CONTROL DESIGN

INTRODUCTION

We begin by defining one basic but very widely used classical control design technique – Proportional-Integral-Derivative control, also called PID control. The PID design approach presented is based on stability margin assessment. Following this are the state-space design methods using pole placement in the design of (a) the full-state regulator and (b) the asymptotic observer. Modern optimal control state-space design techniques are then introduced, beginning with a description of controllability and observability, and covering the tools provided by linear quadratic regulator or the LQR controller, and the Kalman filter as an observer. Here the emphasis is not their derivation or the theoretical bases for these techniques – the emphasis is on concepts associated with their application -- how these tools provide very powerful methods in a control system designer's arsenal for handling difficult design problems that involve complicated systems. The control weighting and "noise" spectral density matrices – design parameters adjusted in the course of generating the control and filter design solutions – are described. A linear quadratic Gaussian controller is formed by the combination of the linear quadratic regulator and Kalman observer. Together they form a dynamic compensator, and we cover the concept of adding integral control called the industrial regulator to both the LQR formulation of a full-state regulator and a combined LQR and Kalman observer. The discretization of controllers designed using continuous-time methods, and the incorporation of system delays are also described.

The outline of this chapter:

- PID control
- State-space design
    - Full-state regulator design by pole placement
    - Asymptotic (Luenberger) observer design by pole placement
    - Compensator equations
    - Separation Principle
- Modern optimal control design
    - Linear Quadratic Regulator (LQR)
        - Controllability
        - Selection of Q & R matrices
    - Optimal Observer (Kalman Filter)
        - Observability
        - Selection of V & W matrices
    - Linear Quadratic Gaussian (LQG) control
    - Addition of Integral Control
        - Industrial Regulator
- Miscellaneous topics
    - Discretizing a continuous-time compensator
    - Accommodating delay
        - Pade' Approximation

PROPORTIONAL – DERIVATIVE – INTEGRAL CONTROL

The Proportional – Integral – Derivative (PID) controller involves 3 independent gains, K_P, K_I, and K_D, the proportional, the integral, and the derivative gains, respectively. Controller gains are selected with the goal of achieving specific performance objectives -- i.e. the speed of response (rise time, settling time, overshoot) which lead directly to dominant pole location definition. To that end we define:

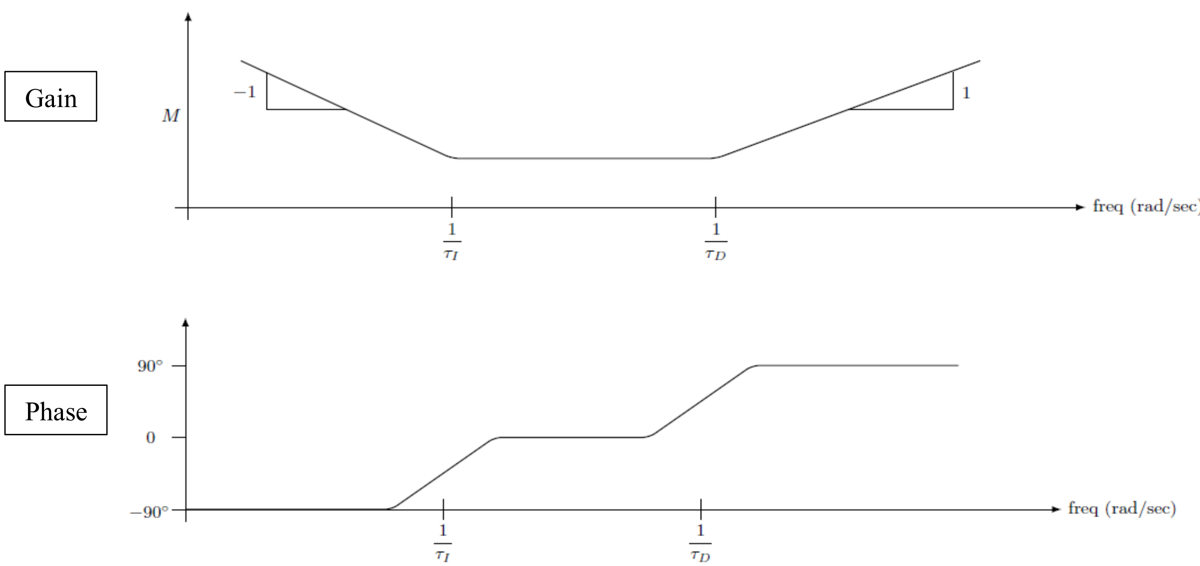
- a natural frequency parameter ω_n which is approximately equal to the controlled systems open loop crossover frequency. K_P is computed to approximately achieve that crossover frequency.

- a derivative gain term to provide phase lead as needed to improve phase margin (PM), and thereby improve settling time and overshoot. K_D is computed to achieve the desired PM.
- An integral gain term to increase the low frequency gain as needed to reject disturbance affects -- i.e. biases and/or dynamic inputs. K_I is selected to boost open-loop low frequency gain without degrading the crossover phase margin. The break frequency associated with K_I is selected to provide this low frequency gain while typically well below the crossover frequency to avoid impacting PM.

The PID controller $(K_P + K_D s + K_I s^{-1})$ can be represented in the convenient form

$$\frac{K}{s}(1+\tau_I s)(1+\tau_D s) \tag{6-1}$$

which makes the frequency response of the PID controller clearly discernible. It comprises an integrator at low frequencies, a break at frequency $1/\tau_I$ to a region of flat response, and finally a 2nd numerator break at $1/\tau_D$:



Usually the derivative break $1/\tau_D$ is very close to the crossover in order to provide the appropriate phase lead in that region.

Expressing (1) in more general form:

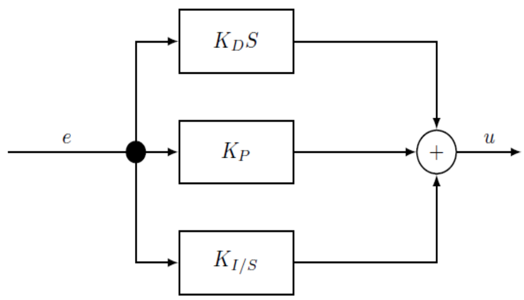
$$\frac{K}{s}\left(1 \;+\; (\tau_I + \tau_D)s \;+\; \tau_I \tau_D s^2\right)$$

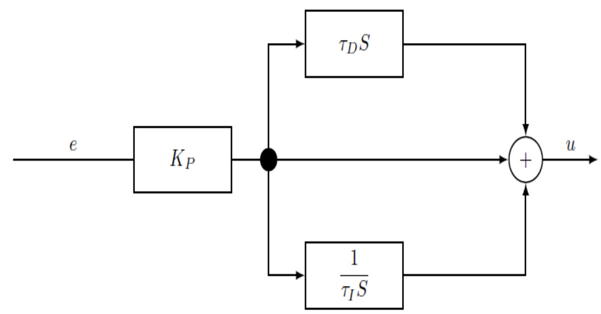
and noting that the PID controller can also be expressed as:

$$\frac{(K_D s^2 + K_P s + K_I)}{s}$$

we see that:

$$K_P = K(\tau_I + \tau_D) \approx K\tau_I$$
$$K_D = K\tau_I\tau_D$$
$$K_I = K$$





where the approximation holds because $\tau_I >> \tau_D$. From this we then have:

$$\frac{K}{s}\left(1 + \tau_I s + \tau_I\tau_D s^2\right)$$
$$= K\tau_I\left(\frac{1}{\tau_I s} + 1 + \tau_D s\right)$$
$$= K_P\left(\frac{1}{\tau_I s} + 1 + \tau_D s\right)$$

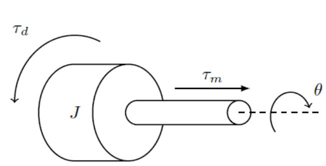
So it is possible to implement the PID controller in the convenient form on the right and thereby very easily select the derivative and integral time constants to place the derivative and integral breaks at desired locations.

The steps for design of the PID controller therefore are:

(1) Select K_P to achieve the crossover frequency desired,
(2) Set τ_D to provide the needed phase lead, and
(3) Set $1/\tau_I$ to provide integral gain at the lower frequencies.

PID Design Example

Perform a PID control design for the 2nd order motor model:



2nd Order Motor model:

Motor inertia $J = 1.0\ in - oz - s^2$
Motor Damping $\beta = 5\ m - oz - s$
Motor Torque constant $K_T = 1\ in - oz/amp$

Dynamic Equations of motion:

$$J\ddot{\theta} + \beta\dot{\theta} = K_T i$$

Transfer function:

$$\frac{\theta(s)}{V(s)} = \frac{KT\ /\ R}{(Js + \beta)s}$$

The state space model is:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\beta/J \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (K_T/R)/J \end{bmatrix} u \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where:

$x_1 = \theta$ Angle (rad)

$x_2 = \omega$ Angular rate (rad/sec)

Three solutions were developed: Proportional control only, Proportional+Derivative, and Proportional+Integral +Derivative.

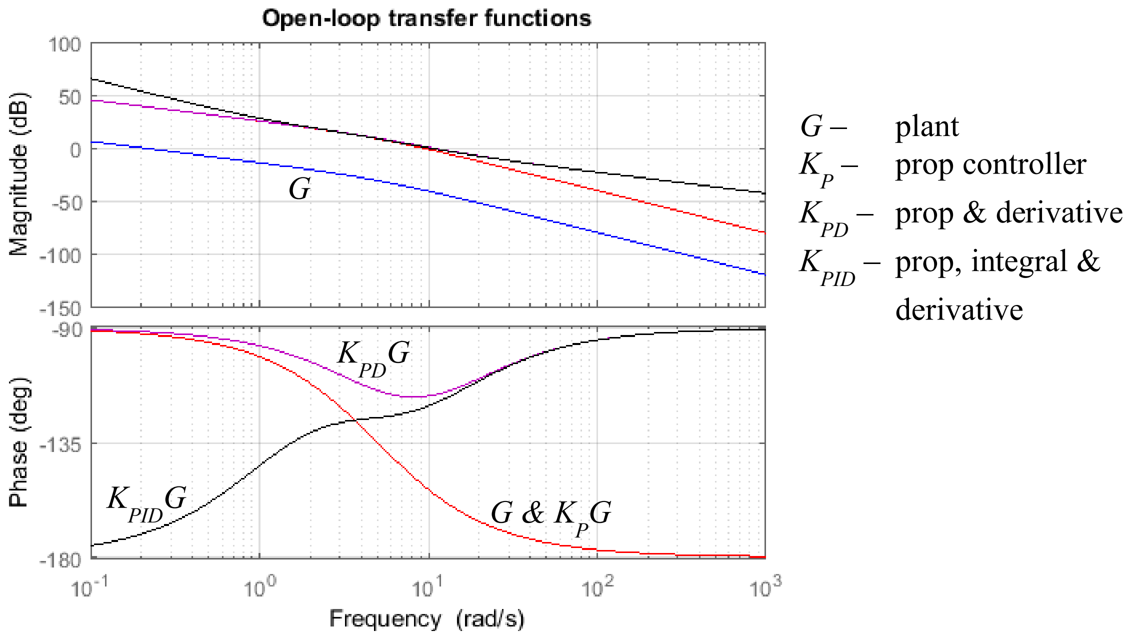
```
%%PID controller design:
%KiKdKpG=[(1/(s/Wi))+(s/Wb) + 1 ] * Kp * G
%Kp is the proportional gain:
%           It is tied to Tr=1.8/Wn
%(s/Wb) + 1 is related to derivate term
%           It is tied to the amount phase lead req
%           nWb=Wn
%1/(sWi) is related to the integral gain
%           Wi will be placed one decade below Wb

%REQUIREMENTS:
    %(1)Tr=0.2secs
    %(2)PM=60deg

%System Parameters ************************************
J=1;
B=5;
Kt=1;
R=1;

%Gs open-loop transfer function
s = tf('s');
```

```
G = (Kt/R) / [(J*s^2)+(B*s)];
%Break frequency Wn of open-loop transfer function
%Wn = 0.196
%*******************************************

%Requirement
%Tr = 0.2 , Tr=1.8/Wn
%Desired crossover frequency
Wn=1.8/0.2;
%Wn = 9 rad/sec

%To move crossover freq add a proportional gain of -39.5dB
Kp=10^(39.5/20);

%Adding proportional gain to plant G
%Kp*G
s = tf('s');
KpG = Kp * G;

%60deg phase margin requirement
%Note for above Kp*G the PM = 26deg = 154-180
                    %Requirement is for PM of 60deg
                    %Add 34 deg to Kp*Gs Bode-Phase-Profile

%Derivative Term needs to add 34 deg of phase lead
%Kp+Kds = Kp ( 1 + Kd/Kp s) = Kp ( Kd/Kp s + 1 )
%       With Kd/Kp = tao --> = Kp ( tao   s + 1 )
%       With    Wb = 1/tao-> = Kp (  s/Wb   + 1 )

%Recall from (  s/Wb   + 1 ) bode phase-profile
%0.67Wb yields 26degs phase lead
%With Wb = 9 for Kp*G(s) Bode phase-profile and 0.6Wb=9
%Solve for new cross over frequency Wb
%Wb=9/0.67 -->Wb =13.43
Wb=9/0.67;

%Now the KdKpGs transfer function
s = tf('s');
KdKpG=(s/Wb + 1 ) * Kp * G;%Note new Wb
%KdKpG bode phase-profile now has 180-117=63 phase lead which meets the 60deg PM
required

%Finally add Ki: 1/(s/W)
%W is the frequency at which we want the
%integral gain break frequency.
%Set W frequency 1 decade below KdKpGs bode phase-profile crossover freq
%W = 1

s = tf('s');
KiKdKpG=[(1/s)+(s/Wb) + 1 ] * Kp * G;
bode(G,'b',KpG,'r', KdKpG,'m', KiKdKpG,'k',{0.1,1000})
```

```
title('Open-loop transfer functions')
grid
```

Open-Loop Frequency Response

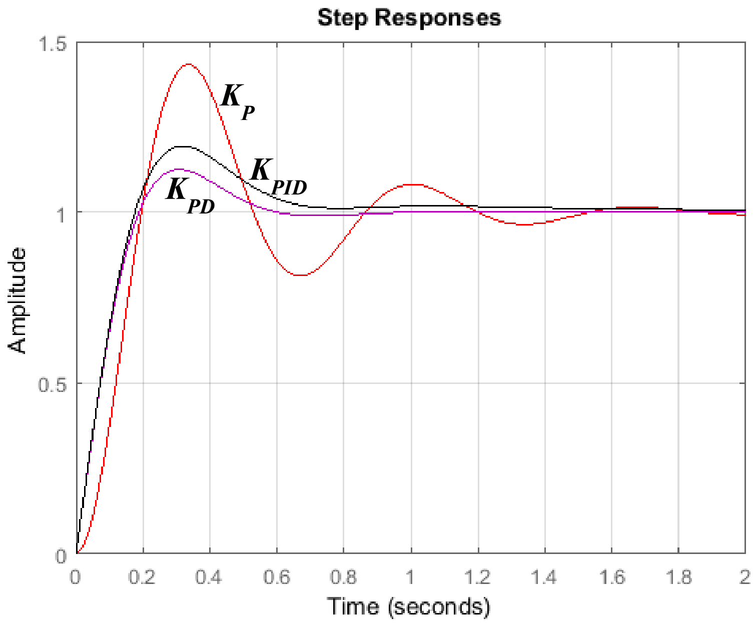
The open loop transfer functions of the controlled plant (G) for all three controllers are shown in the following figure. Note that at the crossover frequency the derivative term adds considerably to the phase margin, some of which is lost when adding in the integral control term. Also note that at low frequencies the integral term causes the gain to grow as frequency decreases with a slope of 2.



Step Responses

Examination of the step response clearly shows the significant improvement in lessening overshoot is afforded by the derivative control, some of which returns with the addition of the integrator. What is not evident yet is the dramatic improvement the dramatic improvement in low frequency tracking performance and disturbance rejection that the integral action achieves, both of which are noted below.

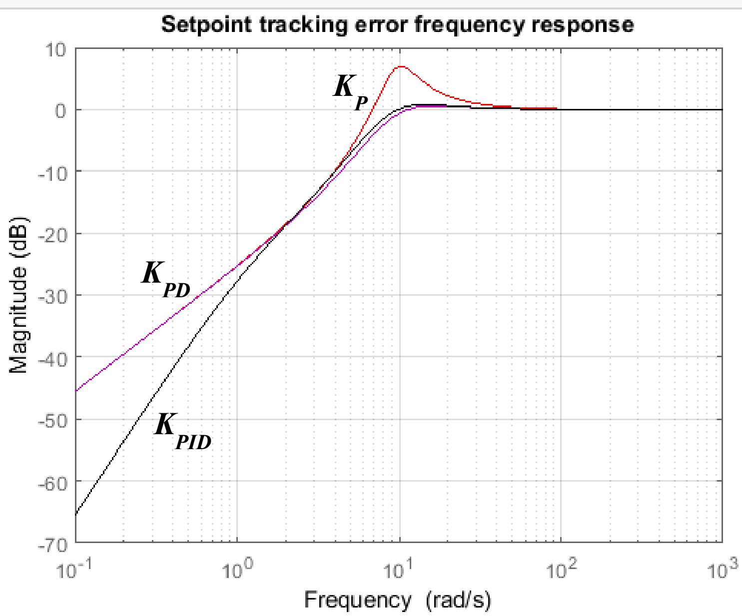
```
CL_KpG = KpG/(1+KpG);
CL_KdKpG = KdKpG/(1+KdKpG);
CL_KiKdKpG = KiKdKpG/(1+KiKdKpG);
figure
stepplot(CL_KpG,'r',CL_KdKpG,'m',CL_KiKdKpG,'k',2)
title('Step Responses'), grid
```



Tracking error frequency response

In the plot of setpoint tracking error frequency response, the derivative control reduces the 'resonant' peaking that occurs at the crossover frequency, which is also the point in frequency above which the control no longer provides any tracking capability. At low frequencies the integral control action greatly improves the tracking accuracy, making the tracking error response significantly small there.

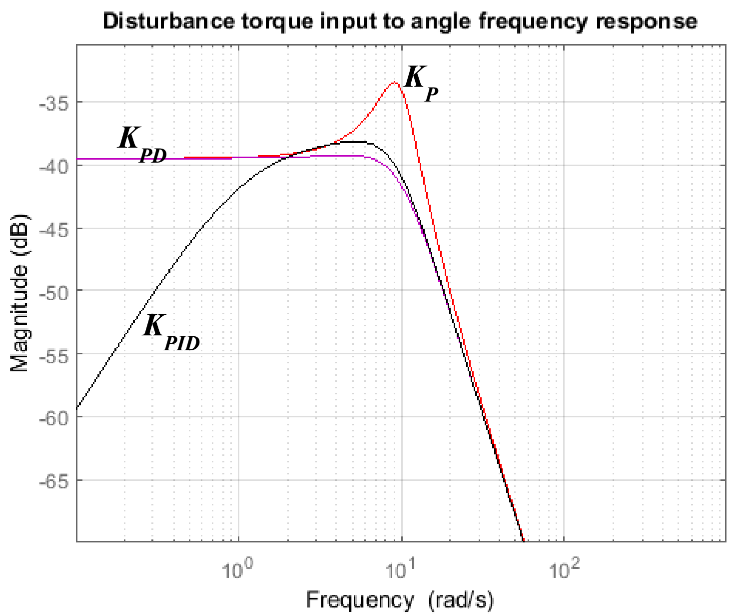
```
e_KpG = 1/(1+KpG);
e_KdKpG = 1/(1+KdKpG);
e_KiKdKpG = 1/(1+KiKdKpG);
figure
bodemag(e_KpG,'r',e_KdKpG,'m',e_KiKdKpG,'k',{0.1,1000})
title('Setpoint tracking error frequency response')
grid
```



Disturbance rejection frequency response

These curves tend to peak near the crossover frequency. At frequencies above the cross over the system naturally rejects disturbances as a result of its inertia. At the lower end, the integral control action greatly reduces the systems response to disturbance (torque) inputs

```
d_KpG = G/(1+KpG);
d_KdKpG = G/(1+KdKpG);
d_KiKdKpG = G/(1+KiKdKpG);
figure
bodemag(d_KpG,'r',d_KdKpG,'m',d_KiKdKpG,'k',{0.1,1000})
title('Disturbance torque input to angle frequency response')
grid
```



State-Space design methods

Full-state regulator design by pole placement

Observer design by pole placement

Compensator equations

Separation Principle

Modern Optimal design methods

Controllability / Observability

LINEAR QUADRATIC FULL-STATE REGULATOR – LQR

SELECTION OF Q & R MATRICES

ADDITION OF INTEGRAL CONTROL

Two techniques for adding integral control action to a controller designed using the LQR method are described. The first available is called the Industrial Regulator and can be derived through the use of the LQR method described above. When full state feedback is available it avoids the use of an observer. The second requires a controller and filter, the filter being needed for generation of the integral state even if all of the other system states are available.

INDUSTRIAL REGULATOR

The Industrial Regulator [ref] is a control law that does not require an observer design – the design of the controller is done to produce a full state regulator using for example, the full-state LQR or pole placement solution methods. Integral action is obtained by feeding back the output y to a summing junction with a reference input y_{REF} and sending that error through an integrator with a gain also produced by the LQR design process. The controlled plant therefore has the form given below:

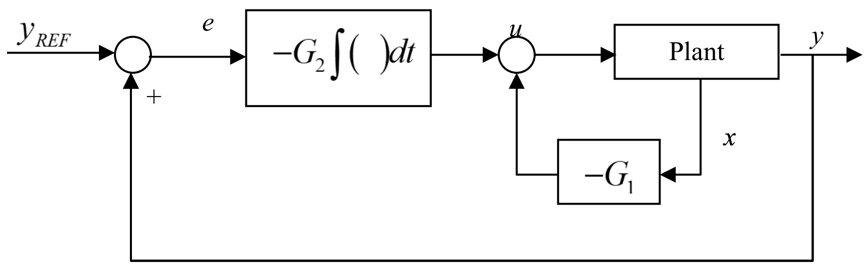


Figure A-4 – Industrial Regulator

The control law by definition contains integral action:

$$u = -G_1 x - G_2 \int \big(e(t)\big) dt \tag{6-2}$$

where $e(t) = y - y_{REF}$. To put this control into a form that is equivalent to that of the full-state regulator problem (i.e. with $u = -Gx$), take the derivative of u and of e:

$$\dot{u} = -G_1 \dot{x} - G_2 e$$
$$\dot{e} = \dot{y} - \dot{y}_{REF} = C\dot{x}$$

where we have assumed that y_{REF} is constant so that it derivative is zero.

Form an augmented state vector

$$z = \begin{bmatrix} \dot{x} \\ e \end{bmatrix}$$

and define the new control $\bar{u} = \dot{u}$.

With these definitions we set up the regulator problem for the augmented state-space system with state variable vector *z.*

$$\dot{z} = \begin{bmatrix} \ddot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \overline{u}$$

or

$$\dot{z} = \overline{A}z + \overline{B}\overline{u} \tag{6-3}$$

with

$$\overline{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \qquad \overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

Thus we will generate controller gain matrices G_1 and G_2 for the Industrial Regulator controller of Figure A-4 by computing the LQR full-state controller gains for the augmented system given above. In other words, synthesize a state feedback controller in terms of *z,* and then use those gain terms in the control law (A1). That controller:

$$\overline{u} = -Gz$$

when expressed in terms of the control signal '*u*', appear as follows

$$\dot{u} = -G_1\dot{x} - G_2 e = -\begin{bmatrix} G_1 & G_2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ e \end{bmatrix}$$

$$u = -G_1 x - G_2 \int e\,dt = -\begin{bmatrix} G_1 & G_2 \end{bmatrix} \begin{bmatrix} x \\ \int e\,dt \end{bmatrix}$$

A unique solution for the gain matrix *G* exists when the pair $\{\overline{A}, \overline{B}\}$ is controllable, or equivalently when the following conditions all hold:

(a) {A, B} is controllable
(b) $m \geq r$, that is – number of inputs $\geq$ number of outputs
(c) $rank \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = n + r$ where *r* is the number of outputs

INDUSTRIAL REGULATOR EXAMPLE

Consider the helicopter system given by the following:

$$\dot{x} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -5.6 & 40 & -.1 \end{bmatrix} x + \begin{bmatrix} 6.3 \\ 0 \\ 40 \end{bmatrix} u + \begin{bmatrix} -0.005 \\ 0 \\ -0.02 \end{bmatrix} w$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x$$

where the state, control, and disturbance variables are:

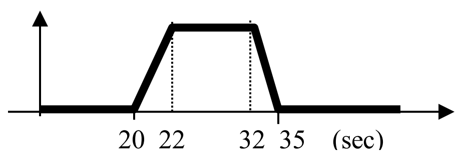
- x_1 – pitch rate (rad/sec),
- x_2 – pitch angle of the fuselage (radians),
- x_3 – horizontal velocity (ft/sec),
- u – control input, rotor blade tilt angle relative to the helicopter body (radians)
- w – head wind velocity (ft/sec)

[2] Shahian, B, Hassul, M., "Control System Design using Matlab", Prentice Hall, 1993.

The objective is to design an Industrial Regulator using the LQR controller design technique such that the following design requirements are met:

- Helicopter must transition from a zero velocity to a nominal and steady forward velocity of +100 ft/sec in 10 seconds in order to accomplish a refueling operation. (i.e. this is the step response); overshoot of <5 ft/sec is acceptable.

- Less than ±5 ft/sec of deviation from the 100 ft/sec setpoint during a wind gust (headwinds) of 30 ft/sec following the profile:



- Control input (rotor tilt) cannot exceed 15 degrees in magnitude at any time (step or disturbance responses).
- Fuselage pitch angle should not exceed 30 degrees in magnitude at any time.

The design procedure to be followed in determining the ultimate design that meets these requirements is as follows:

1) Define the parameters of the full-state LQR problem:
   - Define a Q matrix that will drive the error signal e to zero
   - Perform several design, starting with a large R (i.e.1e10)

(2) Adjust R downward until poles are in appropriate locations

- Good pole locations are expected to be those having a magnitude of 0.2 to provide a rise time of approximately 9 seconds

(3) Simulate each design to determine if the design constraints are met. Start with initial conditions (0 0 0)'.

```
%% Helicopter Industrial Regulator Control Problem
% System matrices:

A = [  -.4        0          -.01
        1         0             0
       -5.6       40         -.1];

B = [6.3          0      40 ]';

C = [0            0      1 ];

E = [-0.005       0      -0.02]';

% Form the Augmented System Matrices
Ap = [A  zeros(3,1) ; C 0 ];
Bp = [   B   ; 0 ];
Ep = [   E   ; 0 ];

% Define the LQR Weighting Matrices
Q = [0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     1  ];
R = 1e8;

% Solve for control gain G
[G,S,Eg] = lqr(Ap,Bp,Q,R)

% Define the gain submatrices for use in performance simulation
G1 = G(1:3);
G2 = G(4);
```

A simulation model of the helicopter and industrial regulator is constructed in Simulink. An augmented B matrix consisting of B and E is generated for use in the 'State-space' block to accommodate both inputs, *u* and *w*.

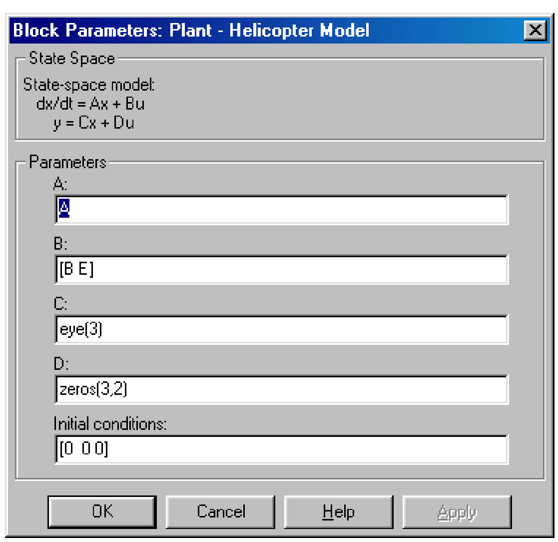


Figure A-5 – The State-space Block input matrices and initial conditions

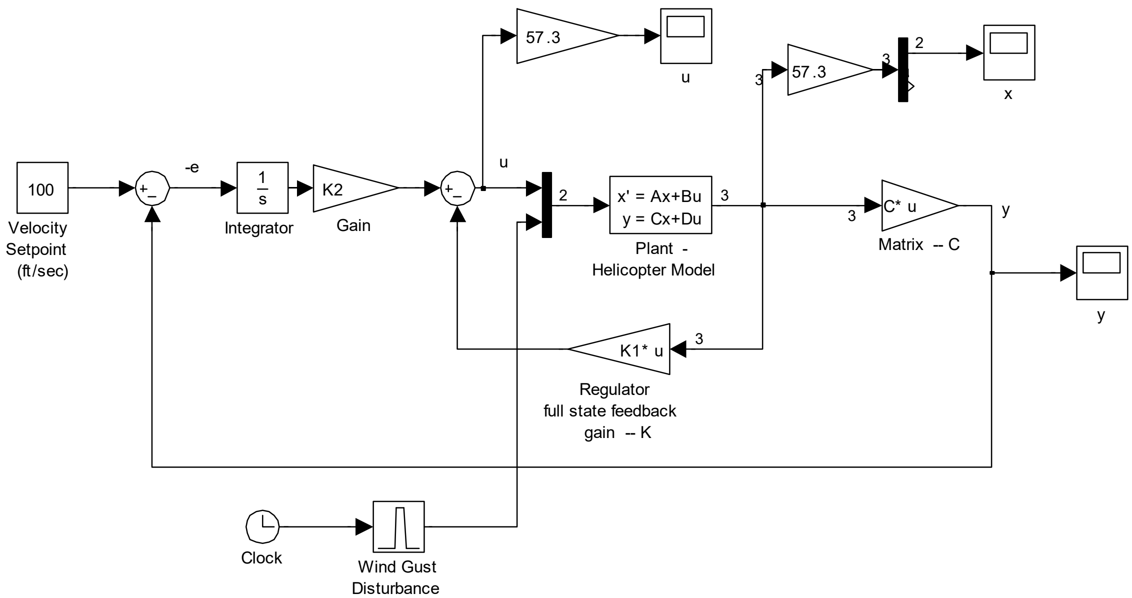
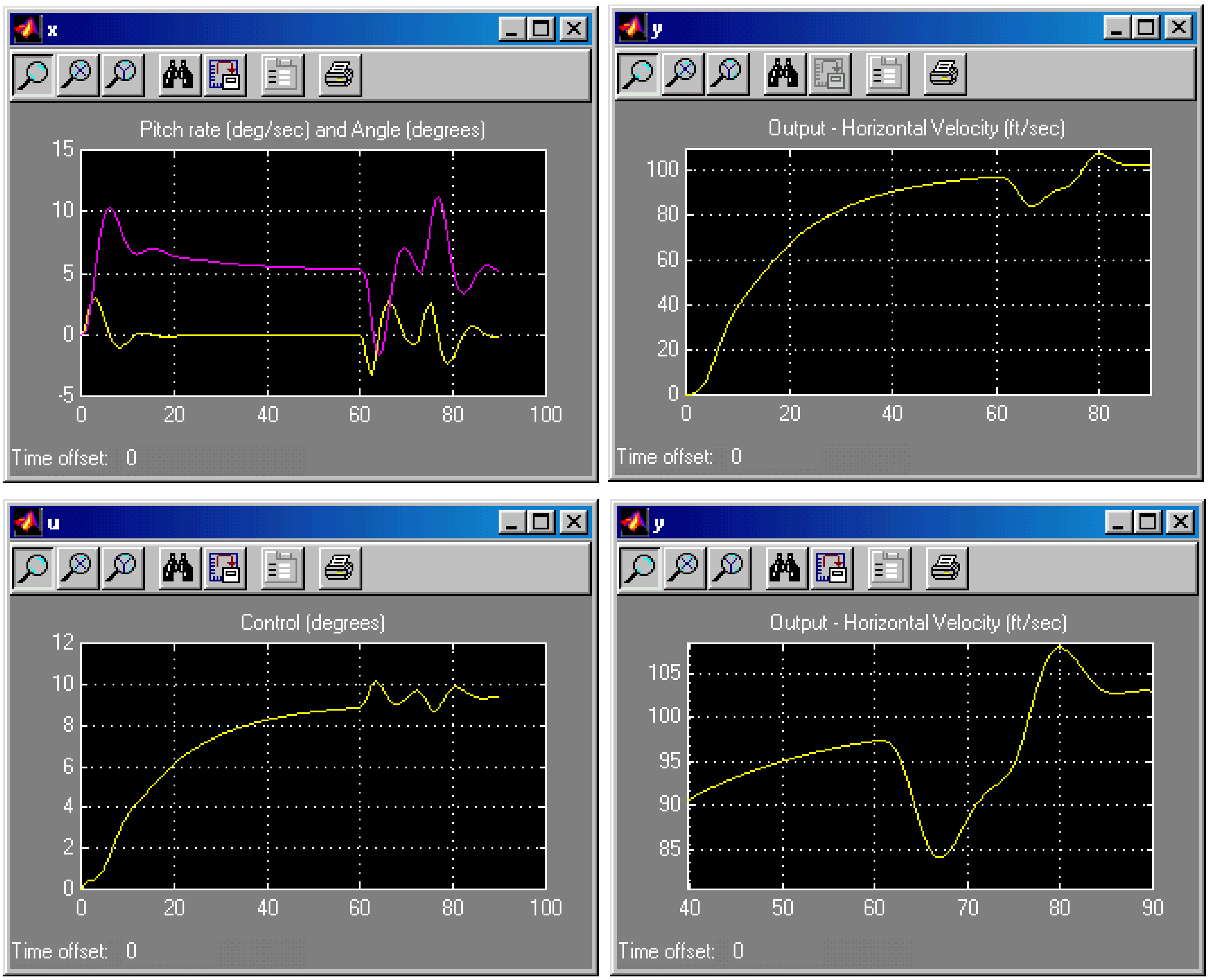


Figure A-5 – Helicopter Model with Industrial Regulator Controller

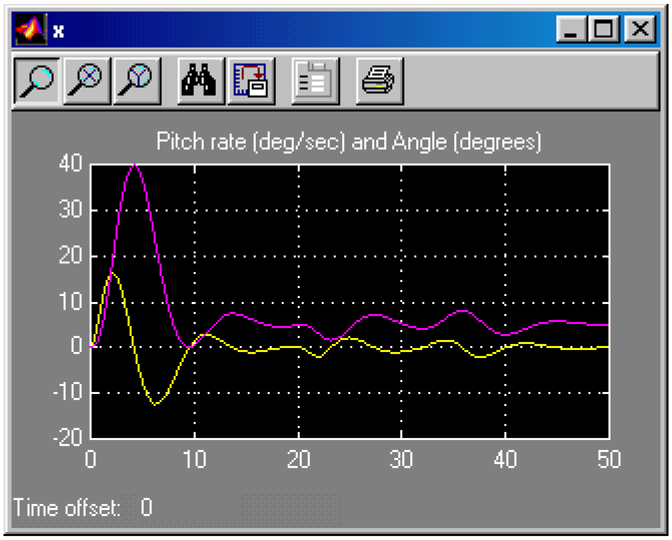
Several designs are developed and tested:

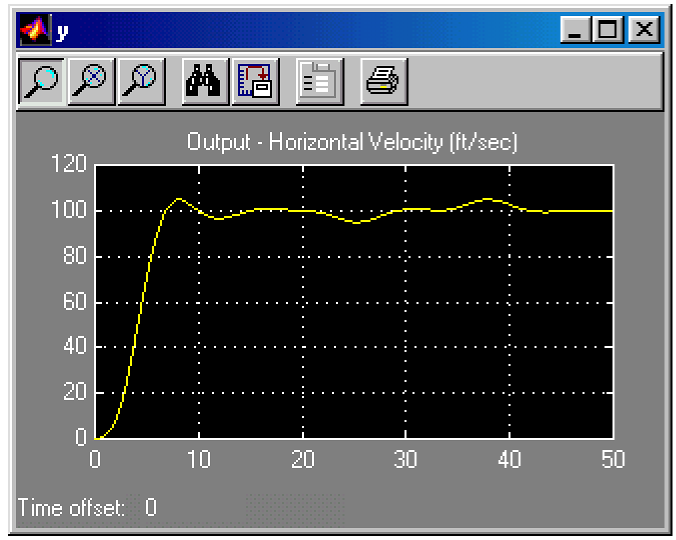
Design	Q=diag(x,x,x,x)	R	Eigenvalues	G = (x, x, x, x)
1	0, 0, 0, 1	1e8	-0.9549,-0.0627, -0.2279±0.6073i	0.1539 0.1465 0.0001 0.0001
2	0, 0, 0, 1	1e6	-0.9190,-0.5092, -0.2526±0.6890i	0.2133 0.2695 0.0022 0.0010
3	0, 0, 0, 1	1e4	-1.2252±0.5166i, -0.4075±1.1222i	0.3245 0.8062 0.0180 0.0100
4	0, 0, 0, 1	1e5	-0.93±0.29i, -0.32±0.86i	0.2735 0.4656 0.0068 0.0032
5	0, 0, 10, 1	1e5	-1.5601,-0.3115, -0.5455±1.1585i	0.3127 0.6703 0.0123 0.0032
6	0, 5e4, 3, 1	1e5	-2.665±2.6266i -0.1919±0.1417i	0.7080 2.5707 0.0188 0.0032

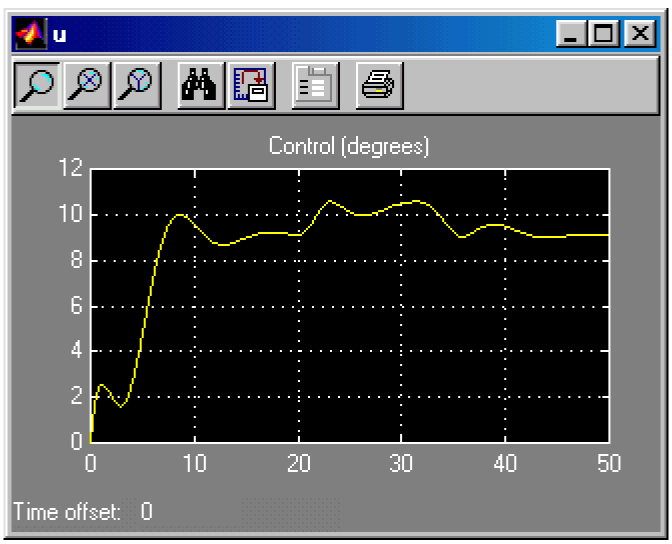
DESIGN #1 – Too slow on step response and wind gust (moved out to 60 seconds) causes excessive velocity disturbances. The pitch angle (purple) is well within the 30 degree limit, and the control level is well within the 15 degree rotor blade tilt limit (lower left). The velocity deviation due to wind exceeds the 5 ft/sec limit as seen in the lower right plot.

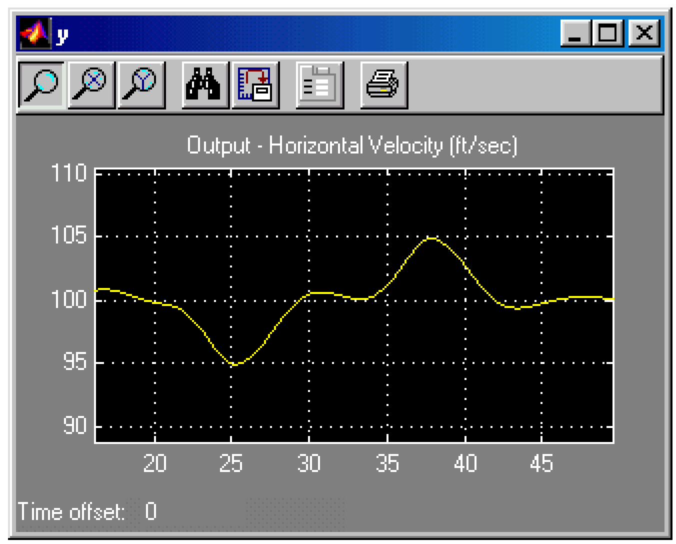


Design #2 – Pitch angle reaches 40 degrees, still exceeding 30 degree limit. Otherwise meets remaining requirements; but it has some undesirable overshoot in horizontal velocity.

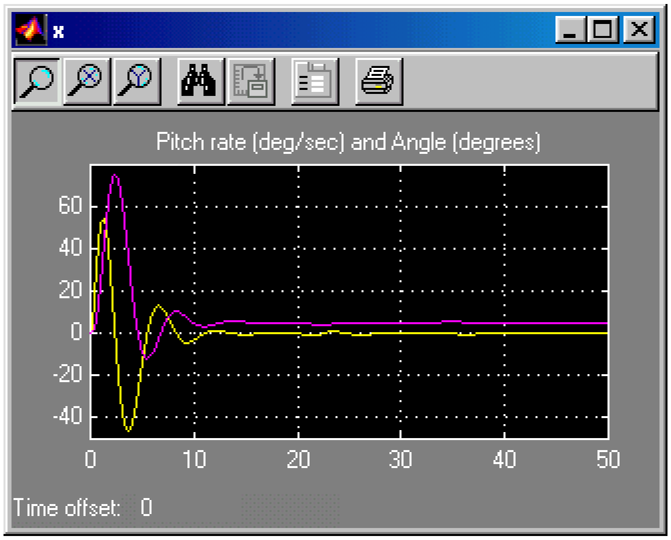


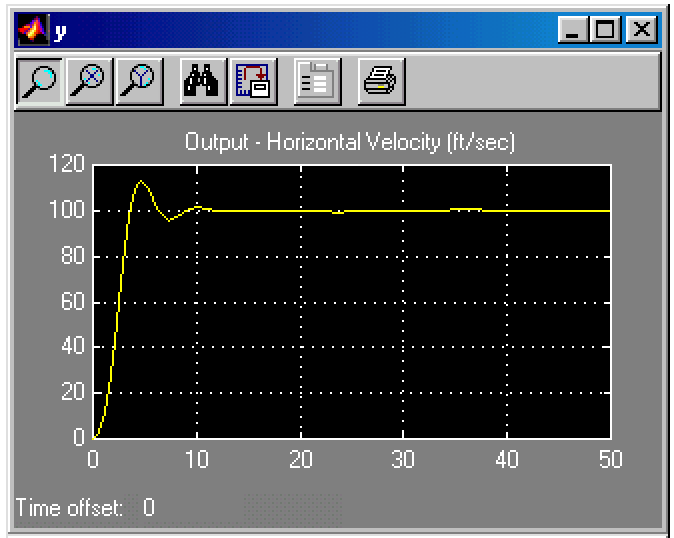


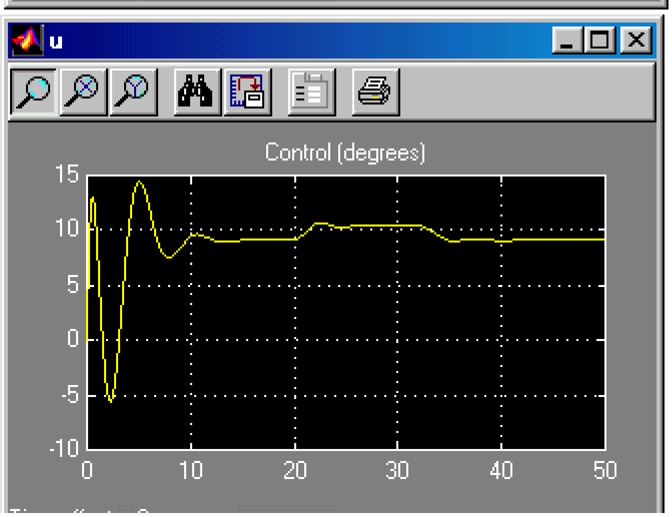


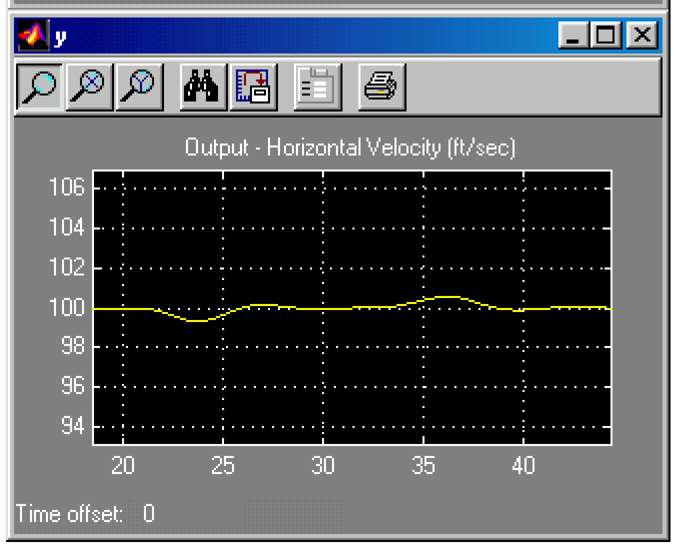


Design #3 – Fuselage pitch angle exceeds 30 degrees significantly, reaching about 75 degrees during the initial transient. The speed of response is very fast and the wind disturbance is acceptable.

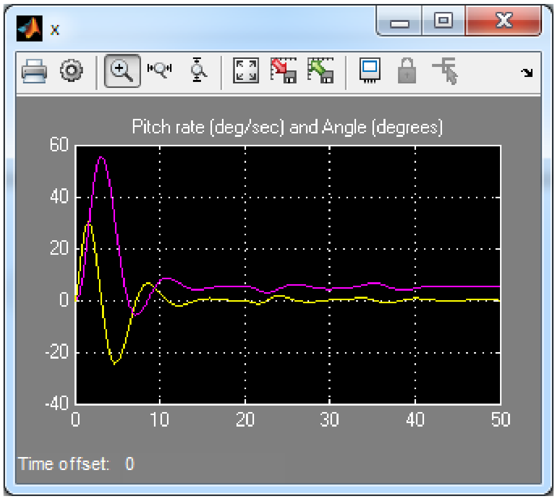


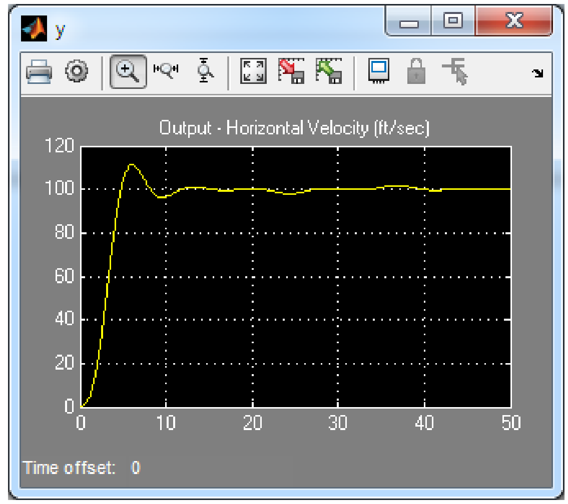


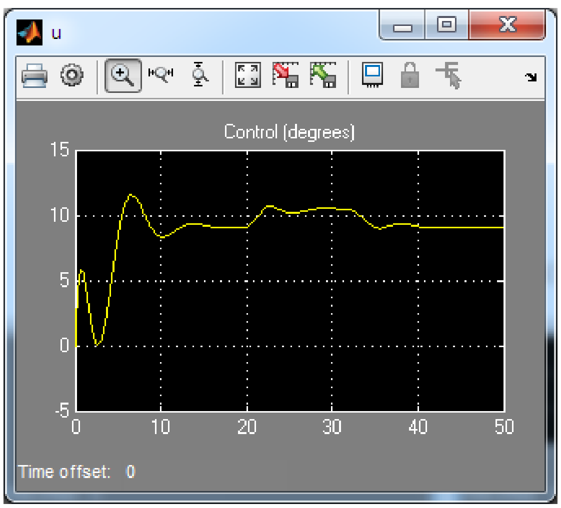


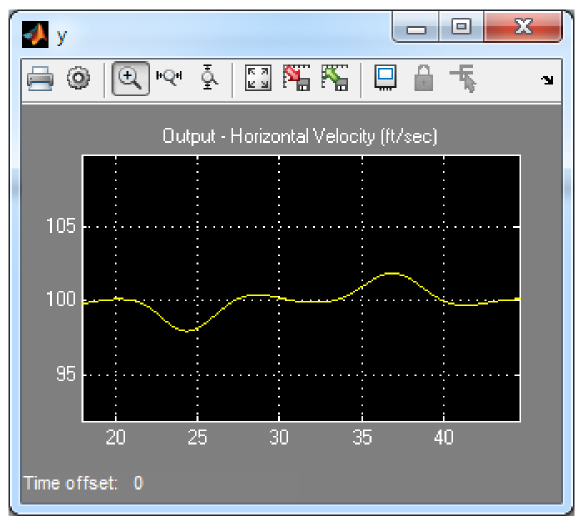


Design #4 – Fuselage pitch angle better but still exceeds 30 degrees significantly, reaching about 55 degrees during the initial transient, down by 20 degrees. The speed of response is very fast and the wind disturbance is acceptable.

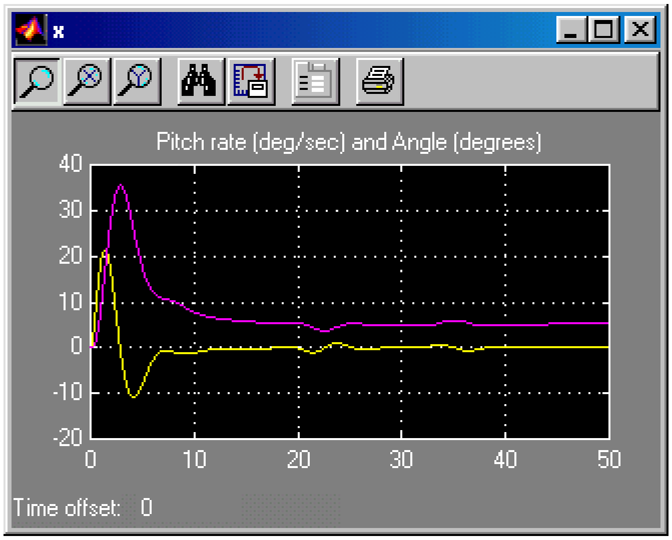


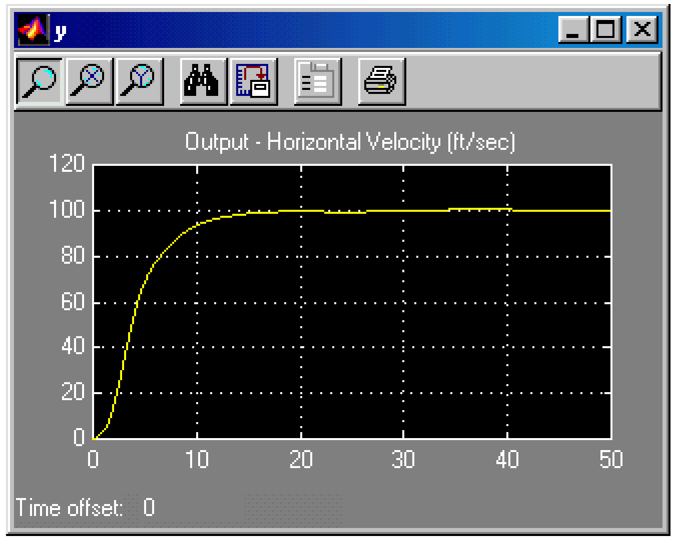


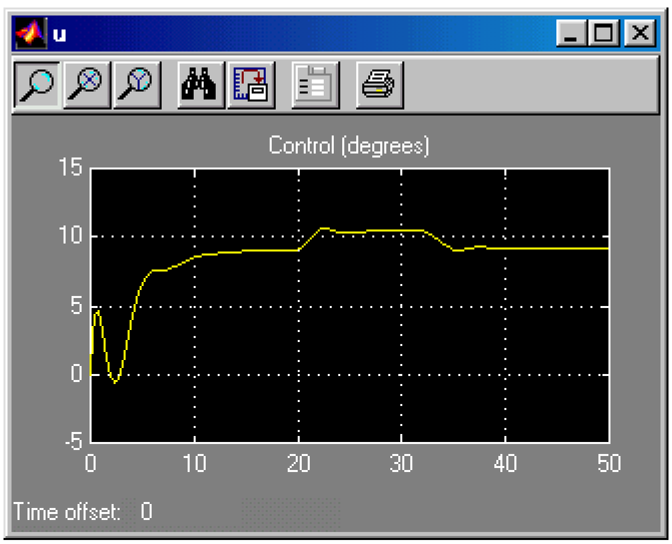


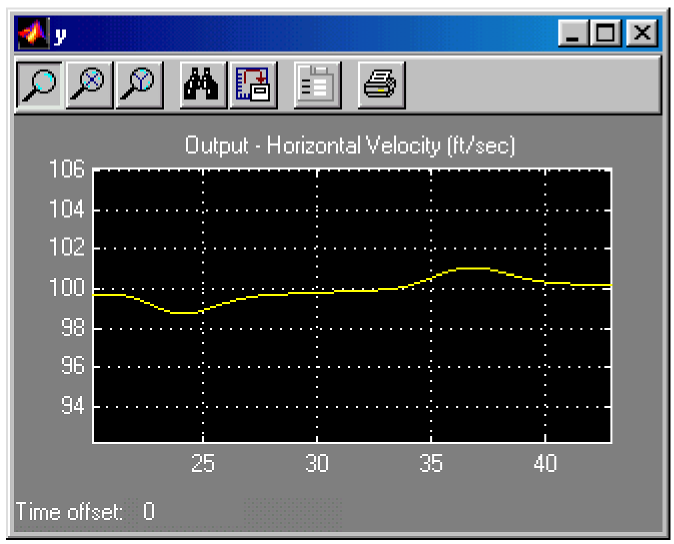


Design #5 – This one has the better performance. The element of the Q matrix weighting the horizontal velocity is added to dampen the transient response and eliminate the overshoot of the horizontal velocity. Note that the fuselage pitch still exceeds the 30 degree limit as is the control less than the 15 degree tilt limit. The wind disturbance is less than 5 ft/sec.

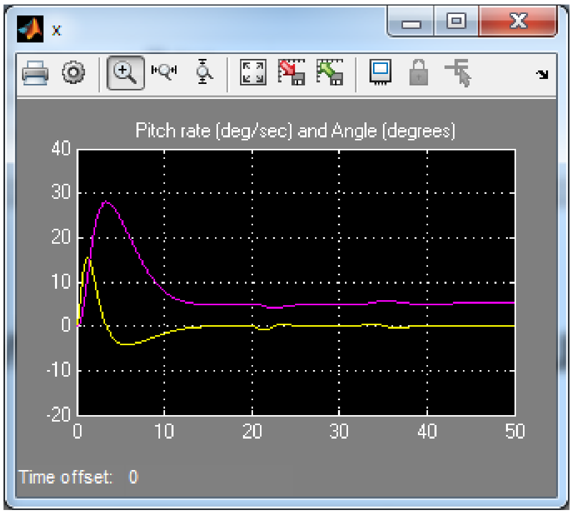


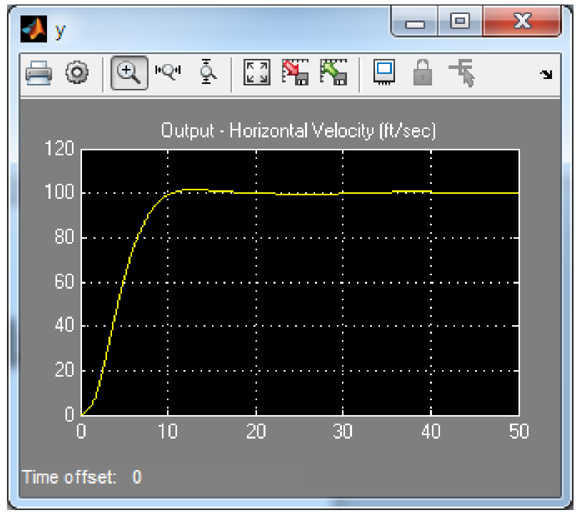


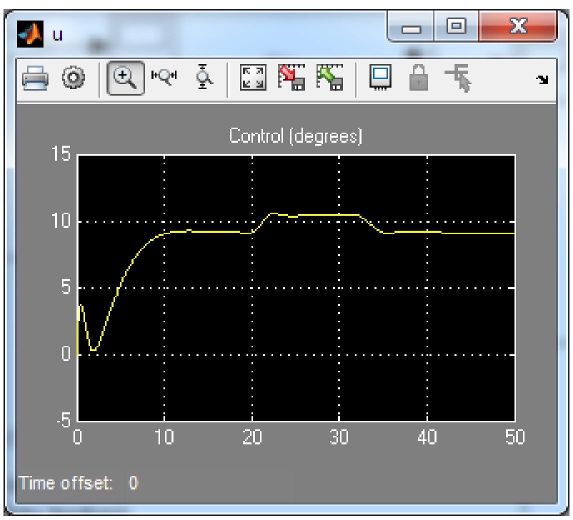


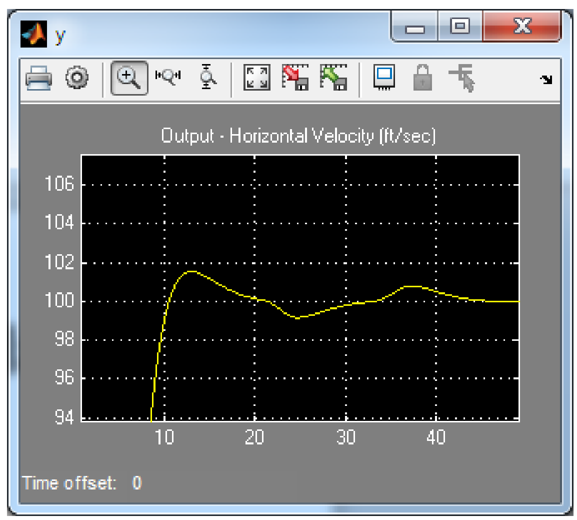


Design #6 – This one meets the pitch angle 30 degree limit and all other requirements. The element of the Q matrix added here introduces a weight on the pitch angle which is expected to reduce pitch angle excursions. We see below in the simulated results that it does. The wind disturbance is less than 2 ft/sec.









Open-Loop Frequency Response of the Industrial Regulator Controlled Plant

Compute the open-loop frequency response of a plant controlled by (1) the LQR Industrial Regulator involving full-state feedback, and then (2) the LQG Industrial Regulator with an observer providing state estimates for control.

LQR Industrial Regulator

The equation for the augmented system is used to derive the open-loop frequency response:

$$\dot{z} = \begin{bmatrix} \ddot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \overline{u}$$

or

$$\dot{z} = \overline{A}z + \overline{B}\overline{u} \qquad \text{(A2)}$$

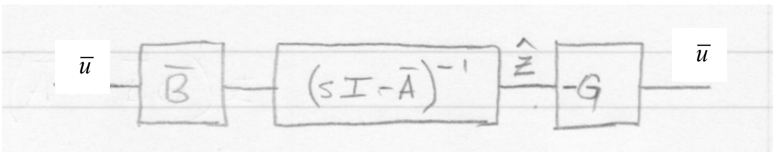
with

$$\overline{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \qquad \overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$\overline{u} = -Gz \qquad \text{(A3)}$$

$$u = -G_1 x - G_2 \int \left(e(t)\right) dt$$

Thus the open-loop response can be computed using the augmented system directly:



The state-space object provided within Matlab:

$$\text{sys\_ol} = ss(\overline{A}, \overline{B}, G, 0)$$

provides a control object representing the complete plant and full-state controller. The frequency response of 'sys_ol' is that open loop response of the Industrial Regulator controlled plant. Code to compute it with this method is given here:

```
sys_ol = ss(Abar,Bbar,G,0);
bode(sys_ol)
title('Bode plots -- Open loop LQR Full-state Ind Reg controlled plant')
grid
```

Alternatively we can compute the frequency response of the controller alone:

$$u = -\left[G_1 x + \frac{G_2}{s} Cx \right]$$

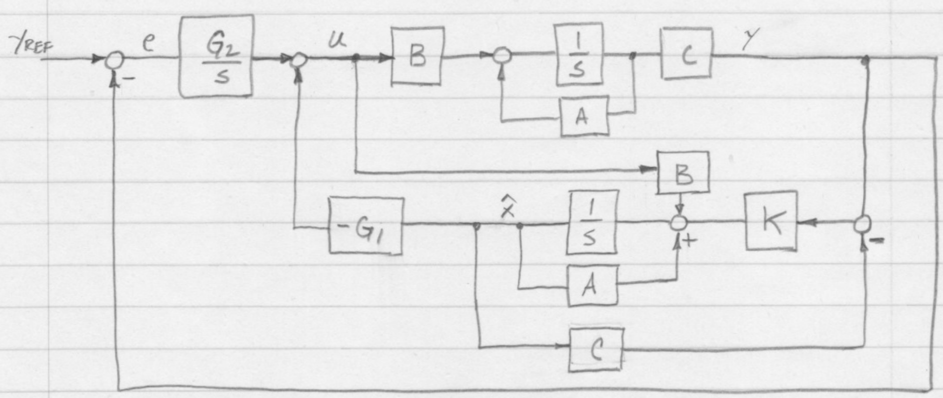
$$\text{sys\_G1} = ss(0, 0_{1xn}, 0, G_1)$$

$$\text{sys\_G2} = ss(0, C, G_2, 0)$$

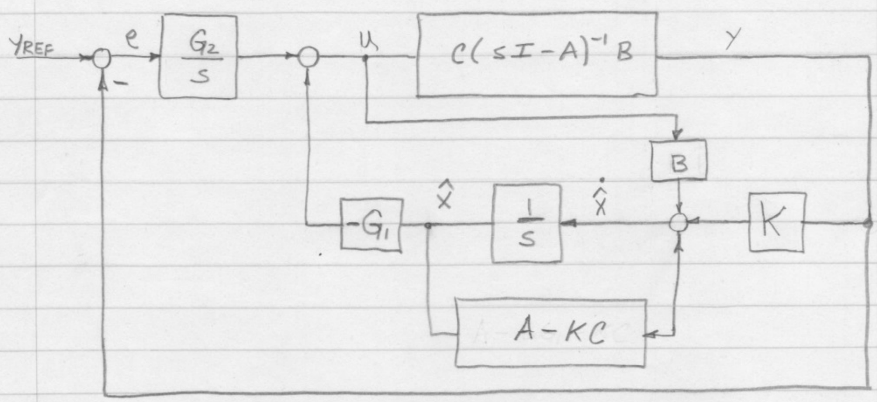
and then append that to the plant. Code to compute it with this method:

```
%% new method -- Compute the open loop response of the LQR Full-state Ind
% Reg controlled plant
G1 = G(1:2);
G2 = G(3);
sys_1 = ss(0,zeros(1,2),0,G1);
sys_2 = ss(0,C,G2,0);
sys_comp = parallel(sys_1, sys_2);
% the compensator frequency response
bode(sys_comp);
title('Bode plots -- LQR Full-state Industrial Regulator Compensator')
grid
% concatenate the compensator with the plant
sys_full_state_plant = ss(A, B, eye(2), zeros(2,1));
sys_ol = series(sys_full_state_plant, sys_comp);
% bode plot Full-state LQR Industrial Regulator open loop respone
bode(sys_ol)
title('Bode plots -- Open loop Full-state LQR Industrial Regulator Controlled plant')
grid
```

The block diagram of the LQG Industrial Regulator is given here:



Manipulating to simplify



Noting that $u = \frac{G_2}{s}(y_{REF} - y) - G_1 \hat{x}$

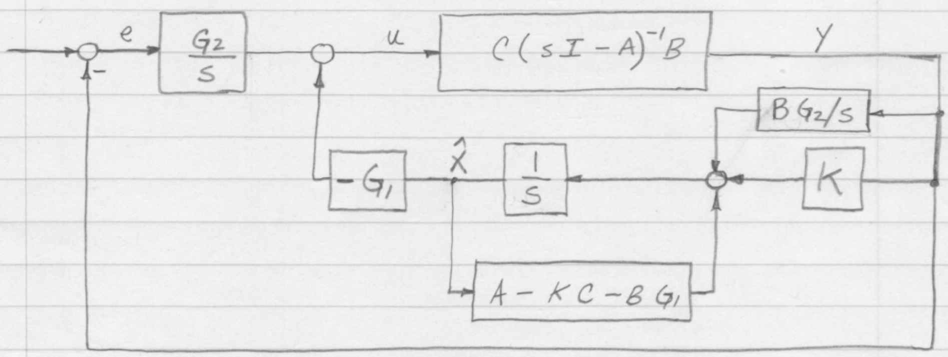
setting the input $y_{REF} = 0$ since it is not relevant to the computation of the open-loop frequency response. Then

$$u = -\frac{G_2}{s}y - G_1\hat{x}$$

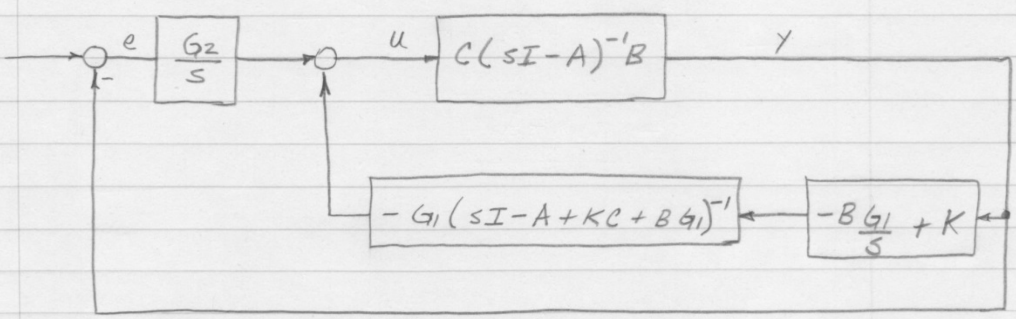
and premultiplying by B:

$$Bu = -BG_2\frac{y}{s} - BG_1\hat{x}$$

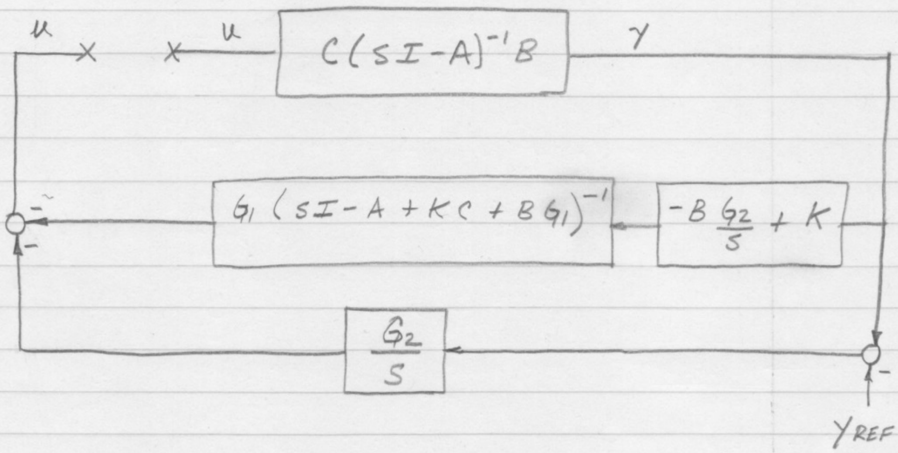
allowing further simplification of the block diagram:



which becomes



Rearranging blocks, moving y_{REF} and opening the loop at u:



So we can compute the freq. response of the compensator as follows:

$A_e = A - BG_1 - KC$

sys_1 = ss(A_e , I_n , G_1 , 0)

sys_2 = ss(0_{nxn} , -B*G_2 , I_n , K)

sys_3 = ss(0 , G_2 , 1 , 0)

sys_4 = series(sys_2, sys_1)

sys_comp = parallel(sys_4, sys_3)

and the open loop response of the controlled plant

sys_plant = ss (A, B, C, 0)

sys_ol = series (sys_plant, sys_comp)

Example code that realizes the above:

```
%%   Open-loop response of LQR/LQG Industrial Regulator (with observer)

sys_plant = ss(A,B,C,0);
bode(sys_plant)
grid
Ae = A - B*G1 - K*C;
sys_1 = ss(Ae, eye(2), G1, 0);
sys_2 = ss(zeros(2,2), -G2*B, eye(2), K);
sys_3 = ss(0, G2, 1, 0);
% concatenate sys_1 ans sys_2
sys_4 = series(sys_2,sys_1);
% sum sys_4 in parallel with sys_3
sys_comp = parallel(sys_4, sys_3);

% the compensator frequency response
bode(sys_comp);
title('Bode plots -- LQG/LQR Industrial Regulator Compensator')
grid

% concatenate the compensator with the plant
sys_ol = series(sys_comp,sys_plant);
% bode plot LQG/LQR Industrial Regulator open loop respone
bode(sys_ol)
title('Bode plots -- Open loop LQG-LQR Industrial Regulator Controlled plant')
grid
```

Integral control by incorporation of an exogenous bias state in an LQG design

We consider the 2nd order spring mass system governed by:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u$$

having position x_1 and velocity x_2. (In this A full-state regulator naturally provides a proportional-derivative type of control action, with the proportional coming from the gain on position, and the derivative from the gain on velocity. An integral control component is inserted by adding an additional state equation

$$\dot{b}=0$$

representing a bias term (a bias has a time derivative of zero), a bias given by the output of an integrator. We append this bias and have it inject an input at the location of the control input *u*. This causes the controller to introduce a compensating control input, i.e. the integral of the output error, to reject the effect of the bias. The system block diagram with the added bias is shown below, where we've indicated that the bias is

$\dot{b}=0$

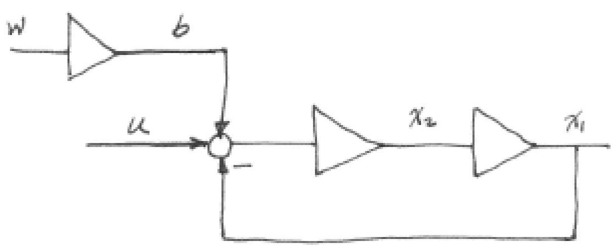


Figure A-1 – Augmentation of state dynamics for addition of integral control

The augmented model is therefore:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ -1 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}u$$

The full-state control design can be obtained for this problem through the solution of the Control Algebraic Ricatti Equation, however a solution is guaranteed to exist only for systems that are stabilizable. Thus the uncontrollable poles must be stable. In this example as show above, the new state x_3 is uncontrollable (check controllability test matrix). To make it stable we add a small negative feedback term α, making:

$$A=\begin{bmatrix}0 & 1 & 0\\ 0 & -1 & 1\\ 0 & 0 & -\alpha\end{bmatrix}$$

This results in an inconsequential change in the system dynamics, but enables the algebraic Riccati Equation solver to find a solution. For this pair {A,B} the controllability test matrix has rank 2. However, with α= 1e-4 the control algebraic Ricatti Equation has a unique solution that can be successfully derived numerically using Matlab's lqr function

```
%% LQR Full-State Controller with Integral Control Action
A = [ 0    1     0;  -.1    0     1;    0     0     -1e-4]
B = [ 0  1  0 ]'
Q = [1     0     0;  ...
     0     0     0;  ...
     0     0     0  ]
 R = .001
 [G,M,E] = lqr(A,B,Q,R,zeros(3,1))
```

```
which produces:

G = 31.5229    7.9401    0.9968

M =

    0.2511    0.0315   -0.0000
    0.0315    0.0079    0.0010
   -0.0000    0.0010    5.0000


E =

  -3.9701 + 3.9826i
  -3.9701 - 3.9826i
  -0.0001
```

Note that gain `K(3)`, which multiplies the bias state, is ~1. Thus the control acts to negate the bias, assumed to be know in the full-state regulator problem. This gain does not change appreciably as the weighting matrices Q and R vary in such a way as to increase the speed of response, moving the complex eigenvalues to the left in the s-plane; however for smaller Q/R ratios (ie when increasing R) this gain will move away from 1 as the optimal solution becomes one that permits state regulation error in favor of conserving control action. Try solving for the optimal gain `K` for lower values of R and note how the other gains change but this gain remains essentially at 1.

The bias state is not a known quantity in general; if it were then we would simply add that control action to *u* and be done with it. To generate the unknown bias state an observer is needed, and thus we develop the Optimal Observer to complete the development of the control law with integral control action. In this case of the filter design, the state dynamics and output equations involve noise signals *v(t)* and *w(t)*.:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x + v$$

Weighting matrices are defined which satisfy the controllability requirement on the pair {A,$W^{1/2}$}. The system must be controllable by the process noise *w*. We set the spectral density matrix so be the diagonal matrix W = diag([0 γ_1 γ_2]), whose square root is simply the square root of the diagonal terms. One can easily verify that the pair {A,$W^{1/2}$} is controllable. Note that the pair {A,C} is observable, thus the requirements for the existence of a solution to the Filter Algebraic Riccati Equation (FARE) . The following m-code, which assumes A is defined above:

```
%% Kalman Filter Design
 A(3,3)=0
 W = diag([0 10 .1]);
 V = 0.0001;
 C = [1 0 0];
 [L,P,E] = lqe(A,eye(3),C,W,V,ones(3,1))
```

yields the solution:

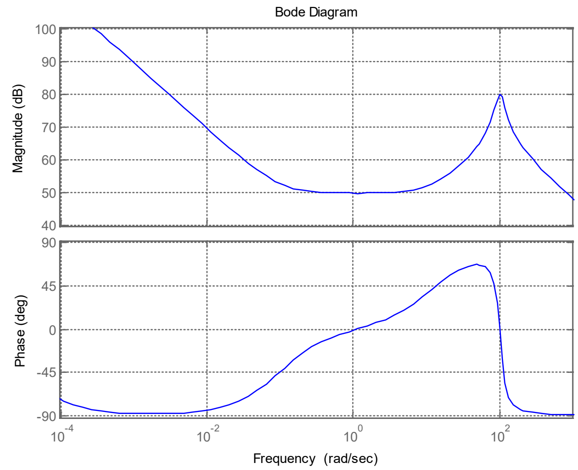
```
L =

  9.8920e-002
  1.0000e+004
  9.8921e+002


P =

 -9.9999e-001  4.8925e-007 -9.0108e-001
  4.8925e-007  9.0000e-001  9.7852e-003
 -9.0108e-001  9.7852e-003  9.1092e-001


E =

 -2.8200e-011 +2.2608e+000i
 -2.8200e-011 -2.2608e+000i
 -9.8920e-002
```

Compensator Equations

Earlier we learned that the compensator given by the combination of the observer and full-state regulator is given by:

$$u(s) = -\left[G(sI - A + BG + LC)^{-1} L\right] y(s)$$

Generating the frequency response of the compensator (Figure A-2) shows that there exists an integral control action in the lower frequency region followed, as frequency increases by a region of phase lead as needed to stabilize the system.



Adding Tracking Reference Signal

A tracking reference is added in order to cause the system to drive the output to a non-zero reference level, as is the case of a tracking servo. In this example the addition of a tracking reference signal is fairly trivial. A block diagram of the controlled system with the reference subtracted from the output is shown in Figure A-3a below, and in another equivalent form in (b).

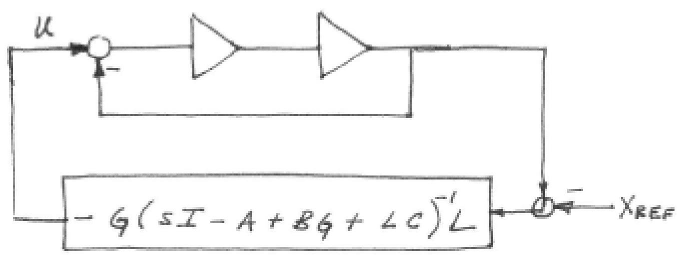


Figure A-3a

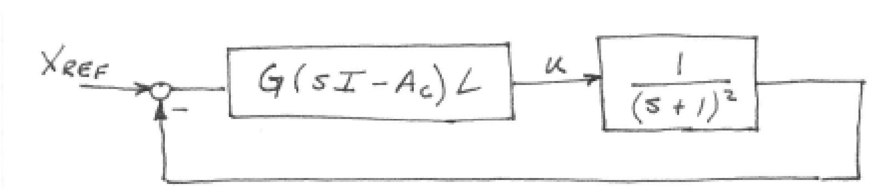


Figure A-3b – Tracking reference signal added

Optimal Observer (Kalman Filter)

Selection of V & W matrices

LINEAR QUADRATIC GAUSSIAN (LQG) CONTROL

MISCELLANEOUS TOPICS

DISCRETIZING A CONTINUOUS-TIME COMPENSATOR

Consider the compensator given as a continuous-time observer and full-state regulator:

$$\dot{\hat{x}} = A_L \hat{x} + Bu + Ly$$
$$u = -G\hat{x}$$

where $A_L = A - LC$. Conversion to discrete-time form:

$$\hat{x}_{n+1} = A_{Ld}\hat{x}_n + B_d u_n + L_d y_n$$
$$u_n = -G\hat{x}_n$$

is accomplished by assuming that over the time interval *T* the control *u(t)* and measurement *y(t)* are constant. The matrix A_{Ld} is known to be the state-transition matrix:

$$A_{Ld} = e^{A_L T}$$

Matrices B_d and K_d are given by:

$$B_d = \int_0^T e^{A_L \tau} B d\tau$$

$$L_d = \int_0^T e^{A_L \tau} L d\tau$$

You can compute these matrices using Matlab:

[A_{Ld}, B_d] = c2d(A_L, B, T)

[A_{Ld}, L_d] = c2d(A_L, L, T)

or compute them yourself as follows. A simple way to compute approximate discrete-time matrices is through Euler Integration. Taking the original equation expressed as follows:

$$\begin{aligned}
\hat{x}\big((n+1)T\big) &= \hat{x}\big(nT\big) + \int_{nT}^{(n+1)T} \dot{\hat{x}}(t)dt \\
&= \hat{x}\big(nT\big) + \int_{nT}^{(n+1)T} \big[A_L\hat{x}(t) + Bu(t) + Ly(t)\big]dt \\
&\approx \hat{x}\big(nT\big) + \big[A_L\hat{x}(nT) + Bu(nT) + Ly(nT)\big]T
\end{aligned}$$

Rewriting in simpler notation:

$$\hat{x}_{n+1} = \big(I + A_L T\big)\hat{x}_n + BTu_n + LTy_n$$

A better approximation can be derived using infinite series representation of the state transition matrix:

$$e^{AT} = I + AT + \frac{1}{2}A^2T^2 + \frac{1}{3!}A^3T^3 + \cdots$$

retaining 3 or 2 terms of the series to compute the state-transition matrix:

$$A_{Ld} \approx I + A_LT + \frac{1}{2}A_L^2T^2$$

$$A_{Ld} \approx I + A_LT$$

Using the latter in the integrals for the control and measurement input matrices:

$$\begin{aligned} B_d &= \int_0^T (I + At)Bd\tau \\ &= \left[I \cdot T + AT^2/2\right]B \end{aligned} \qquad \begin{aligned} L_d &= \int_0^T (I + At)Ld\tau \\ &= \left[I \cdot T + AT^2/2\right]L \end{aligned}$$

Rewriting here in simpler notation:

$$\hat{x}_{n+1} = \left(I + A_LT + A^2\frac{T^2}{2}\right)\hat{x}_n + \left(I \cdot T + A_L\frac{T^2}{2}\right)Bu_n + \left(I \cdot T + A_L\frac{T^2}{2}\right)Ly_n$$

ACCOMMODATING SIGNAL DELAYS

PADE' APPROXIMATION

INCORPORATING DELAYS IN SISO AND STATE-SPACE MODELS

Homework Problems

(6-1) Is the following system controllable? Verify by computation of the Controllability matrix and verification that the rank = 3

$$\dot{x} = \begin{bmatrix} -0.4 & 0 & -0.1 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix} x + \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix} u$$

(6-2) Design an observer for the system below using pole placement and hand calculations to place the closed loop poles at {-4±j4}. Verify your solution using the Matlab place(*A', C', p*), where *p* are the poles of the observer.

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

For initial conditions

$$x(0) = \begin{bmatrix} 1 & 2 \end{bmatrix} \qquad \hat{x}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

simulate with Simulink and plot the transient response of the system and observer for *u(t) = sin(t)*.

(6-3)

For the helicopter system given by the following:

$$\dot{x} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -5.6 & 40 & -.1 \end{bmatrix} x + \begin{bmatrix} 6.3 \\ 0 \\ 40 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x$$

where the state and control variables are:

x_1 – pitch rate (rad/sec),
x_2 – pitch angle of the fuselage (radians),
x_3 – horizontal velocity (ft/sec),
u – control input, rotor blade pitch angle (radians)

(a) Where are the open loop poles of the system?

(b) All of the state variables are available as inputs to the controller. Design a full-state regulator with pole placement. Use place (A, B, p) to place the closed loop poles at -1, -1.5, and -2 rad/sec. What is the feedback gain matrix, K?

(c) Develop a Simulink model of the helicopter as shown here. Add a setpoint to the horizontal velocity of 100 ft/sec in the model and simulate the step response to this input. In other words, set the initial condition *x(0)* to [0 0 0] in the state-space block. Record the horizontal velocity and control input using scope blocks. Copy them to you document along with your Simulink model and any m-code generated.

(6-4) Industrial Regulator Design for Helicopter Control:

– Generate a control law for the HELO by adjusting the Q & R matrices such that the rise time is approximately 10 second and the pitch angle transient limit of 25 degrees max is achieved.

– Plot out the 4 plots as shown in the lecture.

– Suggestion – weight state x_2, the pitch angle – Why?

```
Q = [0    0      0     0
     0    q22    0     0
     0    0      100   0
     0    0      0     10];
```