Dynamics and Motion control

Lecture 3

Feedback control -continuous time design

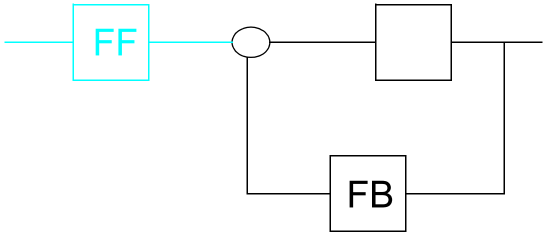
Bengt Eriksson
KTH, Machine Design
Mechatronics Lab
e-mail: benke@md.kth.se

Lecture outline

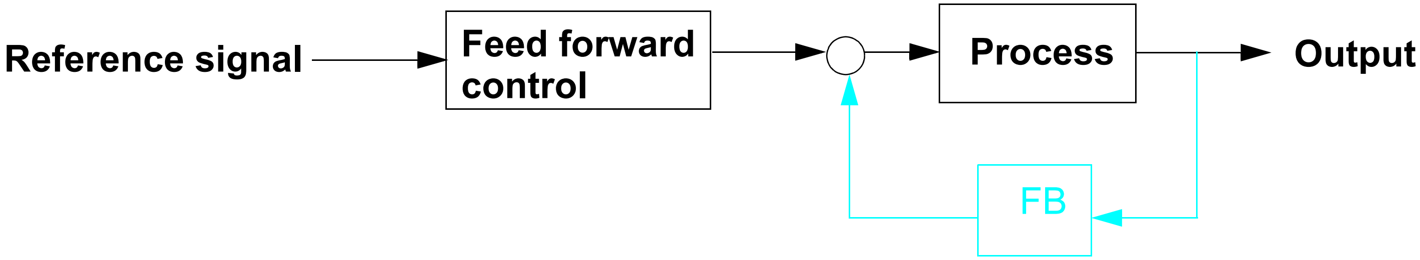
- **1. Introduction**
- 2. Model based Control: a motivating example
- 3. Pole placement design
- 4. Discrete time approximation of the continuous time control
- 5. Example, a PD position controller
- 7. Cascaded motion control architecture
- 8. Antiwindup

3.1. Feedback control properties

- The main principle in control engineering
- Typically model based (but not required to be)
- Produces control signals after an error has occurred
- Disturbance rejection is achieved
- Effect of process parameter variations is reduced
- Leads to a closed loop
- May lead to instability if designed incorrectly
- Sensor noise may be amplified and deteriorate performance



3.1.1 Feed forward control properties



- Produces control signals prior to that an error has occurred
- Uses carefully designed reference signals to make the process follow the references “exactly”

3.1.2 The servo vs. regulator problem

- The **regulator** problem: -> FEEDBACK

  Find a feedback controller that satisfies the specifications on

  -sensor noise, disturbance rejection and robustness to model and parameter uncertainties

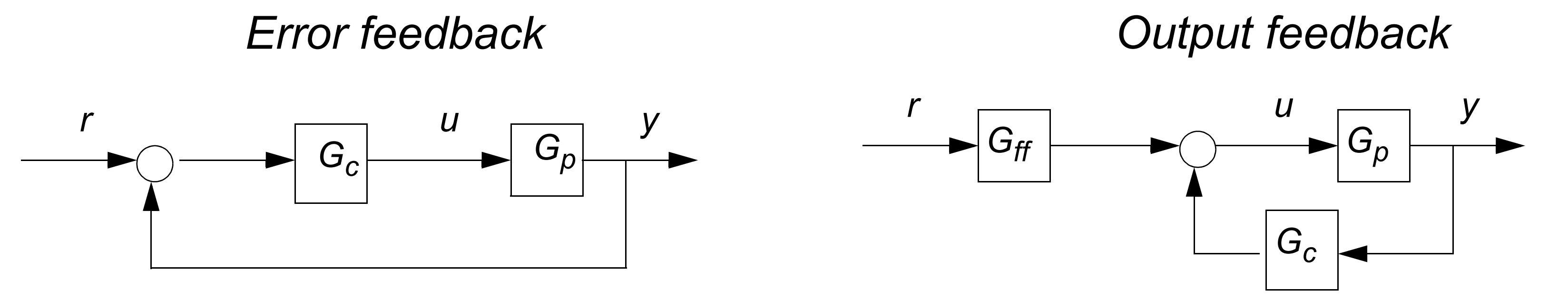
- The **servo** problem: -> FEED FORWARD

  Find a feed forward controller that **tracks the references** according to specifications (a feedback must already exist)

  -Steady state accuracy, overshoot, tracking error, settling time

3.1.3 Error vs. output feedback

- More design freedom with Output feedback, also called 2DOF control



u = control or input signal, y = output or measured signal, r = reference or command signal

- Example PD-control

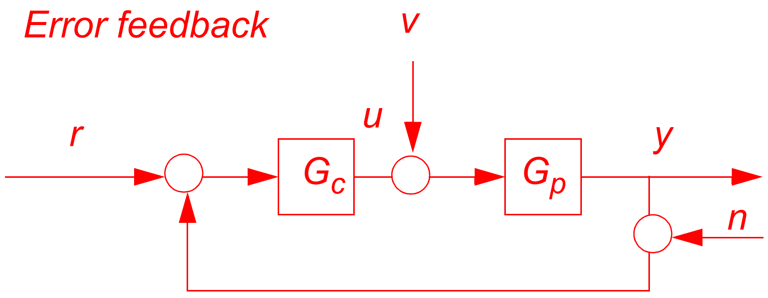
If, $G_c(s) = G_{ff}(s) = P + Ds$, then both structures are equal

Example. If we don't want derivative action on the reference signal we should instead choose $G_{ff}(s) = P$ and $G_c(s) = P + Ds$.

3.1.4 Noise and disturbance models

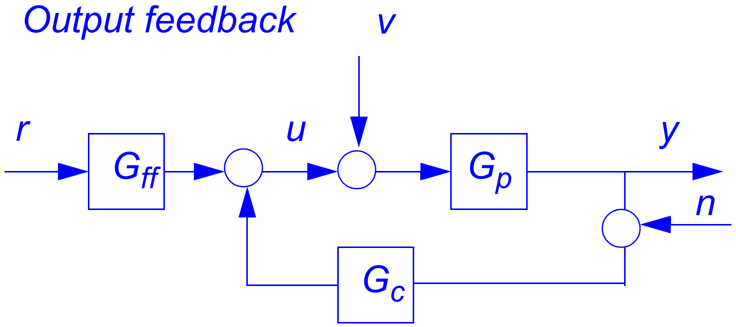
Load disturbance, v(s)
Sensor noise, n(s)

Error feedback



$$y = \frac{G_p G_c}{1 + G_p G_c} r + \frac{G_p}{1 + G_p G_c} v - \frac{G_p G_c}{1 + G_p G_c} n$$

Output feedback



$$y = \frac{G_p G_{ff}}{1 + G_p G_c} r + \frac{G_p}{1 + G_p G_c} v - \frac{G_p G_c}{1 + G_p G_c} n$$

- **TF from reference, *r* to output *y* are different.**

3.1.5 Polynomial models

Set: $G_p(s) = \frac{B(s)}{A(s)}$

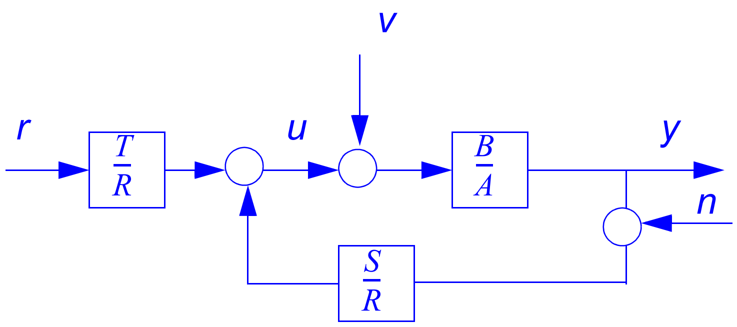
$G_c(s) = \frac{S(s)}{R(s)}$

$G_{ff}(s) = \frac{T(s)}{R(s)}$

Control law

$$u(s) = \frac{T}{R}r - \frac{S}{R}y$$

Output feedback



Closed loop respones

$$y = \frac{BT}{AR+BS}r + \frac{BR}{AR+BS}v - \frac{BS}{AR+BS}n$$

For error feedback is, $T = S$

3.2. Lecture outline

- 1. Introduction
- **2. Model based Control, a motivating example**
- 3. Pole placement design
- 4. Discrete time approximation of the continuous time control
- 5. Example, a PD position controller
- 6. Example, a PD position controller
- 7. Cascaded motion control architecture

3.2.1 Motivating exampl

- **Model based T.F control design**

DC-motor model from lec. 2 $\boldsymbol{J\ddot{\varphi} = k_m i - d\dot{\varphi}}$

$J = 0.08$: rotor inertia
$k_m = 3.6$: torque constant
$d = 0.45$: friction coefficient
i: rotor current
φ: angular position

set $\boldsymbol{y} = \varphi$ and $\boldsymbol{u} = \boldsymbol{i}$ s.t. $G_p(s) = \frac{y(s)}{u(s)} = \frac{k_m}{Js^2 + ds}$

1.) Start simple, try position feedback $\boldsymbol{u = P(u_c - y)}$

2.) Calculate c.l. poles

with: $G_p(s) = \frac{B(s)}{A(s)}$ and $G_c(s) = \frac{S(s)}{R(s)} = \frac{P}{1}$ and $G_{ff} = \frac{T(s)}{R(s)} = \frac{P}{1}$

Closed loop poles, $G_{yr} = \frac{BT}{AR + BS} = \frac{k_m P}{Js^2 + ds + Pk_m}$

-> 2:nd order poly. -> solve for $s_{1,2}$ from $Js^2 + ds + Pk_m = 0$.

3.2.2 Root locus

- **Plot the c.l. poles as a function of one variable**

  the variable could be either a control parameter or a process parameter

  Here we choose P as the varying variable. **Matlab code for the calculations:**

```
m = 0.08; d = 0.45; km = 3.6;
Pvec = 0:.01:1;
for n = 1:length(Pvec), Poles(:,n) = roots([J d Pvec(n)*km]);end
plot(real(Poles(1,:)),imag(Poles(1,:)),'r.',real(Poles(2,:)),imag(Poles(2,:)),'b.')
sgrid
```

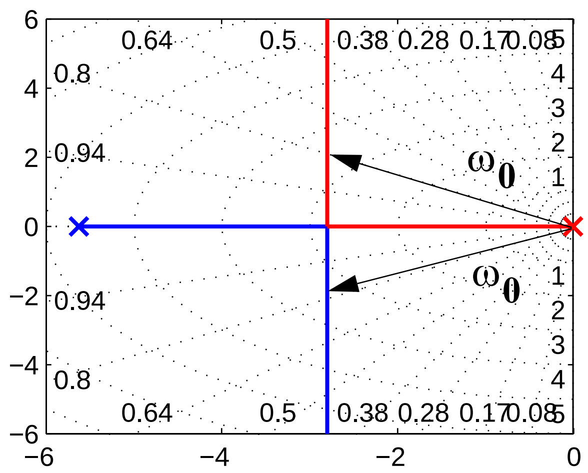
Normally we don't want to have a step response with an overshoot. A robot arm could collide with an object!

Choose fastest poles with $\zeta > 0.8$ -> $P = 0.27$

which gives $\omega_0 = 3.49$

Try:

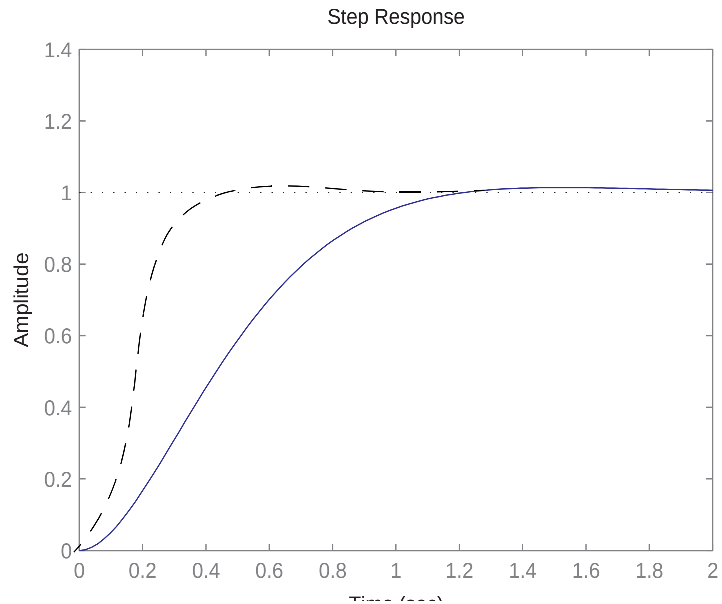
```
G = tf(km,[J d 0]); rlocus(G,Pvec),P = rlocfind(G)
```



3.2.3 Evaluate the closed loop

- **Closed loop** $G_{yr} = \dfrac{k_m P}{Js^2 + ds + k_m P} = \dfrac{(k_m P)/J}{s^2 + (d/J)s + (k_m P)/J} = \dfrac{\omega_0^2}{s^2 + 2\zeta\omega + \omega_0^2}$

  where $\omega_0 = 3.49$ and $\zeta = 0.806$



  But what if we want a faster step response, higher ω_0 with the same $\zeta > 0.8$??

  Try position and velocity feedback!

3.2.4 Position and velocity feedback

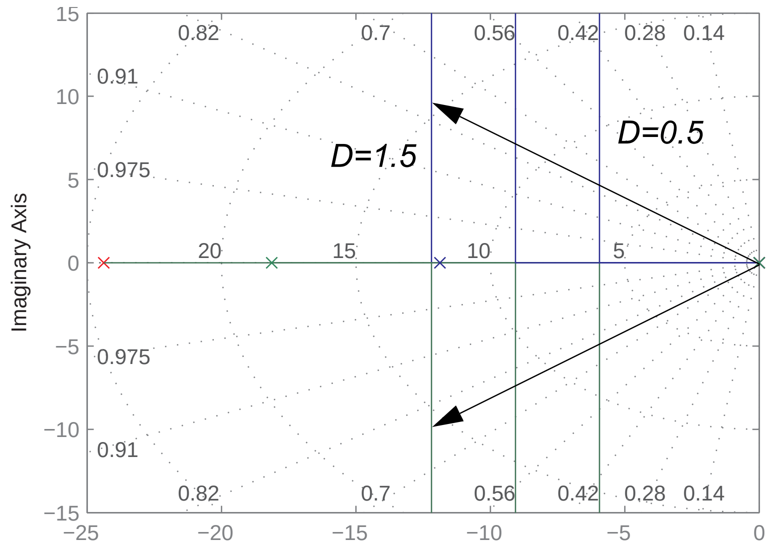
- New control law $u = -P(r-y) - D\frac{dy}{dt}$ -> $G_c(s) = \frac{S}{R} = \frac{Ds+P}{1}, G_{ff}(s) = \frac{T}{R} = \frac{P}{1}$

- How do we choose D? $G_{yu_c} = \frac{BT}{AR+BS} = \frac{k_m P}{Js^2 + (d + Dk_m)s + k_m P}$

  a root locus can not be done on both P and D at the same time lets try multiple root locuses on P with D ={0.5 1.0 1.5};

  With $D = 1.5$ we can choose $(\omega_0, \zeta) = [15, 0.82]$ which is 5 times faster than without velocity feedback with the same ζ.

  Is there a way to get any desired speed ω_0, and damping ζ ?



3.2.5 Instead of guessing D -use Poleplacement

- **Solving for s in the c.l. denominator polynomial with position and velocity feedback gives** $A_{cl} = AR + BS = s^2 + \frac{(d + k_m D)}{J}s + \frac{k_m}{J}P = 0$

  - 2 control parameters and a second order polynomial, that is, we can choose **any** c.l. poles by selecting P & D in a proper way such that.

$$A_{cl}(s) = A_m(s)$$

where A_m is the desired closed loop polynomial e.g. $A_m = s^2 + 2\zeta_m\omega_m s + \omega_m^2$.

This gives

$$\frac{(d + k_m D)}{J} = 2\zeta\omega_m \quad \text{->} \quad D = \frac{-d + 2\zeta\omega_m J}{k_m}$$

$$\frac{k_m}{J}P = \omega_m^2 \qquad P = \frac{\omega_m^2 J}{k_m}$$

Use *solve* in Maple

3.2.6 Maple

```
> restart;
> B:=km/J;A:=s^2+d/J*s;
```

$$B := \frac{km}{J}$$

$$A := s^2 + \frac{d\,s}{J}$$

```
> S:=D*s+P; R:=1;
```

$$S := \mathrm{D}\,s + P$$

$$R := 1$$

```
> Acl:=collect(A*R+B*S,s);
```

$$Acl := s^2 + \left(\frac{d}{J} + \frac{km\,\mathrm{D}}{J}\right) s + \frac{km\,P}{J}$$

```
> Am:=s^2+2*zeta*omega*s+omega^2;
```

$$Am := s^2 + 2\,\zeta\,\omega\,s + \omega^2$$

```
> solve({coeff(Acl,s,1)=coeff(Am,s,1),coeff(Acl,s,0)=coeff(Am,s,0)},{P,D});
```

$$\{\mathrm{D} = \frac{-d + 2\,\zeta\,\omega\,J}{km},\ P = \frac{\omega^2\,J}{km}\}$$

```
>
```

3.3. Lecture outline

- 1. Introduction
- 2. Model based Control, a motivating example
- **3. Pole placement design**
- 4. Discrete time approximation of the continuous time control
- 5. Example, a PD position controller
- 6. Example, a PD position controller
- 7. Cascaded motion control architecture

3.3.1 Poleplacement with s.s. models

Model: $\dot{x} = Ax + Bu$, where $x = [x_1, x_2, \ldots, x_n]^T$.

Control law: $u = -Lx + w$, where $L = [l_1, l_2, \ldots, l_n]$

Closed loop: $\dot{x} = Ax + Bu = Ax - BLx + Bw = (A - BL)x + Bw$

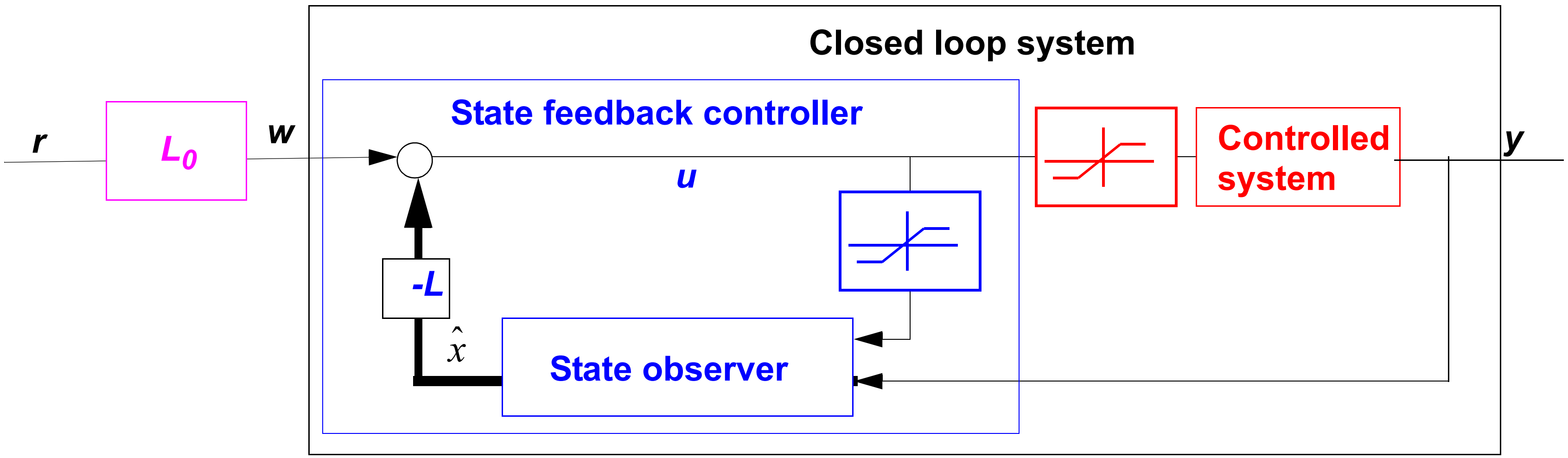
The poles of the c.l. are totally defined by *L*, eigenvalues of the matrix, $(A - BL)$
L is easiest found numerically in Matlab using the 'acker' command.

Advantage, easy to calculate *L* for for any model, also high order.

3.3.2 If the state vector is not measurable

The state vector *x* must be available from measurements or from designing a state observer.

$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) = (A - KC)\hat{x} + Bu + Ky$, design *K* in the same way as *L*.



3.3.3 Numeric solution in Matlab

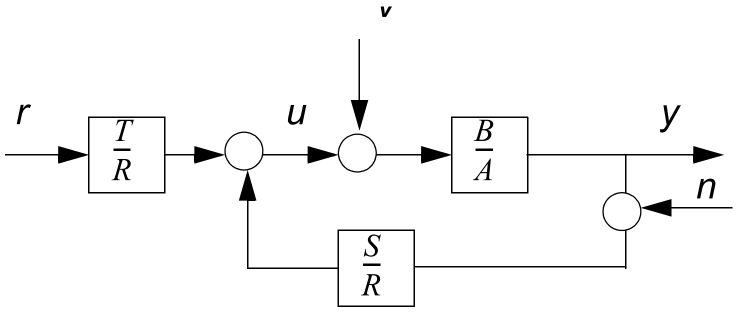
Calculate the state space model of the DC-motor, choose $x_1 = \varphi, x_2 = \dot{\varphi}$

which gives $A = \begin{bmatrix} 0 & 1 \\ 0 & -d/J \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ k_m/J \end{bmatrix}$ and $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

```
A = [0 1;0 -d/J]; B = [0;km/J]; C = [1 0]; D = 0; Gss = ss(A,B,C,D);
w0 = 20; zeta = 0.8;
poles = roots([1 2*zeta*w0 w0^2])
poles =
 -16.0000 +12.0000i
 -16.0000 -12.0000i
L = acker(Gss.a,Gss.b,poles)       % OBS L(1) =  P,  and L(2) = D
L =
    8.8889    0.5861
damp(A-B*L)
        Eigenvalue              Damping      Freq. (rad/s)

 -1.60e+001 + 1.20e+001i     8.00e-001      2.00e+001
 -1.60e+001 - 1.20e+001i     8.00e-001      2.00e+001
```

3.3.4 Poleplacement with T.F. models



1.) Select control structure, $\frac{S(s)}{R(s)}$.

2.) Calculate the c.l. polynomilal
$A_{cl}(s) = AR + BS$ (characteristic equation).

3.) Select a desired c.l. polynomial,
$A_d(s) = A_m(s)A_o(s)$ where $\deg(A_d) = \deg(A_{cl})$

and $\deg(A_m) = \deg(A)$, which gives $\deg(A_o) = \deg(A_d) - \deg(A_m)$.

4.) Solve for the parameters in $R(s)$ and $S(s)$ in the so called Diophantine eq.
$A_{cl}(s) = A_m(s)A_o(s)$.

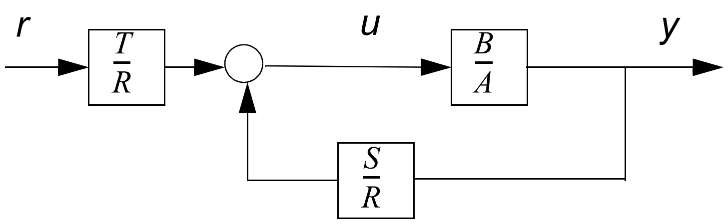
5.) Set the f.f. polynomial to $T(s) = t_0 A_o(s)$ where t_0 is a static gain that gives unit dc-gain in the c.l. T.F. from u_c to y.

3.3.5 Calculate the feedforward part T(s)

$$G_{yr}(s) = \frac{BT}{AR+BS} = \frac{BT}{A_{cl}} = \frac{Bt_0A_o}{A_mA_o} = \frac{Bt_0}{A_m}$$

, chose t_0 such that,

$$\frac{1}{t_0} = \left.\frac{B(s)}{A_m(s)}\right|_{s=0} = \frac{B(0)}{A_m(0)}.$$



i.) the order of $G_{yu_c}(s)$ is the same as $A_m(s)$ and thereby also the order of the process $A(s)$, (see last slide).

ii.) The dc gain is one, $G_{yu_c}(0) = 1$.

3.3.6 Example, position control

$$G_p(s) = \frac{k_m}{Js^2 + ds} = \frac{3.6}{0.1s^2 + 0.45s} = \frac{36}{s^2 + 4.5s}$$

$J = 0.1$: rotor inertia

$k_m = 3.6$: torque constant

$d = 0.45$: friction coefficient

1.) PD-control with l.p. filter structure $\frac{S(s)}{R(s)} = \frac{s_1 s + s_0}{s + r_0}$

2.) C.l polynomial $A_{cl} = AR + BS = s^3 + s^2(4.5 + r_0) + s(4.5r_0 + 36s_1) + 36s_0$

3.) Select desired C.l polynomial

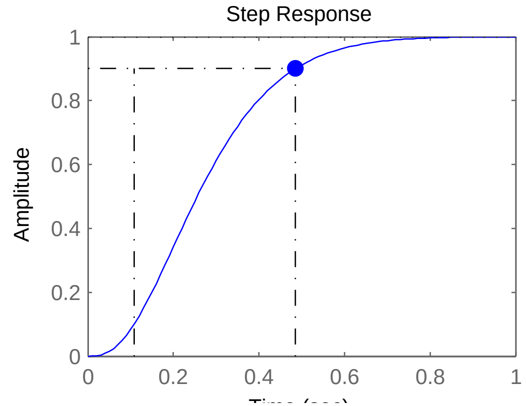
$$A_d = A_m A_o = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$$

From some specifications we want:

Rise time should be less than 0.5 s.

One possible selection of the

c.l. poles is $\omega = 10, \zeta = 0.9, \alpha = 10$



Cont.

4.) Diophantine equation $A_{cl} = A_d$

$$s^3 + s^2(4.5 + r_0) + s(4.5r_0 + 36s_1) + 36s_0 = s^3 + s^2(\alpha + 2\zeta\omega) + s(2\zeta\omega\alpha + \omega^2) + \omega^2\alpha$$

gives:

$$4.5 + r_0 = 28$$
$$4.5r_0 + 36s_1 = 280$$
$$36s_0 = 1000$$

$$r_0 = 23.5$$
$$s_0 = 27.8$$
$$s_1 = 4.8$$

$\longrightarrow$ $$\frac{S}{R} = \frac{4.8s + 27.8}{s + 23.5}$$

Check! roots of :

$$AR + BS = 0$$

5.)

Feed forward part $T(s) = A_o t_0$

T.F from reference to output $G_{yr}(s) = \frac{BT}{AR + BS} = \frac{Bt_0}{A_m} = \frac{36t_0}{s^2 + 28s + 100}$

calculate $t_0 = \frac{100}{36}$ and $T(s) = 2.8s + 28$

gives $G_{yr}(s) = \frac{100}{s^2 + 28s + 100}$

with the control law $u(s) = \frac{T}{R}r - \frac{S}{R}y = \frac{2.8s + 28}{s + 23.5}r - \frac{4.8s + 27.8}{s + 23.5}y$

3.3.7 The choice of S&R

- Normally we need the order of $R(s)$ to be at least the same as for $S(s)$. This gives a proper t.f. $G_c(s) = \frac{S}{R}$. (the order of the numerator is not higher than that of the denominator).

- PD type controllers can however be used. (derivation of position to velocity)

- A time delay of at least one sample will be introduced if the order of $R(s)$ is higher then $S(s)$.

- A good choice is thereby to have the same order of $S(s)$ and $R(s)$, and if the order of $S(s)$ is one less than $A(s)$ then complete control in terms of poles and their c.l. locations is possible.

- Which order is then good to use? -depends on the control problem such as: Integral control, sensor noise, disturbances etc.

3.3.8 PI type feedback for velocity control

Process: $G_p = \frac{B}{A} = \frac{b}{s+a}$, and the Diophantine exp. $AR + BS = A_m A_o$

P-ctrl. $G_c = \frac{S}{R} = \frac{s_0}{1}$ Dio. $s + a + s_0 = s + \alpha \quad (A_o = 1)$

P-ctrl. with LP-filter. $G_c = \frac{S}{R} = \frac{s_0}{s + r_0}$ Dio. $s^2 + (r_0 + a)s + (bs_0 + ar_0) = (s+\alpha)(s+\beta)$

PI-ctrl. $G_c = \frac{S}{R} = \frac{s_1 s + s_0}{s}$, Dio. $s^2 + (a + bs_1)s + bs_0 = (s+\alpha)(s+\beta)$

PI-ctrl. with LP-filt.

$G_c = \frac{S}{R} = \frac{s_1 s + s_0}{s(s + r_0)}$, Dio. $s^3 + (a + r_0)s^2 + (bs_1 + ar_0)s + bs_0 = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$

3.3.9 PID type feedback for position control

Process: $G_p = \frac{B}{A} = \frac{b}{(s+a)s}$ c.l. dynamics $AR + BS$ and the Diophantine exp.

PD-ctrl. $G_c = \frac{S}{R} = \frac{s_1 s + s_0}{1}$ Dio. $s^2 + (bs_1 + a)s + bs_0 = s^2 + 2\zeta\omega s + \omega^2$

PD-ctrl. with LP-filter. $G_c = \frac{S}{R} = \frac{s_1 s + s_0}{s + r_0}$

Dio. $s^3 + (a + r_0)s^2 + (bs_1 + ar_0)s + bs_0 = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$

PID-ctrl. $G_c = \frac{S}{R} = \frac{s_2 s^2 + s_1 s + s_0}{s}$,

Dio. $s^3 + (a + bs_2)s^2 + bs_1 s + bs_0 = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$

PI-ctrl. $G_c = \frac{S}{R} = \frac{s_1 s + s_0}{s}$,

Dio. $s^3 + as^2 + bs_1 s + bs_0 = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$

(not solvable! 3:rd order Dio. with only two control parameters)

PI-ctrl. with LP-filt. $G_c = \frac{S}{R} = \frac{s_1 s + s_0}{s(s + r_0)}$, (observe $\deg R > \deg S$)

Dio. $s^3 + (a + r_0)s^2 + (bs_1 + ar_0)s + bs_0 = (s^2 + 2\zeta\omega s + \omega^2)(s + \alpha)$

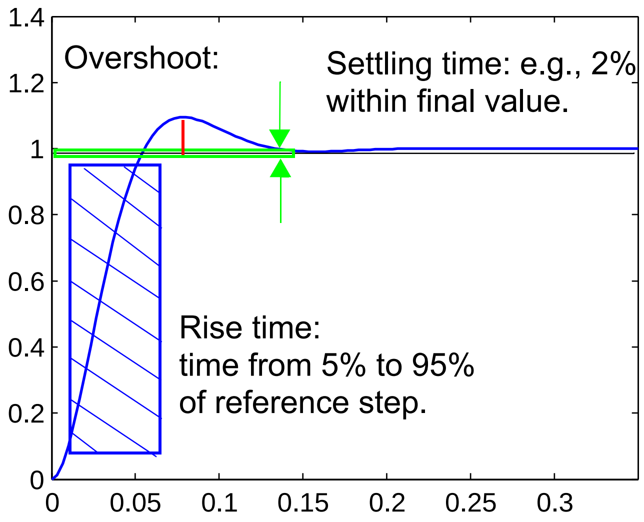
PID-ctrl. with LP-filt. $G_c = \frac{S}{R} = \frac{s_2 s^2 + s_1 s + s_0}{s(s + r_0)}$

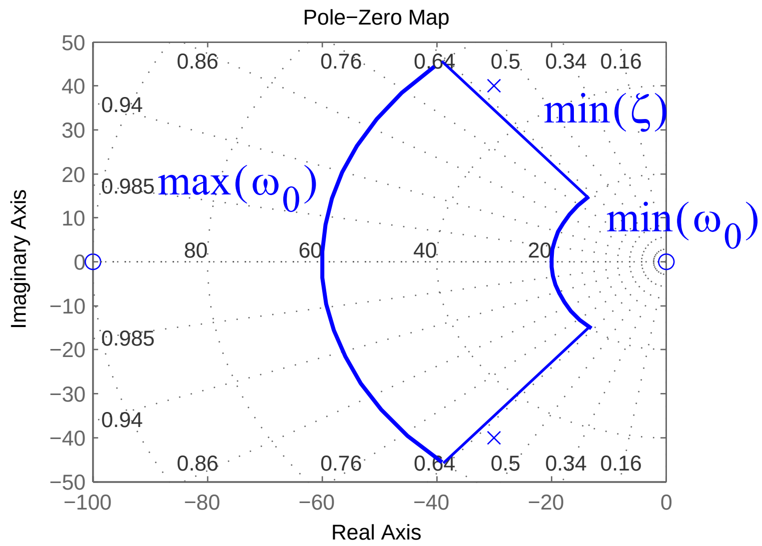
Dio. $s^4 + (r_0 + a)s^3 + (bs_2 + ar_0)s^2 + bs_1 s + bs_0 = (s^2 + 2\zeta_1\omega_1 s + \omega_1^2)(s^2 + 2\zeta_2\omega_2 s + \omega_2^2)$

3.4. Lecture outline

3.4.1 Specifications for poleplacement

- **Intuitively in time domain, but for design in complex plane.**
- **Need for translation between planes**





select c.l. poles:

uneven order process: $(s + \omega_0)$

even order process: $(s^2 + 2\zeta\omega_0 s + \omega_0^2)$

3.4.2 Observe the order of the c.l. system

- The rise time for higer order systems will be slower (superposition)

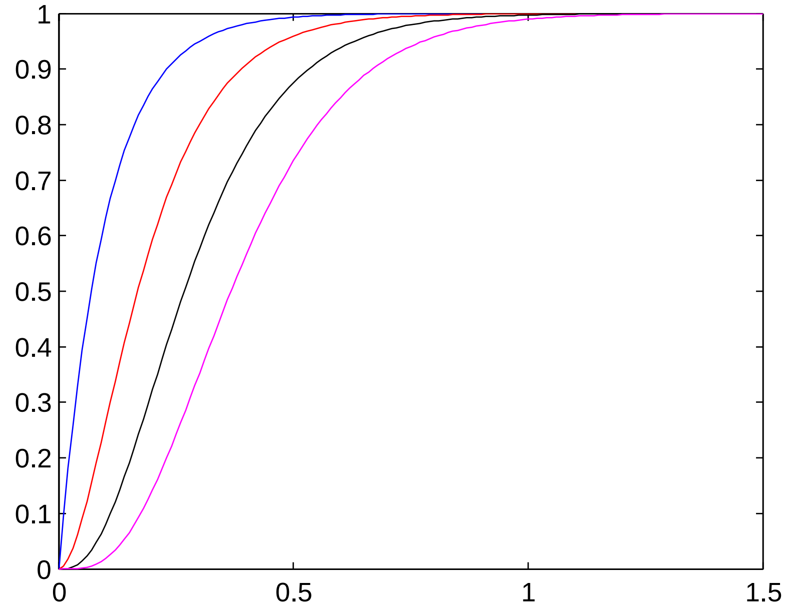
$$\frac{\omega}{s+\omega}$$

$$\frac{\omega^2}{s^2+2\zeta\omega s+\omega^2}$$

$$\frac{\omega^3}{(s^2+2\zeta\omega s+\omega^2)(s+\omega)}$$

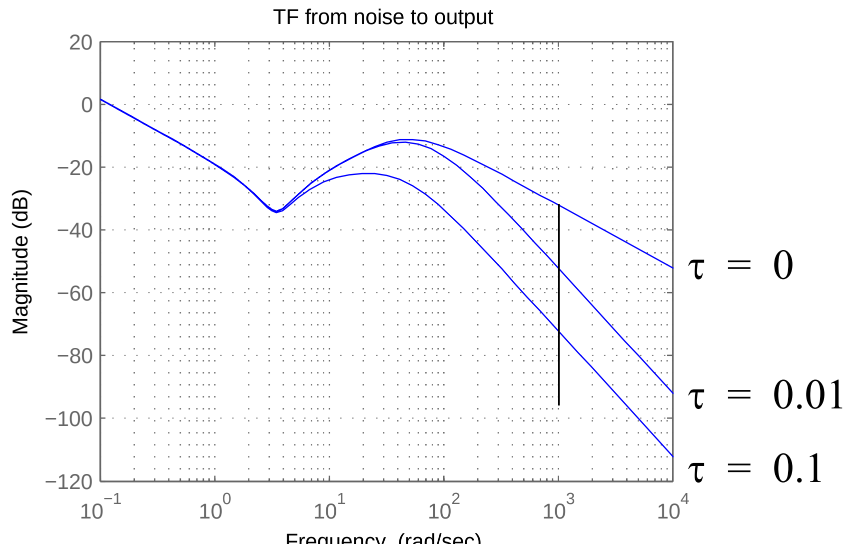
$$\frac{\omega^4}{(s^2+2\zeta\omega s+\omega^2)(s^2+2\zeta\omega s+\omega^2)}$$

$\omega = 10, \zeta = 1$ for all models



3.4.3 Frequency domain specifications

- Example, gain from sensor noise at 50 Hz to output must be less than ()



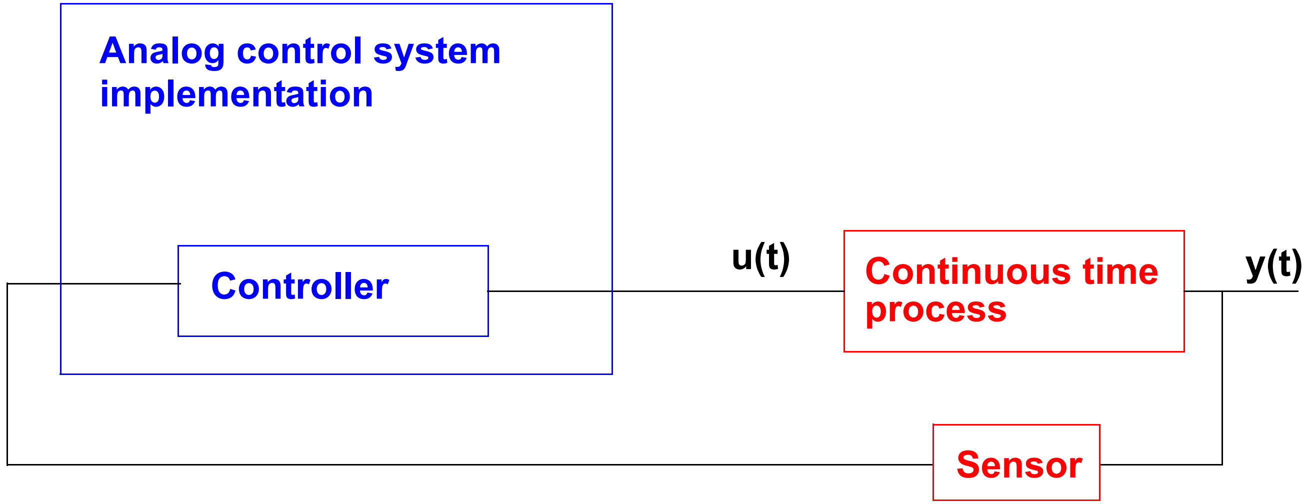
- **Example of a specification on t.f. from noise to output**

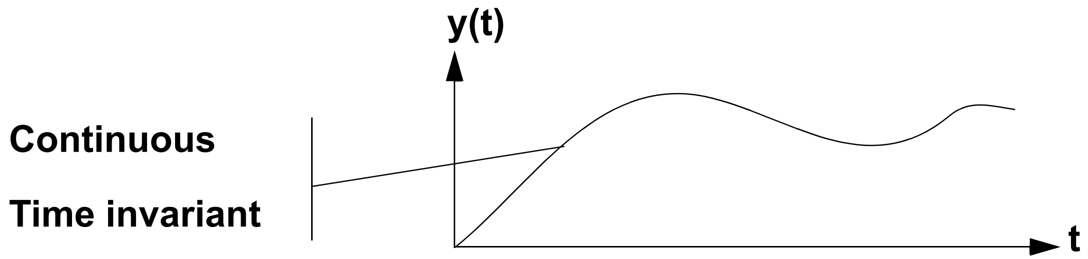
$$G_{yn}(j50) < -40\,\text{db}$$

3.5. Lecture outline

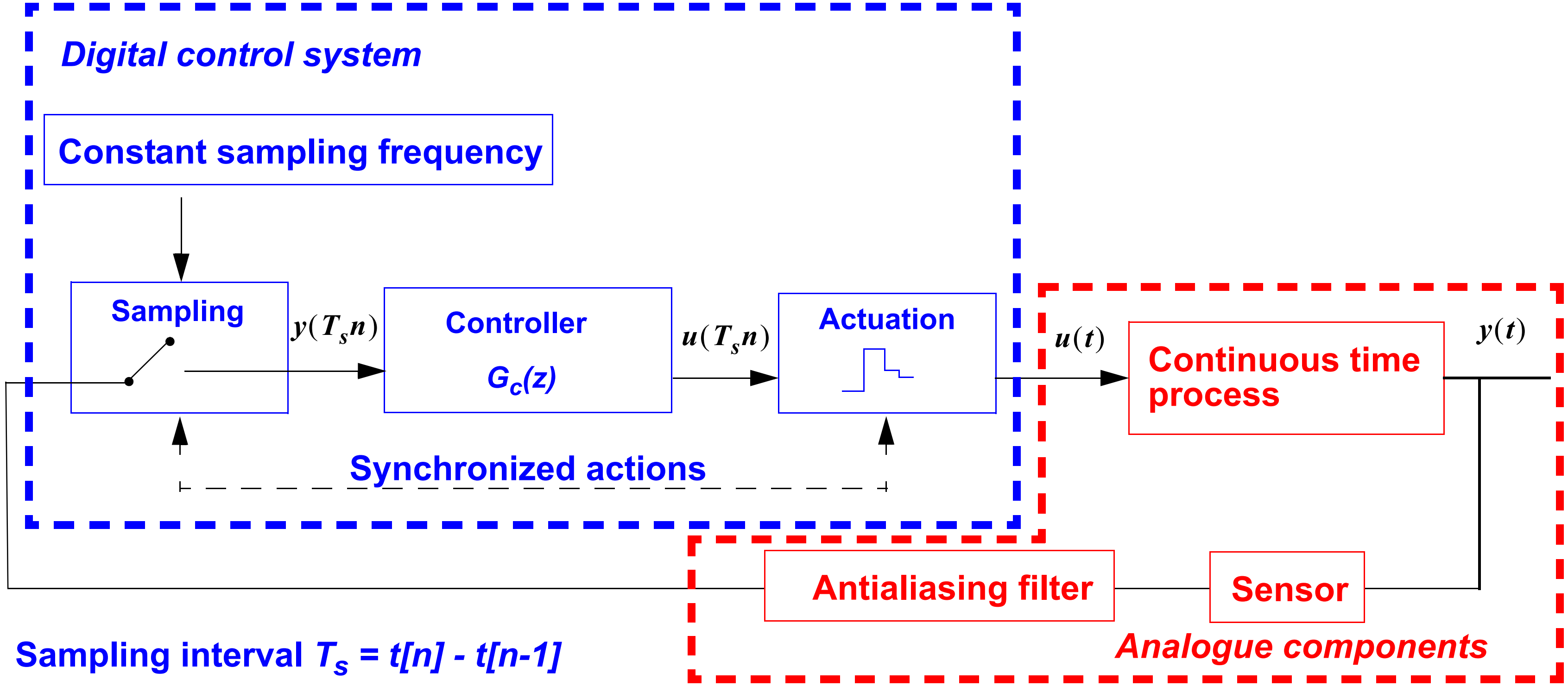
- 1. Introduction
- 2. Model based Control, a motivating example
- 3. Pole placement design
- 4. Specifications for poleplacement design
- **5. Discrete time approximation of the continuous time control**
- 6. Example, a PD position controller
- 7. Cascaded motion control architecture

3.5.1 Continuous vs discrete implementation





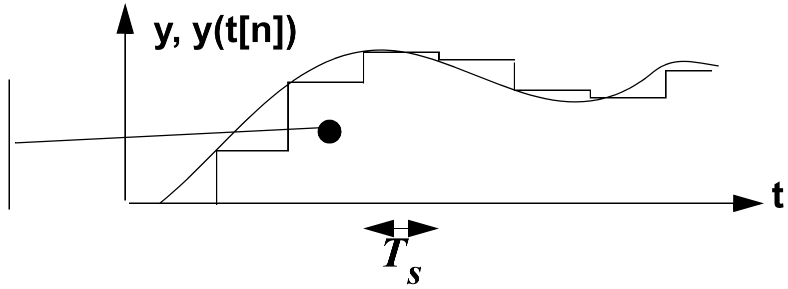
3.5.2 Discrete time control



Sampling interval $T_s = t[n] - t[n-1]$

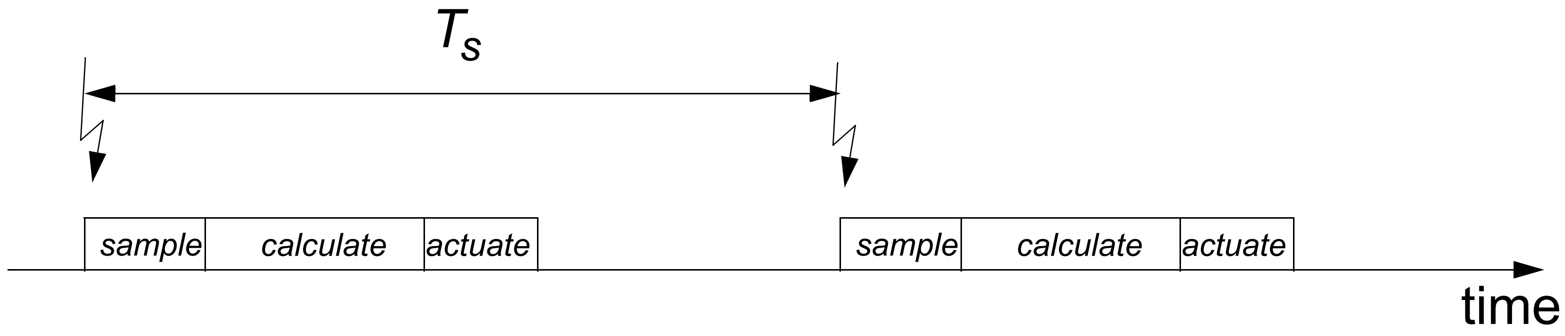
The control signal is typically constant over the sampling interval, ZOH.

Discrete signal
Delayed signal
Time varying



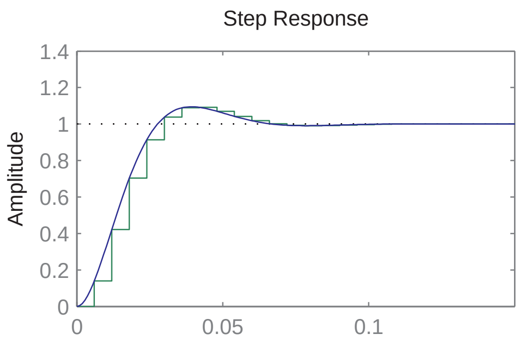
3.5.3 Assumptions/consequences

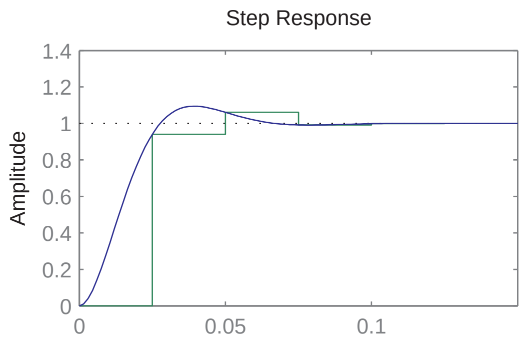
- sampling at constant frequency (constant sampling interval)
- synchronism between sampling and actuation
- zero delay between sampling and actuation (clearly we can not achieve this exactly, execution of the control algorithm takes time)



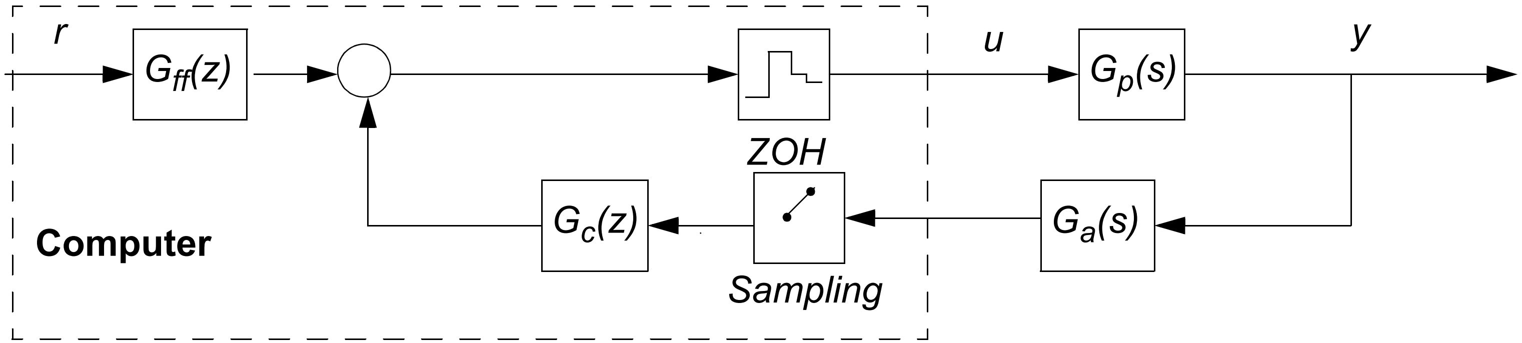
3.5.4 Choice of sampling rates from time domain

- single rate systems
  - high sampling rate is costly
  - the frequency should be set in relation to the fastest dynamics in the closed loop characteristics (i.e. bandwidth, rise-time) of the feedback, observer or model following.
  - or 4-10 samples per rise time





3.5.5 Sampling interval selection based on freq.

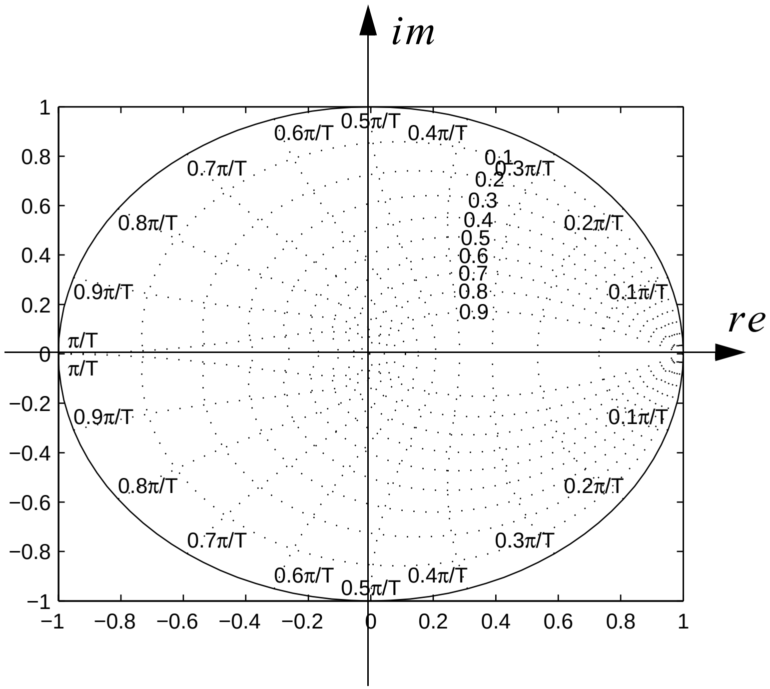


- **The sampling frequency must be faster then the fastest dynamic mode in the control system, which could be either:**
  - in the feedback $\frac{S}{R}$, in the feedforward $\frac{T}{R}$ or in the closed loop $AR + BS$. It can also be taken from the bandwidth or crossover frequency of the controller.
  - If the fastest pole is ω_b then the sampling frequency should be $\omega_s = [10\ldots30]\omega_b$ and thereby sampling time $T_s = \frac{2\pi}{\omega_s}$

3.5.6 Mapping the s-plane to the z-plane

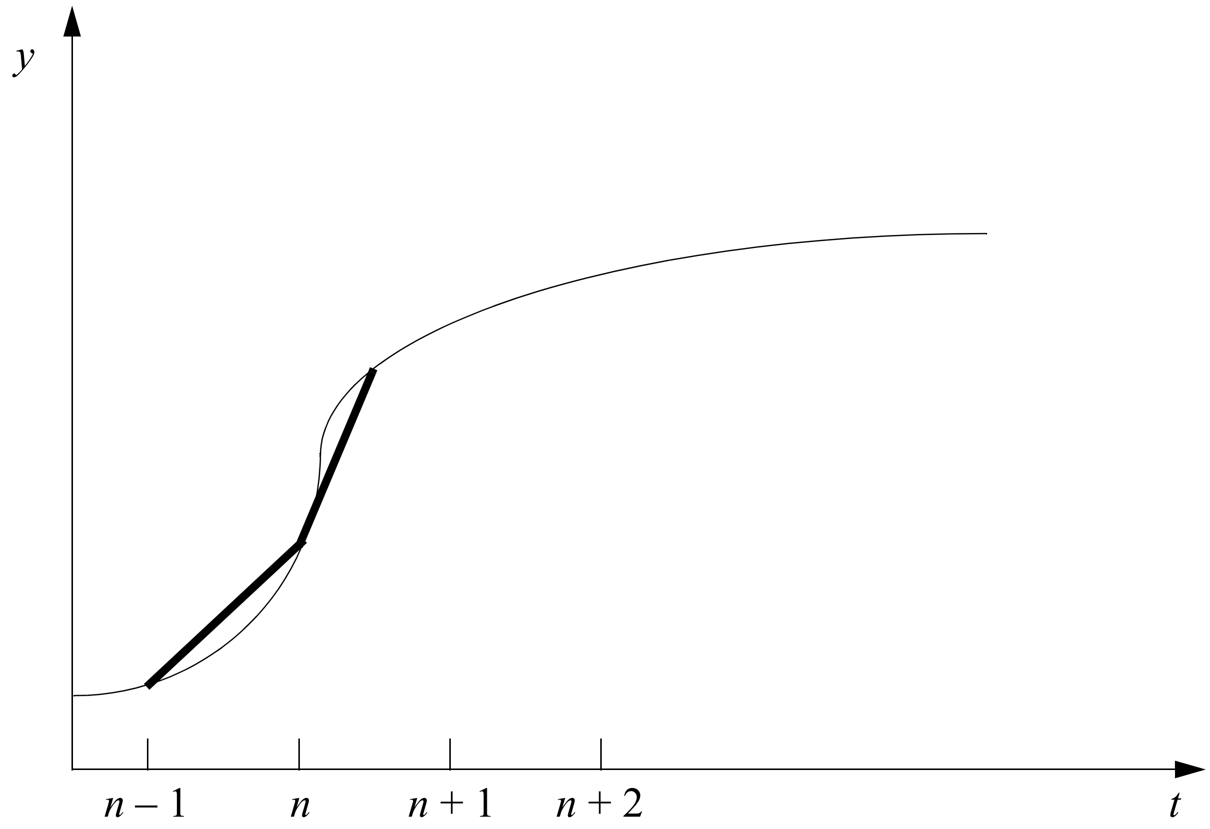
- **Poles**

A continuous time pole $s = a + bi$ is mapped to a discrete time pole by $z = e^{sT_s}$ where T_s is the sampling period.
(From the definition of the z-transform, lecture 2)



The continuous time stability border $s = j\omega$, $\omega = [-\infty, \infty]$ is
$z = e^{j\omega T_s} = \cos(\omega T_s) + i\sin(\omega T_s)$
which is the unit circle.

3.5.7 Approximating the derivative



3.5.8 Transformation of continuous time design

- Forward difference approximation (Euler's method)

$$sx = \frac{dx(t)}{dt} \approx \frac{x(t+T_s) - x(t)}{T_s} = \frac{z-1}{T_s}x(t)$$

- Backward difference

$$sx = \frac{dx(t)}{dt} \approx \frac{x(t) - x(t-T_s)}{T_s} = \frac{1-z^{-1}}{T_s}x(z) = \frac{z-1}{zT_s}x(z)$$

- Tustins approximation, (bilinear transformation), (Trapezoidal method)

$$sx = \frac{dx(t)}{dt} \approx \frac{2}{T_s}\frac{x(t+T_s)-1}{x(t+T_s)+1} = \frac{2(z-1)}{T_s(z+1)}x(z)$$

3.5.9 Using the approximation

Ex. for a PID controller with Euler forward

$$\frac{S(s)}{R(s)} = \frac{s_2 s^2 + s_1 s + s_0}{s(s + r_0)} \quad \longrightarrow \quad \frac{S(z)}{R(z)} \approx \frac{s_2\left(\frac{z-1}{T_s}\right)^2 + s_1\frac{z-1}{T_s} + s_0}{\frac{z-1}{T_s}\left(\frac{z-1}{T_s} + r_0\right)}$$

Use Maple!

Tustin is availabe for numeric approximation in Matlab, Control Toolbox

3.5.10 Convinient in s.s. format

Euler forward

$$\dot{x} \approx \frac{x[n+1] - x[n]}{T_s} = Ax[n] + Bu[n]$$

$$y = Cx[n]$$

$$x[n+1] = x[n] + T_s Ax[n] + T_s Bu[n]$$
$$x[n] = (1 + T_s A)x[n-1] + T_s Bu[n-1]$$
$$y[n] = Cx[n]$$

Delay

Euler backward

$$\dot{x} \approx \frac{x[n] - x[n-1]}{T_s} = Ax[n] + Bu[n]$$

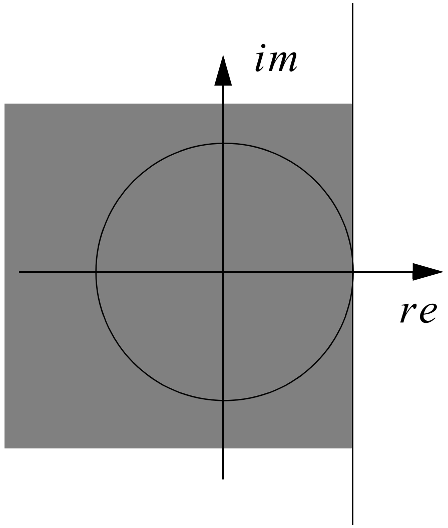
$$y = Cx[n]$$

$$x[n] = x[n-1] + T_s Ax[n] + T_s Bu[n]$$
$$x[n] = (1 + T_s A)^{-1} x[n-1] + (1 + T_s A)^{-1} Bu[n]$$

No delay

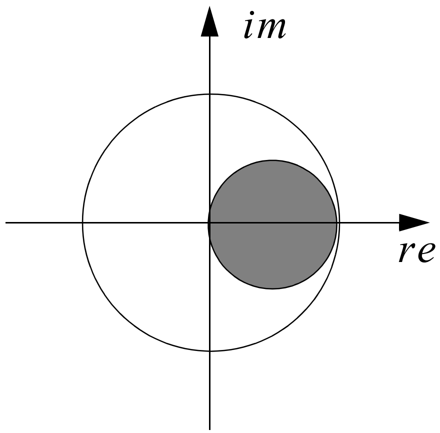
3.5.11 Mapping of poles

- The stability region in the continuous time case (left half plane) corresponds to the unit circle in the discrete time case.



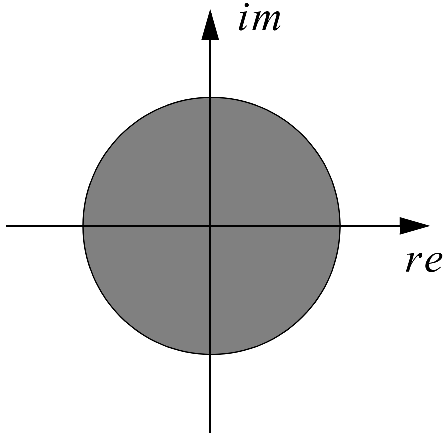
Forward difference

$$z = e^{sT_s} \approx 1 + sT_s$$



Backward difference

$$z = e^{sT_s} \approx \frac{1}{1 - sT_s}$$



Tustin

$$z = e^{sT_s} \approx \frac{1 + (sT_s)/2}{1 - (sT_s)/2}$$

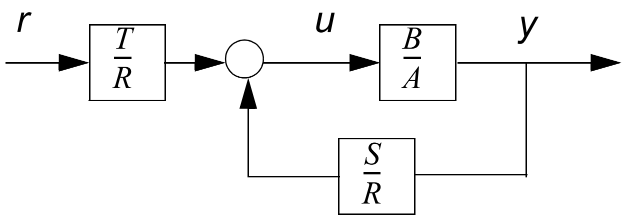
3.5.12 Evaluating the approximation

- **Compare simulated step response (in Simulink)**
  - 1.) With continuos prosess model and continuous controller
  - 2.) With continuous process model discrete time controller.
- **Compare the phase and amplitude margins**
  - 1.) With continuos prosess model and continuous controller
  - 2.) With a zoh model of the process and the discrete time controller
  - It is not possible to make a bode or nyquist plot in matlab for a combined continuous and discrete time model.

3.5.13 Summary

- The continuos time controller is

$$u(s) = \frac{T(s)}{R(s)}r - \frac{S(s)}{R(s)}y \ .$$



- After a discrete time approximation we have

$$u(z) = \frac{T(z)}{R(z)}r - \frac{S(z)}{R(z)}y$$

- Select the sampling period $[10, 30]$ times faster than the c.l. poles. (Observe that the poles are in *rad/s*)

- Use Tustins approximation in Matlab, *i)* for the feedback part $G_c(s) = \frac{S(s)}{R(s)}$ and, *ii)* for the feedforward part $G_{ff}(s) = \frac{T(s)}{R(s)}$ separatly.

3.6. Lecture outline

- 1. Introduction
- 2. Model based Control, a motivating example
- 3. Pole placement design
- 4. Specifications for poleplacement design
- 5. Discrete time approximation of the continuous time control
- **6. Example, a PD position controller**
- 7. Cascaded motion control architecture

3.6.1 Position control with PD controller

Process: $G_p(s) = \frac{B(s)}{A(s)} = \frac{K_t/J}{s(s+d/J)}$, with current as input!

PD-controller with L.P. filter $G_c(s) = \frac{S(s)}{R(s)} = \frac{s_1 s + s_0}{s + r0}$ -> $AR + BS$, third order

c.l. poles, specification $A_m(s)A_0(s) = (s^2 + 2\zeta\omega_1 s + \omega^2)(s + \omega_2)$

Calculate $\{s_1, s_0, r_0\}$ by solving

$$s^3 + \left(\frac{d}{J} + r_0\right)s^2 + \left(\frac{dr_0}{J} + \frac{K_t s_1}{J}\right)s + \frac{K_t s_0}{J} = (s^2 + 2\zeta\omega_1 s + \omega^2)(s + \omega_2)$$, with $\omega_1 = \omega_2 = \omega$

$$\{s1 = \frac{\omega^2 J^2 + 2\zeta\omega^2 J^2 - 2d\zeta\omega J - d\omega J + d^2}{J\,Kt}, r0 = \frac{2\zeta\omega J + \omega J - d}{J}, s0 = \frac{\omega^3 J}{Kt}\}$$

Based on specifications, choose $\omega = 50$ and $\zeta = 0.8$, which gives

$$\frac{S(s)}{R(s)} = \frac{138.8s + 2778}{s + 134.4} \text{ and } \frac{T(s)}{R(s)} = \frac{t_0 A_o}{R} = \frac{55.56s + 2778}{s + 134.4}$$

Approximate a discrete time implementation with e.g. Tustin, select the sampling period from rule of thumb $T_s = \frac{2\pi}{20\omega} \approx 0.006$.

$$\frac{S(z)}{R(z)} = \frac{104.8z - 92.9}{z - 0.43}, \quad \frac{T(z)}{R(z)} = \frac{45.5z - 33.7}{z - 0.43}$$

Control law: $R(z)u(z) = T(z)r(z) - S(z)y(z)$

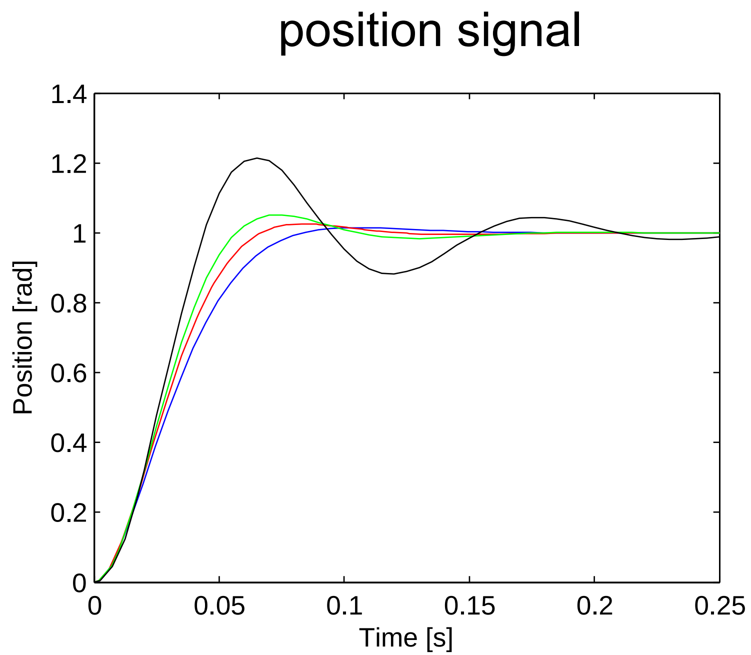
$(z - 0.43)u = (45.5z - 33.7)r - (104.8z - 92.9)y$ shift with z^{-1} gives the control,

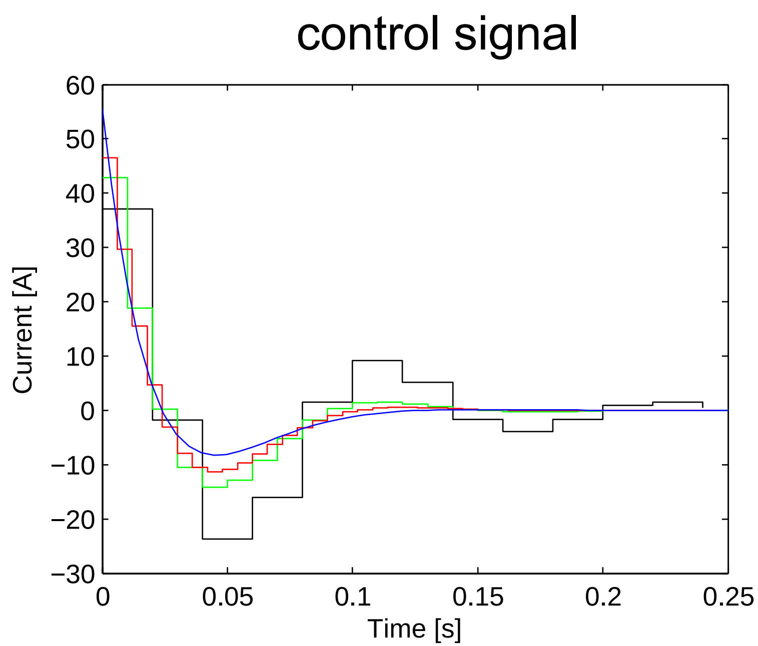
$$u[n] = 0.43u[n-1] + 45.5r[n] - 33.7r[n-1] - 104.8y[n] + 92.9y[n-1]$$

3.6.2 Results in time domain

Simulated step response:

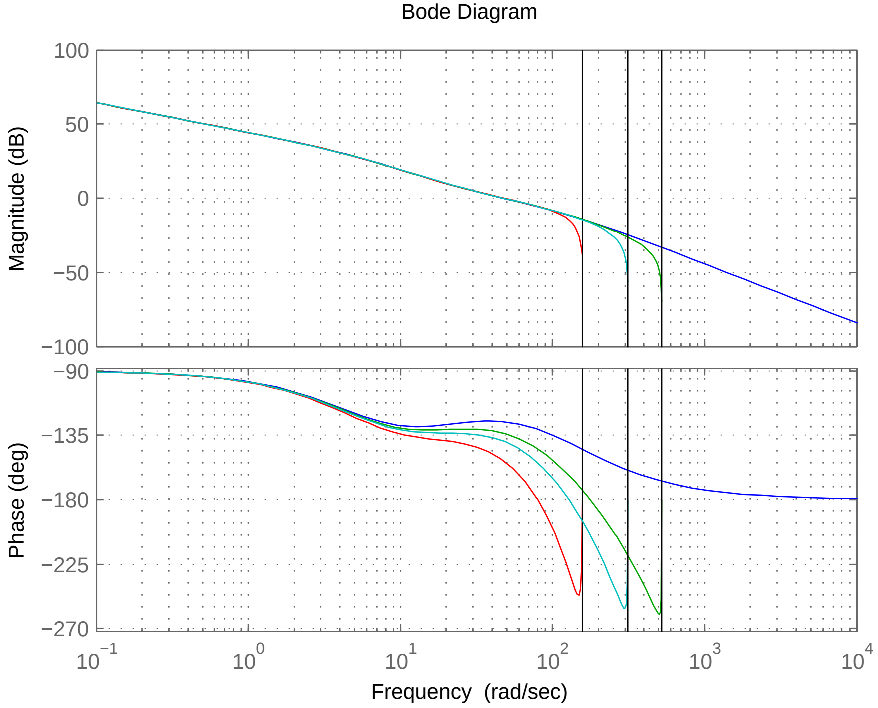
blue line, continuos time controller
red line, discrete time controller with Ts = 6 ms
green line, discrete time controller with Ts = 10 ms
red line, discrete time controller with Ts = 20 ms





3.6.3 Results in frequency domain

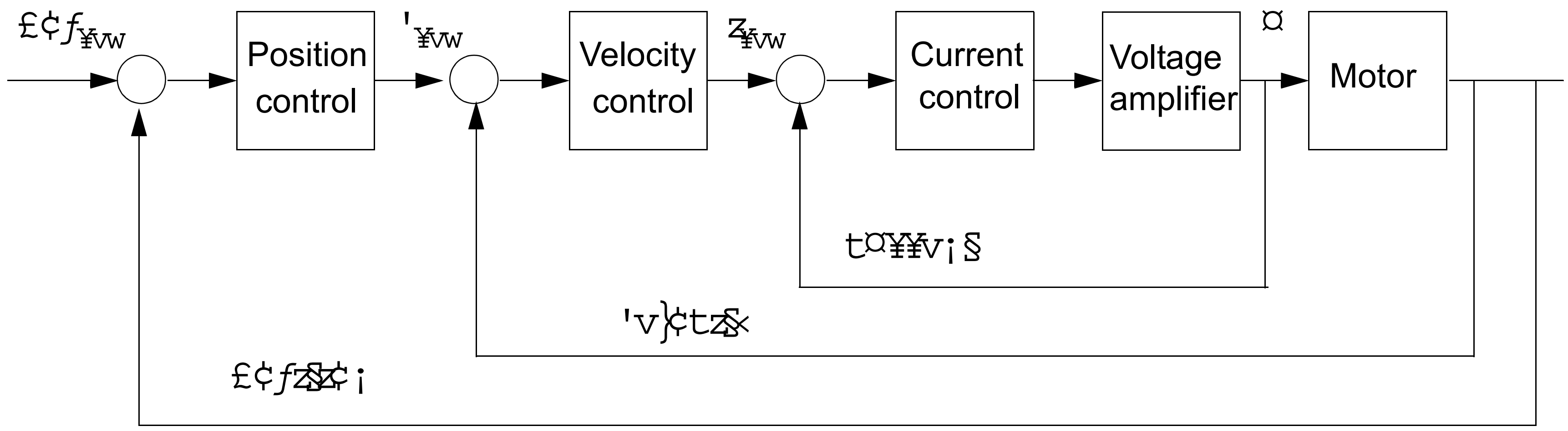
Model/Margin	Phase	Amplitude
Continuous	inf	54
Disc. 6 ms	46	16
Disc. 10 ms	41	12
Disc. 20 ms	27	5.6



3.7. Lecture outline

- 1. Introduction
- 2. Model based Control, a motivating example
- 3. Pole placement design
- 4. Specifications for poleplacement design
- 5. Discrete time approximation of the continuous time control
- 6. Example, a PD position controller
- **7. Cascaded motion control architecture**

3.7.1 Cascaded motion controller



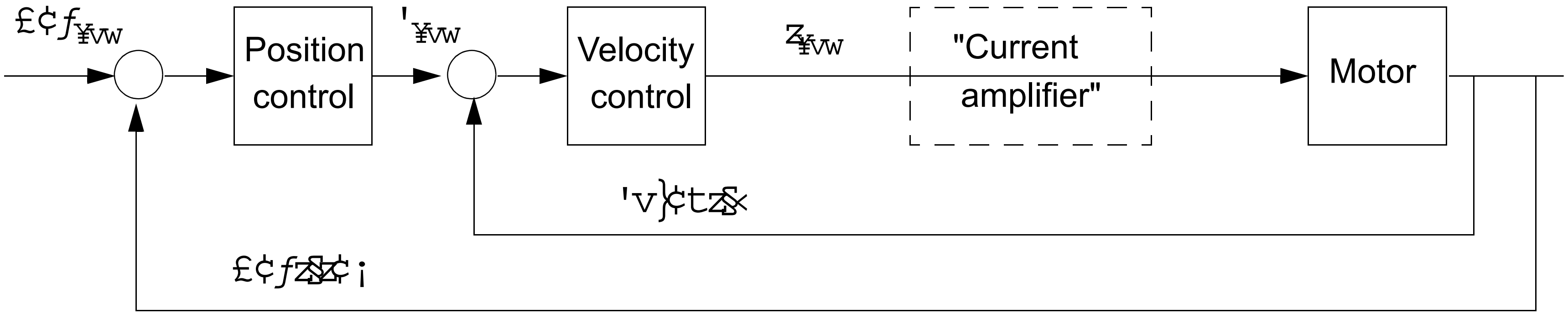
- Design each of the three controllers separately
- Start with the inner current loop
- Then velocity and last position
- The frequency range must be different for each loop

3.7.2 Current loop

- The model $u = R_e i + L\frac{di}{dt} + k_{emk}v$ -> $i = \frac{1}{R+Ls}u - \frac{1}{R+Ls}v$
- The velocity v acts as a disturbance -> the controller needs an integrator
- The control structure $C_i = \frac{Ps+I}{s}$
- The closed loop poles $AR + BS = s^2 + (R_e/L + P/L)s + I/L$
- Desired closed loop poles: real $(s+\omega_1)(s+\omega_2)$, or imag $s^2 + 2\zeta\omega s + \omega^2$

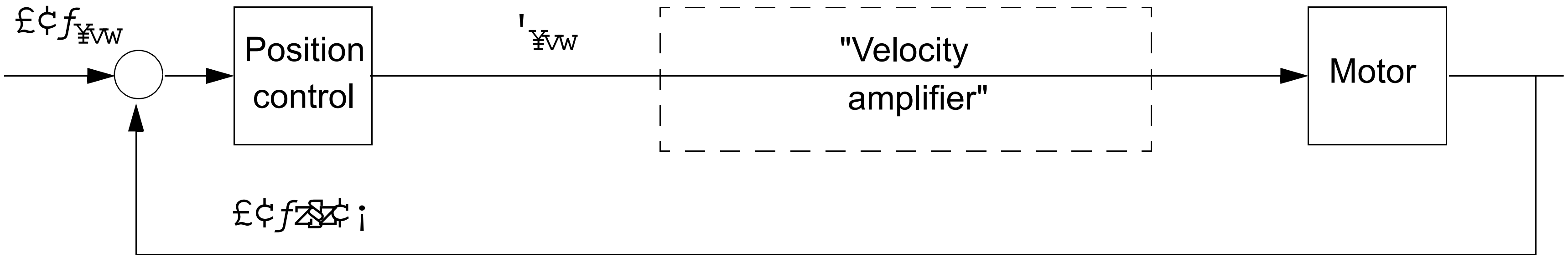
- The time constant L/R_e is normally very fast -> very fast closed loop poles
- Very small sampling period -> difficult with digital implementation
- Often Analogue controller in driver
- The current is not so easy to measure, often noisy

3.7.3 Velocity loop



- The "current amplifier" has T.F. 1 in the frequency range where the velocity controller should be designed.
- Model $v(s) = \frac{k_T}{Js + d} i(s)$
- PI-control is normally sufficient, c.l. poles at around 10 times slower then the current poles.
- The velocity must be measured or derivated from position

3.7.4 Position loop



- The "velocity amplifier" has T.F. 1 in the frequency range where the velocity controller should be designed.
- Model $pos(s) = \frac{1}{s} v(s)$
- P-control is normally sufficient, c.l. poles at around 10 times slower then the velocity poles.

3.7.5 Ex. cascaded control - current loop

- solve the diofantine equation for the controller $\frac{S}{R} = \frac{Ps + I}{s}$ and $\frac{B}{A} = \frac{1}{R + Ls}$

$$AR + BS = s^2 + (R_e/L + P/L)s + I/L = s^2 + (\omega_1 + \omega_2)s + \omega_1\omega_2$$

Control parameters: $P = L(\omega_1 + \omega_2) - R$, $I = L\omega_1\omega_2$

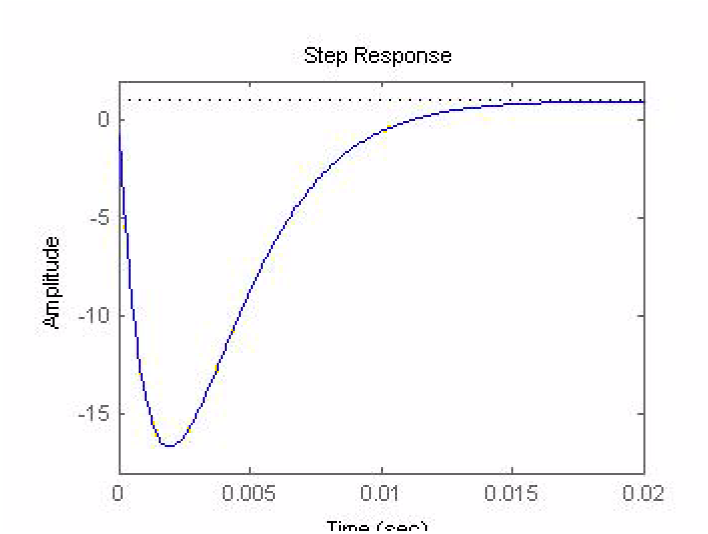
Motor data: $R = 24\Omega$, $L = 1.0\text{mF}$

- Lets try c.l. poles: $\omega_1 = \omega_2 = 500\text{rad/s}$

- For error feedback we get: $\frac{S}{R} = \frac{-23s + 250}{s}$

- Observe! we have c.l. zeros at $BS = 0$ gives a positive zero $s = 10.9$

- How does the step response look like ?



Not very good!

To get negative zeros and if:

$$\omega_1 = \omega_2 = \omega$$

then

$$BS = (2\omega L - R)s + \omega^2 L = 0$$

$$s = \frac{\omega^2 L}{R - 2\omega L} < 0 \quad \text{gives}$$

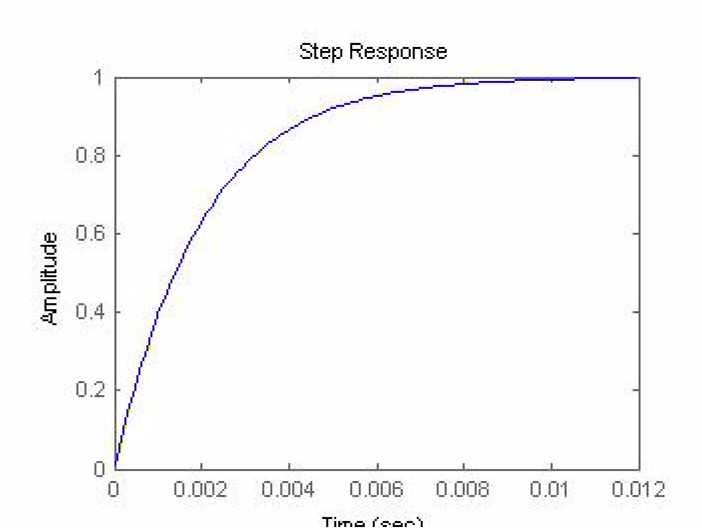
$$\omega > \frac{R}{2L} = \frac{24}{0.002} = 12000$$

- With a c.l. pole at -12000 we need sampling period of approximately 25 μs which is only possible with extremly high performing processors.

3.7.6 Example curren loop cont.

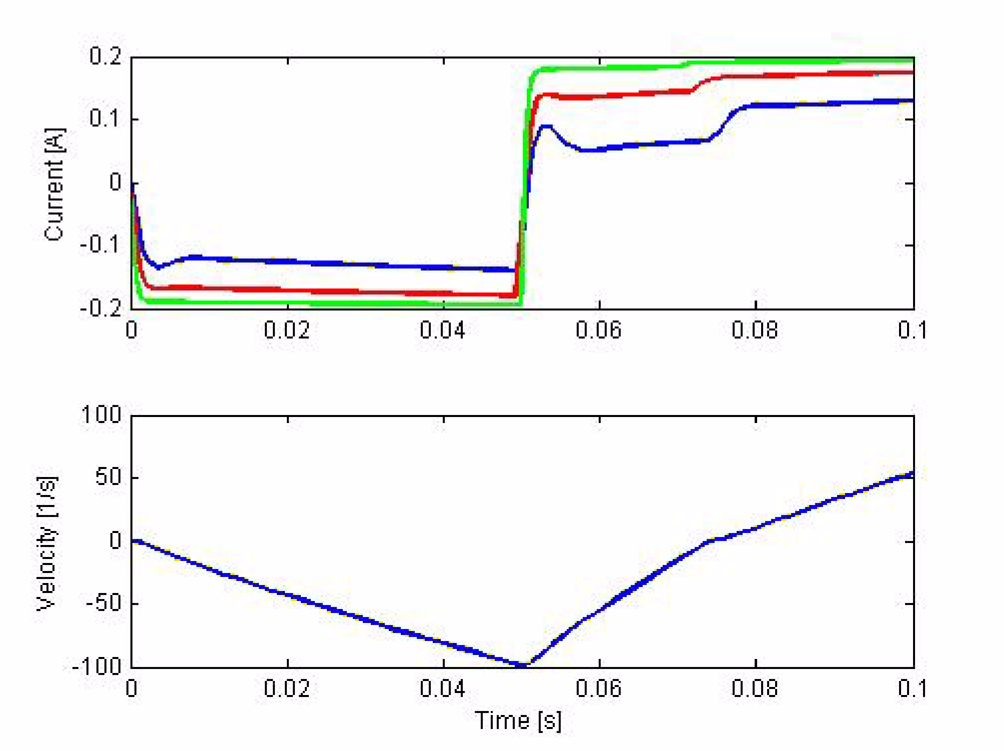
- **Instead do a 2 DOF design:** $u = \frac{T}{R}r - \frac{S}{R}i$

select $T = (s + \omega_2)$ and $t_0 = \omega_1$ gives: $\frac{BT}{AR + BS} = \frac{\omega_1}{(s + \omega_1)}$



3.7.7 Evaluate current response

- **The velocity acts as a disturbance:** $i = \frac{1}{R + Ls}u - \frac{1}{R + Ls}v$



Blue line: $\omega = 500$

Red line: $\omega = 1000$

Green line: $\omega = 2000$

3.7.8 Ex. cont. next step velocity loop

- **The model for designing the velocity loop is** $v(s) = \frac{k_T}{Js+d} i(s)$

There is no integrator in the model, select the control structur $\frac{S}{R} = \frac{Ps+I}{s}$

solve the diophantine equation:

$$AR + BS = s^2 + \left(\frac{d}{J} + \frac{k_T P}{J}\right)s + \frac{k_T I}{J} = (s^2 + (\omega_1 + \omega_2)s + \omega_1\omega_2)$$ gives:

$$P = \frac{(\omega_1 + \omega_2)J - d}{k_T}$$

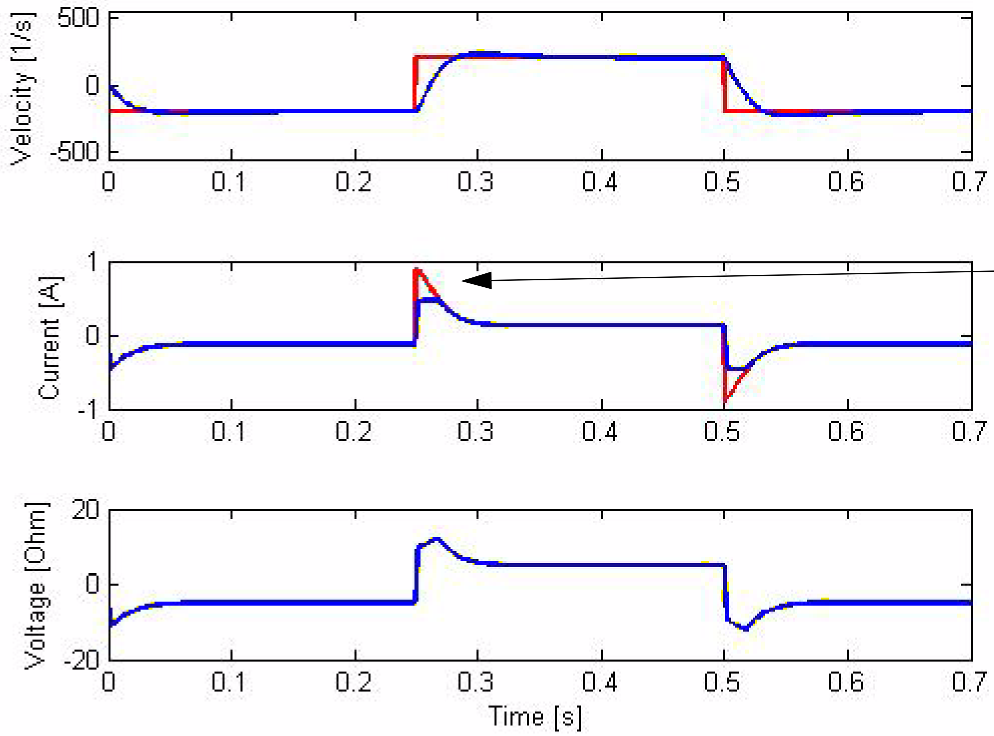
$$I = \frac{\omega_1\omega_2 J}{k_T}$$

select e.g. $\omega_1 = \omega_2 = 50$

Check if the zeros are OK -> $s_z = -30$

Perhaps OK! Lets check

Step response with velocity reference = 200 rad/s



max current is 0.5A

max voltage is 12V

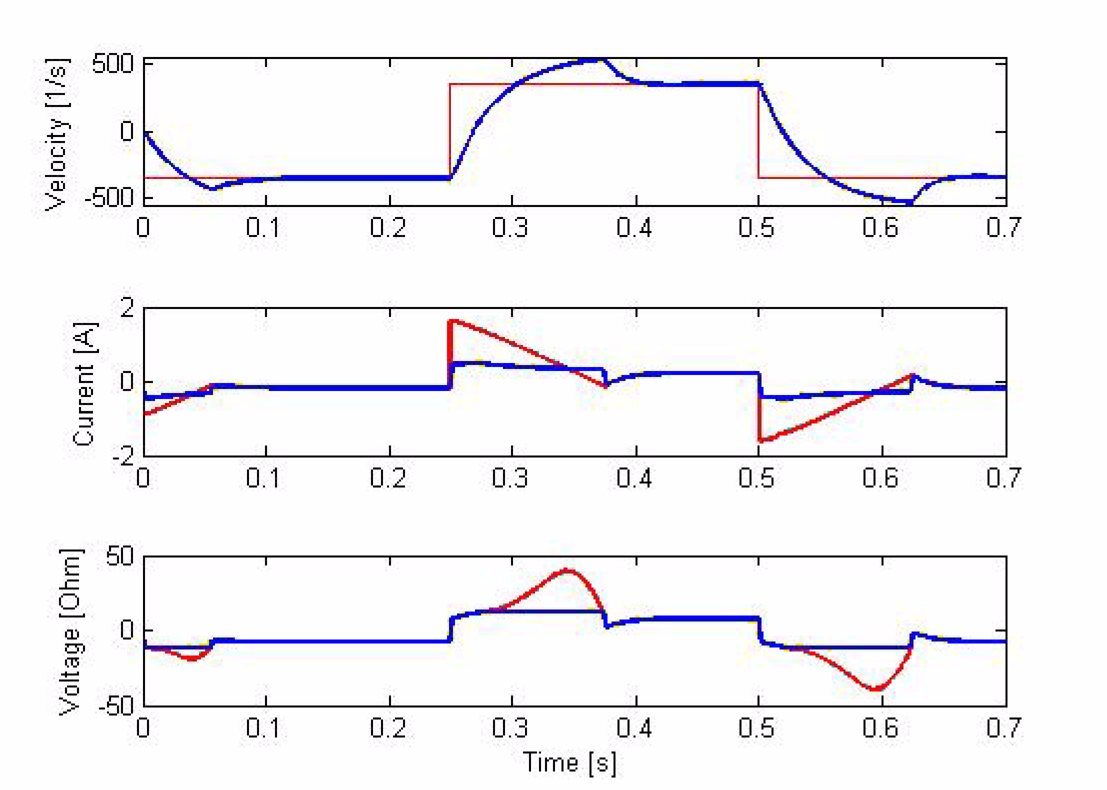
Red lines: reference signals

Blue lines: Actual signals

3.7.9 Saturation problem

reference velocity = 350 rad/s

intgral windup due to saturation of both current and voltage

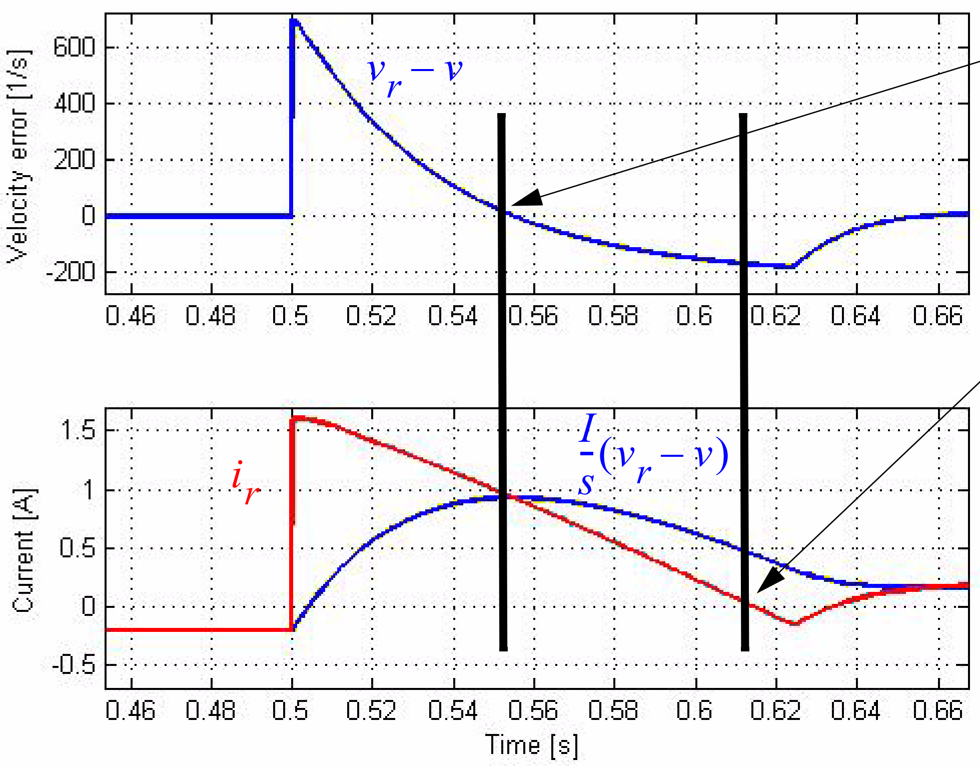


max voltage: 12V

max current: 0.5A

3.7.10 Integral windup- the reason why

- Velocity loop control law: $i_r = \left(P + \frac{I}{s}\right)(v_r - v)$ (PI error feedback)



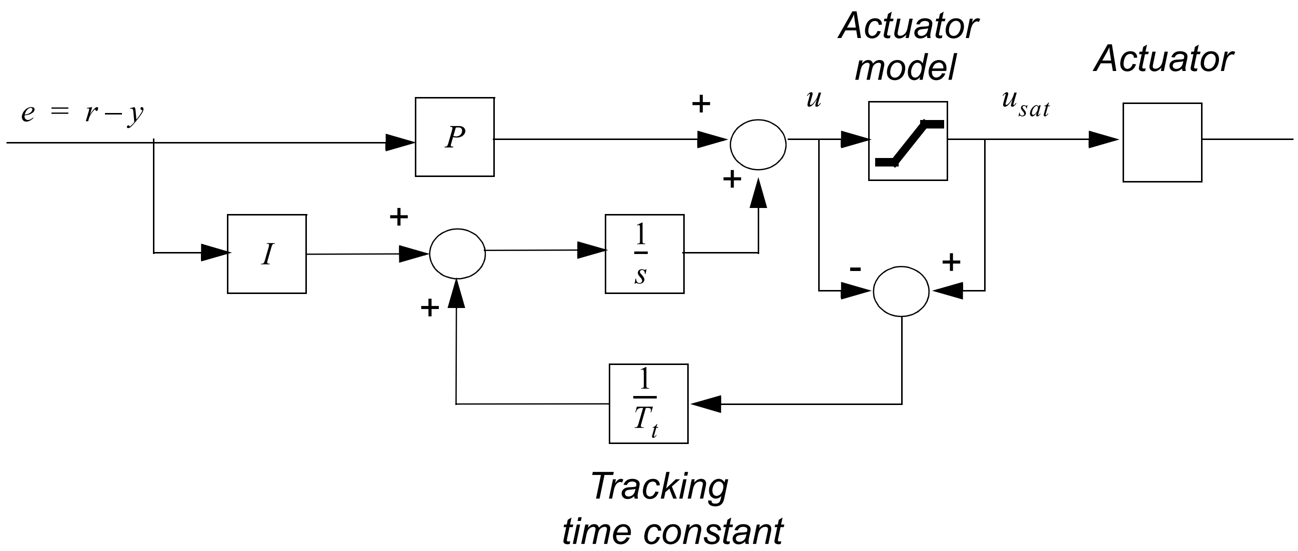
the velocity error $v_r - v$ is negative for time $t > 0.55$.

However the controlsignal i_r is positive until time $t \approx 0.61$.

Because, the integral part $\frac{I}{s}(v_r - v)$ of the controller has acumulated a high value and it takes time before it decreases. The area of the velocity error.

3.8. Windup and Anti windup control

- Windup in the integrator is caused by saturation in the actuator.
- Saturation is caused by, either a large reference step input, a large disturbance or an intial value error.
- Standard antiwindup technic is called Back-Calculation

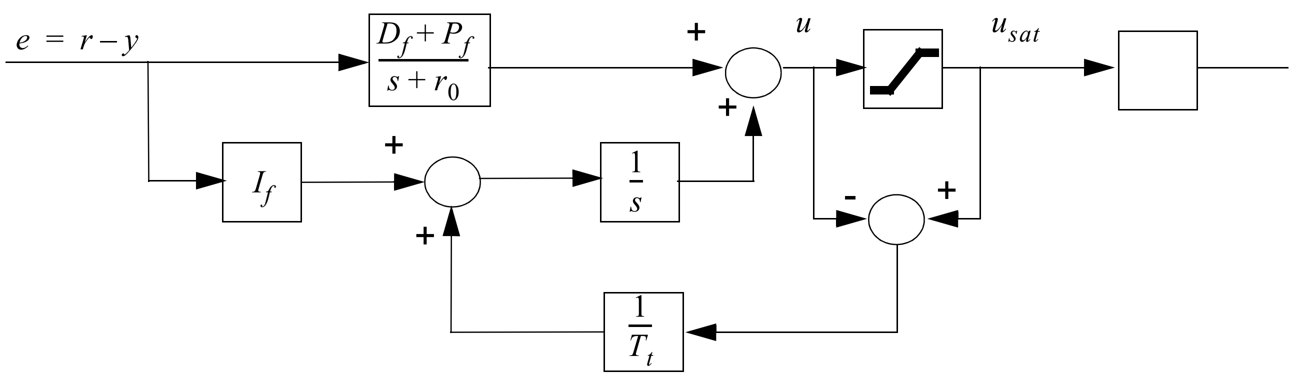


3.8.1 Antiwindup for general controllers

- If the contol has more terms then just the integrator in the denominator must it be factorized in two parts, one with only the integrator end the other with the rest.

  - Example PID controller with a low pass filter,

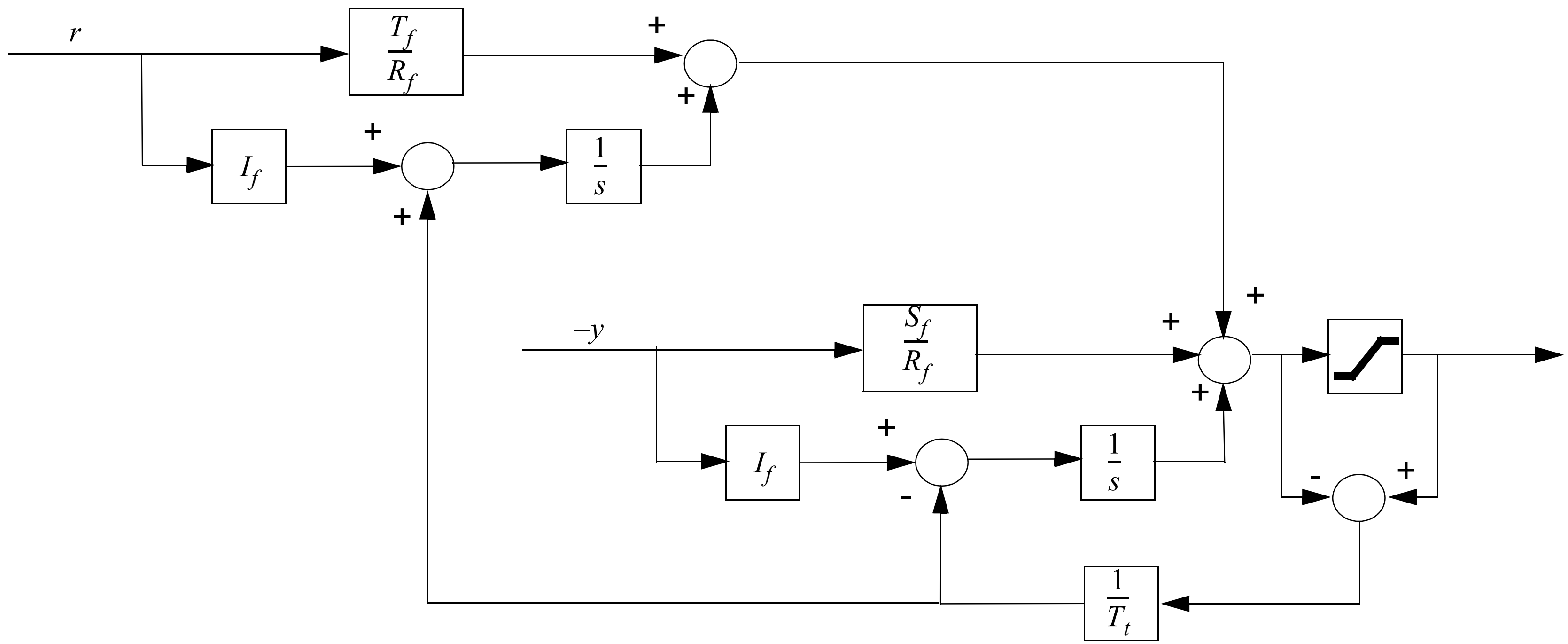
$$C(s) = \frac{Ds^2 + Ps + I}{s(s+r_0)} = \frac{I_f}{s} + \frac{D_f s + P_f}{s + r_0} \quad \text{with:} \quad I_f = \frac{I}{r_0}, P_f = P - I, D_f = D$$



- For a 2DOF control structure with control law: $u = \frac{T}{R}r - \frac{S}{R}y$

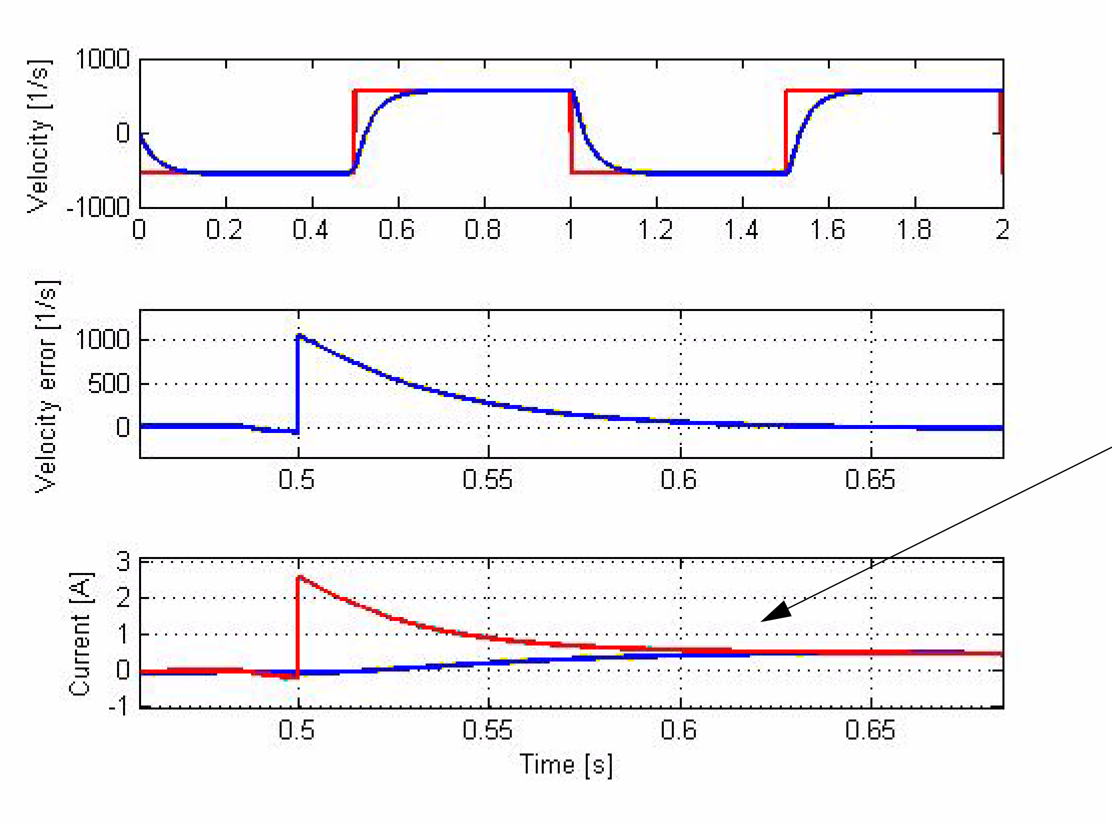
R_f is the part of R without the integrator. T_f and S_f are factorized as above.

$$u = \left(\frac{T_f}{R_f} + \frac{I_f}{s}\right)r - \left(\frac{S_f}{R_f} + \frac{I_f}{s}\right)y$$



3.8.2 Antiwindup for the cascaded controller

Step response, reference velocity = 550 rad/s with antiwindup



Even though the controlsignal

i_r (red line) saturates,

is the output of the integral part

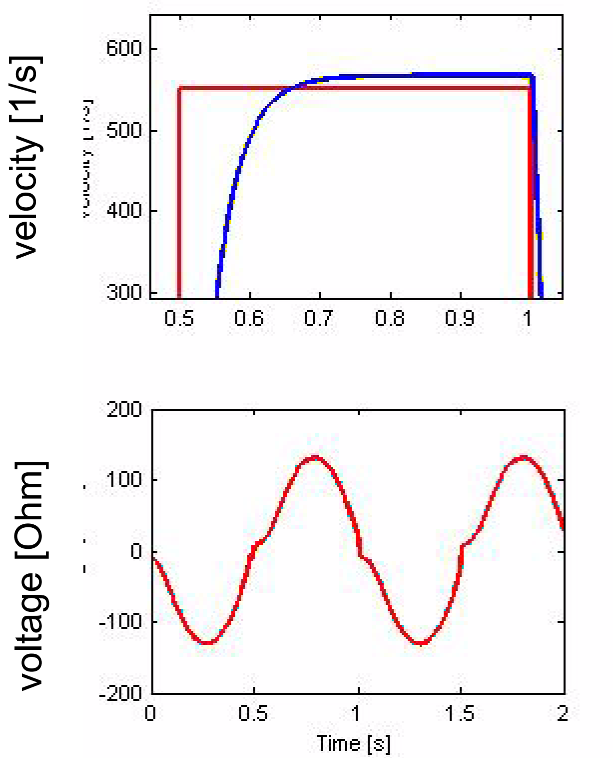
$\frac{I}{s}(v_r - v)$ (blue line) always

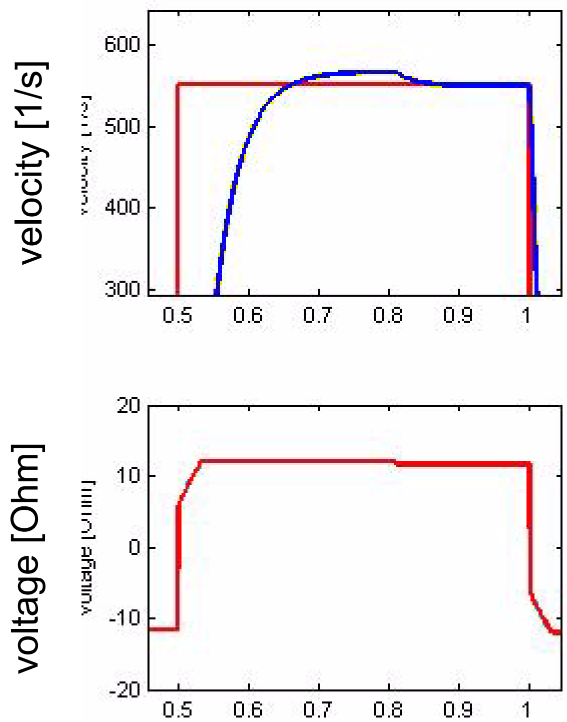
below the saturation level of

0.5A.

3.8.3 A closer analysis of the response

Steady state error in velocity because of a combination of windup in voltage and static friction.





Left plot is without antiwindup and the right with antiwindup in the voltage signal in the current controller.

3.8.4 Summary

- Polynomial approach to poleplacement feedback design
- Mapping of time domain specifications to pole location specifications.
- Selection of sample period for a discrete time implementation based on specifications.
- Approximation of the continuos time controller to a discrete time controller with e.g., Euler or Tustin.
- Evaluate the approximation in time and frequency planes.