

# Discretization of Continuous-time Systems

July 17, 2015

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Discretization

## Use of discretization

Many systems in the real world are continuous systems: chemical reactions, rocket trajectories, power plants, ice cap melting... Computers, however, are mainly digital. If we want to **simulate** the continuous system with a logic device, we need a method to convert the continuous model into a discrete one. This conversion is called "discretization" or "sampling". Discretization also comes in handy when a continuous **filter with useful properties** has been designed and a discrete filter with the same properties is required. Next to these application of discretization, discretization is also used in **control**. To design a digital controller, it is necessary that we have a discrete model of the plant the controller is used for.

# Discretization

## Problem statement

While converting, some information of the continuous model may be lost due to the different nature of the systems. It is important that the loss of information is minimized. Each discretization method has its own qualities and they will all lead to different discrete representations of the same continuous system.

## Discretization methods discussed in this lecture

- Numerical Integration
- Impulse Invariant Method
- Zero-pole Equivalent
- Hold Equivalents

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Numerical Integration

## Relation between a continuous and a discrete system

$$\dot{x} = Ax$$
$$x(t) = e^{At}x(0)$$

The discretization of a system requires a sampling time:  $T_s = \frac{1}{f_s}$

$$x(t + T_s) = e^{AT_s}e^{At}x(0) = e^{AT_s}x(t)$$
$$x(t) = e^{AT_s}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

The integral can be solved using numerical integration.

# Numerical Integration

## General approach

The system transfer function  $H(s)$  is first represented by a differential equation. Next a difference equation, whose solution is an approximation of this differential equation, is derived, e.g.:

$$H(s) = \frac{U(s)}{E(s)} = \frac{a}{s+a}$$

is equivalent to the differential equation:

$$\frac{d}{dt}(u(t)) + au(t) = ae(t)$$

solving this equation results in the following integral:

$$u(t) = \int_0^t (-au(\tau) + ae(\tau))d\tau$$



# Numerical Integration

## General approach

After sampling, this leads to:

$$u(kT) = \int_0^{kT-T} (-au + ae) d\tau + \int_{kT-T}^{kT} (-au + ae) d\tau$$

$$u(kT) = u(kT - T) + \begin{cases} \text{area of } -au(t) + ae(t) \\ \text{over } kT - T \leq \tau < kT \end{cases} \quad (8.1)$$

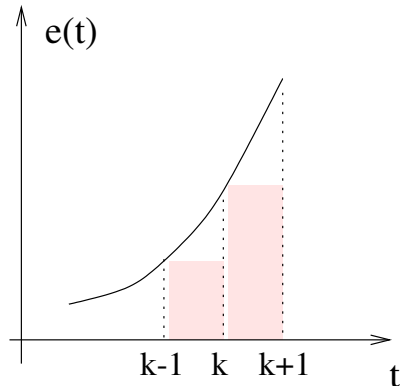
where  $T$  is the sampling time.

The transfer function  $H(s) = \frac{a}{s+a}$  will be used to explain all numerical integration methods.

## Forward rectangular rule (= (Forward) Euler)

### General approach

The area is approximated by the rectangle looking **forward** from  $(k-1)$  toward  $k$  with an amplitude equal to the value of the function at  $(k-1)$ .



# Forward rectangular rule

## Mathematical approach

$$e_k = \frac{x_{k+1} - x_k}{T_s}$$
$$E(z) = \frac{z-1}{T_s} X(z)$$

Which means that we need to apply the following substitution in the Laplace transform of the given transfer function:

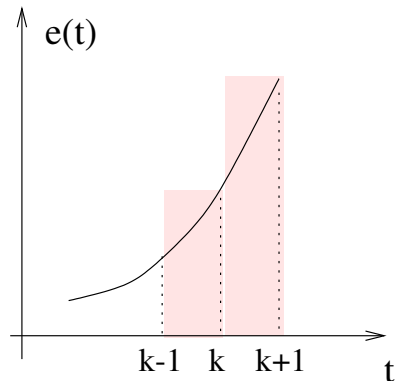
$$s \leftarrow \frac{z-1}{T} \quad (8.2)$$

This is extremely useful while making exercises.

# Backward rectangular rule (=Backward Euler)

## General approach

The area is approximated by the rectangle looking **backward** from  $k$  toward  $(k - 1)$  with an amplitude equal to the value of the function at  $k$ .



# Backward rectangular rule

## Mathematical approach

$$e_{k+1} = \frac{x_{k+1} - x_k}{T_s}$$

$$E(z) = \frac{z-1}{zT_s} X(z)$$

Which means that we need to apply the following substitution in the Laplace transform of the given transfer function:

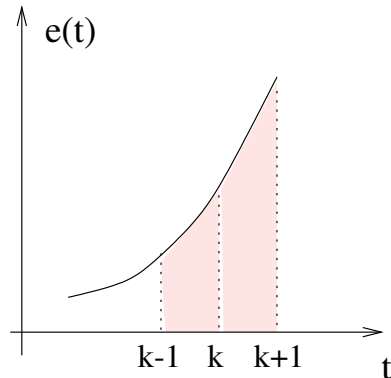
$$s \leftarrow \frac{z-1}{zT_s} \quad (8.3)$$

This is extremely useful while making exercises.

# Bilinear rule (= trapezoidal or Tusting rule)

## General approach

This method makes use of the area of the **trapezoid** formed by the average of the selected rectangles used in the forward and backward rectangular rule. Thus the amplitude equal to the value of the function at  $(k-1)$  and the amplitude equal to the value of the function at  $(k)$  are connected by a line as shown in the illustration.



# Bilinear rule

## Mathematical approach

$$\frac{e_{k+1} + e_k}{2} = \frac{x_{k+1} - x_k}{T_s}$$
$$E(z) = \frac{2}{T_s} \frac{z-1}{z+1} X(z)$$

Which means that we need to apply the following substitution in the Laplace transform of the given transfer function:

$$s \leftarrow \frac{2}{T_s} \frac{z-1}{z+1} \quad (8.4)$$

This is extremely useful while making exercises.

## Bilinear rule

### Example

Given:

$$H(s) = \frac{s+1}{0.1s+1}$$

We now apply substitution (8.4):

$$H(z) = \frac{(2+T)(T-2)z^{-1}}{(0.2+T)+(T-0.2)z^{-1}}$$

Using  $T=0.25s$ , this results in:

$$H(z) = \frac{5(z-0.7778)}{z+0.1111}$$



# Reconstruction

## Reconstruction of the continuous system

To reconstruct the continuous time system, given the discrete model, you need to solve the substitution formulas ((8.2) - (8.4)) to  $z$ :

- $z = 1 + T_s s$

- $z = \frac{1}{1 - T_s s}$

- $z = \frac{1 + T_s \frac{s}{2}}{1 - T_s \frac{s}{2}}$

Next you need to replace  $z$  (in the discrete transfer function) with these expressions.

# Steady State model of numerical methods

	Euler	backward rect.	bilinear transf.
$A_d$	$I + AT_s$	$(I - AT_s)^{-1}$	$(I - \frac{AT_s}{2})^{-1}(I + \frac{AT_s}{2})$
$B_d$	$BT_s$	$(I - AT_s)^{-1}BT_s$	$(I - \frac{AT_s}{2})^{-1}BT_s$
$C_d$	$C$	$C(I - AT_s)^{-1}$	$C(I - \frac{AT_s}{2})^{-1}$
$D_d$	$D$	$D + C(I - AT_s)^{-1}BT_s$	$D + C(I - \frac{AT_s}{2})^{-1}\frac{BT_s}{2}$

# Stability of the numerical integration methods

## Stability

As already mentioned, a discrete system is stable when its poles lie within the unit circle of the z-plane and a continuous system is stable when its poles have a negative real part in the s-plane. Subsequently the ( $s = j\omega$ )-axis is the boundary between poles of stable and unstable continuous systems.

Each of the discretization methods can be considered as a map from the s-plane to the z-plane. It is interesting to know how the  $j\omega$ -axis is mapped by every rule and where the stable part of the s-plane appears in the z-plane. This can be realized by solving formulas (8.2-8.3) to z and replacing s by  $j\omega$ .

# Stability of the numerical integration methods

## Boundaries of the stable regions

Expressions of  $z$  in terms of  $s$ :

- $z = 1 + T_s s$
- $z = \frac{1}{1 - T_s s}$
- $z = \frac{1 + T_s \frac{s}{2}}{1 - T_s \frac{s}{2}}$

By substituting  $s$  by  $j\omega$  the boundaries of the regions in the  $z$ -plane, which originate from the stable portion of the  $s$ -plane, are obtained.

# Stability of the numerical integration methods

## Mapping of the left-hand-s-plane

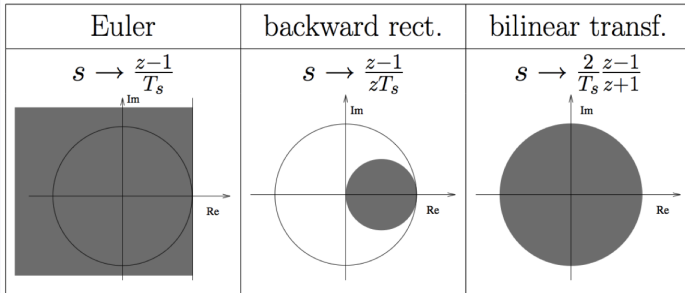
- Forward Euler: a stable continuous-time system may become unstable after discretization;
- Backward Euler: a stable continuous-time system will stay stable after discretization, but the number of degrees of freedom is restricted;
- Bilinear transformation: the entire left-hand-plane is mapped into the unit circle.

# Stability of the numerical integration methods

## Graphical representation

Stable s-plane poles map onto the shaded regions in the z-plane.  
The unit circle is shown for reference.

■ = projection of the left half-plane



## Bilinear rule with prewarping

### Distortion

Looking at the graphical representation of the stability on the previous slide, you can see that the bilinear rule maps the stable region of the s-plane into the stable region of the z-plane. The entire  $j\omega$ -axis is compressed into the  $2\pi$ -length of the unit circle, causing a great deal of distortion.

### Distortion

Distortion is the alternation of the original shape (or other characteristic) of something, such as an object, image, sound or waveform. In some cases distortion is desirable but mostly it needs to be minimized or (when possible) completely eliminated.

# Bilinear rule with prewarping

## Origin of prewarping

NOG IN TE VOEGEN

## Behaviour of the function

The same behaviour of the function is only guaranteed at low frequencies and at the prewarpfrequency you specify.



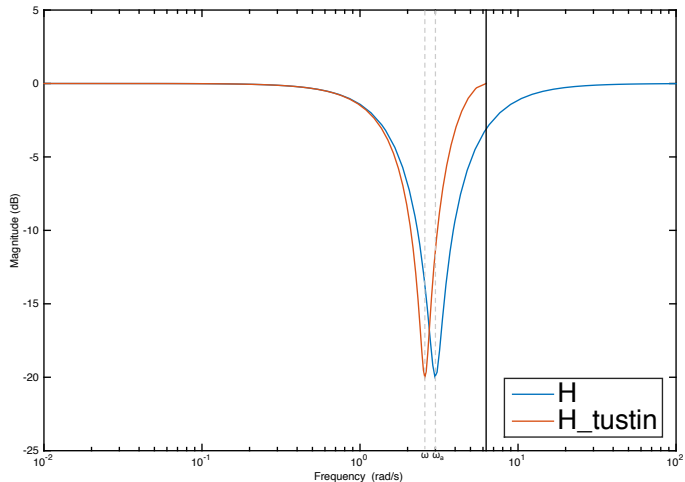
## Bilinear rule with prewarping

### Distortion: frequency domain

The bilinear rule causes every feature, every "bump" that is visible in the frequency response of the continuous-time filter to also be visible in the discrete-time filter, but at a different frequency. This is illustrated by the figure on the next slide.

When the actual frequency of  $\omega$  is input to the discrete-time filter designed by use of the bilinear transform, it is desired to know at what frequency,  $\omega_a$ , for the continuous-time filter that this  $\omega$  is mapped to.

# Bilinear rule with prewarping



## Bilinear rule with prewarping

$\omega_a$  with input  $\omega$

We now look for a relation between  $\omega_a$  and  $\omega$ , using the 's-domain'-  
'z-domain' relation  $z = e^{sT}$  and the substitution  $s = j\omega$ :

$$\begin{aligned} \frac{2}{T} \frac{z-1}{z+1} \Big|_{z=e^{j\omega T}} &= \frac{2}{T} \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} \\ &= \frac{2}{T} \frac{e^{j\omega T/2}(e^{j\omega T/2} - e^{-j\omega T/2})}{e^{j\omega T/2}(e^{j\omega T/2} + e^{-j\omega T/2})} \\ &= j \frac{2}{T} \frac{\sin(\omega T/2)}{\cos(\omega T/2)} \\ &= j \frac{2}{T} \tan\left(\frac{\omega T}{2}\right) = j\omega_a \end{aligned}$$

## Bilinear rule with prewarping

### Frequency warping

The discrete-time system has the same behavior at frequency  $\omega$  as the continuous-time system at frequency  $\omega_a = \frac{2}{T} \tan\left(\frac{\omega T}{2}\right)$ .

Specifically, the gain and phase shift that the discrete-time filter has at frequency  $\omega$  is the same gain and phase shift that the continuous-time filter has at frequency  $\omega_a = \frac{2}{T} \tan\left(\frac{\omega T}{2}\right)$ .

This effect of the non-linear relation between  $\omega$  and  $\omega_a$  is called **frequency warping**.

## Bilinear rule with prewarping

### Frequency prewarping

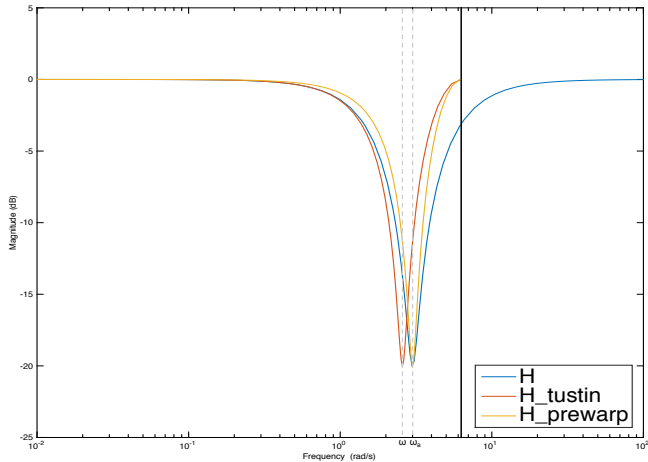
In certain situations, however, we really want the characteristics to be conserved during the discretization.

By setting  $\omega_a = \frac{2}{T} \tan\left(\frac{\omega T}{2}\right)$  for every frequency specification that the designer has control over, the frequency warping will be compensated. This is called **frequency prewarping**.

The digital filter can be made to match the frequency response of the continuous filter at frequency  $\omega_0$  if the following transformation is substituted into the continuous filter transfer function:

$$s \leftarrow \frac{\omega_0}{\tan\left(\frac{\omega_0 T_s}{2}\right)} \frac{z-1}{z+1} \quad (8.4)$$

# Bilinear rule with prewarping



# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Impulse-invariant method

## Practical rule

This method converts a continuous system into a discrete one by matching the impulse response using these transformations.

$\hat{H}_1(p)$	$H_l(z), a=e^{bT}$
$\frac{c}{p-b}$	$\frac{Tc}{1-az^{-1}}$
$\frac{c}{(p-b)^2}$	$T^2 \frac{caz^{-1}}{(1-az^{-1})^2}$
$\frac{c}{(p-b)^3}$	$\frac{T^3 c}{2} \cdot \frac{az^{-1}}{(1-az^{-1})^2} + \frac{2a^2 z^{-2}}{(1-az^{-1})^3} = \frac{T^3 caz^{-1}(1+az^{-1})}{2(1-az^{-1})^3}$
$\frac{c}{(p-b)^4}$	$\frac{T^4 c}{6} \cdot \frac{az^{-1}}{(1-az^{-1})^3} + \frac{6a^2 z^{-2}}{(1-az^{-1})^4} + \frac{6a^3 z^{-3}}{(1-az^{-1})^4}$ $= \frac{T^4 caz^{-1}(1+4az^{-1}+a^2 z^{-2})}{6(1-az^{-1})^4}$



# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Zero-pole equivalent

## General approach

The map  $z = e^{sT}$  is applied to the poles as well as to the zeros of the continuous system. The following rules must be followed:

- 1 If  $s = -a$  is a pole of  $H(s)$ , then  $z = e^{-aT}$ ;
- 2 All finite zeros  $s = -b$  are mapped by  $z = e^{-bT}$ ;
- 3 Zeros at  $\infty$  are mapped to  $z = -1$ ;
- 4 The DC gain is set such that  $\lim_{s \rightarrow 0} G(s) = \lim_{z \rightarrow 1} G_d(z)$ .

## Zero-pole equivalent

### Example

Given:

$$H(s) = \frac{s+1}{0.1s+1}$$

Pole at  $s = -10$  and zero at  $s = -1$ .

New discrete transfer function with equivalent poles and zeros:

$$H(z) = K \frac{z - e^{-T}}{z - e^{-10T}}$$

$K$  is chosen so that  $|H(z)|_{z=1} = |H(s)|_{s=0} \rightarrow K = 4.150$

Using  $T = 0.25$ , this results in:

$$H(z) = 4.150 \frac{z - 0.7788}{z - 0.0821}$$

# Outline

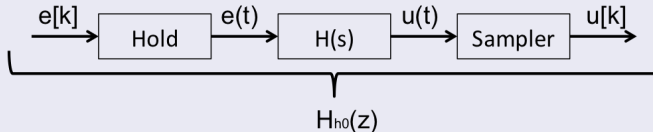
- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Hold equivalent

## General approach

This method uses a discrete system consisting of 3 subsystems, each with its own purpose.

- 1 Hold: approximating  $e_h(t)$  from the samples  $e[k]$
- 2  $H(s)$ : putting the  $e_h(t)$  through the given transfer function  $H(s)$  of the continuous system, resulting in  $u(t)$
- 3 Sampler: sampling  $u(t)$



There are many techniques for holding a sequence of samples.

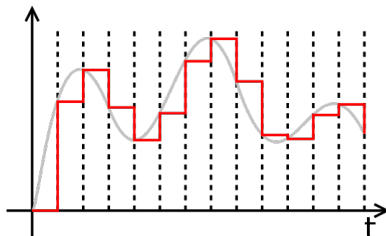
# Zero-order hold equivalent (ZOH)

## Practical rule

The zero-order hold equivalent transfer function  $H_{zoh}(z)$  can be found by computing the following:

$$H_{zoh}(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{H(s)}{s} \right\}$$

This rule is also called the Step-invariant method because it matches the step response of the continuous and the discrete system.



## Zero-order hold equivalent

### Example

$$H(s) = \frac{0.1}{s+0.1}$$

Step 1: multiply by  $1/s$  and perform partial fraction expansion

$$\frac{H(s)}{s} = \frac{0.1}{s*(s+0.1)} = \frac{1}{s} - \frac{1}{s+0.1}$$

Step 2: perform z-transformation

$$\mathcal{Z}\left\{\frac{H(s)}{s}\right\} = \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-0.1T}z^{-1}}$$

Step3: simplify and multiply by  $(1 - z^{-1})$

$$H_{ho}(z) = \frac{1-e^{-0.1T}}{z-e^{-0.1T}}$$

# Zero-order hold equivalent

## Step Invariant Transformation

Zero-order hold equivalent of continuous systems with common transfer functions can easily be found by these transformations.

$f(t)$	$F(s)$	$f(k)$	$F(z)$
$u(t)$ , unit step	$\frac{1}{s}$	$u(k)$ , unit step	$\frac{z}{z-1}$
$tu(t)$	$\frac{1}{s^2}$	$kTu(k)$	$\frac{Tz}{(z-1)^2}$
$e^{-at}u(t)$	$\frac{1}{s+a}$	$(e^{-aT})^k u(k)$	$\frac{z}{z - e^{-aT}}$
$te^{-at}u(t)$	$\frac{1}{(s+a)^2}$	$kT(e^{-aT})^k u(k)$	$\frac{Tze^{-aT}}{(z - e^{-aT})^2}$
$\sin(\omega t)u(t)$	$\frac{\omega}{s^2 + \omega^2}$	$\sin(k\omega T)u(k)$	$\frac{z \sin \omega T}{z^2 - 2z \cos \omega T + 1}$
$\cos(\omega t)u(t)$	$\frac{s}{s^2 + \omega^2}$	$\cos(k\omega T)u(k)$	$\frac{z(z - \cos \omega T)}{z^2 - 2z \cos \omega T + 1}$



# Non-causal first-order (= Triangle-hold equivalent (TRI))

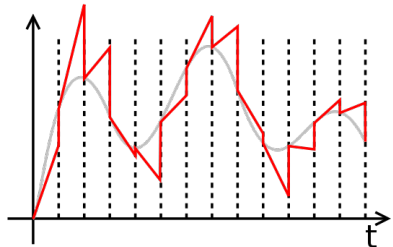
## Practical rule

The non-causal first-order hold equivalent transfer function  $H_{tri}(z)$  can be found by computing the following:

$$H_{tri}(z) = \frac{(z-1)^2}{Tz} \mathcal{Z} \left\{ \frac{H(s)}{s^2} \right\}$$

## Non-causal vs causal

The continuous system is non-causal, but the resulting discrete system is causal.



# Triangle-hold equivalent

## Example

$$H(s) = \frac{1}{s^2}$$

Step 1: multiply by  $1/s^2$  and perform partial fraction expansion

$$H(s) = \frac{1}{s^4}$$

Step 2: perform z-transformation

$$\mathcal{Z}\left\{\frac{H(s)}{s^2}\right\} = \frac{T^3}{6} \frac{(z^2+4z+1)z}{(z-1)^4}$$

Step 3: simplify and multiply by  $\frac{(z-1)^2}{T * z}$

$$H_{tri}(z) = \frac{T^2}{6} \frac{z^2+4z+1}{(z-1)^2}$$

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time**
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

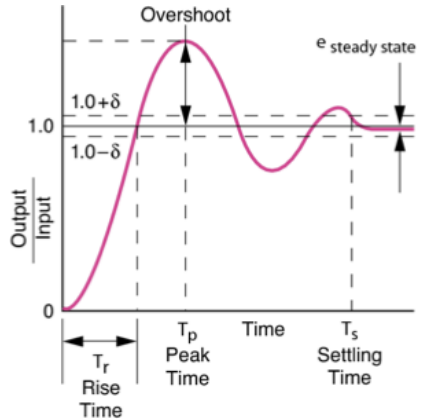
## Sampling time $T_s$ based on the time response

### Rise time

Time needed to reach the steady state for the first time.

### Practical rule

A good rule of thumb is  
 $T_s = T_{rise}/10$ .



## Sampling time $T_s$ based on the frequency response

### Practical rule

A good rule of thumb is  $T_s = 1/(2, 2B)$ .

### Signal vs System

It is very important to understand that although the previous rule of thumb for a system seems the same as the rule of thumb for a signal, they differ! The rule of thumb for a signal uses the bandwidth of the signal and the rule of thumb for a system uses the bandwidth of the system, which can be found in the bodeplot or using MATLAB.

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# MATLAB

## MATLAB Commands for calculations

MATLAB's Control System Toolbox offers extensive support for discretization and re-sampling of linear systems:

- `"c2d(system,sampling time,method)"` used for discretization
- `"d2c(system,sampling time,method)"` used for reconstruction

These commands can also have extra options. It is necessary to specify the options in an additional command:

`"c2dOptions('OptionName',OptionValue)"`.



# MATLAB

## Available options

- 'Method'
- 'PrewarpFrequency'
- 'FractDelayApproxOrder'

## Available methods

- zero-order hold equivalent: "zoh"
- first-order hold equivalent: "foh"
- impulse invariant rule: "impuls"
- zero-pole matching equivalent: "matched"
- bilinear: "tustin"

# MATLAB

## MATLAB Commands for visualisation

- MATLAB can draw the bode plots of the continuous system  
→ command: `"bode(system)";`
- MATLAB can carry out a graphic showing the impulse response of the continuous system and the discretized system  
→ command: `"impulse(H,'b',Hd,'r')"` in which 'b' and 'r' stand for blue and red. By using colours, the step responses of both systems can easily be distinguished;
- MATLAB can carry out a graphic showing the step response of the continuous system and the discretized system  
→ command: `"step(H,'b',Hd,'r')"`.

# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Exercise 1 with MATLAB

## Exercise 1

We will now discuss 4 methods applied on the same continuous system:  $H(s) = \frac{s+1}{s^2+s+1}$

- The sampling time can be determined by the rule of thumb previously explained:  $T_s = 2.2\text{Bandwidth}$ ;
- MATLAB can calculate the bandwidth of a given continuous system, using the command: "bandwidth(system)".

This results in a sampling time of 0.25033 seconds for the given system

# Zero-order hold equivalent

## Example

Given:

$$H(s) = \frac{s+1}{s^2+s+1}$$

Sampling time = 0.25033sec

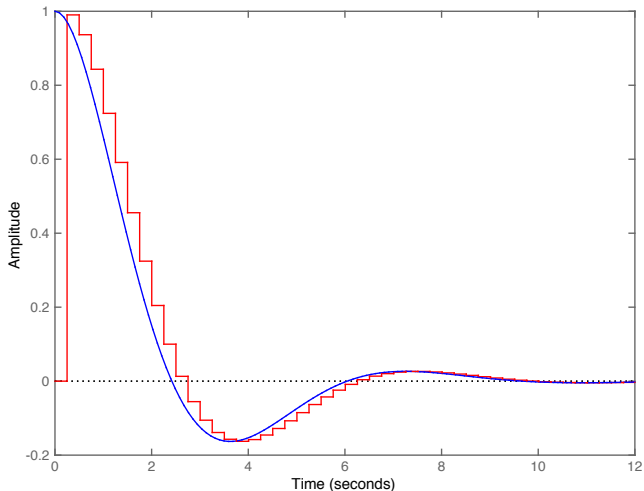
MATLAB commands:

```
H = tf([1 1],[1 1 1])
```

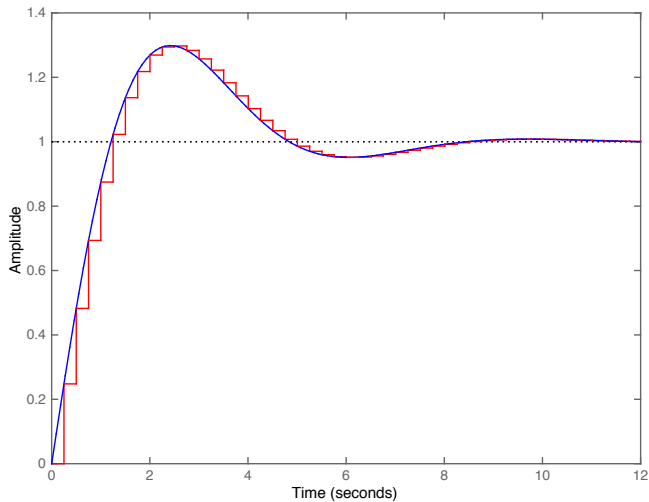
```
Hd = c2d(H,0.25033,'zoh')
```

Result:  $Hd_{zoh}(z) = \frac{0.2479z-0.1927}{z^2-1.723z+0.7785}$

# Zero-order hold equivalent (impulse response)



# Zero-order hold equivalent (step response)



# First-order hold equivalent

## Example

Given:

$$H(s) = \frac{s+1}{s^2+s+1}$$

Sampling time = 0.25033sec

MATLAB commands:

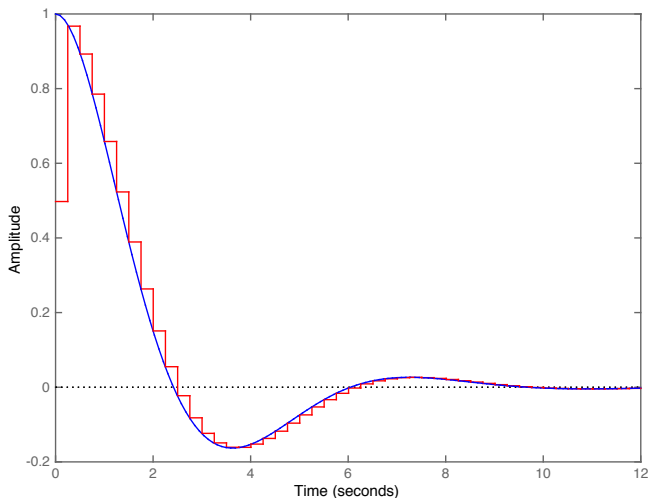
```
H = tf([1 1],[1 1 1]);
```

```
Hd = c2d(H,0.25033,'foh')
```

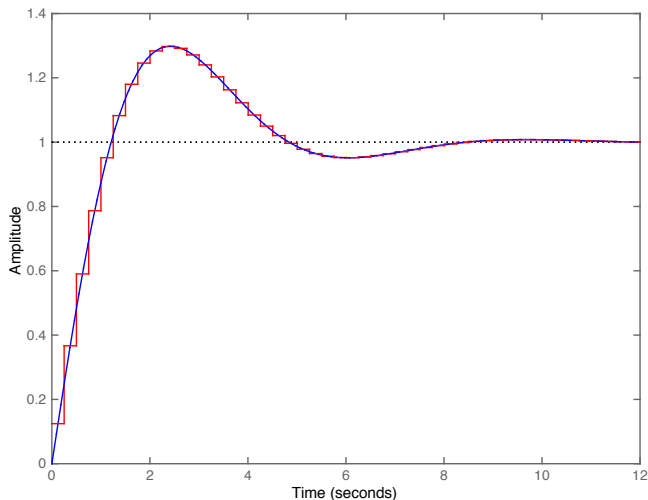
$$\text{Result: } Hd_{foh}(z) = \frac{0.1245z^2+0.02752z-0.09691}{z^2-1.723z+0.7785}$$



# First-order hold equivalent (impulse response)



# First-order hold equivalent (step response)



# Impulse invariant rule

## Example

Given:

$$H(s) = \frac{s+1}{s^2+s+1}$$

Sampling time = 0.25033sec

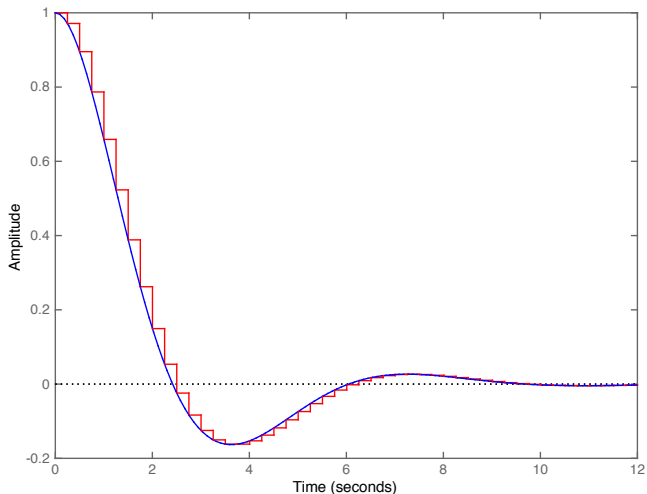
MATLAB commands:

```
H = tf([1 1],[1 1 1]);
```

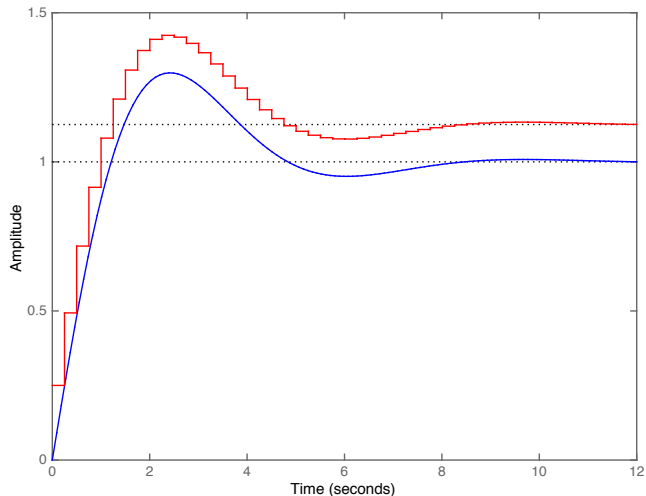
```
Hd = c2d(H,0.25033,'impuls')
```

$$\text{Result: } Hd_{\text{impuls}}(z) = \frac{0.2503z^2 - 0.1883z}{z^2 - 1.723z + 0.7785}$$

# Impulse invariant rule (impulse response)



# Impulse invariant rule (step response)



# Zero-pole equivalent

## Example

Given:

$$H(s) = \frac{s+1}{s^2+s+1}$$

Sampling time = 0.25033sec

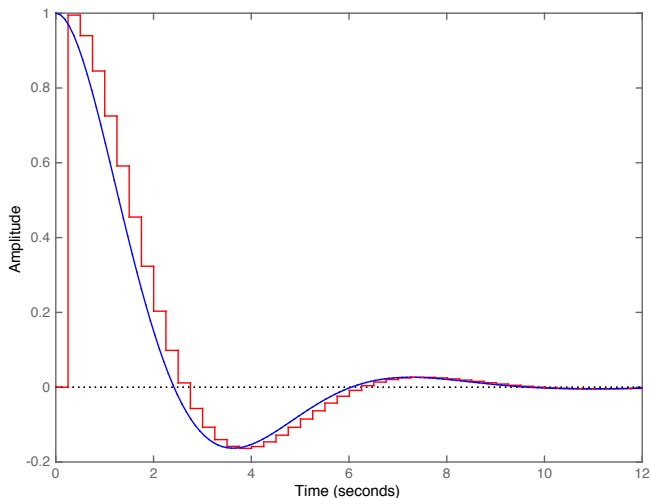
MATLAB commands:

```
H = tf([1 1],[1 1 1]);
```

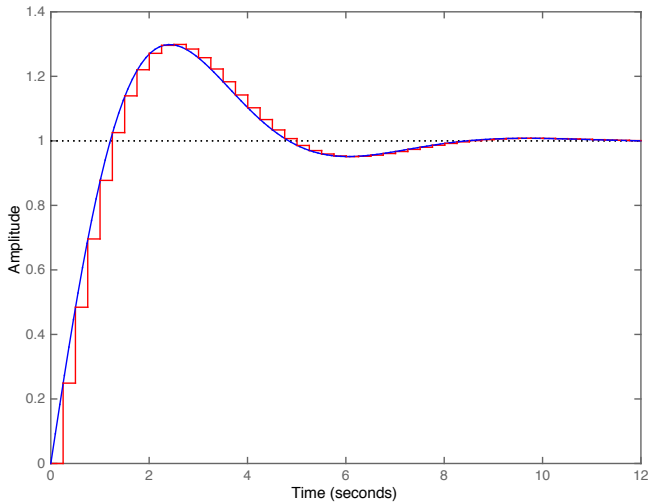
```
Hd = c2d(H,0.25033,'matched')
```

Result:  $Hd_{matched}(z) = \frac{0.249z - 0.1939}{z^2 - 1.723z + 0.7785}$

# Zero-pole equivalent (impulse response)



# Zero-pole equivalent (step response)





## Exercise 2 with MATLAB

### Exercise 2

We will now apply the bilinear rule with and without prewarping on the same continuous system:  $H(s) = \frac{s^2 + 0.5s + 9}{s^2 + 5s + 1}$

Criteria:

- 1 The sampling time is chosen to be 0.5 seconds;
- 2 The discrete system must have the same behaviour as the continuous one at 3rad/s.

The last criteria applies to this specific exercise. It is possible that the parity of the magnitude, of the discrete and the continuous system, at a different frequency (e.g. the peak frequency) is required.

# Tusting rule

## Example

Given:

$$H(s) = \frac{s^2 + 0.5s + 9}{s^2 + 5s + 1}$$

Sampling time = 0.5sec

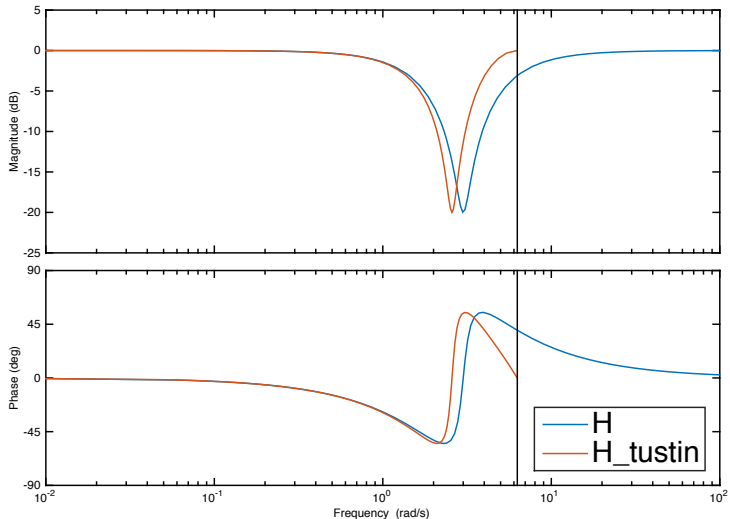
MATLAB commands:

```
H = tf([1,0.5,9],[1,5,9])
```

```
Hdt = c2d(H,0.5,'tustin')
```

$$\text{Result: } Hd_{Tustin}(z) = \frac{0.6z^2 - 0.3111z + 0.5111}{z^2 - 0.3111z + 0.1111}$$

# Tustin rule (bode plot)



# Tustin rule with prewarping

## Example

Given:

$$H(s) = \frac{s^2 + 0.5s + 9}{s^2 + 5s + 1}$$

Sampling time = 0.5sec

MATLAB commands:

```
H = tf([1,0.5,9],[1,5,9])
```

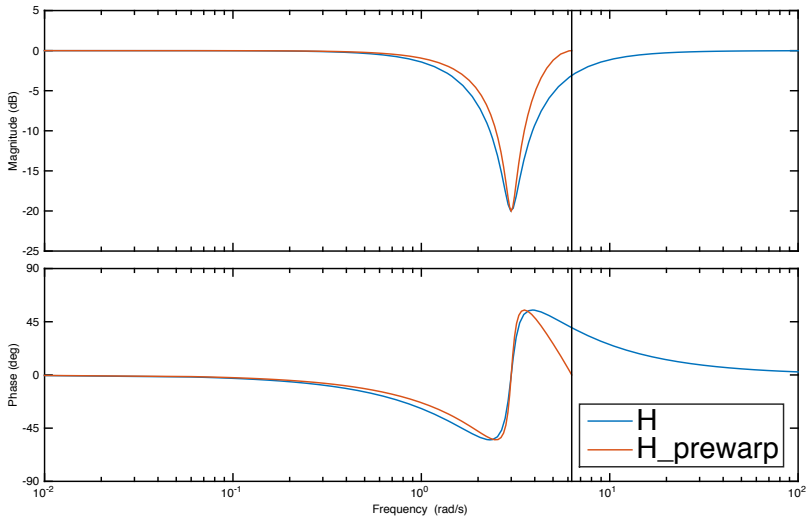
```
damp(H) (results in 3.0Hz)
```

```
discopts = c2dOptions('Method','tustin','PrewarpFrequency',3.0);
```

```
Hdtp = c2d(H,0.5,discopts)
```

Result:  $Hd_{TustinPrewarp}(z) = \frac{0.5915z^2 - 0.07726z + 0.5007}{z^2 - 0.07726z + 0.09215}$

# Tustin rule with prewarping (bodeplot)



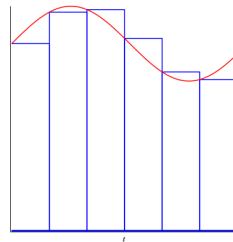
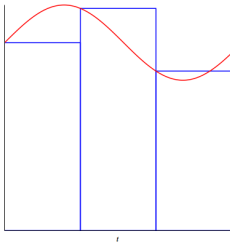
# Outline

- 1 Introduction
- 2 Main Approaches
  - Numerical Integration
  - Impulse Invariant Method
  - Zero-pole Equivalent
  - Hold Equivalent
- 3 Sampling Time
- 4 Discretization and MATLAB
  - Commands
  - Exercises
- 5 Addition Material

# Forward rectangular rule (= (Forward) Euler)

## General approach

The area is approximated by the rectangle looking **forward** from  $(kT - T)$  toward  $kT$  with an amplitude equal to the value of the function at  $(kT - T)$ . A smaller step-size  $T$  leads to a more accurate approximation, as shown in the figures.



# Forward rectangular rule

## Mathematical approach

The general approach (formula (8.1)) applied on the forward rectangle rule, results in an approximation  $u_1$ :

$$\begin{aligned}u_1(kT) &= u_1(kT - T) + T(-au_1(kT - T) \\ &\quad + ae(kT - T)) \\ &= (1 - aT)u_1(kT - T) + aTe(kT - T) \\ aTe(kT - T) &= -(1 - aT)u_1(kT - T) + u_1(kT)\end{aligned}$$



## Forward rectangular rule

### Mathematical approach

By applying the z-transform, we can find the transfer function:

$$\begin{aligned} H_F(z) &= \frac{aTz^{-1}}{1 - (1 - aT)z^{-1}} \\ &= \frac{a}{\frac{z-1}{T} + a} \end{aligned}$$

Which can also be derived using the following substitution in the given transfer function:

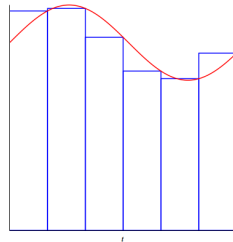
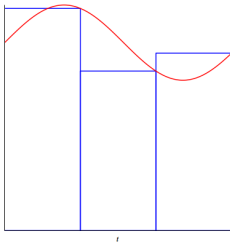
$$s \leftarrow \frac{z-1}{T_s} \quad (8.2)$$

This is extremely useful while making exercises.

# Backward rectangular rule (=Backward Euler)

## General approach

The area is approximated by the rectangle looking **backward** from  $kT$  toward  $(kT - T)$  with an amplitude equal to the value of the function at  $kT$ .



# Backward rectangular rule

## Mathematical approach

The general approach (formula (8.1)) applied on the backward rectangle rule, results in an equation  $u_2$ :

$$\begin{aligned} u_2(kT) &= u_2(kT - T) + T(-au_2(kT) + ae(kT)) \\ &= \frac{u_2(kT - T)}{1 + aT} + \frac{aT}{1 + aT}e(kT) \end{aligned}$$

## Backward rectangular rule

### Mathematical approach

By applying the z-transform, we can find the transfer function:

$$\begin{aligned} H_B(z) &= \frac{aT}{1+aT} \frac{1}{1 - \frac{z^{-1}}{1+aT}} \\ &= \frac{aTz}{z(1+aT) - 1} = \frac{a}{\frac{z-1}{Tz} + a} \end{aligned}$$

Which can also be derived using the following substitution in the given transfer function:

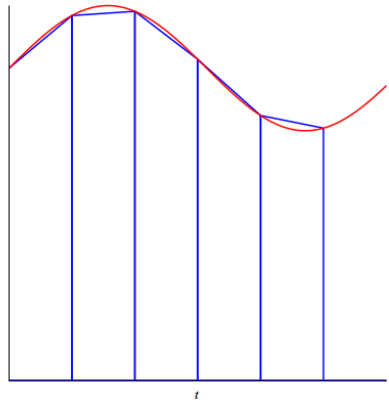
$$s \leftarrow \frac{z-1}{zTs} \quad (8.3)$$

Again this is extremely useful while making exercises.

## Bilinear rule (= trapezoidal or Tusting rule)

### General approach

This method makes use of the area of the **trapezoid** formed by the average of the selected rectangles used in the forward and backward rectangular rule. Thus the amplitude equal to the value of the function at  $(kT - T)$  and the amplitude equal to the value of the function at  $(kT)$  are connected by a line as shown in the illustration.



# Bilinear rule

## Mathematical approach

The general approach (formula (8.1)) applied on the forward rectangle rule, results in an equation  $u_3$ :

$$\begin{aligned} u_3(kT) &= u_3(kT - T) + T/2(-au_3(kT - T) \\ &\quad + ae(kT - T) - au_3(kT) + ae(kT)) \\ &= \frac{1 - (aT/2)}{1 + (aT/2)} u_3(kT - T) \\ &\quad + \frac{aT/2}{1 + (aT/2)} (e_3(kT - T) + e_3(kT)) \end{aligned}$$

## Bilinear rule

### Mathematical approach

By applying the z-transform, we can find the transfer function:

$$\begin{aligned} H_T(z) &= \frac{aT(z+1)}{(2+aT)z + aT - 2} \\ &= \frac{a}{\frac{2}{T} \frac{z-1}{z+1}} \end{aligned}$$

Which can also be derived using the following substitution in the given transfer function:

$$s \leftarrow \frac{2}{T_s} \frac{z-1}{z+1} \quad (8.4)$$

This is extremely useful while making exercises.