2.2 Controllability and Observability .

A system

$$\underline{\dot{x}} = A\underline{x} + \underline{B}\,\underline{u}$$

is said to be completely controllable if it is possible to take the system from any initial state $\underline{x(t_0)}$ to any final state $\underline{x(t_f)}$ in a finite amount of time using a piece-wise continuous forcing function $\underline{u(t)}$, $t_0 \le t \le t_f$

A system

$$\underline{\dot{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$$
$$\underline{y} = \underline{C}\,\underline{x}$$

is said to be completely observable if, given any $\underline{u(t)}$, $t_0 \le t \le t_f$, the matrices $\underline{A}$, $\underline{B}$ and $\underline{C}$ and the vector $\underline{y}(t)$ for $t_0 \le t \le t_f$ are sufficient to determine $\underline{x(t_0)}$.

The concepts of controllability and observability are important in the discussion of so-called optimal feedback controllers (optimal regulators) and optimal state estimators. These will be discussed in another chapter.

The methods for discerning whether a given system is completely controllable or observable are quite simple and will be stated here without proof.

A system

$$\underline{\dot{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$$

is completely controllable if and only if the partitioned matrix

$$\left[\underline{B} \;\vdots\; \underline{A}\,\underline{B} \;\vdots\; \underline{A}^2\underline{B} \;\vdots\; \cdots \;\vdots\; \underline{A}^{n-1}\,\underline{B}\right]$$

has rank n, where n is the order of the system.*

A system

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$$

$$\underline{y} = \underline{C}\,\underline{x}$$

is completely observable if and only if the partitioned matrix

$$\left[\underline{C}^T \mid \underline{A}^T\underline{C}^T \mid (\underline{A}^T)^2\underline{C}^T \mid \ldots \mid (\underline{A}^T)^{n-1}\underline{C}^T\right]$$

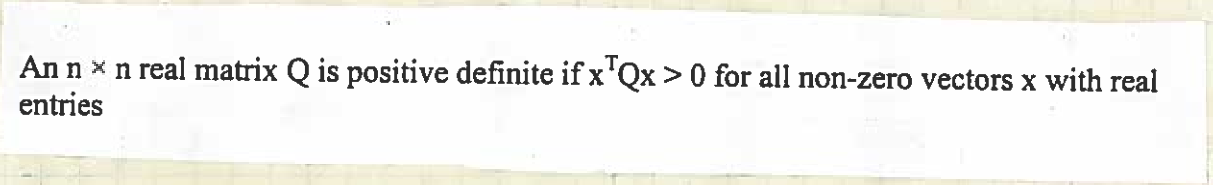
is of rank n, where n is the order of the system.

LQR design
(linear quadratic regulator)

Consider Index of Performance

$$J = \int_0^\infty (\underline{x}^T \underline{Q}\,\underline{x} + \underline{u}^T \underline{R}\,\underline{u})\,dt$$

$\underline{Q}$ & $\underline{R}$ are symmetric & positive definite (will consider them diagonal & positive definite here)


An n × n real matrix Q is positive definite if $x^TQx > 0$ for all non-zero vectors x with real entries

for a controllable system $\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$

solution $\underline{u}$ which minimizes J is

$$\underline{u}(t) = -\underline{R}^{-1}\underline{B}^T\underline{S}\,\underline{x}(t) = -\underline{K}\,\underline{x}(t)$$

where $\underline{S}$ is solution of matrix Riccati Eqn:

$$0 = \underline{S}\,\underline{A} + \underline{A}'\underline{S} + \underline{Q} - \underline{S}\,\underline{B}\,\underline{R}^{-1}\underline{B}^T\underline{S}$$

e.g. $J = \int_0^\infty x_1^2 q_{11} + x_2^2 q_{22} + x_3^2 q_{33} + x_4^2 q_{44} + u_1^2 r_{11}^2$

Q & R must be positive definite

A scalar function

$$V = V(x_1, x_2, \cdots, x_n)$$

is said to be positive (negative) definite in a given region about the origin if at all points in this region V is positive (negative) and, except at the origin, is nowhere zero.

The function is said to be positive (negative) semidefinite if it is postive (negative) throughout the region except at certain points at which it is zero. It must be zero at the origin.

The function is said to be indefinite if in the given region about the origin, it takes on varying signs.

A symmetric matrix $\underline{Q}$ is sign definite, semidefinite or indefinite if the quadratic form $\underline{x}^T \underline{Q}\, \underline{x}$ is sign definite, semidefinite or indefinite.

A theorem known as Sylvester's Theorem states that a symmetric matrix $\underline{Q}$ is positive definite if all the principal minors of the matrix are positive. It is negative definite if $-\underline{Q}$ is positive definite.

Example

The matrix

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

is positive definite if

$$q_{11} > 0$$

$$\begin{vmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{vmatrix} > 0$$

$$\begin{vmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{vmatrix} > 0$$

```
LQR  Linear-quadratic regulator design for state space systems.

   [K,S,E] = LQR(SYS,Q,R,N) calculates the optimal gain matrix K
   such that:

     * For a continuous-time state-space model SYS, the state-feedback
       law u = -Kx  minimizes the cost function

             J = Integral {x'Qx + u'Ru + 2*x'Nu} dt

       subject to the system dynamics  dx/dt = Ax + Bu

       The matrix N is set to zero when omitted.  Also returned are the
          the solution S of the associated algebraic Riccati equation and
          the closed-loop eigenvalues E = EIG(A-B*K).
```

k = gains

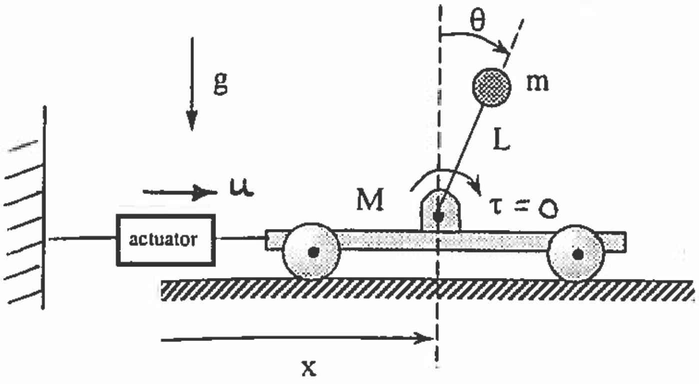
S = colution to Riccati eqn

E = closed-loop eigenvalues

UNIVERSITY OF CALIFORNIA, DAVIS
Dept. of Mechanical and Aerospace Engineering

MAE – 274

The device below is the classic problem of an inverted pendulum on a cart. The idea here is to move the cart and pendulum from some initial position ($x = x_0$, $\theta = \theta_0$) to a desired final position ($x = 0$, $\theta = 0$), within some desired time interval ($0 < t < t_f$) using only a horizontal force applied to the cart (u). NO EXTERNALLY APPLIED PENDULUM TORQUE (τ) WILL BE USED.



System parameters:	
Cart mass	= M
Pendulum mass	= m
Pendulum length	= L
State variables:	
Cart position	= x
Pendulum angle	= θ
Control inputs:	
Horizontal Force	= u
Pendulum Torque	= τ = 0

The equations of motion, linearized about the equilibrium point $\theta = \theta_0 = 0$ can be given by:

Cart mass = 3 kg
Pendulum mass = 0.5 kg
Rod length = 0.4 m

$$z_1 = x \quad (\mathrm{m})$$
$$z_2 = \theta \quad (\mathrm{rad})$$
$$z_3 = \dot{x} \quad (\mathrm{m/sec})$$
$$z_4 = \dot{\theta} \quad (\mathrm{rad/sec})$$
$$u = \text{force (newtons)}$$

$$\begin{bmatrix}\dot z_1\\ \dot z_2\\ \dot z_3\\ \dot z_4\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -1.6345 & -2 & 0.0042\\ 0 & 28.6037 & 5 & -0.0729\end{bmatrix}}_{\underline{A}}\begin{bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ 0.3333\\ -0.8333\end{bmatrix}}_{\underline{B}}[u]$$

$$\underline{C} = [I]_{4\times4} \qquad \underline{D} = \left\{\begin{matrix}0\\0\\0\\0\end{matrix}\right\}$$

```
>> A

A =

         0         0    1.0000         0
         0         0         0    1.0000
         0   -1.6345   -2.0000    0.0042
         0   28.6037    5.0000   -0.0729

>> B

B =

         0
         0
    0.3333
   -0.8333

>> C

C =

     0
     0
     0
     0

>> D

D =

     0

>>
```

4.1

```
>> nn=ctrb(A,B)

nn =

         0    0.3333   -0.6701    2.7095
         0   -0.8333    1.7272  -27.3119
    0.3333   -0.6701    2.7095   -8.3569
   -0.8333    1.7272  -27.3119   64.9441

>> rank(nn)

ans =

     4
```

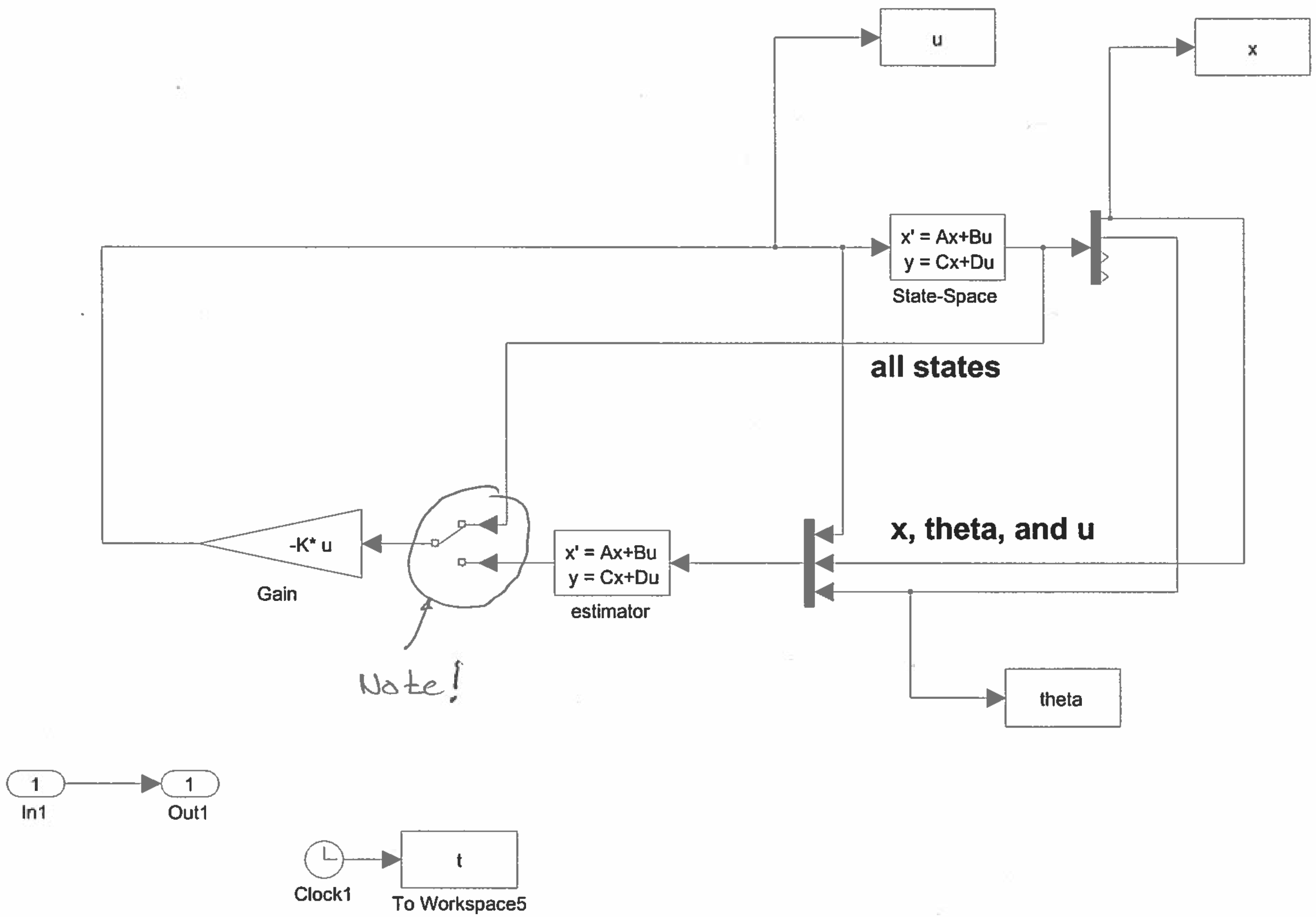
Inverted pendulom

Let $\underline{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\boxed{\underline{R} = 1}$$

$SYS = ss(A, \underline{B}, \underline{C}, D)$

```
>> [K,S,E]=lqr(sys,Q,R)

K =

   -1.0000  -94.4247  -13.0184  -17.9590


S =

    7.0189   18.1994    6.1286    3.6513
   18.1994  858.4174  124.0332  162.9244
    6.1286  124.0332   20.1543   23.6840
    3.6513  162.9244   23.6840   31.0246


E =

   -5.8918
   -4.9429
   -1.6992
   -0.1651
```


u
x
x' = Ax+Bu
y = Cx+Du
State-Space
all states
x, theta, and u
-K* u
Gain
x' = Ax+Bu
y = Cx+Du
estimator
Note!
theta
1
In1
1
Out1
Clock1
t
To Workspace5

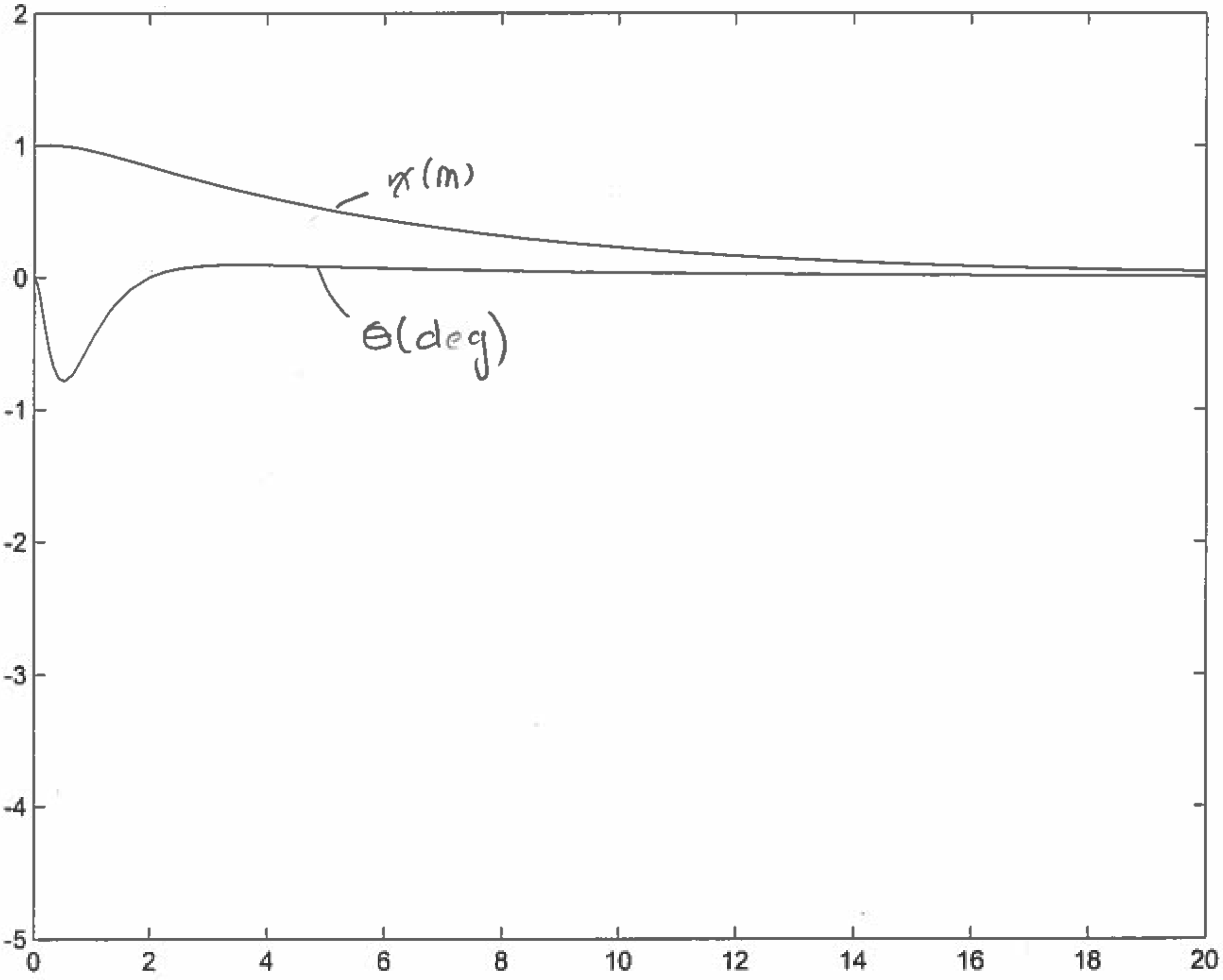
6.1

I. C. $x = 1\ m$

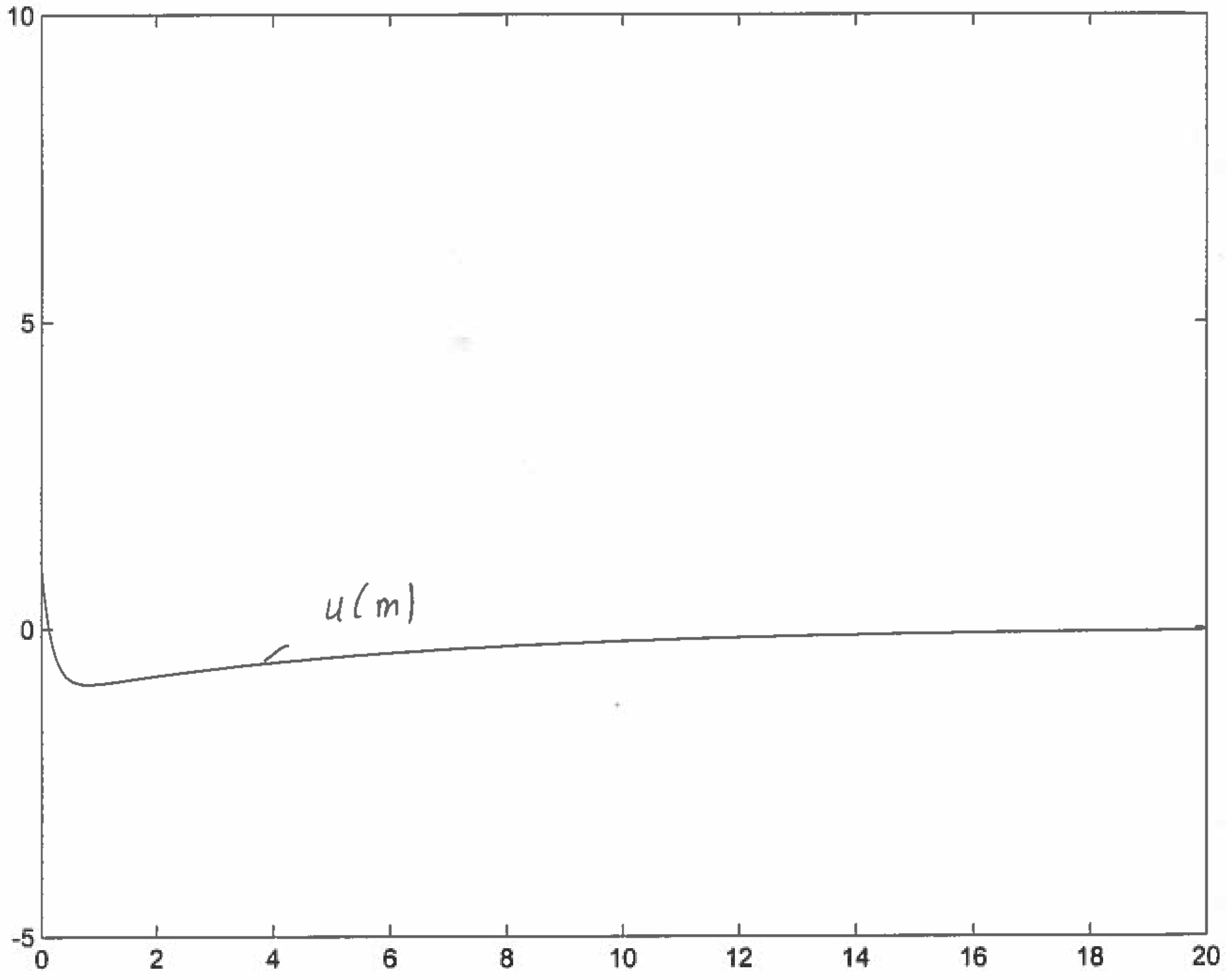
$\theta = 0$

$\dot{x} = 0$

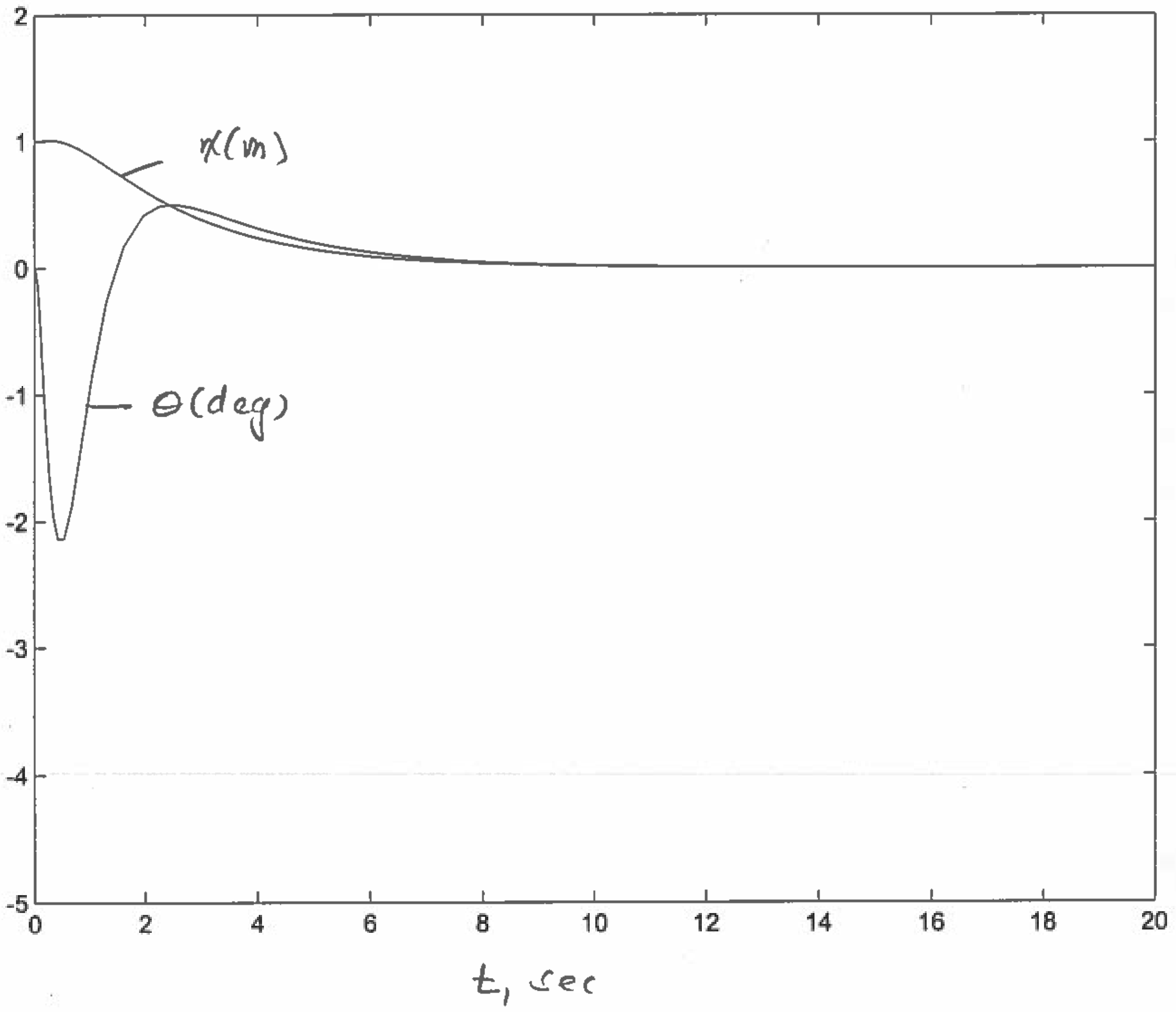
$\dot{\theta} = 0$

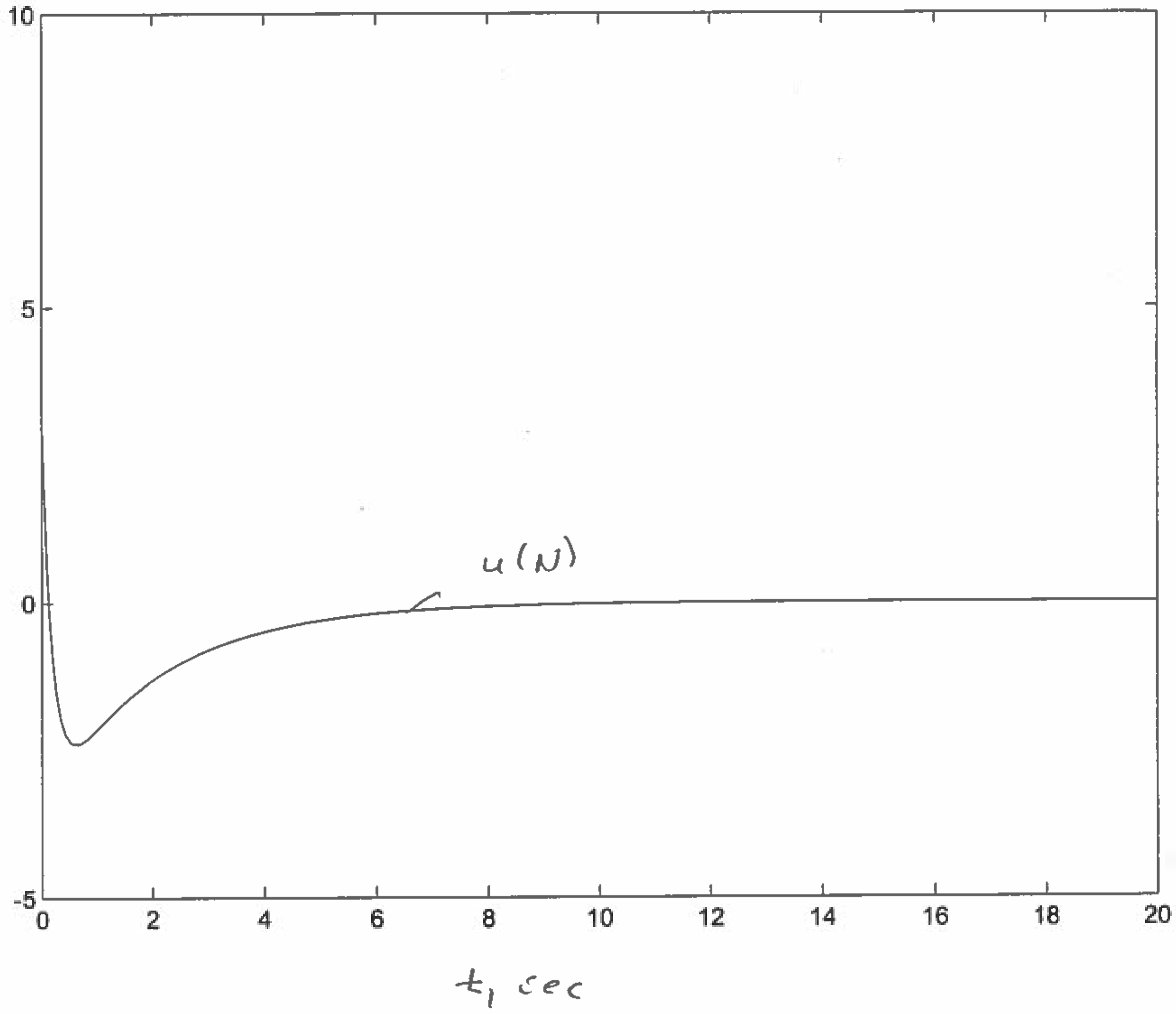


t, sec


10
5
0
-5
0
2
4
6
8
10
12
14
16
18
20
u(m)

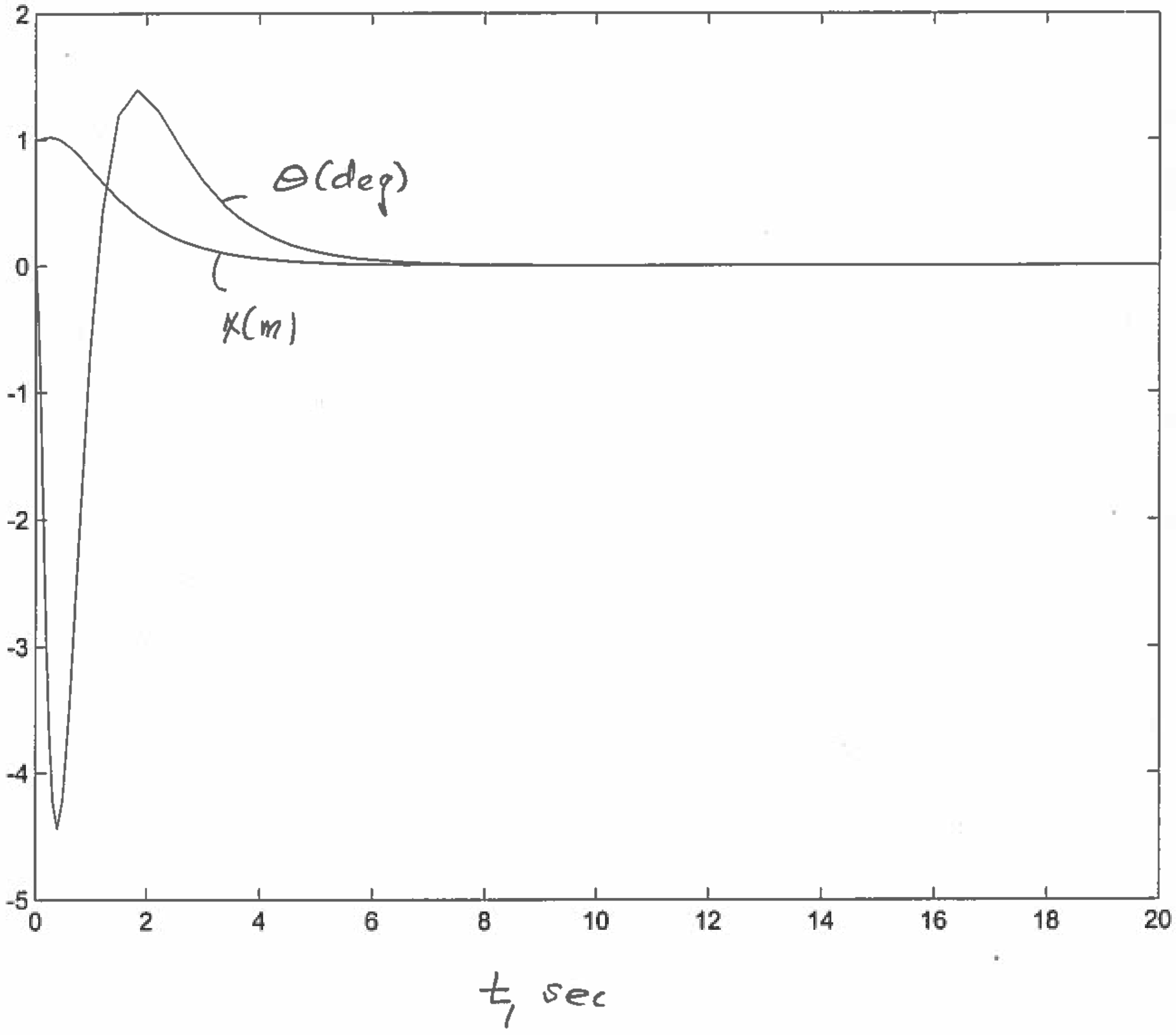
```
Q =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> R=.1*R → Note

R =

    0.1000

>> [K,S,E]=lqr(sys,Q,R)

K =

   -3.1623 -102.7493  -15.4936  -19.8476


S =

    3.0023    6.3591    2.2065    1.2620
    6.3591  103.7140   17.8126   19.4550
    2.2065   17.8126    3.8754    3.4094
    1.2620   19.4550    3.4094    3.7455


E =

   -6.9557
   -4.1525
   -1.8583
   -0.4815
```


2
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12
14
16
18
20
x(m)
θ(deg)
t, sec

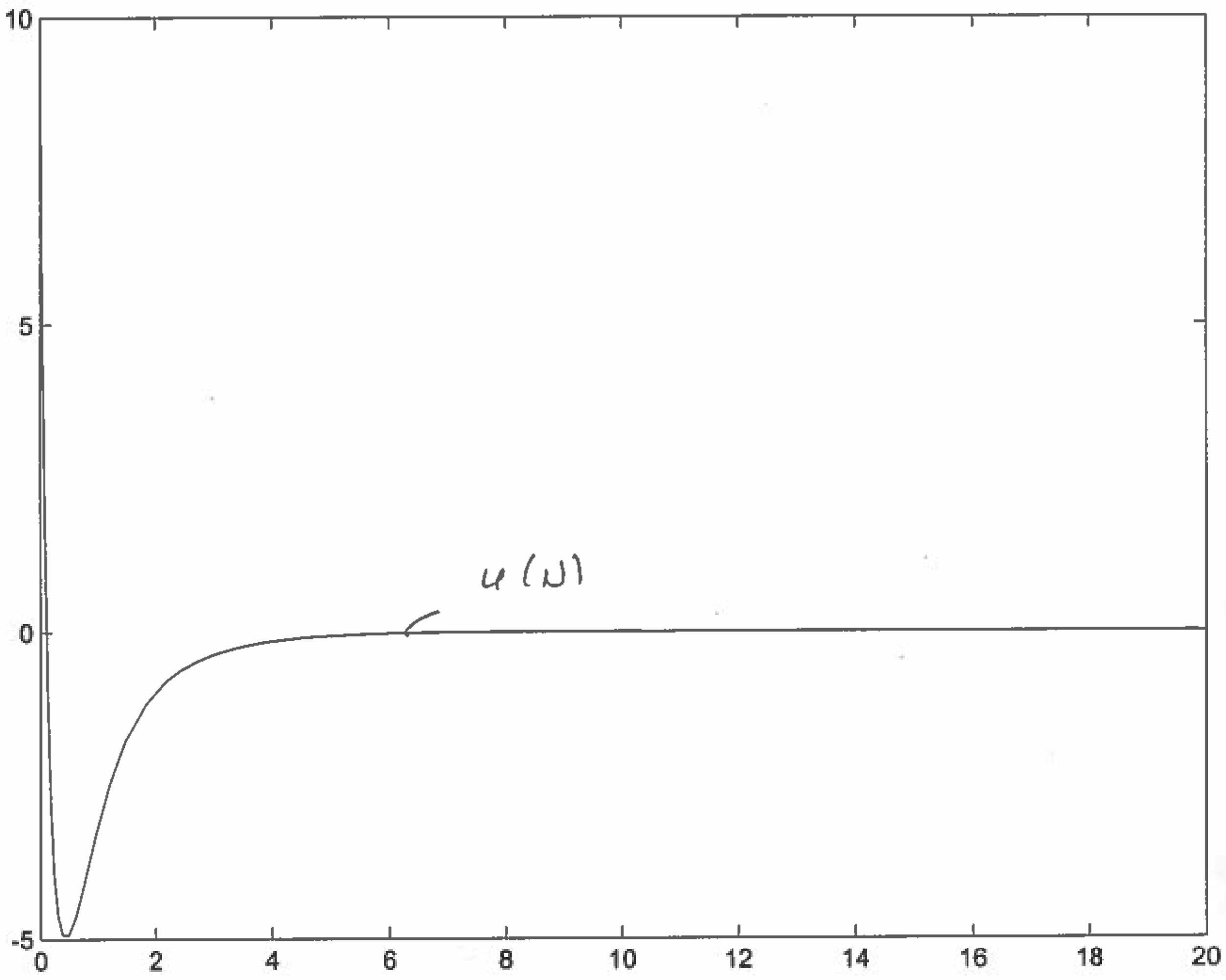

10
5
0
-5
0
2
4
6
8
10
12
14
16
18
20
u(N)
t, sec

```
Q =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> R=.1*R        Note

R =

    0.0100

>> [K,S,E]=lqr(sys,Q,R)

K =

  -10.0000 -137.7483  -24.6255  -28.3794


S =

    1.8626    2.8730    1.0547    0.5418
    2.8730   20.9612    4.8020    3.5737
    1.0547    4.8020    1.4614    0.8800
    0.5418    3.5737    0.8800    0.6926


E =

 -11.3900
  -2.6164 + 1.0990i
  -2.6164 - 1.0990i
  -0.8908

>>
```

R = .01



R = .01



t, sec

Estimator Design

Assume a reasonable model of our system:

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$$

Assume $\underline{u}$ is available & we construct an electronic model of our system

$$\dot{\hat{\underline{x}}} = \underline{A}\,\hat{\underline{x}} + \underline{B}\,\underline{u} \qquad \hat{\underline{x}} \rightarrow \text{"estimate of" } \underline{x}$$

By using actual $\underline{u}$ & known initial condition, we could generate $\hat{\underline{x}}$.

But modeling errors, etc, would soon cause <u>poor</u> estimates.

∴ we create a <u>feedback</u> system

$$\dot{\hat{\underline{x}}} = \underline{A}\,\hat{\underline{x}} + \underline{B}\,\underline{u} + \underline{K}^T \boxed{(\underbrace{\underline{y}}_{\text{measured output vector}} - \underline{C}\,\hat{\underline{x}})} \quad \text{estimate error}$$

or
$$\dot{\hat{\underline{x}}} = (\underline{A} - \underline{K}^T\underline{C})\,\hat{\underline{x}} + \underline{B}\,\underline{u} + \underline{K}^T\underline{y}$$

$$\dot{\hat{\underline{x}}} = \underline{A}_{obs}\,\hat{\underline{x}} + \underline{B}_{obs}\begin{Bmatrix}\underline{u}\\ \underline{y}\end{Bmatrix} \qquad \underline{B}_{obs} = [\,\underline{B} \,\vdots\, \underline{K}^T\,]$$

$\underline{K}$ = estimator gain matrix

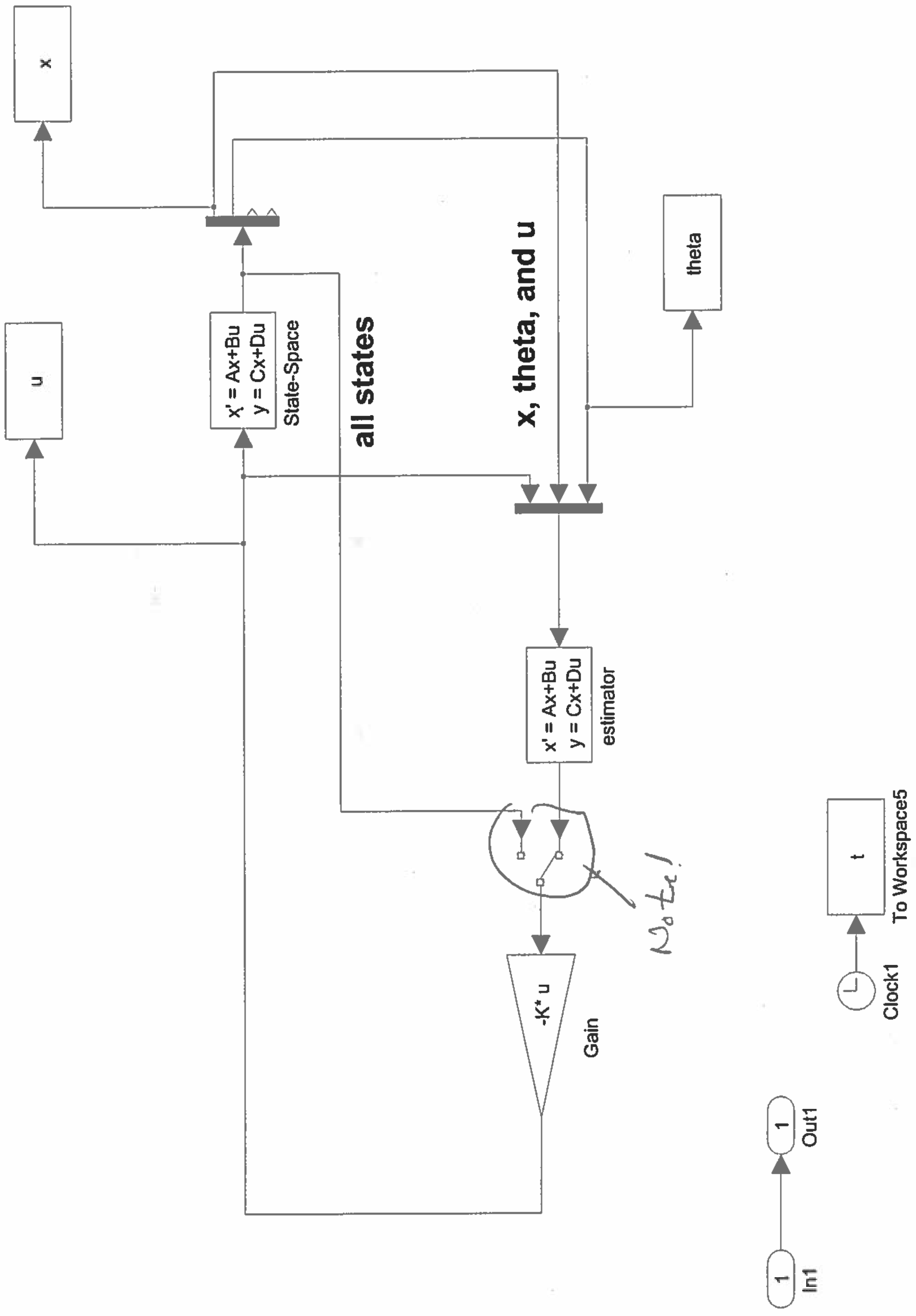
Can determine $\underline{K}$ in MATLAB using "place" command with <u>desired</u> estimator root locations <u>assuming</u> <u>observability</u>.

Now assume only x & θ can be measured

$$\underline{\dot{\hat{x}}} = \underline{A}\,\underline{\hat{x}} + \underline{B}\,\underline{u} + K^{T}(y - C\hat{x})$$

$$= \underbrace{(A - K^{T}C^{T})}_{\bar{A}_{OBS}}\,\underline{x} + \underbrace{(\underline{B} \mid K^{T})}_{\bar{B}_{obs}} \left\{\begin{array}{c} u \\ \hline y \end{array}\right\}$$

```
>> nn=obsv(A,C)

nn =

    1.0000         0         0         0
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
         0   -1.6345   -2.0000    0.0042
         0   28.6037    5.0000   -0.0729
         0    3.3891    4.0210   -1.6432
         0  -10.2577  -10.3645   28.6300

>> rank(nn)

ans =

     4
```


x
u
theta
x' = Ax+Bu
y = Cx+Du
State-Space
all states
x, theta, and u
x' = Ax+Bu
y = Cx+Du
estimator
-K* u
Gain
Note!
t
To Workspace5
Clock1
1
Out1
1
In1

```
>> Ac

Ac =

         0         0    1.0000         0
         0         0         0    1.0000
         0   -1.6345   -2.0000    0.0042
         0   28.6037    5.0000   -0.0729

>> Bc

Bc =

         0
         0
    0.3333
   -0.8333

>> Cc

Cc =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> Dc

Dc =

     0
     0
     0
     0

>> Cc1

Cc1 =

     1     0     0     0
     0     1     0     0
```

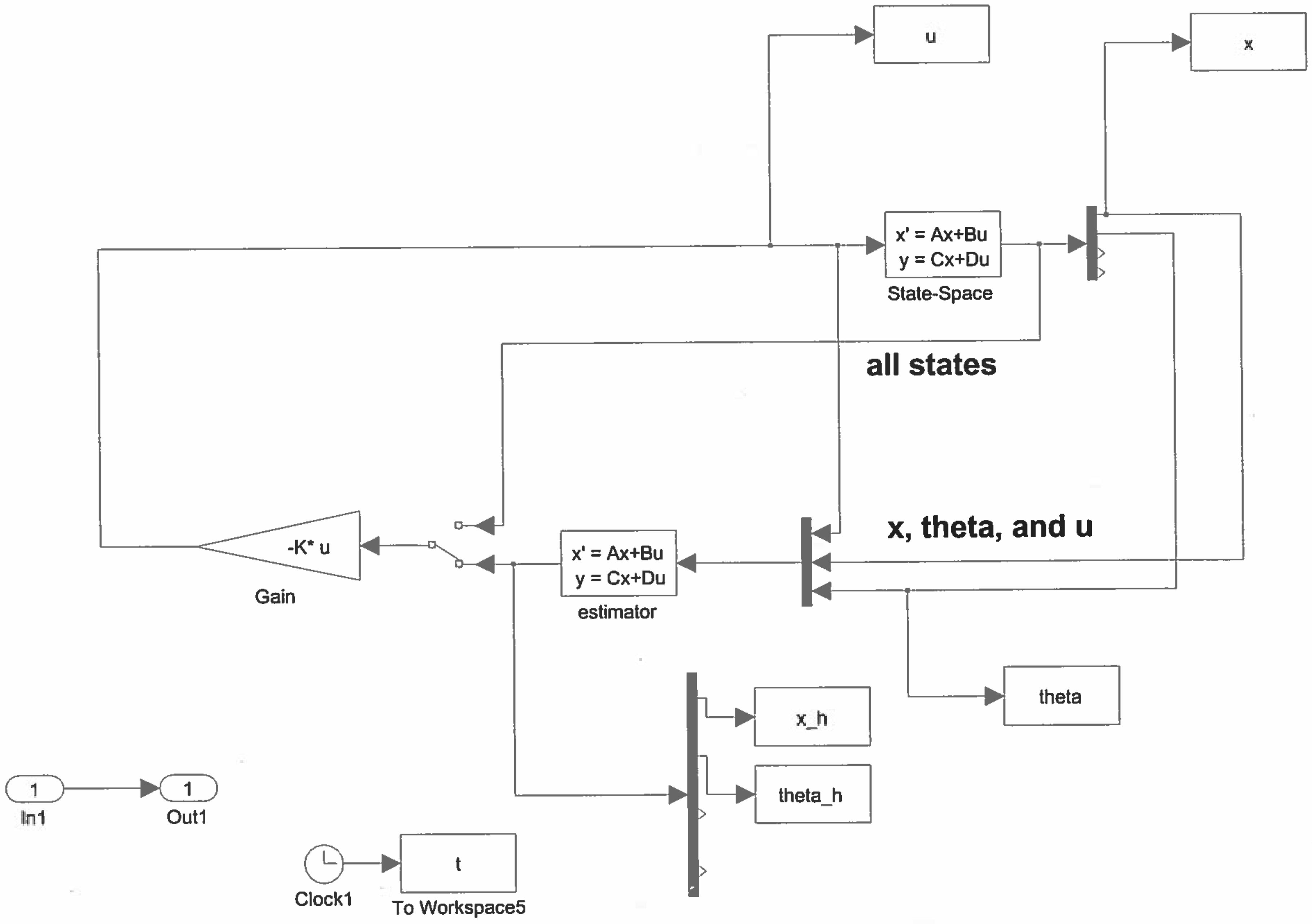
only measuring x + θ

```
>> Cobs

Cobs =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
>> Dobs

Dobs =

     0     0     0
     0     0     0
     0     0     0
     0     0     0

>>
```

R = .01

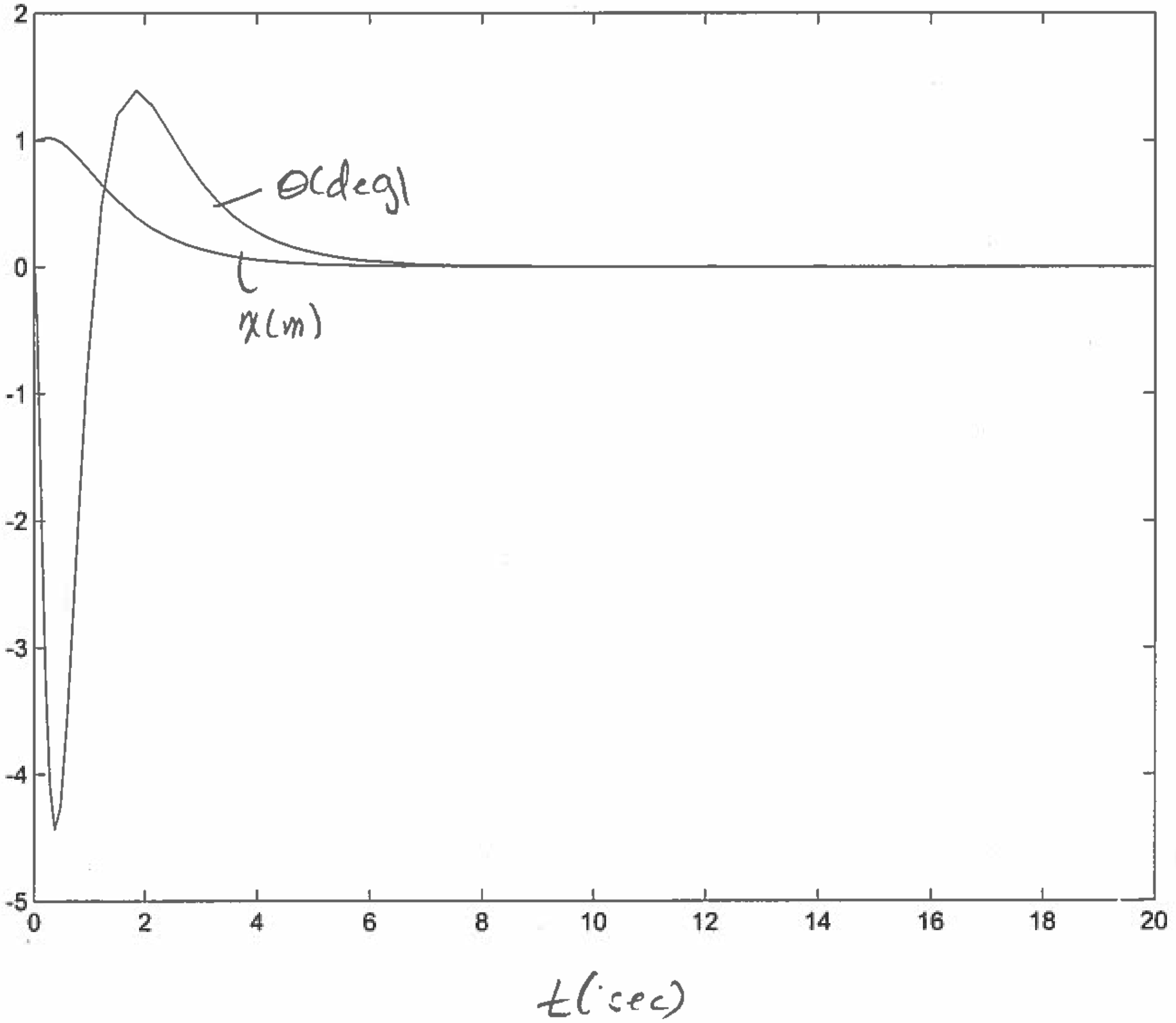
```
lambda =

  -15.0000  -15.1000  -15.2000  -15.3000

>> k=place(Ac',Cc1',lambda)

k =

   28.3176    5.1300  173.1674  143.1836
    0.0787   30.2095   -0.5367  256.0438

>> Aobs=Ac-k'*Cc1

Aobs =

  -28.3176   -0.0787    1.0000         0
   -5.1300  -30.2095         0    1.0000
 -173.1674   -1.0978   -2.0000    0.0042
 -143.1836 -227.4401    5.0000   -0.0729

>> eig(Aobs)

ans =

  -15.0000
  -15.3000
  -15.1000
  -15.2000


>> Bobs=[Bc k']

Bobs =

         0   28.3176    0.0787
         0    5.1300   30.2095
    0.3333  173.1674   -0.5367
   -0.8333  143.1836  256.0438


Cobs =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> Dobs

Dobs =

     0     0     0
     0     0     0
     0     0     0
     0     0     0

>>
```

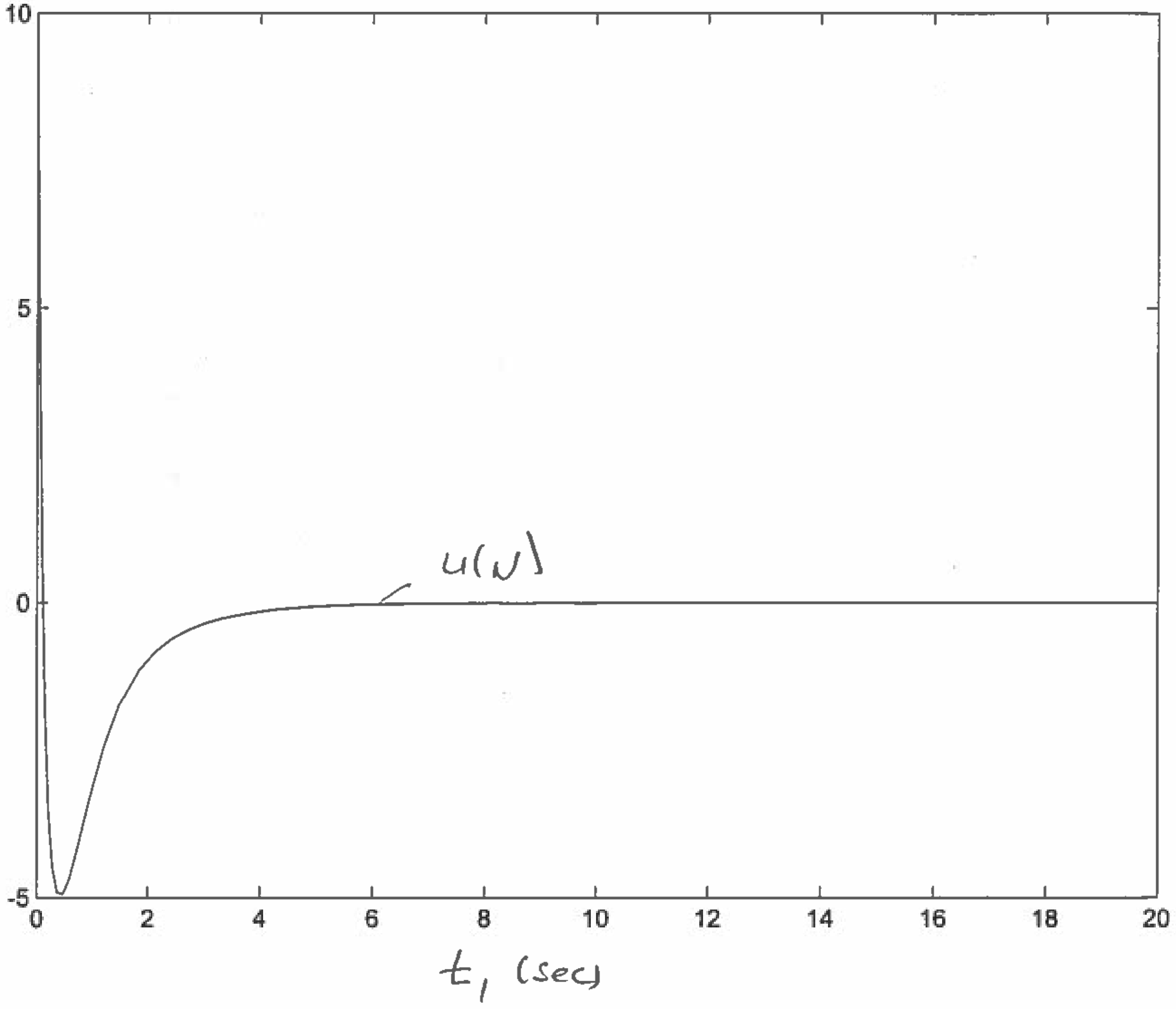

u
x
x' = Ax+Bu
y = Cx+Du
State-Space
all states
x, theta, and u
-K* u
Gain
x' = Ax+Bu
y = Cx+Du
estimator
theta
x_h
theta_h
1
In1
1
Out1
Clock1
t
To Workspace5

19.1

$R = .01$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$




10
5
0
-5
0
2
4
6
8
10
12
14
16
18
20
u(N)
t, (sec)

MAE 275

Creating a normal acceleration (n_z) control system for an F-16 using a "modern" output feedback approach and a "classical" loop shaping approach.

Aircraft: F-16

Flight Condition: Sea Level, U0 = 502 ft/sec

The vehicle model is taken from:

Stevens, B. L., and Lewis, F. L, *Aircraft Control and Simulation,* Wiley, 2nd Edition, pp.428-433.

```
>> A16

A16 =

   -1.0189    0.9051         0
    0.8223   -1.0774         0
         0    1.0000         0

>> B16

B16 =

   -0.0022
   -0.1756
         0

>> C16

C16 =

 -511.4700  -47.6400         0
         0         0    1.0000

>> D16

D16 =

   -1.0790
         0

>> eig(A16)

ans =

         0
   -1.9113
   -0.1850

>>
```

$$\{x\} = \begin{Bmatrix} \alpha \\ q \\ \theta \end{Bmatrix}$$

$$u = \delta_e$$

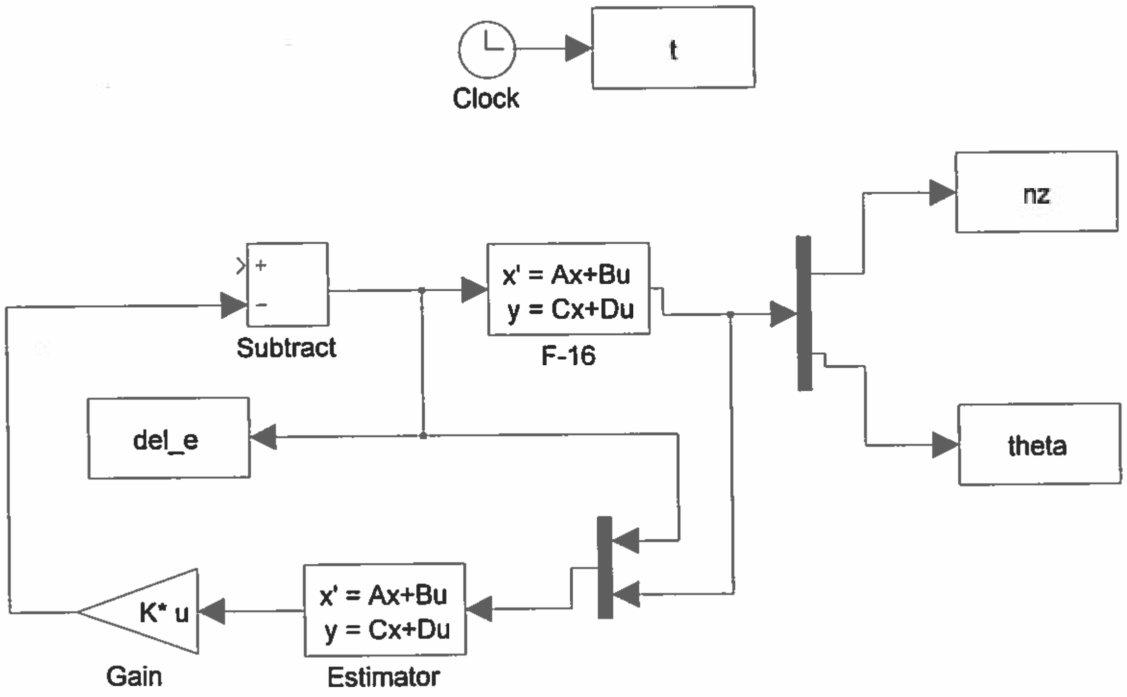
$$y = \begin{Bmatrix} nz \\ \theta \end{Bmatrix}$$

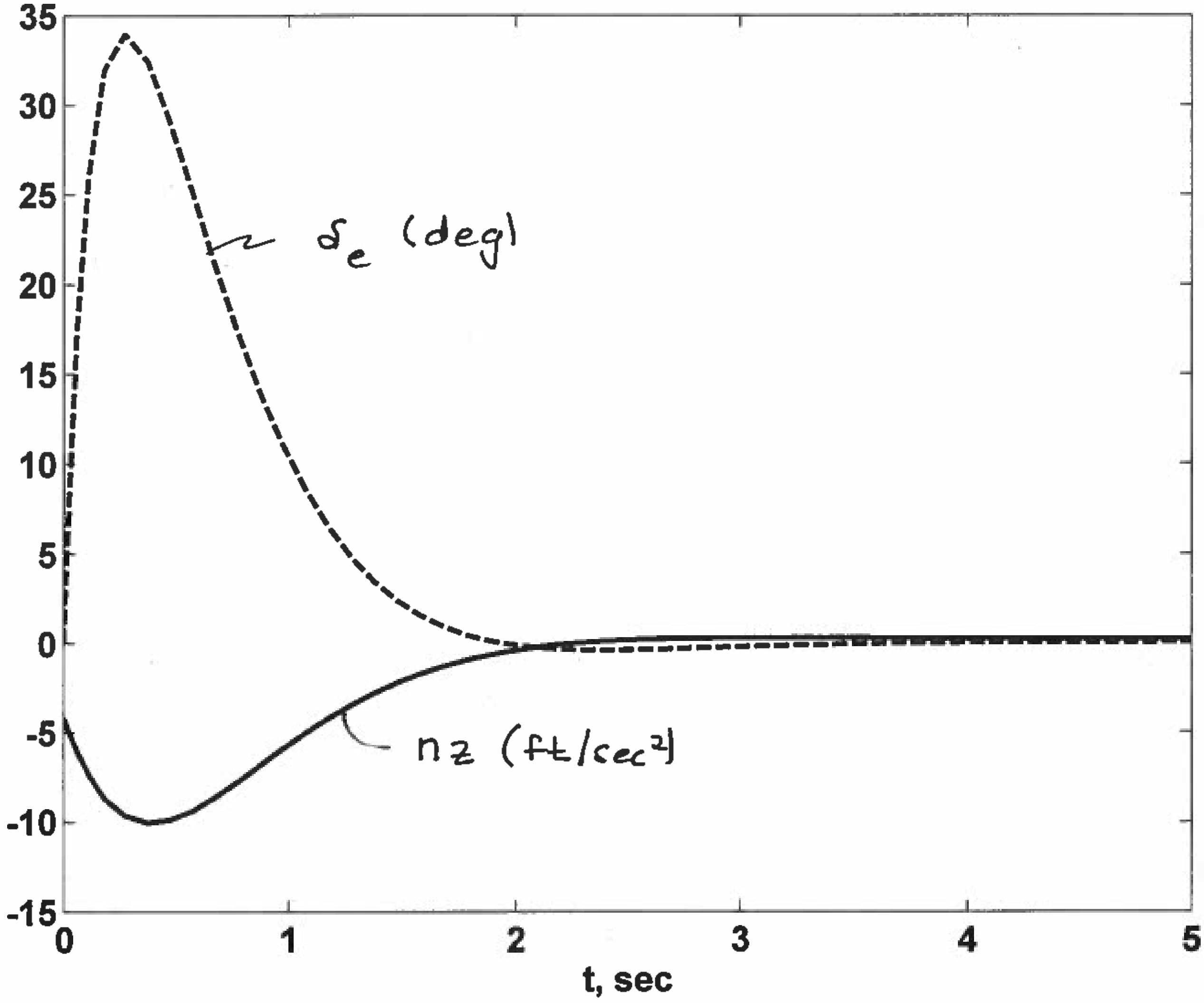
```
>> nn=ctrb(A16,B16);
>> rank(nn)

ans =

     3

>> sys=ss(A16,B16,C16,D16);
>> nn=obsv(sys)

nn =

 -511.4700  -47.6400         0
         0         0    1.0000
  481.9494 -411.5832         0
         0    1.0000         0
 -829.4681  879.6371         0
    0.8223   -1.0774         0

>> rank(nn)

ans =

     3

>> C161=C16(1,:); } only measuring nz
>> D161=D16(1,:); }
>> sys=ss(A16,B16,C161,D161);
>> nn=obsv(sys)

nn =

 -511.4700  -47.6400         0
  481.9494 -411.5832         0     rank(nn) = 2
 -829.4681  879.6371         0

>> Q16

Q16 =

     1     0     0
     0     2     0
     0     0     1

>>
```

```
>> R16

R16 =

     5.

>> [K,S,E]=lqr(sys,Q16,R16)

K =

 -6.9891e-001 -9.5933e-001 -4.4721e-001


S =

   1.5207e+001  1.9720e+001  8.6600e+000
   1.9720e+001  2.7082e+001  1.2631e+001
   8.6600e+000  1.2631e+001  7.9166e+000


E =

 -1.9129e+000
 -1.7664e-001 +1.0502e-001i
 -1.7664e-001 -1.0502e-001i

>>
```

```
>> k=place(A16',C16',[-10 -10.2 -10.2]);
>> A16obs=A16-k'*C16;
>> eig(A16obs)

ans =

 -1.0200e+001
 -1.0000e+001
 -1.0200e+001

>> B16obs=[B16 k']

B16obs =

 -2.1500e-003 -1.8382e-002  2.4435e-001
 -1.7555e-001 -7.9870e-002  4.4296e+001
            0 -8.6516e-003  1.5097e+001

>> C16obs=[1 0 0;0 1 0;0 0 1];
>> C16obs=[1 0 0;0 1 0;0 0 1]

C16obs =

     1     0     0
     0     1     0
     0     0     1

>> D16obs

D16obs =

     0     0     0
     0     0     0
     0     0     0

>>
```

modern

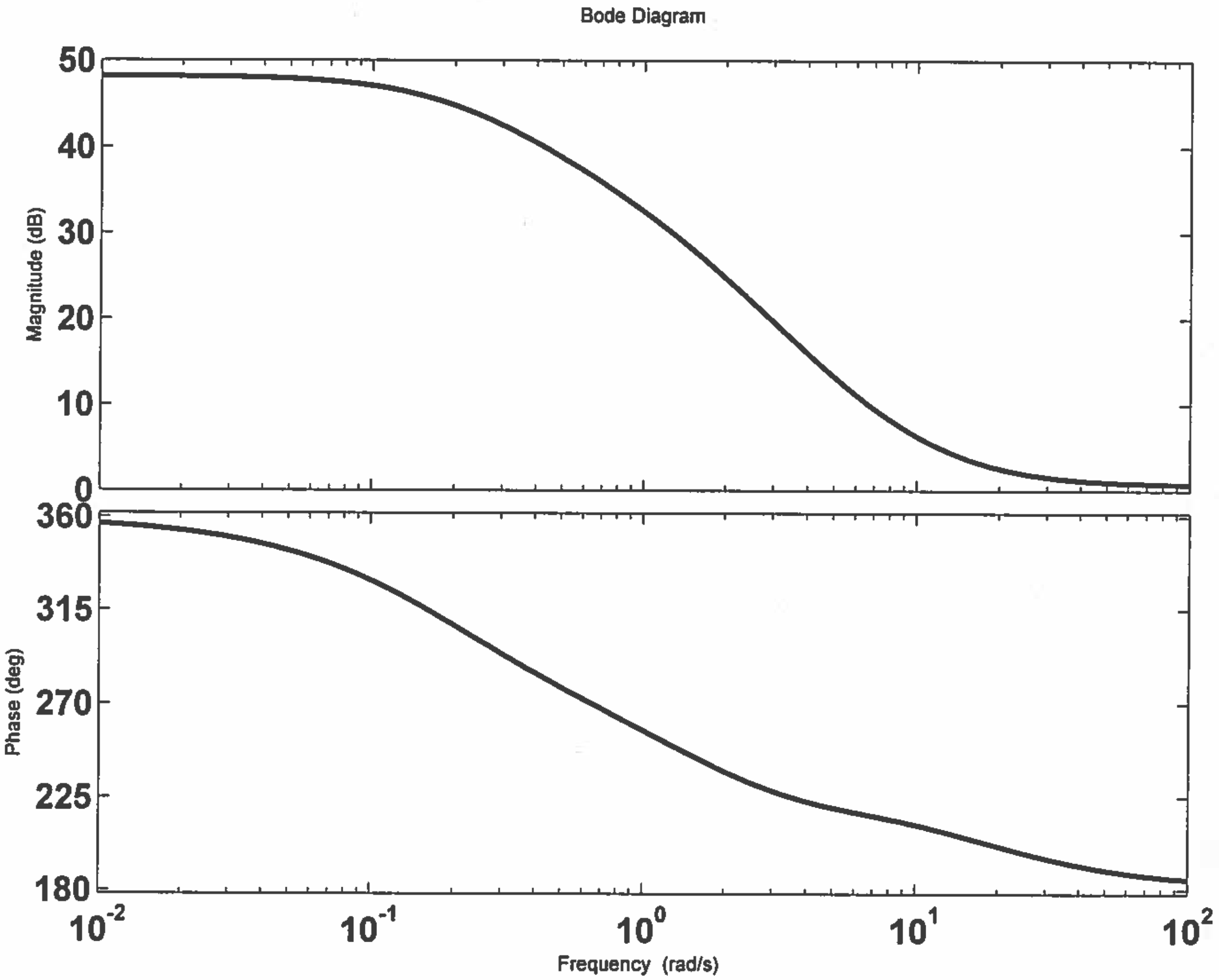

Clock
t
nz
theta
+
−
Subtract
x' = Ax+Bu
y = Cx+Du
F-16
del_e
x' = Ax+Bu
y = Cx+Du
Estimator
K* u
Gain



Modern

```
>> [a,b,c,d]=linmod('F16_classical');

>> [a,b,c,d]=minreal(a,b,c,d);
3 states removed.
>> [num,den]=ss2tf(a,b,c,d);
>> g=tf(num,den );
>> w1=logspace(-2,2,200);
>> zpk(g)

Zero/pole/gain:
-1.079 (s-13.09) (s+6.419)
--------------------------
   (s+1.911) (s+0.185)

>> bode(g,w1)
```

$\frac{n_z}{\delta_e}$

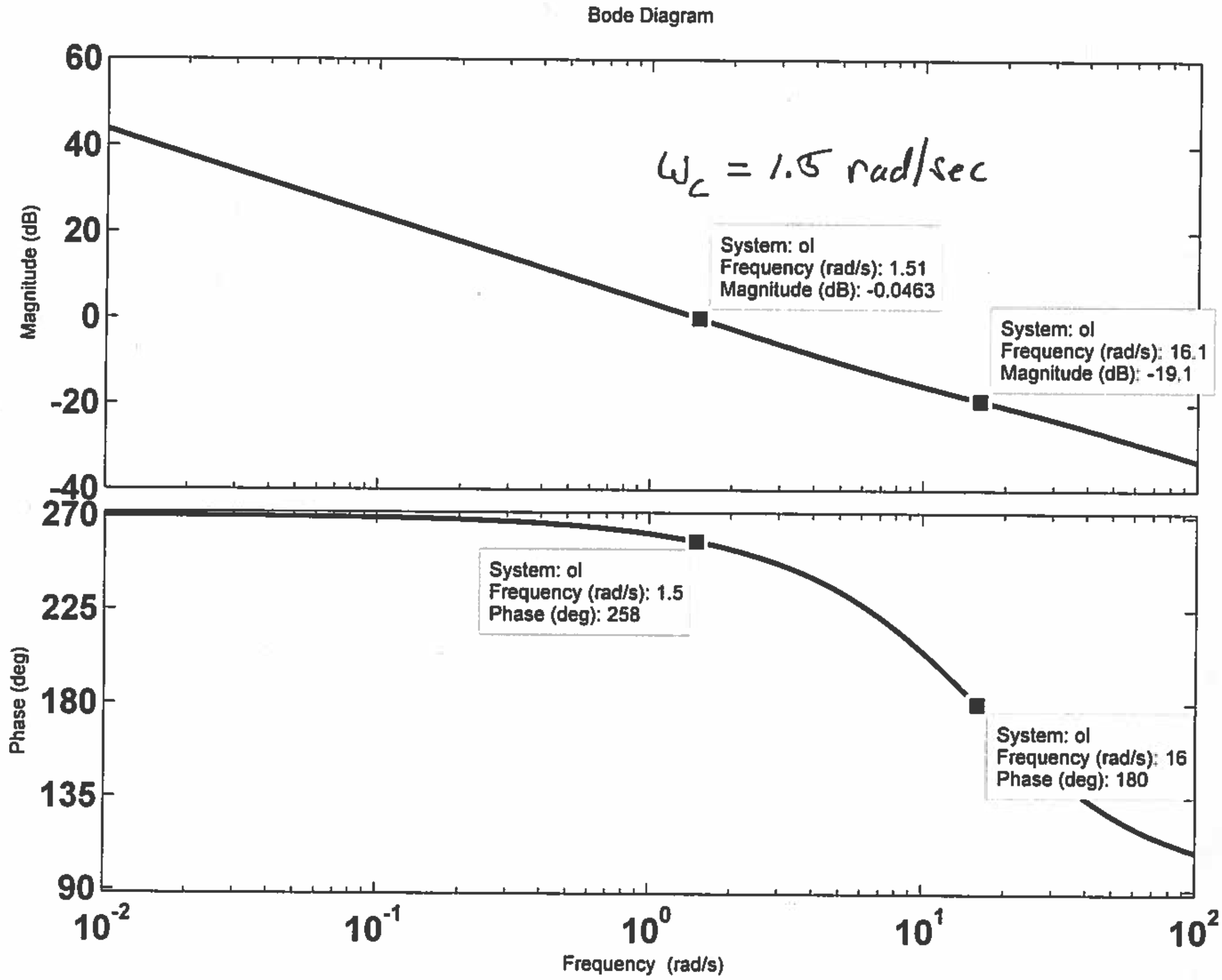


$$GC = \frac{2.06\,(s+2)(s+.185)}{s(s+20)(s+6.4)}$$

$$\frac{nz}{nz_e}$$

G.M = 19.1 dB

P.M. = 78°



$\omega_c = 1.5$ rad/sec

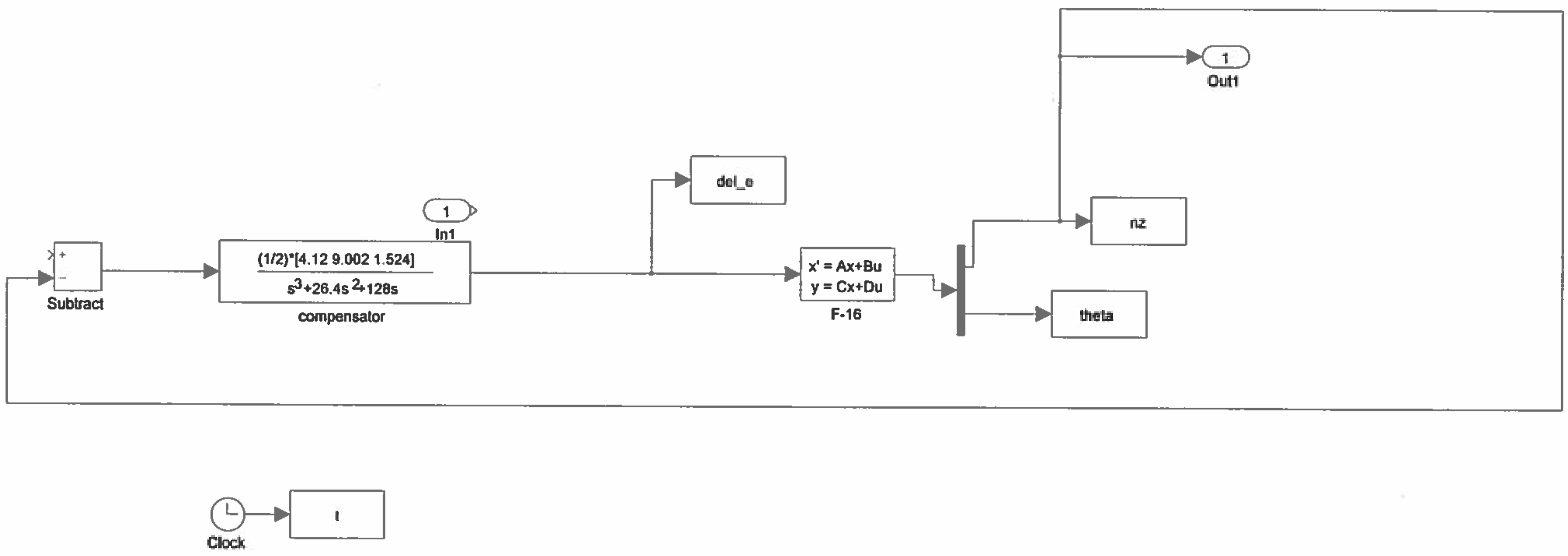
System: ol
Frequency (rad/s): 1.51
Magnitude (dB): -0.0463

System: ol
Frequency (rad/s): 16.1
Magnitude (dB): -19.1

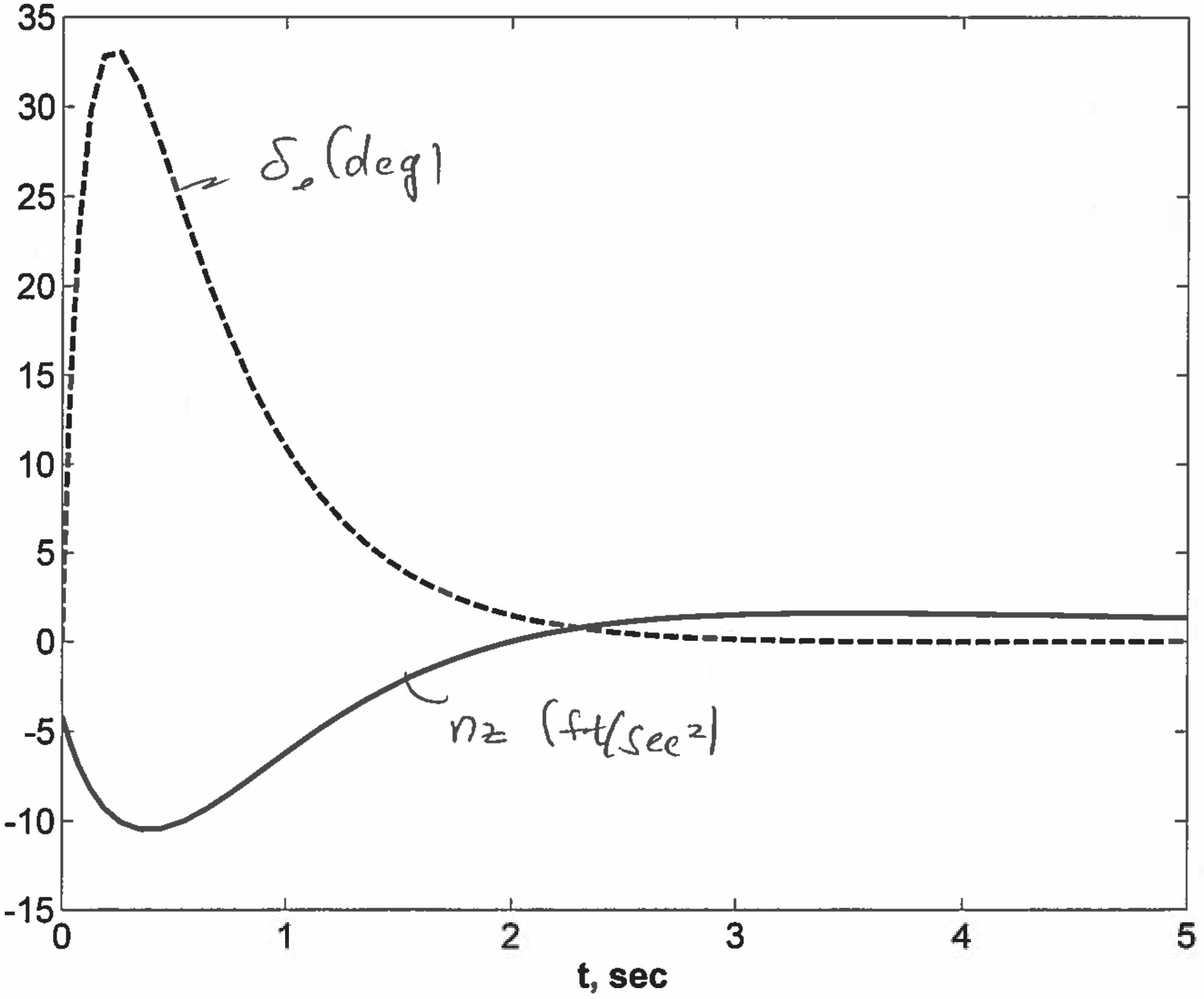
System: ol
Frequency (rad/s): 1.5
Phase (deg): 258

System: ol
Frequency (rad/s): 16
Phase (deg): 180

classical


Subtract
+
-
1
In1
(1/2)*[4.12 9.002 1.524]
s3+26.4s 2+128s
compensator
del_e
x' = Ax+Bu
y = Cx+Du
F-16
nz
theta
1
Out1
Clock
t

classical



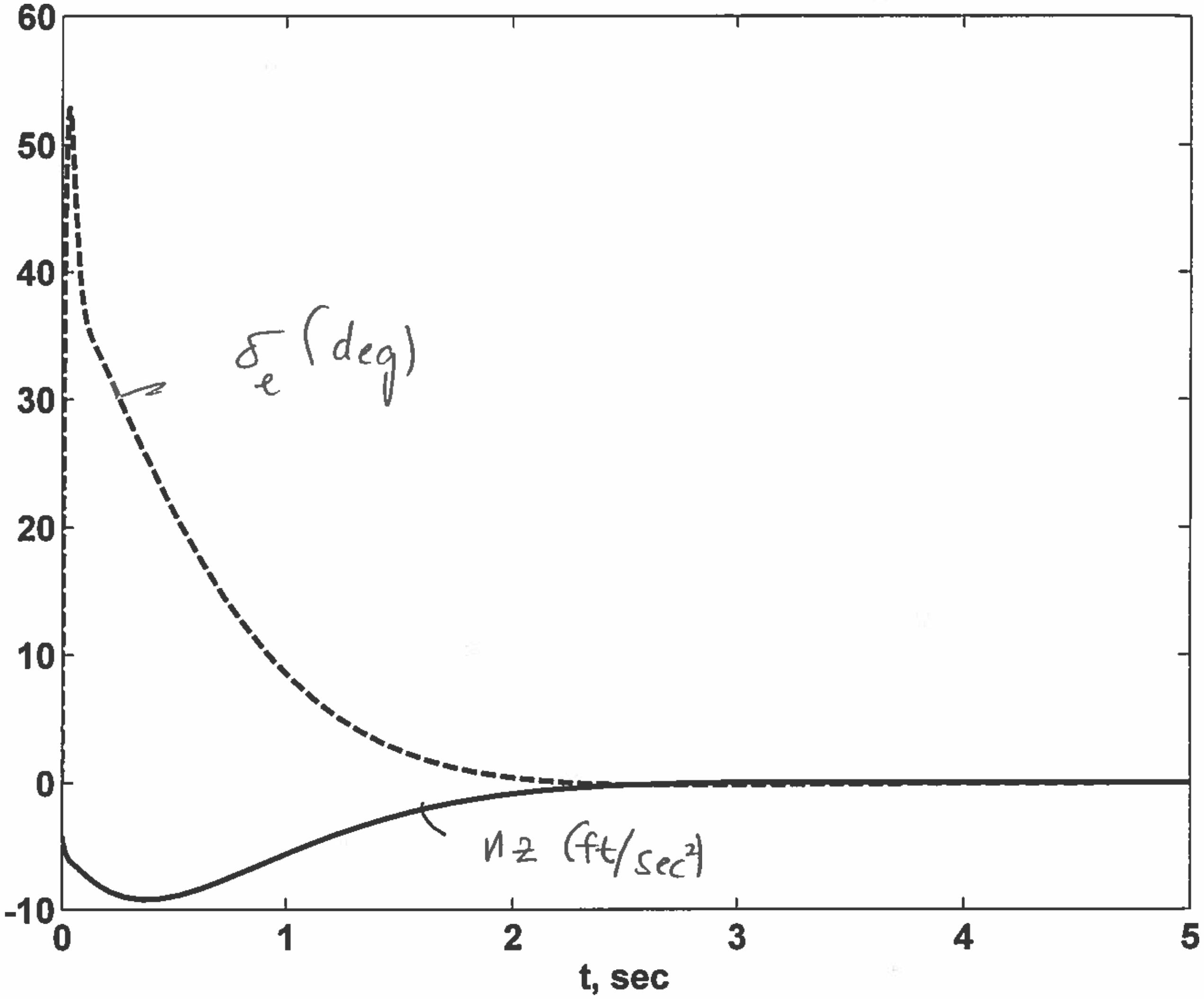
```
 LQRY  Linear-quadratic regulator design with output weighting.

    [K,S,E] = LQRY(SYS,Q,R,N) calculates the optimal gain matrix K
    such that:

      * if SYS is a continuous-time system, the state-feedback law
        u = -Kx  minimizes the cost function

              J = Integral {y'Qy + u'Ru + 2*y'Nu} dt
                                                .
        subject to the system dynamics  x = Ax + Bu,  y = Cx + Du

      * if SYS is a discrete-time system, u[n] = -Kx[n] minimizes

              J = Sum {y'Qy + u'Ru + 2*y'Nu}

        subject to  x[n+1] = Ax[n] + Bu[n],   y[n] = Cx[n] + Du[n].

    The matrix N is set to zero when omitted.  Also returned are the
    the solution S of the associated algebraic Riccati equation and
    the closed-loop eigenvalues E = EIG(A-B*K).
```

```
>> k1=place(A16',C16',[-50 -52 -54])

k1 =

  1.0e+003 *

    0.0001   -0.0025   -0.0000
   -0.1079    1.4280    0.0801

>> A16obs1=A16-k1'*C16;
>> eig(A16obs1)

ans =

  -54.0000
  -50.0000
  -52.0000

>> B16obs1=[B16 k1']

B16obs1 =

  1.0e+003 *

   -0.0000    0.0001   -0.1079
   -0.0002   -0.0025    1.4280
         0   -0.0000    0.0801

>>
```

New estimator poles
needed for stability

```
>> sys=ss(A16,B16,C16,D16);
>> Q161='          [.01 0; 0.01]    outputs nz, θ
>> N=[0;0];
>> [K1,S1,E1]=lqry(sys,Q161,R16,N)

K1 =

 -170.9752  -47.3782   -0.4028


S1 =

  1.0e+004 *

    4.6868    0.8573   -0.0401
    0.8573    0.1851    0.0019
   -0.0401    0.0019    0.0501


E1 =

  -5.3896 + 2.7339i
  -5.3896 - 2.7339i
  -0.0020

>>
```

Modern output regulator

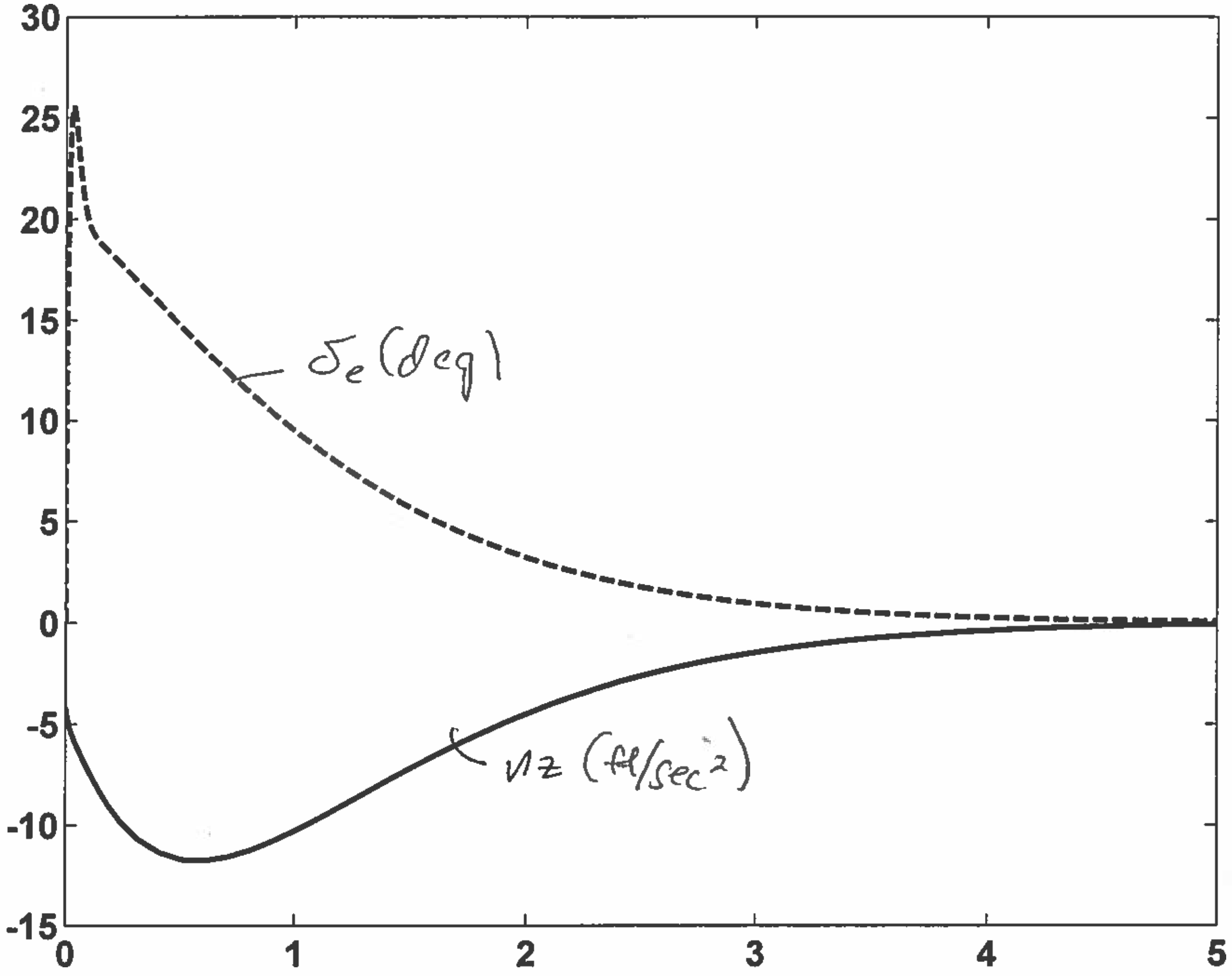


```
>> R16=20

R16 =

    20
```

Q161 = [.01 0; 0 .01]

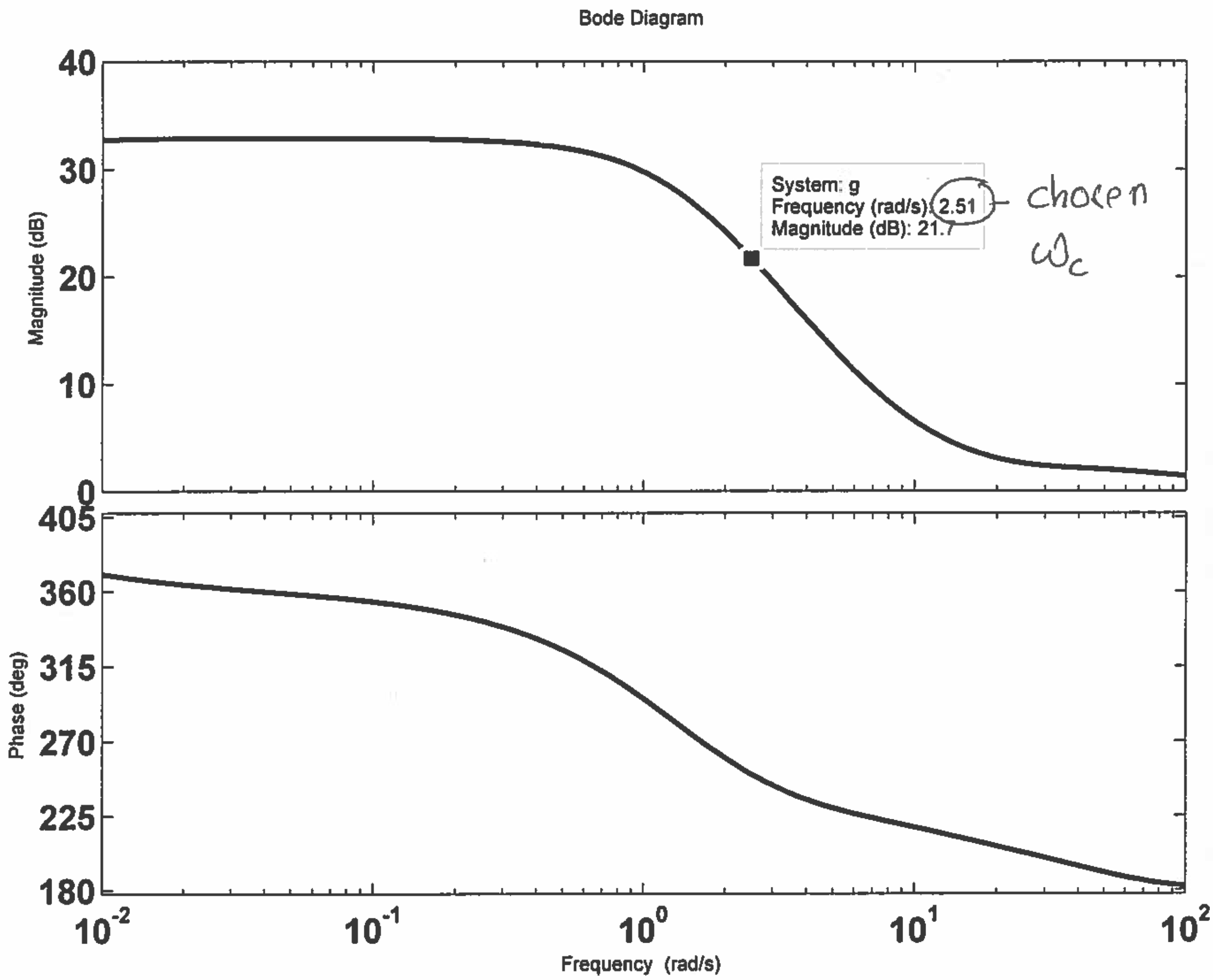
```
>> [K1,S1,E1]=lqry(sys,Q161,R16,N)

K1 =

   -6.0639   -4.0306   -0.0224


S1 =

  1.0e+003 *

    1.4912    0.7044   -0.0006
    0.7044    0.4538    0.0026
   -0.0006    0.0026    0.0051


E1 =

  -1.4075 + 0.2767i
  -1.4075 - 0.2767i
  -0.0020
```

modern, modified output regulator

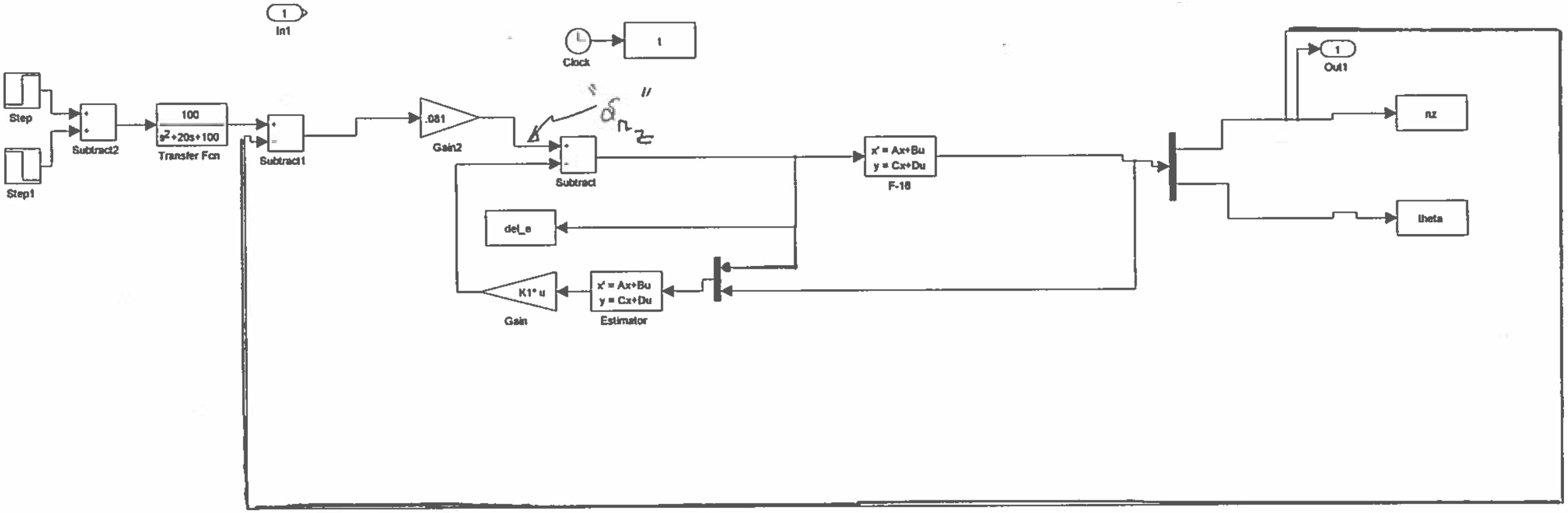


modern, modified output regulation

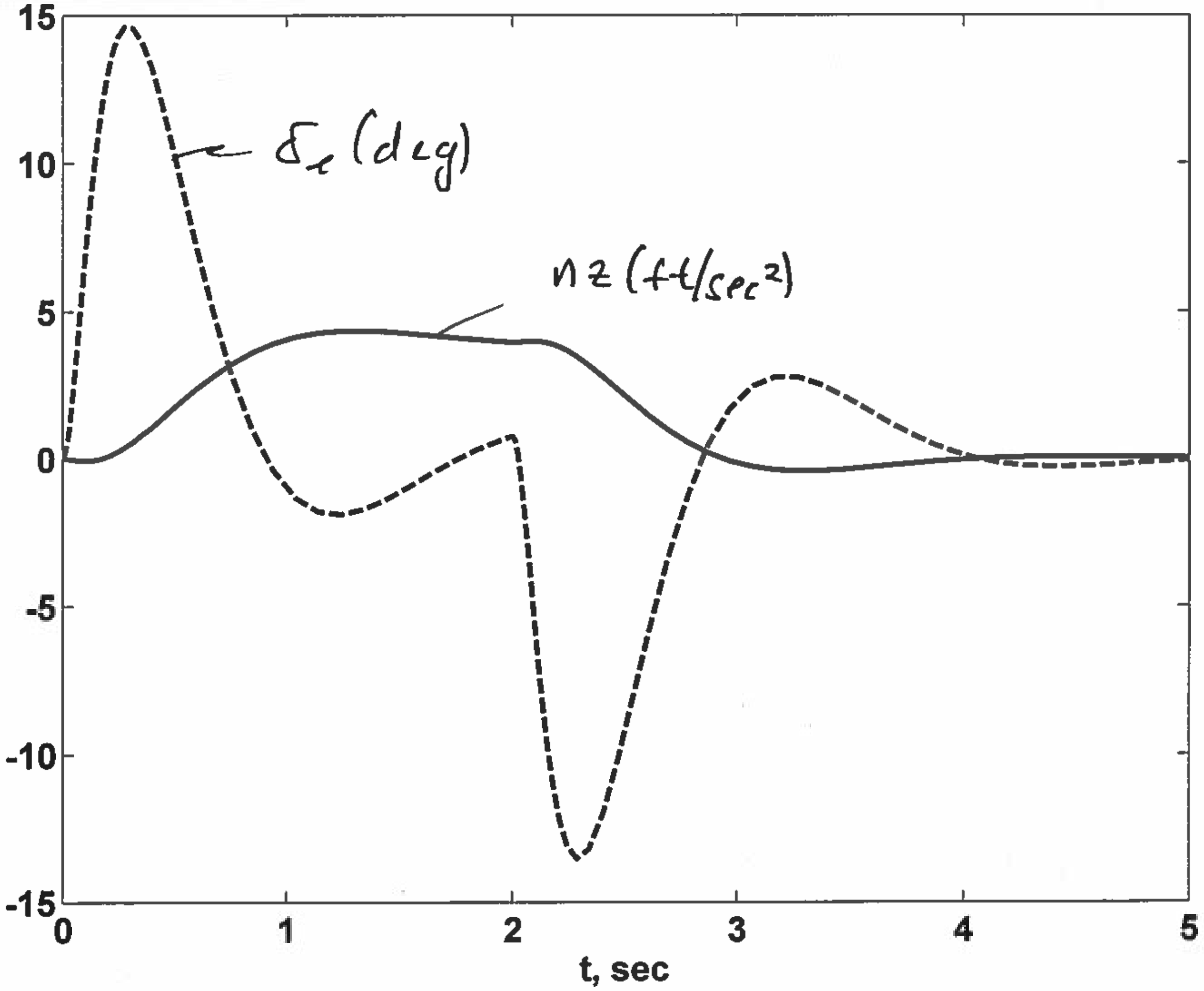
$\frac{nz}{\delta_{n_z}}$



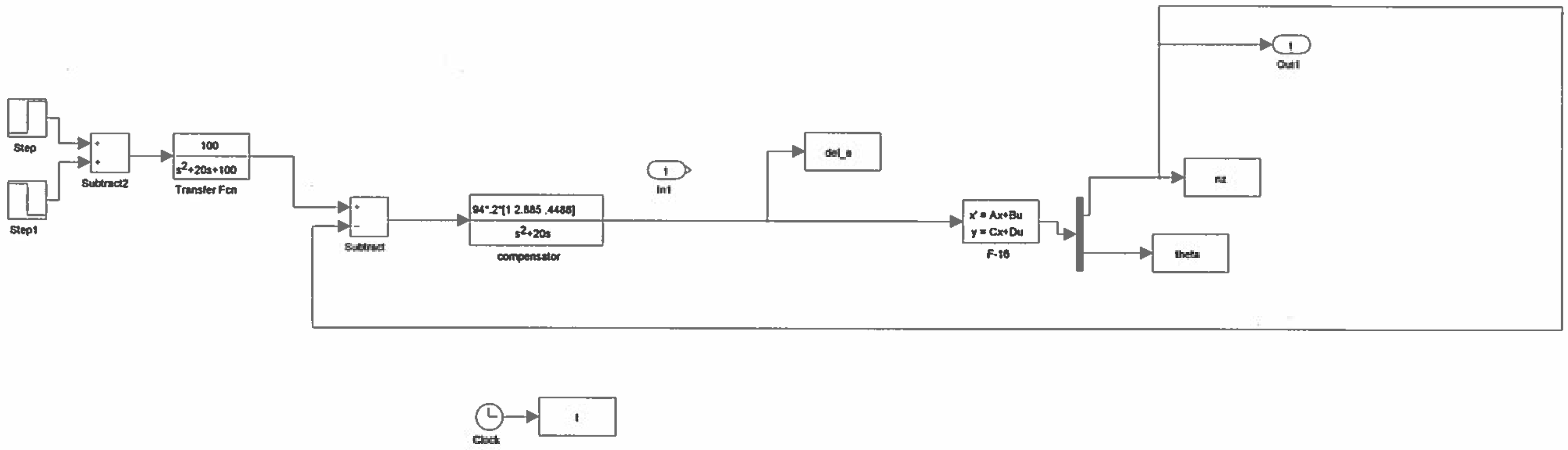
Modern, output regulation,
command following



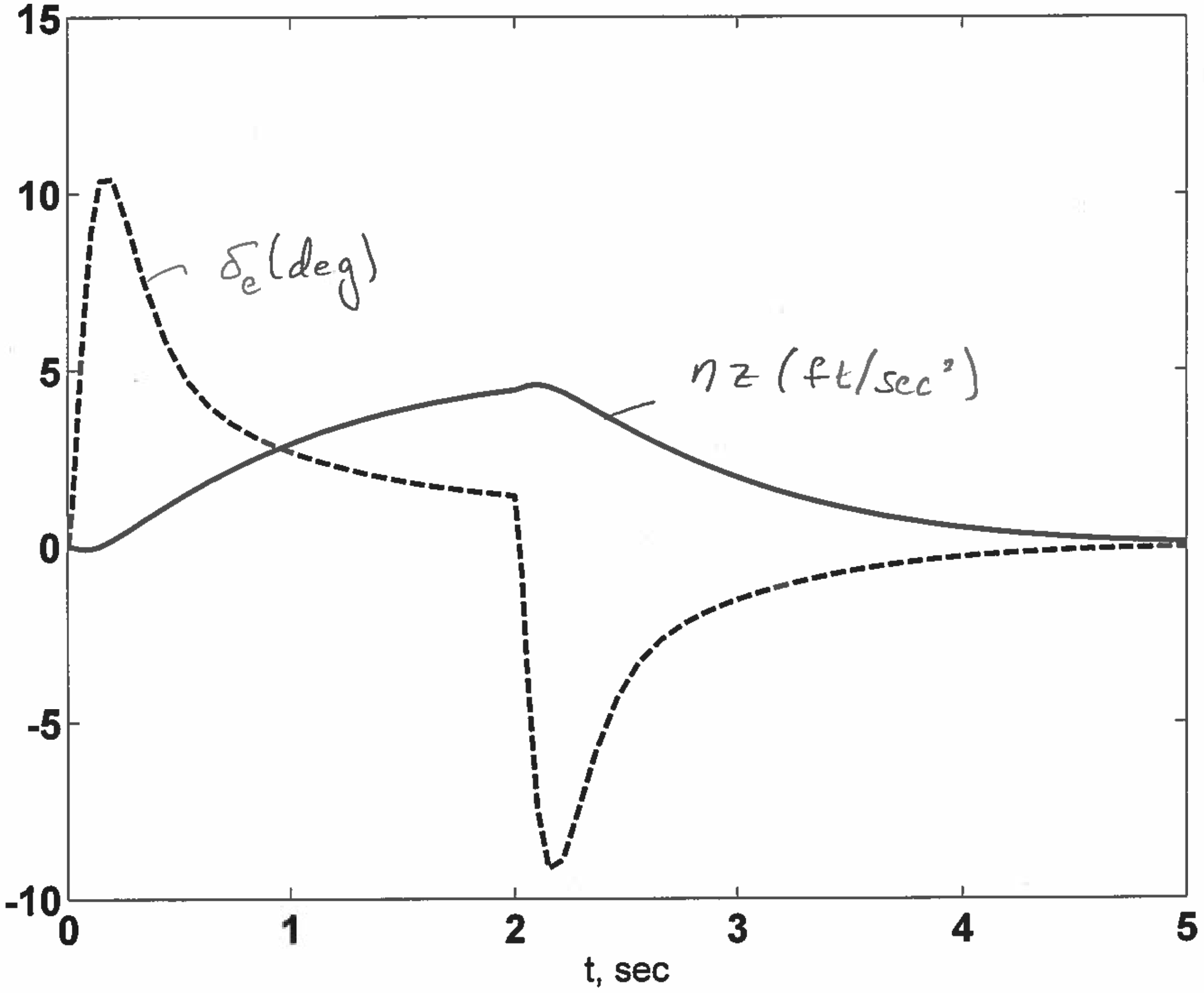
Modern, modified output regulation



classical, with command input



classical, with command input



Dynamic Inversion

Consider the vehicle described by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du. \end{aligned} \tag{1}$$

If D is invertible then full state feedback gives the decoupling control law

$$u = D_R^{-1}(v - Cx) \tag{2}$$

$D_R^{-1} = D^T(DD^T)^{-1}$

where D_R^{-1} denotes a right inverse of D and v is a pseudocontrol. Thus, (1) can be written

$$\begin{aligned}\dot{x} &= \left(A - BD_R^{-1}C\right)x + BD_R^{-1}v \\ \boxed{y} &\boxed{= v.} \end{aligned} \tag{3}$$

If the ith row of D is zero, the ith output is differentiated until the input appears explicitly

$$\begin{aligned}\dot{y}_i &= C_i\dot{x} = C_iAx + C_iBu \\ &\cdots \\ y_i^{(d_i)} &= C_iA^{d_i}x + C_iA^{(d_i-1)}Bu. \end{aligned} \tag{4}$$

d_i = relative order for output y_i

The superscript (d_i) denotes the smallest number of times that it is necessary to differentiate y_i before an element of u appears. Defining

$$y^{(d)} = Fx + Eu \tag{5}$$

where

$$E = \left[E_1^T \cdots E_i^T \cdots E_n^T\right]^T \tag{6}$$

with

$$E_i = \left\{ \begin{array}{ll} D_i, & \text{for } d_i = 0 \\ C_iA^{(d_i-1)}B, & \text{for } d_i \geq 1 \end{array} \right\} \tag{7}$$

and

$$F = \left[F_1^T \cdots F_i^T \cdots F_n^T\right]^T \tag{8}$$

where

$$F_i = \left\{ \begin{array}{ll} C_i, & \text{for } d_i = 0 \\ C_iA^{d_i}, & \text{for } d_i \geq 1 \end{array} \right\}. \tag{9}$$

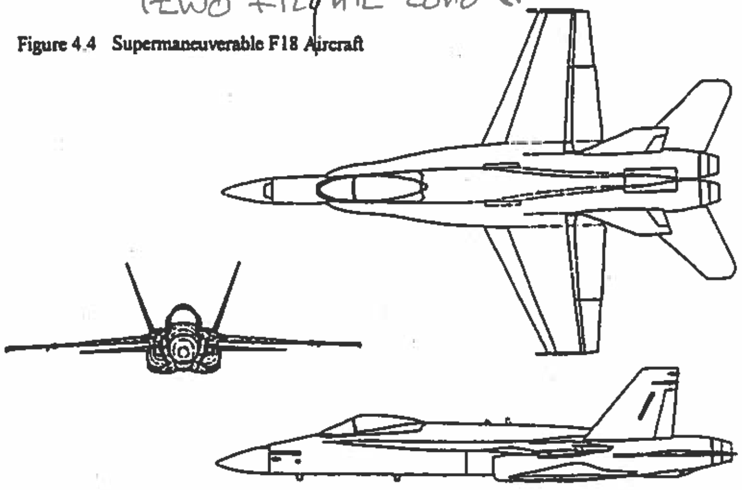
Now, inversion gives

$$\begin{aligned}\dot{x} &= \left(A - BE_R^{-1}F\right)x + BE_R^{-1}v \\ \longrightarrow u &= E_R^{-1}\underbrace{(v - Fx)}_{\text{full state feedback!}} \longleftarrow \\ \boxed{y^{(d)}} &\boxed{= v.} \end{aligned} \tag{10}$$

The advantage of dynamic inversion is that is essentially automates the process of "gain scheduling." That is, scheduling the control system characteristics with flight condition is handled by implementing Eqs. 10. One disadvantage is the sensitivity of system stability and performance to errors in the model (**A**, **B**, etc). in Eqs. 10.

Example
(two flight cond's)

Figure 4.4 Supermaneuverable F18 Aircraft



β Side-Slip Angle = $\sin^{-1}(\frac{V}{V_T})$, (degrees)

q Perturbational Pitch Rate (degrees/sec)

r Perturbational Yaw Rate (degrees/sec); Command Input Signal

δ_A Deflection of Aileron Effector (degrees)

δ_{DT} Deflection of Differential Tail Effector (degrees)

δ_R Deflection of Rudder Effector (degrees)

δ_{RTV} Deflection of Pitch Thrust Vectoring Effector (degrees)

δ_{YTV} Deflection of Yaw Thrust Vectoring Effector (degrees)

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Y_\beta & \sin(\alpha) & -\cos(\alpha) \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta DT} & Y_{\delta A} & Y_{\delta R} & Y_{\delta RTV} & Y_{\delta YTV} \\ L_{\delta DT} & L_{\delta A} & L_{\delta R} & L_{\delta RTV} & L_{\delta YTV} \\ N_{\delta DT} & N_{\delta A} & N_{\delta R} & N_{\delta RTV} & N_{\delta YTV} \end{bmatrix} \begin{bmatrix} \delta_{DT} \\ \delta_A \\ \delta_R \\ \delta_{RTV} \\ \delta_{YTV} \end{bmatrix}$$

$$= A_{lat/dir} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + B_{lat/dir} \begin{bmatrix} \delta_{DT} \\ \delta_A \\ \delta_R \\ \delta_{RTV} \\ \delta_{YTV} \end{bmatrix}$$

$$A_{lat/dir}^{m3h10} = \begin{bmatrix} -0.1292 & 0.1738 & -0.9833 \\ -8.643 & -1.129 & 0.5986 \\ 1.519 & -0.01327 & -0.1105 \end{bmatrix}$$

$$B_{lat/dir}^{m3h10} = \begin{bmatrix} -0.006987 & -0.005249 & 0.01285 & 0 & 0.006894 \\ 5.096 & 6.075 & 0.51 & 0.1781 & 0.02478 \\ 0.1908 & -0.1522 & -0.3872 & 0.0008849 & -0.3397 \end{bmatrix}$$

$$A_{lat/dir}^{m4h10} = \begin{bmatrix} -0.1544 & 0.09691 & -0.9939 \\ -9.965 & -1.721 & 0.599 \\ 2.169 & -0.01995 & -0.1447 \end{bmatrix}$$

$$B_{lat/dir}^{m4h10} = \begin{bmatrix} -0.01187 & -0.006276 & 0.01785 & 0 & 0.002272 \\ 9.643 & 12.16 & 0.9326 & 0.1495 & 0.01088 \\ 0.2768 & -0.2727 & -0.7155 & -0.000305 & -0.1492 \end{bmatrix}$$

M=5
h=10,000ft

$$A_{lat/dir}^{m5h10} = \begin{bmatrix} -0.1932 & 0.06234 & -0.9968 \\ -12.37 & -2.164 & 0.6034 \\ 3.119 & -0.0211 & -0.1802 \end{bmatrix}$$

$$B_{lat/dir}^{m5h10} = \begin{bmatrix} -0.01652 & -0.007459 & 0.02203 & 0 & 0.002821 \\ 15.18 & 18.04 & 1.412 & 0.3035 & 0.01689 \\ 0.3458 & -0.3975 & -1.099 & -0.002071 & -0.2315 \end{bmatrix}$$

$$A_{lat/dir}^{m7h10} = \begin{bmatrix} -0.2701 & 0.03162 & -0.9984 \\ -17.77 & -3.177 & 0.5446 \\ 5.987 & -0.0205 & -0.2555 \end{bmatrix}$$

$$B_{lat/dir}^{m7h10} = \begin{bmatrix} -0.02472 & -0.00764 & 0.02855 & 0 & 0.01074 \\ 29.4 & 25.95 & 2.503 & 1.789 & 0.09002 \\ 0.4006 & -0.3672 & -1.982 & -0.02069 & -1.234 \end{bmatrix}$$

$$A_{lat/dir}^{m9h10} = \begin{bmatrix} -0.321 & 0.02008 & -0.9987 \\ -17.6 & -5.716 & 0.5193 \\ 9.433 & -0.02149 & -0.3391 \end{bmatrix}$$

$$B_{lat/dir}^{m9h10} = \begin{bmatrix} -0.0301 & 0 & 0.03051 & 0 & 0.02583 \\ 46.68 & 19.41 & 4.054 & 5.024 & 0.2782 \\ 0.2385 & -0.3226 & -2.951 & -0.06746 & -3.813 \end{bmatrix}$$

$$A_{lat/dir}^{m4h20} = \begin{bmatrix} -.0112 & 0.1408 & -0.9889 \\ -8.538 & -1.171 & 0.5146 \\ 1.619 & -0.01304 & -0.103 \end{bmatrix}$$

$$B_{lat/dir}^{m4h20} = \begin{bmatrix} -0.007388 & -0.004613 & 0.01238 & 0 & 0.008751 \\ 6.245 & 7.673 & 0.611 & 0.3615 & 0.04034 \\ 0.1998 & -0.1822 & -0.4722 & 0.0008652 & -0.553 \end{bmatrix}$$

$$A_{lat/dir}^{m5h20} = \begin{bmatrix} -0.1354 & 0.09036 & -0.9949 \\ -10.37 & -1.469 & 0.5126 \\ 2.281 & -0.01482 & -0.1277 \end{bmatrix}$$

$$B_{lat/dir}^{m5h20} = \begin{bmatrix} -0.01091 & -0.005695 & 0.01555 & 0 & 0.00489 \\ 9.93 & 12.12 & 0.9416 & 0.3977 & 0.02817 \\ 0.2757 & -0.2797 & -0.7419 & -0.001175 & -0.3861 \end{bmatrix}$$

$$A_{lat/dir}^{m6h20} = \begin{bmatrix} -0.166 & 0.0629 & -0.9971 \\ -12.97 & -1.761 & 0.5083 \\ 3.191 & -0.01417 & -0.1529 \end{bmatrix}$$

$$B_{lat/dir}^{m6h20} = \begin{bmatrix} -0.0142 & -0.00686 & 0.01851 & 0 & 0.005817 \\ 14.38 & 16.76 & 1.316 & 0.7007 & 0.0402 \\ 0.3389 & -0.385 & -1.051 & -0.00475 & -0.5511 \end{bmatrix}$$

M=7.5

h = 20,000 ft

$$A_{lat/dir}^{m75h20} = \begin{bmatrix} -0.1982 & 0.03905 & -0.9984 \\ -17.3 & -2.505 & 0.4624 \\ 4.688 & -0.01064 & -0.1942 \end{bmatrix}$$

$$B_{lat/dir}^{m75h20} = \begin{bmatrix} -0.01816 & -0.007716 & 0.02168 & 0 & 0.01018 \\ 22.53 & 23.71 & 1.905 & 1.697 & 0.08794 \\ 0.4211 & -0.238 & -1.555 & -0.01765 & -1.206 \end{bmatrix}$$

$$A_{lat/dir}^{m9h20} = \begin{bmatrix} -0.2257 & 0.02638 & -0.9989 \\ -19.08 & -3.708 & 0.4264 \\ 6.586 & -0.01925 & -0.2379 \end{bmatrix}$$

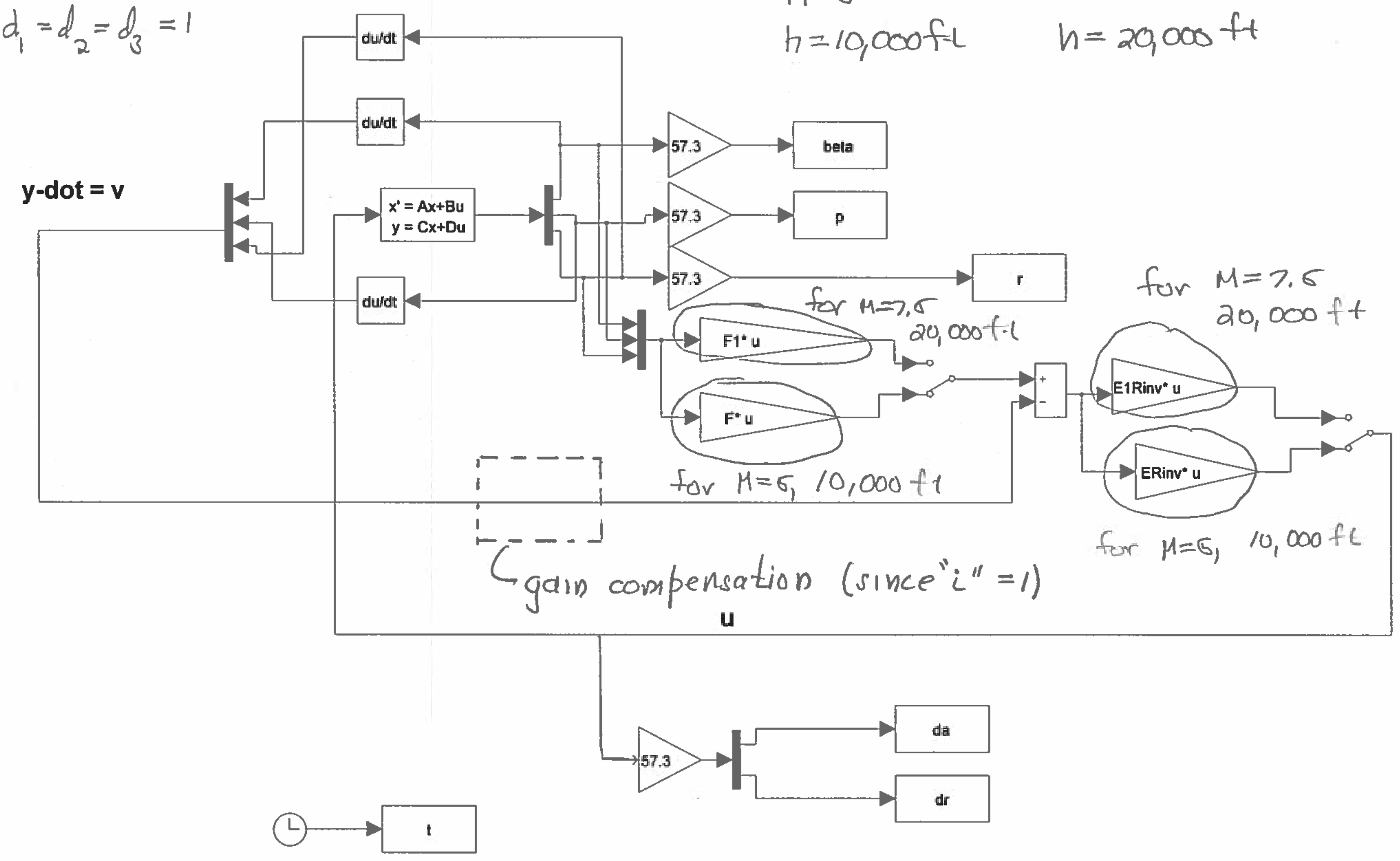
$$B_{lat/dir}^{m9h20} = \begin{bmatrix} -0.02089 & -0.005593 & 0.02314 & 0 & 0.01809 \\ 31.29 & 25.88 & 2.879 & 3.359 & 0.1875 \\ 0.3459 & -0.1457 & -2.152 & -0.04183 & -2.571 \end{bmatrix}$$

relative order = 1 for all outputs

$d_1 = d_2 = d_3 = 1$

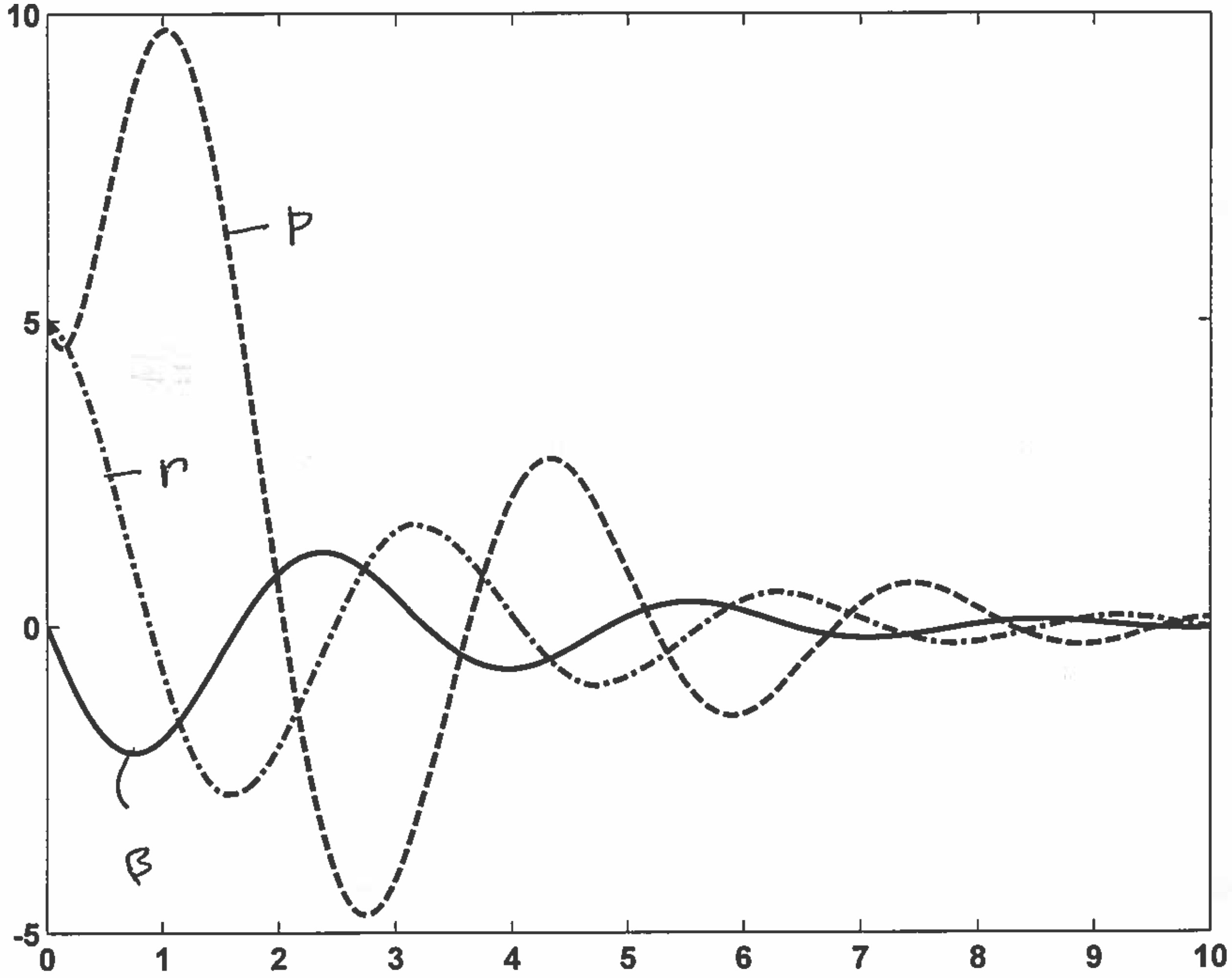
M=5 h=10,000ft

M=7.5 h=20,000 ft



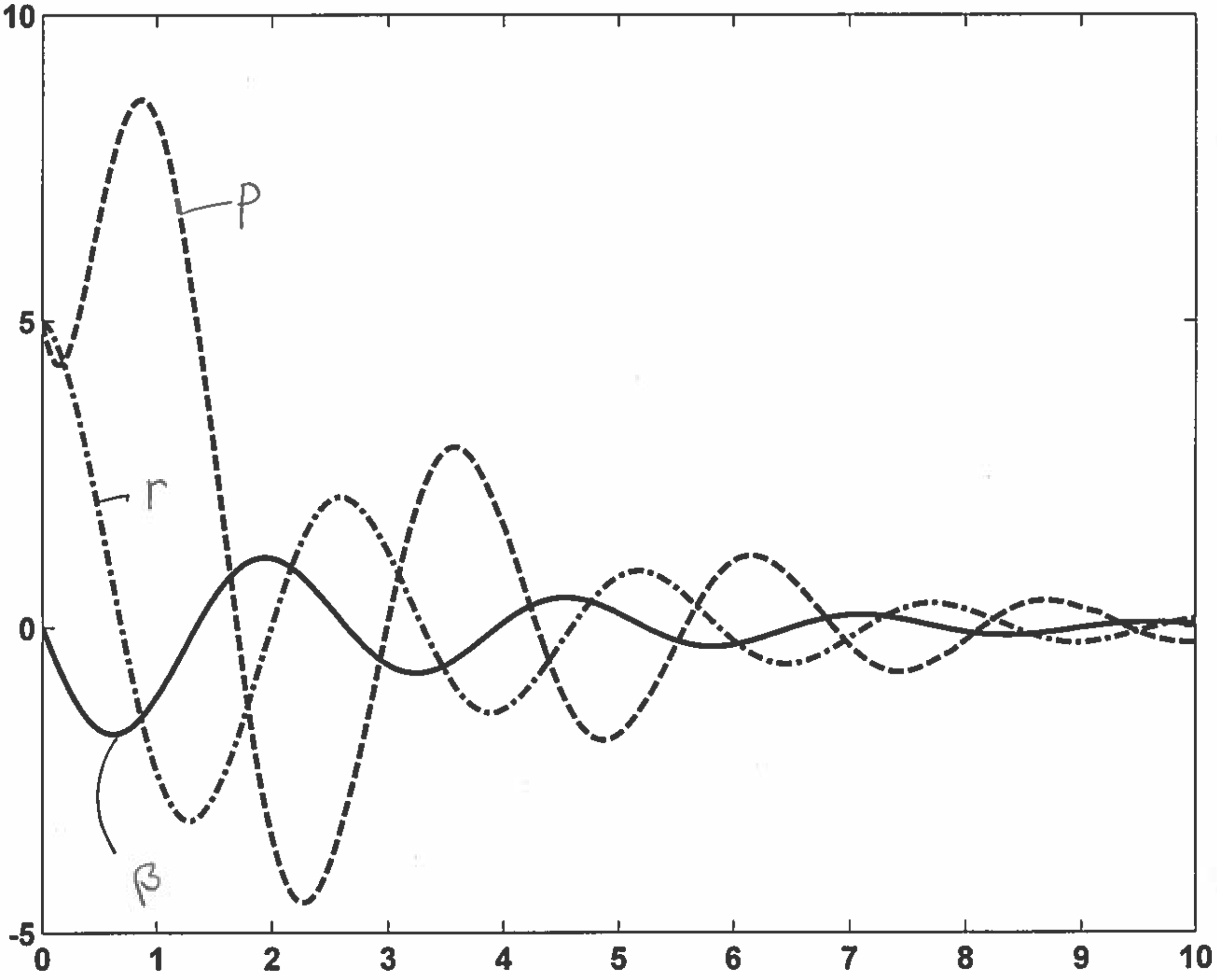
$p_o = r_o = 5$ deg/sec

$M = 5$, $h = 10{,}000$ ft



$P_0 = r_0 = 5$ deg/sec

$M = 7.5 \quad h = 20{,}000$ ft

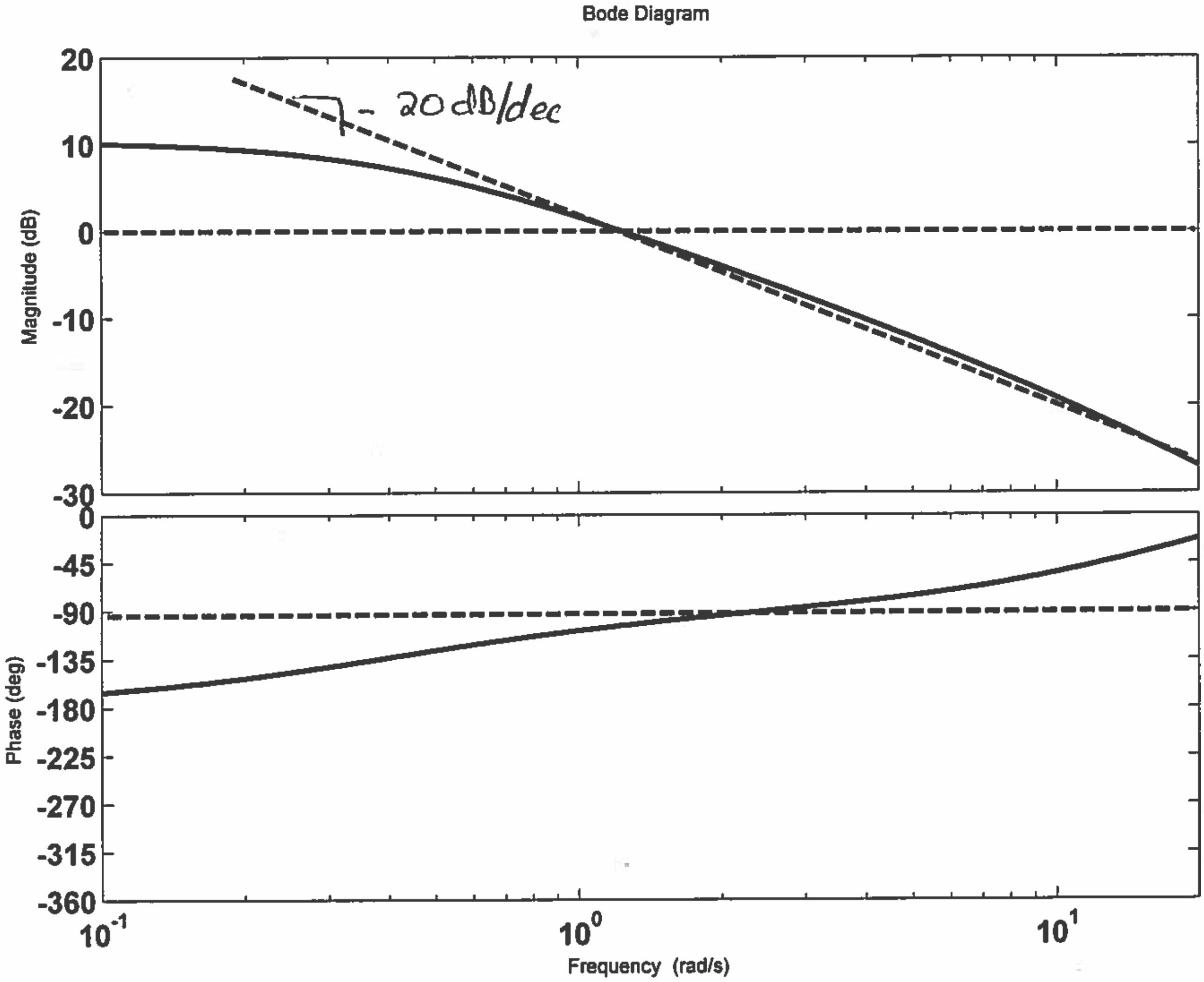


not perfect $1/s$ since E_R^{-1} involved near singularity

$\frac{\beta}{\vec{v}_i}$

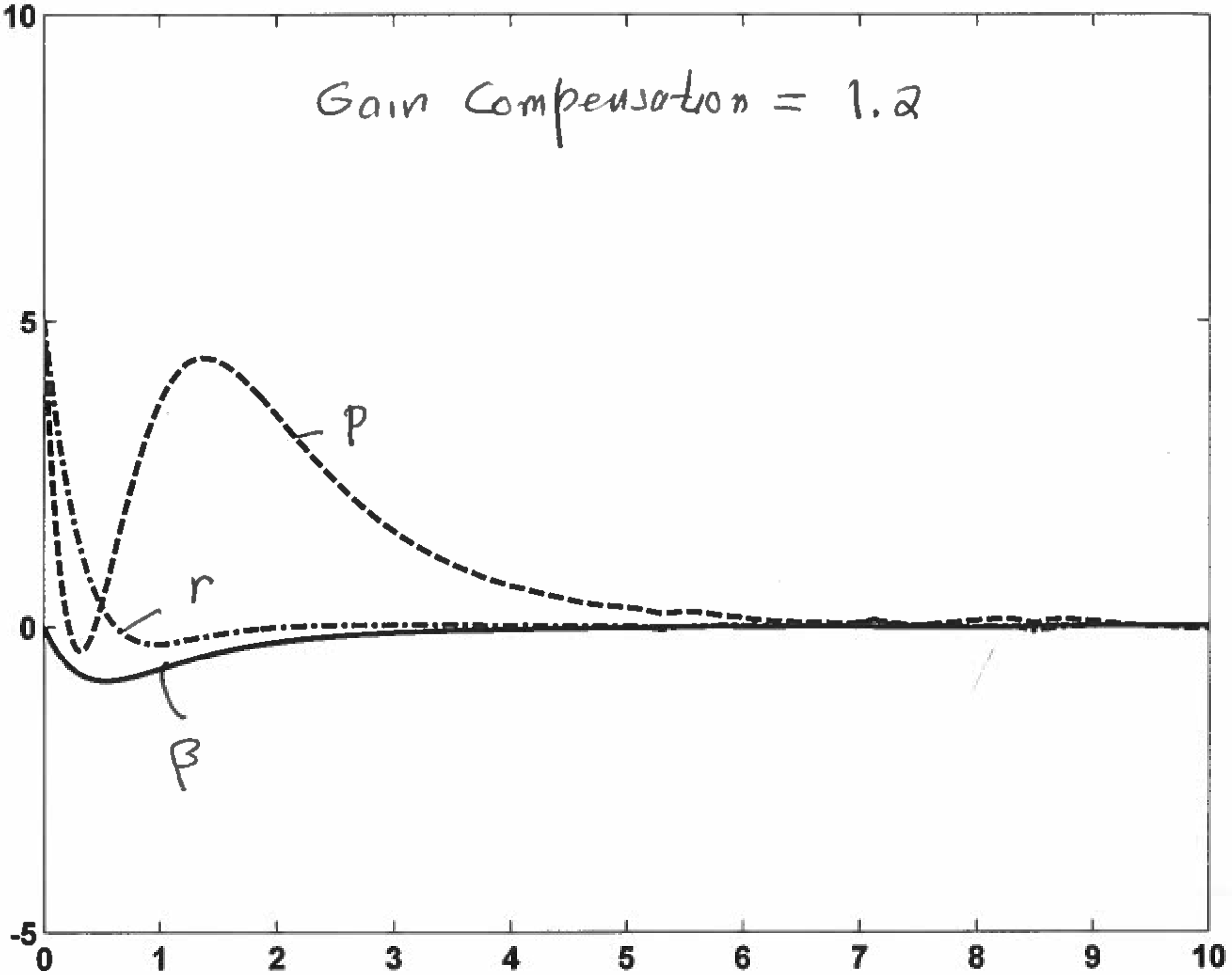
$M = 7.6$

$h = 20{,}000$ ft



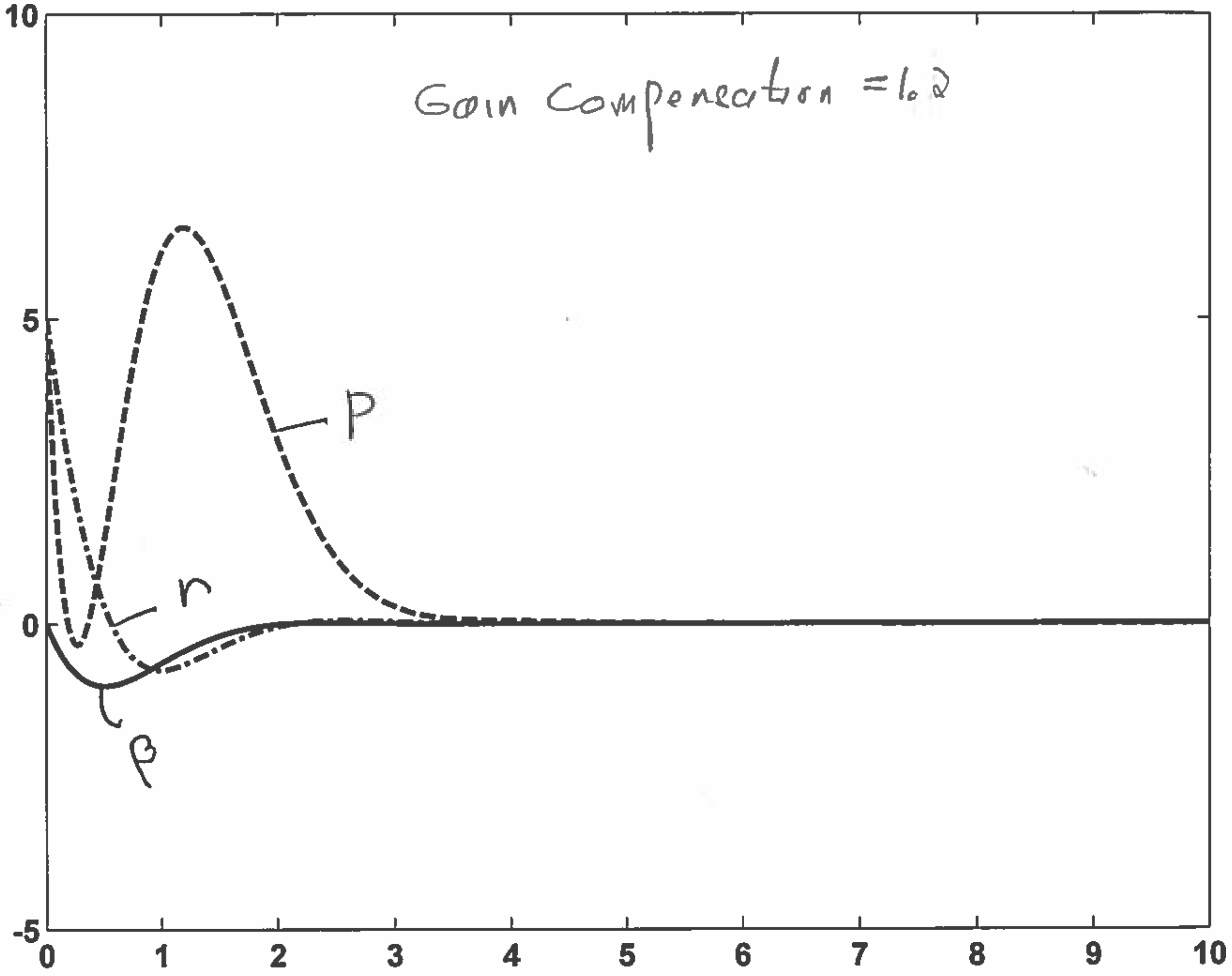
$P_0 = r_0 = 5$ deg/sec

$M = 6$, $h = 10,000$ ft



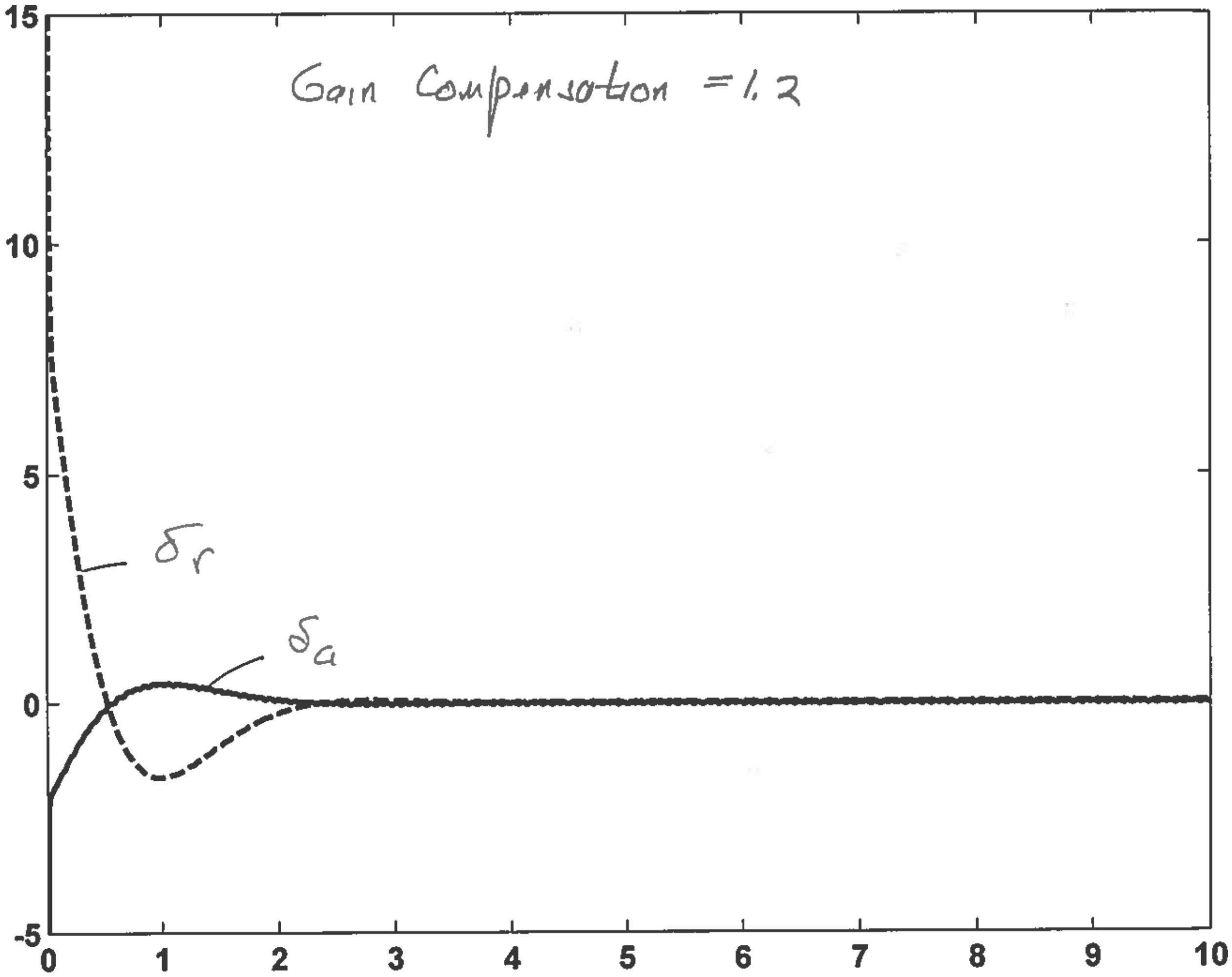
$p_0 = r_0 =$ 5 deg/sec

M = 7.5 h = 20,000 ft



$P_0 = r_0 = 5$ deg/sec

$M = 2.5$ $h = 20,000$ ft


Gain Compensation = 1.2
δ_r
δ_a