Hands On
Design of a Phase Lead Network for a DC Motor Model

0. Introduction

A hands on activity is proposed within the Automatic Control area in an Information Science Engineering curriculum, which can contain basic and elective courses of Automatic Control for BSc and MSc degrees.
Through this design example, it is shown that a 'learning by doing approach', which enhances the development of theoretical and practical issues proposed to the student at the same time, should support the teaching activity. On the other hand, the use of 'real or realistic examples' taken from different engineering backgrounds helps to engage students and attract their interest towards difficult theoretical activities. Moreover, the design of proper 'manual and semi-automated procedures' that are tailored to the considered application examples drives the students to learn the engineering approach to solve practical problems.
The typical problem under consideration concerns the teaching of the design of phase lead (lag) control networks by using basic tools of the Control System Toolbox, as well as the derivation of transfer function and state-space models from a physical model of a real system. It is known that the design of control networks can be performed using empirical approaches or semi-automated methodologies. To this aim, the synthesis of a phase lead or lag network can be easily achieved from the required response transient time or phase margin performances, by means of the use of the root locus or Bode diagram, and through the analysis of proper frequency function.

1. DC Motor Model Derivation

The DC motor represented in Figure 1 is controlled through its armature (variables with subscript 'a') via the voltage and current V_a and i_a, respectively. Its field (variables with subscript 'e') is excited by the constant voltage and current V_e and i_e, respectively. Figure 1 indicates also the load inertia J, the resistance torque due to the frictional torque of the rolling bearings and the ventilation resistance f (also depending on the angular speed $\omega(t)$), and the load torque C_c.

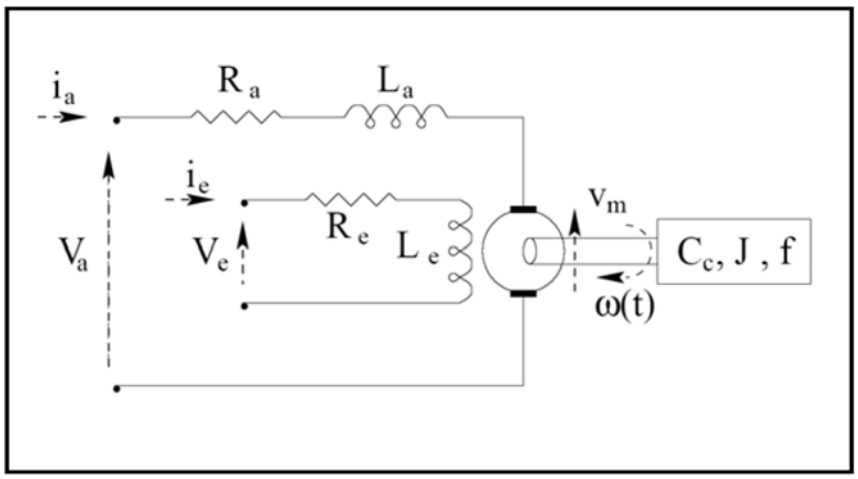

Figure 1: DC motor scheme.

The variables of the DC motor considered in this problem are: $R_a = 3\Omega$, $L_a = 30mH$, $k_m = 2Nm/A$, $J = 3kg\,m^2$, and $f = 5 \cdot 10^5 Nms/rad$. In particular, the parameter R_a represents the armature resistance, L_a is its inductance, whilst the constant k_m represents the counter-electromotive force generated by the motor rotating at angular velocity $\omega(t)$. If the armature voltage $V_a(t)$ and the load torque $C_c(t)$ are the inputs of the model, whilst its outputs are the armature current $i_a(t)$ and the angular velocity $\omega(t)$, from the scheme of Figure 1 it is possible to derive the state-state model in the form of Eq. (1):

$$\begin{cases} \begin{bmatrix} \dot{i}_a(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{k_m}{L_a} \\ \dfrac{k_m}{J} & -\dfrac{f}{J} \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{L_a} & 0 \\ 0 & -\dfrac{1}{J} \end{bmatrix} \begin{bmatrix} V_a(t) \\ C_c(t) \end{bmatrix} \\ \omega(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} \end{cases} \tag{1}$$

The matrices of the state-space mode of Eq. (1), (A, B, C, D), are represented by the relations of Eq. (2):

$$A = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{k_m}{L_a} \\ \dfrac{k_m}{J} & -\dfrac{f}{J} \end{bmatrix},\ B = \begin{bmatrix} \dfrac{1}{L_a} & 0 \\ 0 & -\dfrac{1}{J} \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2}$$

when the controlled output is $y(t) = \omega(t)$, with:

$$x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix},\ u(t) = \begin{bmatrix} V_a(t) \\ C_c(t) \end{bmatrix} \tag{3}$$

If the relative angular position $\alpha(t)$ is also required, the relations of Eq. (3) have the form of Eq. (4):

$$x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \\ \alpha(t) \end{bmatrix},\ y(t) = \begin{bmatrix} \omega(t) \\ \alpha(t) \end{bmatrix} \tag{4}$$

When a Single-Input Single-Output (SISO) model of the DC motor is considered for control design purpose, the control input can be defined as $u(t) = \mathrm{V_a}(t)$, whilst the controlled output is

$y(t) = \alpha(t)$. Under these assumptions, the matrices of the DC motor model have the form of the relations of Eq. (5):

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_m}{L_a} & 0 \\ \frac{k_m}{J} & -\frac{f}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B = \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \; D = \begin{bmatrix} 0 \end{bmatrix} \tag{5}$$

The transfer function of the SISO model can be determined by using the relation of Eq. (6):

$$G(s) = C(sI - A)^{-1}B + D \tag{6}$$

that is directly implemented by the Matlab function `[b,a] = ss2tf(A,B,C,D)`. It converts the state-space representation (A,B,C,D) into the equivalent transfer function $G(s)$, whose coefficients of the numerator and denominator polynomials are defined by the Matlab vectors `b` and `a`, respectively. The DC motor matrices (A,B,C,D) are defined in the script file provided for the hands on activities (file `DC_motor_model_3rd_order.m`).

2. Control Design Problem Definition

Given the SISO model of the DC motor, the controller design problem requires to determine a phase lead or lag network described by the transfer function $R(s)$ of Eq. (7):

$$R(s) = K \frac{1 + s/s_z}{1 + s/s_p} \tag{7}$$

By using the continuous time root locus for the controlled system $G(s)$ of Eq. (6), with reference to the network $R(s)$, the problem requires to determine the values of the pole and the zero, s_p and s_z, as well as the DC gain K, such that the transient conditions (8) are fulfilled in terms of settling time T_a and maximum overshoot $S\%$:

$$\begin{cases} T_a \le 0.5s \\ S\% \le 35\% \;\; (\delta \ge 0.33) \end{cases} \tag{8}$$

with respect to the reference unit step input.

3. Controller Design Solution

By executing the Matlab script file `DC_motor_model_3rd_order.m`, the transfer function $G(s)$ defining the 3rd order model of the DC motor is obtained in the form of Eq. (9):

$$G(s) = \frac{22.22}{s^3 + 100\,s^2 + 44.61\,s} \tag{9}$$

which contains a pole in $s = 0$, thus resulting a 'type 1' system. The response of this model presents zero steady state error when the reference input is a unit step function. As already remarked, the transient response of this system has to fulfil the following conditions:

$$\begin{cases} T_a \le 0.5s \\ S\% \le 35\% \;\; (\delta \ge 0.33) \end{cases} \tag{10}$$

when controlled by the network $R(s)$ of Eq. (7) for a proper definition of the parameters s_p, s_z, and K, and for the reference unit step $r(t)$.
The Matlab script file `DC_motor_model_3rd_order.m` defines the transfer function of the controlled system, by using the following MATLAB code:

```
Gs_sys = ss(A,B(:,1),C(2,:),D(1,2));

[numGs,denGs] = tfdata(Gs_sys,'v');

Gs = tf(numGs,denGs);
```

as the matrices of the state-space model of the DC motor have been already defined in the same script file.

The root locus of the closed-loop system $G(s)$ can be obtained from the following MATLAB function:

```
>>rlocus(Gs)
```

that is depicted in Figure 2.

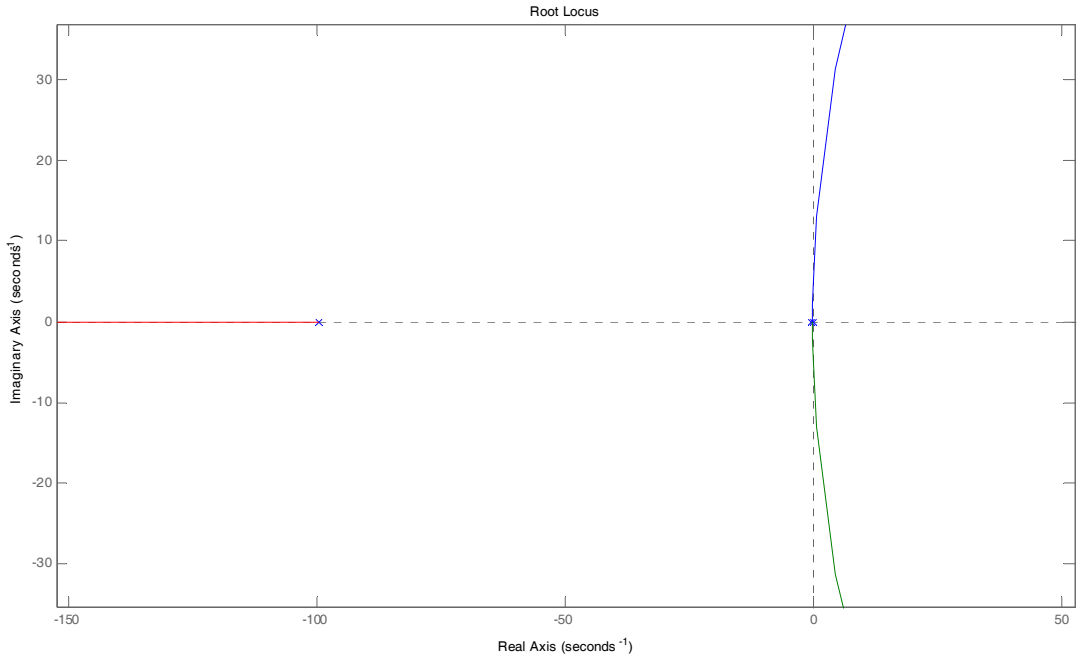

Figure 2: Root locus of the system G(s).

Figure 2 highlights that most of the root locus of $G(s)$ develops in the right half plane, and only a limited part of it that is requiref for the design of the network $R(s)$ is in the left half place, corresponding to a stable system. By using the MATLAB function `rlocfind(Gs)`, the critical (ultimate) gain K_c can be easily determined, which defines the stability region for the closed-loop system, described by the relation of Eq. (11):

$$1 + K\,G(s) = 0 \tag{11}$$

with:

$$0 \le K \le K_c \tag{12}$$

The MATLAB command `rlocfind` is used for the interactive selection of the gain from the root locus plot generated by `rlocus`. The command `rlocfind` puts up a crosshair cursor in the graphics window, which is used to select a pole location on the existing root locus. In this way, the following code is used with the root locus plot of Figure 2:

```
>> Kc=rlocfind(Gs)

Select a point in the graphics window

selected_point =

0.0000 + 6.1929i

Kc =

172.5982
```

The current selection of one of the intersection point between the root locus of $G(s)$ and the imaginary axis determines the critical (ultimate) gain $K_c = 172.6$, as shown in Figure 3.

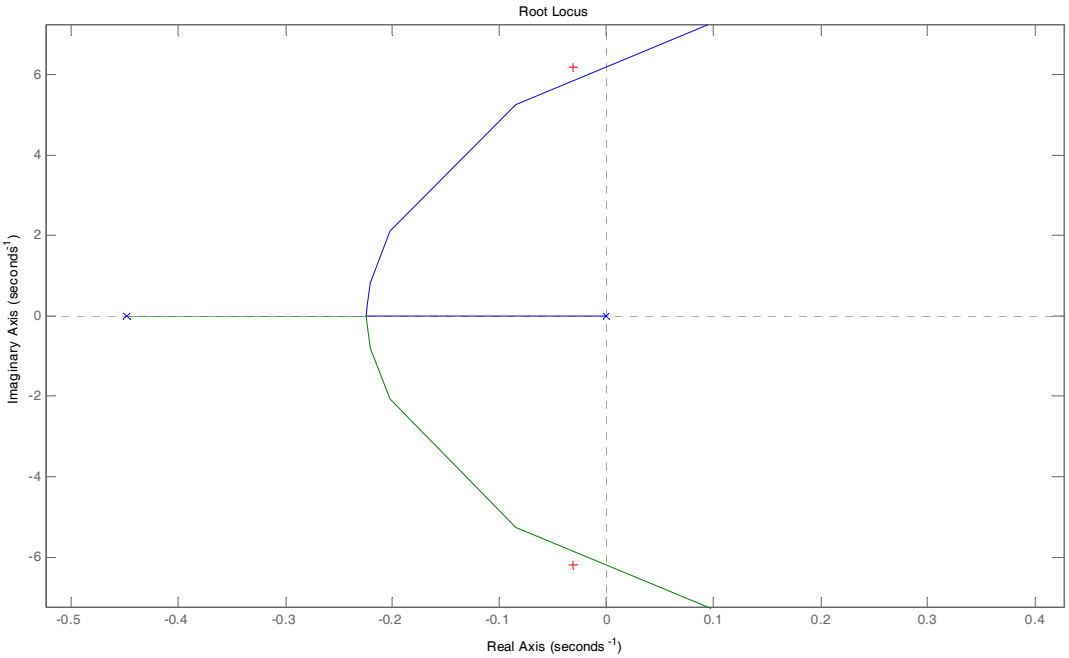

Figure 3: Detail of the root locus of G(s).

Moreover, the following command displays the roots of the polynomial of the denominator `denGs` of the transfer function $G(s)$, which are:

```
>> roots(denGs)

ans =

0

-99.5536

-0.4481

>>
```

thus highlighting that the controlled system $G(s)$ has a pole $s = 0$ ('type 1' system) and is simply stable. Note that the denominator `denGs` of the transfer function $G(s)$ was determined with the following function:

```
[numGs,denGs] = tfdata(Gs_sys,'v')
```

Moreover, by using the closed-loop scheme implemented in the Simulink environment in Figure 4 (file '`DC_motor_controller.mdl`'), the settling time and the maximum overshoot can be easily computed as follows:

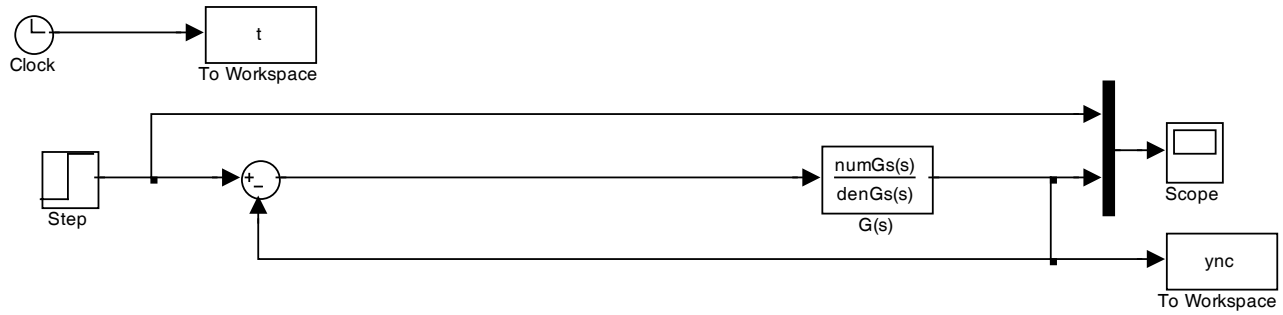

Figure 4: Closed-loop system of G(s) with $K = 1$.

```
>> lsiminfo(ync,t)

ans =

SettlingTime: 17.5024

Min: 0

MinTime: 0

Max: 1.1861

MaxTime: 7.5500

>>
```

The achieved results show that a simple proportional controller $R(s) = K$ would not be able to meet the required performances of Eq. (10). In fact, if the scheme of Figure 5 is considered, by using again the root locus of $G(s)$, the best achievable value of the settling time T_a can be obtained, as remarked in the following.

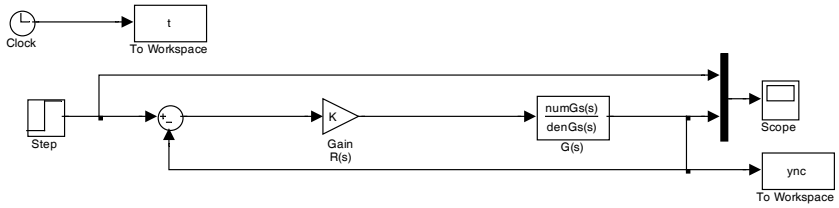

Figure 5: Proportional controller for the DC motor model described by $G(s)$.

The best value of T_a is thus obtained by selecting the gain K corresponding to the point highlighted in Figure 6, i.e. the break out point of the root locus of $G(s)$.

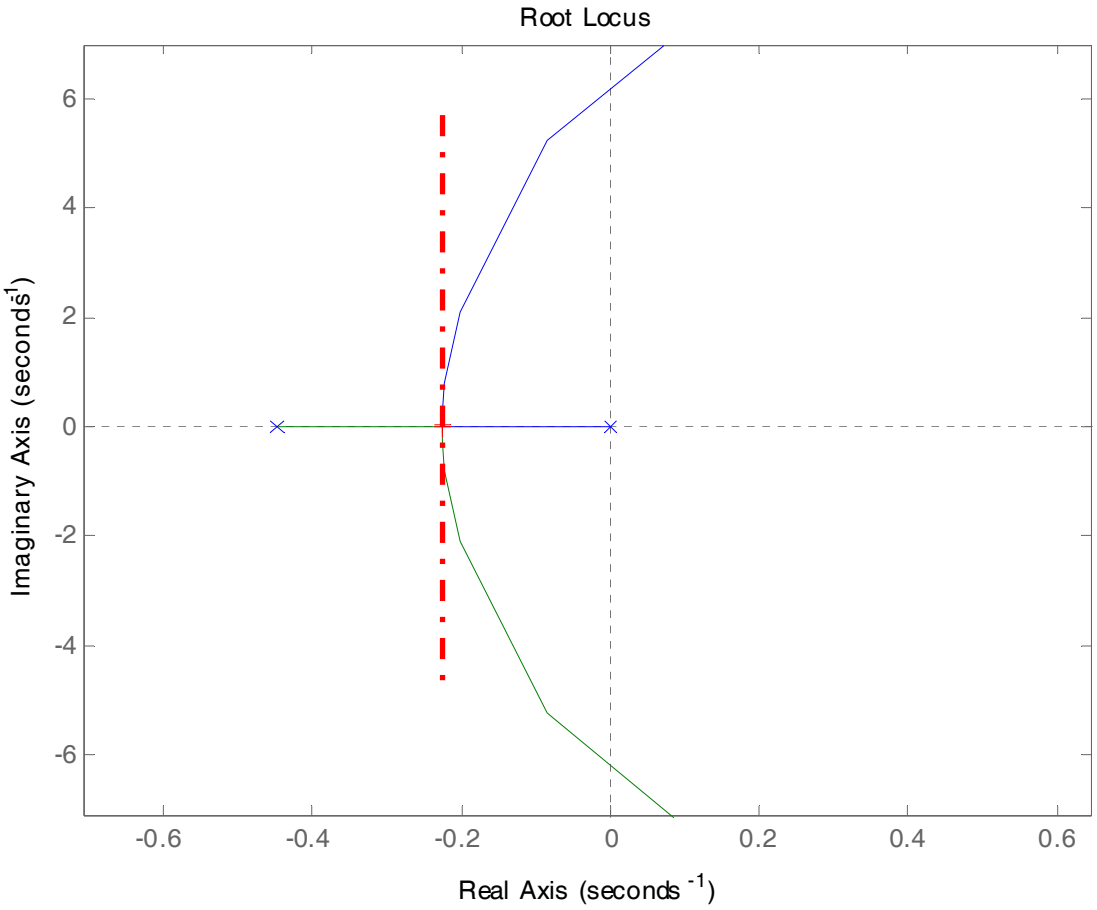

Figure 6: Root locus for the selection of the constant gain K.

```
>> rlocus(Gs)

>> K=rlocfind(Gs)

Select a point in the graphics window

selected_point =

  -0.2228 - 0.0169i

K =

    0.2257

>>
```

The corresponding value of the gain is $K = 0.2257$, and the response of the system is depicted in Figure 7. In these conditions, the response of the system is the same of a first order model, described by the relation of Eq. (13):

$$y(t) = 1 - e^{-t/\tau} \tag{13}$$

where τ corresponds to the 'dominant' time constant (dominant pole) of the closed-loop system depicted in Figure 5. The dominant pole can be determined by means of the following relations:

```
>> Grk=K*Gs/(1+K*Gs)

Transfer function:
               5.015 s^3 + 501.5 s^2 + 223.7 s
-----------------------------------------------------------------
s^6 + 200 s^5 + 1.009e004 s^4 + 8927 s^3 + 2492 s^2 + 223.7 s

>> [numGrk,denGrk]=tfdata(Grk,'v')

>> roots(denGrk)

ans =

         0
  -99.5541
  -99.5536
   -0.4481
   -0.2240
   -0.2236

>>
```

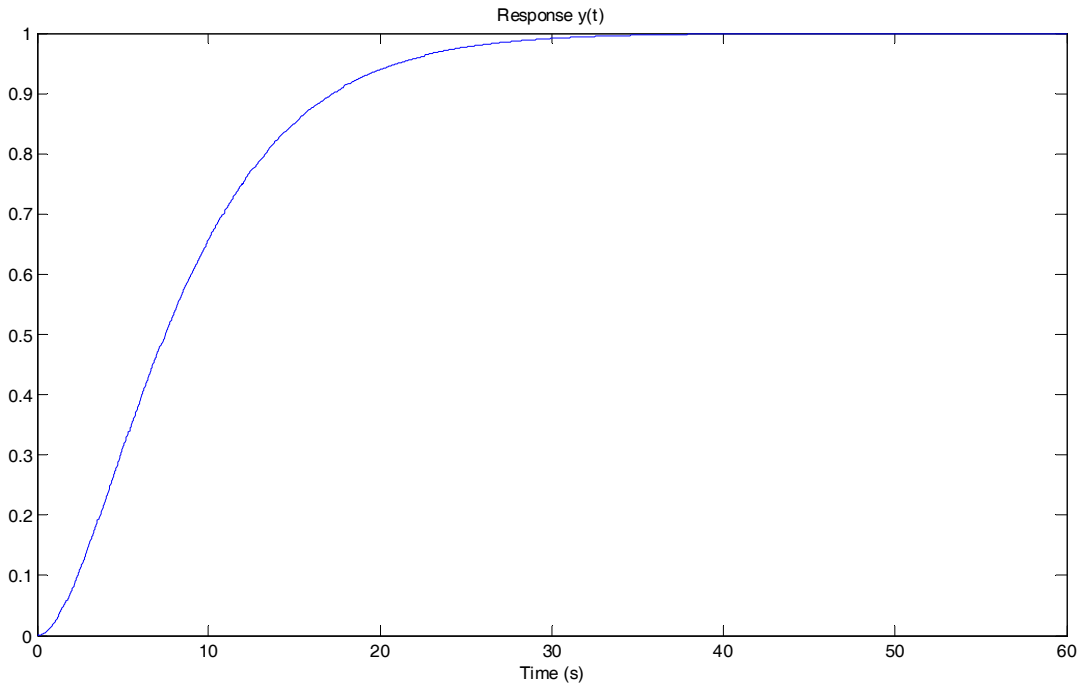

Figure 7: Response of the approximate 1st order system with proportional gain K.

Therefore, the approximate value of the settling time T_a ($\pm 5\%$) is given by the relation of Eq. (14):

$$T_a \cong \frac{3}{\sigma} \tag{14}$$

where σ is the real part of the 'dominant' pole shown in Figure 6. In this case:

```
>> Ta=3/0.22

Ta =

   13.6364

>>
```

Of course, this value would be the exact settling time of the response, if the overall system were described by a 1st order model. However, in order to determine the actual value of the settling time, the scheme of Figure 5 leads to the following results:

```
>> lsiminfo(ync,t)

ans =

    SettlingTime: 26.0718
             Min: 0
         MinTime: 0
             Max: 1.0000
         MaxTime: 60

>>
```

As already remarked, the achieved results show also that a simple proportional controller $R(s) = K$ is unable to compensate the considered system in order to obtain the required performance. In particular, the settling time cannot be further reduced to meet the requirements. Therefore, the control strategy proposed in this lecture has to rely on a dynamic controller, and in particular on a phase lead network, since $G(s)$ has three poles. Therefore, the following phase lead network $R(s)$ is proposed:

$$R(s) = K\frac{1+s/10}{1+s/150} \tag{15}$$

In this way, the choice of the pole of $R(s)$, $s_p = -150$, does not modify the asymptotic development of the root locus of $G(s)$, whilst its zero $s_z = -10$ is properly selected to suitably modify the development of the root locus closest to the imaginary axis. The MATLAB code below defines the phase lead network for $K = 1$:

```
>> s=tf('s')

>> Rs=(1+s/10)/(1+s/150)

>> [numRs,denRs]=tfdata(Rs,'v')
```

The closed loop gain function $G_a(s) = R(s)G(s)$ is defined in MATLAB as:

```
>> Ga=Rs*Gs
```

whose root locus is depicted in Figure 8.

```
>> rlocus(Ga)
```

Figure 8 highlights how the root locus is 'attracted' by the zero placed in $s_z = -10$, thus leading to decrease the overall settling time, as depicted by the vertical blue line corresponding to the points with constant T_a, and described by the relation of Eq. (16):

$$T_a = \frac{3}{\sigma} = \frac{3}{\delta \omega_n} \tag{16}$$

With respect to the uncontrolled system $G(s)$, Figure 8 shows the advantages of the use of the proposed phase lead network of Eq. (15), which leads to improve the achievable settling time. In fact, the 'deformation' of the root locus of $G_a(s)$ due to the attraction effect of the network zero reduces the settling time of the closed loop system, depending on the choice of the network DC gain K. As the modified root locus is able to increase the absolute value of the real part of the poles (σ or $\delta \omega_n$) by changing the value of K, this reduces the corresponding value of the settling time T_a, as highlighted by the relation of Eq. (16).

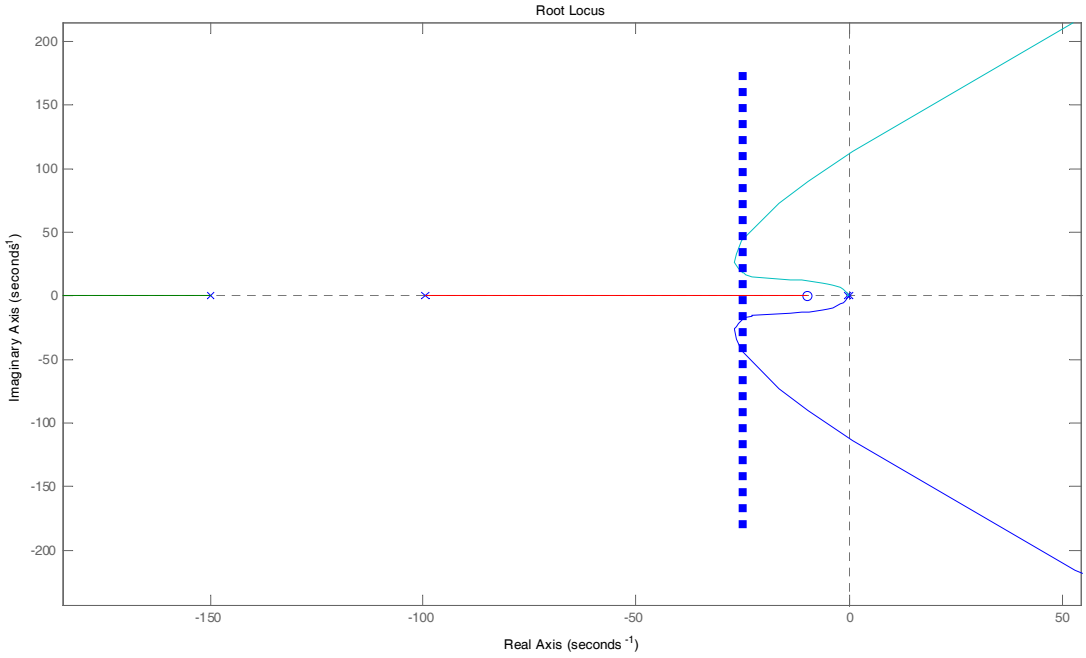

Figure 8: Root locus of the transfer function $G_a(s) = R(s)G(s)$.

On the other hand, by considering again the root locus of $G_a(s)$, the locus of the points with constant damping factor δ can be obtained from the following MATLAB code:

```
>> rlocus(Ga)
```

```
>> sgrid
```

The plot is depicted in Figure 9. In particular, the MATLAB command `sgrid` generates the s-plane grid lines for the considered root locus or pole-zero map. In this way, it generates the required grid over the existing continuous s-plane root locus. The lines of constant damping ratio δ and natural frequency ω_n are thus drawn, as shown in Figure 9.

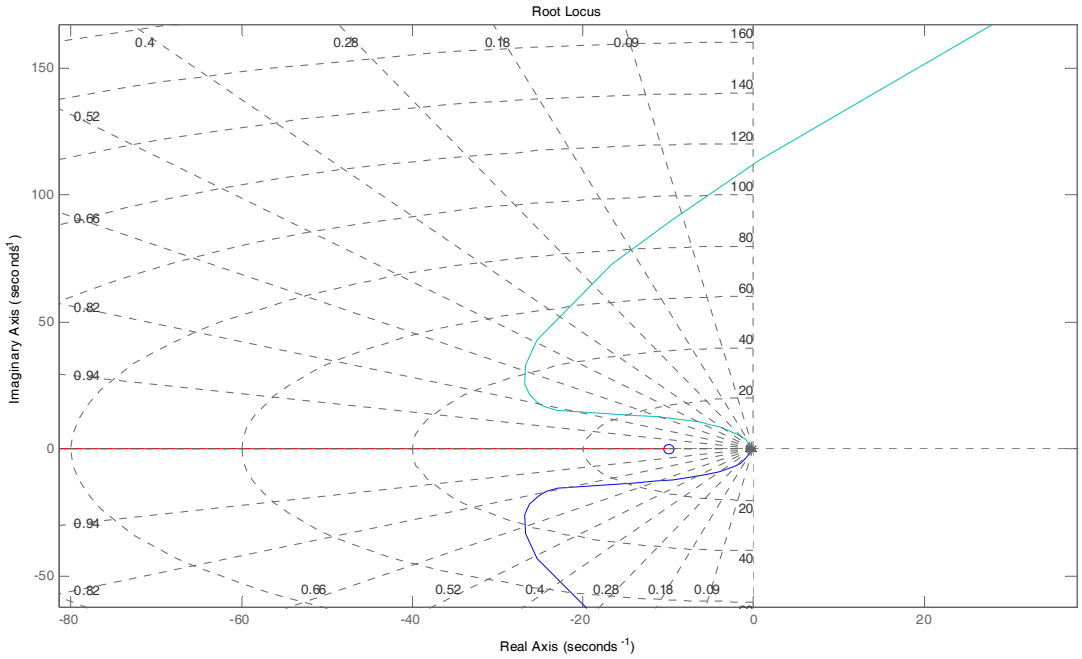

Figure 9: Root locus of $G_a(s)$ with constant damping ratio δ and natural frequency ω_n.

Moreover, Figure 10 shows the detail of Figure 9 for the required values of $\delta \geq 0.3$.

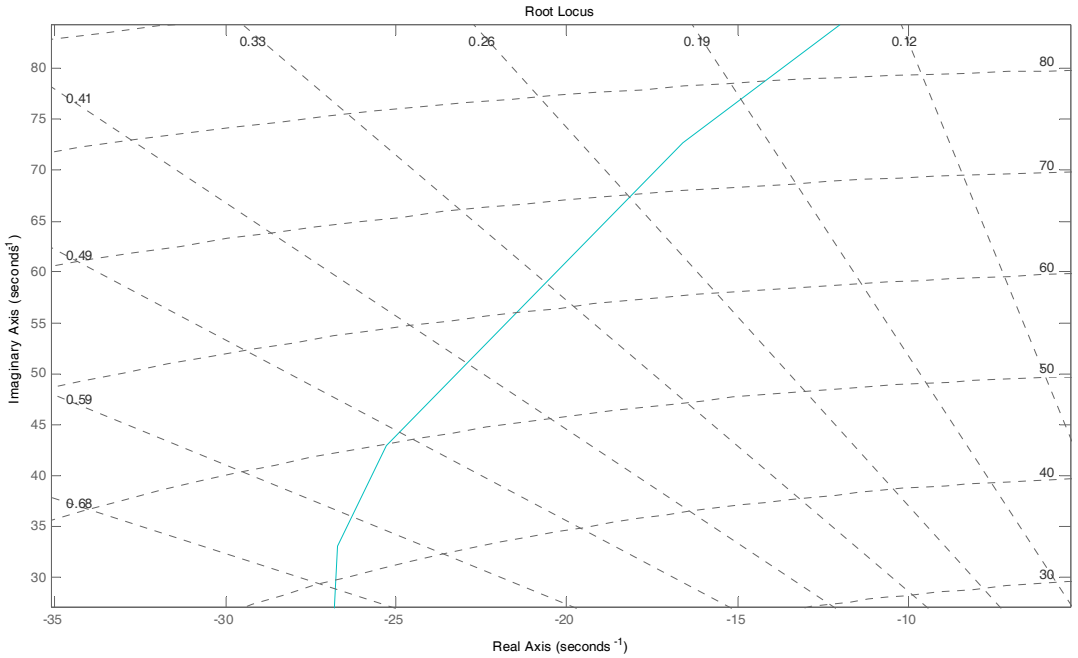

Figure 10: Detail of the root locus for the required values of $\delta \geq 0.3$.

By means of the use of the command `rlocfind`, a tentative value of the gain K can be determined with the crosshair cursor selected with the mouse for a pole of the root locus of $G_a(s)$ near to $\delta \geq 0.3$.

```
>> K=rlocfind(Ga)
```

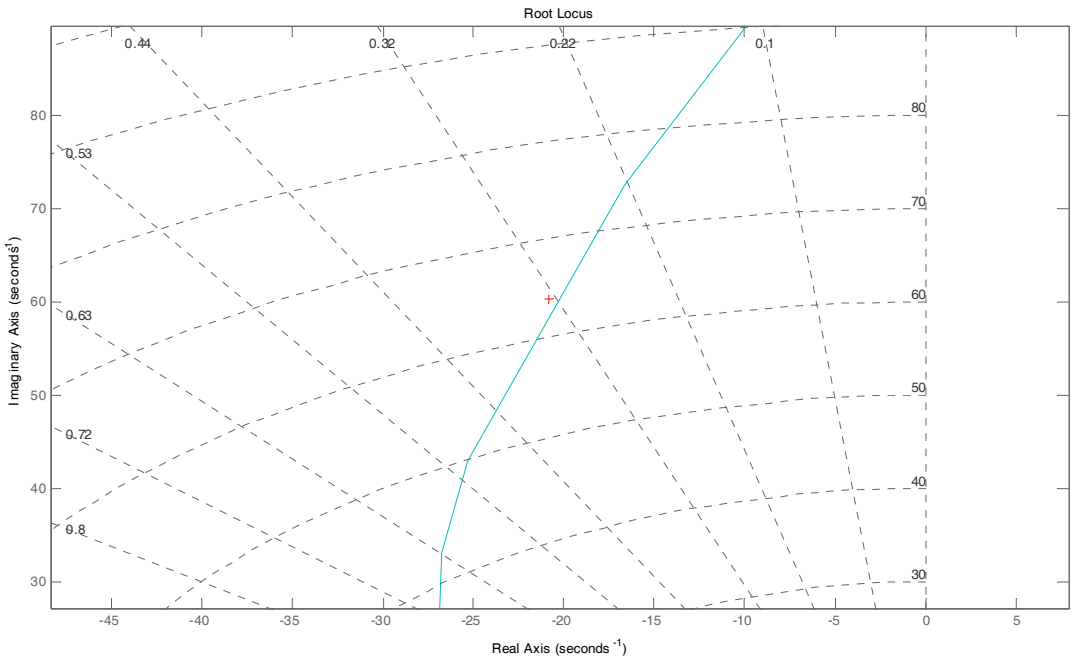

Figure 11: Tentative value of the gain K.

```
Select a point in the graphics window

selected_point =

 -19.6907 +62.0544i


K =

  2.9492e+003

>>
```

By reducing the simulation time to 5s, the characteristics of the transient response of the controlled system are reported in the following, obtained again by means of the command `lsiminfo`:

```
>> lsiminfo(yc,t)

ans =

    SettlingTime: 0.1946
             Min: 0
         MinTime: 0
             Max: 1.4813
         MaxTime: 0.0569

>>
```

It can be noted that the settling time requirement is now verified ($T_a < 0.5s$), but the maximum overshoot $S\%$ is higher than 48%. In general, the settling time T_a decreases ($\downarrow$) by increasing ($\uparrow$) the proportional gain K, whilst the maximum overshoot $S\%$ decreases ($\downarrow$) when the proportional gain K is reduced ($\downarrow$). These general 'meta-rules', which can be exploited for the design of a phase lead or lag network, are summarised in Table 1.

Table 1: Meta-rules highlighting the relation between gain, settling time and maximum overshoot.

K	$\uparrow$	$\downarrow$
T_a	$\downarrow$	$\uparrow$
$S\%$	$\uparrow$	$\downarrow$

In this case, as the actual maximum overshoot is higher than required ($S\% \geq 48\%$), the network gain K has to be reduced. A second tentative value is thus selected, $K = 2000$, smaller than the previous value, by using a trial and error procedure.

```
>> K = 2000

K =

        2000

>> lsiminfo(yc,t)

ans =
```

```
    SettlingTime: 0.2383
             Min: 0
         MinTime: 0
             Max: 1.3824
         MaxTime: 0.0747

>>
```

Also for this value, the settling time is verified ($T_a < 0.5s$), but the maximum overshoot is too large ($S\% > 35\%$). A further and lower value of the gain K is thus tried:

```
>> K = 1500

K =

        1500

>> lsiminfo(yc,t)

ans =

    SettlingTime: 0.2379
             Min: 0
         MinTime: 0
             Max: 1.3335
         MaxTime: 0.0802

>>
```

which finally allows to satisfy the required performances, i.e. $T_a \cong 0.24s > 0.5s$ and $S\% \cong 33\% < 35\%$. This concludes the design of the phase lead network.

The final response of the closed loop system implemented in Simulink as sketched in Figure 12 is shown in Figure 13.

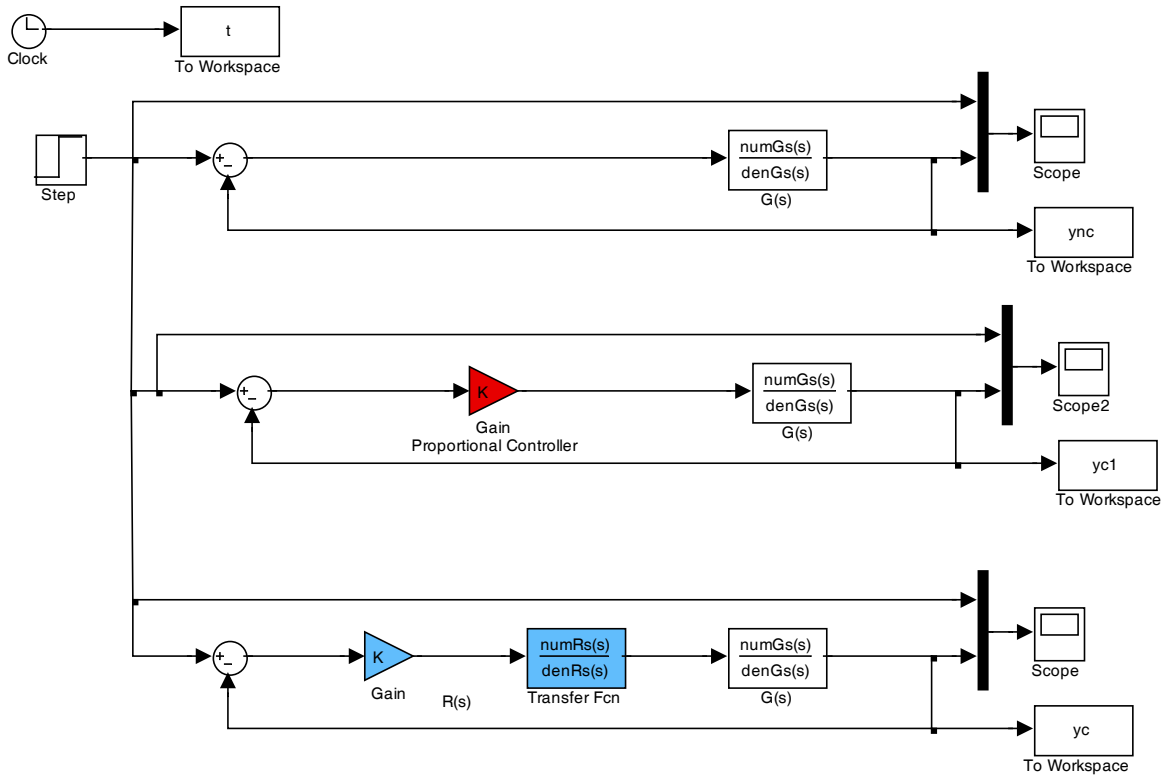

Figure 12: Overall Simulink scheme of the controller design.

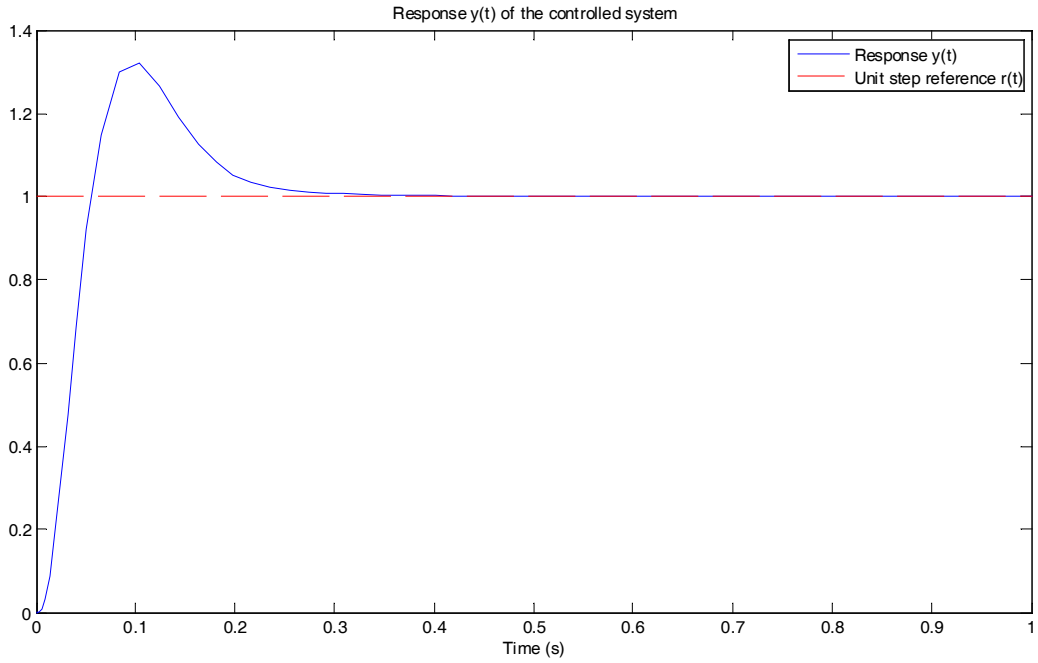

Figure 13: Response of the controlled system and its reference signal.