Chapter #1

EEE 8007

Digital Control

- **State Space Representation of Discrete Time Control Systems**
- **State Feedback**
- **Discrete Linear Quadratic Regulator (LQR)**
- **Closed Loop Estimators**

State Space Representation of Discrete Time Control Systems

Most controllers are digital, hence we need to transform the continuous systems to discrete.

Many methods for that

1) First Order Hold (exists in Matlab)
2) Tustin's Bilinear Method (exists in Matlab)
3) Bilinear with pre-warping (exists in Matlab)
4) Zero pole matched (exists in Matlab)
5) Impulse (exists in Matlab)
6) Forward where Φ=I+AT, Γ=BT
7) Backward where $\Phi=(I-AT)^{-1}BT$, $\Gamma=C(I-AT)^{-1}$, $C=C(I-AT)^{-1}$, $D=D+(I-AT)^{-1}BT$

Forward:

$$\dot{\mathbf{X}}(kT)=\frac{\mathbf{X}((k+1)T)-\mathbf{X}(kT)}{(k+1)T-kT}=\frac{\mathbf{X}((k+1)T)-\mathbf{X}(kT)}{T}$$

Hence $\dfrac{\mathbf{X}((k+1)T)-\mathbf{X}(kT)}{T}=f\left(\mathbf{X}(kT),\mathbf{U}(kT),kT\right)$

or $\mathbf{X}((k+1)T)=Tf\left(\mathbf{X}(kT),\mathbf{U}(kT),kT\right)+\mathbf{X}(kT)$ or
$\mathbf{X}((k+1)T)=g\left(\mathbf{X}(kT),\mathbf{U}(kT),kT\right)$

Hence, for time-varying (linear or not) discrete time systems the state space representation is:

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{U}(k), k\right)$$
$$\mathbf{Y}(k) = g\left(\mathbf{X}(k), \mathbf{U}(k), k\right)$$

If the system is linear and there is no direct coupling between the input and the output then:

$$\mathbf{X}(k+1) = \mathbf{\Phi}(k)\mathbf{X}(k) + \mathbf{\Gamma}(k)\mathbf{U}(k)$$
$$\mathbf{Y}(k) = \mathbf{C}(k)\mathbf{X}(k)$$

And for LTI systems:

$$\mathbf{X}(k+1) = \mathbf{\Phi}\mathbf{X}(k) + \mathbf{\Gamma}\mathbf{U}(k)$$
$$\mathbf{Y}(k) = \mathbf{C}\mathbf{X}(k)$$

The relation of the matrices Φ and A, Γ and B can be found in any text book and is:

$$\mathbf{\Phi} = e^{\mathbf{A}T} = \mathbf{I} + \mathbf{A}T + \frac{\mathbf{A}^2T^2}{2!} + \frac{\mathbf{A}^3T^3}{3!} + \ldots$$
$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{A}n} dn\, \mathbf{B}$$

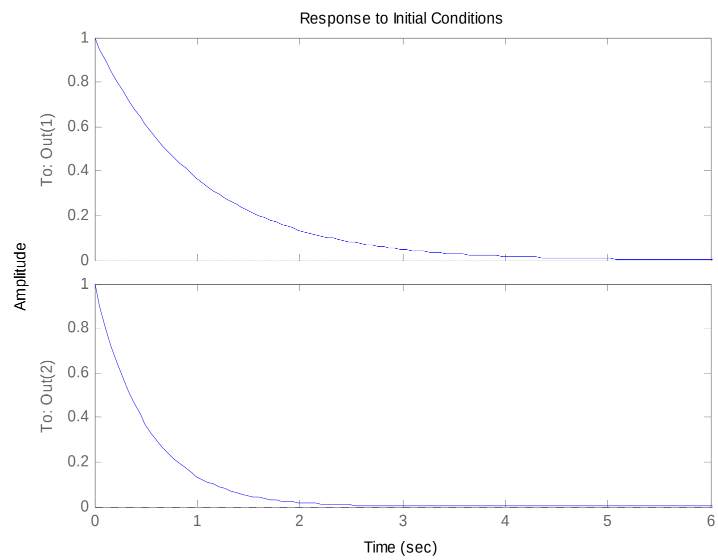
In Matlab use: c2d

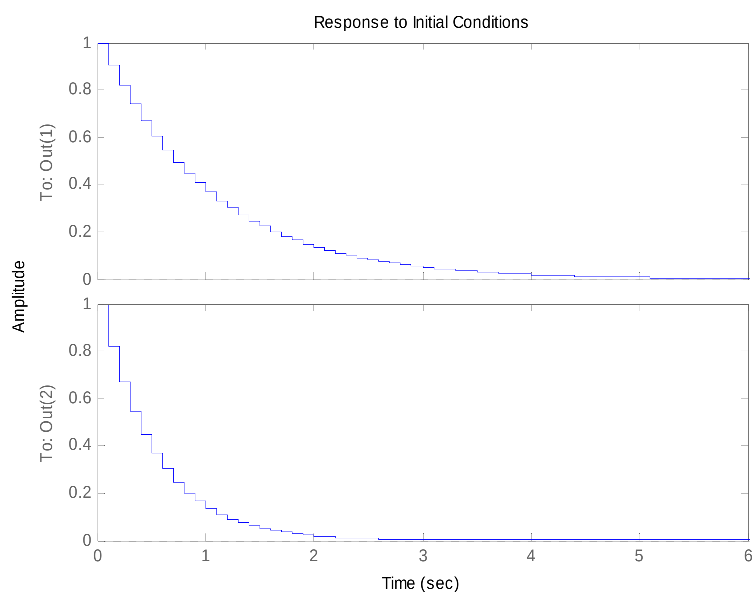
All the other concepts are the same:

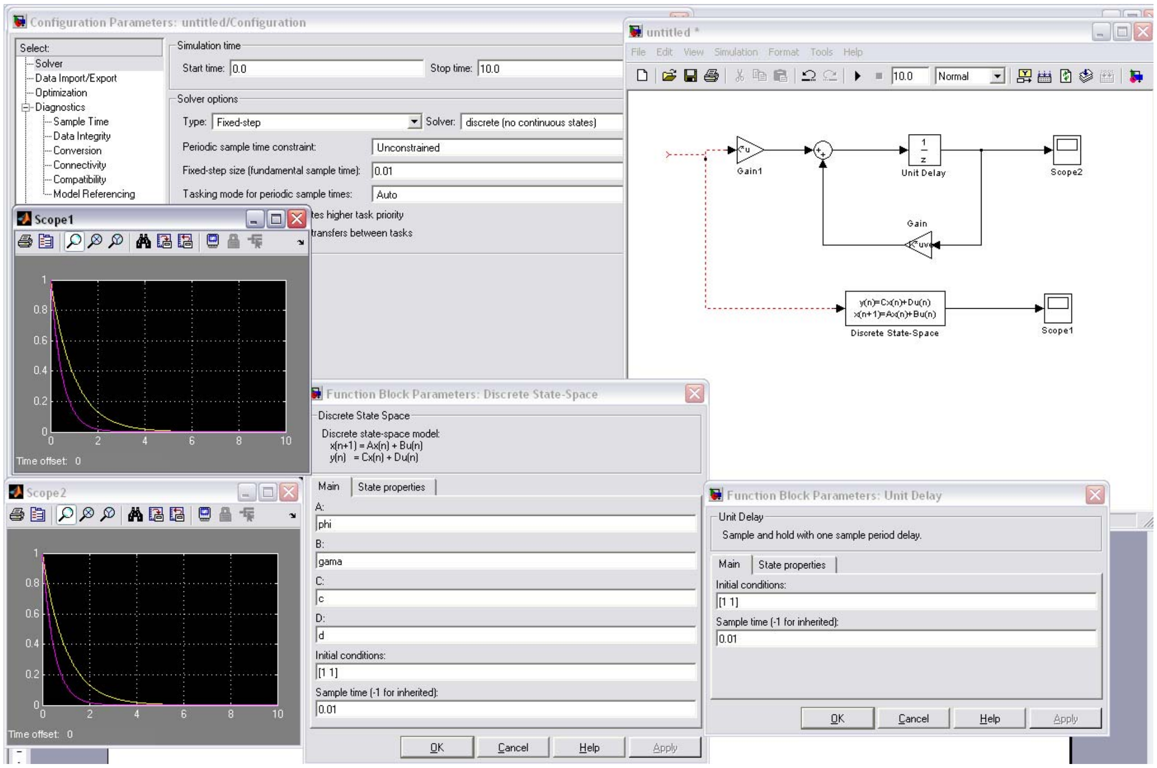
TF of digital system: $\mathbf{C}\left[z\mathbf{I} - \mathbf{\Phi}\right]^{-1}\mathbf{\Gamma}$ and hence poles of system: $eig\left(\mathbf{\Phi}\right)$

```
A=[-1 0;0 -2]; B=[1;0]; C=eye(2); D=0;
sys_c=ss(A,B,C,D);
sys_d=c2d(sys_c, 0.1);
initial(sys_c,[1,1]);
figure
```

```
initial(sys_d,[1,1]);
```







State Feedback

By following the same steps, as in continuous time systems:

$\dot{\mathbf{X}}(k+1) = \mathbf{\Phi}_{cl}\mathbf{X}(k) + \mathbf{\Gamma}_{cl}\mathbf{U}(k)$ where $\Phi_{cl}=\Phi-\Gamma K$, $\Gamma_{cl}=\Gamma F$, $C_{cl}=C-DK$, $D_{cl}=DF$. Thus we have described the closed loop system into the same form as the open loop system.

Since the poles of the OL can be found by the $|zI-\Phi|=0$, the close loop poles are: $|zI-(\Phi-\Gamma K)|=0$.

If we want the close loop poles at $P=[P_1\ P_2\ \dots\ P_n]^T$, (n is the order of the system) then:

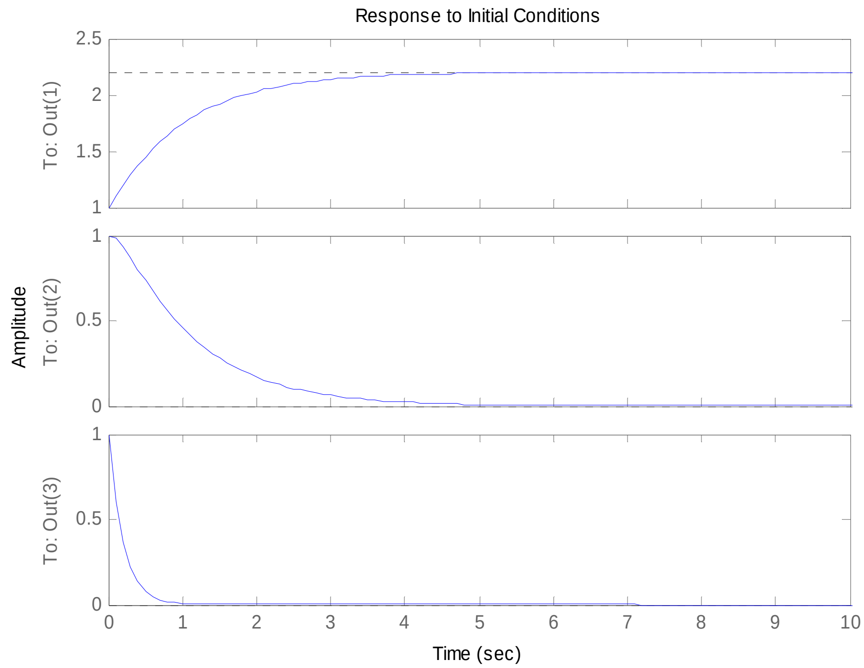
$|zI-\Phi + \Gamma K|=(z-P_1)(z-P_2)\ldots(z-P_3) \Leftrightarrow K=[K_1\ K_2\ \ldots]^T$. Pole placement

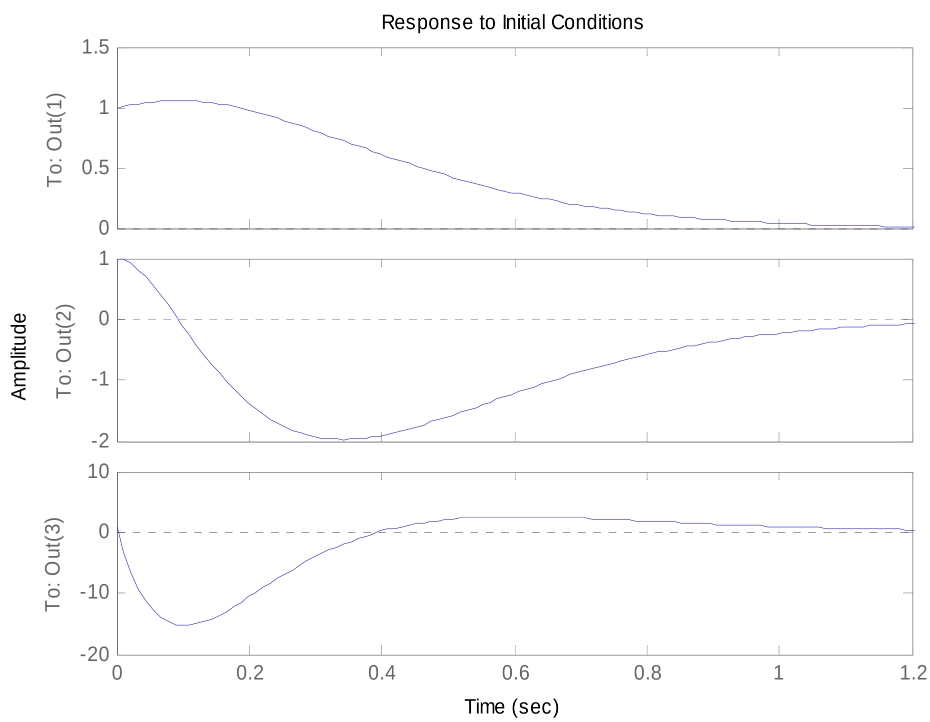
Check **CONTROLLABILITY**

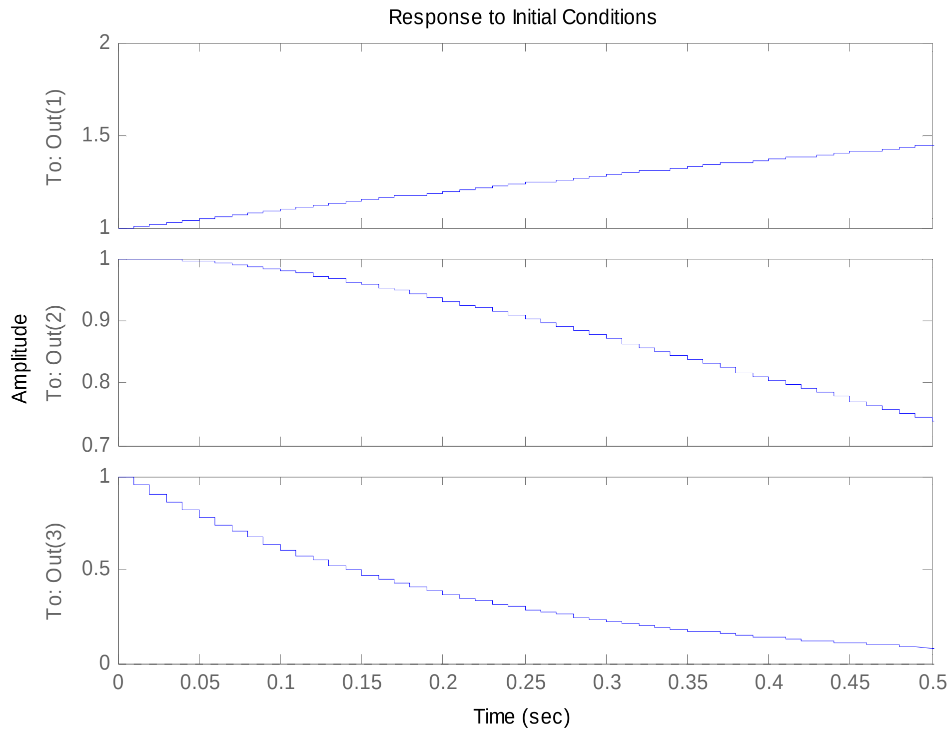
$$\mathbf{S} = \left[\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \mathbf{\Phi^2\Gamma} \quad \ldots \quad \mathbf{\Phi^{n-1}\Gamma}\right]$$

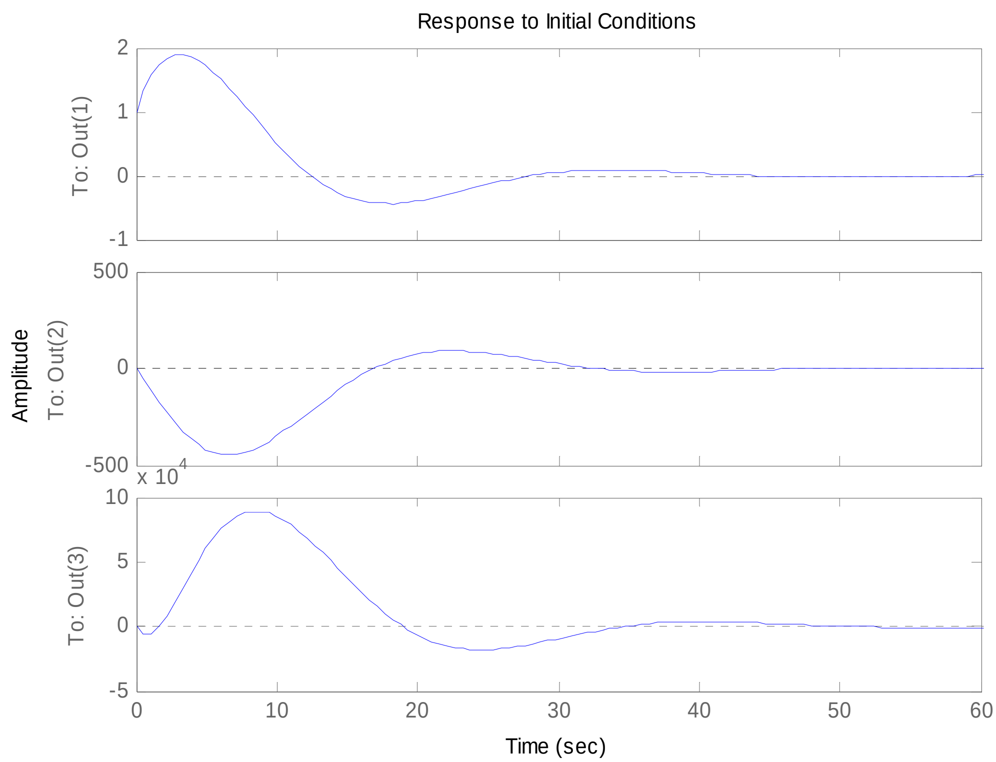
Example:

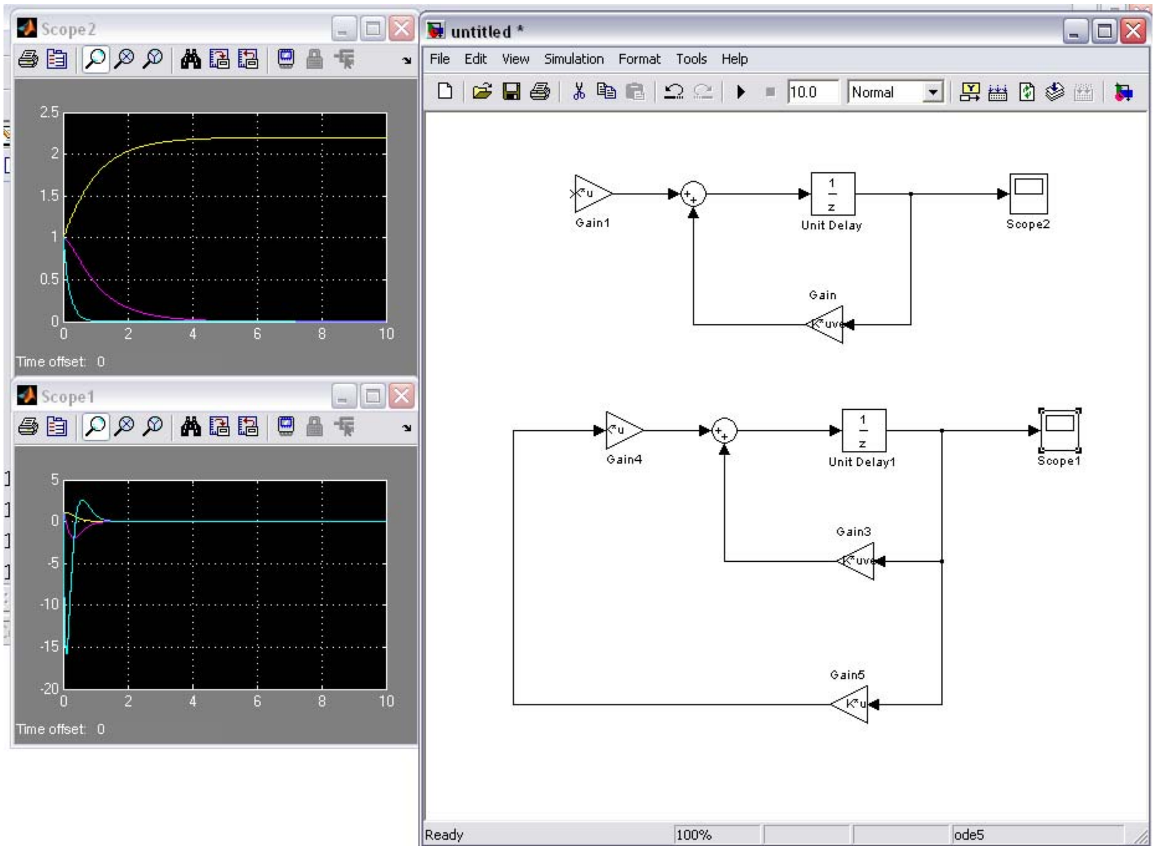
```
clc
close all
clear all
A=[0 1 0;0 -1 1;0 0 -5];B=[0 0 5]';C=eye(3);D=0
sys_c=ss(A,B,C,D)
initial(sys_c,[1 1 1])

K=place(A,B,[-10 -5+2*j -5-2*j])


Acl=A-B*K;
sys_c_cl=ss(Acl,[],C,[])
figure
initial(sys_c_cl,[1 1 1])


Ts=0.01;
sys_d=c2d(sys_c,Ts)

figure
initial(sys_d,[1 1 1])

[F,G,C,D,Ts1]=ssdata(sys_d)
rank(ctrb(F,G))
K_d=place(F,G,[-0.5 -0.1+0.2*j -0.1-0.2*j])

Fcl=F-G*K_d;
figure
sys_d_cl=ss(Fcl,[],C,[])
initial(sys_d_cl,[1 1 1])
```


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
2.5
2
1.5
1
0.5
0
0 1 2 3 4 5 6 7 8 9 10
Time (sec)


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
1.5
1
0.5
0
-1
-2
10
-10
-20
0 0.2 0.4 0.6 0.8 1 1.2
Time (sec)


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
Time (sec)
2
1.5
1
0.9
0.8
0.7
0.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
Time (sec)
2
1
0
-1
500
-500
x 10^4
10
5
-5
20
30
40
50
60



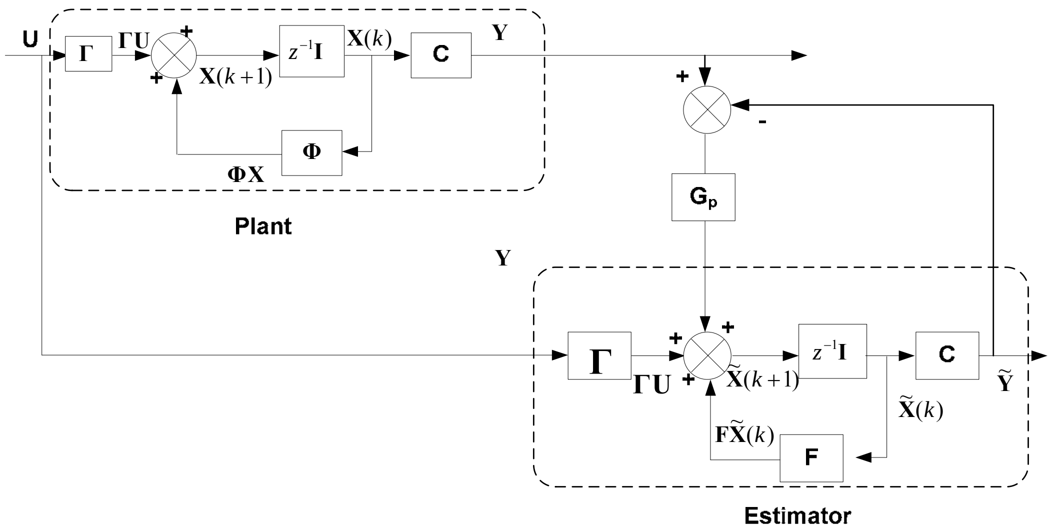
LQR Control

To use an LQR controller use the command:

>>dlqr()

Discrete Time Estimators

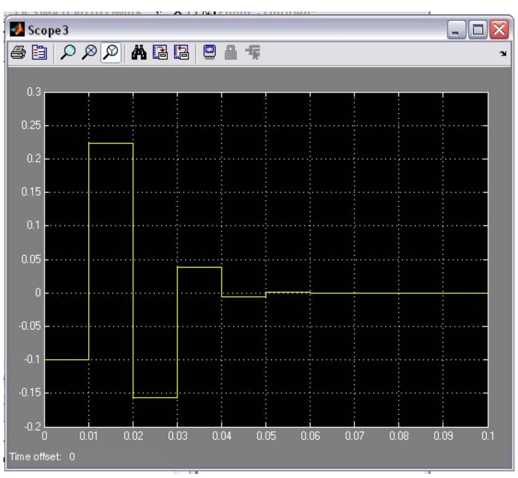
The same method can be applied at the DTE:

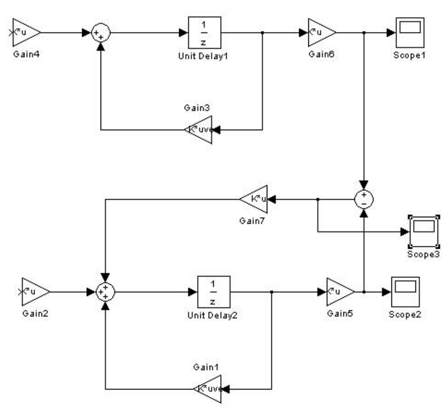


End the error dynamics are: $\mathbf{e}(k+1) = \left(\mathbf{\Phi} - \mathbf{G_p C}\right)\mathbf{e}(k)$.

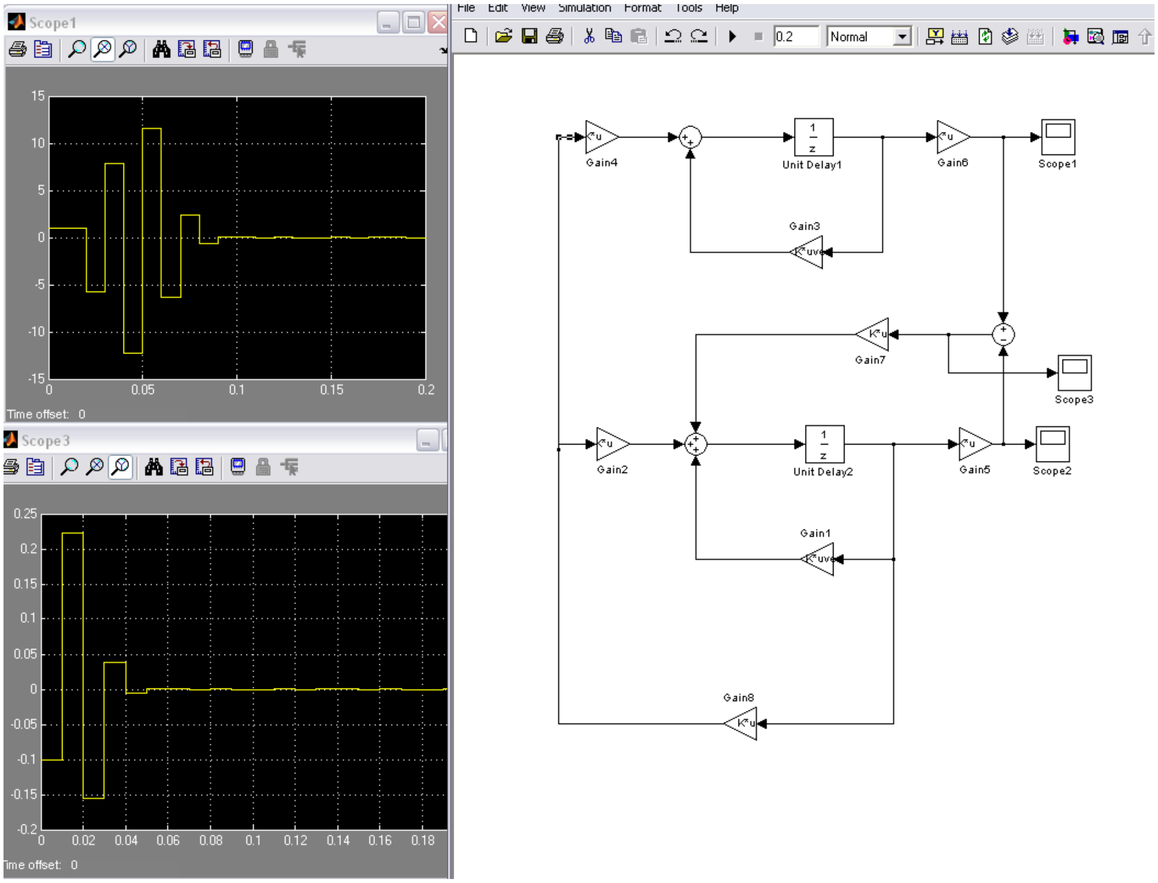
10% error of ICs:

```
F=[1 0.01 0; 0 0.99 0.0097; 0 0 0.9512];G=[0 0.002
0.0488]'; C=eye(1,3);D=0;
rank(obsv(F,C))
Gp=place(F',C',[-0.1 -0.1+0.1*j -0.1-0.1*j])' ;
```





```
K_d=place(F,G,[-0.2 -0.2+0.2*j -0.2-0.2*j])
```



As it can be seen we estimate the **X**(k) from the estimation of **X**(k+1) after a delay. To calculate the estimation of **X**(k+1) we use **Y**(k) and not **Y**(k+1). So this estimator predicts **X**(k+1) from **Y**(k). For this reason is called prediction Estimator (or priory estimator).

Current Estimators:

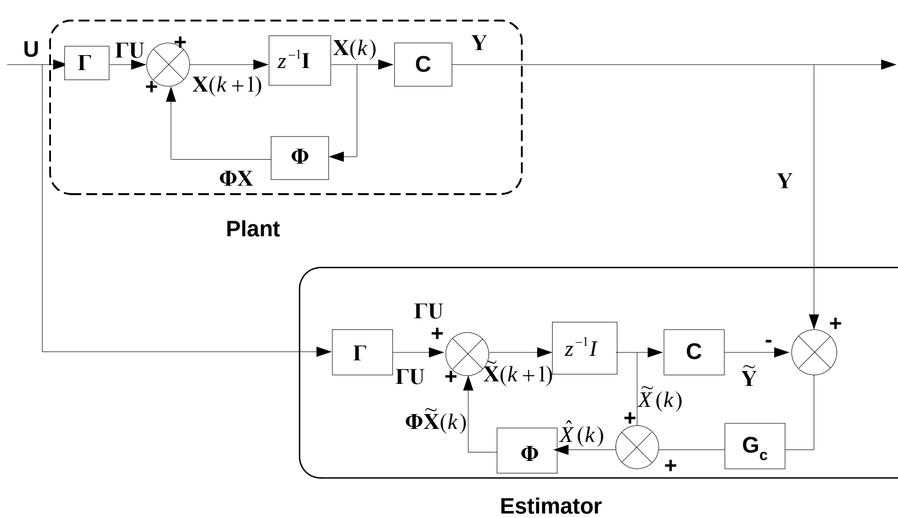
Until now the estimated state vector **X**(k) was calculated by the one sample delay of **X**(k+1). The **X**(k+1) was estimated by the measurements of **Y**(k). What about if we use the sample **Y**(k+1) to calculate the **X**(k+1) and hence the **X**(k). Then this estimator is called current estimator – or "posteriori" estimator (used in **Kalman Filter**).

The current estimator will be symbolised by: $\hat{\mathbf{X}}(k)$ and its relation with the $\tilde{\mathbf{X}}(k)$ is: $\hat{\mathbf{X}}(k) = \tilde{\mathbf{X}}(k) + G_c\left(\mathbf{Y}(k) - \mathbf{C}\tilde{\mathbf{X}}(k)\right)$

Hence it is the same as the prediction plus some extra information. But it must be noticed that for the calculation of the predictor the current is used: $\tilde{\mathbf{X}}(k) = \mathbf{\Phi}\hat{\mathbf{X}}(k-1) + \mathbf{\Gamma}\mathbf{U}(k-1)$ (OL estimation of current value)

The estimator is predicting the states and then it is using the current information of **Y**(k) to find the correct the estimating value

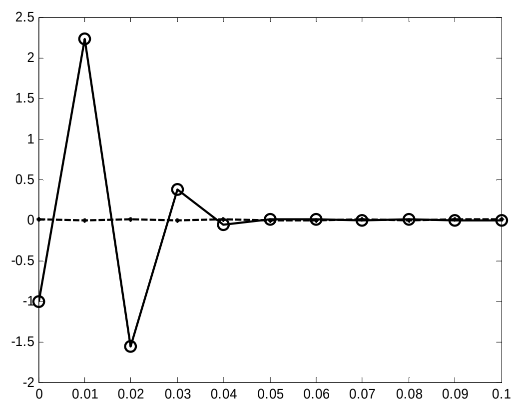
Hence:

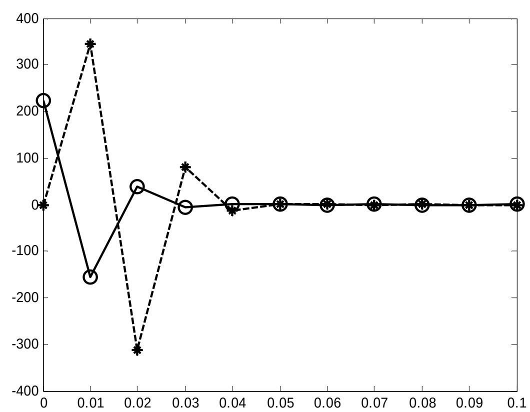


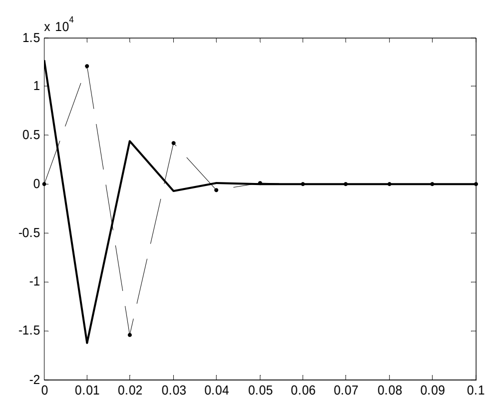
And the error dynamics are: $\mathbf{e}(k+1) = \left(\mathbf{\Phi} - \mathbf{G_c C \Phi}\right)\mathbf{e}(k)$

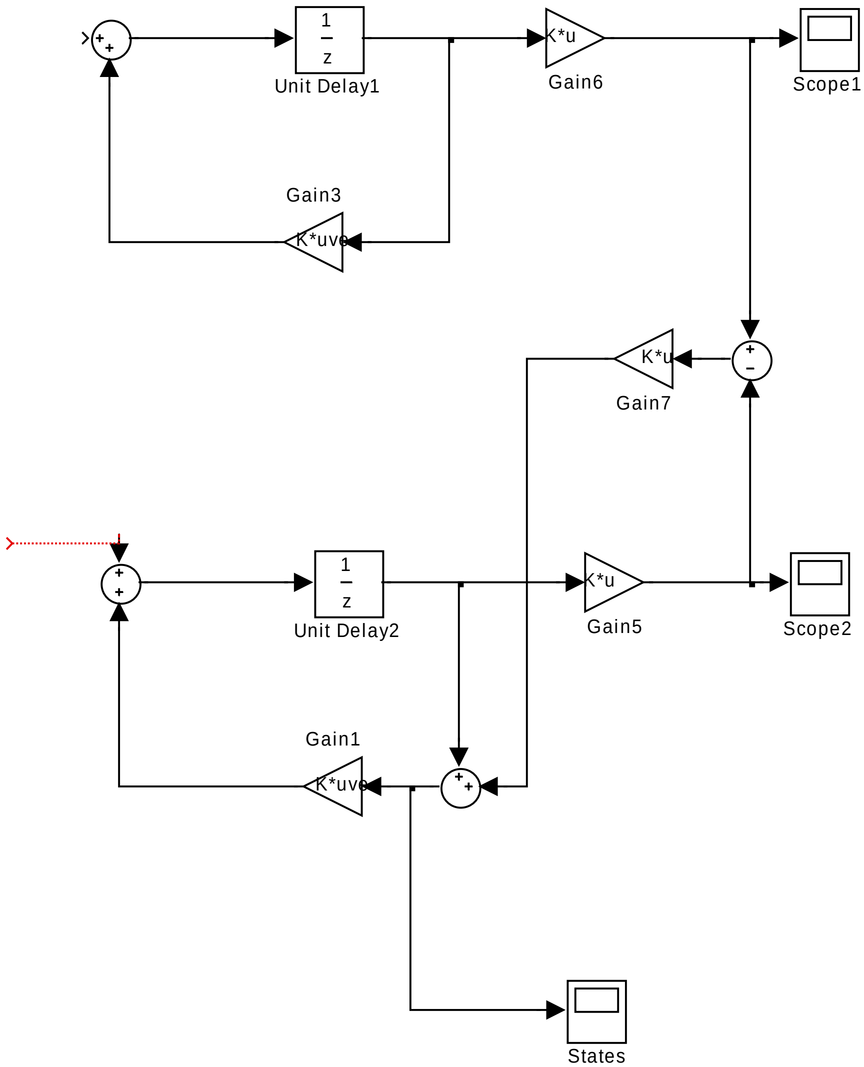
The previous example with a current estimator:

```
>> Gc=place(F',F'*C',[-0.1 -0.1+0.1*j -0.1-0.1*j])'
```








1
z
Unit Delay1
K*u
Gain6
Scope1
Gain3
K*uvec
K*u
Gain7
1
z
Unit Delay2
K*u
Gain5
Scope2
Gain1
K*uvec
States