
W_b=644;
Z=34.5;
W_l=12000;
%
%end
```

```
--------------------------makeplant.m-------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  makeplnt.m                                                              %
%  Routine to generate the A,B,C,D matrices for Equal Tether Twin Lift    %
%                                                                          %
%   NOTE: Run loaddata.m prior to running this script                       %
%                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Definition of variables for the Twin Lift Helicopter System           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
clc;
disp('')
disp('... Calculating TLHS Data ...')
disp('')

%%
%% Mass and Moment Calculations
M_h=W_h/g;
M_b=W_b/g;
I_b=M_b*L*L/12;
M_l=W_l/g;

%%
%% Tether Length Parameters
H_a=2*H_s*H_m/(H_s + H_m);
S=(H_m - H_s)/(H_m + H_s);
w_a_sq=g/H_a;

%%
%Normalized Quantities
H_a_hat=H_a/L;
Z_hat=Z/L;

%%
%Mass Parameters
mew=(W_l + W_b)/(2*W_h);
delta_l=W_l/(W_l +W_b);
epsl=(M_h*hp)/I_y;
ep_b=2*I_b/(M_h*L*L);

%%
%Definitons
psi=1 + ep_b + 4*mew*Z_hat*Z_hat*delta_l*(1 - delta_l);
J=inv(mew*delta_l*Z_hat*w_a_sq);
T=inv(J*psi);
D=-w_a_sq*(1 + mew + (hp + H_s)*mew*epsl + (4*T*delta_l*Z_hat/w_a_sq));
E=-(X_u + M_u*(hp + H_s));
F=(hp + H_s)*epsl*mew*w_a_sq*S*H_s - w_a_sq*H_s - 4*T*delta_l*Z_hat*H_s;
V=1 + (hp + H_s)*epsl + (1/mew) + (4*delta_l*delta_l*Z_hat*Z_hat/psi);

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Definition of the Ap, Bp, Cp, and Dp Matrices used to     %
%   describe the Plant of the Twin Lift Helicopter System.    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Ap Matrix
```

```
Ap(1,1)=Z_w/(1 + mew);
Ap(2,4)=1;
Ap(3,5)=1;
Ap(4,2)=-mew*w_a_sq;
Ap(4,3)=-(g*(1 + mew) +mew*w_a_sq*hp);
Ap(4,4)=X_u;
Ap(4,6)=-2*mew*w_a_sq*S*H_s;
Ap(4,8)=-2*mew*w_a_sq*S;
Ap(5,2)=-epsl*mew*w_a_sq;
Ap(5,3)=-epsl*mew*w_a_sq*(hp + H_a);
Ap(5,4)=M_u;
Ap(5,5)=M_q;
Ap(5,6)=-2*epsl*mew*w_a_sq*S*H_s;
Ap(5,8)=-2*epsl*mew*w_a_sq*S;
Ap(6,10)=1;
Ap(7,11)=1;
Ap(8,12)=1;
Ap(9,2)=mew*w_a_sq*S/2;
Ap(9,3)=mew*w_a_sq*S*hp/2;
Ap(9,6)=-g*(1 + mew*S*H_s/H_a);
Ap(9,8)=mew*w_a_sq;
Ap(9,9)=X_u;
Ap(10,2)=mew*epsl*w_a_sq*S/2;
Ap(10,3)=mew*epsl*w_a_sq*S*hp/2;
Ap(10,6)=-epsl*mew*w_a_sq*S*H_s;
Ap(10,8)=epsl*mew*w_a_sq;
Ap(10,9)=M_u;
Ap(10,10)=M_q;
Ap(11,2)=2*T*S;
Ap(11,3)=2*T*S*hp;
Ap(11,6)=4*T*H_s;
Ap(11,7)=-4*T*H_a_hat;
Ap(11,8)=4*T;
Ap(11,11)=Z_w*T*J;
Ap(12,2)=-(mew*w_a_sq*S/2)*V;
Ap(12,3)=-(mew*w_a_sq*S/2)*V*hp;
Ap(12,6)=g*(1 + mew*S*H_s/H_a) + F;
Ap(12,7)=4*T*delta_l*Z_hat*H_a_hat;
Ap(12,8)=D;
Ap(12,9)=E;
Ap(12,10)=-M_q*(hp + H_s);
Ap(12,11)=-Z_w*T*J*delta_l*Z_hat;

%%Bp Matrix
Bp(1,1)=Z_thc/(1 + mew);
Bp(4,2)=X_blc;
Bp(5,2)=M_blc;
Bp(9,4)=X_blc;
Bp(10,4)=M_blc;
Bp(11,3)=Z_thc*T*J;
Bp(12,3)=-Z_thc*T*J*delta_l*Z_hat;
Bp(12,4)=-(X_blc + M_blc*(hp + H_s));

%%Cp Matrix
Cp(1,1)=1;
Cp(2,2)=1;
Cp(3,6)=(hp + H_s);
Cp(3,7)= Z_hat;
Cp(3,8)=1;
Cp(4,9)=1;
Cp(:,10:12)=zeros(4,3);
```

```
%%Dp Matrix
Dp=zeros(4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Convert Ap, Bp, Cp, and Dp from units of radians per        %
%  second to units of degrees.                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cf=180/pi;
v1=[1,1,cf,1,cf,cf,1,1,1,cf,1,1];
con_A=diag(v1,0);
con_A_inv=inv(con_A);
v2=[cf,cf,cf,cf];
con_B=diag(v2,0);
con_B_inv=inv(con_B);
Ap=(con_A)*(Ap)*(con_A_inv);
Bp=(con_A)*(Bp)*(con_B_inv);
Cp=(Cp)*(con_A_inv);
Dp=(Dp)*(con_B_inv);

%%
%Clean Up
clear cf v1 v2 con_A con_A_inv con_B con_B_inv

%%
%% System poles and zeros
poles_sys=eig(Ap);
zeros_sys=tzero(Ap,Bp,Cp,Dp);

-----------------------------AVM.m----------------------------------
clc;
%% Average vertical motion plant
A_avm=Ap(1,1);
B_avm=Bp(1,1);
C_avm=Cp(1,1);
D_avm=Dp(1,1);
poles_avm=eig(A_avm);
zeros_avm=tzero(A_avm,B_avm,C_avm,D_avm);
avm_sys = ss(A_avm,B_avm,C_avm,D_avm);
figure()
pzmap(avm_sys)
%% Plant Dimensions
[ns_avm,nc_avm] = size(B_avm);
no_avm = nc_avm;
%% Eigenvectors and Eigenvalues
[evec,eval] = eig(A_avm);
%% Modal Analysis
%
tinit = 0;
tinc  = 0.001;
tfin  = 0.2;
t     = [tinit:tinc:tfin];   % Vector of uniformly spaced time points
u     = [0*t'];         % Set input u to zero for all time in order to
generate zero input response;
                              % i.e. response to an initial condition x_o.

x = lsim(ss(A_avm, B_avm, eye(ns_avm,ns_avm), 0*ones(ns_avm,nc_avm)), u, t,
evec);
figure; plot(t,x)
grid
title('Fast Instability')
ylabel('States')
```

```
xlabel('Time (seconds)')
pause
% Excite Slow Instability.
x = lsim(ss(A_avm,B_avm , eye(ns_avm,ns_avm), 0*ones(ns_avm,nc_avm)), u, t,
real(evec));
figure; plot(t,x)
grid
title('Slow Instability:')
ylabel('States (deg, deg/sec)')
xlabel('Time (seconds)')
%% Transmission Zeros
plantzeros = tzero(ss(A_avm,B_avm,C_avm,D_avm))       % transmission zeros
%% Plant Transfer Funtion from u to x
Plant_zpk = zpk(ss(A_avm,B_avm,C_avm,D_avm));
%% Controllability
rank(ctrb(A_avm,B_avm))
%% Observability
rank(obsv(A_avm,C_avm))
%% Frequency Response
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A_avm, B_avm, C_avm, D_avm),w);
sv     = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Outputs: \Sigma dz(ft/sec) ; Inputs: \Sigma \theta_{c}')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Pland SVD analysis at DC
dc            =  C_avm*inv(-A_avm)*B_avm
[udc,sdc,vdc] = svd(dc)

%% Designing an LQG/LTR Compensator for average vertical motion
A1 = [zeros(nc_avm,nc_avm+ns_avm);B_avm A_avm];
B1 = [eye(nc_avm); zeros(nc_avm)];
C1 = [zeros(nc_avm) C_avm];
D1 = [zeros(no_avm,nc_avm)];
Ll1 = inv(C_avm*inv((-A_avm))*B_avm);
Lh1 = inv(-A_avm)*B_avm*Ll1;
Llqg1 = [Ll1;Lh1];
mu = 0.1;
sigma1 = are(A1',(C1'*inv(mu)*C1),(Llqg1*Llqg1'));
H1 = sigma1*C1'*inv(mu);
%% Target Closed loop poles and zeros
tclpoles = eig(A1-H1*C1);
tclzeros = tzero(A1-H1*C1,H1,C1, D1);
%% Target Open Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A1,H1,C1,D1),w);
tlsv   = 20*log10(sv);
semilogx(w, tlsv)
title('Target Open Loop Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
```

```
ylabel('Singular Values (dB)')
pause
%% Target Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A1-H1*C1,H1,-C1,eye(no_avm,no_avm)),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Sensitivity Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Target Complementary Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A1-H1*C1,H1,C1,D1),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Comp Sensitivity Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Designing Pre-Filter
z = 1.2;
fil1    = ss((tf([z],[1 z])));
s                    = j*20;
tol_s                = C1*inv(s*eye(size(A1))-A1)*H1 + D1;
[tolu, tols, tolv ] = svd(tol_s);
%% Target Closed Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
[ntacl1, ntbcl1, ntccl1, ntdcl1 ] = series(fil1.a, fil1.b, fil1.c, fil1.d,
A1-H1*C1,H1,C1,D1);
sv     = sigma(ss(ntacl1, ntbcl1, ntccl1, ntdcl1),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('T_{ry}, Singular Values (dB)')
pause
%% Target Step Responses
tinit = 0;
tinc  = 0.005;
tfin  = 4.0;
t     = [tinit:tinc:tfin]';                       % Vector of uniformly spaced
time points
r1    = 5*[ones(size(t))];
ty1   = lsim(ss(ntacl1, ntbcl1, ntccl1, ntdcl1), r1,t);

plot(t,ty1)
grid
```

```
title('Target Responses to Step Reference Command')
ylabel('(degrees)')
xlabel('Time (seconds)')
pause
%% LTR at output
rho1 = 1e-13;
R1 = rho1*eye(1);
[G1,X1,poles1] = lqr(A1,B1,C1'*C1,R1);
%% Model Based Compensator followed by integrator bank
Ak1 = [0*ones(nc_avm,no_avm)  G1; zeros(ns_avm+nc_avm,no_avm)  A1-B1*G1-
H1*C1];
Bk1 = [zeros(nc_avm,no_avm); H1];
Ck1 = [eye(nc_avm,no_avm)  zeros(nc_avm, ns_avm+nc_avm)];
Dk1 = [zeros(no_avm,no_avm)];
kpoles = eig(Ak1)                               % Compensator Poles
kzeros = tzero(Ak1, Bk1, Ck1, Dk1)                 % Compensator Zeros

winit  = -2;
wfin   = 4;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);
sv     = sigma(ss(Ak1, Bk1, Ck1, Dk1),w);
sv     = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Compensator Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Open Loop System
[al1, bl1, cl1, dl1 ] = series(Ak1, Bk1, Ck1, Dk1, A_avm, B_avm, C_avm,
D_avm);

olpoles = eig(al1)                          % Open Loop Poles
olzeros = tzero(al1,bl1,cl1,dl1)               % Open Loop Zeros

winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(al1, bl1, cl1, dl1), w);
sv      = 20*log10(sv);
figure; semilogx(w, sv, w, tlsv)
%clear sv
title('Open Loop Singular Values at Error (Recovered and Target)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause


figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
[ali1, bli1, cli1, dli1 ] = series(A_avm, B_avm, C_avm, D_avm, Ak1, Bk1,
Ck1, Dk1);
```

```
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(ali1, bli1, cli1, dli1 ), w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Closed Loop System
acl1     = al1-bl1*cl1;
bcl1     = bl1;
ccl1     = cl1;
dcl1     = dl1;

clpoles = eig(acl1)                % Closed Loop Poles
damp(clpoles)
clzeros = tzero(acl1,bcl1,ccl1,dcl1) % Closed Loop Zeros (r to y)
%% Sensitivity at Error
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl1, bcl1, -ccl1, eye(1)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Sensitivity at Input
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(ali1-bli1*cli1, bli1, -cli1, eye(1)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Complementary Sensitivity
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl1, bcl1, ccl1, dcl1),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Complementary Sensitivity Singular Values at Ouput')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
```

```matlab
pause
%% Add Pre-filter to Closed Loop Transfer Function Matrix From r to y
[nacl1, nbcl1, nccl1, ndcl1 ]   = series(fil1.a, fil1.b, fil1.c, fil1.d,
acl1,bcl1,ccl1,dcl1);
sv                            = sigma(ss(nacl1, nbcl1, nccl1, ndcl1),w);
sv                            = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Reference to Controls (r to u)
[aru1, bru1, cru1, dru1 ] = series(acl1, bcl1, -ccl1, eye(1), Ak1, Bk1,
Ck1, Dk1);
[aru1, bru1, cru1, dru1 ] = series(fil1.a, fil1.b, fil1.c, fil1.d, aru1,
bru1, cru1, dru1 );
winit                    = -1;
wfin                     = 2;
nwpts                    = 200;
w                        = logspace(winit,wfin,nwpts);
sv                       = sigma(ss(aru1, bru1, cru1, dru1),w);
sv                       = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Reference to Control Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Input Disturbance to Output
[ady1, bdy1, cdy1, ddy1 ] = series(A_avm, B_avm, C_avm, D_avm, acl1, bcl1,
-ccl1, eye(1));
winit                    = -2;
wfin                     = 2;
nwpts                    = 200;
w                        = logspace(winit,wfin,nwpts);
sv                       = sigma(ss(ady1, bdy1, cdy1, ddy1),w);
sv                       = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Input Disturbance to Output Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Input Disturbance Analysis at DC
s = j*7
tdy1                 = cdy1*inv(s*eye(size(ady1))-ady1)*bdy1 +  ddy1;
[utdy1 stdy1 vtdy1] = svd(tdy1)
%% Closed Loop Step Responses
y1    = lsim(ss(nacl1,nbcl1, nccl1, ndcl1), r1, t);
figure; plot(t,y1, t, ty1)
plot(t,y1)
grid
title('Average Vertical Velocity')
ylabel('Outputs \Sigma dz(ft/sec)')
xlabel('Time (seconds)')
pause
u1    = lsim(ss(aru1, bru1, cru1, dru1), r1, t);
figure;
plot(t,u1)
```

```
grid
title('Average Collective Control')
ylabel('Controls \Sigma \theta_{c}')
xlabel('Time (seconds)')
pause
```

------------------------SM.m -------------------------------

```
clc;
%% Symmetric motion plant
A_sm=Ap(2:5,2:5);
B_sm=Bp(2:5,2);
C_sm=Cp(2,2:5);
D_sm=Dp(2,2);
sm_sys = ss(A_sm,B_sm,C_sm,D_sm);
figure()
pzmap(sm_sys)
%% Plant Dimensions
[ns_sm,nc_sm] = size(B_sm);
no_sm = nc_sm;
%% Natural Modes: Poles (Eigenvalues), Eigenvectors
[evec,eval] = eig(A_sm)
tinit = 0;
tinc  = 0.001;
tfin  = 0.2;
t     = [tinit:tinc:tfin];   % Vector of uniformly spaced time points
u     = [0*t'];         % Set input u to zero for all time in order to
generate zero input response;
% Excite Fast Instability.
% This mode
%            is associated with a pole at s = 14.1050.
%            is associated primarily with theta_2 dot.
x = lsim(ss(A_sm, B_sm, eye(ns_sm,ns_sm), 0*ones(ns_sm,nc_sm)), u, t,
evec(:,1));
figure; plot(t,x)
grid
title('Fast Instability')
ylabel('States')
xlabel('Time (seconds)')
pause

%
% Excite Slow Instability.
% This mode
%            is associated with a poles at s = 4.5299.
%            is associated primarily with theta_1 and theta_1 dot
%
x = lsim(ss(A_sm, B_sm, eye(ns_sm,ns_sm), 0*ones(ns_sm,nc_sm)), u, t,
real(evec(:,4)));
figure; plot(t,x)
grid
title('Slow Instability')
ylabel('States')
xlabel('Time (seconds)')
pause
%% Transmission Zeros
plantzeros = tzero(ss(A_sm,B_sm,C_sm,D_sm))       % transmission zeros
%% SYSTEM TRANSFER FUNCTIONS: From u_i to x_j
Plant_zpk = zpk(ss(A_sm,B_sm,C_sm,D_sm)) % Zeros, Poles, and Gains fron u_i
to x_j
```

```
%% Controllability
rank(ctrb(A_sm,B_sm))
%% Observability
rank(obsv(A_sm,C_sm))
%% FREQUENCY RESPONSE: Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A_sm, B_sm, C_sm, D_sm),w);
sv     = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Outputs: ;      Inputs: ')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% PLANT SVD ANALYSIS at DC
dc              =  C_sm*inv(-A_sm)*B_sm
[udc,sdc,vdc] = svd(dc)
%% Designing an LQG/LTR Compensator for symmetric motion
[ns_sm,nc_sm] = size(B_sm);
no_sm = nc_sm;
A2 = [zeros(nc_sm) zeros(nc_sm,ns_sm); B_sm A_sm];
B2 = [eye(nc_sm);zeros(ns_sm,1)];
C2 = [zeros(nc_sm) C_sm];
D2 = [zeros(no_sm,nc_sm)];
Ll2 = inv(C_sm*inv((-A_sm))*B_sm);
Lh2 = inv(-A_sm)*B_sm*Ll2;
Llqg2 = [Ll2;Lh2];
mu = 0.1;
B_fare = (C2'*inv(mu)*C2);
sigma2 = are(A2',B_fare,(Llqg2*Llqg2'));
H2= sigma2*C2'*inv(mu);
%% Target Closed loop poles and zeros
tclpoles2 = eig(A2-H2*C2);
tclzeros2 = tzero(A2-H2*C2,H2,C2, D2);
%% Target Open Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A2,H2,C2,D2),w);
tlsv   = 20*log10(sv);
semilogx(w, tlsv)
title('Target Open Loop Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Target Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A2-H2*C2,H2,-C2,eye(no_sm,no_sm)),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Sensitivity Singular Values at Output')
```

```
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Target Complementary Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A2-H2*C2,H2,C2,D2),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Comp Sensitivity Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Designing Pre-Filter
z = 1.2;
fil2   = ss((tf([z],[1 z])));
s                    = j*20;
tol_s                = C2*inv(s*eye(size(A2))-A2)*H2 + D2;
[tolu, tols, tolv ] = svd(tol_s);
%% Target Closed Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
[ntacl2, ntbcl2, ntccl2, ntdcl2 ] = series(fil2.a, fil2.b, fil2.c, fil2.d,
A2-H2*C2,H2,C2,D2);
sv     = sigma(ss(ntacl2, ntbcl2, ntccl2, ntdcl2),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('T_{ry}, Singular Values (dB)')
pause
%% Target Step Responses
tinit = 0;
tinc  = 0.005;
tfin  = 10.0;
t     = [tinit:tinc:tfin]';% Vector of uniformly spaced time points
r2    = 1*[ones(size(t))];
ty1   = lsim(ss(ntacl2, ntbcl2, ntccl2, ntdcl2), r2,t);

plot(t,ty1)
grid
title('Target Responses to Step Reference Command')
ylabel('(degrees)')
xlabel('Time (seconds)')
pause
%% LTR at output
rho2 = 1e-13;
R2 = rho2*eye(1);
[G2,X2,poles2] = lqr(A2,B2,C2'*C2,R2);
%% Model Based Compensator followed by integrator bank
Ak2 = [0*ones(nc_sm,no_sm)  G2; zeros(ns_sm+nc_sm,no_sm)  A2-B2*G2-H2*C2];
Bk2 = [zeros(nc_sm,no_sm); H2];
Ck2 = [eye(nc_sm,no_sm)  zeros(nc_sm, ns_sm+nc_sm)];
Dk2 = [zeros(no_sm,no_sm)];
```

```
kpoles = eig(Ak2)                                    % Compensator Poles
kzeros = tzero(Ak2, Bk2, Ck2, Dk2)                      % Compensator Zeros

winit  = -2;
wfin   = 4;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);
sv     = sigma(ss(Ak2, Bk2, Ck2, Dk2),w);
sv     = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Compensator Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Open Loop System
[al2, bl2, cl2, dl2 ] = series(Ak2, Bk2, Ck2, Dk2, A_sm, B_sm, C_sm, D_sm);

olpoles = eig(al2)                               % Open Loop Poles
olzeros = tzero(al2,bl2,cl2,dl2)                    % Open Loop Zeros

winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(al2, bl2, cl2, dl2), w);
sv      = 20*log10(sv);
figure; semilogx(w, sv, w, tlsv)
%clear sv
title('Open Loop Singular Values at Error (Recovered and Target)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause


figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
[ali2, bli2, cli2, dli2 ] = series(A_sm,B_sm,C_sm,D_sm, Ak2, Bk2, Ck2,
Dk2);

winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(ali2, bli2, cli2, dli2 ), w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Closed Loop System
acl2    = al2-bl2*cl2;
```

```
bcl2     = bl2;
ccl2     = cl2;
dcl2     = dl2;

clpoles = eig(acl2)                % Closed Loop Poles
damp(clpoles)
clzeros = tzero(acl2,bcl2,ccl2,dcl2) % Closed Loop Zeros (r to y)
%% Sensitivity at Error
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl2, bcl2, -ccl2, eye(1)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Sensitivity at Input
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(ali2-bli2*cli2, bli2, -cli2, eye(1)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Complementary Sensitivity
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl2, bcl2, ccl2, dcl2),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Complementary Sensitivity Singular Values at Ouput')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Add Pre-filter to Closed Loop Transfer Function Matrix From r to y
[nacl2, nbcl2, nccl2, ndcl2 ]   = series(fil2.a, fil2.b, fil2.c, fil2.d,
acl2,bcl2,ccl2,dcl2);
sv                            = sigma(ss(nacl2, nbcl2, nccl2, ndcl2),w);
sv                            = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Reference to Controls (r to u)
```

```
[aru2, bru2, cru2, dru2 ] = series(acl2, bcl2, -ccl2, eye(1), Ak2, Bk2,
Ck2, Dk2);
[aru2, bru2, cru2, dru2 ] = series(fil2.a, fil2.b, fil2.c, fil2.d, aru2,
bru2, cru2, dru2 );
winit                = -1;
wfin                 = 2;
nwpts                = 200;
w                    = logspace(winit,wfin,nwpts);
sv                   = sigma(ss(aru2, bru2, cru2, dru2),w);
sv                   = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Reference to Control Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Input Disturbance to Output
[ady2, bdy2, cdy2, ddy2 ] = series(A_sm,B_sm,C_sm,D_sm, acl2, bcl2, -ccl2,
eye(1));
winit                = -2;
wfin                 = 2;
nwpts                = 200;
w                    = logspace(winit,wfin,nwpts);
sv                   = sigma(ss(ady2, bdy2, cdy2, ddy2),w);
sv                   = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Input Disturbance to Output Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Input Disturbance Analysis at DC
s = j*7
tdy2            = cdy2*inv(s*eye(size(ady2))-ady2)*bdy2 +  ddy2;
[utdy1 stdy1 vtdy1] = svd(tdy2)
%% Closed Loop Step Responses
y1    = lsim(ss(nacl2,nbcl2,nccl2, ndcl2), r2, t);
figure;
plot(t,y1)
grid
title('Horizontal Separation')
ylabel('Outputs \Delta x(ft)')
xlabel('Time (seconds)')
pause
u1    = lsim(ss(aru2, bru2, cru2, dru2), r2, t);
figure;
plot(t,u1)
grid
title('Differential Collective Control')
ylabel('Controls \Delta B_{lc} (deg)')
xlabel('Time (seconds)')
pause
```

-----------------------------ASM.m-----------------------------

```
clc;
A_asm=Ap(6:12,6:12);
B_asm=Bp(6:12,3:4);
```

```
C_asm=Cp(3:4,6:12);
D_asm=Dp(3:4,3:4);
poles_asm=eig(A_asm);
zeros_asm=tzero(A_asm,B_asm,C_asm,D_asm);
asm_sys = ss(A_asm,B_asm,C_asm,D_asm);
figure()
pzmap(asm_sys)
%% Plant Dimensions
[ns_asm,nc_asm] = size(B_asm);
no_asm = nc_asm;
%% Natural Modes: Poles (Eigenvalues), Eigenvectors
[evec,eval] = eig(A_asm)
tinit = 0;
tinc  = 0.001;
tfin  = 0.2;
t     = [tinit:tinc:tfin];   % Vector of uniformly spaced time points
u     = [0*t' 0*t'];         % Set input u to zero for all time in order to
generate zero input response;
% Excite Fast Instability.
% This mode
%             is associated with a pole at s = 14.1050.
%             is associated primarily with theta_2 dot.
x = lsim(ss(A_asm, B_asm, eye(ns_asm,ns_asm), 0*ones(ns_asm,nc_asm)), u, t,
evec(:,1));
figure; plot(t,x)
grid
title('Fast Instability')
ylabel('States')
xlabel('Time (seconds)')
pause
%
% Excite Slow Instability.
% This mode
%             is associated with a poles at s = 4.5299.
%             is associated primarily with theta_1 and theta_1 dot
%
x = lsim(ss(A_asm, B_asm, eye(ns_asm,ns_asm), 0*ones(ns_asm,nc_asm)), u, t,
real(evec(:,4)));
figure; plot(t,x)
grid
title('Slow Instability')
ylabel('States')
xlabel('Time (seconds)')
pause
%% Transmission Zeros
plantzeros = tzero(ss(A_asm,B_asm,C_asm,D_asm))       % transmission zeros
%% SYSTEM TRANSFER FUNCTIONS: From u_i to x_j
Plant_zpk = zpk(ss(A_asm,B_asm,C_asm,D_asm)) % Zeros, Poles, and Gains fron
u_i to x_j
%% Controllability
rank(ctrb(A_asm,B_asm))
%% Observability
rank(obsv(A_asm,C_asm))
%% FREQUENCY RESPONSE: Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A_asm, B_asm, C_asm, D_asm),w);
sv     = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
```

```
title('Outputs,Inputs: ')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% PLANT SVD ANALYSIS at DC
dc              =  C_asm*inv(-A_asm)*B_asm
[udc,sdc,vdc] = svd(dc)
%% Designing an LQG/LTR Compensator for anti-symmetric motion
A3 = [zeros(nc_asm,ns_asm+nc_asm); B_asm A_asm];
B3 = [eye(nc_asm);zeros(ns_asm,nc_asm)];
C3 = [zeros(nc_asm) C_asm];
D3 = [zeros(no_asm,nc_asm)];
Ll3 = inv(C_asm*inv((-A_asm))*B_asm);
Lh3 = inv(-A_asm)*B_asm*Ll3;
Llqg3 = [Ll3;Lh3];
mu = 1.5;
B_fare3 = (C3'*inv(mu*eye(2))*C3);
sigma3 = are(A3',B_fare3,(Llqg3*Llqg3'));
H3= sigma3*C3'*inv(mu*eye(2));
%% Target Closed loop poles and zeros
tclpoles3 = eig(A3-H3*C3);
tclzeros3 = tzero(A3-H3*C3,H3,C3, D3);
%% Target Open Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A3,H3,C3,D3),w);
tlsv   = 20*log10(sv);
semilogx(w, tlsv)
title('Target Open Loop Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Target Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A3-H3*C3,H3,-C3,eye(no_asm,no_asm)),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Sensitivity Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Target Complementary Sensitivity Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
sv     = sigma(ss(A3-H3*C3,H3,C3,D3),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Comp Sensitivity Singular Values at Output')
grid
xlabel('Frequency (rad/sec)')
```

```
ylabel('Singular Values (dB)')
pause
%% Designing Pre-Filter
z = 1.2;
fil3   =  ss(tf({z 0; 0 z}, {[1 z] 1; 1 [1 z]}));
s                    = j*20;
tol_s                = C3*inv(s*eye(size(A3))-A3)*H3 + D3;
[tolu, tols, tolv ] = svd(tol_s);
%% Target Closed Loop Singular Values
winit  = -1;
wfin   =  2;
nwpts  = 200;
w      = logspace(winit,wfin,nwpts);   % Form vector of logarithmically
spaced freq points
[ntacl3, ntbcl3, ntccl3, ntdcl3 ] = series(fil3.a, fil3.b, fil3.c, fil3.d,
A3-H3*C3,H3,C3,D3);
sv     = sigma(ss(ntacl3, ntbcl3, ntccl3, ntdcl3),w);
sv     = 20*log10(sv);
semilogx(w, sv)
title('Target Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('T_{ry}, Singular Values (dB)')
pause
%% Target Step Responses
tinit = 0;
tinc  = 0.005;
tfin  = 100.0;
t     = [tinit:tinc:tfin]';% Vector of uniformly spaced time points
r1    = -1*[ones(size(t)) zeros(size(t))];
r2    = 5*[zeros(size(t)) ones(size(t))];
ty1   = lsim(ss(ntacl3, ntbcl3, ntccl3, ntdcl3), r1,t);
ty2   = lsim(ss(ntacl3, ntbcl3, ntccl3, ntdcl3), r2,t);

plot(t,ty1(:,1),'b', t,ty1(:,2),'r')
grid
title('Target Responses to Step Reference Command')
ylabel('')
xlabel('Time (seconds)')
pause


plot(t,ty2(:,1),'b', t,ty2(:,2),'r')
grid
title('Target Responses to Step Reference Command')
ylabel('')
xlabel('Time (seconds)')
pause
%% LTR at output
rho3 = 0.001;
R3 = rho3*eye(nc_asm);
[G3,X3,poles3] = lqr(A3,B3,C3'*C3,R3);
%% Model Based Compensator followed by integrator bank
Ak3 = [0*ones(nc_asm,no_asm)  G3; zeros(ns_asm+nc_asm,no_asm)  A3-B3*G3-
H3*C3];
Bk3 = [zeros(nc_asm,no_asm); H3];
Ck3 = [eye(nc_asm,no_asm)  zeros(nc_asm, ns_asm+nc_asm)];
Dk3 = [zeros(no_asm,no_asm)];
kpoles = eig(Ak3)                              % Compensator Poles
kzeros = tzero(Ak3, Bk3, Ck3, Dk3)                % Compensator Zeros

winit  = -2;
```

```
wfin    = 4;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(Ak3, Bk3, Ck3, Dk3),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Compensator Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Open Loop System
[al3, bl3, cl3, dl3 ] = series(Ak3, Bk3, Ck3, Dk3, A_asm, B_asm, C_asm,
D_asm);

olpoles = eig(al3)                             % Open Loop Poles
olzeros = tzero(al3,bl3,cl3,dl3)                  % Open Loop Zeros

winit    = -1;
wfin     = 2;
nwpts    = 200;
w        = logspace(winit,wfin,nwpts);
sv       = sigma(ss(al3, bl3, cl3, dl3), w);
sv       = 20*log10(sv);
figure; semilogx(w, sv, w, tlsv)
%clear sv
title('Open Loop Singular Values at Error (Recovered and Target)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause


figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
[ali3, bli3, cli3, dli3 ] = series(A_asm,B_asm,C_asm,D_asm, Ak3, Bk3, Ck3,
Dk3);

winit    = -1;
wfin     = 2;
nwpts    = 200;
w        = logspace(winit,wfin,nwpts);
sv       = sigma(ss(ali3, bli3, cli3, dli3 ), w);
sv       = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Open Loop Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Closed Loop System
acl3     = al3-bl3*cl3;
bcl3      = bl3;
ccl3      = cl3;
dcl3      = dl3;
```

```
clpoles = eig(acl3)                 % Closed Loop Poles
damp(clpoles)
clzeros = tzero(acl3,bcl3,ccl3,dcl3) % Closed Loop Zeros (r to y)
%% Sensitivity at Error
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl3, bcl3, -ccl3, eye(2)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Error')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Sensitivity at Input
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(ali3-bli3*cli3, bli3, -cli3, eye(2)),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Sensitivity Singular Values at Input')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Complementary Sensitivity
winit   = -1;
wfin    = 2;
nwpts   = 200;
w       = logspace(winit,wfin,nwpts);
sv      = sigma(ss(acl3, bcl3, ccl3, dcl3),w);
sv      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Complementary Sensitivity Singular Values at Ouput')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Add Pre-filter to Closed Loop Transfer Function Matrix From r to y
[nacl3, nbcl3, nccl3, ndcl3 ]   = series(fil3.a, fil3.b, fil3.c, fil3.d,
acl3,bcl3,ccl3,dcl3);
sv                              = sigma(ss(nacl3, nbcl3, nccl3, ndcl3),w);
sv                              = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Closed Loop Singular Values (r to y)')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Reference to Controls (r to u)
[aru3, bru3, cru3, dru3 ] = series(acl3, bcl3, -ccl3, eye(2), Ak3, Bk3,
Ck3, Dk3);
[aru3, bru3, cru3, dru3 ] = series(fil3.a, fil3.b, fil3.c, fil3.d, aru3,
bru3, cru3, dru3 );
```

```
winit                   = -1;
wfin                    = 2;
nwpts                   = 200;
w                       = logspace(winit,wfin,nwpts);
sv                      = sigma(ss(aru3, bru3, cru3, dru3),w);
sv                      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Reference to Control Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Form Input Disturbance to Output
[ady3, bdy3, cdy3, ddy3 ] = series(A_asm,B_asm,C_asm,D_asm, acl3, bcl3, -
ccl3, eye(2));
winit                   = -2;
wfin                    = 2;
nwpts                   = 200;
w                       = logspace(winit,wfin,nwpts);
sv                      = sigma(ss(ady3, bdy3, cdy3, ddy3),w);
sv                      = 20*log10(sv);
figure; semilogx(w, sv)
%clear sv
title('Input Disturbance to Output Singular Values')
grid
xlabel('Frequency (rad/sec)')
ylabel('Singular Values (dB)')
pause
%% Input Disturbance Analysis at DC
s = j*7
tdy3              = cdy3*inv(s*eye(size(ady3))-ady3)*bdy3 +  ddy2;
[utdy1 stdy1 vtdy1] = svd(tdy3)
%% Closed Loop Step Responses
y1    = lsim(ss(nacl3,nbcl3, nccl3, ndcl3), r1, t);
figure;
plot(t,y1(:,1))
grid
title('Response to Step Reference Command for Load Deviation from Center ')
ylabel('Outputs Load Deviation from Center , x_{L}-\Sigma x')
xlabel('Time (seconds)')
pause
%return

y2    = lsim(ss(nacl3,nbcl3, nccl3, ndcl3), r2, t);
figure;
plot(t,y2(:,2))
grid
title('Response to Step Reference Command for Average horizontal velocity')
ylabel('Outputs \Sigma \dot x')
xlabel('Time (seconds)')
pause
%return


u1    = lsim(ss(aru3, bru3, cru3, dru3), r1, t);
figure; plot(t,u1(:,2))
grid
title('Response to Step Reference Command for Differential Cyclic Control')
ylabel('Controls \theta_{c}')
xlabel('Time (seconds)')
pause
```

```
%return


u2    = lsim(ss(aru3, bru3, cru3, dru3), r2, t);
figure;plot(t,u2(:,2))
grid
title('Response to Step Reference Command for Average Cyclic Control')
ylabel('Controls \Sigma \beta_{lc}')
xlabel('Time (seconds)')
pause
```

-------------------THE END ----------------------------------