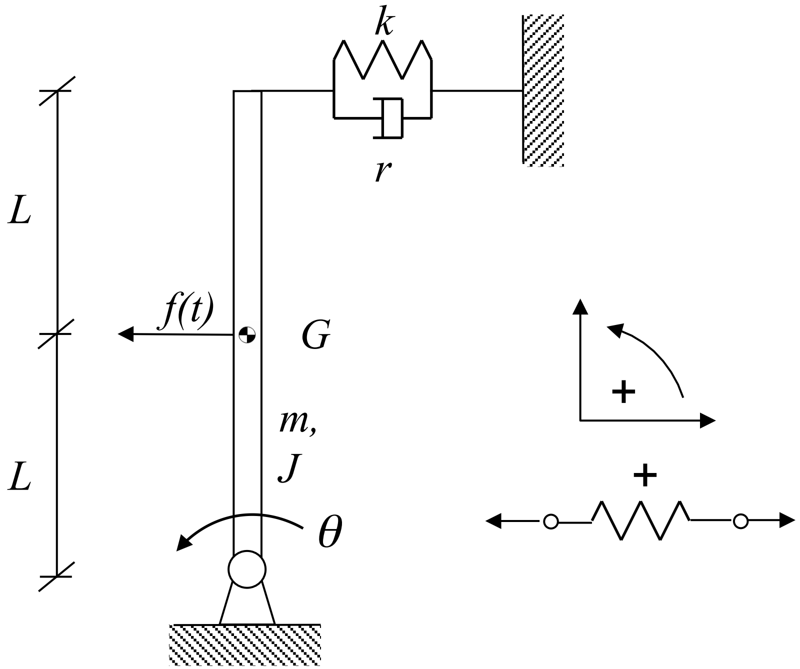
Master of Science Course in Mechanical Engineering
Politecnico di Milano

MatLab Exercise: Proportional regulators

Control and Actuating Devices for Mechanical Systems

Mechanical system



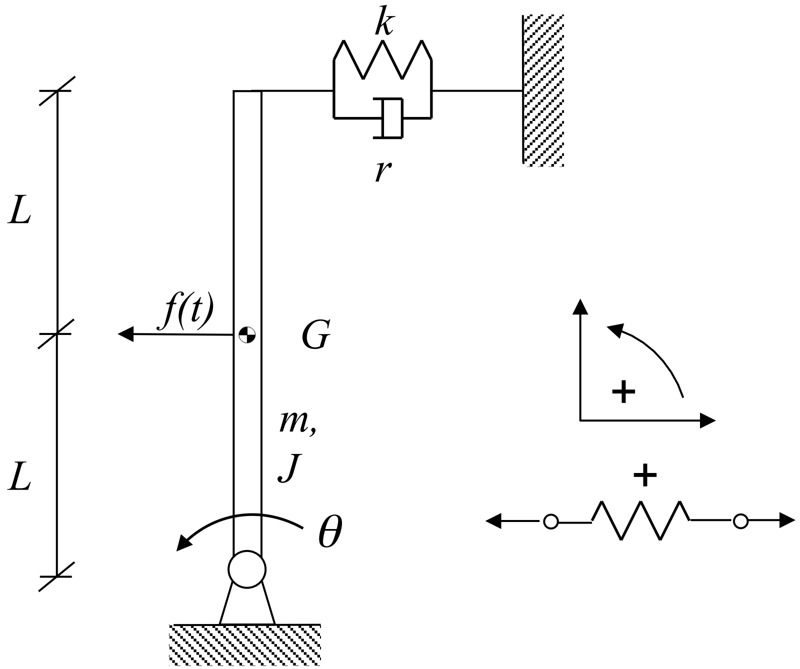
L = 1 m
m = 80 kg
J = 2 kgm^2
k = 250 N/m
c = 25 Ns/m

Equation of motion:

$$\left(mL^2+J\right)\ddot{\vartheta}+4cL^2\dot{\vartheta}+\left(4kL^2-mgL\right)\vartheta=f\left(t\right)L=T\left(t\right)$$

$$m^*\ddot{\vartheta}+c^*\dot{\vartheta}+k^*\vartheta=f\left(t\right)L=T\left(t\right)$$

Laplace domain



L = 1 m
m = 80 kg
J = 2 kgm^2
k = 250 N/m
c = 25 Ns/m

$$m^* \ddot{\vartheta} + c^* \dot{\vartheta} + k^* \vartheta = f(t) L = T(t)$$

$$\left(m^* s^2 + c^* s + k^*\right) \Theta(s) = F(s) L = T(s)$$

$$G_m(s) = \frac{\Theta(s)}{T(s)} = \frac{1}{\left(m^* s^2 + c^* s + k^*\right)}$$

TF of the passive system (no control)

Definition of generalized quantities

MatLab code

```
%----------------------------------------
% System data

g = 9.81;       %[m/s^2] gravity acceleration

m = 80;         %[kg] bar mass
J = 20;         %[kg*m^2] bar moment of inertia
L = 1;          %[m] bar length

k = 250;        %[N/m] stiffness coeff.
c = 25;         %[Ns/m] damping coeff.

% Generalized mass, damping and stiffness
ms = m*L^2 + J;
cs = 4*c*L^2;
ks = 4*k*L^2 - m*g*L;
```

How to create TFs using MatLab?

Transfer functions

TFs can be defined with the command ***tf*** in MatLab . As an example, the TF $G_m(s)$ can be defined as:

```
num_Gm = [1];                    %numerator
den_Gm  = [ms cs ks];            %denominator

Gm = tf(num_Gm,den_Gm);   %TF
```

Vectors *num_Gm* and *den_Gm* contain the coefficients of the numerator and of the denominator of G_m in descending powers of s

How to create TFs using MatLab?

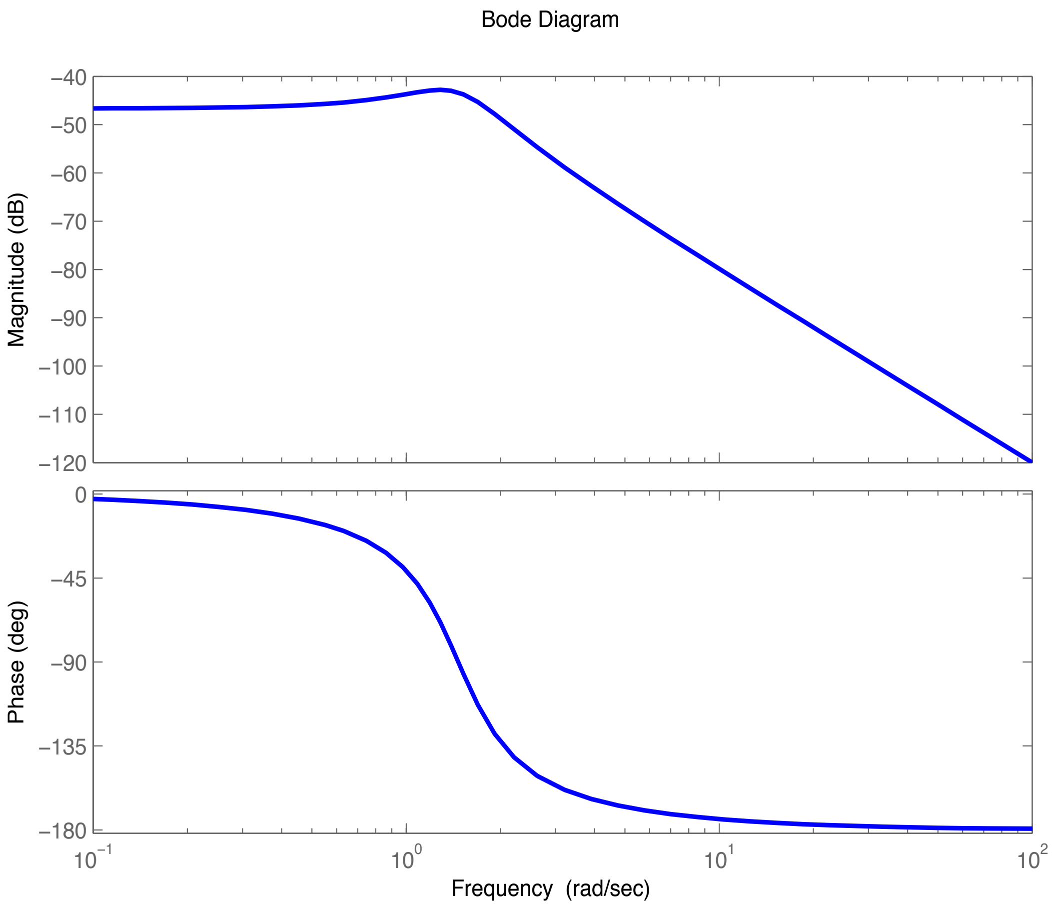
Poles of $G_m(s)$ can be calculated by means of the MatLab command ***pole***

```
>> Gm = tf([1],[ms  cs  ks])

Transfer function:

                1
---------------------------------
100 s^2 + 100 s + 215.2


>> pole(Gm)
  -0.5000 + 1.3791i
  -0.5000 - 1.3791i
```

Bode diagrams

Bode diagrams of the transfer function $G_m(s)$ can be drawn using the MatLab command ***bode***

>> bode(Gm)



Poles of $G_m(s)$

$$\lambda_{1,2} = -\alpha + i\omega = -0.5 + 1.38i$$

Natural frequency

$$\omega_0 = \sqrt{\frac{k}{m}} = 1.38\text{rad/s} = 0.23\text{Hz}$$

Damping factor

$$h = \frac{c}{2m\omega_0} = 0.34$$

The mechanical system is stable and underdamped

Laplace domain

Equation of motion (no control)

$$m^*\ddot{\vartheta} + c^*\dot{\vartheta} + k^*\vartheta = f(t)L = T(t)$$

Proportional regulator (P-regulator)

$$T(t) = f(t)L = k_p\left(\vartheta_{ref} - \vartheta\right)$$

Equation of motion of the **feedback control system**

$$m^*\ddot{\vartheta} + c^*\dot{\vartheta} + k^*\vartheta = k_p\left(\vartheta_{ref} - \vartheta\right)$$

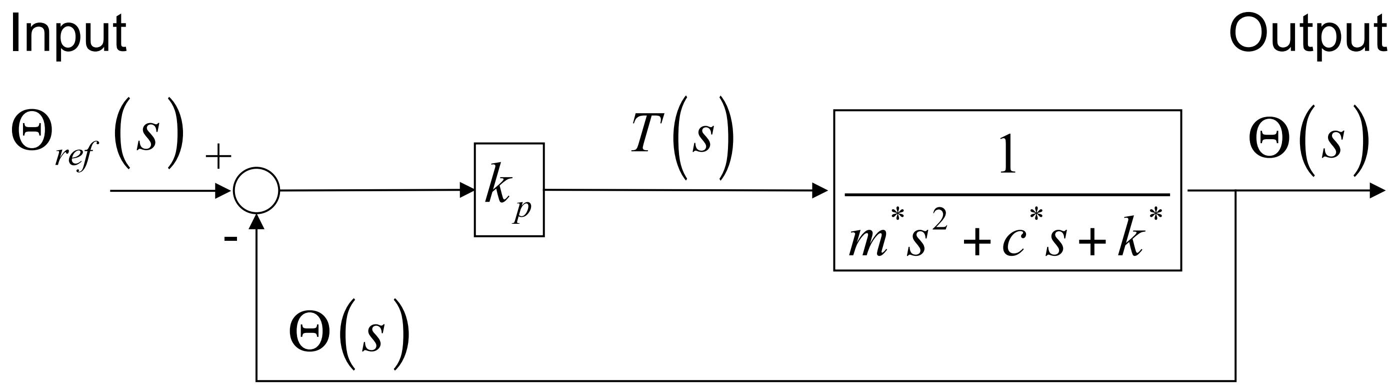
Equation of motion of the feedback control system in the **Laplace domain**

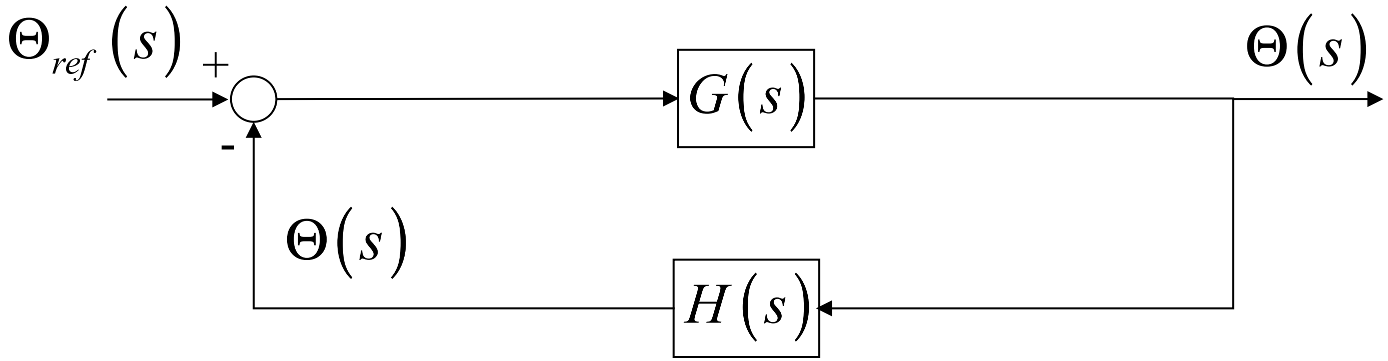
$$\left(m^*s^2 + c^*s + k^*\right)\Theta(s) = k_p\left(\Theta_{ref}(s) - \Theta(s)\right)$$

Laplace domain

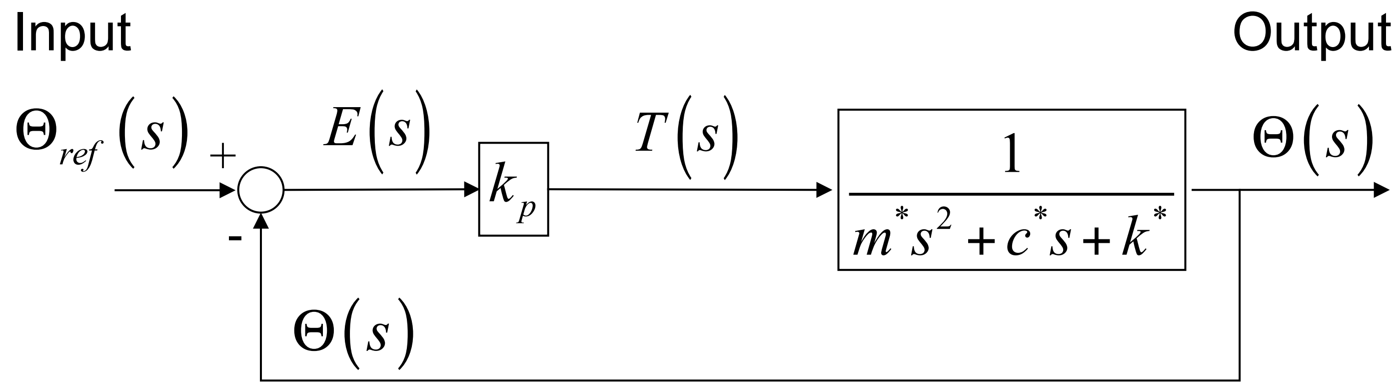
$$m^*\ddot{\vartheta} + c^*\dot{\vartheta} + k^*\vartheta = T(t) = k_p\left(\vartheta_{ref} - \vartheta\right)$$

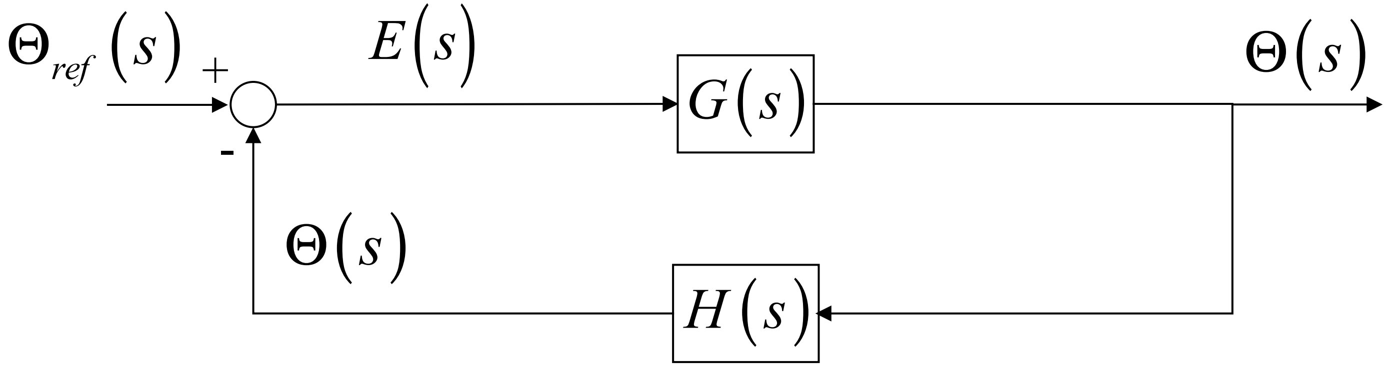
$$\left(m^* s^2 + c^* s + k^*\right)\Theta(s) = T(s) = k_p\left(\Theta_{ref}(s) - \Theta(s)\right)$$





Laplace domain





$$G(s) = \frac{\Theta(s)}{E(s)} = \frac{k_p}{m^* s^2 + c^* s + k^*}$$

$$H(s) = 1$$

Laplace domain

$$GH(s) = G(s)H(s) = \frac{k_p}{m^* s^2 + c^* s + k^*}$$ open loop transfer function

$$L(s) = \frac{G(s)}{1 + G(s)H(s)} = \frac{k_p}{m^* s^2 + c^* s + k^* + k_p}$$ closed loop transfer function

Stability analysis can be performed:

1) using Nyquist criterion (we must draw the Nyquist diagram of *GH*(*s*));
2) using Bode criterion, if and only if the uncontrolled system is stable (we must draw the Bode diagram of *GH*(*s*));
3) studying the poles of the closed loop transfer function *L(s*);
4) studying the eigenvalues of the state matrix of the feedback control system.

Stability analysis-Undirect methods

To assess stability of the feedback control system using undirect methods, the transfer function $G(s)H(s)$ must be defined

```
num_GH = [kp];                %numerator
den_GH  = [ms cs ks];         %denominator

GH = tf(num_GH,den_GH);   %open-loop TF
```

Vectors num_GH and den_GH contain the coefficients of the numerator and of the denominator of GH in descending powers of s

Once calculated the transfer function G(s)H(s), Nyquist and Bode criteria can be applied

Stability analysis-Undirect methods

Nyquist criterion

A closed-loop system is stable if the Nyquist diagram of G(s)H(s) encircles the point -1+0i in counter-clockwise sense as many times as there are poles of G(s)H(s) in the right-half of the s-plane (i.e. unstable poles of the open-loop system G(s)H(s)):

$$N = -P$$

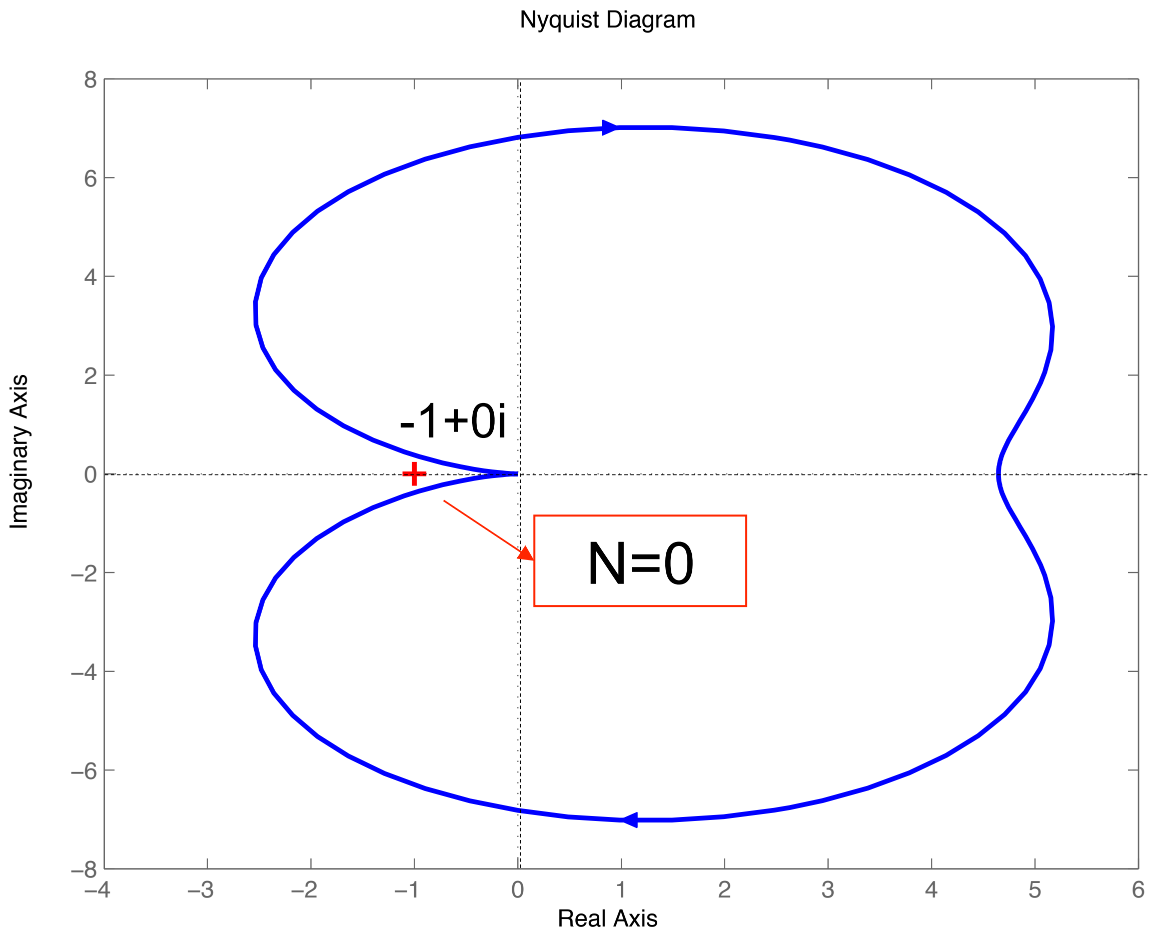
P: number of unstable poles of G(s)H(s)

N: number of encirclements of point -1+0i

- Poles of G(s)H(s) can be calculated by means of the MatLab command ***pole***
- The Nyquist diagram of G(s)H(s) can be calculated by means of the MatLab command ***nyquist***

Stability analysis-Undirect methods

Nyquist criterion



kp=1000

P=0

\>> pole(GH)

-0.5000 + 1.3791i

-0.5000 - 1.3791i

\>> nyquist(GH)

Since N=P=0, the system is stable according to the Nyquist criterion

Stability analysis-Undirect methods

Bode criterion

Hypothesis:

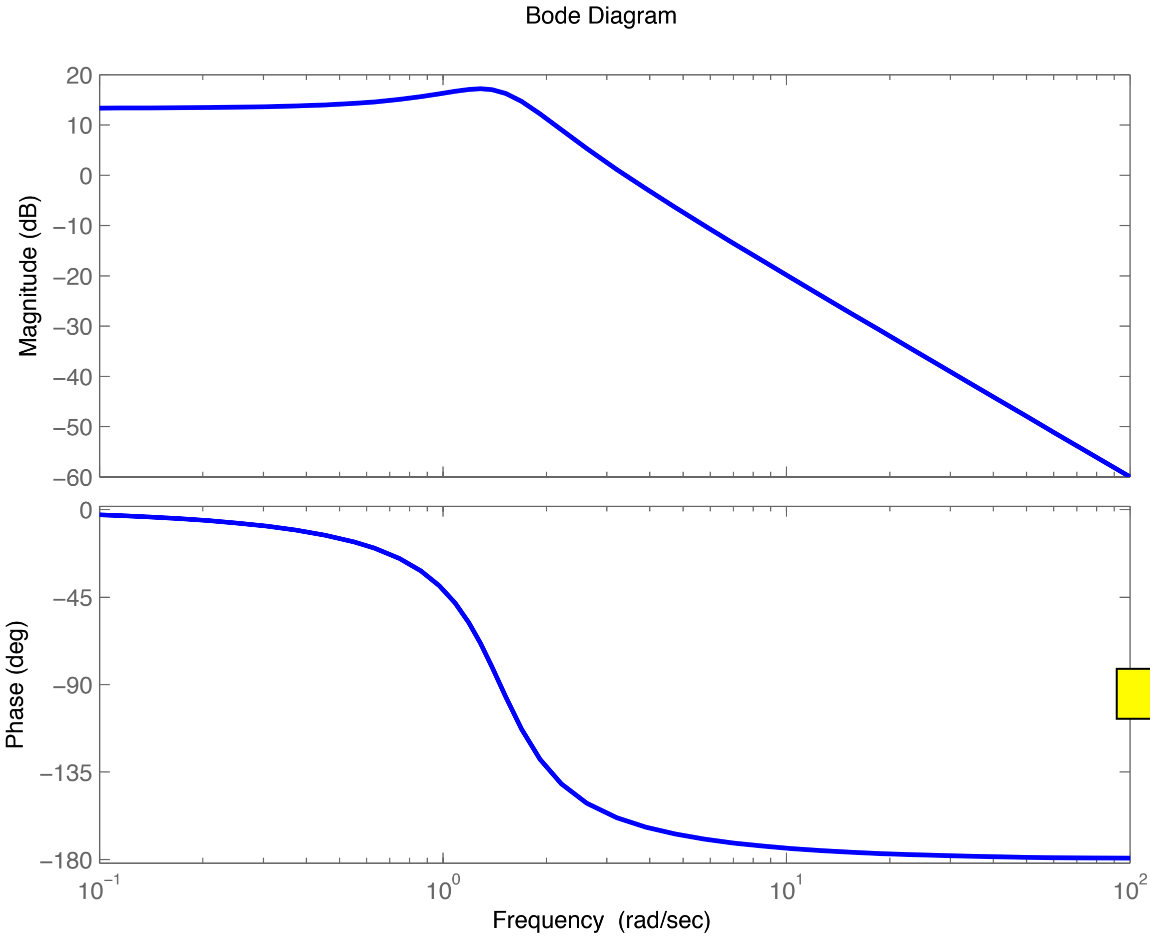
- G(s)H(s) has no unstable poles
- the amplitude diagram of G(s)H(s) intersects only once the 0dB axis

If the amplitude of G(s)H(s) at the phase crossover frequency (i.e. the value of *s* when the phase diagram is equal to -180°) is larger than one, the feedback control system is unstable (sufficient condition).

Bode diagrams of a transfer function can be drawn using the MatLab command ***bode***

Stability analysis-Undirect methods

Bode criterion



\>> bode(GH)

\>> pole(GH)

-0.5000 + 1.3791i

-0.5000 - 1.3791i

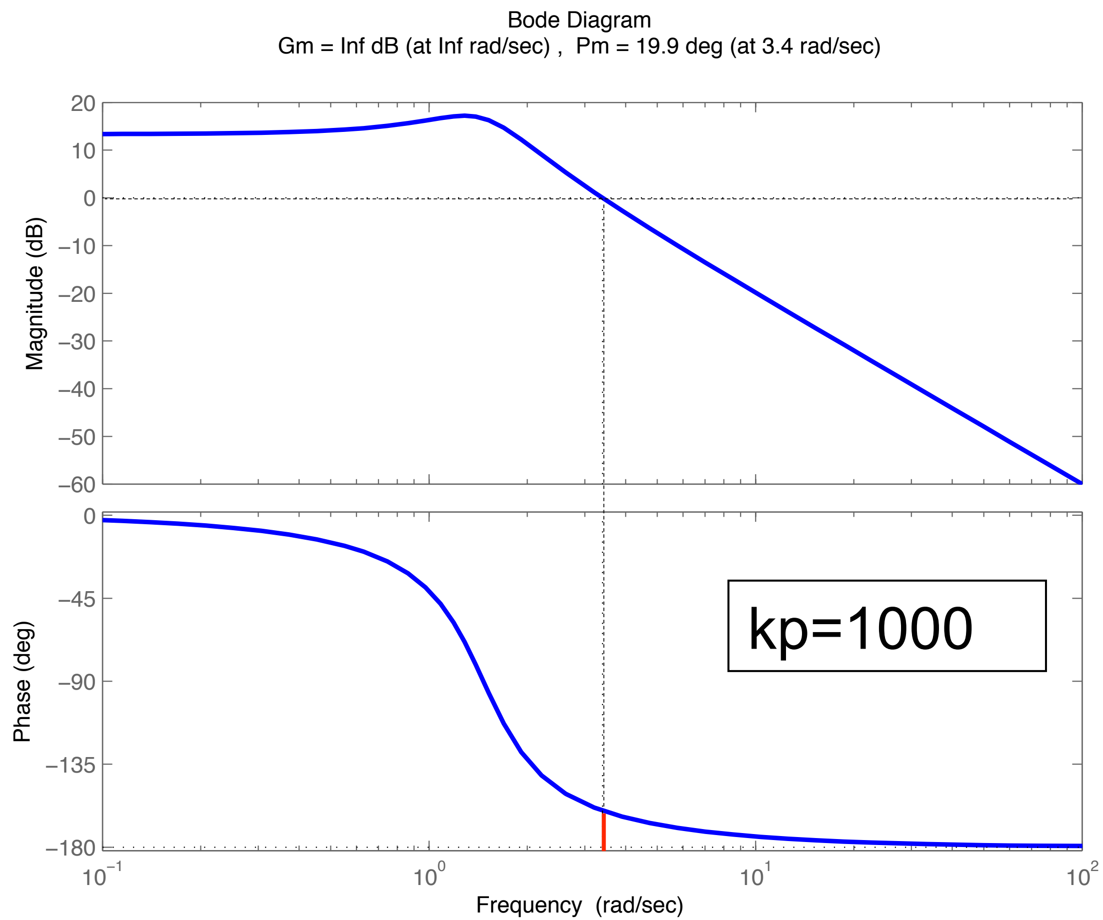
The phase is always higher than -180°

kp=1000

Stability analysis-Undirect methods

Relative stability

Gain and phase margins can be evaluated using the MatLab command ***margin***



>> margin(GH)

Pm = 19.9 deg

Gm = inf

Stability analysis-Direct methods

$$L(s) = \frac{\Theta(s)}{\Theta_{ref}(s)} = \frac{G(s)}{1+G(s)H(s)} = \frac{k_p}{m^* s^2 + c^* s + k^* + k_p}$$

Stability of the feedback control system can be assessed by calculating the poles of the closed-loop transfer function L(s).

Poles of L(s) can be calculated by means of the MatLab command ***pole***

```
num_L = [kp];                  %numerator
den_L  = [ms  cs  (ks + kp)];  %denominator

L = tf(num_L,den_L);           %closed-loop TF

pole(L)                        %poles of L(s)
```

Stability analysis-Direct methods

$$L(s)=\frac{\Theta(s)}{\Theta_{ref}(s)}=\frac{G(s)}{1+G(s)H(s)}=\frac{k_p}{m^* s^2+c^* s+k^*+k_p}$$

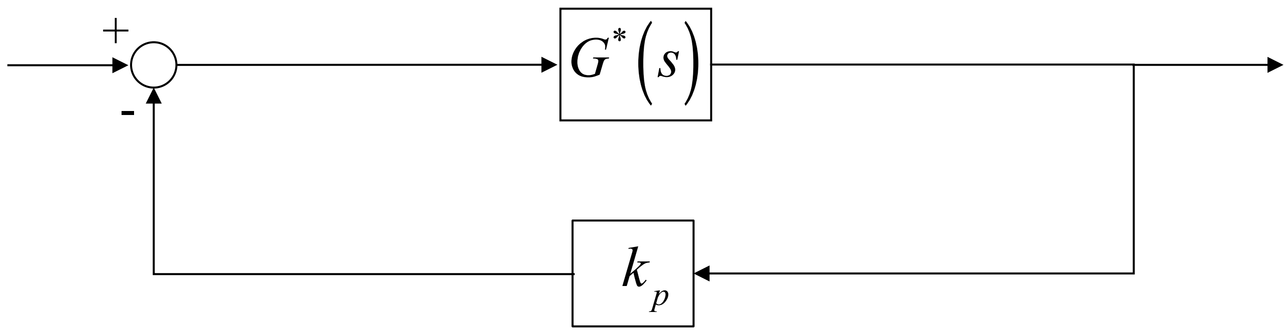
However calculated poles of L(s) are associated to a specific value of the proportional gain k_p

To evaluate how poles of L(s) vary as k_p increases, **root locus** analysis can be used

Root locus analysis can be performed in MatLab using the command ***rlocus***

Stability analysis-Direct methods

The MatLab command ***rlocus*** is used to analyze the following negative feedback loop

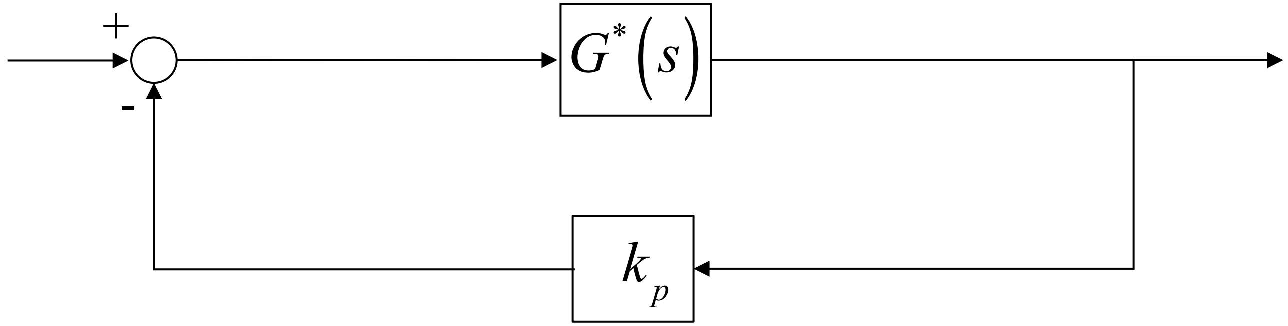


The following TF must be defined on the purpose:

$$G^*(s) = \frac{N_{G^*}}{D_{G^*}} = \frac{N_{GH}/k_p}{D_{GH}} = \frac{1}{m^* s^2 + c^* s + k^*}$$

Stability analysis-Direct methods

The MatLab command ***rlocus*** is used to analyze the following negative feedback loop



The associated closed-loop TF is:

$$L^*(s) = \frac{G^*}{1+k_p G^*} = \frac{N_{G^*}}{D_{G^*}+k_p N_{G^*}} = \frac{N_{G^*}}{N_{GH}+D_{GH}} = \frac{N_L/k_p}{D_L} = \frac{1}{m^* s^2 + c^* s + k^* + k_p}$$

L*(s) has thus the same poles and the same zeros of L(s)

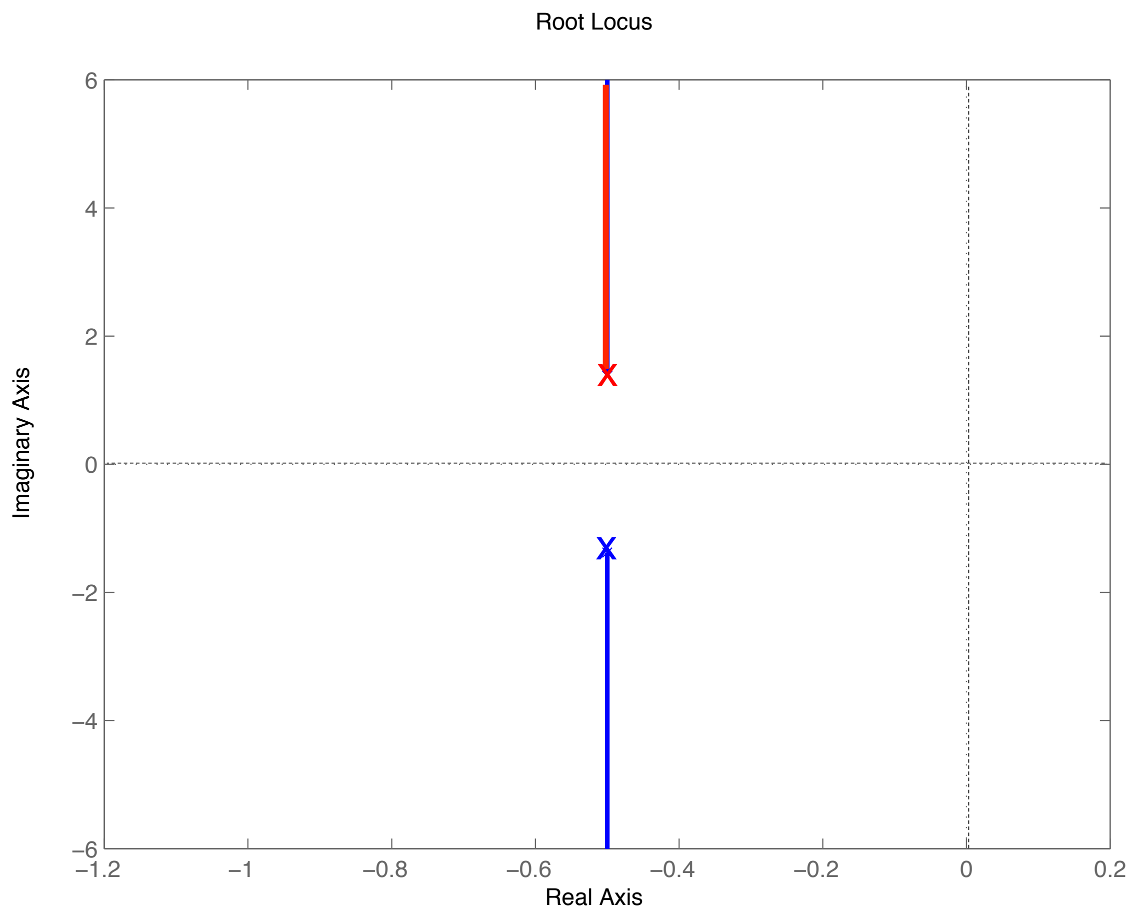
Stability analysis-Direct methods

Root locus

```
>> Gs = tf(num_GH/kp,den_GH);
>> rlocus(Gs)
```

No branches of the root locus go in the right-half of the s-plane

The systems is stable for any value of k_p



Stability analysis-Direct methods

Eigenvalues of the state matrix of the feedback control system

State-space representation

$$\begin{cases} m^*\ddot{\theta} + c^*\dot{\theta} + k^*\theta = T = k_p\left(\theta_{ref} - \theta\right) \\ \dot{\theta} = \dot{\theta} \end{cases} \Rightarrow \begin{cases} \ddot{\theta} = -\dfrac{c^*}{m^*}\dot{\theta} - \dfrac{\left(k^* + k_p\right)}{m^*}\theta + \dfrac{k_p}{m^*}\theta_{ref} \\ \dot{\theta} = \dot{\theta} \end{cases}$$

$$\underline{\dot{x}} = \begin{Bmatrix} \ddot{\theta} \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} -\dfrac{c^*}{m^*} & -\dfrac{\left(k^* + k_p\right)}{m^*} \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta} \\ \theta \end{Bmatrix} + \begin{Bmatrix} \dfrac{k_p}{m^*} \\ 0 \end{Bmatrix} \theta_{ref} = \left[A_c\right]\underline{x} + \left[B_c\right]u_c$$

$$\underline{x} = \begin{Bmatrix} \dot{\theta} \\ \theta \end{Bmatrix}$$ State vector

$$\left[A_c\right] = \begin{bmatrix} -\dfrac{c^*}{m^*} & -\dfrac{\left(k^* + k_p\right)}{m^*} \\ 1 & 0 \end{bmatrix}$$ State matrix of the control system

$$u = \theta_{ref}$$ Input vector

$$\left[B_c\right] = \begin{Bmatrix} \dfrac{k_p}{m^*} \\ 0 \end{Bmatrix}$$

Stability analysis-Direct methods

Eigenvalues of the state matrix of the feedback control system

Eigenvalues of $[A_c]$ can be calculated using the MatLab command ***eig***

```
% Eigenvalues of the state matrix of the feedback control system
Ac = [-cs/ms  -(ks + kp)/ms
    1  0];
[Vc,Dc] = eig(Ac);
disp('Eigenvalues of the state matrix of the feedback control system [Ac]:')
diag(Dc)
```

Vc: matrix of the eigenvectors of matrix Ac

Dc: matrix having as diagonal elements the eigenvalues of matrix Ac

Performance

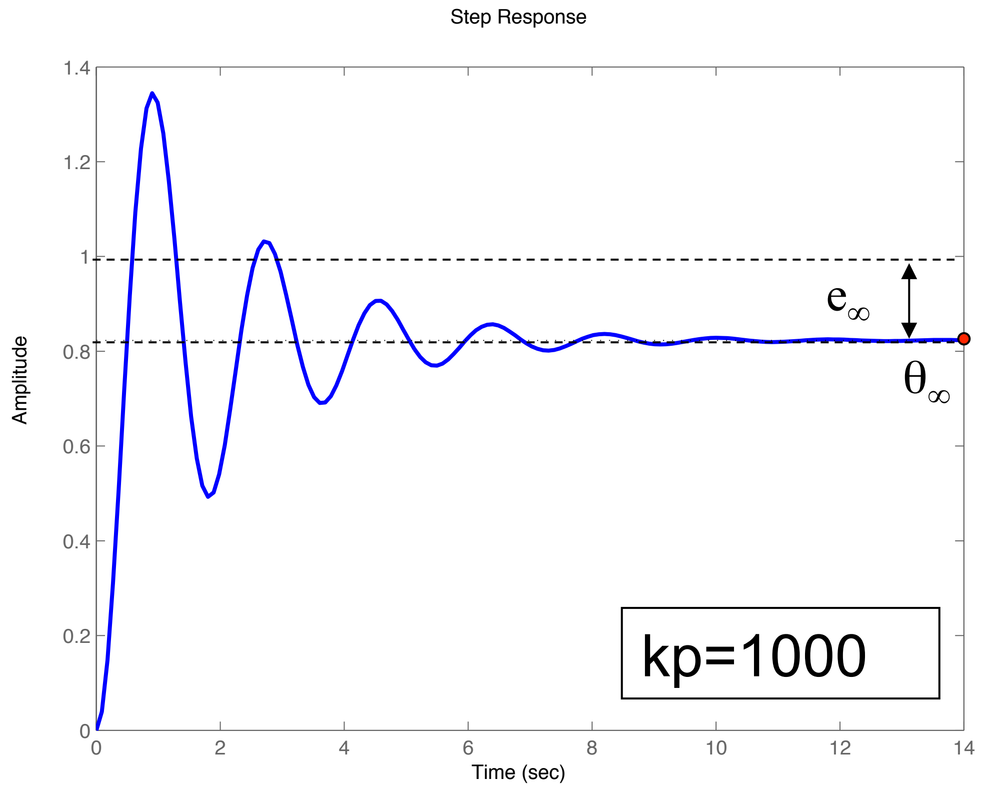
Once verified stability of the feedback control system, its performances can be assessed:

- evaluating the step response of the closed-loop systems in the time domain
- evaluating the Bode diagrams of the closed-loop transfer function L(s)

Performance

Step response

The response of the feedback control system to a unit step input can be evaluated by means of the MatLab command ***step***



step(L)

% steady-state value of L(s)

theta_inf = dcgain(L)

% steady-state error

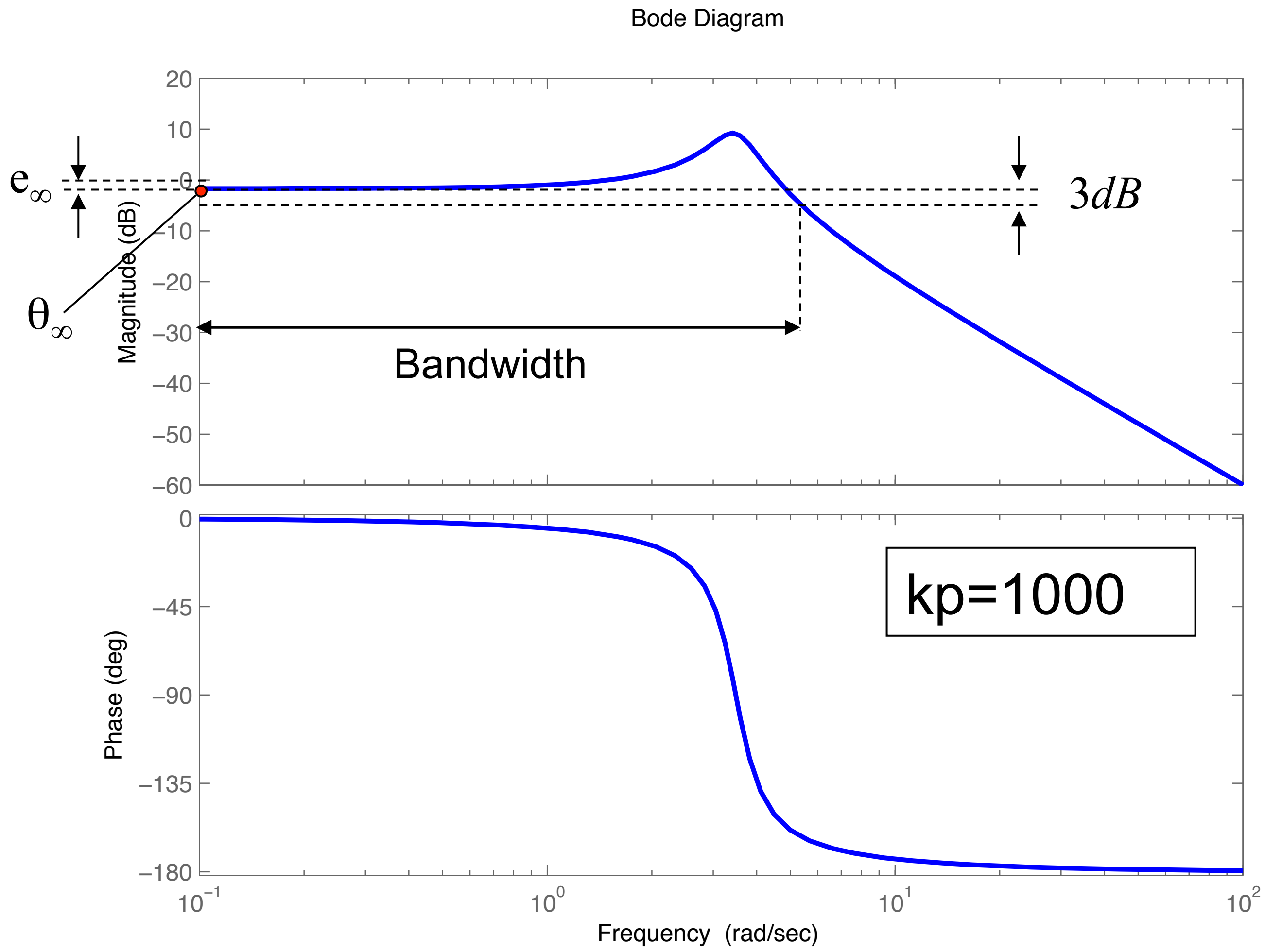
e_inf = 1 - theta_inf

The MatLab command ***dcgain*** calculates the steady-state value of the closed-loop transfer function L(s) and thus the steady-state error

Performance

Bode diagrams of L(s)

Bode diagrams of L(s) can be calculated by means of the MatLab command ***bode***



>> bode(L)

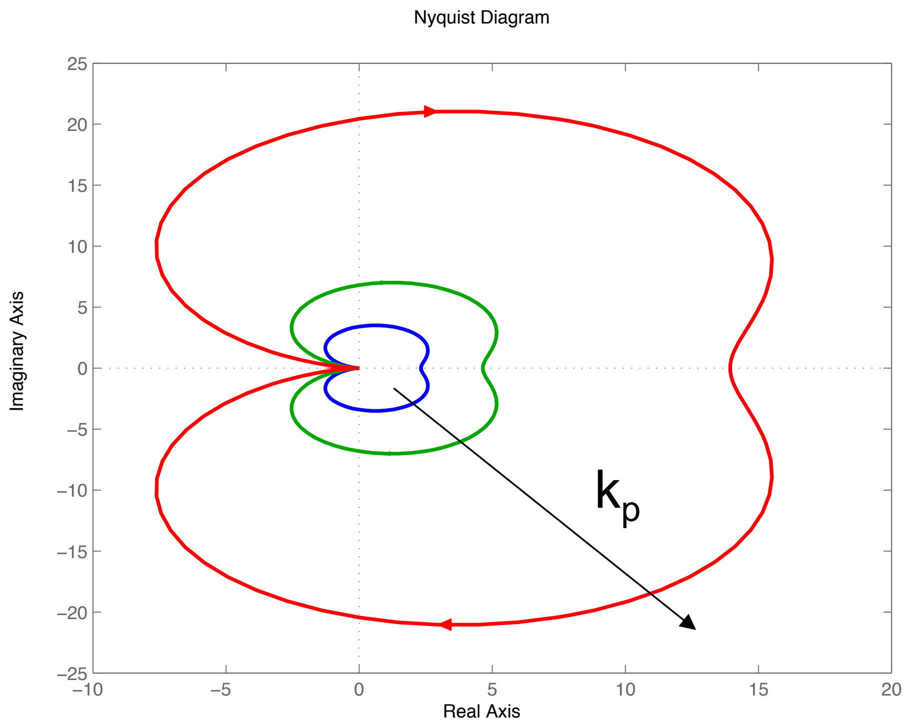
>> bandwidth(L)

The MatLab command ***bandwidth*** calculates the bandwidth of the system

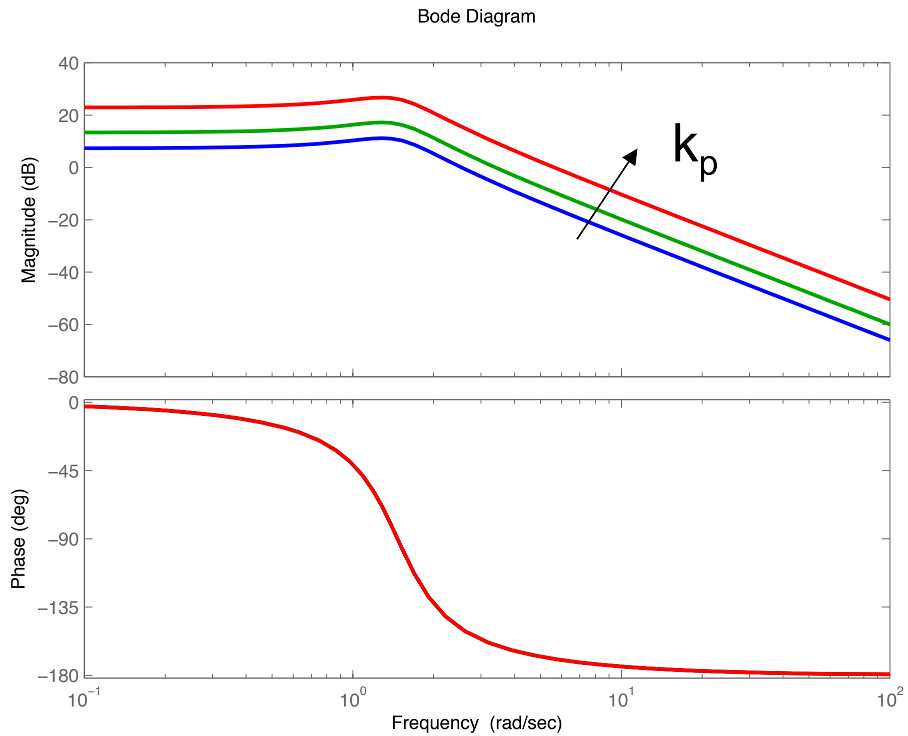
What happens if k_p varies?

Stability analysis-Undirect methods

Nyquist diagram of G(s)H(s)



Bode diagrams of G(s)H(s)

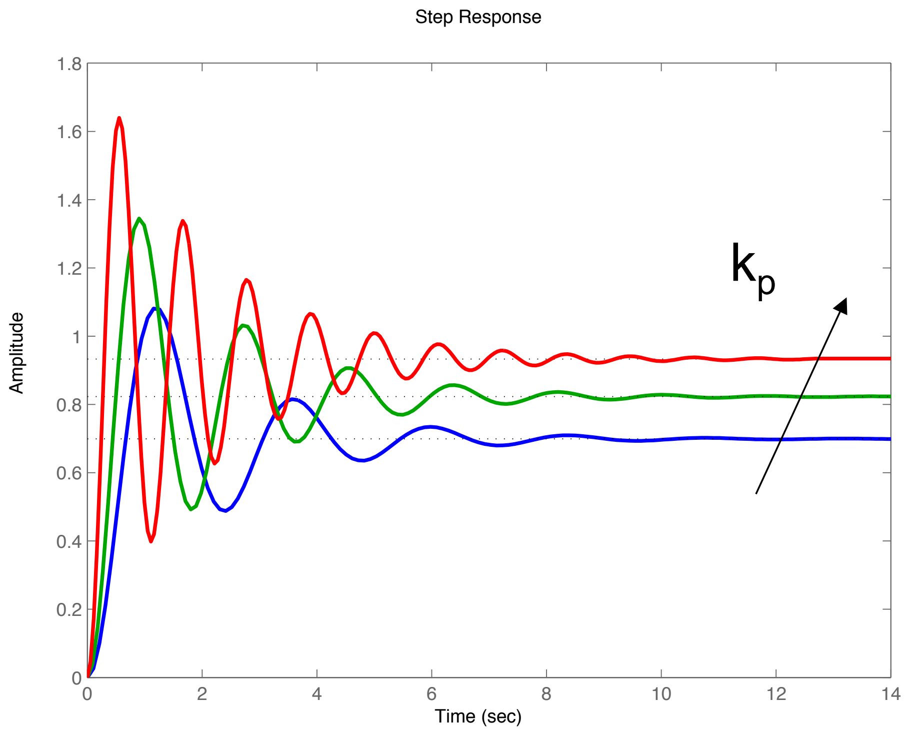


Whatever k_p, the feedback control system is always stable

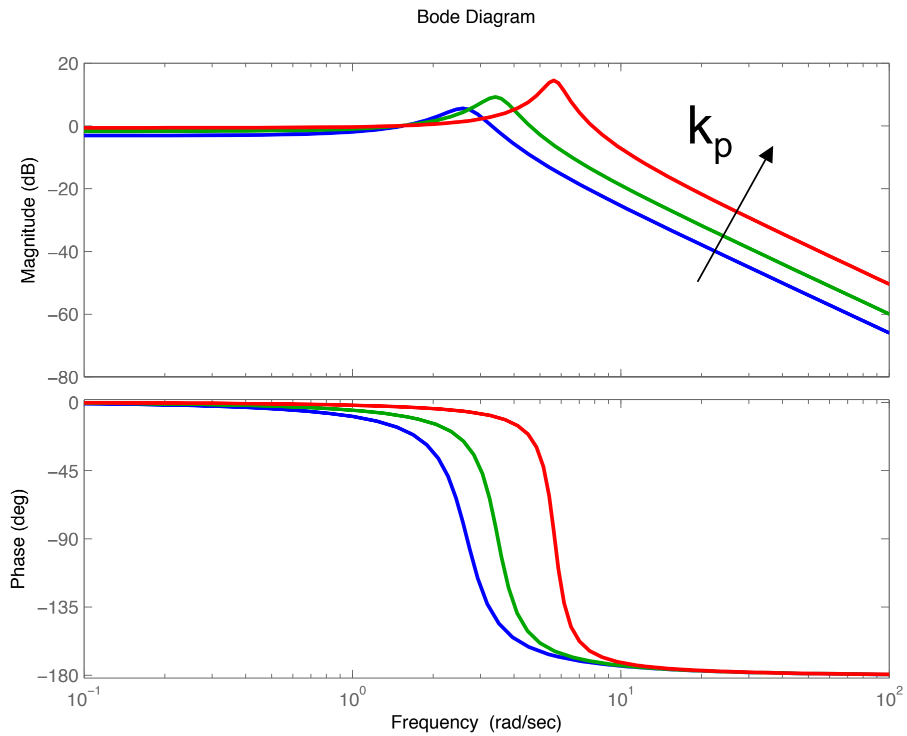
What happens if k_p varies?

Performance analysis

Step response of L(s)



Bode diagrams of L(s)



Homework

Evaluate:

1. stability (use direct and undirect methods)
2. and performance (step-response and Bode diagrams)

of the feedback control system in the following cases:

- k = 125 N/m
- k = 196.2 N/m

Summary of MatLab commands

tf: creates a TF

pole: calculate the poles of a TF

bode: draw the Bode diagrams of a TF

nyquist: draw the Nyquist diagram of a TF

margin: calculates phase and gain margin of a TF

rlocus: draw the root locus of a TF

eig: calculates the eigenvalue of a matrix

dcgain: calculates the static gain of a TF

bandwidth: calculates the bandwidth of a TF

step: draws the response of a system to a unit step input