A Matlab® interactive tool for computer aided control systems design in frequency domain: FRTool

Robin De Keyser and Clara Ionescu
Ghent University, Electrical energy, Systems and Automation,
Technologiepark 913, B9052 Gent, Belgium

1. Introduction

Looking back at the history of control engineering, one finds that technology and ideas combine themselves until they reach a successful result, over the timeline of several decades (Bernstein, 2002). Simple dynamical compensators (such as PID, Phase-Lead, Phase-Lag, etc) have done in the past a remarkably good and efficient job in real-life control applications. Theoretical insight in the closed-loop behavior is provided by powerful and well-developed theories – which are considered as basic knowledge for every control engineer – such as the Root Locus and the Frequency Response techniques. In the past, these theories have been extensively used as *analysis tools* – e.g. the stability analysis of a given control system based on closed loop poles (Root Locus - RL) or Nyquist criterion (Frequency Response - FR). However, analysis implies *that a controller is already available*, irrespective of its design method.

Nowadays, thanks to the computational and graphical power of modern computers, many of these theories can be implemented as interactive graphical design tools. In this way, control engineering moves away from being an abstract and mathematical-oriented discipline and it evolves gradually towards a mature engineering discipline. An extensive reflection upon the role of information science in control engineering has been given in (Dormido, 2004). This new way of interactive control education provides practical insights into control systems fundamentals (Wittenmark *et al*, 1998; Dormido *et al*, 2002). Such combinations of interactive environment and animation bring visualization to a new level and aid learning and active participation by control engineering students (Kheir *et al*, 1996; Johansson *et al*, 1998).

Recently, a Root Locus RL toolbox has been introduced in Matlab – in this chapter, a Frequency Response toolbox (FRtool) will be presented. Although some other computer aided design (CAD) tools based on frequency response have been developed in the past (Balakrihsnan & Boyd, 1994; Satoh *et al*, 1994; Piguet & Gillet, 1997; Pouliz & Pouliezos, 1997; Piguet *et al*, 1999), the one presented in this contribution is highly interactive, graphical, easy-to-use and posing an elegant simplicity (especially for non-experts, as limited control engineering insight is required). The Matlab® controller design tool based on the root locus method (*rltool*) cannot handle systems with time-delay without approximating the dead-time by a rational transfer function. To tackle this problem, the FRtool operates with frequency diagrams (Nichols charts) – and in this way, the dead-time can be treated without any approximation.

The paper is structured as follows: the FRtool graphical interface is depicted in the next section, followed in the 3rd section by a description of the underlying frequency response concepts. In the 4th section, four examples are given:

i) a time-delayed and integrating transfer function of a (chemical) process;

ii) a second order system (velocity control);

iii) a mass spring damper system, defined by a 4th order transfer function with poor damping factors and

iv) a high order system (6th order)

A final section concludes this contribution.

2. FRTool Graphical Interface

Probably the most important feature of the FRtool is the user-friendly graphical interface (drag & drop and zoom included). It also has the possibility to display design specifications as graphical restrictions on the Nichols plot – including a real-time update while dragging controller's poles and zeros. It can also import/export process and controller from or to the Matlab workspace and has options to print Nichols, Nyquist or Bode curves and closed-loop responses, as depicted in figure 1.

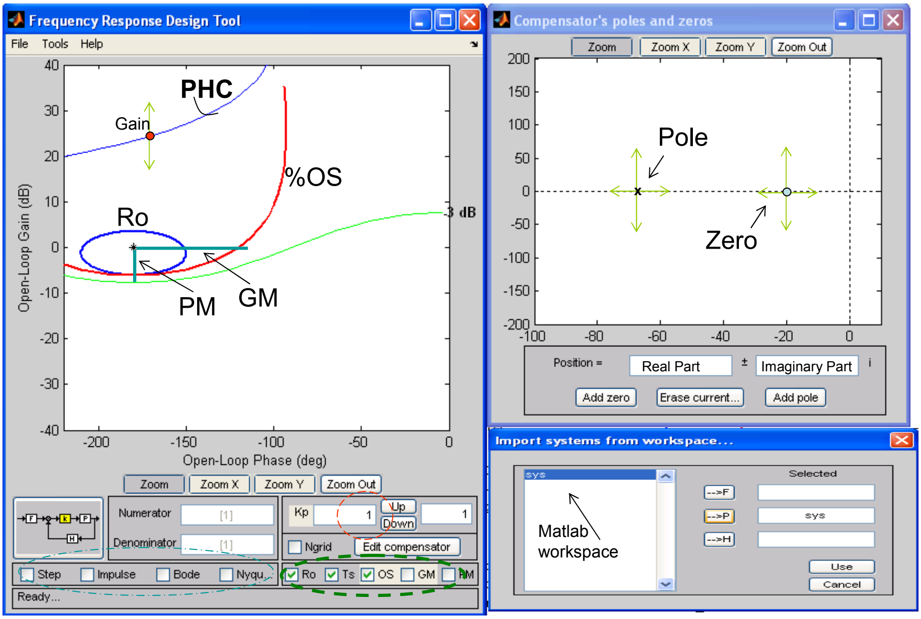


Fig. 1. Graphical interface of FRtool. After the system has been imported from Matlab workspace (see window in the lower right part), it appears as a curve in the Nichols chart corresponding to the loop frequency response (PHC).

The design specifications can be introduced using the options denoted in the bolded dashed green line; e.g. the overshoot ($\%OS$) and the robustness (Ro) specifications are visible in the chart. Closed-loop performance can be evaluated with the options in the dashed-dotted blue line. The upper right window in figure 1 is used to design the compensator by dragging compensator's poles and zeros with the mouse. Additionally, pre-filter $F(s)$ or feedback $H(s)$ transfer functions can be added within the control scheme. In order to design a controller with FRtool, the user does not need a detailed knowledge of the frequency-response background theory. Tools lessen the need for prior insights, but should provide a better understanding of the problem to let the user make better decisions. In figure 1, the PHC-curve corresponds to the loop Nichols curve, with a default controller equal to 1 (and in this example also H=1). Notice that the PHC-curve is above the critical point in the Nichols chart: -1+j0 – denoted by a *star* corresponding to an open-loop gain of 0dB and phase -180°. Consider now figure 2 and a transfer function available in the Matlab workspace:

$$P(s) = \frac{2500}{s(s+25)} e^{-0.2s} \tag{1}$$

It is obvious that transfer function (1) denotes a system which is not trivial to control: it is marginally stable due to the integrator and it has a significant dead-time, two aspects which make the control design a challenging task.

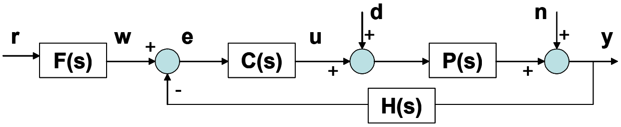
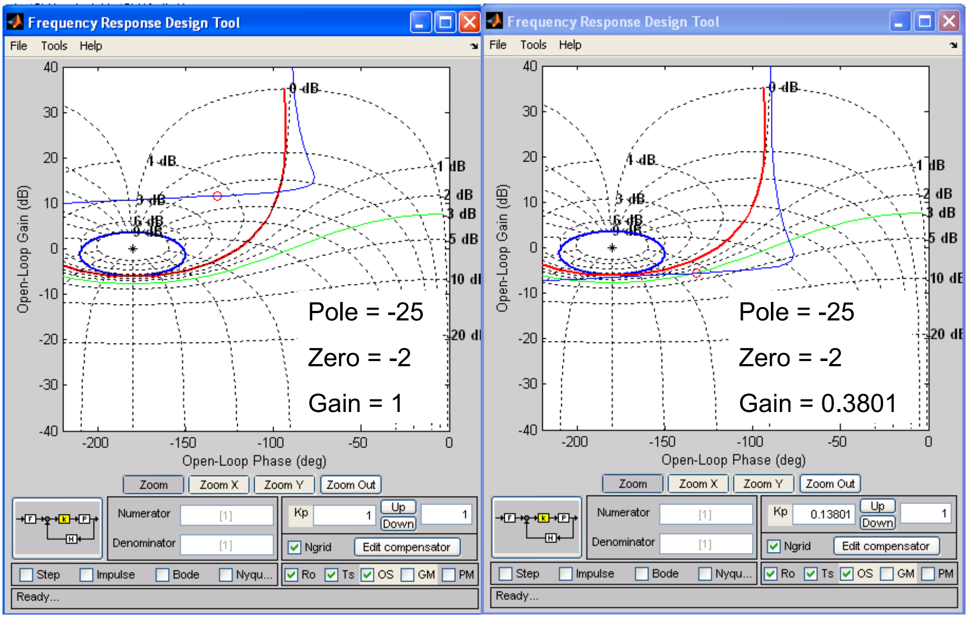


Fig. 2. General scheme of a control loop: r – reference, w – setpoint, e – error, u – the manipulated variable, d and n – disturbances, y – the controlled variable, F(s) is a pre-filtering transfer function and H(s) is a feedback transfer function; C(s) and P(s) denote the controller, respectively the process transfer function.

When imported in FRTool, the process from (1) with controller C=1 gives the result as depicted in figure 1. One may observe that the closed loop system will be unstable, and the user has then to *play with the controller's poles and zeros* to fulfill the specifications by pure visual inspection in the Nichols chart. Figures 3 and 4 depict various situations of zeros-poles placement, affecting the shape of the Nichols curve. Additionally, the user may choose to plot the Nichols grid (see dotted grid in figures 3-4), denoting the circles of Hall in the Nyquist plane (Nise, 1995). This grid shows the relationship between open-loop and closed-loop frequency responses.



Pole = -25

Zero = -2

Gain = 1

Numerator [1]

Denominator [1]

Kp 1 Up Down 1

☑ Ngrid

☐ Step ☐ Impulse ☐ Bode ☐ Nyqu... ☑ Ro ☑ Ts ☑ OS ☐ GM ☐ PM

Pole = -25

Zero = -2

Gain = 0.3801

Numerator [1]

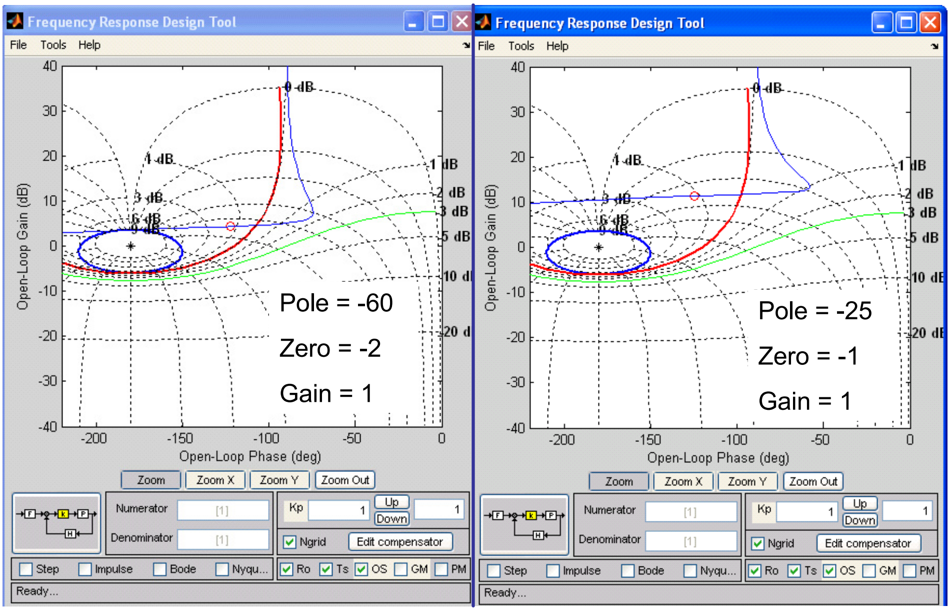
Denominator [1]

Kp 0.13801 Up Down 1

☑ Ngrid

☐ Step ☐ Impulse ☐ Bode ☐ Nyqu... ☑ Ro ☑ Ts ☑ OS ☐ GM ☐ PM

Fig. 3. Graphical interface of FRtool. It can be observed that the change in gain produces only a translation of the Nichols curve along the vertical axis, while its shape (dynamics) is influenced by the position of the controller's poles and zeros.



Pole = -60

Zero = -2

Gain = 1

Numerator [1]

Denominator [1]

Kp 1 Up Down 1

☑ Ngrid

☐ Step ☐ Impulse ☐ Bode ☐ Nyqu... ☑ Ro ☑ Ts ☑ OS ☐ GM ☐ PM

Pole = -25

Zero = -1

Gain = 1

Numerator [1]

Denominator [1]

Kp 1 Up Down 1

☑ Ngrid

☐ Step ☐ Impulse ☐ Bode ☐ Nyqu... ☑ Ro ☑ Ts ☑ OS ☐ GM ☐ PM

Fig. 4. Graphical interface of FRtool. Referenced to the pole/zero positions in figure 3, this picture illustrates the effect of a change in the pole position (left) and in the zero position (right).

3. Design Specifications

An important feature of a controller design tool is the possibility to define practically-meaningful design specifications – which will guide the designer in the tuning process. These specifications have to be converted to graphical restrictions to make the designer's job easier. Some of the *traditional* design specifications are gain margin and phase margin (Nise, 1995). However, these specifications have not necessarily a clear physical meaning to a potential user (unless this user is e.g. a control engineer) – they are based on mathematical insight and system theory. Therefore, more *practical* specifications - which can be easily interpreted by any user - are settling-time and overshoot of the closed-loop time response, and of course robustness of the design (Nise, 1995). In this section, the background theory on which these specifications are taken into account in FRtool is explained with reference to their graphical equivalents. The basic (textbook) transfer function of a PID is:

$$C(s) = K_p(1 + \frac{1}{T_i s} + T_d s) = K\frac{(s - z_1)(s - z_2)}{s} \tag{2}$$

with K_p the proportional gain, T_i the integrative and T_d the derivative constants, K the equivalent controller gain in zero-pole chart, z_1 and z_2 zeros. Similarly, in FRtool a *general form* of the compensator (filter) can be defined as the following transfer function:

$$C(s) = K\frac{(s - z_1)(s - z_2)...}{(s - p_1)(s - p_2)...} \tag{3}$$

Robustness is probably the most important design specification in practice. Since the controller tuning is done around a nominal operating point, the control performance should still be acceptable irrespective of model changes (e.g. in chemical plants) or model-mismatches (e.g. in mass-produced mechatronic devices such as DVD-players, hard-disks, etc). Denote $G(s)$ as the (open-)loop transfer function and $T(s)$ as the closed-loop transfer function, then the closed-loop sensitivity is defined as the closed-loop relative change over the open-loop relative change (the controller does not change, but the process may):

$$\frac{\partial T(s)/T(s)}{\partial G(s)/G(s)} = \frac{dT(s)}{dG(s)} \cdot \frac{G(s)}{T(s)} = \frac{1}{1 + G(s)} \tag{4}$$

Thus the sensitivity should be as small as possible, resulting in $|1 + G(s)|$ to be as big as possible. The equivalent representation in the complex plane is given in figure 5.

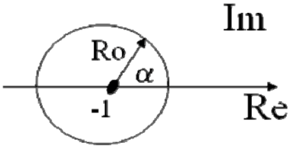


Fig. 5. Graphical representation of the robustness-function.

Robustness is described as how in-sensitive a closed-loop is to changes in the process. In FRtool, robustness (*Ro*) is a design parameter specified by the user, with value 0<*Ro*<1. It has the meaning of $|1+G(j\omega)|$, thus a larger *Ro*-value means more robustness. To fulfill the robustness spec, the Nyquist curve has to stay outside the circle with radius *Ro*.

Every point of this circle can be translated into the Nichols chart, leading to the ellipse-type blue curve in figures 2, 3 and 4. The inner part of this ellipse denotes a forbidden zone for the Nichols curve (with $\alpha = 0 \cdots 360°$):

$$\begin{aligned} &\mathrm{R} = -1 + Ro * \cos(\alpha) \qquad \mathrm{I} = Ro * \sin(\alpha) \\ &M_{dB} = 20\log\sqrt{\mathrm{R}^2+\mathrm{I}^2} \quad \Phi_{\deg} = -180 + \arctan\left(\frac{\mathrm{I}}{\mathrm{R}}\right) \end{aligned} \tag{5}$$

The other two specifications – overshoot (%*OS*) and settling time (*T*$_s$) – are derived from the dominant second order closed-loop transfer function:

$$T(s) \cong \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{6}$$

From (6) the overshoot and settling time in time-domain are obtained via the damping factor (ξ) and the natural frequency (ω_n). Further on, given the user-defined specifications for %*OS* and *T*$_s$, it is straightforward to obtain ξ and ω_n from:

$$\begin{aligned} &\%OS = 100e^{-\xi\pi/\sqrt{1-\xi^2}} \\ &T_s \cong \frac{4}{\xi\omega_n} \end{aligned} \tag{7}$$

Having (7), the time-domain specifications can be translated into frequency-domain specifications by using the following formulas (Nise, 1995):

$$\begin{aligned} &M_p = \frac{1}{2\xi\sqrt{1-\xi^2}} \\ &\omega_{BW} = \omega_n\sqrt{1-2\xi^2+\sqrt{4\xi^4-4\xi^2+2}} \end{aligned} \tag{8}$$

These parameters correspond then to the maximum closed-loop magnitude M_p (which is related to overshoot specification) and the closed-loop bandwidth ω_{BW} (which is related to the settling time specification).

From (7) and (8) it results that the designed controller should induce in the closed loop transfer function a peak-magnitude smaller than M_p, and the -3dB-bandwith frequency should be bigger than ω_{BW}. Notice in figure 2 the representation of M_p by the red curve indicated by %*OS*; the Nichols curve must stay below this %*OS* curve. The frequency ω_{BW} is denoted by the little red circle on the Nichols curve in figure 2; it must be above the green -3dB line in order to fulfill the specification.

4. Illustrative Examples

4.1 First Order Plus Integrator and Time-Delay

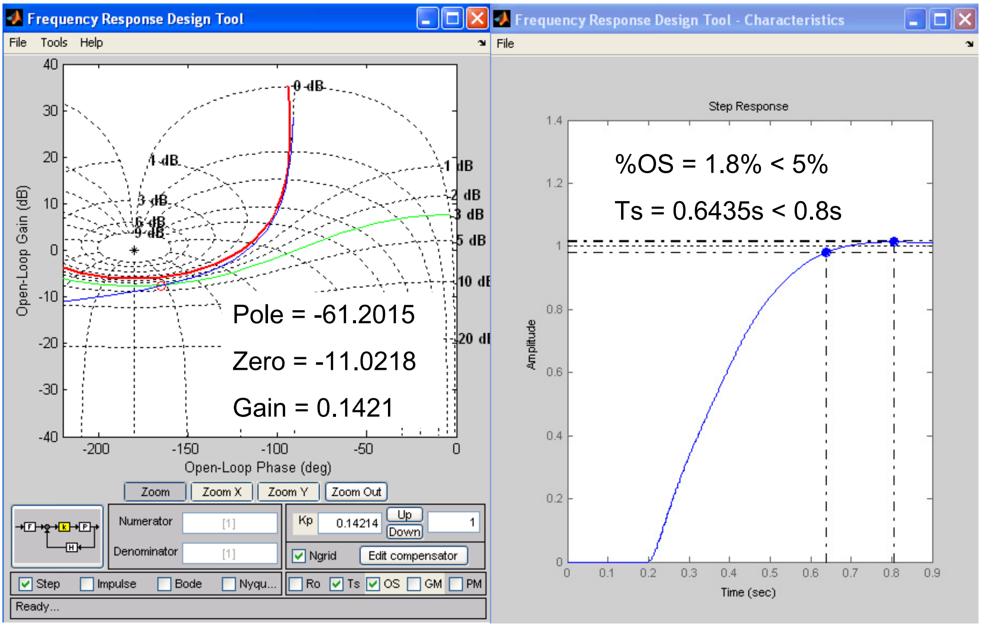
Given the challenging process (1), design a phase-lead controller so that the closed-loop will satisfy the specifications for overshoot $\%OS < 5\%$ and settling time $T_s < 0.8s$. Defining the system in the Matlab command window:

```
sys=tf(2500,[1 25 0],'InputDelay',0.2)
```

importing it in the FRtool P-block (as Process transfer function) and defining the given specifications, the Nichols curve will be the one shown in figure 2 (blue line), which represents the process and controller C=1. Suppose the user wants to design a phase-lead controller, whose transfer function is defined by:

$$C(s) = K\frac{s-z}{s-p} \tag{9}$$

where the condition $|p| > |z|$ is satisfied. The optimal situation with the corresponding phase-lead controller design that satisfies the specifications is given in figure 6 along with the corresponding step response of the closed-loop.



Pole = -61.2015

Zero = -11.0218

Gain = 0.1421

%OS = 1.8% < 5%

Ts = 0.6435s < 0.8s

Numerator [1]

Denominator [1]

Kp 0.14214

☑ Ngrid

☑ Step □ Impulse □ Bode □ Nyqu...

□ Ro ☑ Ts ☑ OS □ GM □ PM

Ready...

Fig. 6. Optimal phase-lead controller design fulfilling the required specifications and its step response for first example.

4.2 Second Order System

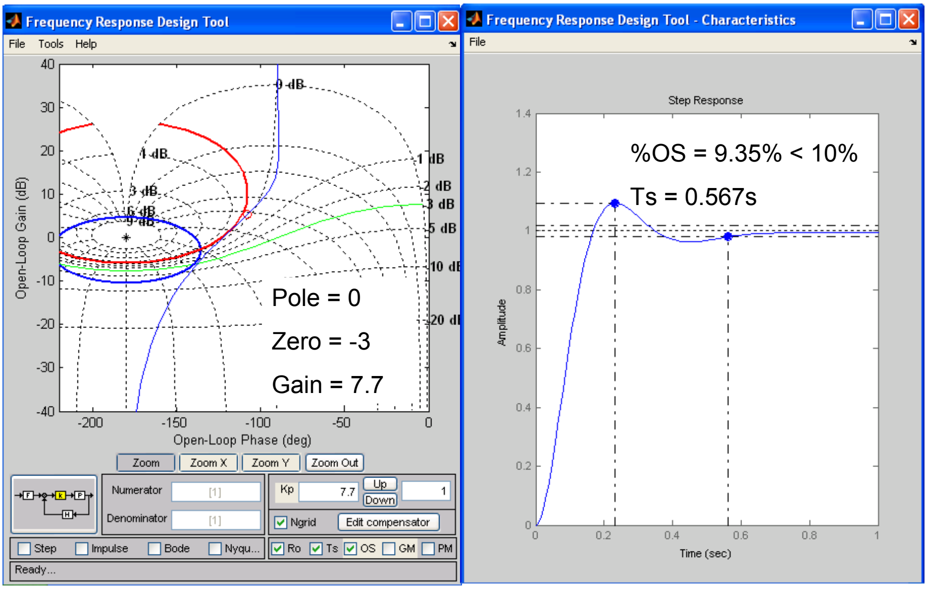
A second example is given to illustrate the robustness property of the designed controller. Consider a flight control system (angular) velocity control (it can be as well a servo-control system, antenna, disk drive, DVD, etc) represented by:

$$P(s) = \frac{32}{(s+4)(s+16)} \tag{10}$$

and the following specifications for a PI-controller design: robustness Ro>0.7 (a reasonably high robustness specification on a scale 0…1); overshoot %OS<10% and a *minimum* settling time T_s. The optimal PI-controller design and its corresponding step response are given in figure 7. The robustness can be evaluated in case the process model changes (the controller is fixed). To illustrate this case, consider the following transfer function, assuming that (10) has changed:

$$P^*(s) = \frac{48}{(s+5)(s+12)} \tag{11}$$

As it can be observed from (11), the changes do not affect only the gain (+50%), but also change the dynamics of the process quite significantly. In figure 8 can be evaluated the performance of the same PI-controller designed as in figure 7, but applied on the process model (11).



Pole = 0

Zero = -3

Gain = 7.7

%OS = 9.35% < 10%

Ts = 0.567s

Numerator [1]

Denominator [1]

Kp 7.7 1

☑ Ngrid

□ Step □ Impulse □ Bode □ Nyqu... ☑ Ro ☑ Ts ☑ OS □ GM □ PM

Fig. 7. Optimal PI-controller design fulfilling the required specifications and its step response for second example.

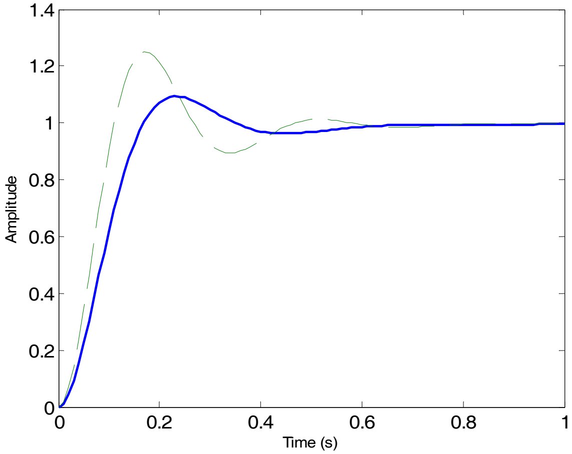


Fig. 8. Robustness evaluation of the PI-controller. The continuous line represents the nominal process model, and the dashed line represents the changed process model.

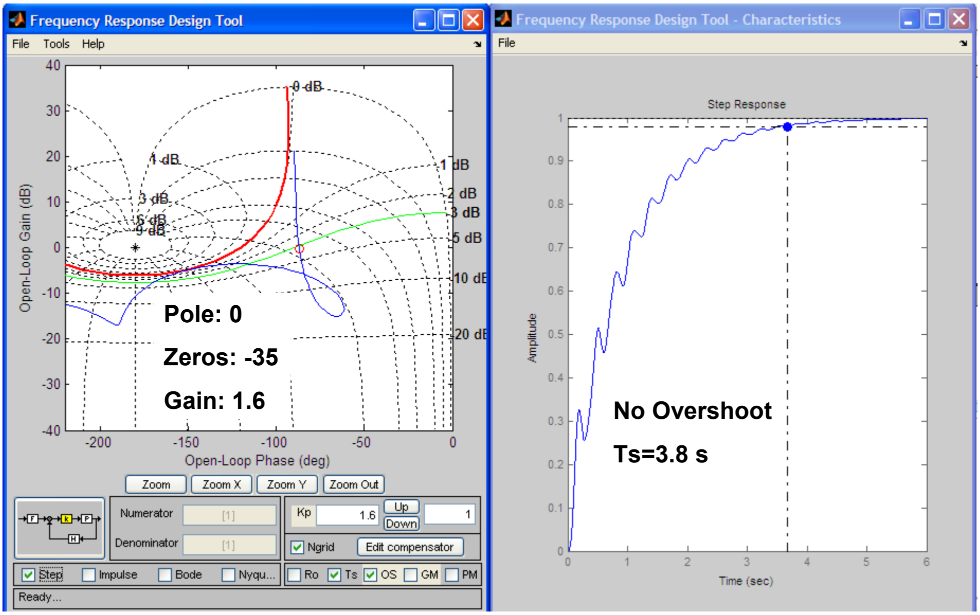
It can be concluded that the performance of the controller does not fulfill anymore the initial specifications (%OS>10%) but it has still a reasonably good behavior, taking into account the fairly large changes in the process transfer function. This is due to the fact that a relatively high robustness was asked in the specifications.

4.3 High Order + Poorly Damped System

Consider a mass-spring-damper system driven by an electrical motor, with two masses, three springs and one damper, given by the transfer function:

$$P(s) = \frac{800}{2.498s^4 + 16.65s^3 + 4473s^2 + 14400s + 1360000} \tag{12}$$

This system has two poorly damped eigenfrequencies $\omega_1 = 20.8$ rad/s and $\omega_2 = 39.1$ rad/s, with damping factors $\xi_1 = 0.08$ and $\xi_1 = 0.05$. Although this type of vibrational process should in practice be better controlled by a more sophisticated controller, it is possible to design with the CAD package, a suboptimal controller, of PID type. The result is then indicated in figure 9.



Pole: 0

Zeros: -35

Gain: 1.6

Numerator [1]

Denominator [1]

Kp 1.6 1

☑ Ngrid

☑ Step ☐ Impulse ☐ Bode ☐ Nyqu... ☐ Ro ☑ Ts ☑ OS ☐ GM ☐ PM

Ready...

No Overshoot

Ts=3.8 s

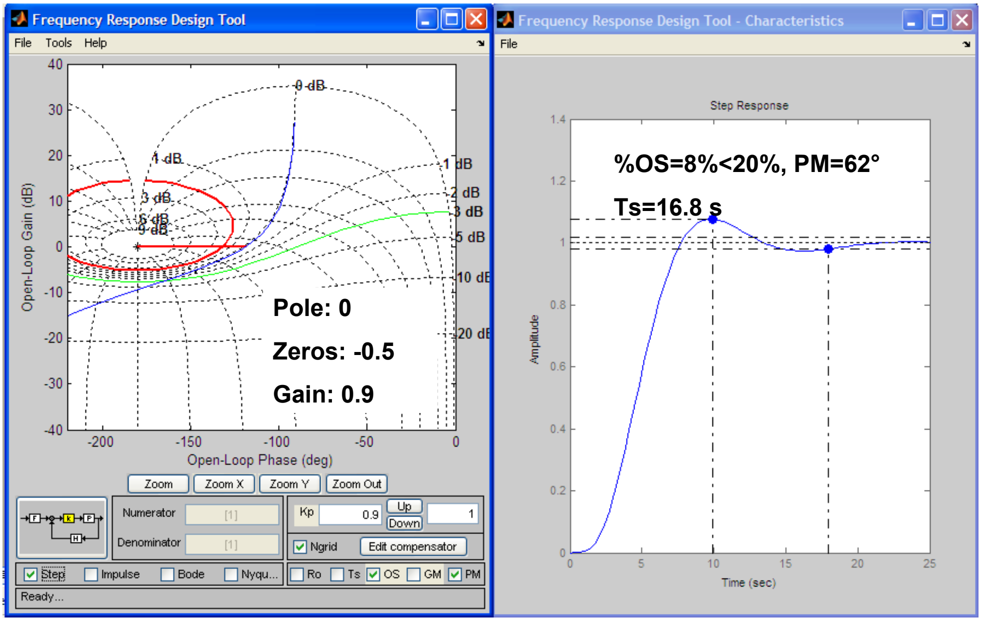
Fig. 9. Optimal PID-controller design fulfilling the required specifications and its step response for the high order plus poorly damped example.

4.4 High Order System

Consider the high order system presented by:

$$P(s) = \frac{1}{(s+1)^6} \tag{13}$$

and the specifications: phase margin *PM*=62°, overshoot %OS<20% and a the settling time T_s as small as possible. The PID controller design in FRTool is given in figure 10, along with the step response.



Pole: 0

Zeros: -0.5

Gain: 0.9

%OS=8%<20%, PM=62°

Ts=16.8 s

Numerator [1]

Denominator [1]

Kp 0.9 1

☑ Ngrid

☑ Step □ Impulse □ Bode □ Nyqu... □ Ro □ Ts ☑ OS □ GM ☑ PM

Fig. 10. Optimal PID-controller design fulfilling the required specifications and its step response for the high order example.

5. Conclusions

In this contribution, a novel controller design toolbox has been presented for the Matlab environment. It is based on the frequency response (Nichols chart) representation of systems and entitled FRtool – Frequency Response toolbox. The toolbox can perform controller design with given *practical* specifications such as overshoot, settling time, robustness (and also 'classical' gain and phase margins as an option). It can also deal with time-delay systems and it has been illustrated on four examples, of which two examples far from being second order systems (6th order, respectively, 4th order with poor damping).

Notice that the design procedure is extremely simple and close-at-hand, given the highly interactive graphical user interface of FRtool. Insight in all details of the underlying control engineering principles is not really required in order to use the CAD tool. This makes the toolbox attractive to students and engineers who are not control experts, and it offers an elegant solution for obtaining satisfactory results.

The CACSD tool has been used during five years by several hundreds of engineering students – not specializing in control engineering – in a *basic* control engineering course. The experience is that the majority of them are able to manage the tool and produce a good controller design within about fifteen minutes.

The software package is a development of Ghent University EeSA department; it consists of a set of Matlab files and is freely provided by email request.

6. References

Balakrihsnan, V. & Boyd S. (1994) Trade-offs in frequency-weighted H_∞-control, *Proceedings of the IEEE/IFAC Joint Symp. Computer Aided Control Systems Design,* Arizona, pp. 469–474

Bernstein, D. (2002) Feedback control: an invisible thread in the history of technology, *IEEE Ctrl Syst Mag*, 22(2), pp. 53-68

Dormido, S., Gordillo, F., Dormido-Canto, S. & Aracil J. (2002) An interactive tool for introductory nonlinear control systems education, *Proceedings of the 15th. IFAC World Congress* b'02, Barcelona, Spain

Dormido, S. (2004) Control Learning: Present and Future, *Annual Review in Control*, **28**, pp. 115-136

Johansson, M., Gäfvert, M., Åström, K. (1998) Interactive tools for education in automatic control, *IEEE Control Systems Magazine*, 18, pp. 33-40

Kheir, N., Åström, K., Auslander, D., Cheok, K., Franklin, G., Masten, M. & Rabins M. (1996) Control system engineering education, *Automatica*, **32**, pp. 147-166

Nise, N. (1995) *Control Systems Engineering,* 2nd edition, Addison-Wesley, chapter 10

Piguet, Y., Gillet D. (1997) Java-based remote experimentation for control algorithms prototyping, *Proc. of the American Control Confrence*, San Diego, U.S.A. pp. 1465-1469

Piguet, Y., Holmberg, U., Longchamp, R. (1999) Instantaneous performance visualization for graphical control design methods, *14th IFAC World Congress*, Beijing, China, 6p.

Poulis, D., Pouliezos, A. (1997) Computer assisted learning for automatic control, *IFAC Symposium on Advances in Control Education*, Estambul, Turquía, pp. 181-184

Satoh, T., Ishihara, T. & Inooka, H. (1994) Computer-aided control system design accounting pole-zero cancellations by the method of inequalities, *Proceedings of the IEEE/IFAC Joint Symp. Computer Aided Control Systems Design,* Arizona, pp. 481–488

Wittenmark, B, Häglund, H., Johansson, M. (1998) Dynamic pictures and interactive learning, *IEEE Control Systems Magazine*, 18**, p**p. 26-32