Servo Machine for antenna azimuth control

Digital Control System

Lorenzo Vianello
Lorenzo Moreal
Alessandro Cipolla
Alessandro Mirante

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

- State Space
- Discrete Emulation
- Regulator
  - Pole placement
  - Estimator
- Disturbance rejection
- Reference input
- Deabeat controller

Model

State Space representation

The equation of motion is

$$J\theta'' + B\theta = Tc + \mathrm{Td}$$

If we set

$$\frac{B}{J} = a$$

$$u = \frac{Tc}{B}$$

$$wd = Td/B$$

Through simple passages we obtain:

$$\frac{1}{a}\theta'' + \theta' = u + wd$$

State space representation

Assuming that

$$\frac{1}{a} = 10\ s$$

Our equation become

$$10\theta'' + \theta' = u + wd$$

Using Laplace transform I can easily obtain the transfer function

$$\Theta(s) = \frac{1}{s\left(\frac{s}{a} + 1\right)}[u(s) - wd(s)]$$

Where $G(s)$ is the transfer function without disturbances

$$G(s) = \frac{\theta(s)}{u(s)} = \frac{1}{s\left(\frac{s}{a} + 1\right)}$$

State space representation

matrix equation

We chose as state of the system

$$x = \begin{pmatrix} \theta \\ \theta' \end{pmatrix}$$

So the matrix equation become

$$\begin{pmatrix} \theta' \\ \theta'' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -0.1 \end{pmatrix} \begin{pmatrix} \theta \\ \theta' \end{pmatrix} + \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} u + \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} w_d$$

$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \theta' \end{pmatrix}$$

Then we can transform them in their discrete equivalent form :

State Space transformation

It can be done with lot of emulations techniques with differents results, for example applying a ZOH with not delay we have:

$$\phi = e^{FT}$$

$$\Gamma = \int_0^T e^{Fw} G dw$$

Where T=1s is the sampling period.

State space transformation

$$x(k+1) = \Phi x(k) + \Gamma u \quad y(k) = Hx(k)$$

The new matrices are

$$\Phi = \begin{pmatrix} 1 & 0.9516 \\ 0 & 0.9048 \end{pmatrix} \quad \Gamma = \begin{pmatrix} 0.04837 \\ 0.09516 \end{pmatrix}$$

The transfer function is

$$G_1(z) = \frac{0.048(z+0.97)}{(z-1)(z-0.9048)}$$

Controllers
state space pole placement

We now assume that we have all the states available, so we can proceed to the control-law design: the control law is simply the feedback of a linear combination of all the states, that is

$$u = -Kx = -(K_1 K_2 ...) \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \end{pmatrix}$$

Substituting in the equation we have:

$$x(k+1) = \Phi x(n) - \Gamma K x(n)$$

The control- law design then consists of picking the elements of K so that the roots of are in desirable locations.

$$\alpha(z) = (z - \beta_1)(z - \beta_2)(z - \beta_3).... = 0$$

Controllability Matrix

Before calculate the values of K we must ensure that the system is controllable

To check the controllability of the system we need to calculate the rank of the controllability matrix

In our system the controllability matrix is:

$$\begin{pmatrix} 0.0484 & 0.1389 \\ 0.0952 & 0.0861 \end{pmatrix}$$

We want to place the poles in

$$s = 0.5 \pm \frac{\sqrt{3}}{2}$$

so the correspondent discrete roots are

$$z = e^{-sT}$$

Gains matrix

The gains matrix obtained is

$$(6.1157, 8.6494)$$

And the closed loop $\Phi \mathrm{cl} = \Phi - K\Gamma$ is

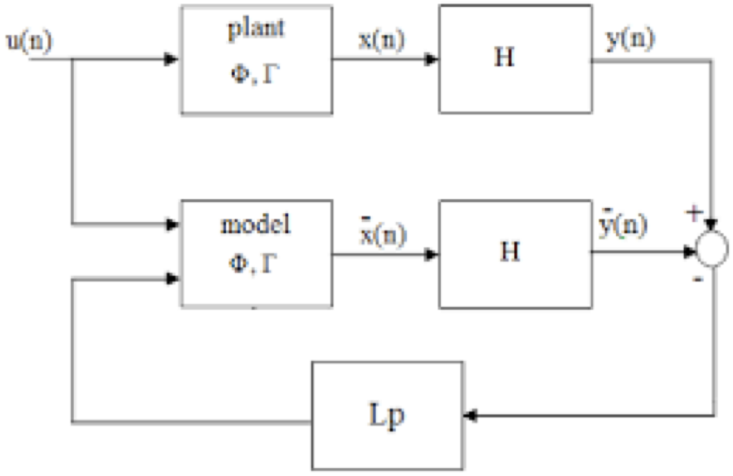
$$\Phi_K = \begin{pmatrix} 0.7042 & 0.5332 \\ -0.582 & 0.08174 \end{pmatrix}$$

Speed Observer

- Looking the H matrix we can see that we can not measure all the state, but only θ value.

So our purpose is to reconstruct all the states given measurements of a portion of them.

Using the following scheme



Closed-loop Estimator

Speed Observer

And its equation is:

$$\hat{x}(n+1) = \Phi\hat{x}(n) + \Gamma u(n) + L[y(n) - H\hat{x}(n)]$$

Typically L can be chosen so that the system is stable, and to select it we use the same approach seen for design the control law.

Speed Observer

- The observability matrix in our system is:

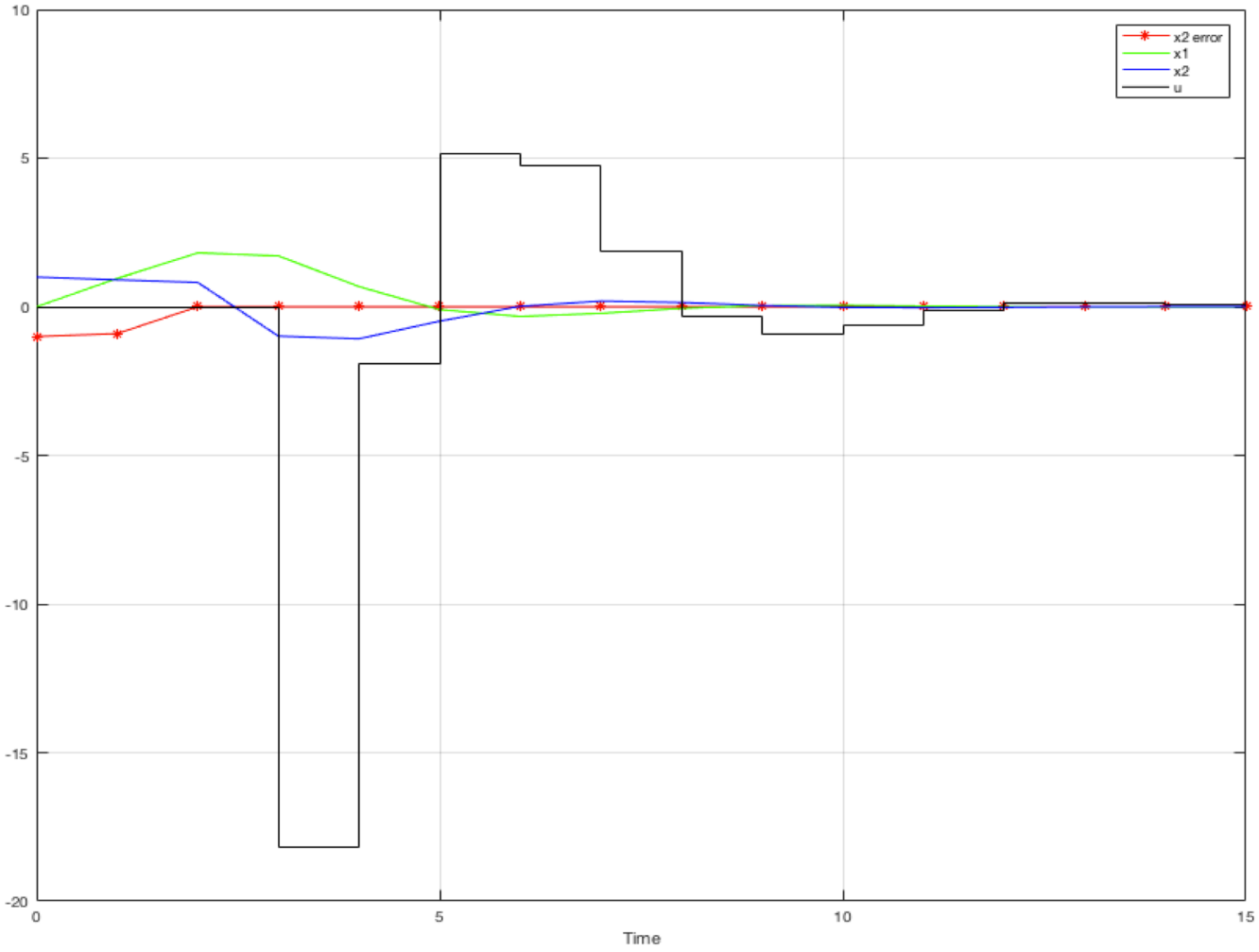
$$\begin{pmatrix} 1 & 0 \\ 0.7042 & 0.5332 \end{pmatrix}$$

It is full rank so we can implement the observer; if the desired characteristic equation is $\alpha(z) = z^2$

the values of the parameters L_P calculated are

$$Lp = \begin{matrix} 1{,}9048 \\ 0{,}8603 \end{matrix}$$

State Estimation

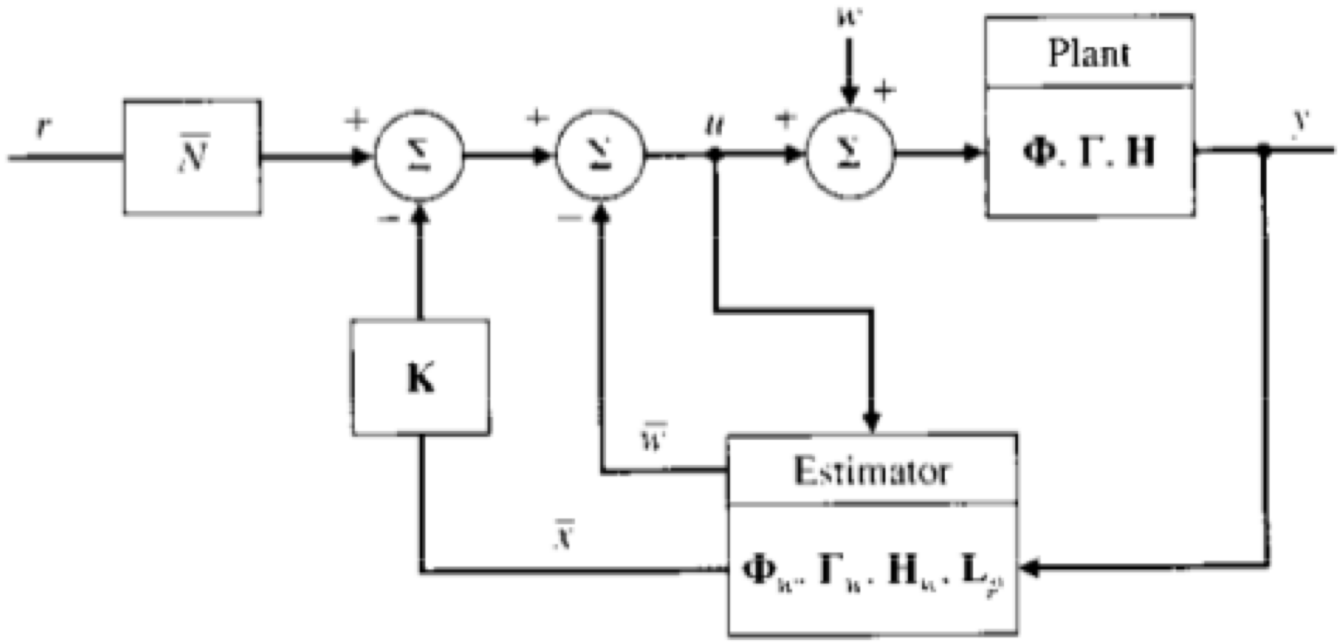


Disturbance estimation

Our approach is to estimate the disturbance signal in the estimator and then to use that estimate in the control law so as to force the error to zero.

This approach is called disturbance rejection.

Following the presented scheme



Disturbance estimation

For purposes of disturbance estimation we augment the system model with the disturbance model:

$$\begin{pmatrix} x(k+1) \\ w(k+1) \end{pmatrix} = \begin{pmatrix} \Phi & \Gamma_w \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \end{pmatrix} + \begin{pmatrix} \Gamma_u \\ 0 \end{pmatrix} u(k)$$

$$y(k) = \begin{pmatrix} H & 0 \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \end{pmatrix}$$

Disturbance estimation

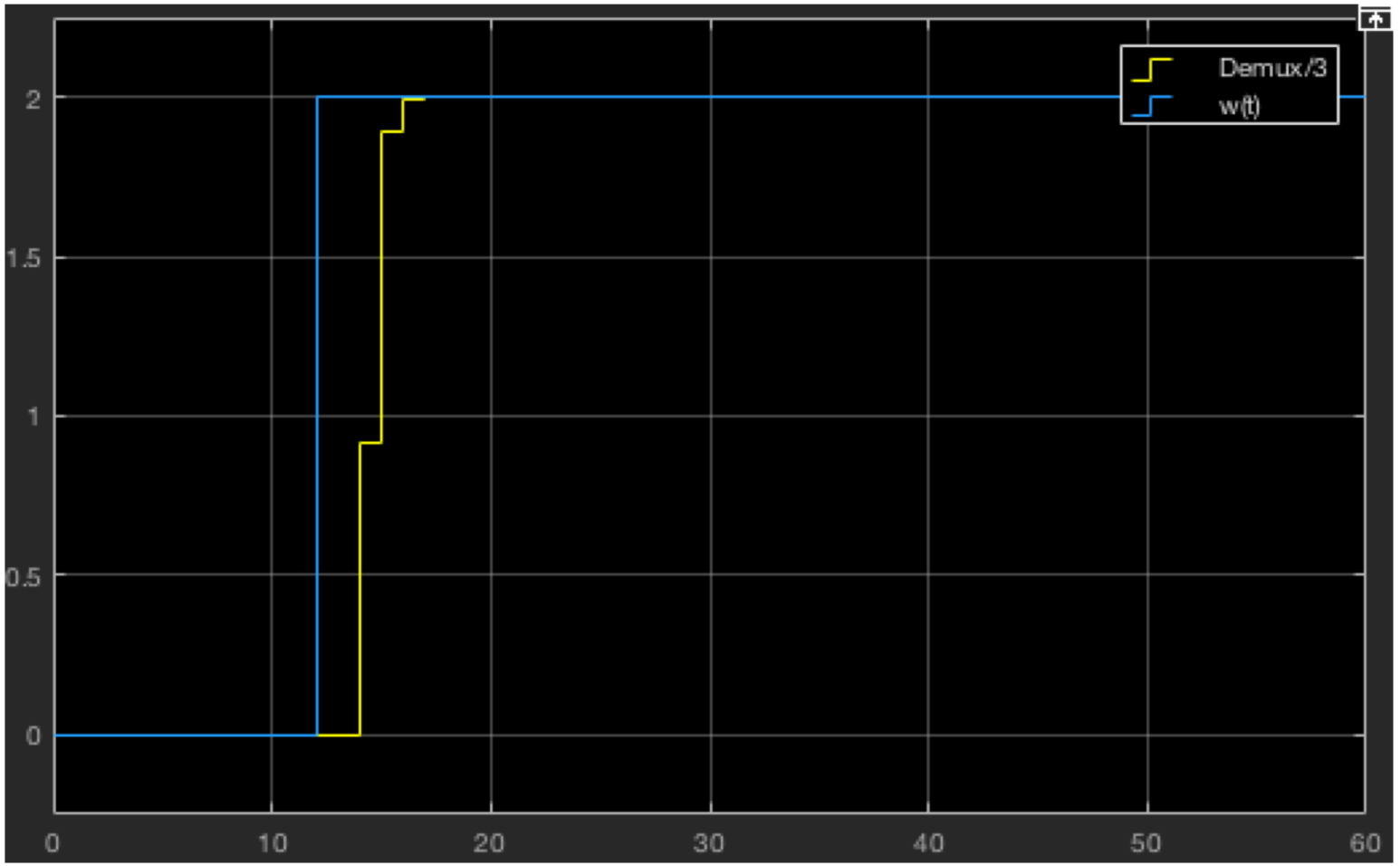
- From them if we desire poles of the estimator in

$$(0\ 0\ 0.1)$$

We obtain

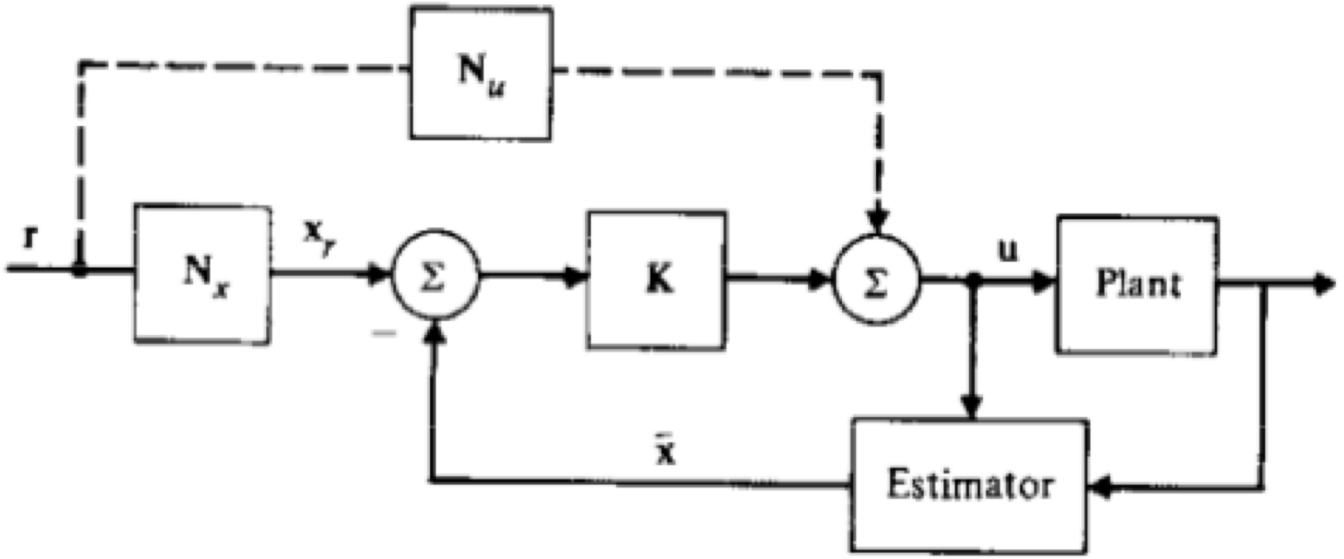
$$Lp' = \begin{matrix} 2.8 \\ 2.18 \\ 9.46 \end{matrix}$$

Disturbance Estimation



Reference input

- The structure of a reference input consist of a state command matrix Nx that defines the desired value of the state x_r.



Reference input

- The basic idea in determining N_x is that it should transform the reference input, r, to a reference state

that is an equilibrium one for that r.

More specifically, we have defined N_r so that

$$N_x r = x_r$$

$$u = -K(x - x_r)$$

In our project we calculated

$$N_x = (1\ 0)^T$$

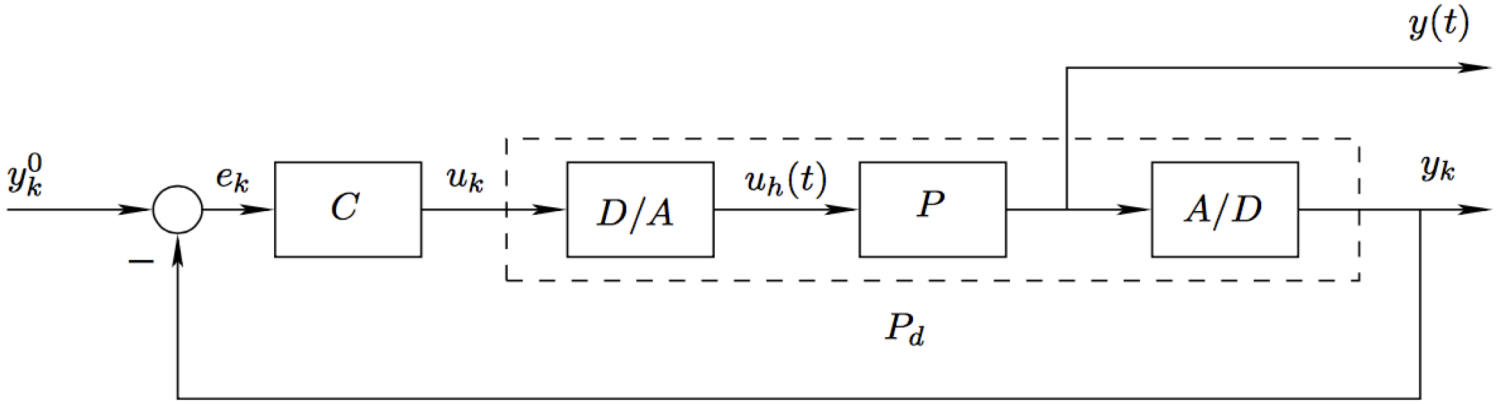
$$N_u = 6.11$$

With

$$N_{bar} = 58.9401$$

Deadbeat control

Deadbeat is an analytical method for the synthesis of digital control laws.



The deadbeat specifications refer to the closed loop system response for test signals (step, ramp,...):

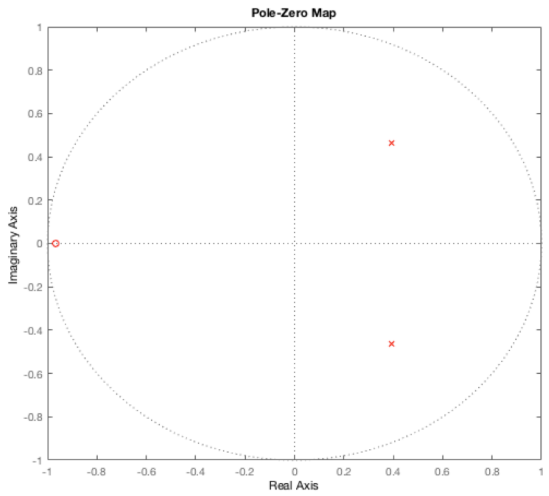
- the output reaches the final value in finite time, i.e. in a minimum number of period
- zero steady state error

The control law obtained must respect the causality property.

Deadbeat control – step input

The zero order hold equivalent of P(s) is considered:

$$P(z) = \frac{0.29584(z + 0.9672)}{z^2 - 0.7859z + 0.3679}$$



Step input: $R(z) = \frac{z}{z-1}$

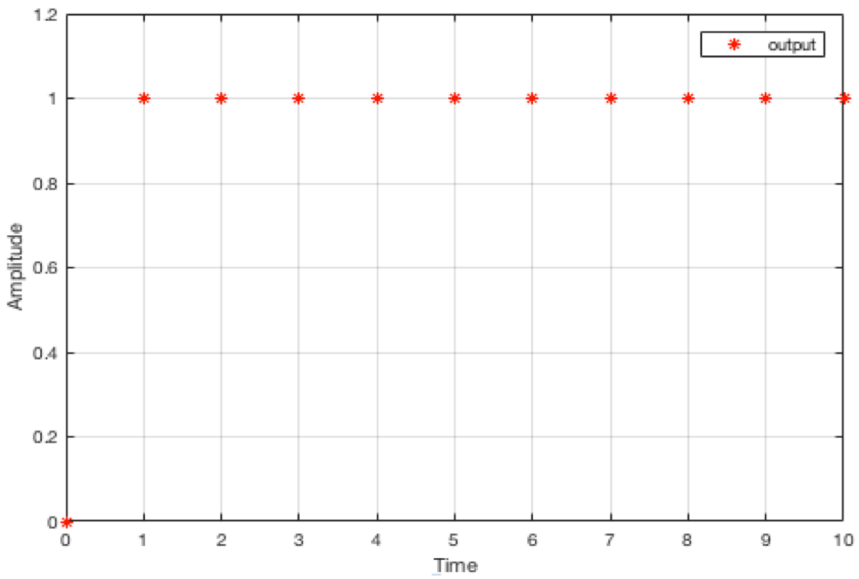
Appling the control: $C(z) = \frac{K_d}{P(z)} \frac{1}{z-1}$

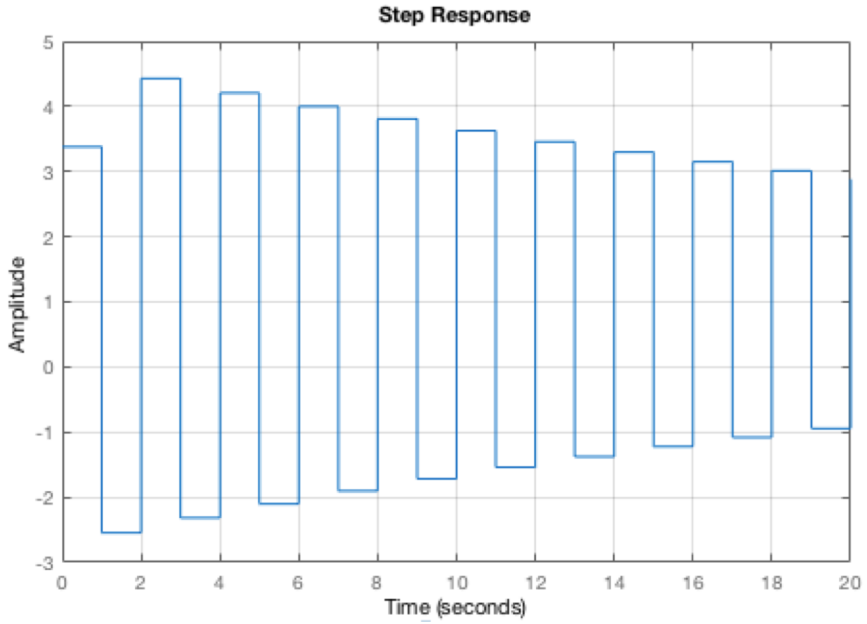
We obtain a desired closed-loop transfer function: $T(z) = \frac{1}{z}$

Sensitivity control transfer function: $S_u(z) = \frac{U(z)}{R(z)} = \frac{3.3802(z^2 - 0.7859z + 0.3679)}{z(z + 0.9672)}$

There is a root at $z = -0.9672$ this is the source of the oscillation in the control response

Deadbeat control – step input response

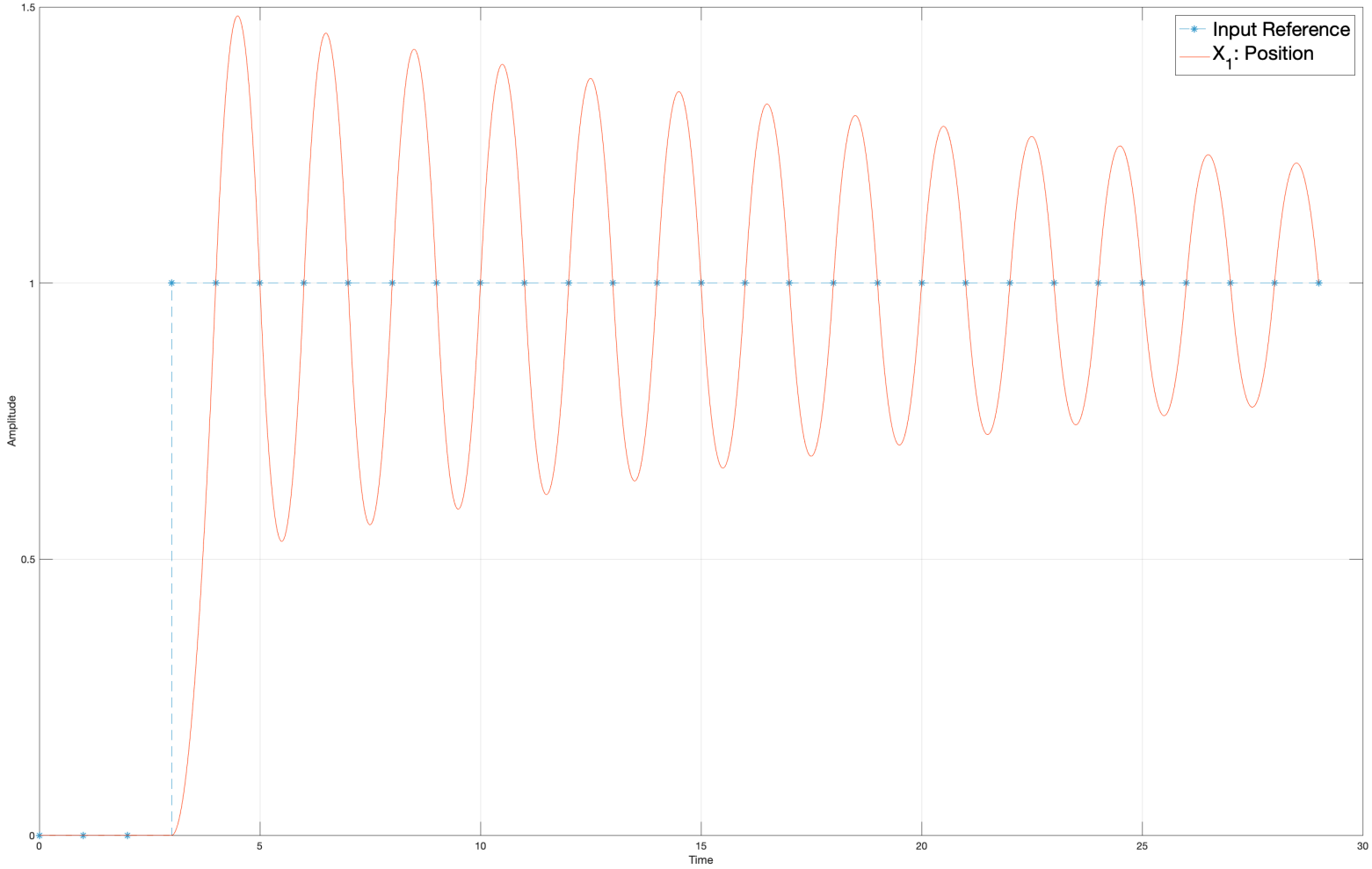




The control signal has large oscillation that causes the “intersample-ripple” in the output response.

Note that, if only the output at the sample point had been determined, the system would appear to have very good response.

Deadbeat Controller – intersampling ripple



Anti-ripple Deadbeat control - step responce

To ensure that in the system, after the transient, no oscillations ("ripple") occur between the sampling instants, the following condition must be imposed:

$y(t)$ constant for $t > NT$ for step inputs

this condition is reached if the following conditions are met on the error signal and on the control signal (or the equivalent on the loop function F(z)):

$$P^* = \lim_{s \to 0} \frac{1}{P(S)}$$

$e(h)=0$ for $h \geq k$

$m(h) = P^* K_d$

$D_F(z)$ has a solution in $z = 1$

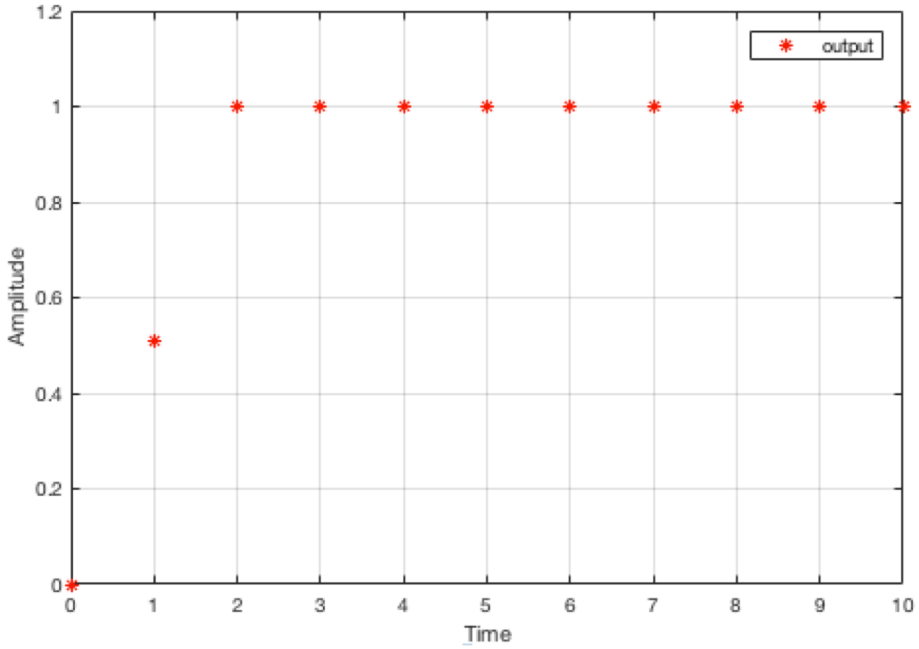
$N_F(z) + D_F(z) = z^k$

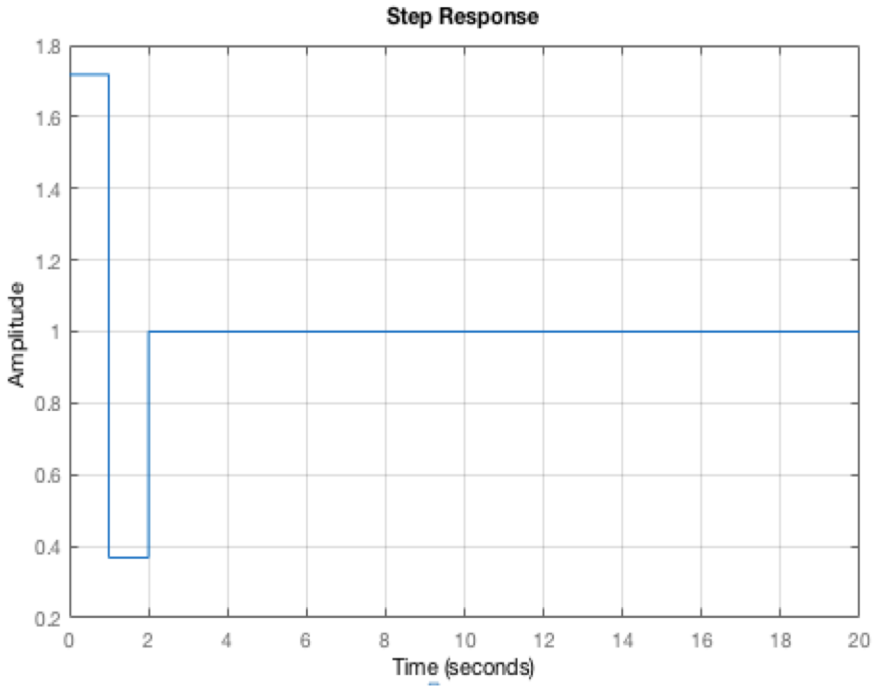
no pole-zero cancellation between $C(z)$ and $P(z)$

Applying these conditions, we arrived through an analytical procedure to the anti-ripple deadbeat controller:

$$C(z) = \frac{1.7183(z^2 - 0.7859z + 0.367)}{z + 0.4917} \frac{1}{z - 1}$$

Anti-ripple Deadbeat control – step responce

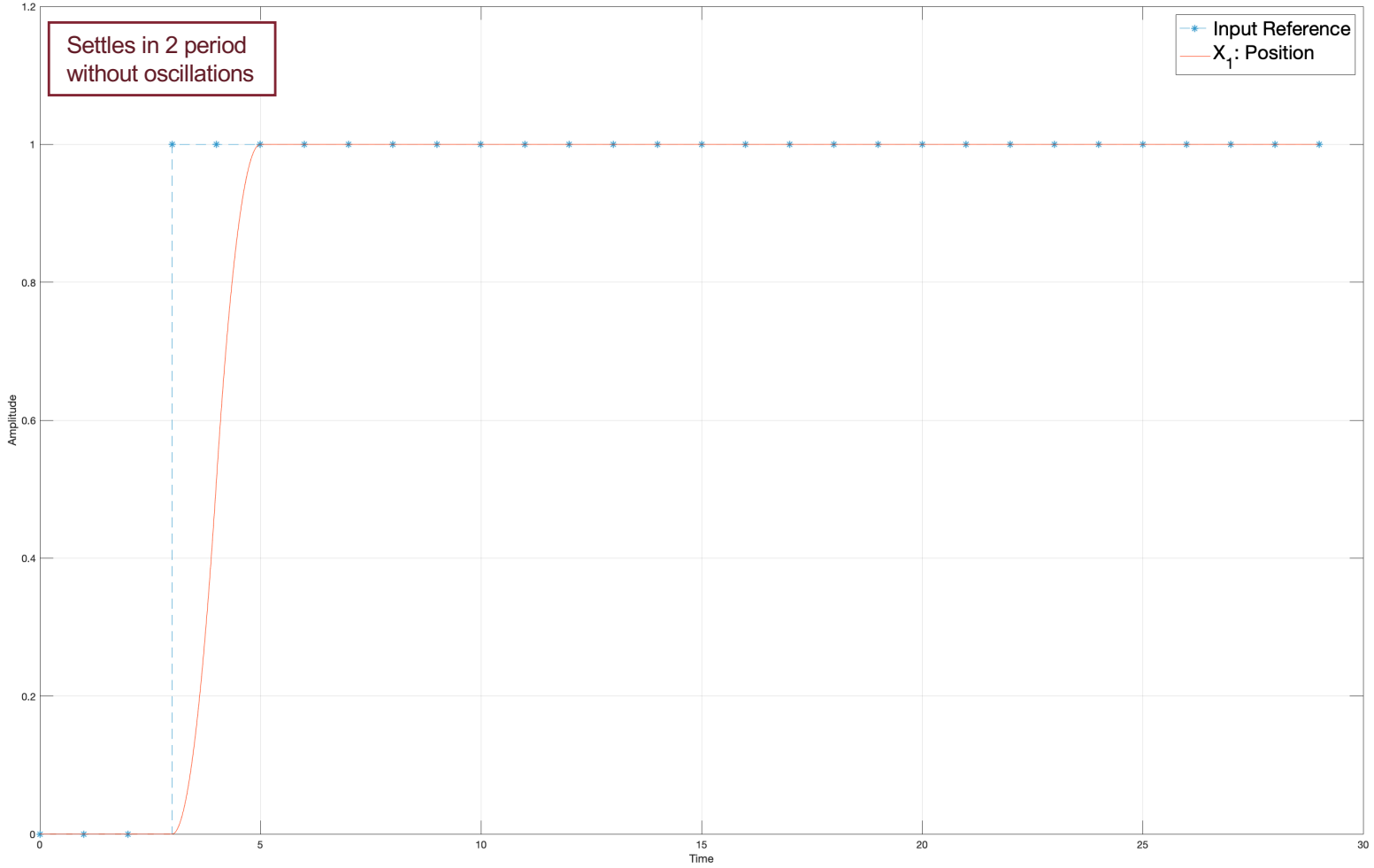




Both the control signal and the output settle in finite time.

y(t) now is not being excited by an oscillation in the control and so there isn't the inter-sampling ripple effect.

Deadbeat Controller -



Deadbeat Controller - Ramp input

Our plant (Z-transform)

$$P(z) = \frac{0.29584(z + 0.9672)}{z^2 - 0.7859z + 0.3679}$$

Ramp Input (Z - transform)

$$R(z) = \frac{z}{(z-1)^2}$$

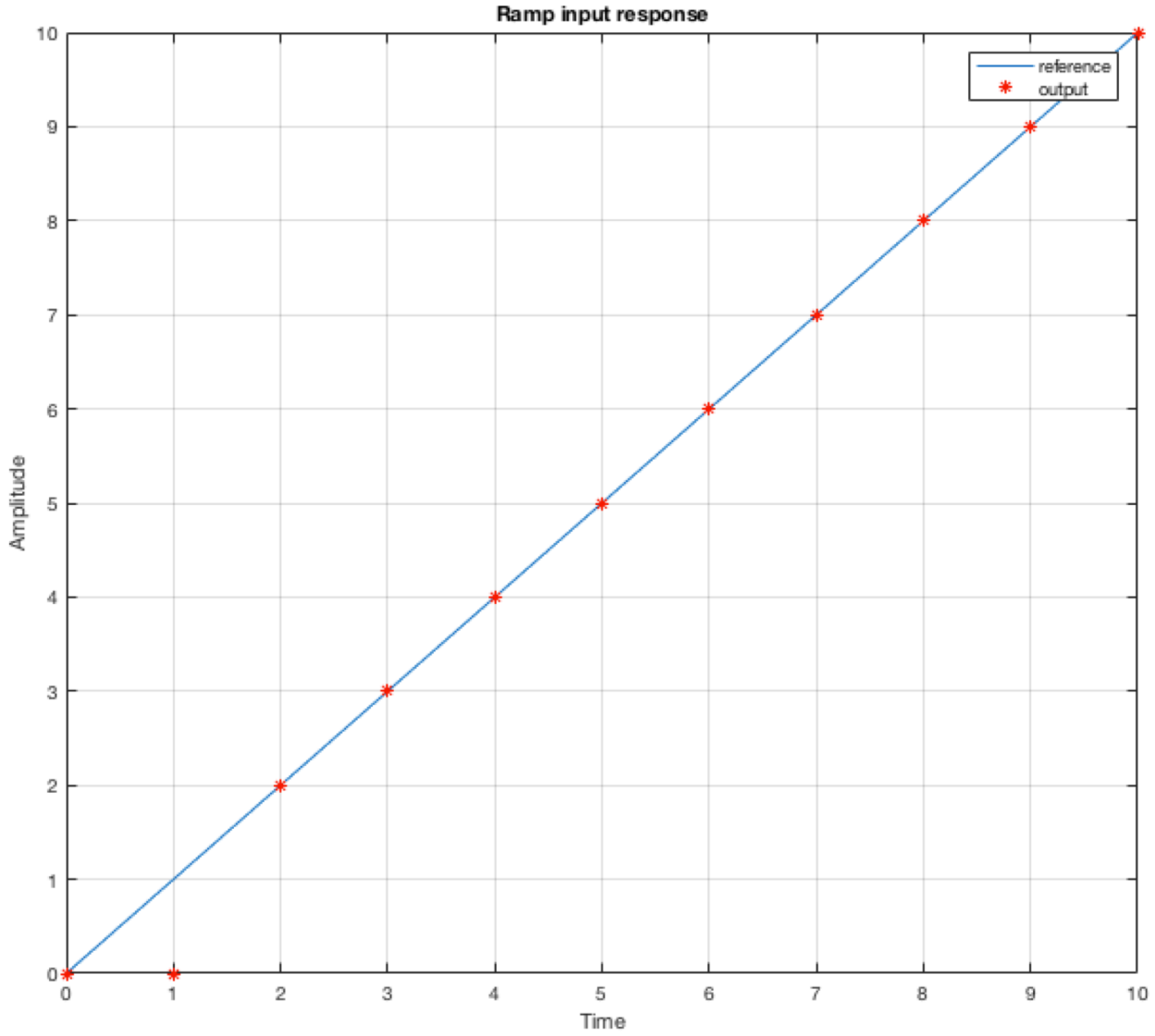
Aimed Trasnfer Function (Z-transform)

$$T(z) = \frac{2z - 1}{z^2}$$

Deadbeat Controller

$$C(z) = \frac{T(z)}{P(z)(1 - T(z))} = \frac{6.7604\,(z - 0.5)\,(z^2 - 0.7859z + 0.3679)}{(z + 0.9672)(z-1)^2}$$

Deadbeat control – ramp responce



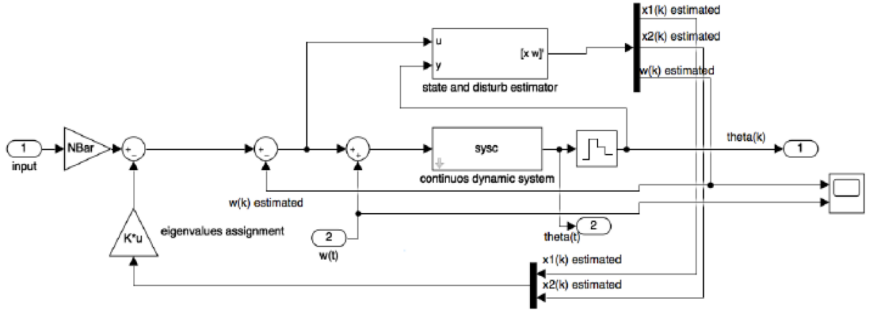
Development Simulator

Provide a way to check the operation of these control system techiques

- Physical Realizability of the controllers (deadBeat)
- Intersampling phenomena
- Very High Resolution (integration with ODE45)

```
function dxdt = ODE_sys(t,x,hummer)

a=0.1;
dxdt= zeros(2,1);

dxdt(1) = x(2);
dxdt(2) = - a*x(2) + 0.1*hummer;


end
```

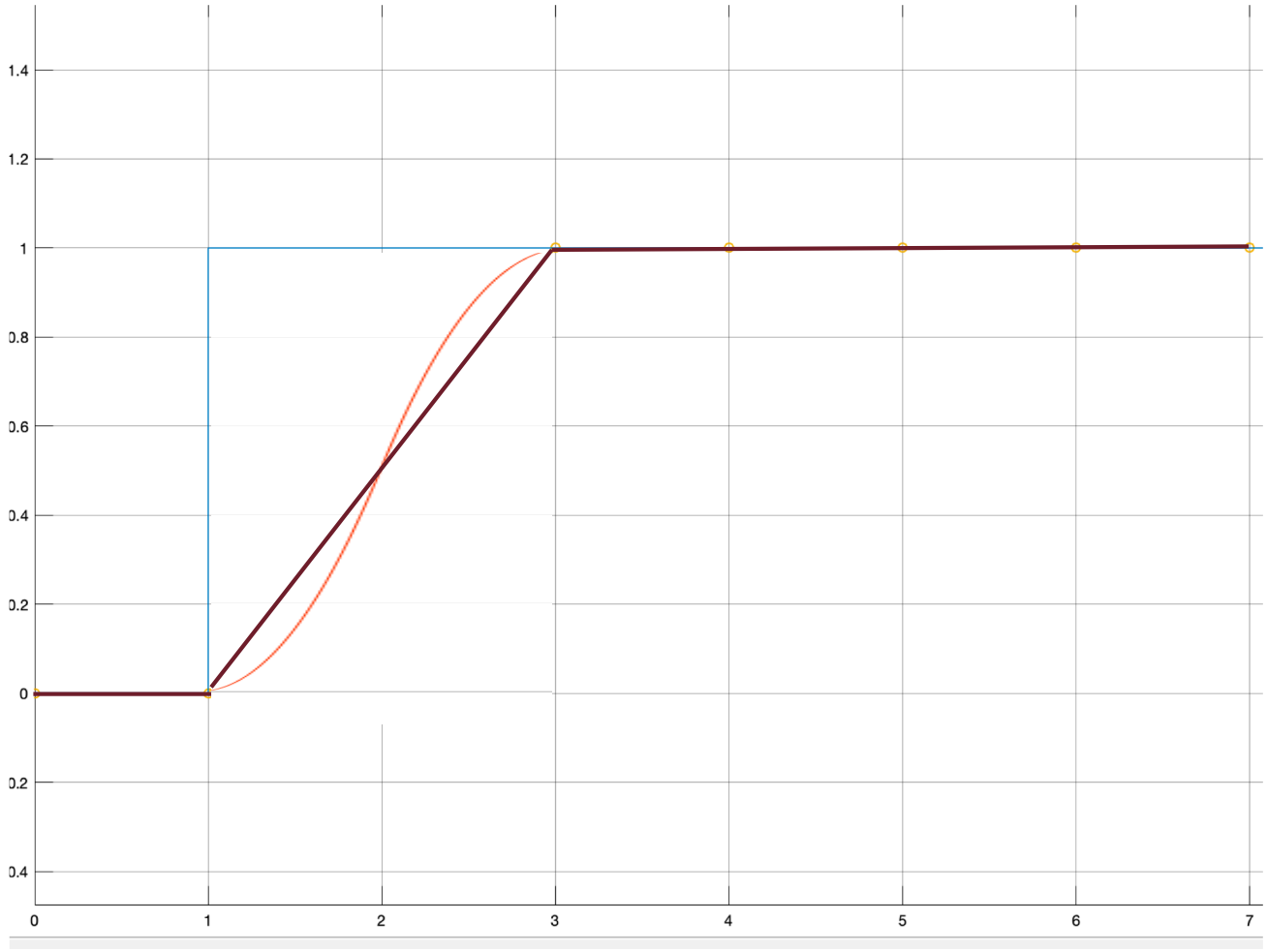
Simulink → Matlab Code



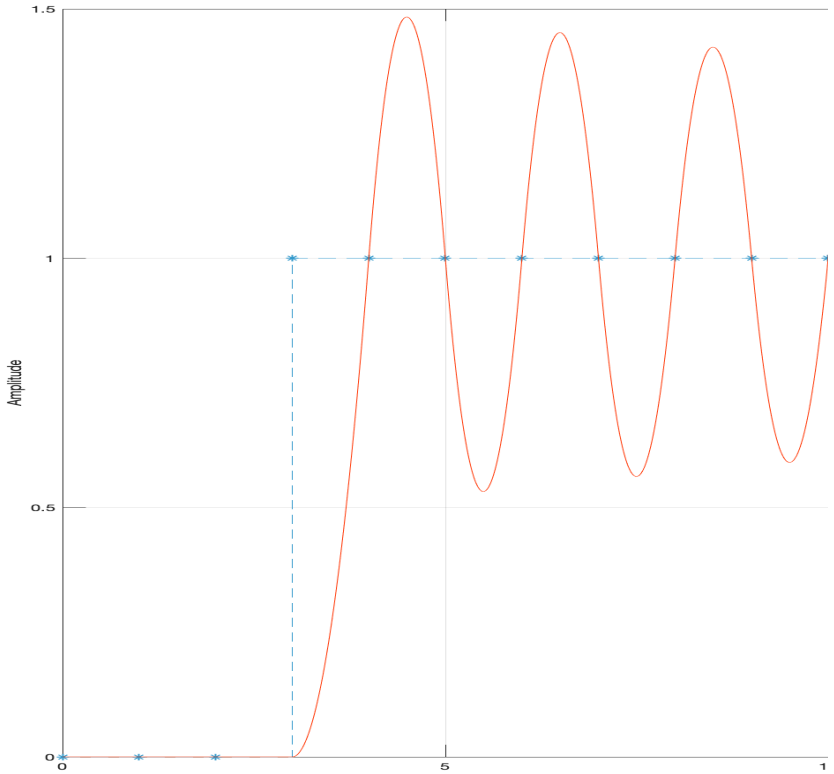
Very Poor resolution for mixed model, so lost of informations.
We have seen that for 'mixed' simulations matlab discretize the overall system

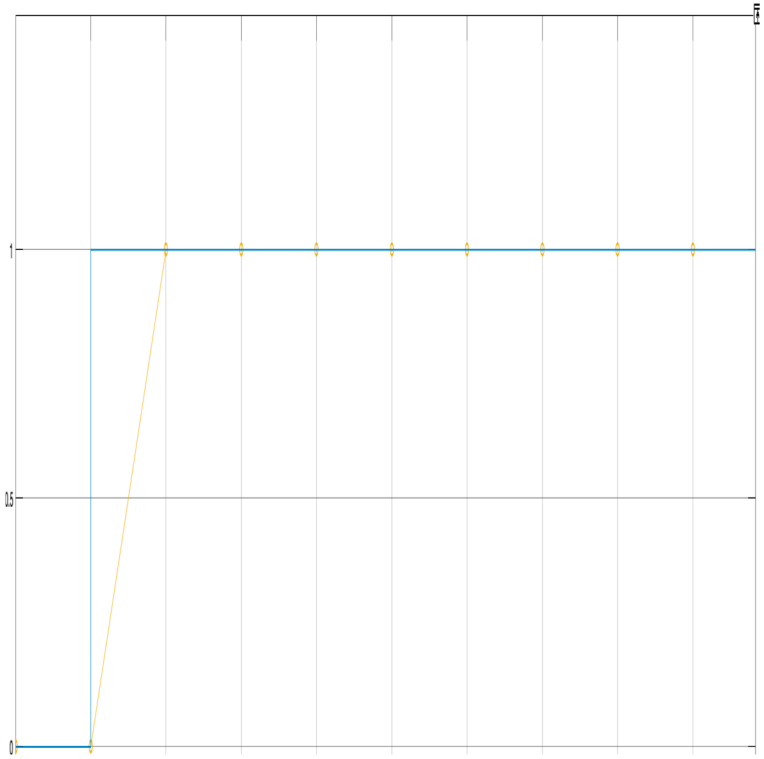
```
options = odeset('Reltol',reltol,'Abstol',abstol);

x0=[0 0];
while par<time_simulation-sampling_time

     e(k)=(u(k)-FB);

    %------------------------Deadbeat Control-------------------------------

     x_control(:,k+1) = A_c*x_control(:,k) + B_c*e(k) ;
     m(k) = C_c*x_control(:,k)+ D_c*e(k);

    %--------------------Augmented Plant State Space-----------------------

    x(1:2,k+1) = Ad*x(1:2,k) + Bd.*(m(k) - K*x_hat(1:2,k)+...
                 tau(k) - x_hat(3,k));
    y(k) = Cd*x(1:2,k);


    %----------------------Continuos Time---------------------------------

    t_int=par:microsteps:par+delta;

    hummer = m(k) - K*x_hat(1:2,k)+ tau(k) - x_hat(3,k);

    [t0,y0] = ode45(@(t,x) ODE_sys(t,x,hummer),t_int,x0,options);

    x0=y0(end,:);

    t_1(j*(numofsteps+1)+1:(j+1)*(numofsteps+1),1)=t0;
    y_1(j*(numofsteps+1)+1:(j+1)*(numofsteps+1),1:2)=y0;

    j=j+1;
```

Development Simulator – Simulink vs Matlab Coding



Development Simulator – Simulink vs Matlab Coding





Development Simulator

POSSIBLE SCENSARIO

Pole Placement
State Space Estimation
Disturbance Estimation and Rejection

Controlller: DeadBeat for Ramp input

- Input: ramp + costant
- State: Position
- Disturbance: Wind Gusts 100

Evolution of the system is seen in continuous but the controllers and other components work in discrete time