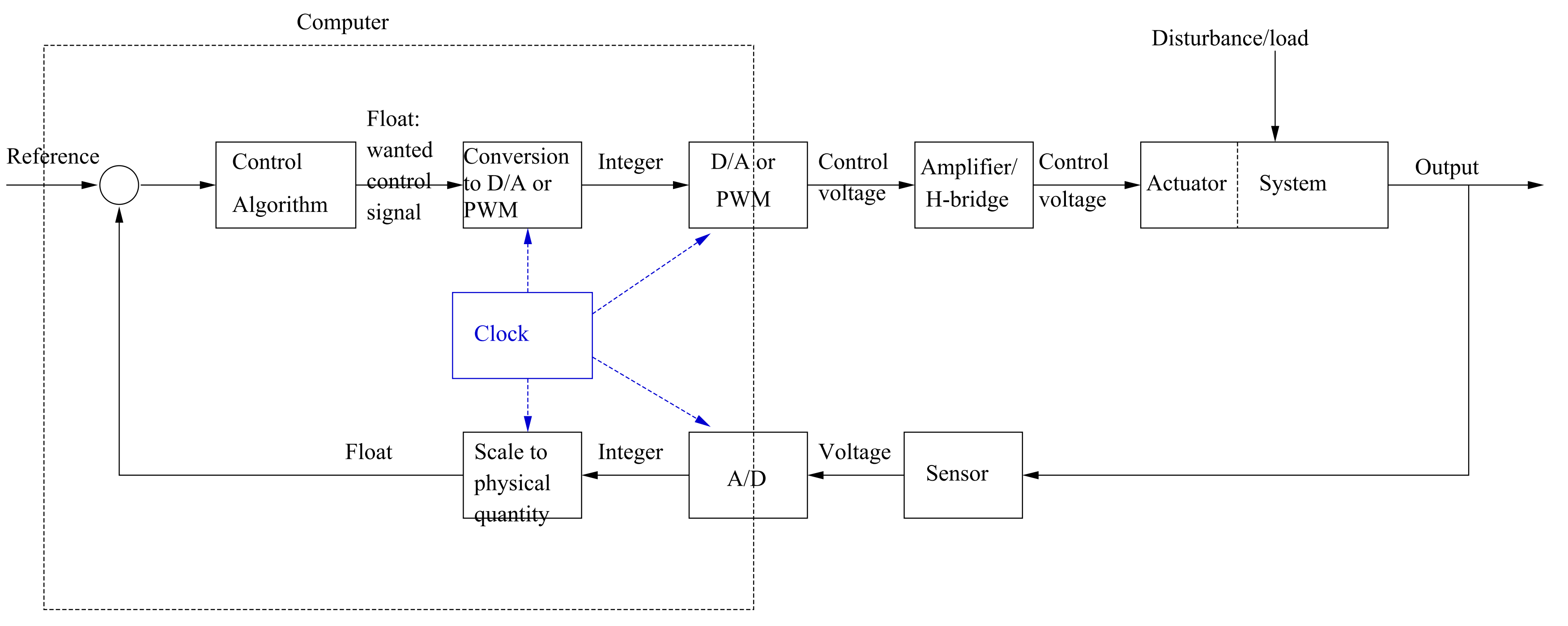
Outline

Introduction to digital control

- Effect of sampling

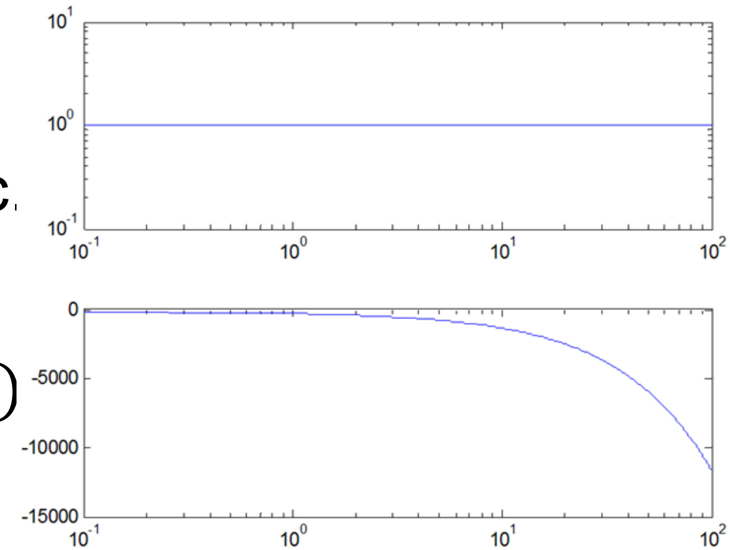
Control in a lab environment

Standard Control system

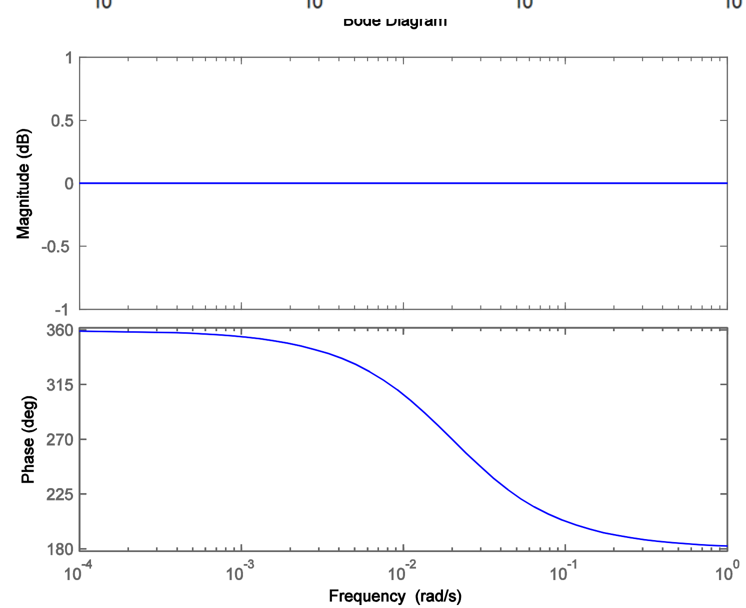


Time delays caused by the system, the sensor or sampling

- A function f(t) which is delayed Λ sec.
- has the Laplace transform
- $F(s) = \int_0^\infty f(t-\Lambda)e^{-s\Lambda}dt = e^{-\Lambda s}F(s)$

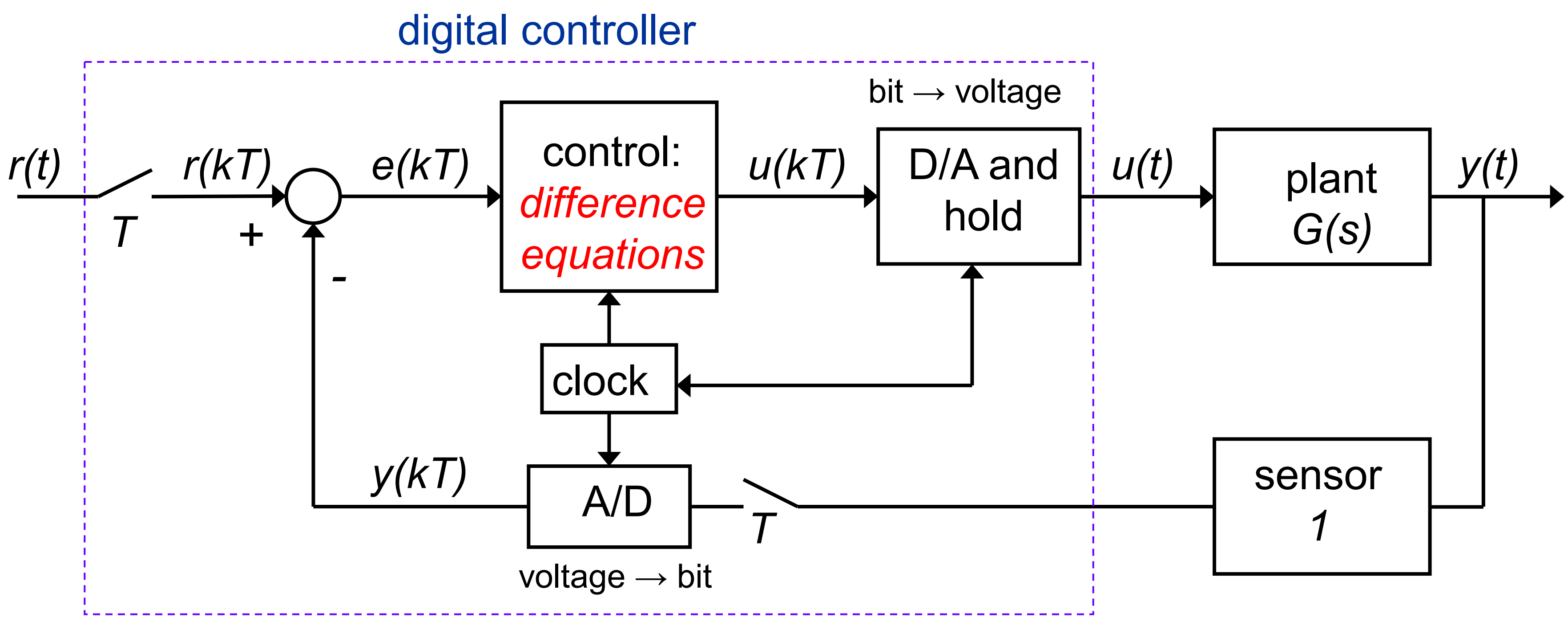


- Pade approximation:
- $e^{-\Lambda s} \sim \frac{1-\Lambda/_2 s}{1+\Lambda/_2 s}$
- For small Λ $\quad e^{-\Lambda s} \sim \frac{1}{1+\Lambda s}$
- Minimize the delays if possible

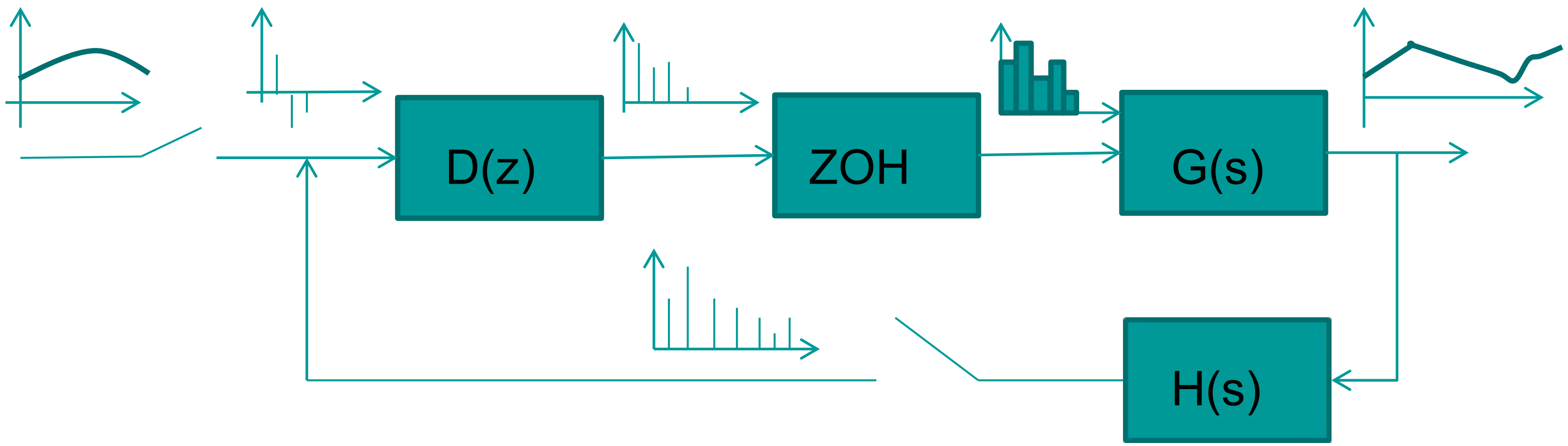


Discretisation

- Digital Control System
  - *T* is the sample time (*s*)
  - Sampled signal : $x(kT) = x(k)$



Discrete control



The sample time must be constant !!!!

Discretization of the controller

Continuous control vs. digital control

- Basically, we want to simulate the controller D(s)
- D(s) contains *differential equations* (time domain) – must be translated into *difference equations.*

Variables are measured/calculated values at kT

Derivatives are approximated (Euler's method)

- $\dot{x}(k) \approx \frac{x(k+1)-x(k)}{T}$

Discretization

Example

Using Euler's method, find the difference equations.

$$D(s) = \frac{U(s)}{E(s)} = K_0 \frac{s+a}{s+b}$$

Differential equation

$$(\mathrm{s}+\mathrm{b})\mathrm{U}(\mathrm{s}) = K_0(s+a)E(s) \xrightarrow{L^{-1}} \dot{u} + bu = K_0(\dot{e} + ae)$$

Using Euler's method

$$\frac{u(k+1)-u(k)}{T} + bu(k) = K_0\left(\frac{e(k+1)-e(k)}{T} + ae(k)\right) \quad \Rightarrow$$

$$u(k+1) = (1-bT)u(k) + K_0(aT-1)e(k) + K_0 e(k+1)$$

Implementation of controller

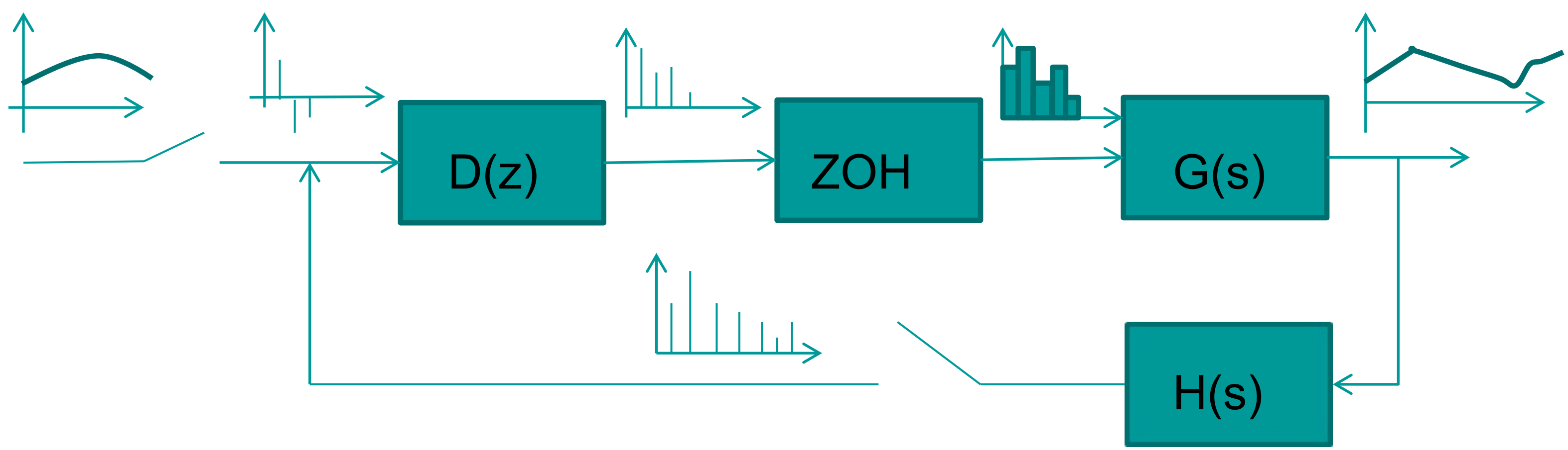
$$u(k+1) = (1-bT)u(k) + K_0(aT-1)e(k) + K_0 e(k+1)$$

Pseudo code

x=0 initialization of past values for first time trough the loop
K_o = gain
C1 = 1-bT
C2=K_o(aT-1)
Read r and y from the A/D converter
e = r-y
u = x+K_oe
Send u to the D/A converter
x= C1u+C2e (update x for next run through the loop)
Wait until T seconds from last read

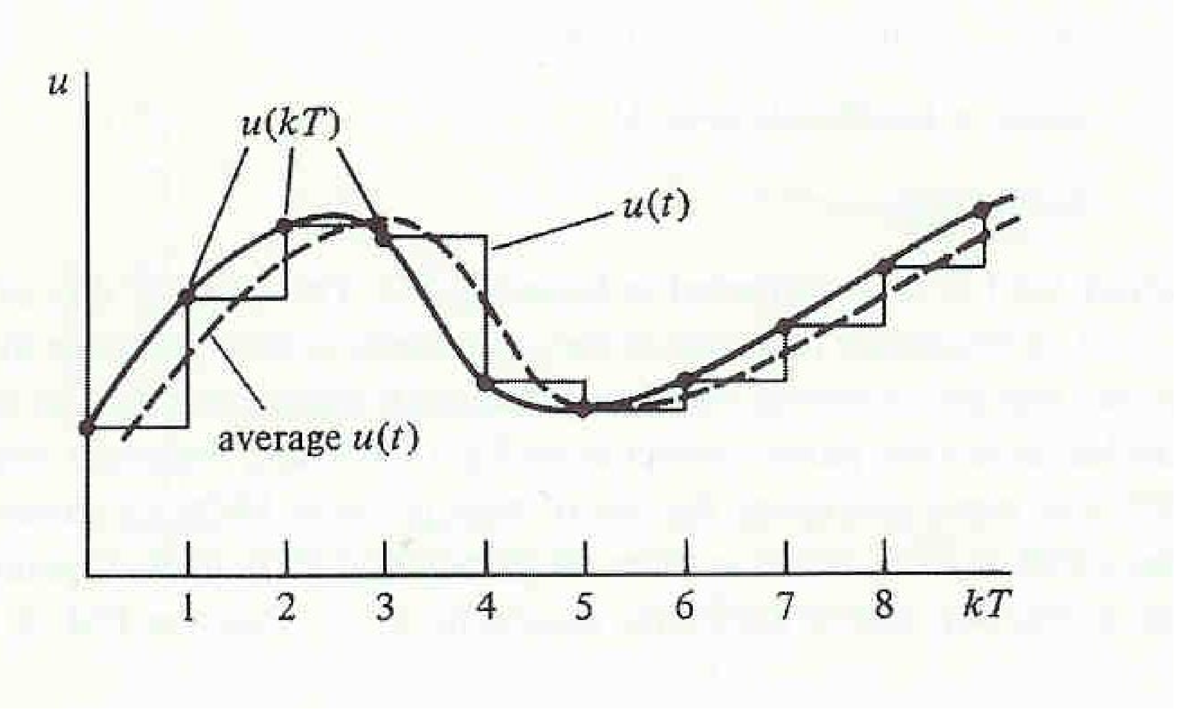
Discretisation : The “hold “

Most discret controller outputs are implemented as a series of steps – the controller output value at a sampletime T are hold constant until the next sampletime T+T_s where a new controller output value are ready => ZOH is a realistic description



Effect of sampling

D/A in output from controller



The single most important impact of implementing a control digitally is the delay associated with the hold.

Effect of sampling

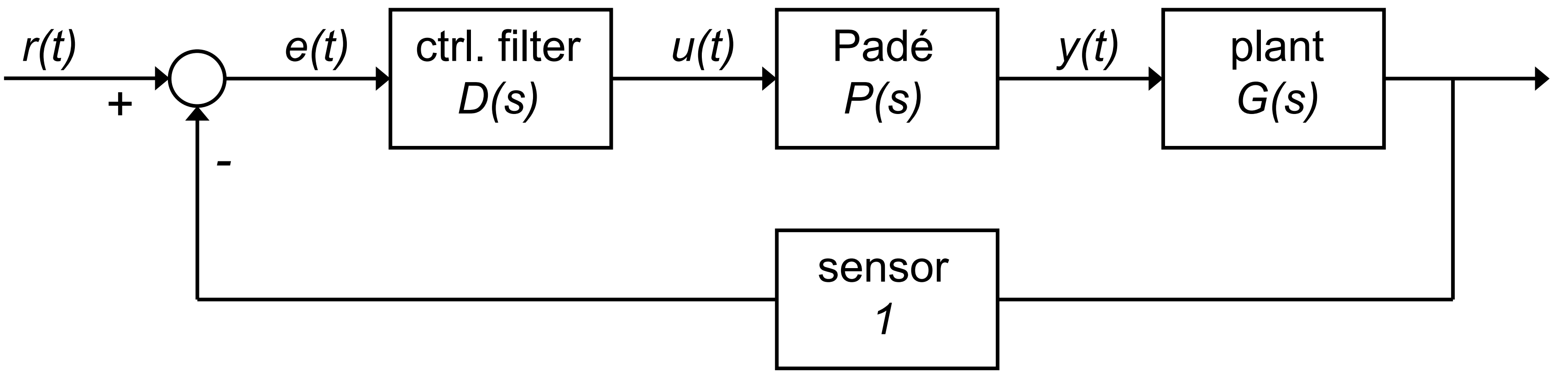
- Analysis
  - Approximately 1/2 sample time delay

$$T_d = \frac{T}{2}$$

  - Can be approx. by Padè
    (and cont. analysis as usual)

  Positive right plane zero

$$P(s) = \frac{s - 2/T}{s + 2/T}$$



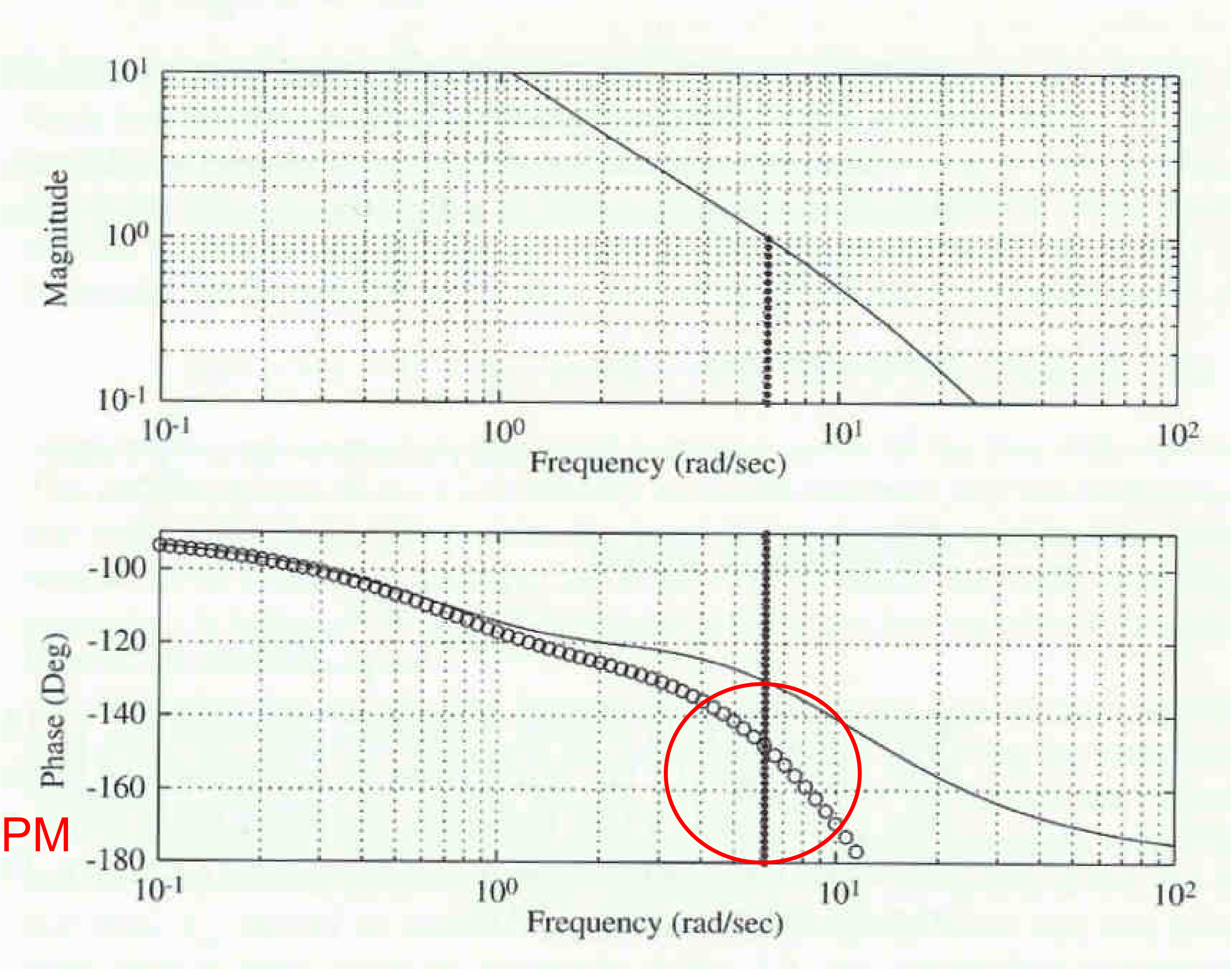
Effect of sampling

Example of phase lag by sampling

Example from before with sample rate = 10 Hz

Notice PM reduction

All delays will cause PM reduction



If the sample time is varying the stability is unknown !!!!

Discretization

Significance of sampling time T

Example controller *D(s)* and plant *G(s)*

$$D(s) = 70\frac{s+2}{s+10} \quad , \quad G(s) = \frac{1}{s(s+1)} = \frac{1}{s^2+s}$$

Compare – investigate using Matlab

1) Closed loop step response with *continuous* controller.

2) Closed loop step response with *discrete* controller.
Sample rate = 20 Hz

3) Closed loop step response with *discrete* controller.
Sample rate = 40 Hz

Discretisation

Controller *D(s)*
and plant *G(s)*

$$D(s) = 70\frac{s+2}{s+10}$$

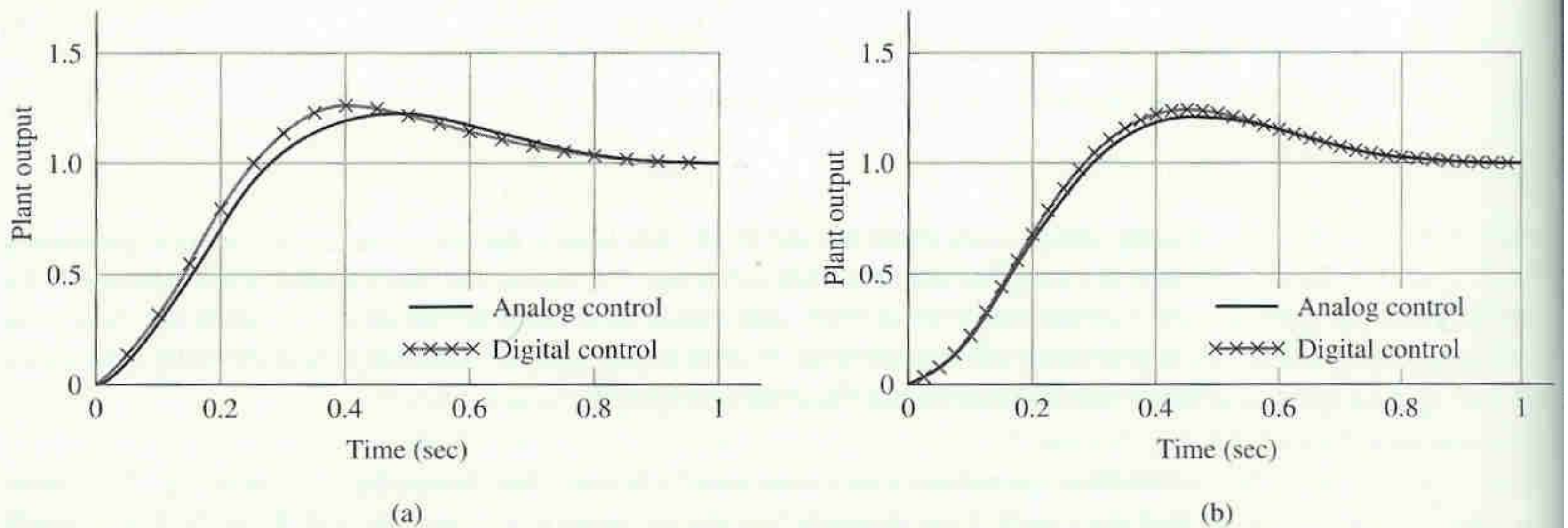
$$G(s) = \frac{1}{s^2+s}$$

Matlab – continuous and discrete controller

```
numD = 70*[1  2];   denD = [1   10];
numG = 1;   denG = [1   1   0];
sysOL = tf(numD,denD) * tf(numG,denG);
sysCL = feedback(sysOL,1);
sysCLd = c2d(sysCL,T,'zoh');  % use zero order hold
                                 and sample time T to convert
                                 from continous to discrete
step(sysCL,'-',sysCDd,'--');
```

Discretisation

Figure 3.2
Continuous and digital step response using Euler's method for discretization: (a) 20 Hz sample rate, (b) 40 Hz sample rate



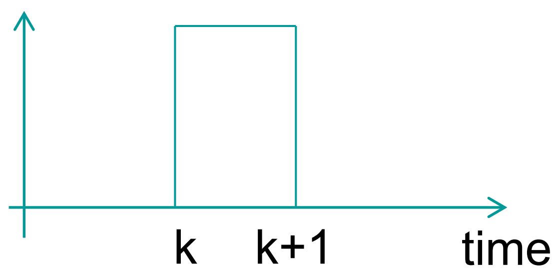
Notice, high sample frequency (small sample time *T*) gives a good approximation to the continuous controller

Design using discrete equivalents

- Design a continuous compensation
- Find the discrete equivalent best approximates the continuous controller
- Relate the sample time to the closed loop dynamics
- Use discrete analysis, simulation or experiments to verify

Discrete representation of D(s) and ZOH

- Given a input e(k)
- D(s) preceded by ZOH corresponds to a positive step at sample time k and a negative step at sample time k+1



A positive step input at time k

$$E(s) = \frac{e(k)}{s}$$

A negative stepinput at time k+1

$$E(s) = \frac{-e(k)}{s}$$

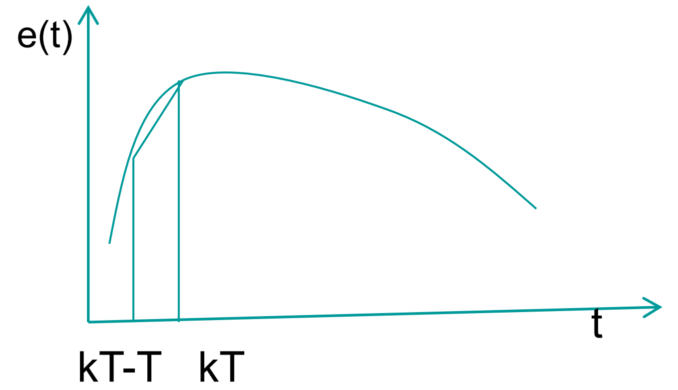
$$D(z) = Z\left\{\frac{D(s)}{s}\right\} + z^{-1}Z\left\{\frac{-D(s)}{s}\right\} = (1 - z^{-1})Z\left\{\frac{D(s)}{s}\right\}$$

Matlab: Dz = c2d(D,T,'zoh');

Tustin's Method

Trapezoidal integration

e(t) is approximated by a straight line between two samples



$$u(k) = u(k-1) + \frac{T}{2}\big(e(k-1) + e(k)\big)$$

or

$$\frac{U(z)}{E(z)} = \frac{T}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right)$$

Hd = c2d(H, 0.1, 'tustin');

Discrete control
z-plan specifications

- The z-transform is the mathematical tool for the analysis of linear discrete systems

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

$$Z\{f(k-1)\} = z^{-1} F(z)$$

- The transfer function of a discrete system given by a difference equation can be found

Difference equation to discrete transfer function

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2)$$

$$Y(z) = (-a_1 z^{-1} - a_2 z^{-2}) Y(z) + (b_0 + b_1 z^{-1} + b_2 z^{-2}) U(z)$$

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{N(z)}{D(z)} = K \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)}$$

z^{-1} is equivalent to one sample delay

Correspondence with cont. signals

General s - plane pole : $s = \sigma + j\omega$

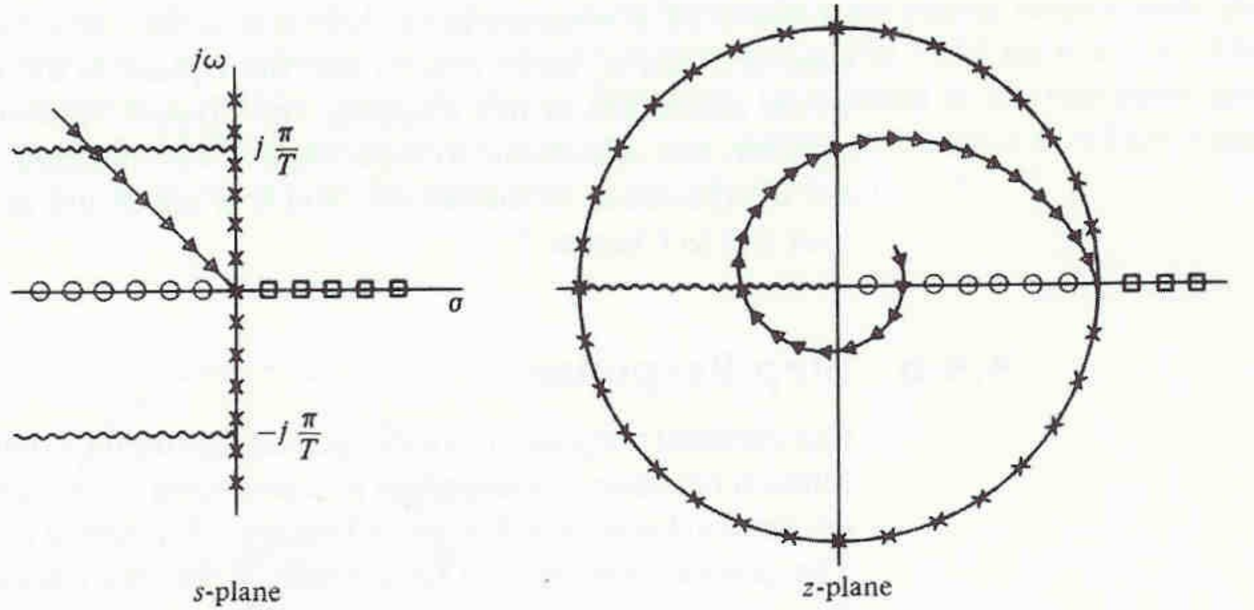
$s = j\omega \quad \leftrightarrow \quad |z| = 1$

$s = \sigma, \quad \sigma \geq 0 \quad \leftrightarrow \quad z = r, \quad r \geq 1$

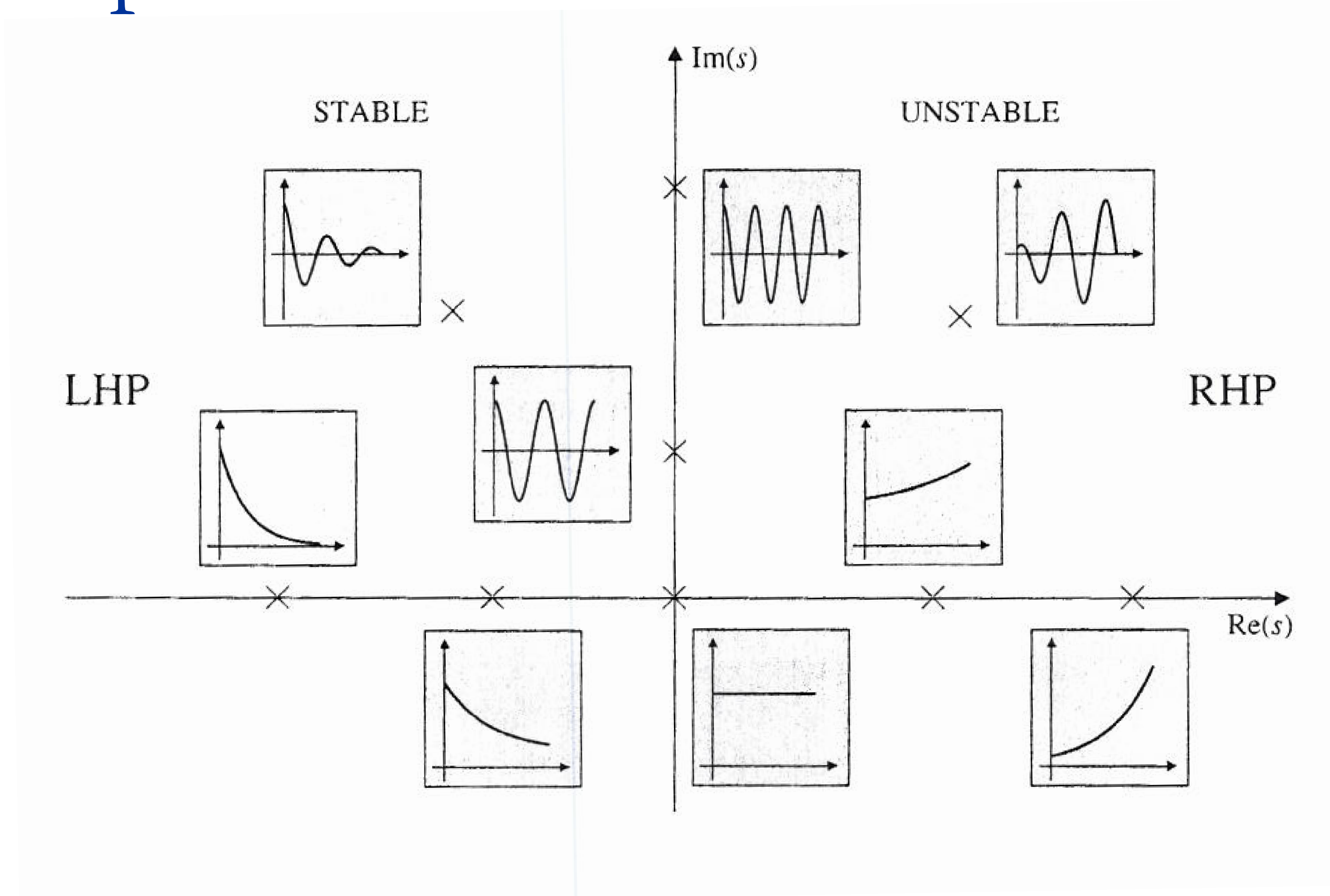
$s = \sigma, \quad \sigma \leq 0 \quad \leftrightarrow \quad z = r, \quad 0 \leq r \leq 1$

Pole map

$$\boxed{z = e^{sT}}$$

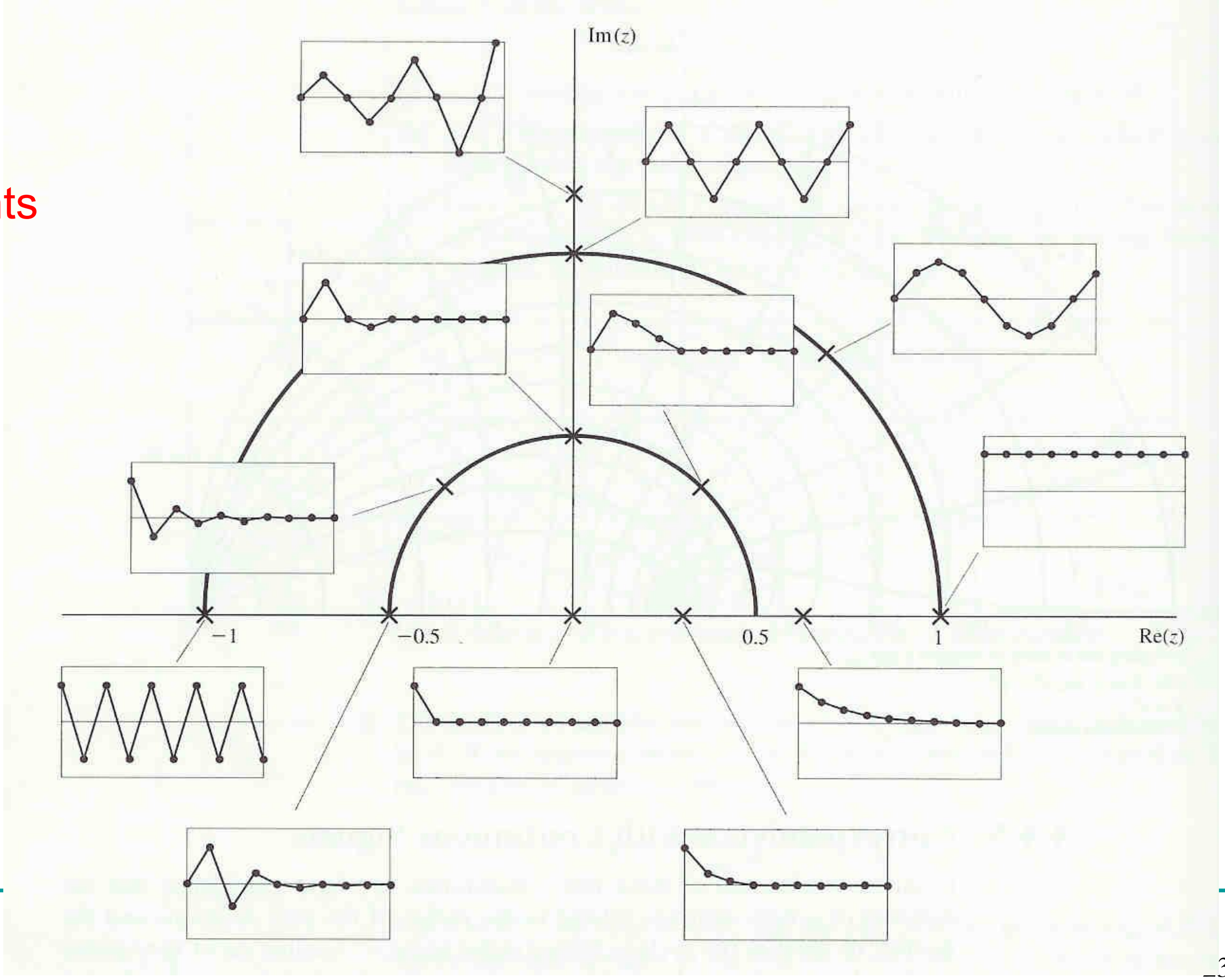


Time functions associated with poles in the s plane

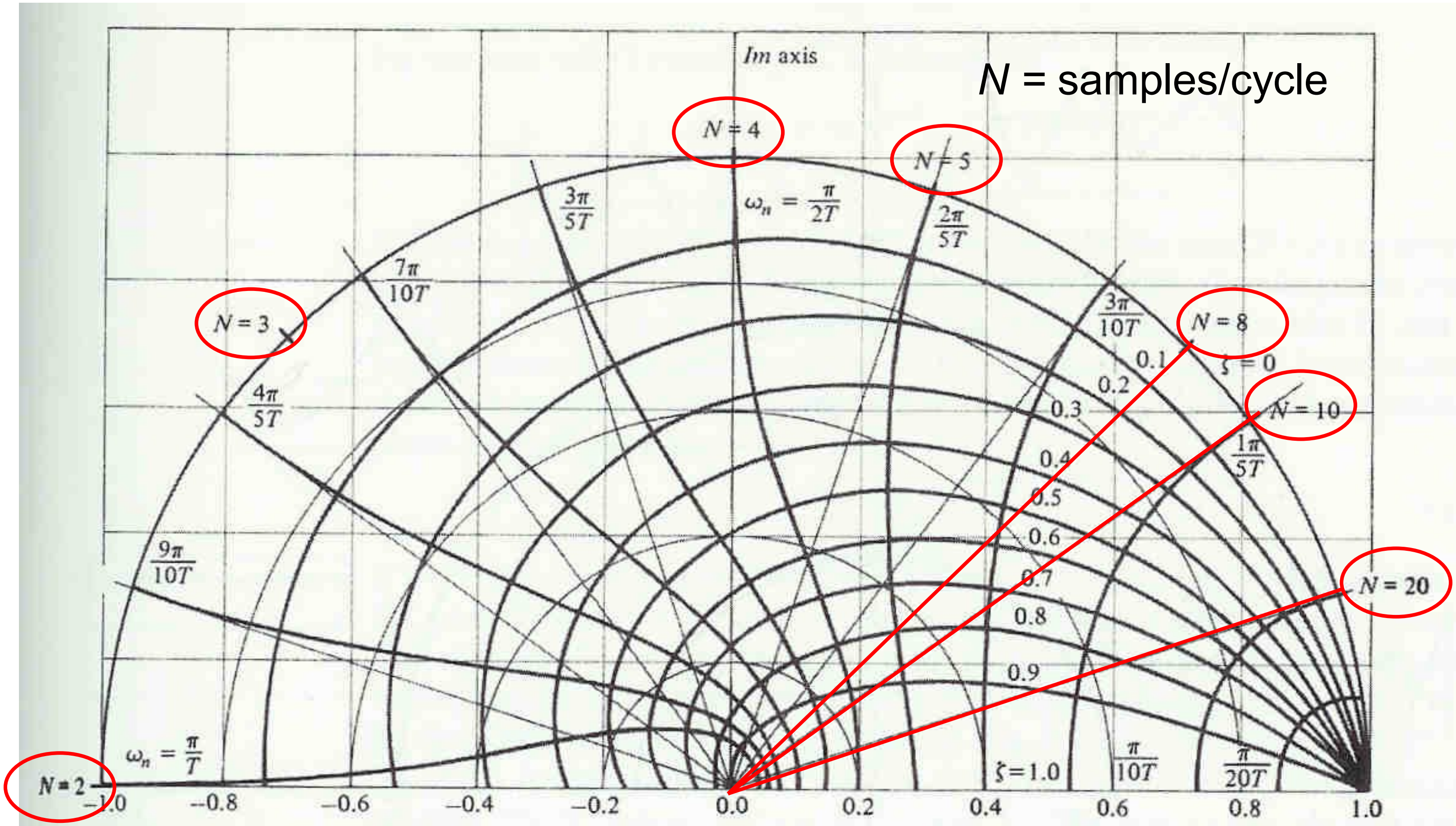


Time sequences associated with points in the z-plane.

Pole placements



Signal analysis – discrete signals



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

- The duration settling of a time signal is related to the radius of the pole locations

  the closer r is to 0 the shorter settling time
- The number of the samples per cycle N is related to the angle θ

$$N = \frac{360}{\theta} \quad \frac{\text{samples}}{\text{cycle}}$$

The rule of thumb is to sample 20 times faster than the time constant

BREAK

Control in a lab environment

Things to remember

The controller design and implementation procedure

- Make a dynamic model of the system
- Descide on inputs, outputs and disturbances
- Descide the controller structure.
- Adjust the dynamic model to describe the input/output relations
- Controller design – inner loop first
- Test by simulation
  - linear model
  - taking known non linearities into account
- Implementation – gradually
- Adjust parameters
- Compensation for real life issues

Dynamic Model

Usually

- The system is approximated by a combination of first and second order elements

- The elemets are found by e.g.
  - Stepresponses
  - Deduction based on physical laws

The real world is different from the Matlab world.

- The model is linear – the real world in non-linear
  - Non-linearities can be e.g
    - saturation,
    - slew rate,
    - non-linear amplifiers – dead zones
    - friction
- High frequency dynamics are not included in the model
- Disturbances
- Noise

The real world is different from the Matlab world.

- High frequency poles could be left out of the model. They can cause problems if the bandwith is increased.

Example

- Crane : the wire can be modeled via two complex conjugated poles causing wire oscillations

Controller design

Dimension the controllers starting from the innermost loops

- Simple first
- Start out using a proportional controller – it might do the job – it shows bad scaling
- Add an I elemet in the controller to remove stationary error
  - An extra zero (lag) can avoid infinit DC gain
- Add an D element to get a higher bandwith – to get a faster system
  - An extra pole (lead) can avoid high frequency gain

Simulation

- Test by simulation
- Is the controller output adjusting the wanted system output in the right direction
- If the structure is wrong – make a re-design

- Test the signals from the controller to the system. Is the level of the signals realistic – otherwise adjust the controllers

Implementation

- Discretize the controller
- The sampletime is found from the fastes dynamics in the system to be controlled
- The sampletime must be constant.
- Implement the controllers seperately if possible
- Implement the controller from low level to high level corresponding to inner loops first.
- The controller parameters are initial they must be adjusted in lab.
- Rise time,setling time and overshoot can be tested in the linear operating area – not from min to max.

Test purposes

Test for ability to

- follow a reference input
- suppress a disturbance
- suppress noise

Test strategy

- Small steps !!! Not the entire system immediately
- Test the controllers seperately
- Stabilize from the lowest level
- For cascade structures –
  - test from the inner loop and adjust
  - Add outer loops one by one

Investigation of signals from the controller in closed loop

- Investigate the output signal from the controller to the system in the closed loop
- If the singnal is jumping between max and min then e.g. the gain is to high
- If the signal never reach the limitations the design could have been faster or a smaller e.g. moter could have been choosen.

Test using step response- very common

- Use a stepsize sufficiently large to show the dynamics, small enough to avoid saturations
- When using a small step – keep the noise in mind.
- Don’t use a step from minimum to maximum input value – you will post probably enter a non-linear zone.
- Rise time, setling time and overshoot are only defined for the linear controller, a test from maximum to minimum input will most likely be affcted by system limitations.
- Large step input can be violent/tough – in some cases ramp inputs are more manageable and closer to reality

Typical problems

- Controller parameter values from the off-line controller design are initial. They have to be adjusted in the lab.
- Dead zones caused by friction must be removed using offsets
- ‘I’ control in systems with saturations can cause integrator wind up
- Trembling behaviour can be caused by a low phase margin or impact from external forces.

Noise

- Noise is not included in the model
- Measurement noise often originate from discrete transducers. Analog transducers doesn’t have the same problem.
- Calculate the transfer function from noise to output-maybe you will find high gains
- The transfer function from the origin of the noise to the output must be stable