CHEN4011
Advanced Modelling and COntrol

Curtin University

Dr. Ranjeet Utikar (RU)

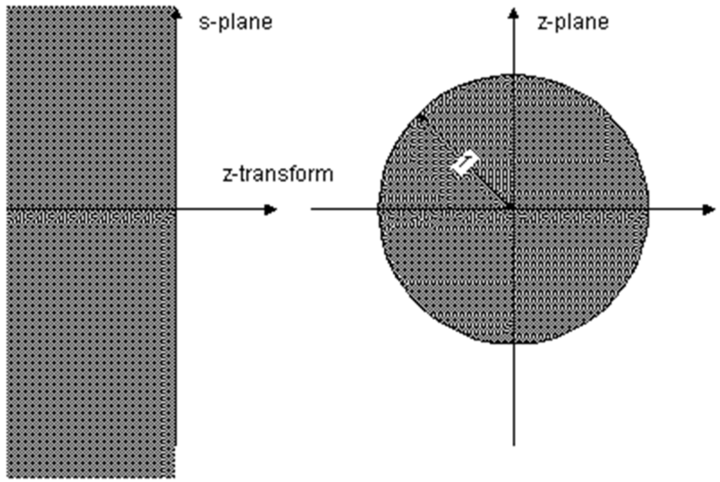
Curtin University
Malaysia

Dr. Jobrun Nandong (JN)

Introduction to Digital Control

Relationship z-plane with S - Plane

$$z = e^{sT}$$



- Processes are **<u>stable</u>** if they do not possess the poles that lies outside the **unit-circle in the z-plane.**

Origin of Z - Transform

$$f(nT_s) = \begin{cases} f(t) & \text{for } n = 0,1,2,\ldots \\ 0 & \text{otherwise} \end{cases} = f^*(t)$$

Recall the Laplace transform (LT) of **f(t):** $\mathbf{F(s) = \int_0^\infty e^{-st} f(t) dt}$

We can also apply LT to f*(t): $F^*(s) = \int_0^\infty e^{-st} f^*(t) dt$

Since f*(t) only exists at sampling instant: $F^*(s) = \sum_{n=0}^{\infty} f(nT) e^{-nTs}$

Defining: $z = e^{sT}$

$$\Rightarrow \mathbf{F^*(s) = \sum_{n=0}^{\infty} f(nT) e^{-nTs} = \sum_{n=0}^{\infty} f(nT) z^{-n} = F(z)}$$

Definition of Z- Transform

$$Z\{f(t)\} = F(z) = \sum_{n=0}^{\infty} f(nT)z^{-n}$$

- Z – transform is merely a Laplace transform for a **sampled data sequence**, as such inherits many of the properties of Laplace transform.
- Z – transform allows:
  - Development of **input-output models** for discrete-time system
  - Can be used to analyze how discrete-time processes react to **external input changes**.

Z - Transform

- Exists only if the **summation of infinite terms** takes **finite values**.
- Depends on sampling period T
- **Impossible to distinguish two functions**, which have the **same samples values** at the sampling instants.
- E.g. the values of a unit ***step function*** and ***cosine wave*** sampled at uniform intervals of period T are the same.

Same Sampled Value Function

$$\mathbf{Z}[\textbf{Unit Step}] = \frac{\mathbf{z}}{\mathbf{z}-1}$$

$$\mathbf{Z}[\cos\omega\mathbf{t}] = \frac{\mathbf{z}^2 - \mathbf{z}\cos\omega\mathbf{t}}{\mathbf{z}^2 - 2\mathbf{z}\cos\omega\mathbf{t} + 1}$$

If $\omega t = 2n\pi = nT$

$$\mathbf{Z}[\cos(\mathbf{nT})] = \frac{\mathbf{z}^2 - \mathbf{z}}{\mathbf{z}^2 - 2\mathbf{z} + 1} = \frac{\mathbf{z}}{\mathbf{z}-1}$$

Example - Z Transform

- Given the transform: $$\mathbf{F(s)} = \frac{1}{\mathbf{s(s+a)}} \qquad \mathbf{a} > 0$$

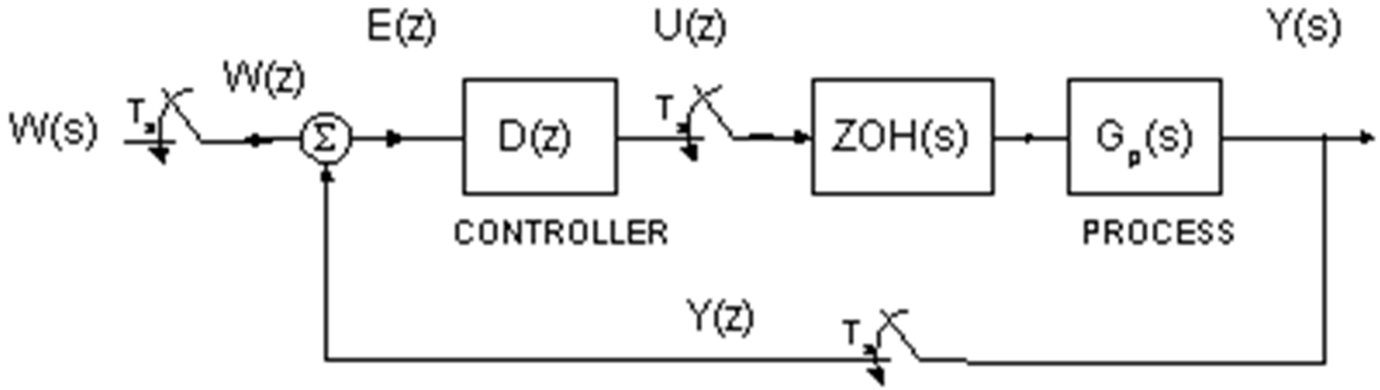
- Find: $$\lim_{\mathbf{n}\to\infty} \mathbf{F(nT)}$$

$$\mathbf{F(s)} = \frac{1}{\mathbf{a}}\left(\frac{1}{\mathbf{s}} - \frac{1}{\mathbf{s+a}}\right)$$

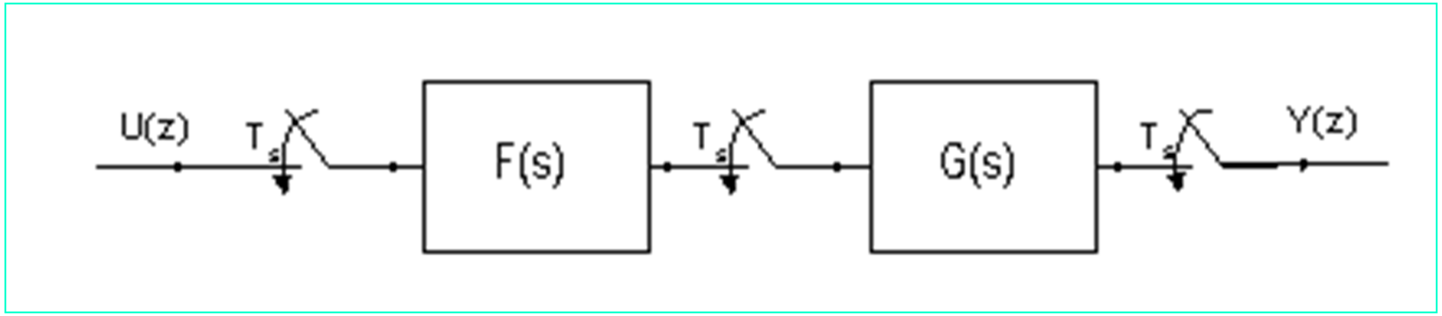
$$\therefore \mathbf{F(z)} = \frac{1}{\mathbf{a}}\left(\frac{1}{1-\mathbf{z}^{-1}} - \frac{1}{1-\mathbf{e}^{-\mathbf{aT}}\mathbf{z}^{-1}}\right)$$

Block Diagram Manipulation

- Manipulation of block diagrams of sampled data systems are very similar to that for those in the Laplace domain.
- **The z-transform is a special case of the Laplace transform**.
- The **presence of samplers**, there are some extra rules to follow
- ZOH denotes zero-order hold element



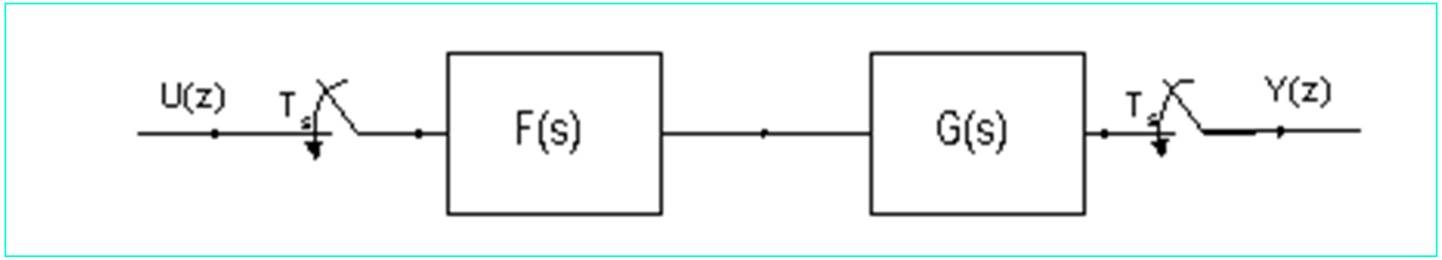
Block Diagram – System A



$$Y(z) = Z\{F(s)\}Z\{G(s)\}U(z)$$

$$\Rightarrow Y(z) = F(z)G(z)U(z)$$

Block Diagram – System B



$$Y(z) = Z\{F(s)G(s)\}U(z)$$

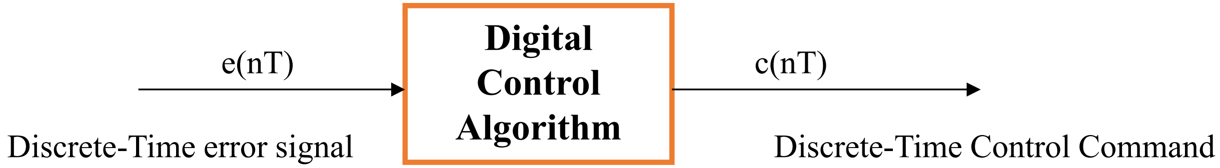
$$\Rightarrow Y(z) = FG(z)U(z)$$

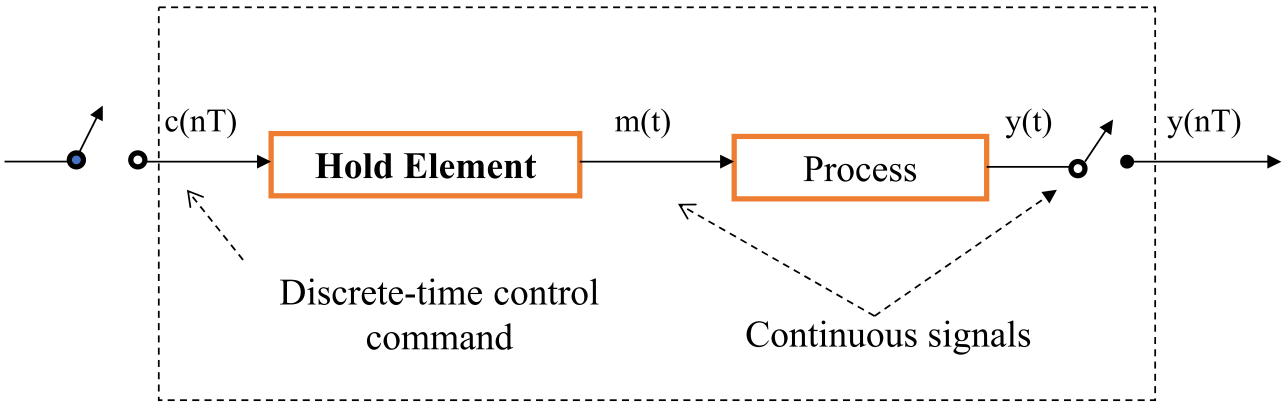
$$\textit{In general}: Z\{F(s)\}Z\{G(s)\} \neq Z\{F(s)G(s)\}$$

Discrete-Time Response

- **Two primary distinct** components in dynamic systems whose responses should be considered:
- **Digital control algorithm,** i.e. discrete element with discrete-time input and output signals
- **Process with the hold elements,** i.e. continuous element of the Direct Digital Control (DDC) Loop

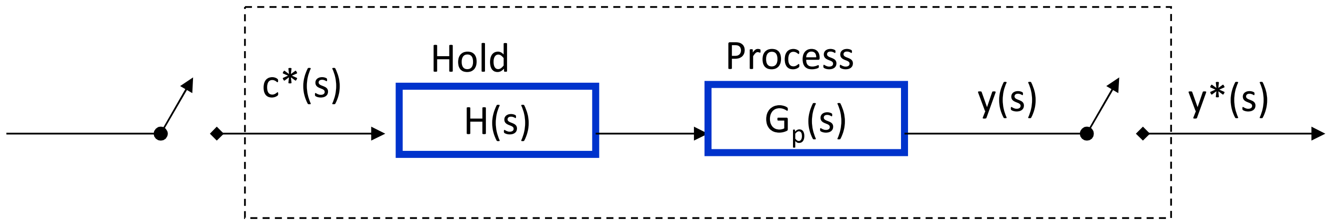
Components DDC Loop with Discrete – Time I/O

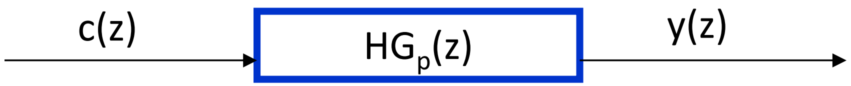




Discrete-Time Analysis

- Pulse transfer function relates y(nT) to c(nT) in z-domain.





$$\frac{Y(z)}{C(z)} = HG_p(z) = Z\left[H(s)G_p(s)\right]$$

MATLAB functions – Conversion of Continuous System to Discrete

- MATLAB built-in function to convert a continuous transfer function $G_p(s)$ to discrete system $G_p(z)$ => cdc
- Syntax:
- **SYSD = c2d(SYSC,TS,METHOD)** computes a discrete-time model SYSD with
- **sample time TS** that approximates the continuous-time model SYSC.
- The string METHOD selects the discretization method among the following:
- **'zoh'** Zero-order hold on the inputs
- **'foh'** Linear interpolation of inputs
- **'impulse'** Impulse-invariant discretization
- **'tustin'** Bilinear (Tustin) approximation.
- **'matched'** Matched pole-zero method (for SISO systems only).
- **'least-squares'** Least-squares minimization of the error between
- frequency responses of the continuous and discrete
- systems (for SISO systems only).

Example 1 – Continuous System to Discrete System Conversion

- Consider a continuous process:

$$G_p(s) = \frac{2(1-s)}{(3s+1)(s+1)}$$

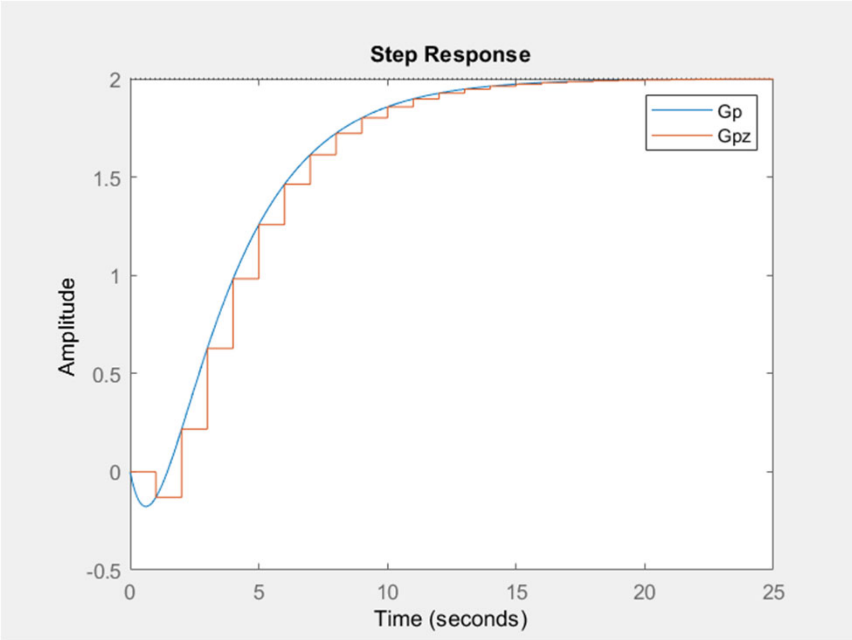
- Use Zero-Order Hold and sampling period T = 1
- Type on MATLAB Command Window:

```
>> s = tf ('s');
>> Gp = 2*(1-s)/((3*s+1)*(s+1));
>> Gpz = c2d(Gp,1,'zoh')
Gpz =
    -0.1304 z + 0.4887
  ----------------------
  z^2 - 1.084 z + 0.2636
Sample time: 1 seconds
Discrete-time transfer function.
```

Example 1 cont..

- Compare the responses of Gp and Gpz to 1 unit step change.
- Use the ‘step’ function to generate the plots

    >> step(Gp,Gpz)



Example 1 – Other hold elements

First-Order Hold Element

```
>> Gpz1 = c2d(Gp,1,'foh')

Gpz1 =

  -0.1374 z^2 + 0.3141 z + 0.1817
  -------------------------------
     z^2 - 1.084 z + 0.2636

Sample time: 1 seconds
Discrete-time transfer function.
```

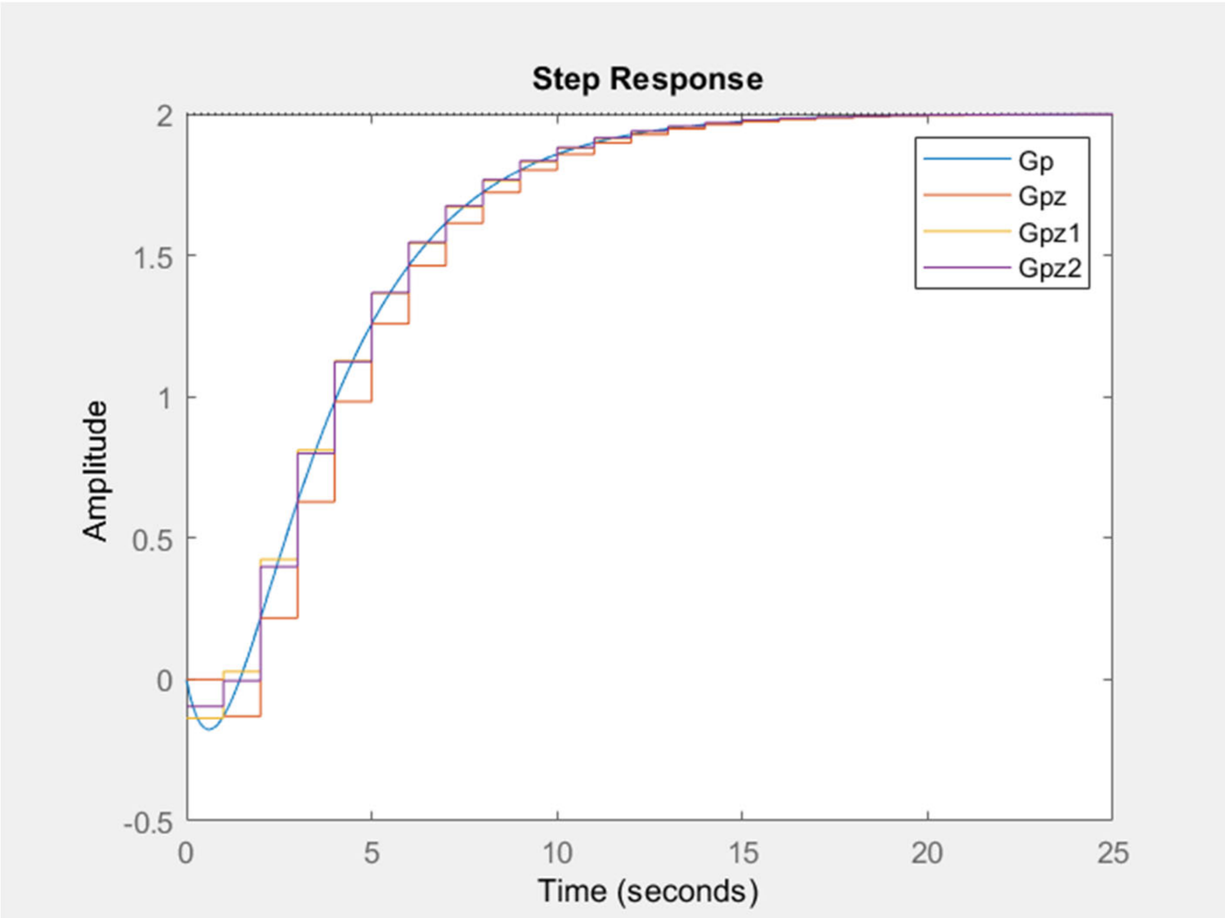
Tustin approximation

```
>> Gpz2 = c2d(Gp,1,'tustin')

Gpz2 =

  -0.09524 z^2 + 0.1905 z + 0.2857
  --------------------------------
     z^2 - 1.048 z + 0.2381

Sample time: 1 seconds
Discrete-time transfer function.
```

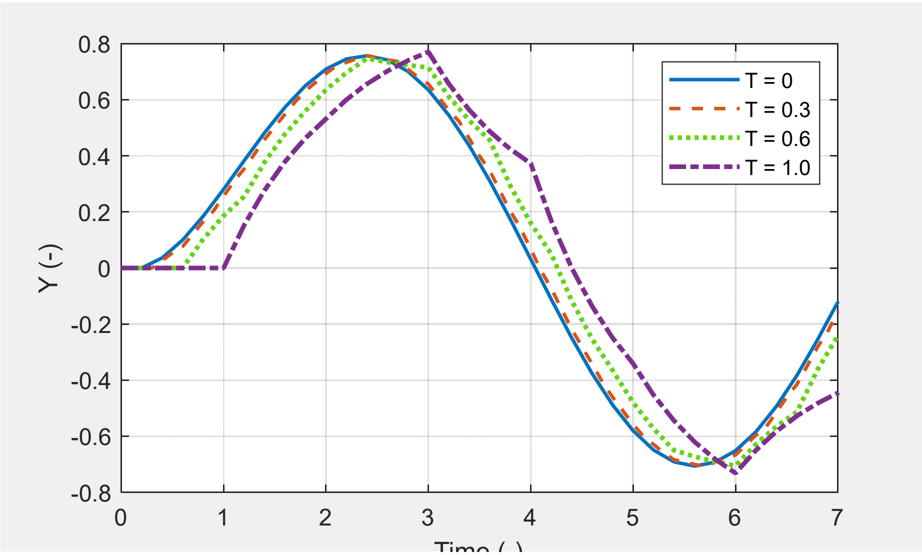


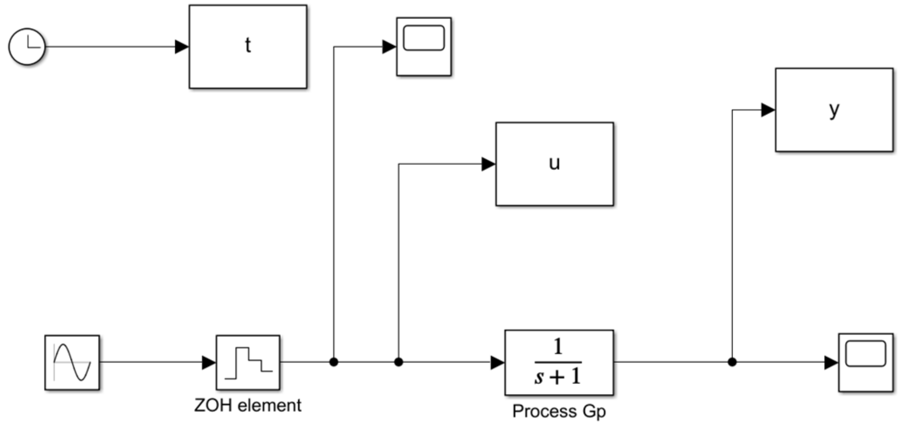
- Comparisons of responses of discrete systems of a continuous system using different approximations:

1. Zero-Order Hold (Gpz)
2. First-Order Hold (Gpz1)
3. Tustin (Gpz2)

- Signal reconstruction using First-Order Hold and Tustin approximation are more accurate than Zero-Order Hold

Effect of Sampling Time (T) on Signal Reconstruction





- Larger sampling period T leads to more loss in signal information
- Signal reconstruction is deformed more as T increases
- Signal information loss due to large sampling period is called ***"Signal aliasing"***

Position Form of PID Controller

- At each sampling time the actual value (position) of the output signal is calculated.
- PI saves:
  - Current error e_n
  - Sum of all previous errors

$$S_{n-1} = \sum_{i=1}^{n-1} \varepsilon_i$$

$$c_n = K_c \left[\varepsilon_n + \frac{T}{\tau_I}(S_{n-1} + \varepsilon_n) \right] + c_s$$

- PID saves previous error as well:

$$c_n = K_c \left[\varepsilon_n + \frac{T}{\tau_I}(S_{n-1} + \varepsilon_n) + \frac{\tau_D}{T}(\varepsilon_n - \varepsilon_{n-1}) \right] + c_s$$

Velocity Form of PID Controller

- At each sampling time the change from the preceding period of the output signal is calculated.

- At n^{th} and $(n-1)^{th}$ sampling periods

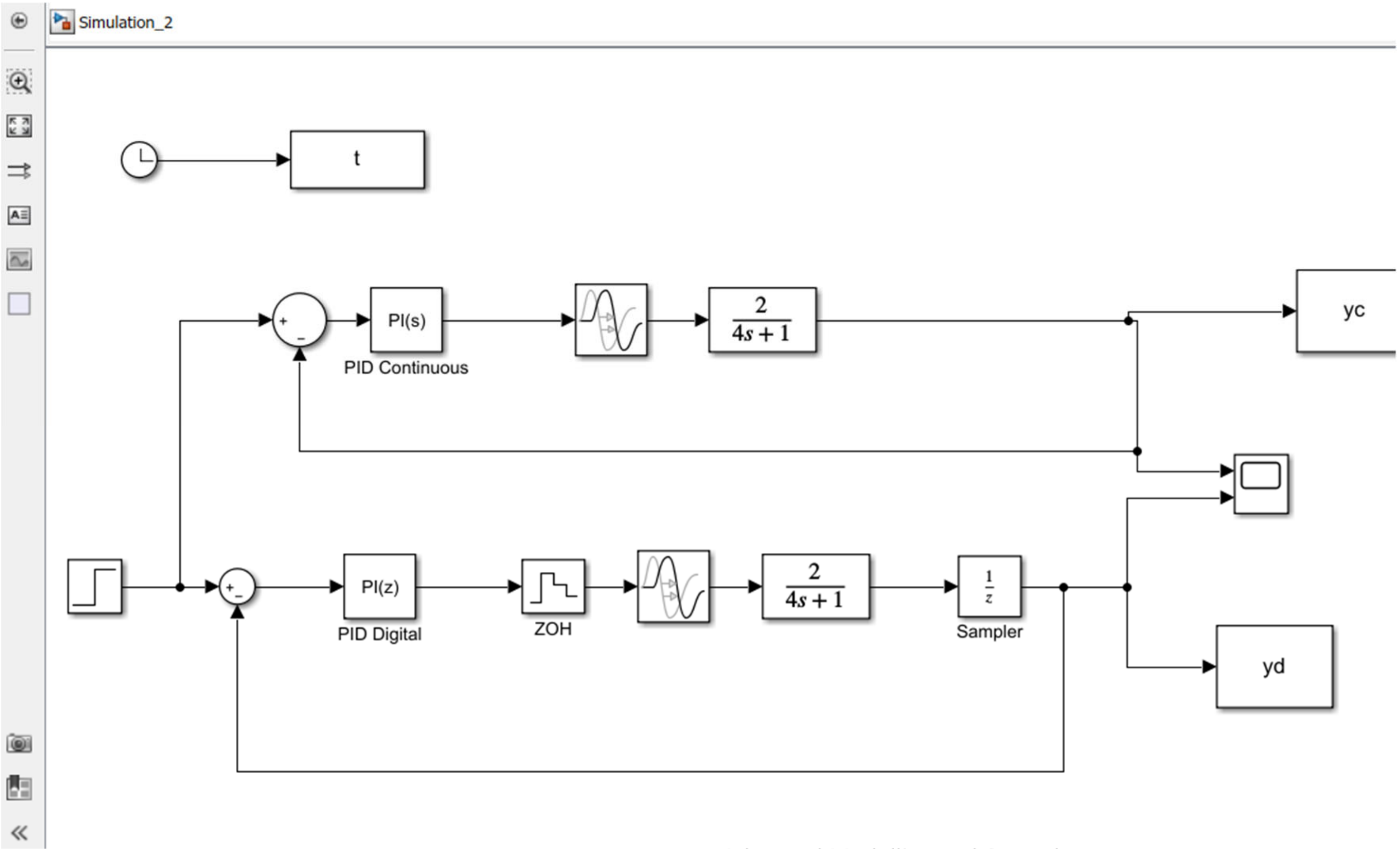
$$c_n = K_c\left[\varepsilon_n + \frac{T}{\tau_I}\sum_{k=0}^{n}\varepsilon_k + \frac{\tau_D}{T}\left(\varepsilon_n - \varepsilon_{n-1}\right)\right] + c_s$$

$$c_{n-1} = K_c\left[\varepsilon_n + \frac{T}{\tau_I}\sum_{k=0}^{n-1}\varepsilon_k + \frac{\tau_D}{T}(\varepsilon_{n-1} - \varepsilon_{n-2}\right] + c_s$$

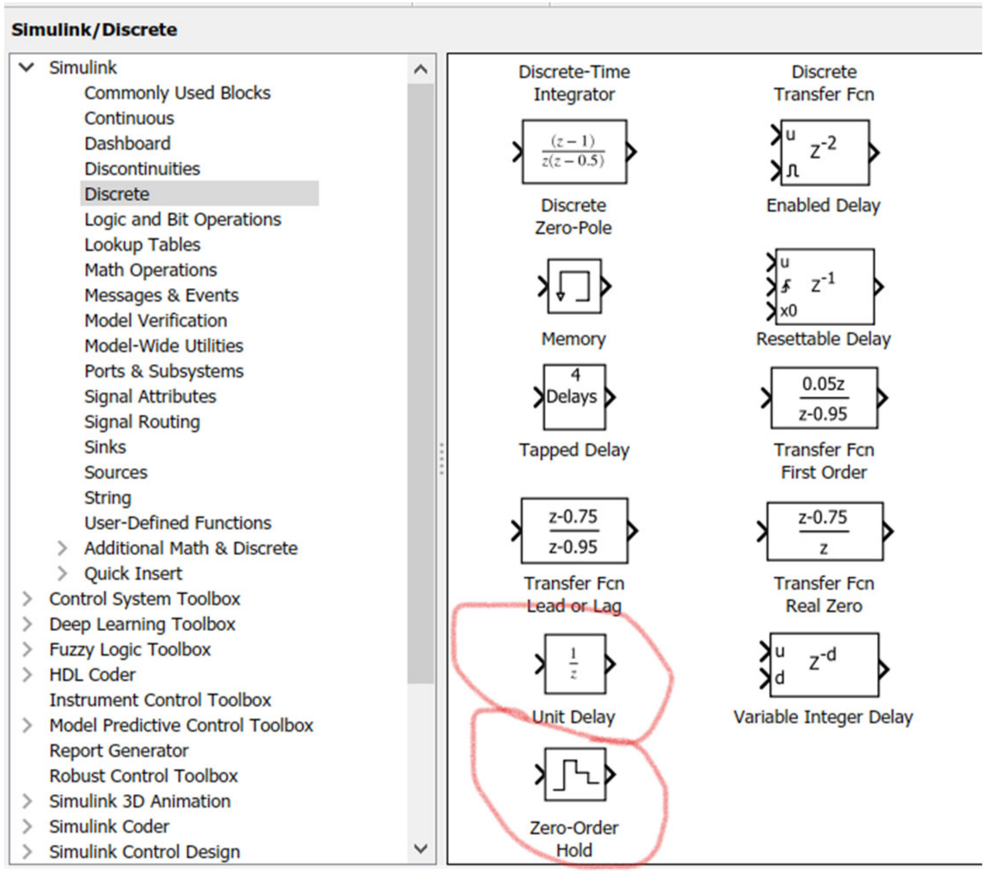
- Subtraction:

$$\Delta c_n = c_n - c_{n-1} = K_c\left(1 + \frac{T}{\tau_I} + \frac{\tau_D}{T}\right)\varepsilon_n - K_c\left(1 + \frac{2\tau_D}{T}\right)\varepsilon_{n-1} + K_c\frac{\tau_D}{T}\varepsilon_{n-2}$$

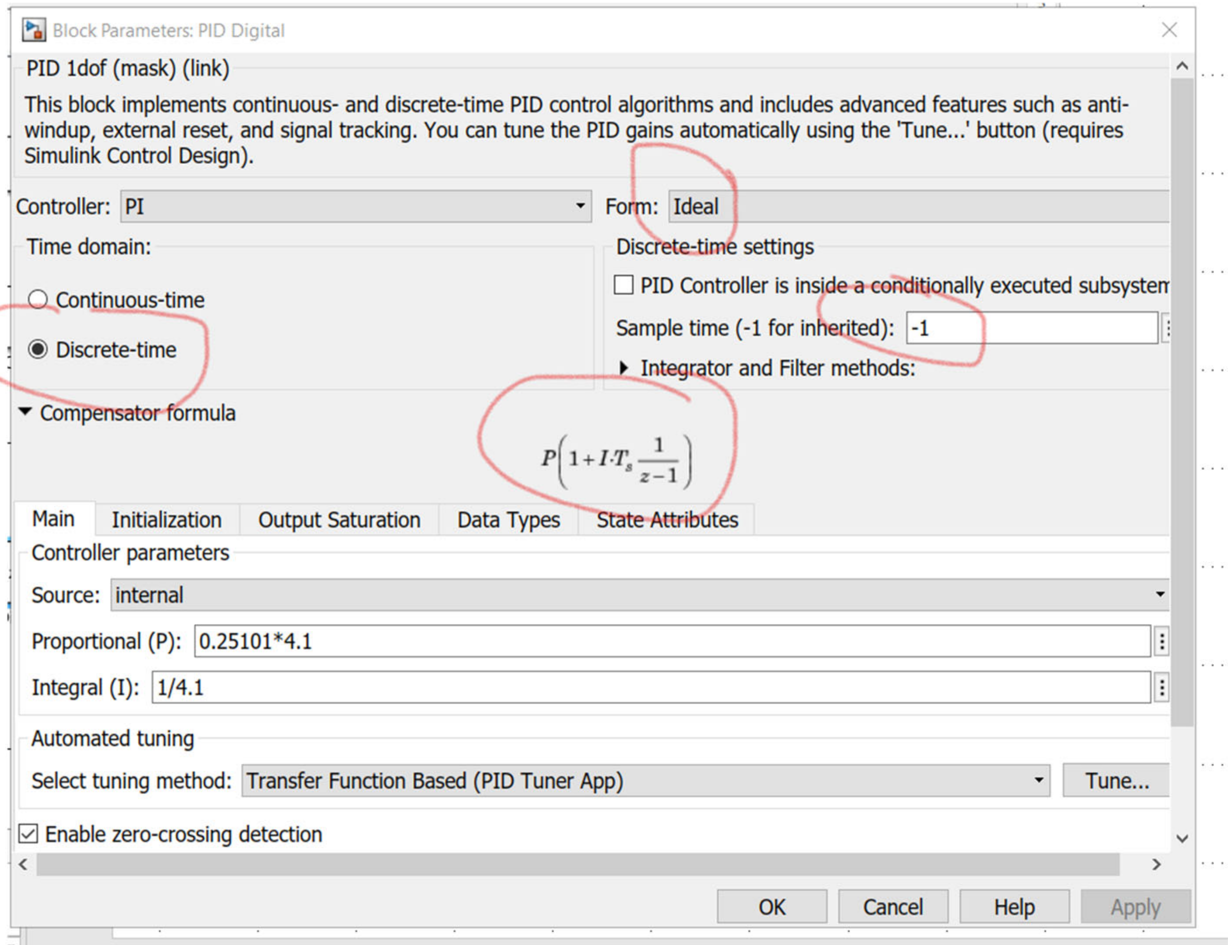
MATLAB Simulink - Digital PID Controller



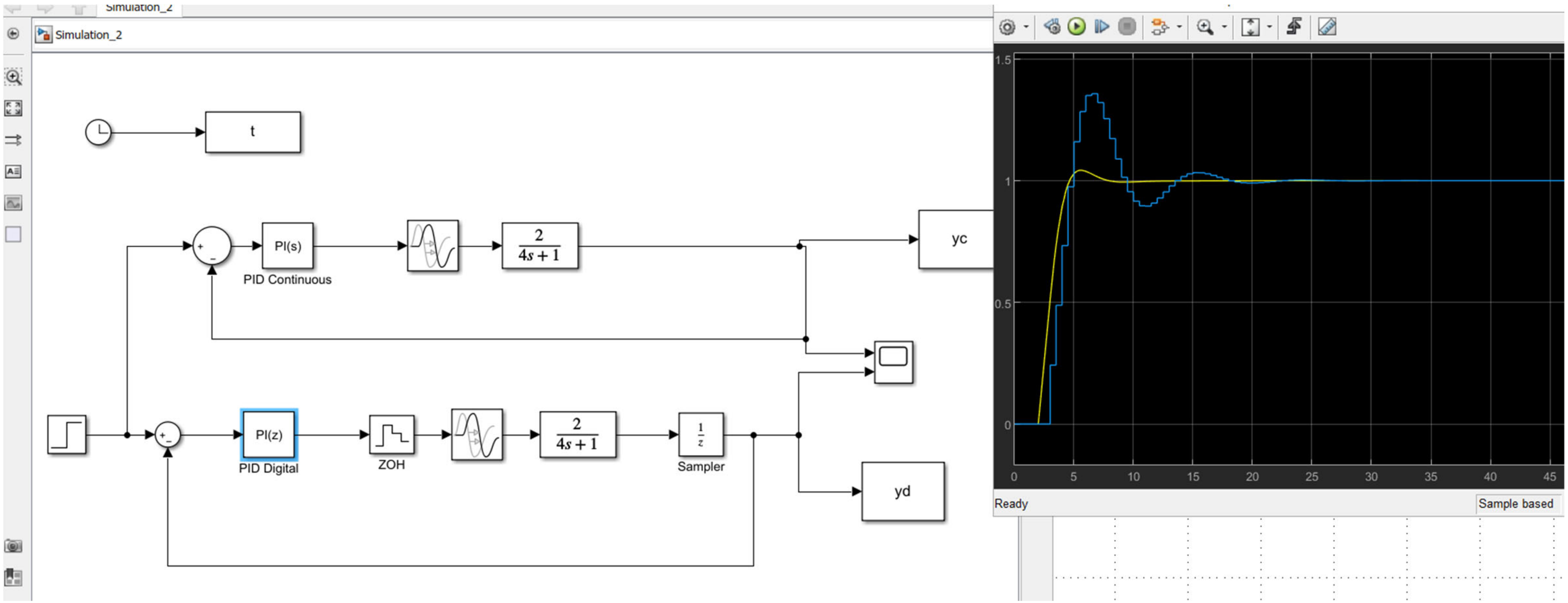
Simulink Blocks



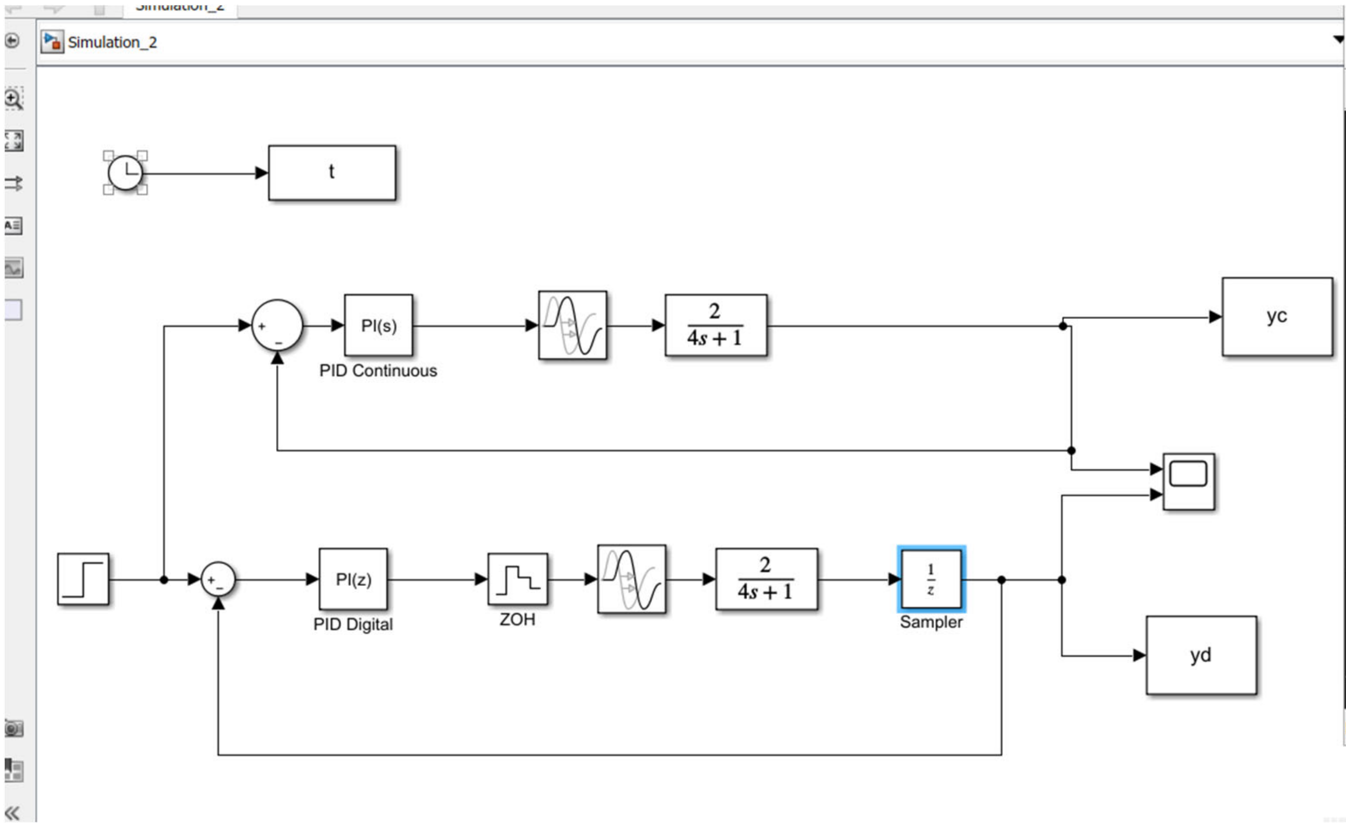
Setting PID Controller – Continuous to Digital Form

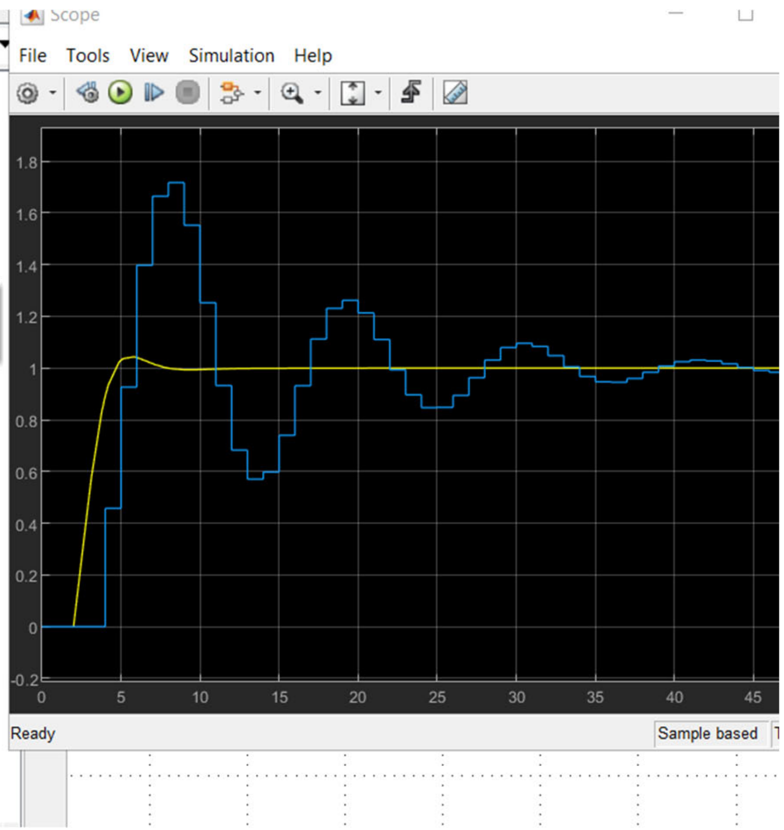


Example 2 - Continuous vs. Digital PI Controller T = 0.5 s

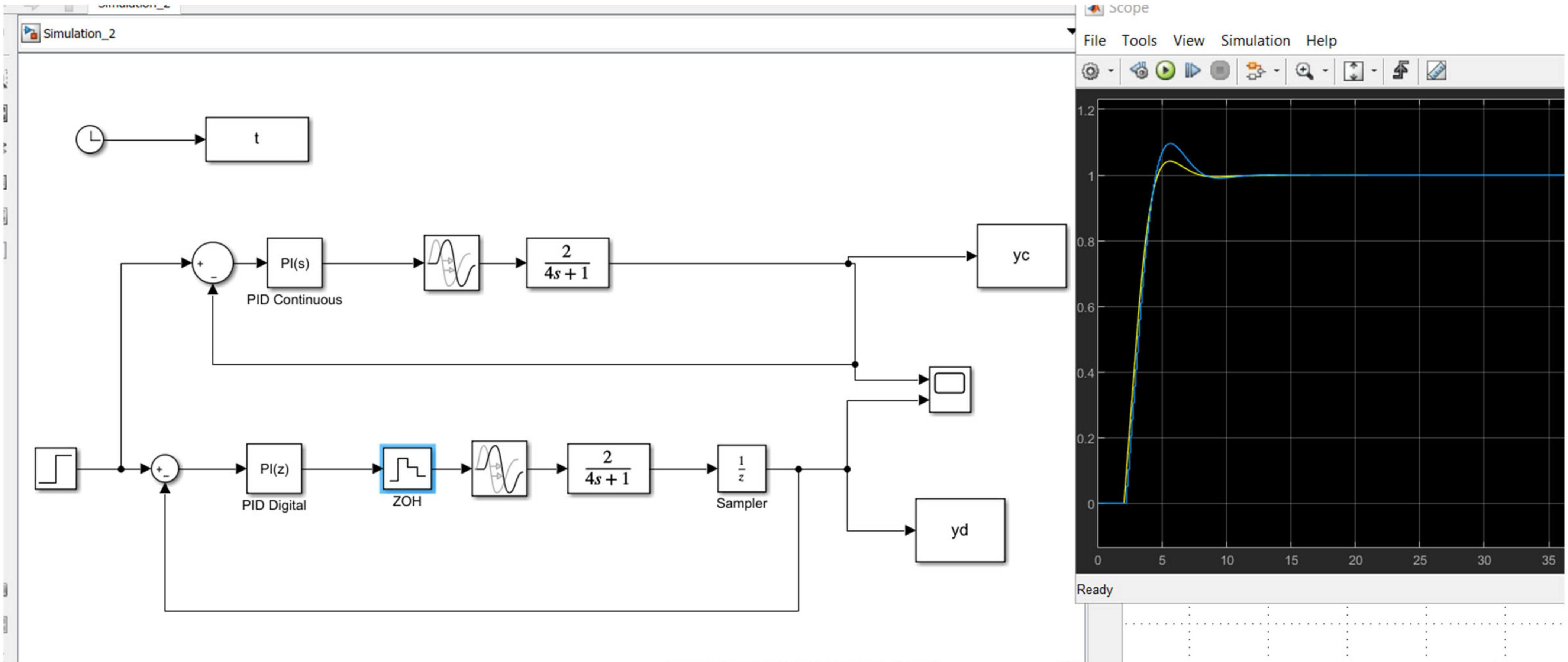


Continuous vs. Digital PI Controller
T = 1 s





Continuous vs. Digital PI Controller
T = 0.1 s



Effect of Sampling Period on Digital PI Controller Performance

- Long sampling period leads to slower response compared to the continuous PI controller
- Smaller sampling period leads to faster response and approaching that of the continuous PI controller as the sampling period decreases
- In practice, there is a trade-off between short sampling period and storage/computational capacity
- Short sampling period requires more computational and storage capacity but gives good control performance
- Long sampling period requires less storage and computational capacity but lower control performance