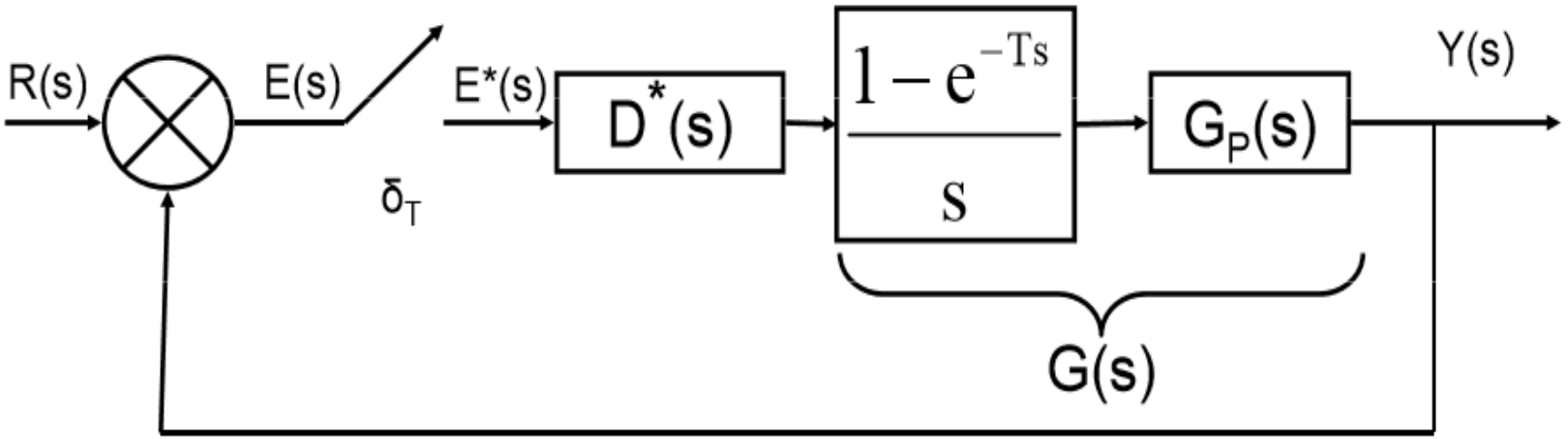
- PID controller is the most common form of feedback. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s.
- In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used
- Practically all PID controllers made today are based on microprocessors



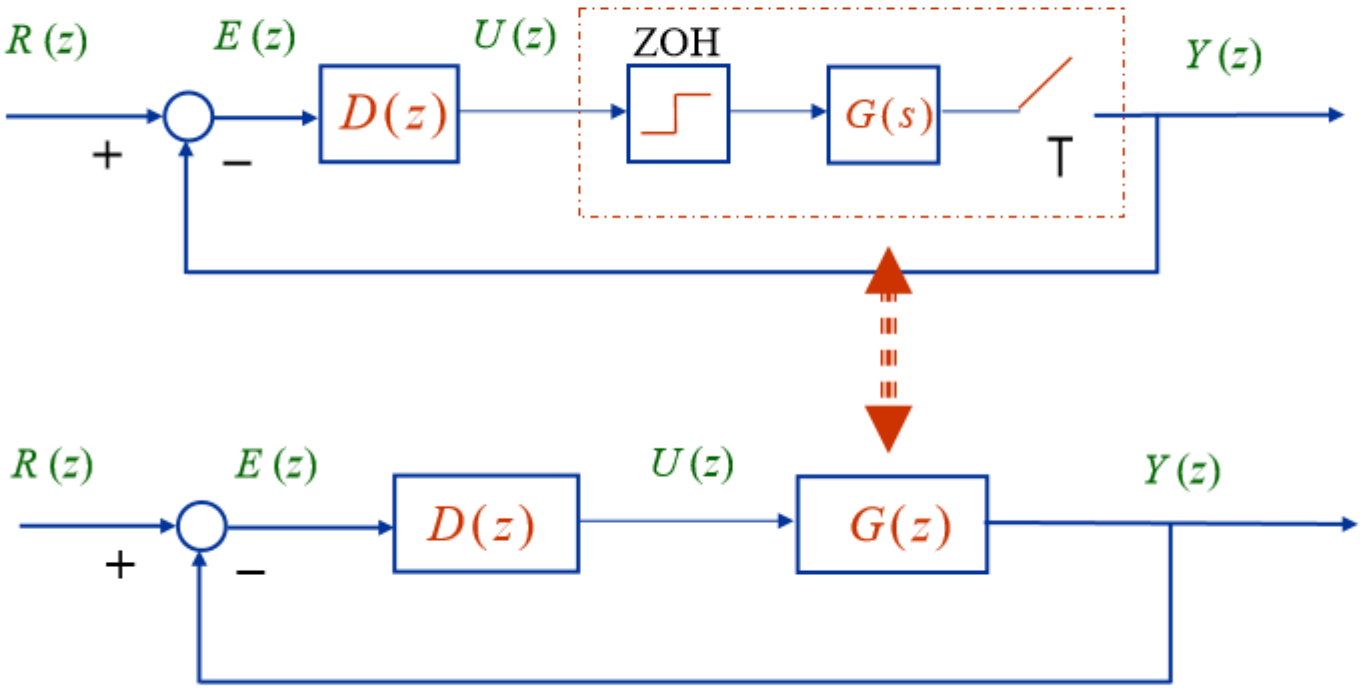
$$\begin{cases} Y(s) = G(s)D^*(s)E^*(s) \\ E(s) = R(s) - Y(s) \end{cases} \Rightarrow \begin{cases} Y^*(s) = G^*(s)D^*(s)E^*(s) \\ E^*(s) = R^*(s) - Y^*(s) \end{cases}$$

$$\Rightarrow Y(z) = G(z)D(z)\big(R(z) - Y(z)\big)$$

$$\boxed{\frac{Y(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)G(z)}}$$

DIGITAL PID CONTROL SYSTEM DESIGN VIA POLE PLACEMENT TECHNIQUE

In this approach, we discretize the continuous-time plant first or directly work on a discrete-time plant to design a digital controller using the well known PID framework.



where $G(z) = (1 - z^{-1})\mathbf{Z}\left\{\frac{G(s)}{s}\right\}$ and $D(z)$ is taken to be a PID controller.

PID CONTROL

PID control is widely used in process control and most of industrial control systems. Unknown source reports that more than 90% of industrial processes are actually controlled by PID type of controllers. PID control consists of three essential components, namely, **P** (proportional control), **I** (integral control) and **D** (derivative control).

Proportional Control

A discrete implementation of proportional control is identical to continuous. The continuous is

$$u(t) = k_p e(t) \quad \Rightarrow \quad D(s) = k_p$$

The discrete is

$$u(k) = k_p e(k) \quad \Rightarrow \quad D(z) = k_p$$

where $e(t)$ or $e(k)$ is the error signal as given in the feedback block diagram.

Derivative Control

The continuous derivative control is

$$u(t) = k_d \dot{e}(t) \quad \Rightarrow \quad D(s) = k_d s$$

The discrete derivative control is

$$u(k) = k_d \frac{e(k) - e(k-1)}{T} \quad \Rightarrow \quad D(z) = k_d \frac{1 - z^{-1}}{T} = k_d \frac{z-1}{Tz}$$

Integral Control

The continuous integral control is $u(t) = k_i \int_{t_0}^{t} e(t)dt \quad \Rightarrow \quad D(s) = k_i \frac{1}{s}$

The discrete integral control is

$$u(k) = u(k-1) + k_i T e(k) \quad \Rightarrow \quad D(z) = \frac{k_i T}{1 - z^{-1}} = \frac{k_i T z}{z-1}$$

Digital PID Control (conventional version)

$$D(z) = k_P + k_I \frac{z}{z-1} + k_D \frac{z-1}{z} = \frac{(k_P + k_I + k_D)z^2 - (k_P + 2k_D)z + k_D}{z(z-1)}$$

Digital PI Control (conventional version)

Digital PI control consists of only P and I actions and is given by

$$D(z) = k_P + k_I \frac{z}{z-1} = \frac{(k_P + k_I)z - k_P}{z-1}$$

Digital PD Control (conventional version)

Digital PD control consists of only P and D actions and is given by

$$D(z) = k_P + k_D \frac{z-1}{z} = \frac{(k_P + k_D)z - k_D}{z}$$

Digital PID Control (via bilinear transformation)

$$D(z)=\left(k_p+\frac{k_i}{s}+k_d s\right)_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}=k_p+\frac{k_i T(z+1)}{2(z-1)}+\frac{2k_d(z-1)}{T(z+1)}=\frac{\alpha_2 z^2+\alpha_1 z+\alpha_0}{(z-1)(z+1)}$$

where α_0, α_1 and α_2 are design parameters.

Digital PI Control (via bilinear transformation) – the same as the previous version

$$D(z)=\left(k_p+\frac{k_i}{s}\right)_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}=k_p+\frac{k_i T(z+1)}{2(z-1)}=\frac{\alpha_1 z+\alpha_0}{z-1}$$

Digital PD Control (via bilinear transformation)

$$D(z)=\left(k_p+k_d s\right)_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}=k_p+\frac{2k_d(z-1)}{T(z+1)}=\frac{\alpha_1 z+\alpha_0}{z+1}$$

DESIGN EXAMPLE

Consider a car (BMW), which has a weight $m = 1000\ kg$. Assuming the average friction coefficient $b = 100$, design a cruise control system such that the car can reach $100\ km/h$ from $0\ km/h$ in $8\ s$ with an overshoot less 20%.

Assuming the sampling period $T = 0.6$ seconds, design a digital PI controller that achieve the above specifications.

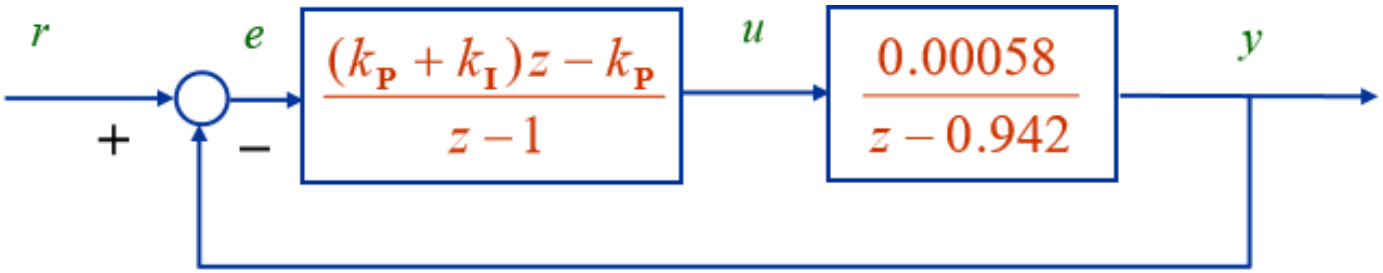
$$G(s) = \frac{1}{ms+b} = \frac{1}{1000s+100} \quad \Rightarrow$$

$$G(z) = (1-z^{-1})\mathbf{Z}\left\{\frac{G(s)}{s}\right\} = (1-z^{-1})\mathbf{Z}\left\{\frac{0.001}{s(s+0.1)}\right\} = \frac{z-1}{z}\mathbf{Z}\left\{\frac{0.01}{s} - \frac{0.01}{s+0.1}\right\}$$

$$= \frac{0.00058}{z-0.942}$$

discretized plant with T = 0.6

DISCRETIZED PLANT WITH DIGITAL PI CONTROLLER



The resulting closed-loop transfer function from r to y is given by

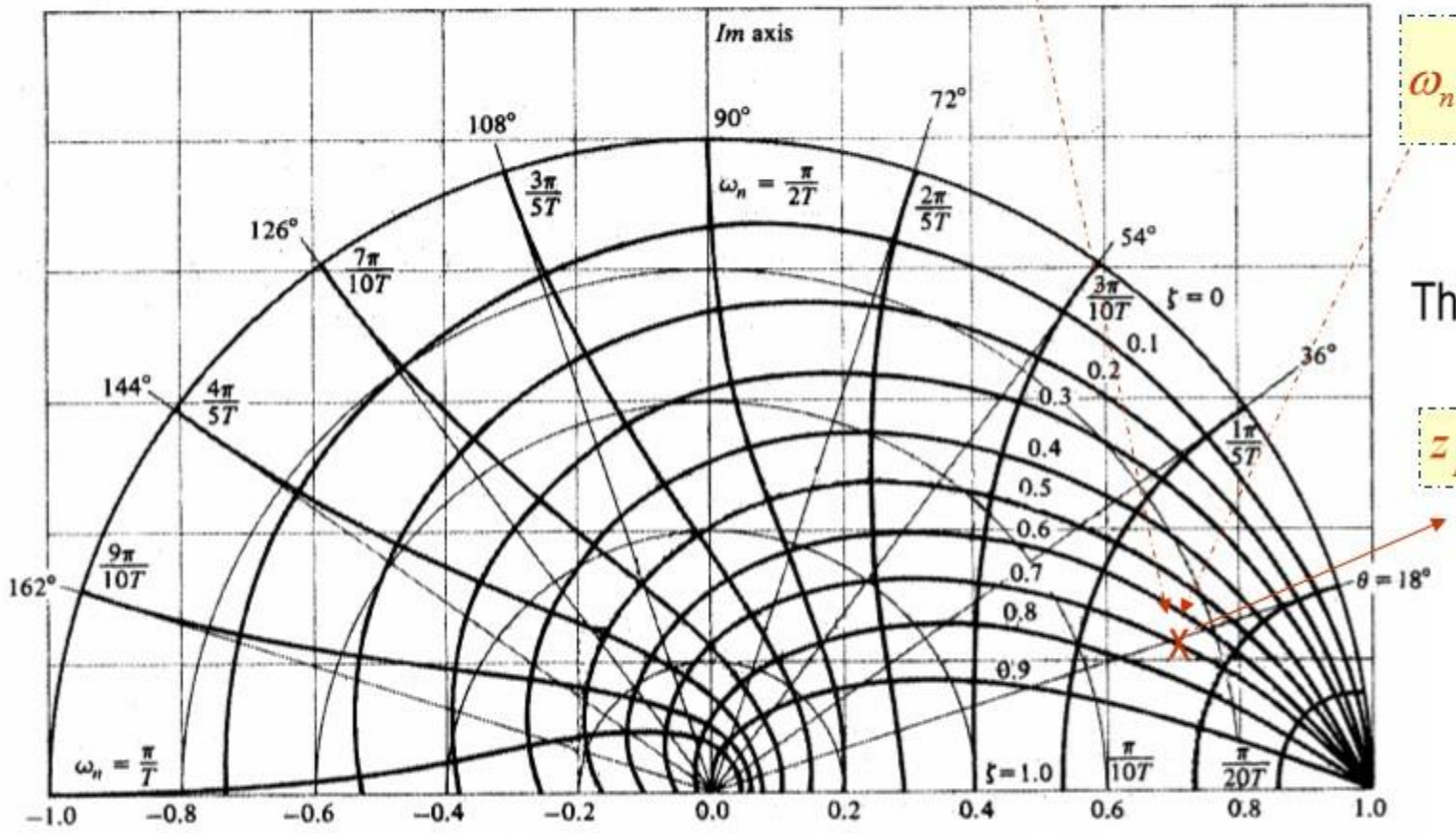
$$H(z) = \frac{G(z)D(z)}{1+G(z)D(z)} = \frac{\dfrac{0.00058}{z-0.942}\cdot\dfrac{(k_P+k_I)z-k_P}{z-1}}{1+\dfrac{0.00058}{z-0.942}\cdot\dfrac{(k_P+k_I)z-k_P}{z-1}}$$

$$= \frac{0.00058\,(k_P+k_I)z - 0.00058\,k_P}{z^2 + [0.00058\,(k_P+k_I) - 1.942\,]z + (0.942 - 0.00058\,k_P)}$$

DESIRED CLOSED-LOOP TRANSFER FUNCTION

From the design in we obtain the desired $\zeta = 0.7$ and $\omega_n = 0.82$ in continuous setting,

which would achieve the design specifications. Using the following chart with $T = 0.6$



z = plane loci of roots of constant ζ and ω_n

$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$

$z = e^{Ts}$

T = sampling period

$$\omega_n = 0.82 \approx 0.16\frac{\pi}{T}$$

The desired poles:

$$z_p = 0.7 + j0.2$$

$$H_{\text{desired}}(z) = \frac{(1-z_p)(1-\bar{z}_p)}{(z-z_p)(z-\bar{z}_p)} = \frac{0.13}{z^2 - 1.4z + 0.53}$$

DETERMINATION OF THE PI CONTROLLER PARAMETERS

Comparing the denominator of the actual closed-loop transfer function

$$H(z) = \frac{0.00058\,(k_P + k_I)z - 0.00058\,k_P}{z^2 + [0.00058\,(k_P + k_I) - 1.942\,]z + (0.942 - 0.00058\,k_P)}$$

with that of the desired one

$$H_{\text{desired}}(z) = \frac{0.13}{z^2 - 1.4z + 0.53}$$

we obtain

$$\begin{cases} 0.00058\,(k_P + k_I) - 1.942 = -1.4 \\ 0.942 - 0.00058\,k_P = 0.53 \end{cases} \Rightarrow \begin{array}{l} k_I = 224 \\ k_P = 710 \end{array}$$

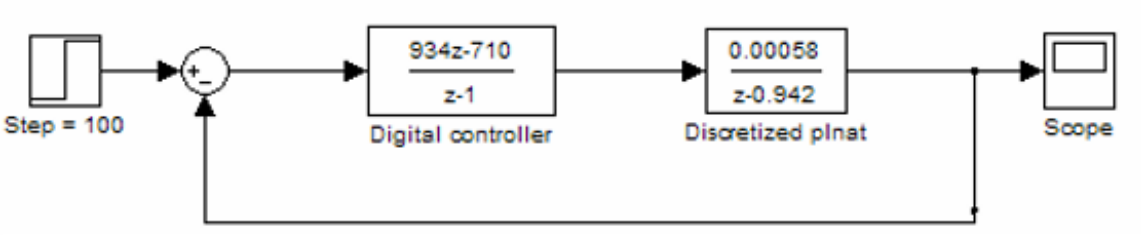
and a digital PI controller

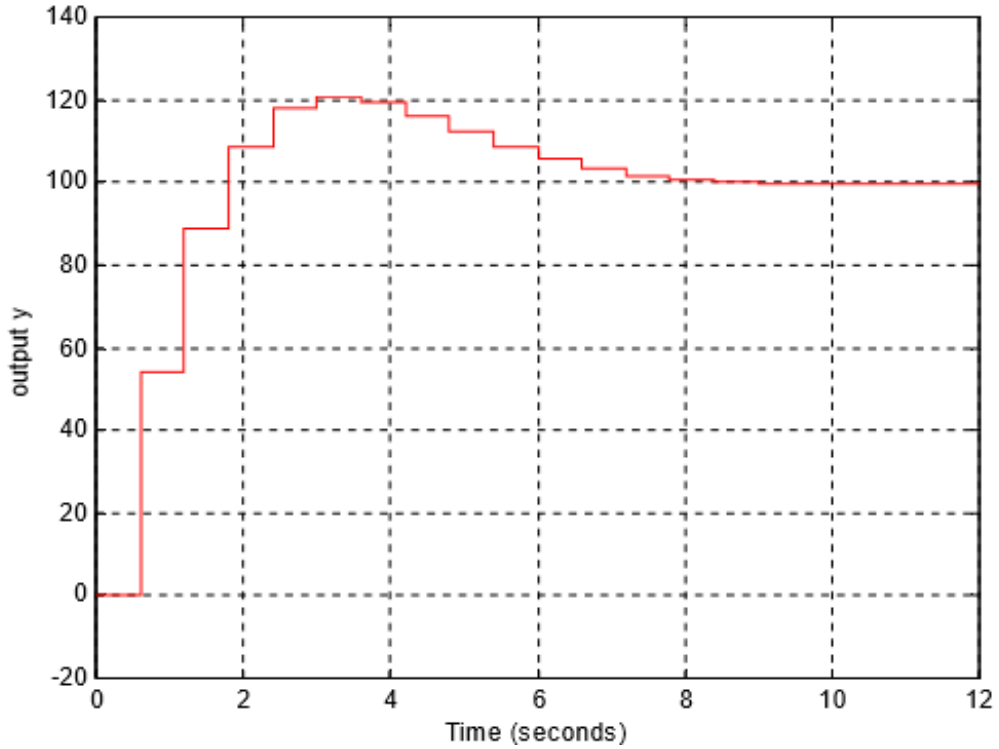
$$D(z) = k_P + k_I \frac{z}{z-1} = \frac{(k_P + k_I)z - k_P}{z-1} = \frac{934\,z - 710}{z-1}$$

Note: we cannot do much with the numerators of these transfer functions. It does affect the overall performance.

SIMULATION OF THE DIGITAL CONTROLLER WITH DISCRETIZED PLANT

We simulate the digital controller with the discretized plant to see whether the specifications are fulfilled in the discrete-time setting

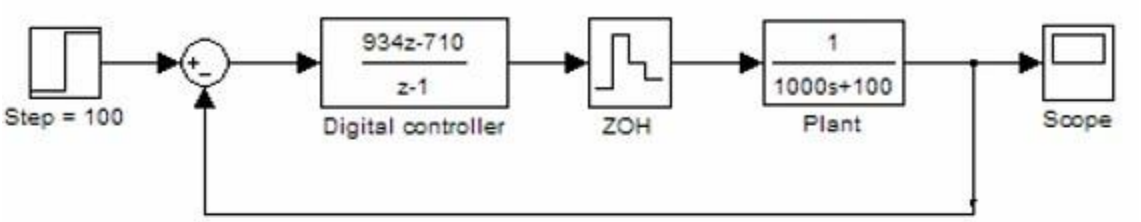


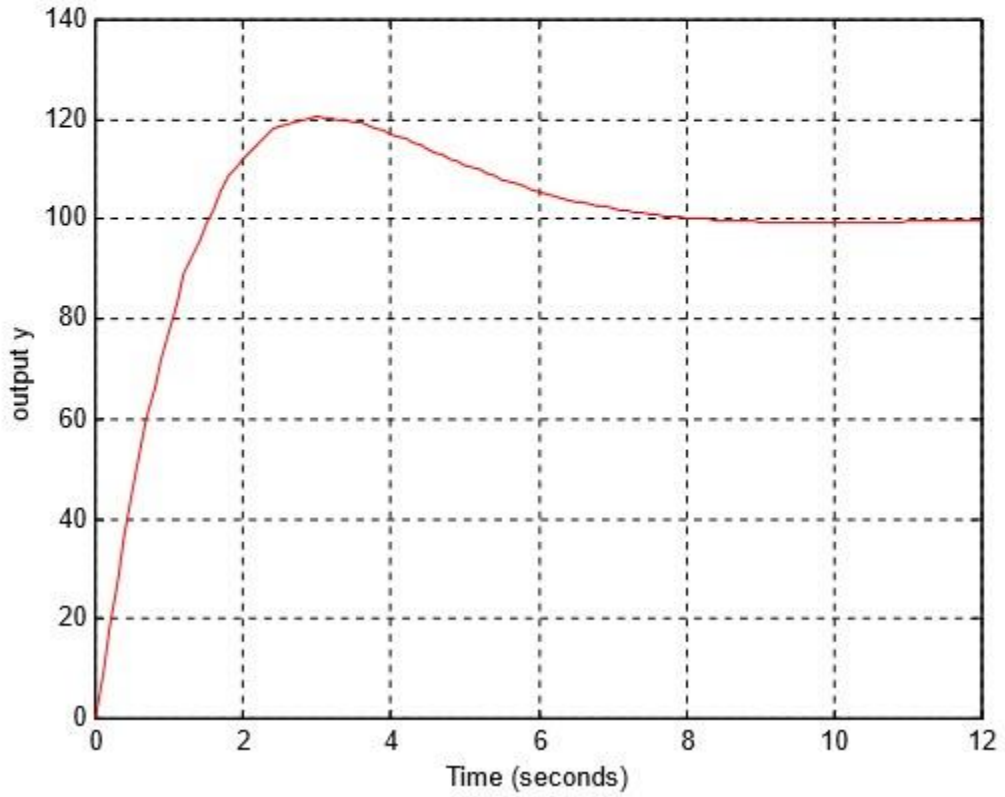


Remark: The overshoot is slightly larger than the design specification. But, the settling time meets the specification. The performance can be fine tuned by re-selecting the desired pole locations in z-plane.

SIMULATION OF THE DIGITAL CONTROLLER WITH ACTUAL PLANT

We simulate the digital controller with the actual plant.

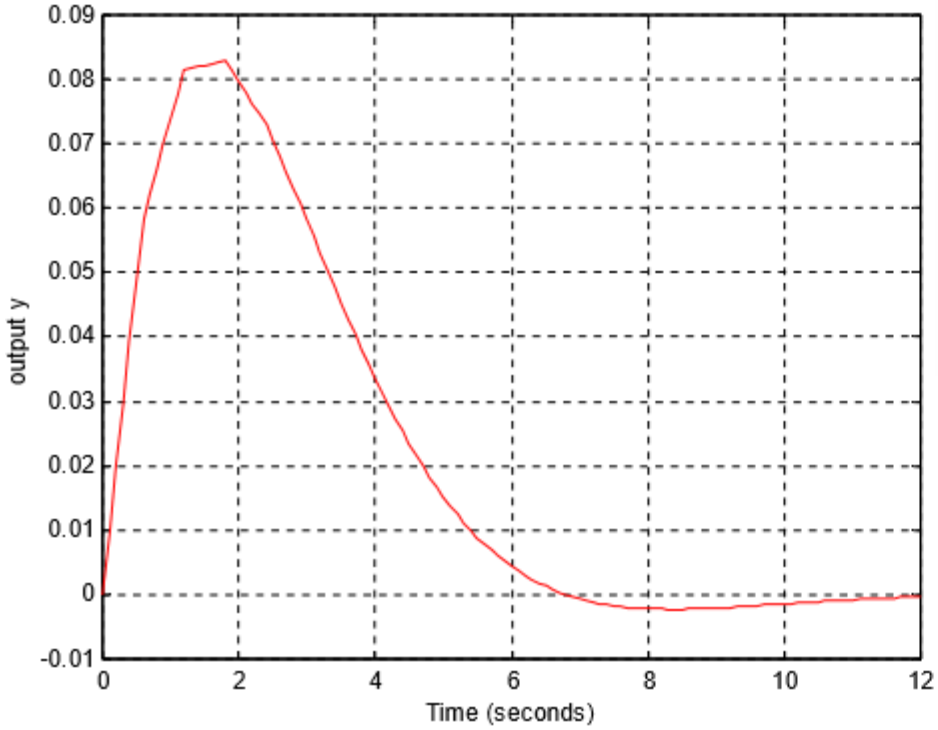


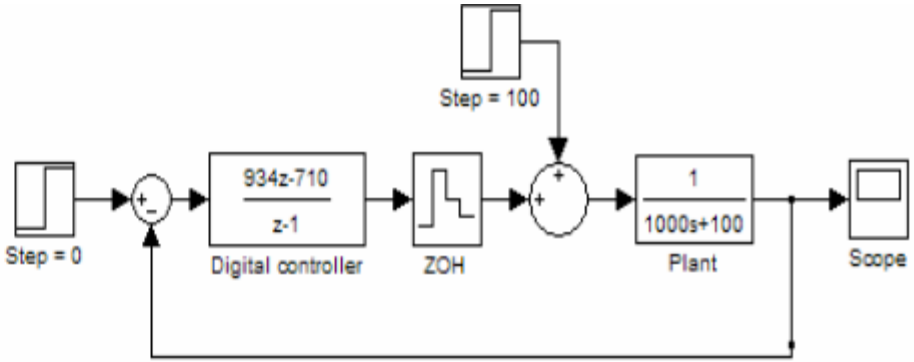


Remark: When the control law is implemented onto the actual continuous-time plant, the overshoot and the settling time are above the same as those obtained with the discretized system. All design specifications are met with a sampling period $T = 0.6$ seconds.

DISTURBANCE REJECTION

A step disturbance can be rejected when the controller has an integral action. Since we are using a PI controller, step disturbance should be rejected in our design although we haven't explicitly considered such a property in our design. We verify this through simulation by injecting a step function into the plant input. To see the effect of the disturbance on the system output, we let the reference to be zero.





A step disturbance with magnitude = 100 nicely is rejected from the output ! How about a ramp disturbance? D.I.Y.

HIGHER ORDER SYSTEMS

When the given plant has a dynamic order higher than 1 and/or a general PID controller is used, the overall closed-loop transfer function from rto ywill have an order larger than 2, e.g.,

$$H(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}, \qquad m \le n, \; n > 2$$

In this case, we should place the poles of the above transfer function by comparing it to the following desired transfer function

$$H_{\text{desired}}(z) = \frac{*}{(z-\alpha_1)\cdots(z-\alpha_{n-2})\cdot(z-z_p)(z-\bar{z}_p)}$$

i.e., by placing all the rest poles close to the origin, which is the fastest location in digital control. Eventually, dynamics associated with the poles close to the origin will die out very fast and the overall system is dominated by the pair left. This is left for students to practice in tutorial questions.

DEADBEAT CONTROLLER DESIGN

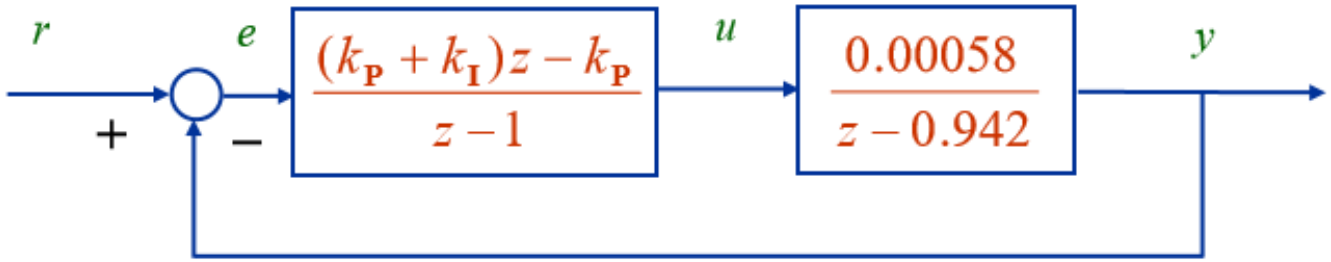
A unique feature of digital control is that we can design a control law such that the resulting system output is capable of following the reference input in a finite number of steps, i.e., in finite time interval, which can never be achieved in continuous-time setting. Such a control law is called deadbeat controller, which in fact places all the closed-loop system poles at the origin. Thus, the desired closed-loop transfer function under the deadbeat control is

$$H_{\text{desired}}(z) = \frac{1}{z^n}$$

The deadbeat control can only guarantee the system output to reach the target reference in n steps. The resulting overshoot could be huge and the control input could be very high (which is equivalent to that the energy required to achieve such a performance is large). As such, deadbeat control is generally impractical and rarely used in practical situations.

DESIGN EXAMPLE

We now design a deadbeat PI controller for the cruise control system. Recall



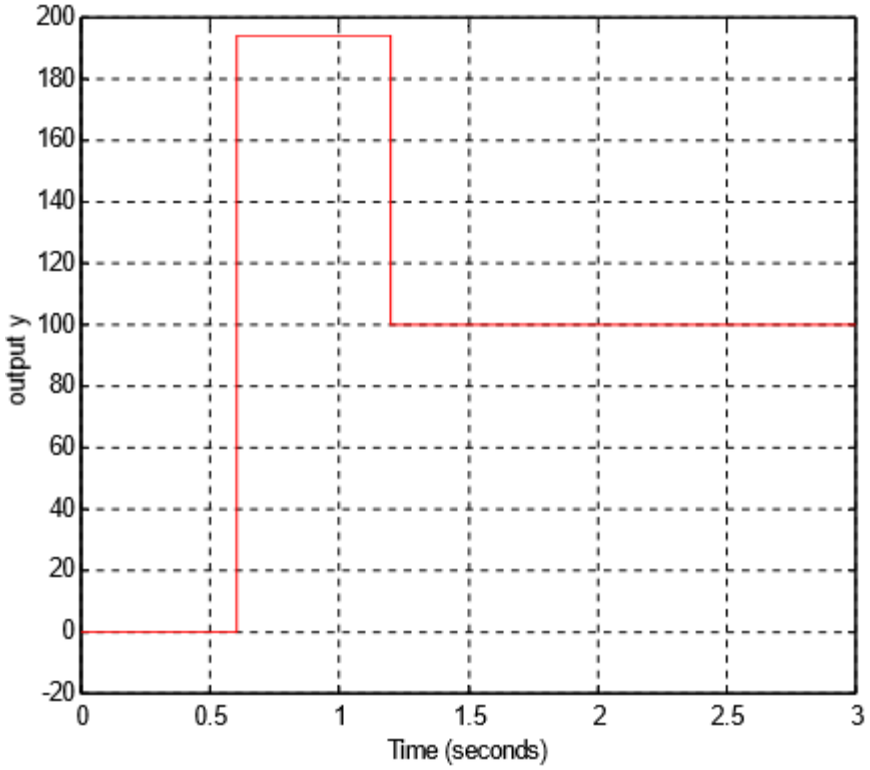
The resulting closed-loop transfer function from r to y is given by

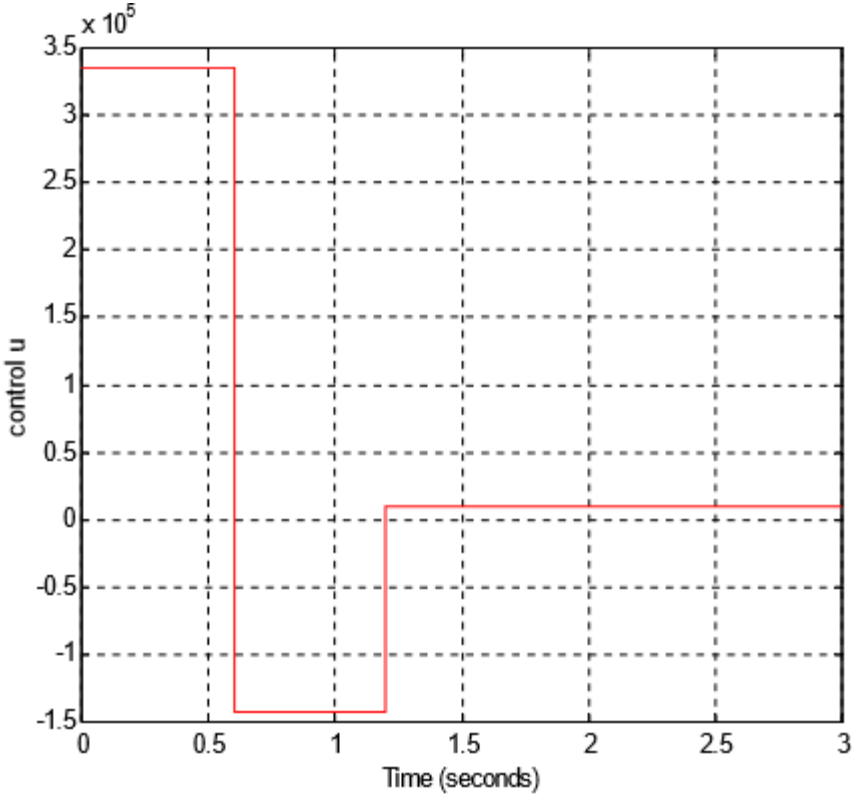
$$H(z) = \frac{0.00058\,(k_P + k_I)z - 0.00058\,k_P}{z^2 + [0.00058\,(k_P + k_I) - 1.942\,]z + (0.942 - 0.00058\,k_P)} \quad \succ \quad \frac{*}{z^2}$$

$$\Rightarrow \quad k_P = 1624.137931\,, \quad k_I = 1724.137931$$

$$\Rightarrow \quad D(z) = \frac{(k_P + k_I)z - k_P}{z-1} = \frac{3348.275862\,z - 1624.137931}{z-1}$$

SIMULATION RESULTS





The system out settles down to the target reference in 2steps, i.e., 2 x 0.6 = 1.2seconds (very fast). But, it has about 100% overshoot and huge control input. Such a controller cannot be implemented in real life, period ! However, there could be applications (such as military applications) that deadbeat control might be desirable.