

## **LTI System Control Toolbox (LTITOOL)**

INSTITUTION:

Cátedra de Instrumentación y Control de Procesos  
Departamento de Ingeniería de Procesos – FIQ – UNL

AUTOR:

Prof. Dr. Eduardo J. Adam  
mailto: eadam.fiq@gmail.com

AÑO: 2019

**Developers and Collaborators of this edition:**

EDUARDO J. ADAM, (eadam.fiq@gmail.com)  
Faculta de Ingeniería Química - Universidad Nacional del Litoral,  
Santa Fe, Santa Fe, Argentina.

**Special thanks:**

Special thanks to Ernesto S. Burgos for the development of the GUI-Editor and for  
for his selfless collaboration, which would otherwise it has made it impossible  
to achieve the GUIs here presented.

ERNESTO S. BURGOS, (e.sergio.burgos@gmail.com)  
Universidad Tecnológica Nacional - Facultad Regional Paraná,  
Paraná, Entre Ríos, Argentina

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation Mode</b>	<b>3</b>
2.1	Installing for the First Time . . . . .	3
2.2	Updating the Version . . . . .	3
<b>3</b>	<b>Open Loop LTI Systems</b>	<b>5</b>
3.1	Impulse response . . . . .	6
3.2	Step response . . . . .	8
3.3	Tool Developed for Open Loop Systems . . . . .	10
3.4	Conclusions . . . . .	13
<b>4</b>	<b>Closed Loop LTI Systems</b>	<b>15</b>
4.1	Tools for Feedback LTI Systems . . . . .	19
4.1.1	Analysis and Simulation Based on $G(s)H(s)$ . . . . .	19
4.1.2	Analysis and Simulation of Feedback System . . . . .	23
4.2	Conclusions . . . . .	28
<b>5</b>	<b>Traditional Advanced Control Strategies</b>	<b>29</b>
5.1	Feedforward-Feedback Control . . . . .	29
5.2	Cascade Control . . . . .	30
5.3	Tools for Traditional Advanced Control Strategies . . . . .	31
5.3.1	Analysis and Simulation of a Feedforward-Feedback Control System . . . . .	31
5.3.2	Analysis and Simulation of a Cascade Control System . . . . .	35
5.4	Conclusions . . . . .	38
<b>6</b>	<b>Additional Tools for LTI System Control</b>	<b>39</b>
6.1	Optimal Setting of PID Controller . . . . .	39
6.2	Developed Additional Tools . . . . .	39
6.2.1	PID Optimal Controller Setting . . . . .	39



# Chapter 1

## Introduction

This particular toolbox is designed to facilitate the teaching, learning and comprehension process of the linear control theory applied to linear time invariant (LTI) systems.

For this reason, this user manual makes use of classical theoretical concepts of linear control systems, based particularly on the work of Adam [1]. footnote Where the entire nomenclature is based on the cited text.

This toolbox has 5 sections where the it is possible study

1. Properties of open loop linear system as poles-zeros of transfer function and their relation to the dynamic responses to step and impulse.
2. a first introduction to the analysis of linear systems with unit feedback, which includes pole and zero maps of the open and closed loop system, root locus, frequency analysis (Bode and Nyquist diagrams), dynamic responses of the feedback system to step set-point changes.
- 3.
- 4.



## Chapter 2

# Installation Mode

### 2.1 Installing for the First Time

This toolbox are installed like any other Octave toolboxes. Consequently, the installation procedure is as detailed below.

**Step 1.** Download the file `ltitool.tar.gz` To download the file for the first time you must go the website `http://www.fiq.unl.edu.ar/control/index.php?page=software`.

**Step 2.** Run the following command in the directory where the file was downloaded.

```
>>> pkg install ltitool.tar.gz
```

In this way, the toolbox is already installed and ready to use. Then, write the following commands to execute the toolbox mentioned before:

```
>>> pkg load ltitool
>>> ltitool
```

### 2.2 Updating the Version

If you already have LTITOOOL installed, then when a new version is available, a window like the one is shown in Fig. 2.1.

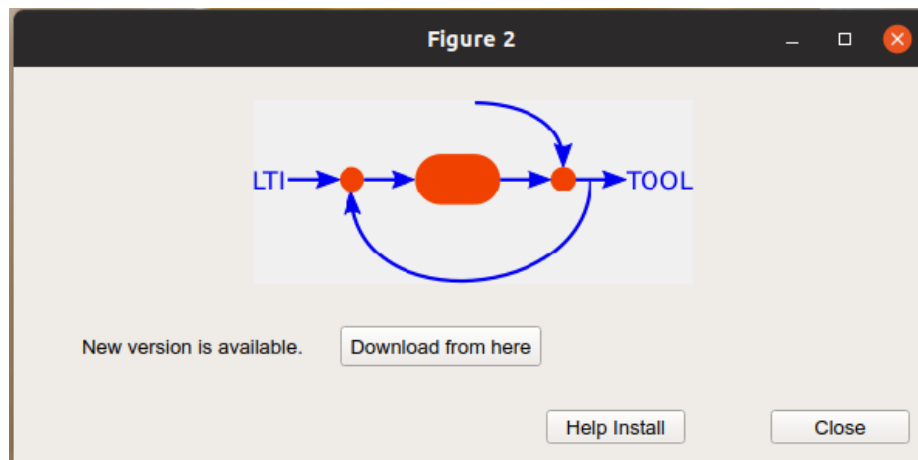


Figure 2.1: Updating to a new version.

Simply, you have to press the download button and then you have to follow the procedure indicated below.

**Step 1.** Uninstall the version you currently have installed. To do this, execute the command

```
>>> pkg uninstall ltitool.tar.gz
```

Alternatively, it might be convenient to close and reopen Octave, but this should not be necessary.

**Step 2.** Run the following command in the directory where the file was downloaded.

```
>>> pkg install ltitool.tar.gz
```

Finally, you have the new version installed and to execute it you must follow the steps indicated in the previous section.



## Chapter 3

# Open Loop LTI Systems

The transient response analysis carried out in this chapter is based on using a representation with transfer functions of the system output according to his input. The use of this particular representation is more intuitive for the process engineer, since it consists of a simple input-output representation, where if it take into account that for its study the input has been set a priori, the transient output can predicted immediately from the analysis of the system transfer function.

To do this, consider the open loop system of the Fig. 3.1.

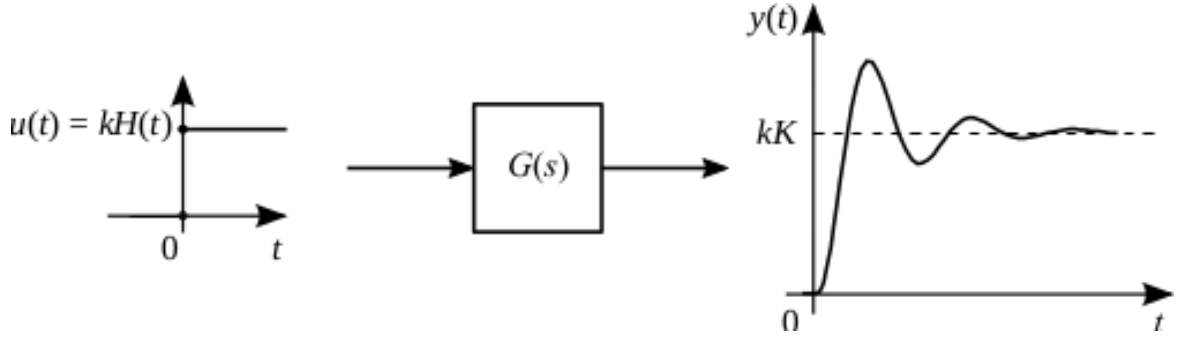


Figure 3.1: Open loop LTI system

where  $Y(s) = G(s)U(s)$  with  $G(s)$  a polynomial ratio written as,

$$G(s) = \frac{b_m s^m \dots b_1 s + b_0}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (3.1)$$

Note that the dynamic response turns out to be,

$$y(t) = \mathcal{L}^{-1} \left[ \frac{b_m s^m \dots b_1 s + b_0}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} U(s) \right] \quad (3.2)$$

where the available inputs in this application are step and impulse.

### 3.1 Impulse response

Impulse response is the system dynamic response subjected to the input signal equal to a Dirac delta input (denote as  $\delta(t)$ ).

Thus, being the temporary expression of an unit impulse  $u(t) = \delta(t)$ , its Laplace transform results  $U(s) = \mathcal{L}[u(t)] = \mathcal{L}[\delta(t)] = 1$  then,  $Y(s) = G(s)$  and the dynamic LTI system response is,

$$y(t) = \mathcal{L}^{-1}[G(s)] . \quad (3.3)$$

In conclusion, *the impulse response of a linear system is equal to the inverse transformation of its plant transfer function.*

#### Example 1

Consider a linear system whose transfer function is,

$$G(s) = \frac{(s+1)}{s^2 + 0.5s + 1} \quad (3.4)$$

subject to a Dirac Delta input. We pretend to obtain the dynamic system response subject to such excitation.

To do this, before carrying out a simulation, we will make a study based on the transfer function of the system.

1. **Stability:** It is possible to determine this based on the poles of the system transfer function, in this case under open loop. To do this, using Octave, you can determine these poles using the following commands:

```
% System transfer function
s=tf('s');
Gs=(s+1)/(s^2+0.5*s+1);

% Poles and Zeros
polesGs=roots(Gs.den{1,1})
```

La respuesta en la ventana de comandos de Octave es:

```
polesGs =

-0.25000 + 0.96825 i
-0.25000 - 0.96825 i
```

2. **Initial and Final Value:** These values can be determined by the initial (IVT) and final (FVT) values theorems.

Taking into account the following remark:

**R** **Remarca 3.1** The initial value ( $t = 0^+$ ) of the impulse response of the self-regulating linear system itself with  $n > m$  is a finite magnitude.

**Demostración** See Adam's textbook [1], among others.  $\square$

Where it is denoted as  $n$  the denominator polynomial order and  $m$  that corresponding to the numerator.

According to IVT,

$$\begin{aligned} y(0^+) &= \lim_{s \rightarrow \infty} sG(s)U(s) \\ &= \lim_{s \rightarrow \infty} s \frac{(s+1)}{(s^2 + 0.5s + 1)} 1 \\ &= 0 . \end{aligned}$$

**R** **Remarca 3.2** The final value ( $t \rightarrow \infty$ ) of the impulse response of a self-regulating linear system itself with  $n \geq m$  is zero.

**Demostración** See Adam's textbook [1], among others.  $\square$

According to FVT,

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} sG(s)U(s) = \lim_{s \rightarrow 0} s \frac{(s+1)}{s^2 + 0.5s + 1} 1 = 0 .$$

3. **Impulse Response:** With the previously information it is possible to predict the system dynamic behavior. For this particular case, we can predict that the system leaves zero (through the IVT) and returns to zero (through the FVT). In addition, it will oscillate around the final value since such system has two complex conjugate roots, according to previous calculations.

Also, using the `impulse` Octave command you can determine the dynamic open-loop response of the system for a Dirac delta input, as indicated below.

```
% System transfer function
s=tf('s');
Gs=(s+1)/(s^2+0.5 s+1);

% Impulse response
impulse(Gs)
```

obtaining the following graph:

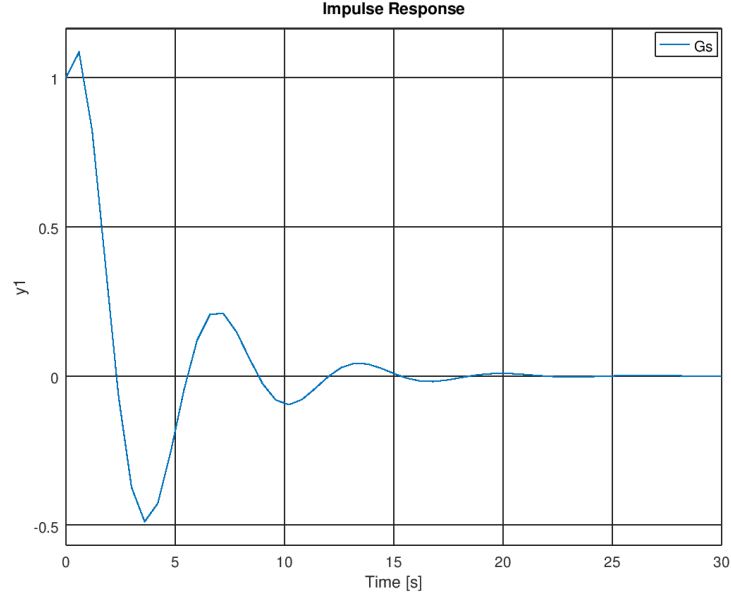


Figure 3.2: System impulse response (3.4).

## 3.2 Step response

Now, consider the system dynamic response subjected to a step input signal with amplitude  $k$ .

The step temporal expression is,  $u(t) = kH(t)$  where,  $H(t)$  is the Heavyside function and therefore, the Laplace transform of  $u(t)$  is  $U(s) = k/s$ . Consequently, the temporal response of the linear system results,

$$y(t) = \mathcal{L}^{-1} \left[ G(s) \frac{k}{s} \right] . \quad (3.5)$$

That is to say: the step response with amplitude  $k$  of a linear system is equal to the inverse Laplace transform of the transfer function multiplied by  $k/s$ .

### Example 2

Consider a linear system of the Exam. 3.1 whose transfer function is given by (3.4). Obtain the dynamic response of such system for a step input.

Firstly, it is studied

1. **Stability:** This depends on the roots of the characteristic equation and is not affected by the type of input. Therefore, the system will have a stable open loop dynamic response for the step input.

2. **Initial and Final Values:** They can be determined by IVT and FVT.

For IVT, the following remark is established below.

**R** **Remarca 3.3** The initial value of the step response of the self-regulating linear system itself is zero if  $n > m$  and nonzero if  $n = m$ .

**Demostración** See Adam's textbook [1], among others.  $\square$

According to Eq. (3.4), it is easy to proof that,

$$y(0^+) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s G(s)U(s) = \lim_{s \rightarrow \infty} \cancel{s} \frac{(s+1)}{s^2 + 0.5s + 1} \cancel{\frac{k}{s}} = 0 .$$

For FVT, the following remark is established below.

**R** **Remarca 3.4** The final value of the step response of the self-regulating linear system itself is equal to  $kK$ .

**Demostración** See Adam's textbook [1], among others.  $\square$

Where it is denoted as  $k$  the step amplitude and  $K$  to the static gain of the open loop plant.

According to the FVT and taking into account Eq. (3.4) it is possible to write,

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s G(s)U(s) = \lim_{s \rightarrow 0} \cancel{s} \frac{(s+1)}{s^2 + 0.5s + 1} \cancel{\frac{k}{s}} = k$$

3. **Step response:** With the previously determined information, the system dynamic behavior can be predicted for the amplitude input step  $k$ . For this example particular, it can be predicted that the system leaves zero (IVT) and reaches a final value equal to  $kK$  (FVT). In addition, the dynamic response will oscillate around the final value since such system has two complex conjugate roots, according to previously calculus.

In addition, using the Octave command `step`, the dynamic open loop response of the system can be determined, as indicated below.

```
% System transfer function
s=tf('s');
Gs=(s+1)/(s^2+0.5*s+1);

% Step response
step(Gs)
```

Thus the Fig. 3.3 is obtained.

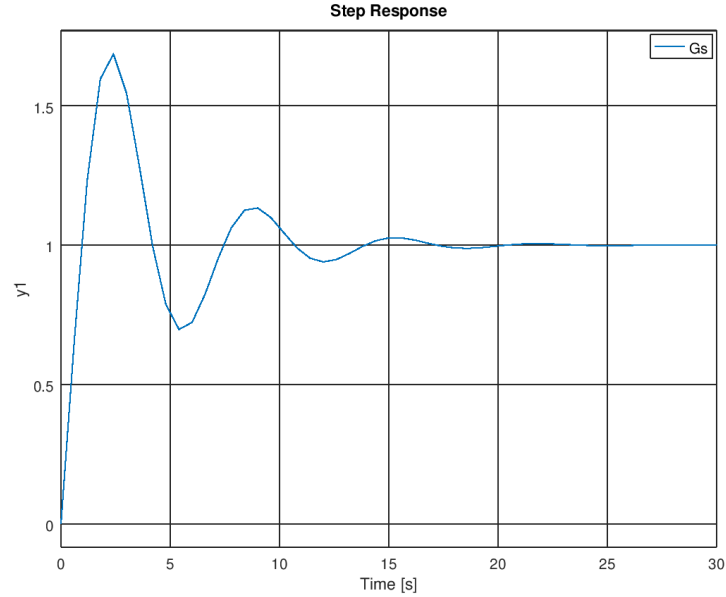


Figure 3.3: System step response (3.4).

### 3.3 Tool Developed for Open Loop Systems

This tool allows:

1. To determine poles and zeros maps of a transfer function.
2. Similarly, impulse and step dynamic responses of  $G(s)$  are presented in the main application window. Also, it is necessary to specify,
  - (a) step amplitude, in case of choosing for this option and
  - (b) simulation time.

Also in this developed tool additional information is presented, as

1. Stability or instability of the LTI system.
2. Final value reached by the open loop system once the transient has finished.
3. Approximate settling time, which is calculated based on the dominant poles of the system. Such calculus can be improved if the simulation time is extended.

#### Example 3

It is pretend to obtain the open-loop response to the step and delta for the transfer function,

$$G(s) = \frac{s + 1}{s^2 + 0.5s + 1} \quad (3.6)$$

the application can be invoked in command mode as indicated below.

```
% packages are loaded.
```

```

pkg load control signal ltitool

% Clean memory and command window.
clear all, clc

% Complex variable "s"
s = tf("s");
% plant
Gol=(s+1)/(s^2+0.5*s+1);
% simulation parameters
k=1;          % step amplitude
tFinal=20;    % simulation time

% -----
% Poles-Zeros Map
% -----
title_string='Poles and Zeros Map of G(s)';
legend_zeros_string='Zeros of G(s)';
legend_poles_string='Poles of G(s)';

figure(1),
xpzmap(title_string,legend_zeros_string,legend_poles_string, ...
Gol.num{1,1},Gol.den{1,1});

% -----
% Simulation
% -----
% Step response
figure(2), OLstepresponse(Gol.num{1,1},Gol.den{1,1},k,0.0,tFinal,1000);

% Impulse response
figure(3), OLimpulseresponse(Gol.num{1,1},Gol.den{1,1},0.0,tFinal,1000);

% -----
% Additional information
% -----
[stability,yinf,Test]=additionalinfo(Gol.num{1,1},Gol.den{1,1}, ...
tFinal,1000)

```

#### The command window reports

```

>> stability = Asympt. Stable
>> yinf = 1
>> Test = 16.060

```

Notice that in the case of the step response, the command window reports that

$y_{inf}=1$  while for the impulse response  $y_{inf}=0$ .

#### Example 4

The same example can be solved directly by invoking the analysis and simulation window of open-loop linear systems, simply by typing in the command window,

```
>> pkg load control signal ltitool,  
>> oLWnd;
```

and then Octave will present the window that it shows in Fig. 3.4.

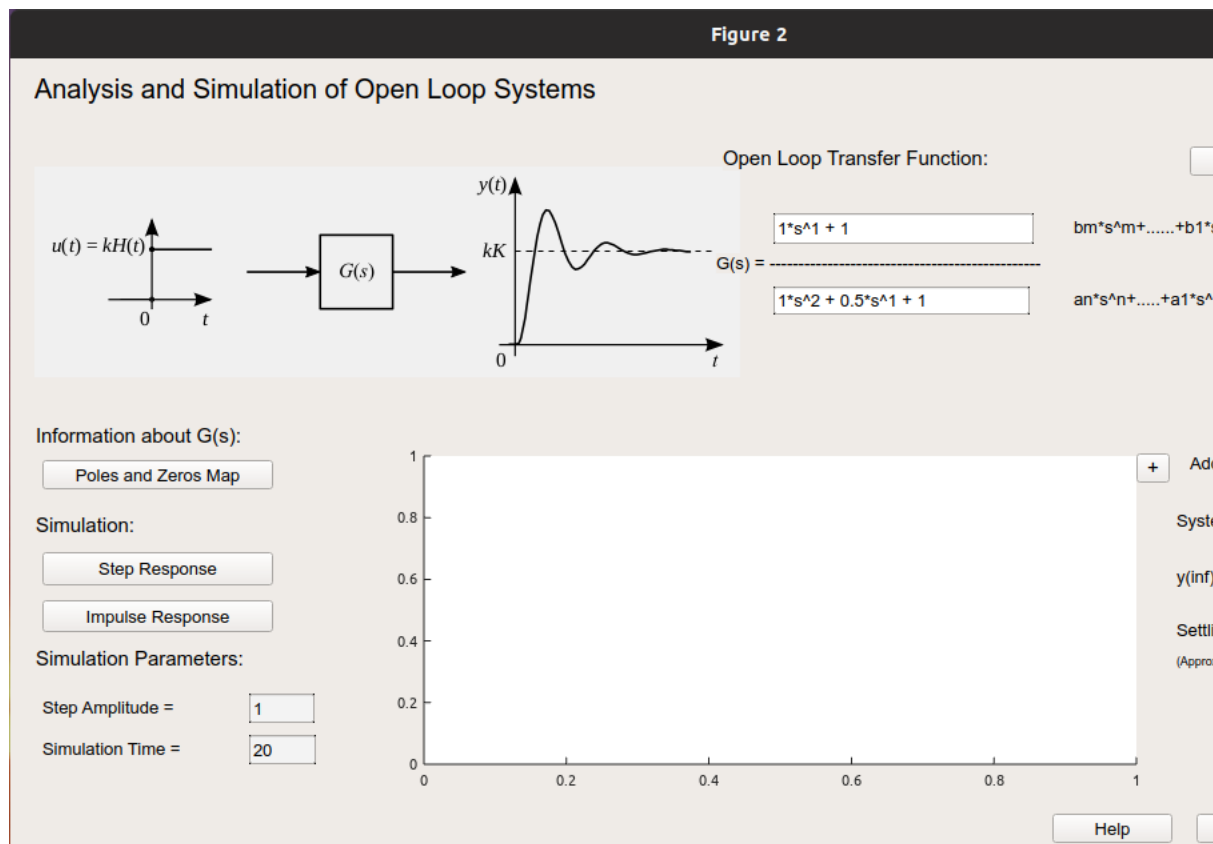


Figure 3.4: Analysis and simulation window for LTI systems in open loop.

Notice that,

- The window has preloaded data that can be changed.
- It is possible to specify, step amplitude and simulation time.
- It can be obtained easily (i) poles-zeros map, (ii) step response or, (iii) impulse response.
- The + button throws a graph out of the application window.



By way of example, the Fig. 3.5 shows the step response along with the additional information about stability, reached steady state value and settling time.

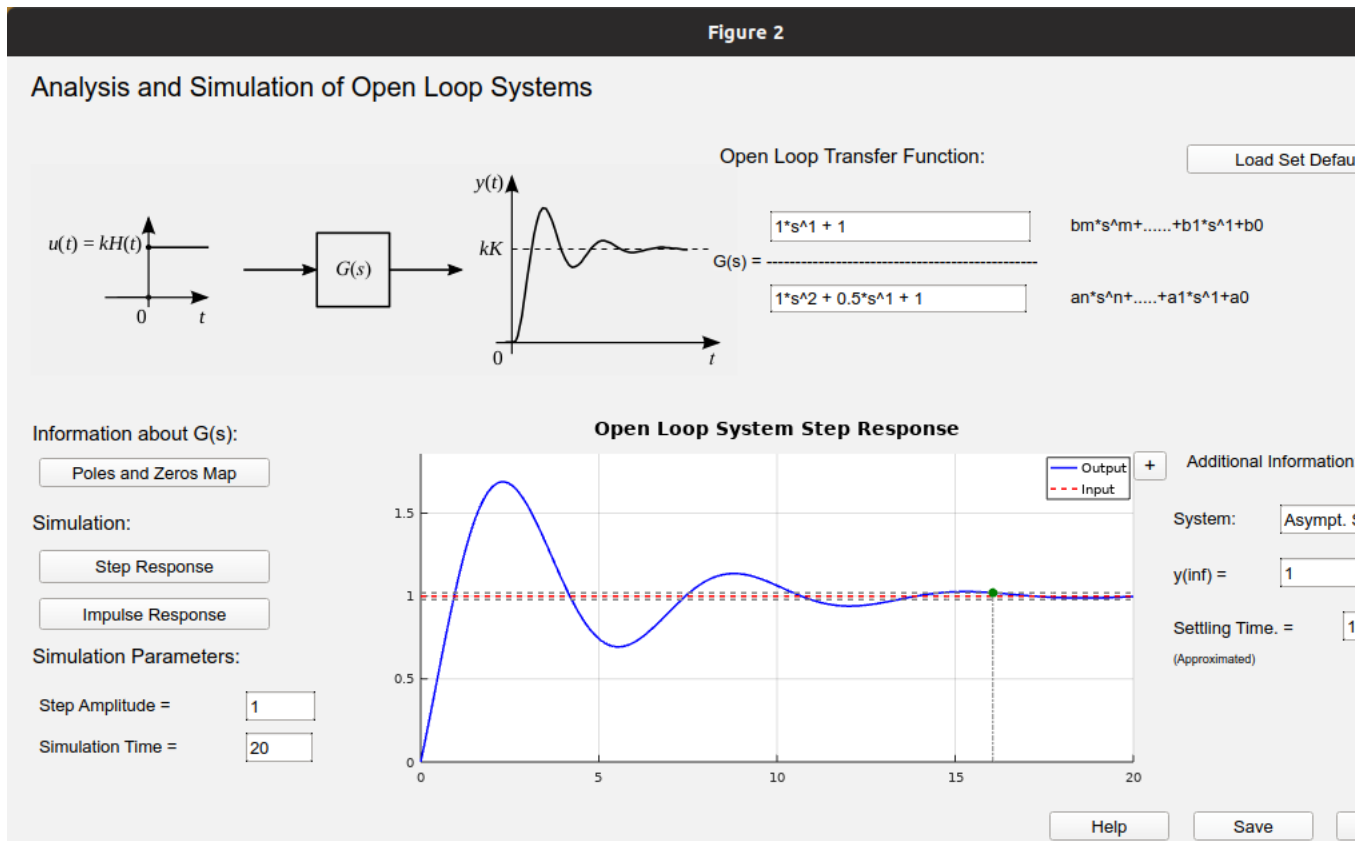


Figure 3.5: Open loop LTI system analysis and simulation window after pressing the step response button.

## 3.4 Conclusions



## Chapter 4

# Closed Loop LTI Systems

Consider the closed loop system given by Fig. 4.1.

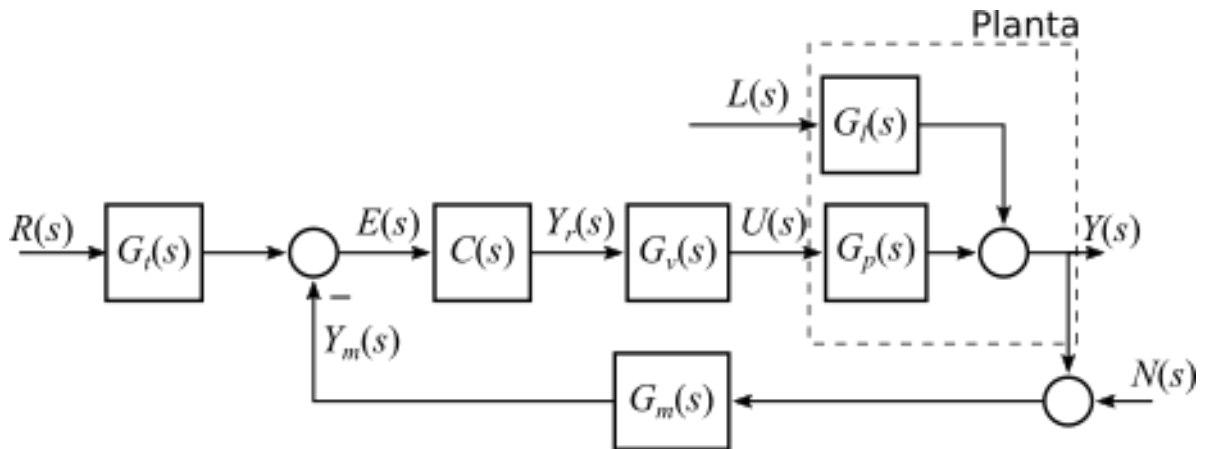


Figure 4.1: Closed loop LTI system.

The closed loop system given by Fig. 4.1 can be redrawn as presented in Fig. 4.2 for a setpoint change where, according to Adam [1, see Chap. 9], if  $G_t(s) = G_m(s)$  the feedback system has a similar behavior that a unity feedback system and consequently it is possible to write,  $G(s) = G_p(s)G_v(s)G_m(s)$  y  $H(s)=C(s)$ .

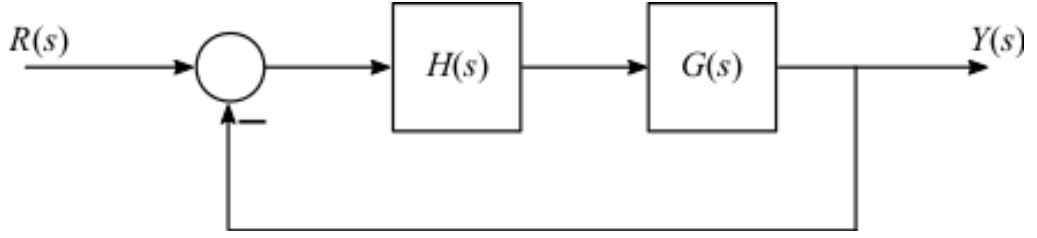


Figure 4.2: Closed loop LTI system with unity feedback.

For the stability study, remember that it must take into account (Adam, [1]):

**Teorema 4.0.1 — Routh-Hurwitz stability Criterion.** The necessary and sufficient condition of stability is that all the terms in the first column of the Routh array have the same sign.

In this toolbox version, two tools have been developed to allow studying of stability by mean of the root locus and Bode and Nyquist stability criteria.

**Example 3.1.** Firstly, consider a unity feedback system whose transfer function is presented in Fig. 4.3, with a PI controller where  $K_r$  is a variable gain and  $T_I = 5$ .

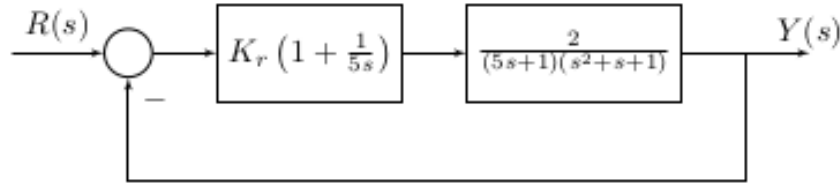


Figure 4.3: Unity feedback system given by Exam. 1.

The open loop transfer function,  $G(s)H(s)$  results,

$$G(s)H(s) = \frac{K_r \cancel{(5s+1)}}{5s} \frac{2}{\cancel{(5s+1)}(s^2+s+1)} = \frac{K^*}{s(s^2+s+1)} \quad (4.1)$$

where for this particular case  $K^* = 2K_r/5$ .

For this example, the closed loop transfer function results,

$$\frac{Y(s)}{R(s)} = G_{lc}(s) = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{K^*}{s^3 + s^2 + s + K^*}. \quad (4.2)$$

To study the stability of the feedback system, the Routh stability criterion will be apply based on the characteristic equation of feedback loop system.

Being the characteristic equation of this example,  $D(s) = s^3 + s^2 + s + K^*$  and therefore the Routh array results

$$\begin{array}{c|cc} s^3 & 1 & 1 \\ s^2 & 1 & K^* \\ s^1 & 1 - K^* & 0 \\ s^0 & K^* & 0 \end{array}$$

It follows that the system will be *asymptotically stable* if and only if,  $0 < K^* < 1$ .

Now it is pretended to study the initial and final values reached for a step change in the setpoint  $R(s) = k/s$ . To do this, applying the initial value theorem (IVT),

$$y(0^+) = \lim_{s \rightarrow \infty} sY(s) = \lim_{s \rightarrow \infty} s G_{lc}(s)R(s) = \lim_{s \rightarrow \infty} \cancel{s} \frac{K^*}{s^3 + s^2 + s + K^*} \frac{k}{\cancel{s}} = 0 \quad .$$

Then, applying the final value theorem (FVT),

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s G_{lc}(s)R(s) = \lim_{s \rightarrow 0} \cancel{s} \frac{K^*}{s^3 + s^2 + s + K^*} \frac{k}{\cancel{s}} = k$$

From the results of both theorems it is shown that the system starts from zero and reaches the setpoint value. This result is subject to the system being stable in closed loop; this is,  $0 < K^* < 1$ .

Now adopting  $K^* = 0.5$  (o sea  $Kr = 1.25$ ) to reach a  $GM = 2$ , the characteristic closed loop equation results  $D(s) = s^3 + s^2 + s + 0.5$  presenting three poles in  $s_1 = -0.64780$  y  $s_{2-3} = -0.17610 \pm j0.86072$ .

Note that both systems do not have zeros.

These results can be easily obtained with the Octave commands,

```
pkg load control

% G(s)H(s) transfer function
Kast=0.5;
s=tf('s');
GH=Kast/(s*(s^2+s+1));

% G(s)H(s) Poles-Zeros
polesGH=roots(GH.den{1,1})
% Closed loop Poles-Zeros
Gcl=feedback(GH,1)
polesGcl=roots(Glc.den{1,1})
```

where the answer in the Octave command window is,

```
polesGH =

-0.50000 + 0.86603i
-0.50000 - 0.86603i
```

```
0.00000 + 0.00000i
```

```
Transfer function 'Glc' from input 'u1' to output ...
```

```

          0.5
y1:  -----
      s^3 + s^2 + s + 0.5
```

```
Continuous-time model.
```

```
polesGcl =
```

```
-0.17610 + 0.86072i
```

```
-0.17610 - 0.86072i
```

```
-0.64780 + 0.00000i
```

The feedback system step response can be obtained by means of the following Octave command:

```
% Step response
step(Glc)
```

that it shown in Fig. 4.4.

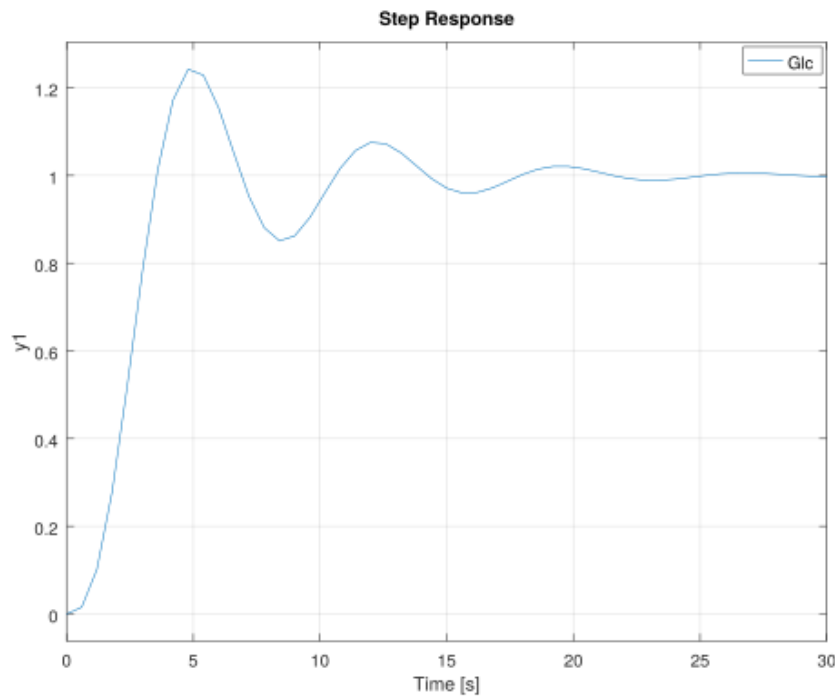


Figure 4.4: Unit step response of the feedback system of the Fig. 4.3.

## 4.1 Tools for Feedback LTI Systems

As mentioned before, two tools for the analysis and simulation of closed loop LTI systems were developed. These ones apply to

1. Feedback system based on  $G(s)H(s)$  according to Fig. 4.2 and,
2. Closed loop system under the representation of ??.

### 4.1.1 Analysis and Simulation Based on $G(s)H(s)$

The tool here developed consists of a function set that allows for a given variable gain value  $K$ :

1. Determine the map of poles and zeros of the transfer function  $G(s)H(s)$  and of the closed loop transfer function with unit feedback for setpoint change.
2. Present the root locus of a feedback system, Bode and Nyquist diagrams.
3. Furthermore, it can also be determined, the dynamic responses of the feedback system given by Fig. 4.2. For this you must specify,
  - (a) step amplitude and
  - (b) simulation time

Besides in this tool, additional information about,

1. stability or instability of the feedback linear system;
2. final value reached by the open loop system once the transient has passed;
3. Approximated settling time, which is calculated based on the dominant pole(s) of the system. Such calculation can be improved if the simulation time is extended.
4. The gain value  $K^*$  and the ultimate gain value  $K_u^*$  of  $G(s)H(s)$ , phase and gain cross frequencies  $\omega_{-180}$  and  $\omega_{cg}$  respectively; together with the gain and phase margins.

**Ejemplo 3.2** As an example, the resolution of the Exam. 3.1 is shown using the developed functions, which can be invoked directly from the Octave command window or, through an .m extension file, as is usually done.

```
% packages are loaded.
pkg load control signal ltitool

% Clean memory and command window.
clear all, clc

% -----
% Default values for the variables are defined
% -----
% G(s)H(s) :
s=tf('s');
Gs=2.0/((5*s+1)*(1*s^2+1*s+1)); delay=0;
Hs=(5*s+1)/(5*s);
Kr=1.25;
GH=minreal(Kr*Gs*Hs);

% Simulation parameters
AmplitudSP=1; % step amplitude
tFinal=30;    % simulation time

% -----
% Analisis Tools
% -----
% Root locus
figure(1), clf(figure(1),"reset");
[Kast,Kast_ult,wultimo,GM,wcg,PM]=Ghrlocusplot(GH.num{1,1},GH.den{1,1},Kr,10);

% Bode diagrams
figure(2), clf(figure(2),"reset");
[Kast,Kast_ult,wultimo,GM,wcg,PM]=GHbodeplot(GH.num{1,1},GH.den{1,1},delay,Kr);

% Nyquist diagram
figure(3), clf(figure(3),"reset");
```



```
[Kast,Kast_ult,wultimo,GM,wcg,PM]=GHnyquistplot (GH.num{1,1},GH.den{1,1},delay,Kr);

% -----
% Simulation Tools
% -----
% Step response
figure(4), clf (figure(4),"reset");
GHstepresponse (GH.num{1,1},GH.den{1,1},AmplitudSP,tFinal,1000);

% Impulse response
figure(5), clf (figure(5),"reset");
GHimpulseresponse (GH.num{1,1},GH.den{1,1},tFinal,1000);

% Additional information
[estabilidad,yinf,Test]=additionalinfo (GH.num{1,1},GH.den{1,1},AmplitudSP,tFinal,1000);
```

As an example, the root locus diagram of this application is shown.

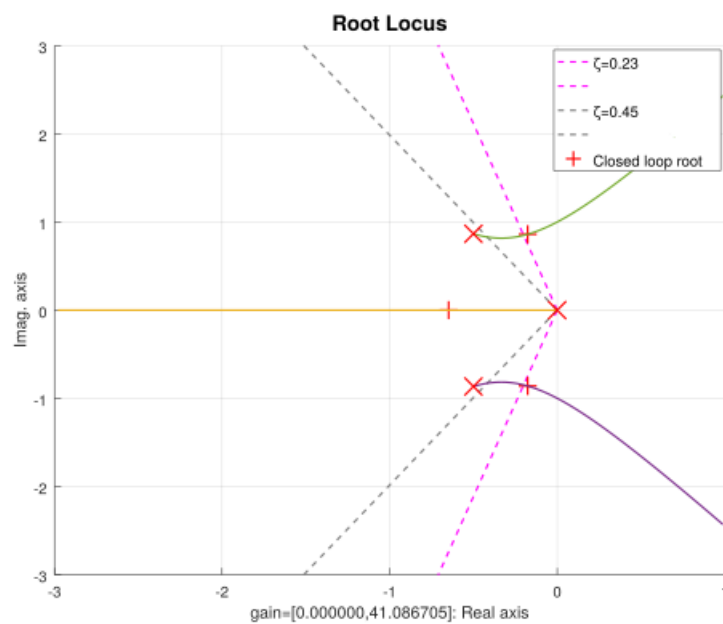


Figure 4.5: Root locus diagram of Exam. 3.2.

### Ejemplo 3.3

Figure 4.6 shows the application window performed for the analysis of closed-loop LTI systems based on the expression of  $G(s)H(s)$ .

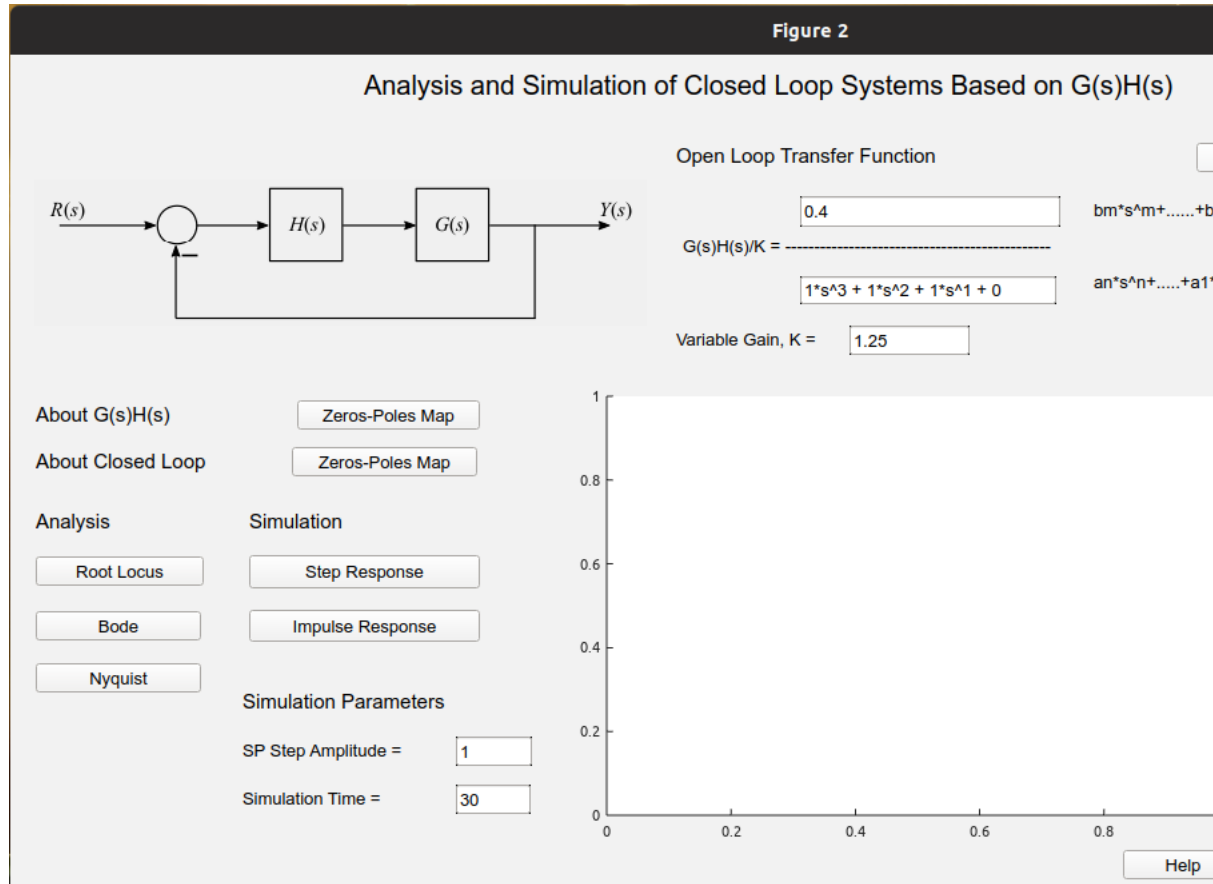


Figure 4.6: Closed loop system analysis and simulation window based on  $G(s)H(s)$  with preloaded data.

Note that the data has been preloaded by a file that saves the data from the last time you worked with this application. There is also a button (Load Default Values Botton) that load the preloaded data for this particular example, which if it pressed, default values are reloaded.

Figure 4.7 shows the numerical simulation of the feedback system when  $K^* = 0.5$ . Similarly, the poles and zeros maps of  $G(s)H(s)$  can be obtained from the feedback system, together the root locus, Bode and Nyquist diagrams.

In addition, the additional information button provides information of the feedback system related to stability, establishment value, and establishment time,  $GM$ ,  $PM$ ,  $\omega_{-180}$  y  $\omega_{cg}$ . Note that the values  $K^*$  and  $GM$  here reported are coincident with that it was presented by the Exam. 3.1.

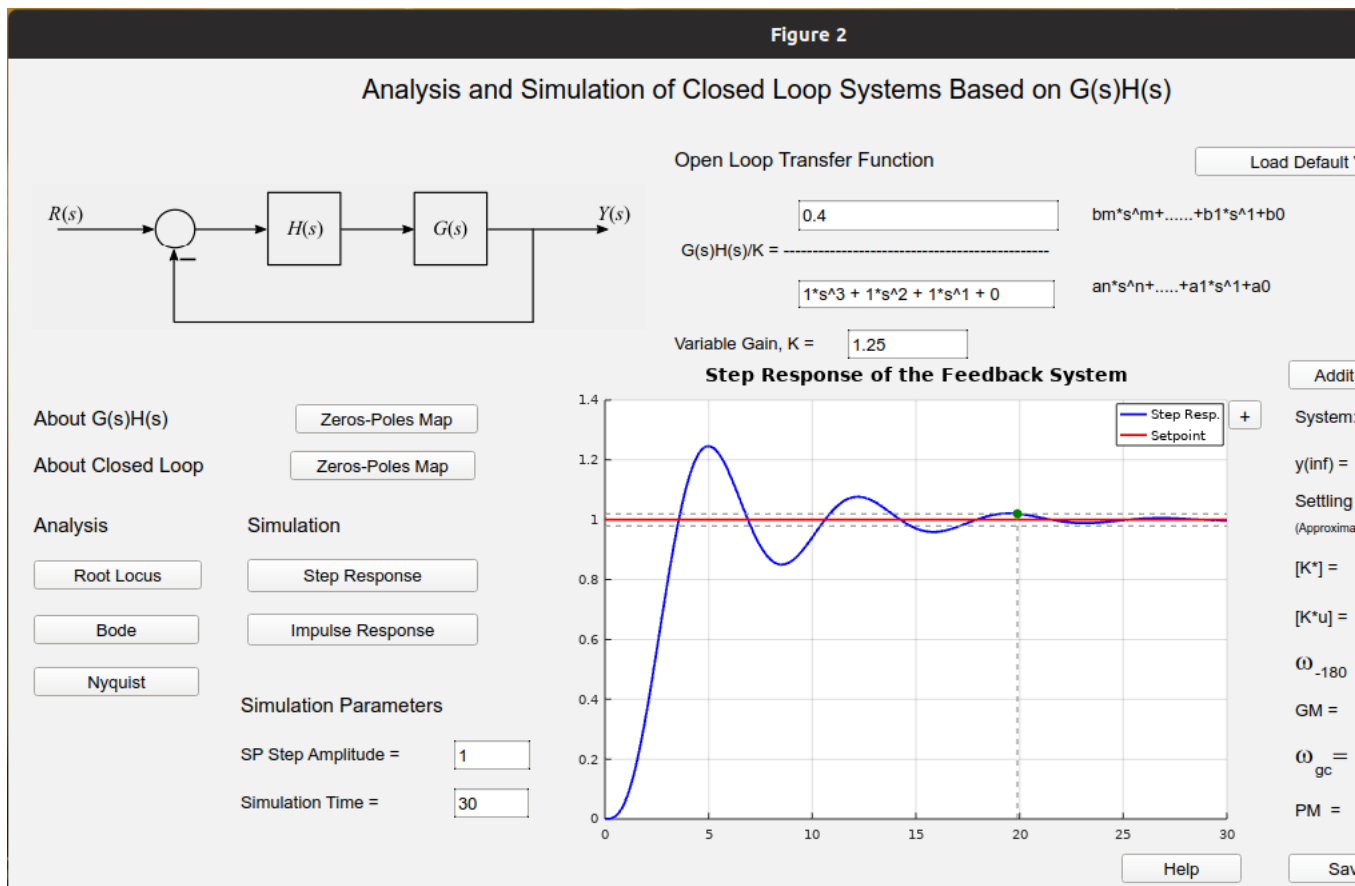


Figure 4.7: Closed loop system analysis and simulation window based on  $G(s)H(s)$  with step response.

#### 4.1.2 Analysis and Simulation of Feedback System

In this subsection, it will be considered the closed loop system of Fig. 4.1.

##### Example 3.4

Figure 4.8 shows the developed window for analysis and simulation of closed loop system.

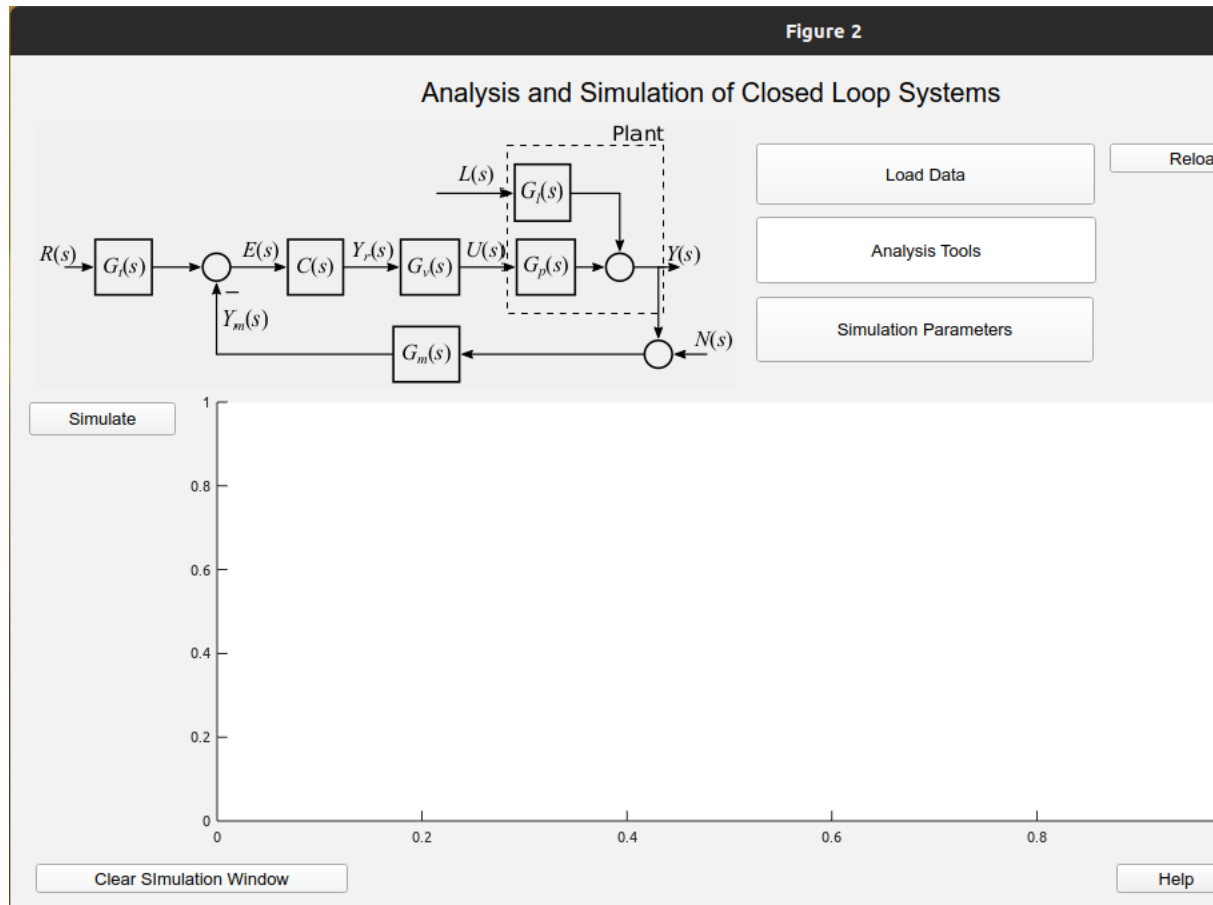


Figure 4.8: Main window for analysis and simulation of closed loop system.

Notice that, if the user press load data button then a window like the one in the Fig. 4.9 will be present. In it, the data corresponding to the Fig. 4.1 must be loaded.

**Figure 3**

**Transfer Functions of the Closed Loop System**

Plant

$1*s^1 + 1$        $bm*s^m + ..... + b1*s^1 + b0$        $1*s^1 + 1$

$G_p(s) = \frac{1*s^2 + 1*s^1 + 1}{1*s^2 + 1*s^1 + 1}$        $G_l(s) = \frac{1*s^2 + 1*s^1 + 1}{1*s^2 + 1*s^1 + 1}$

Measuring Element      Input Converter      Final Control Element

$1$        $1$        $1$

$G_m(s) = \frac{0.1*s^1 + 1}{0.1*s^1 + 1}$        $G_t(s) = \frac{0.1*s^1 + 1}{0.1*s^1 + 1}$        $G_v(s) = \frac{0.2*s^1 + 1}{0.2*s^1 + 1}$

Controller

☒ PID       $C(s) = K_r(1 + 1/TI_s + TD_s)$

☒ P Mode       $1$       ☒ I Mode       $1$       ☐ D Mode       $0$

☐ Compensator

$G_c(s) = \frac{1*s^1 + 1}{0.1*s^1 + 1}$

OK

Figure 4.9: Window for defining of feedback system transfer function.

On the other hand, the user has a tool for analysis of feedback systems, if the analysis tools button is pressed. Based on the preloaded data in the previous window, this tool provides in a single window with

1. Poles-Zeros Map of  $G(s)H(s)$ .
2. Poles-Zeros Map of the feedback system.
3. Root locus plot
4. Bode and Nyquist diagrams

as it is shown in Fig. 4.10.

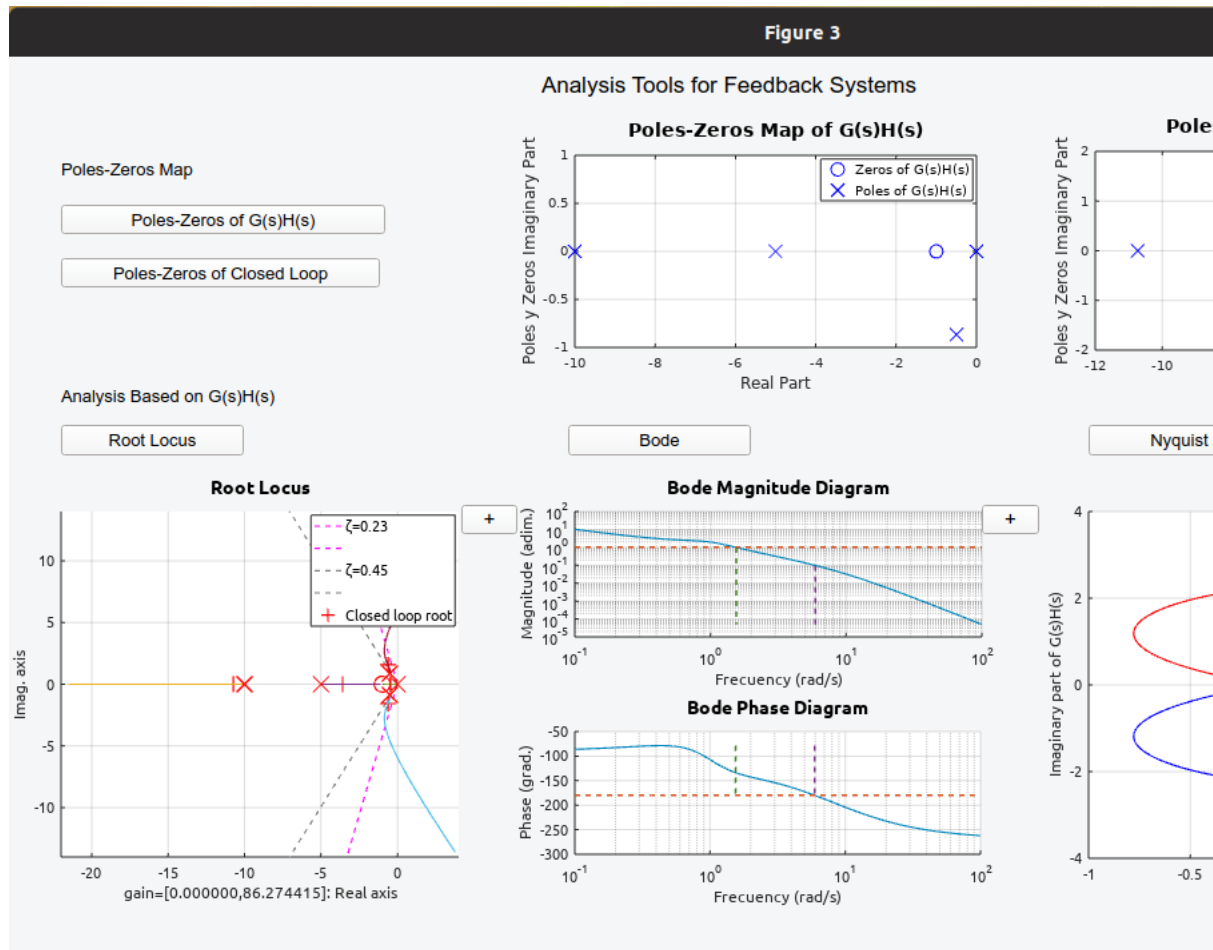
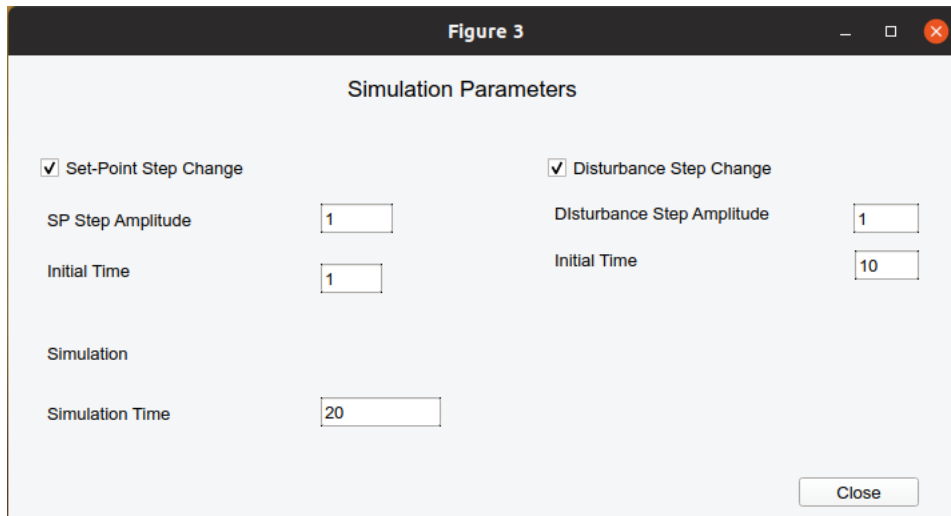


Figure 4.10: Tool Window for analysis of feedback system.

Furthermore, the user has a simulation parameter window where allows setting step amplitude and initial time of set-point and disturbance changes and, the simulation time.



The image shows a software window titled "Figure 3" with a subtitle "Simulation Parameters". It contains several input fields and checkboxes. On the left, there is a checked checkbox for "Set-Point Step Change", followed by "SP Step Amplitude" (value 1), "Initial Time" (value 1), and "Simulation Time" (value 20). On the right, there is a checked checkbox for "Disturbance Step Change", followed by "Disturbance Step Amplitude" (value 1), and "Initial Time" (value 10). A "Close" button is located at the bottom right.

Parameter	Value
Set-Point Step Change	<input checked="" type="checkbox"/>
SP Step Amplitude	1
Initial Time	1
Disturbance Step Change	<input checked="" type="checkbox"/>
Disturbance Step Amplitude	1
Initial Time	10
Simulation Time	20

Figure 4.11: Tool window for defining simulation parameters.

Finally, in Fig. 4.12 the simulation of this example is shown.

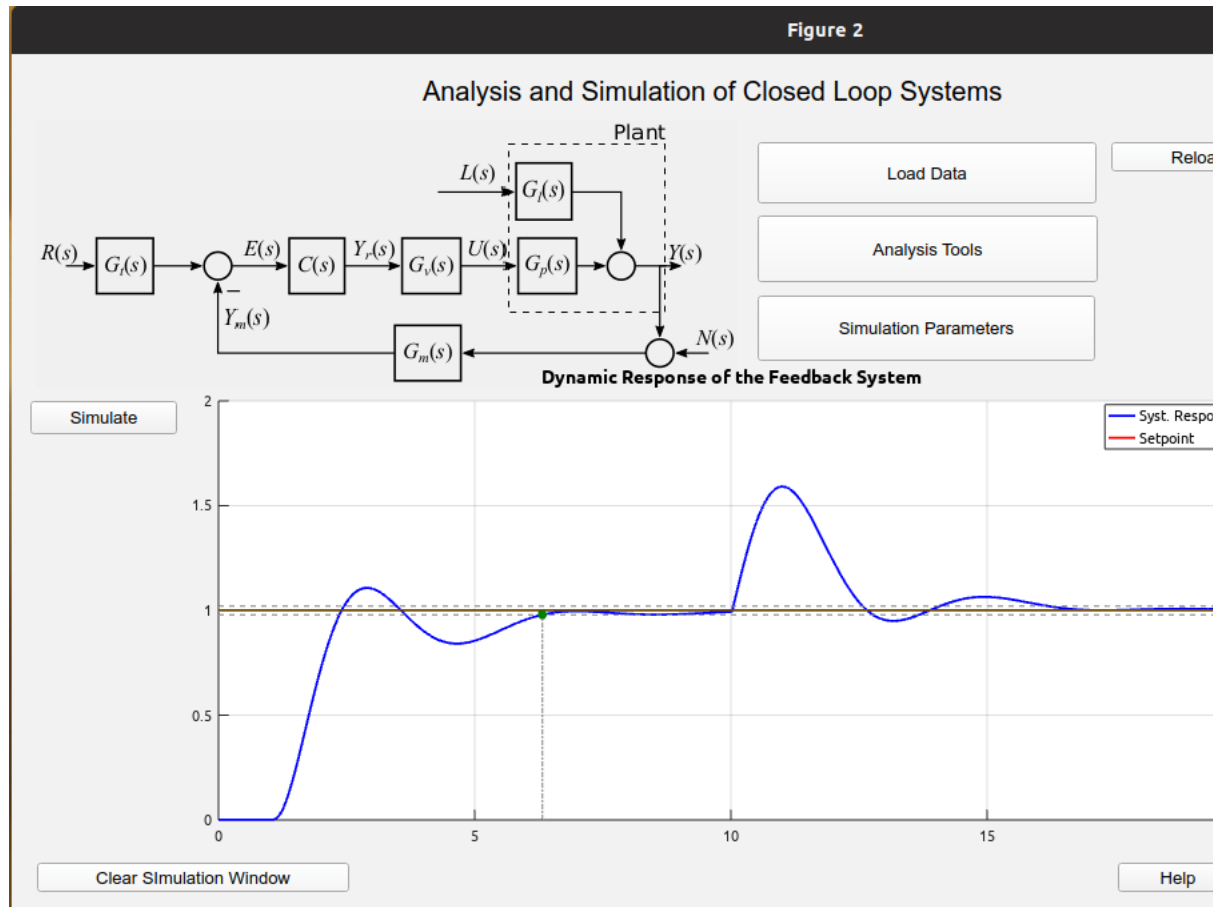


Figure 4.12: Window with a simulation example.

## 4.2 Conclusions



## **Chapter 5**

# **Traditional Advanced Control Strategies**

This chapter presents tools for the analysis and simulation of traditional advanced control strategies such as feedforward-feedback control, cascade control, among others that are currently under development.

### **5.1 Feedforward-Feedback Control**

Consider the feedforward-feedback control system according to Fig. 5.1.

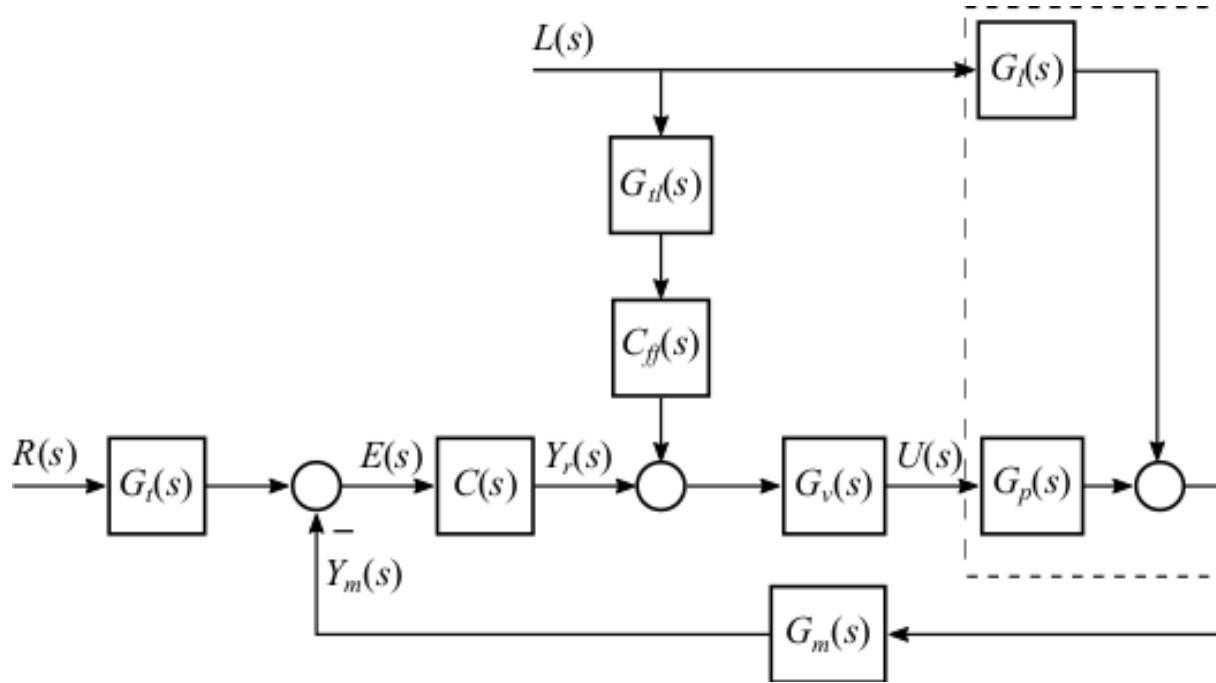


Figure 5.1: Feedforward-feedback control system.

## 5.2 Cascade Control

Consider the cascade control system according to Fig. 5.2.

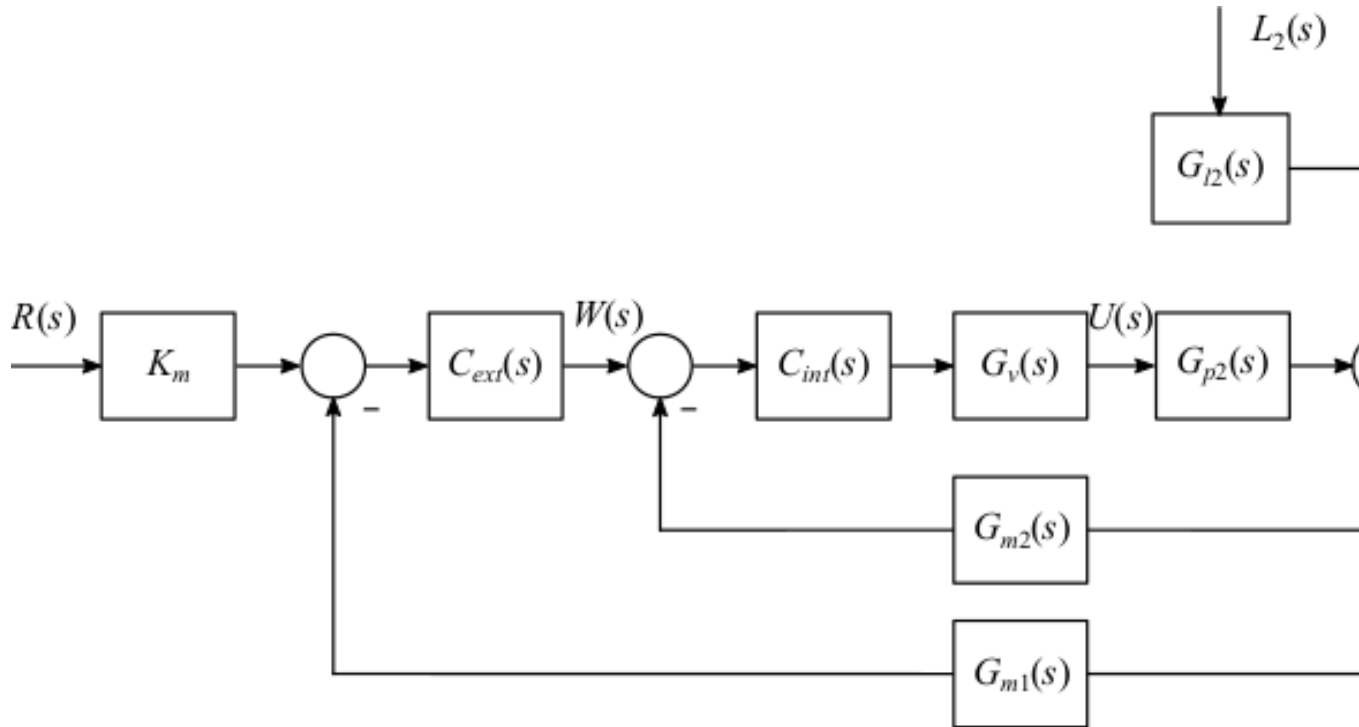


Figure 5.2: Cascade control system.

## 5.3 Tools for Traditional Advanced Control Strategies

### 5.3.1 Analysis and Simulation of a Feedforward-Feedback Control System

#### Example 5.1

Figure 5.3 shows the developed main window for traditional advanced control strategies that includes feedforward-feedback control and cascade control schemes.

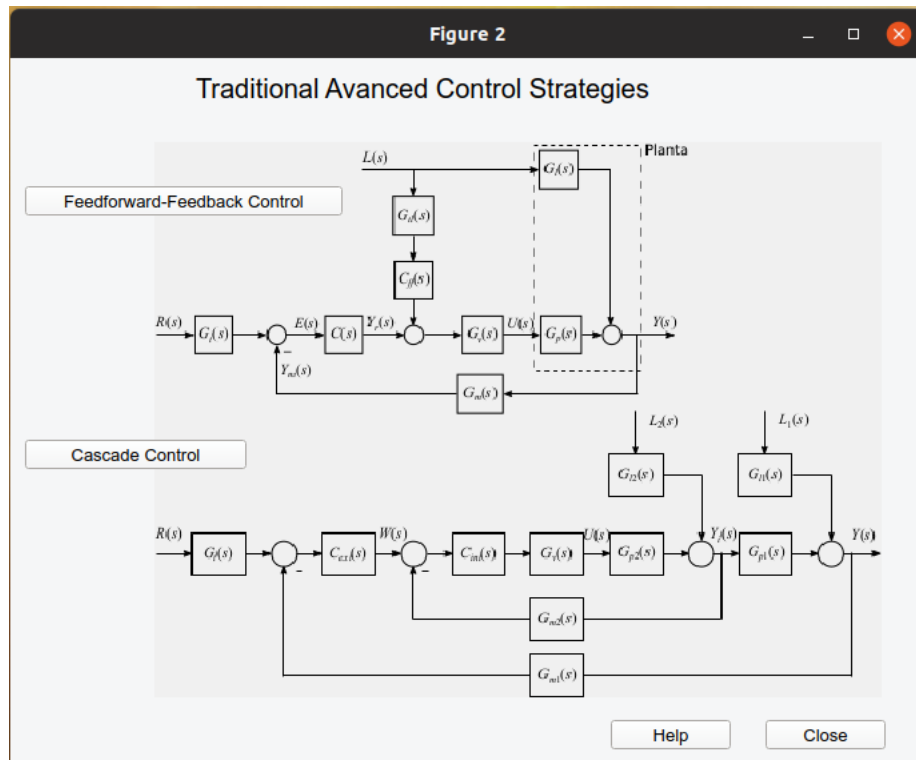


Figure 5.3: Main window for Traditional Advanced Control Strategies.

Figure 5.4 shows the developed main window to study the combined control scheme known as feedforward-feedback control.

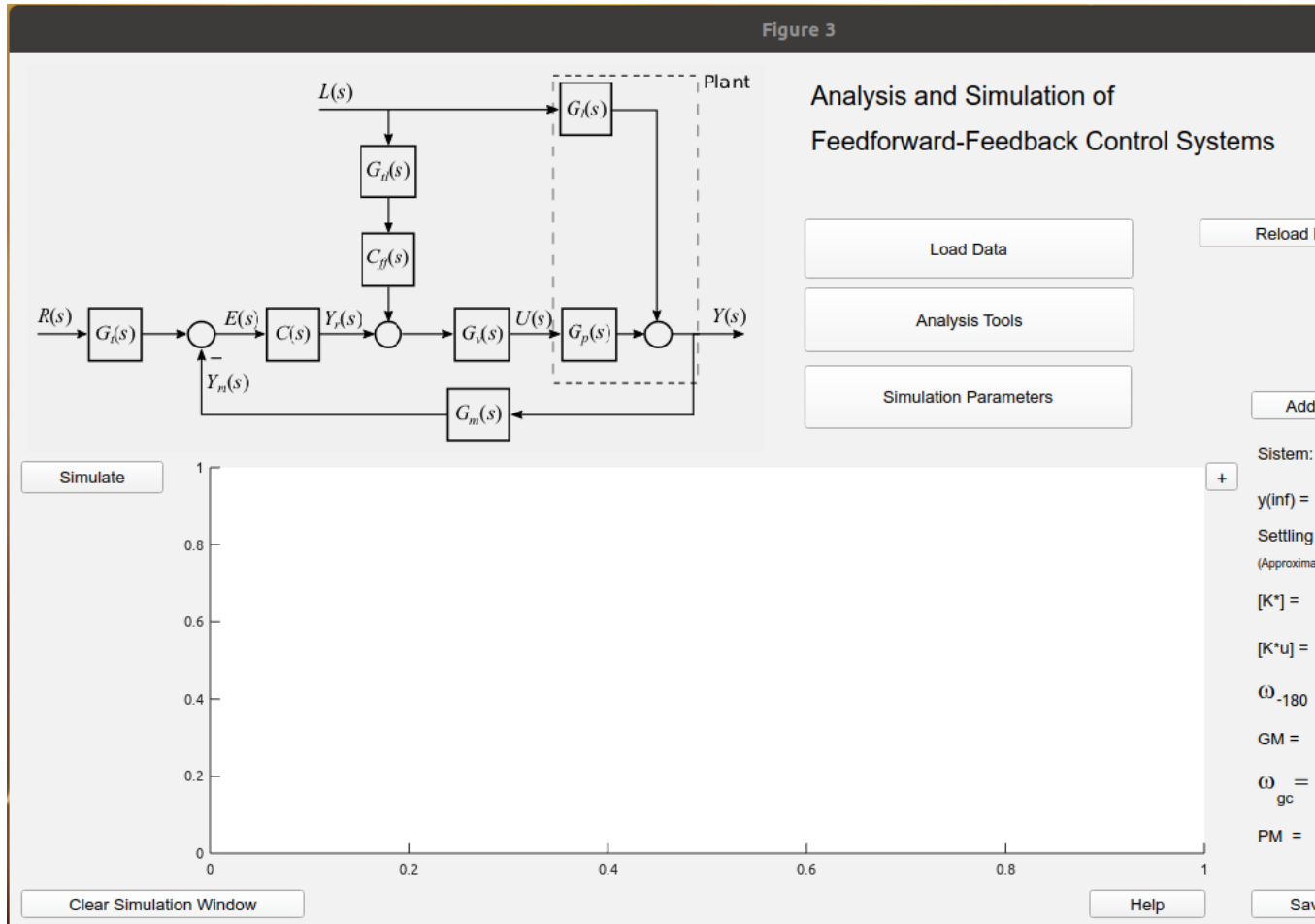


Figure 5.4: Main window for feedforward-feedback control.

Note that the load data bottom opens a window to load data for this example and it is similar to Fig. 4.9.

Figure 5.5 shows the developed analysis tool window which is similar to developed window for closed loop system (Fig. 4.10).

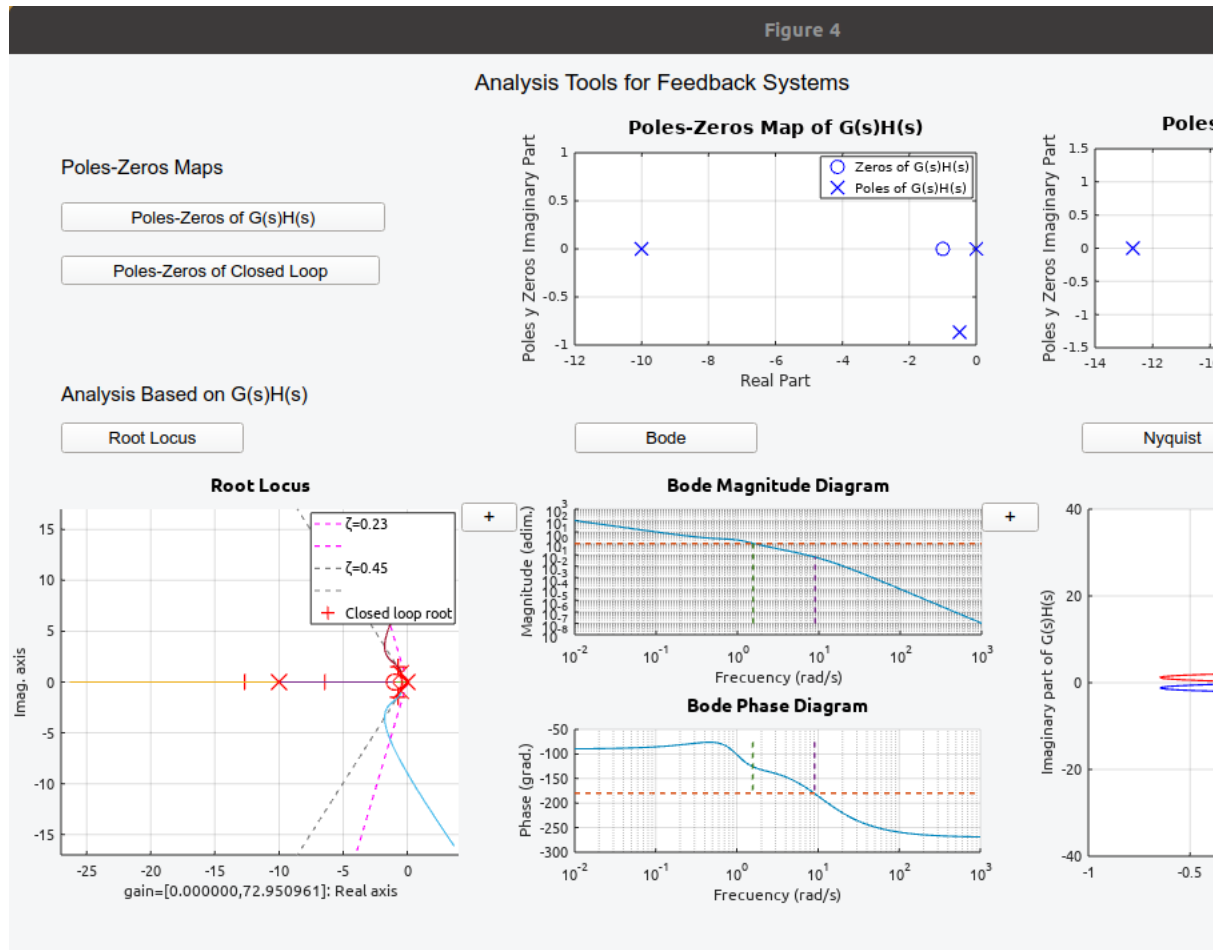


Figure 5.5: Analysis tool window for feedforward-feedback control system.

Finally, Fig. 5.6 shows the simulations obtained with this example.

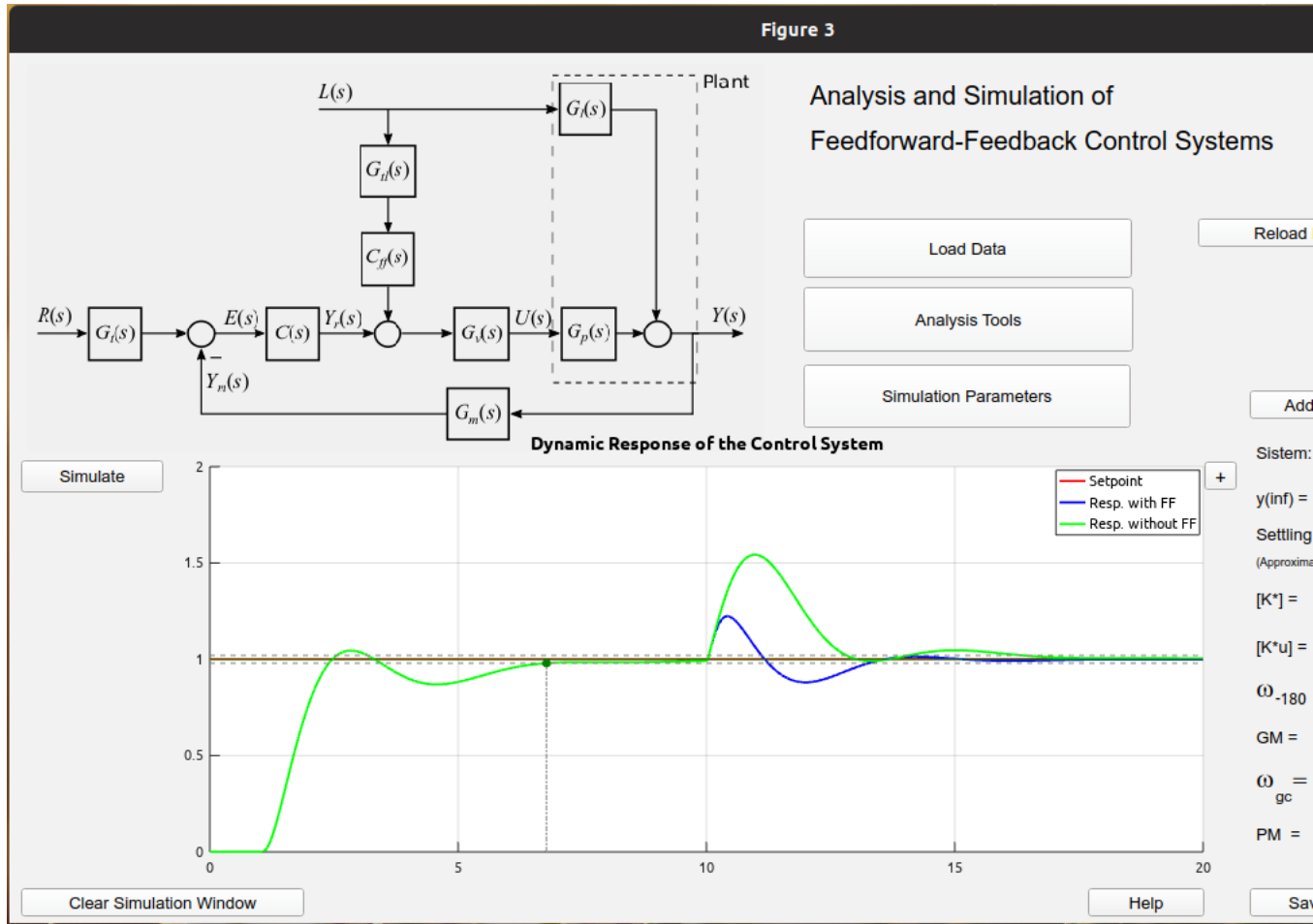


Figure 5.6: Simulation of feedforward-feedback control example.

### 5.3.2 Analysis and Simulation of a Cascade Control System

#### Example 5.2

If the user press the cascade control bottom indicated in Fig. 5.3 then, a window developed for cascade control scheme is opened. Figure 5.7 shows the developed main window to study this combined control scheme.

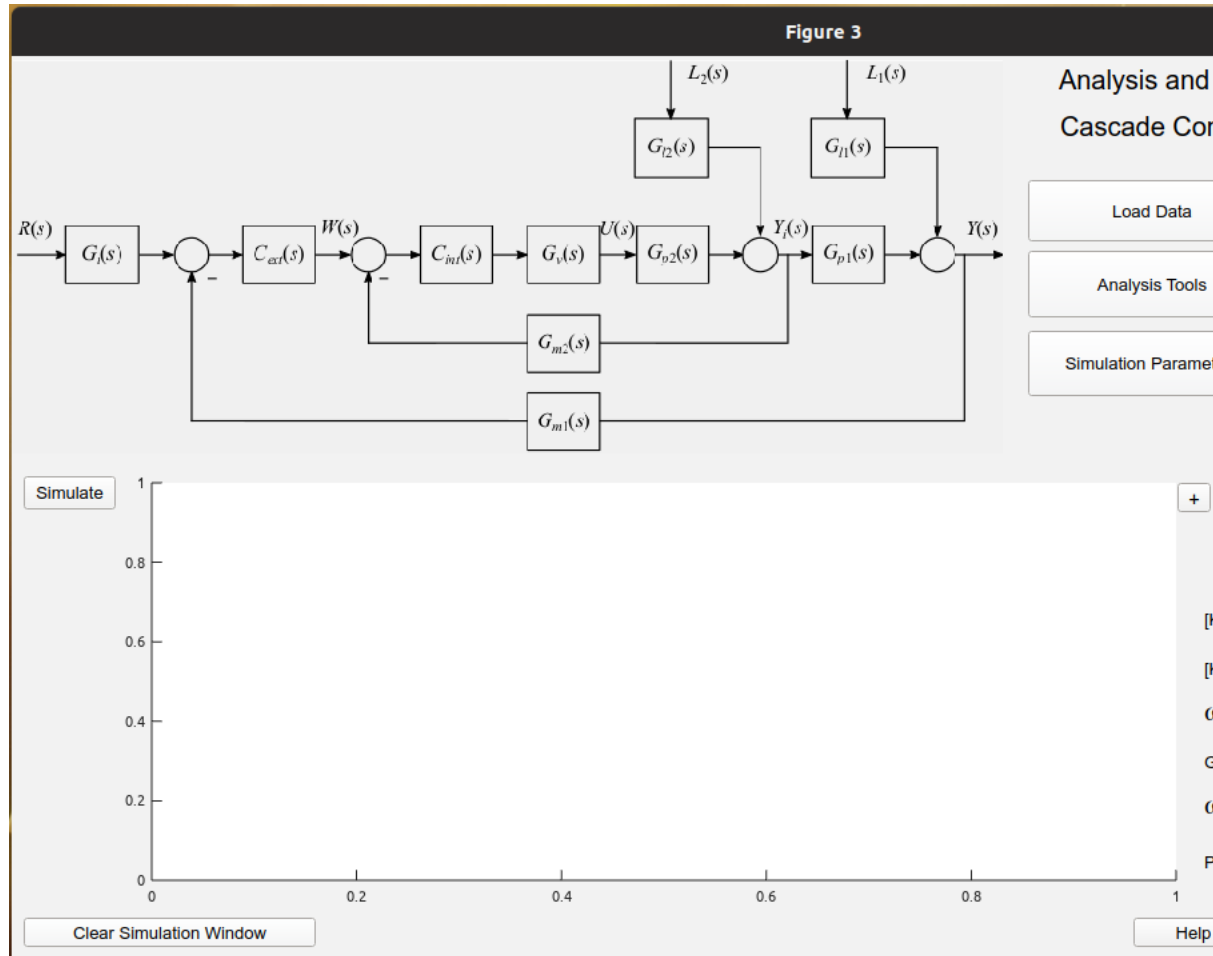


Figure 5.7: Main window for cascade control.

Note that the load data bottom opens a window to load data for this example and it is similar to Fig. 4.9. Also, when the bottom analysis tools is pressed the window indicated in Fig. is opened.



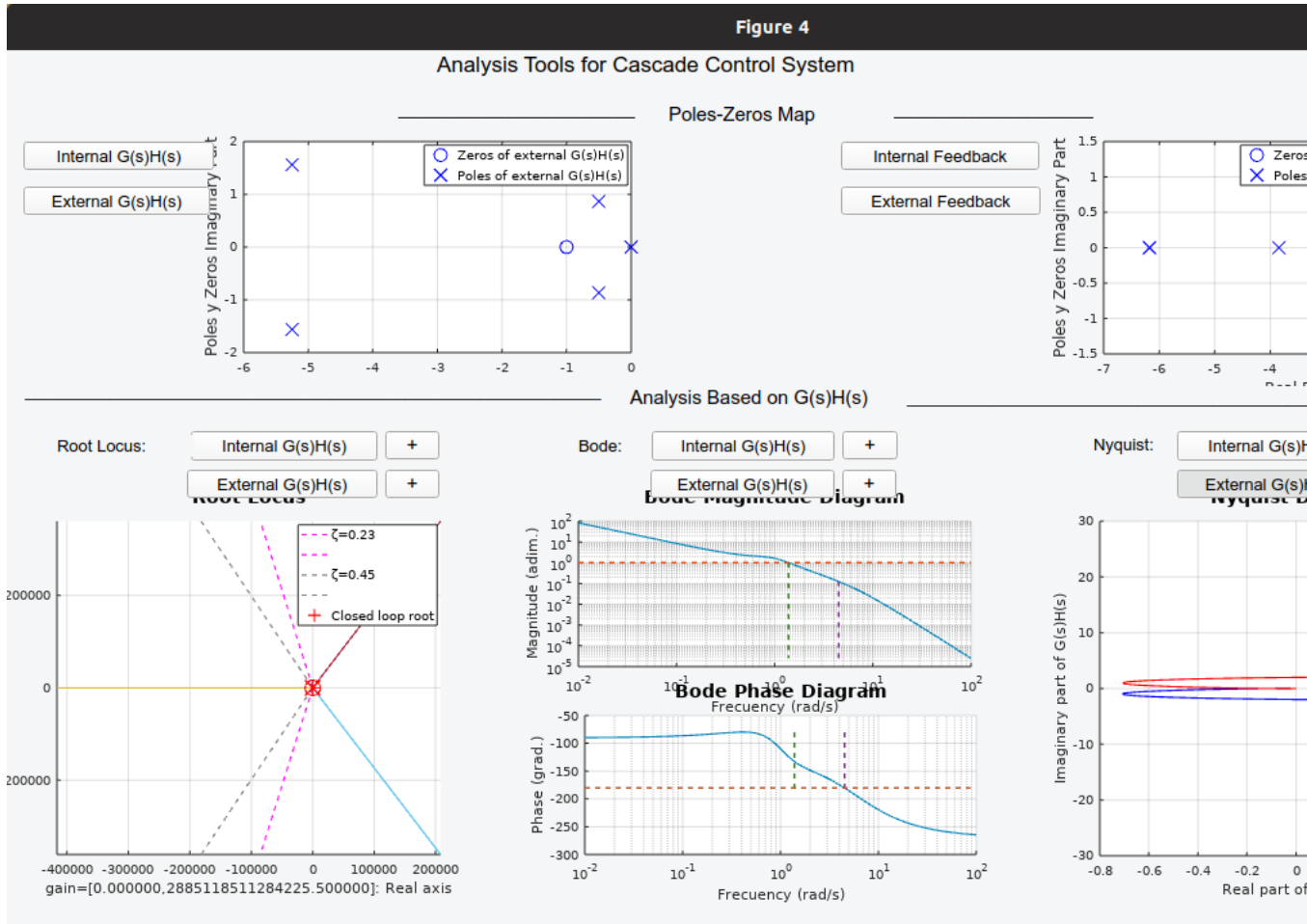


Figure 5.8: Analysis tool window for control control system.

Note that in mentioned window, the user can study internal and external loop independently.

Finally, Fig. 5.9 shows the simulations obtained with this example.

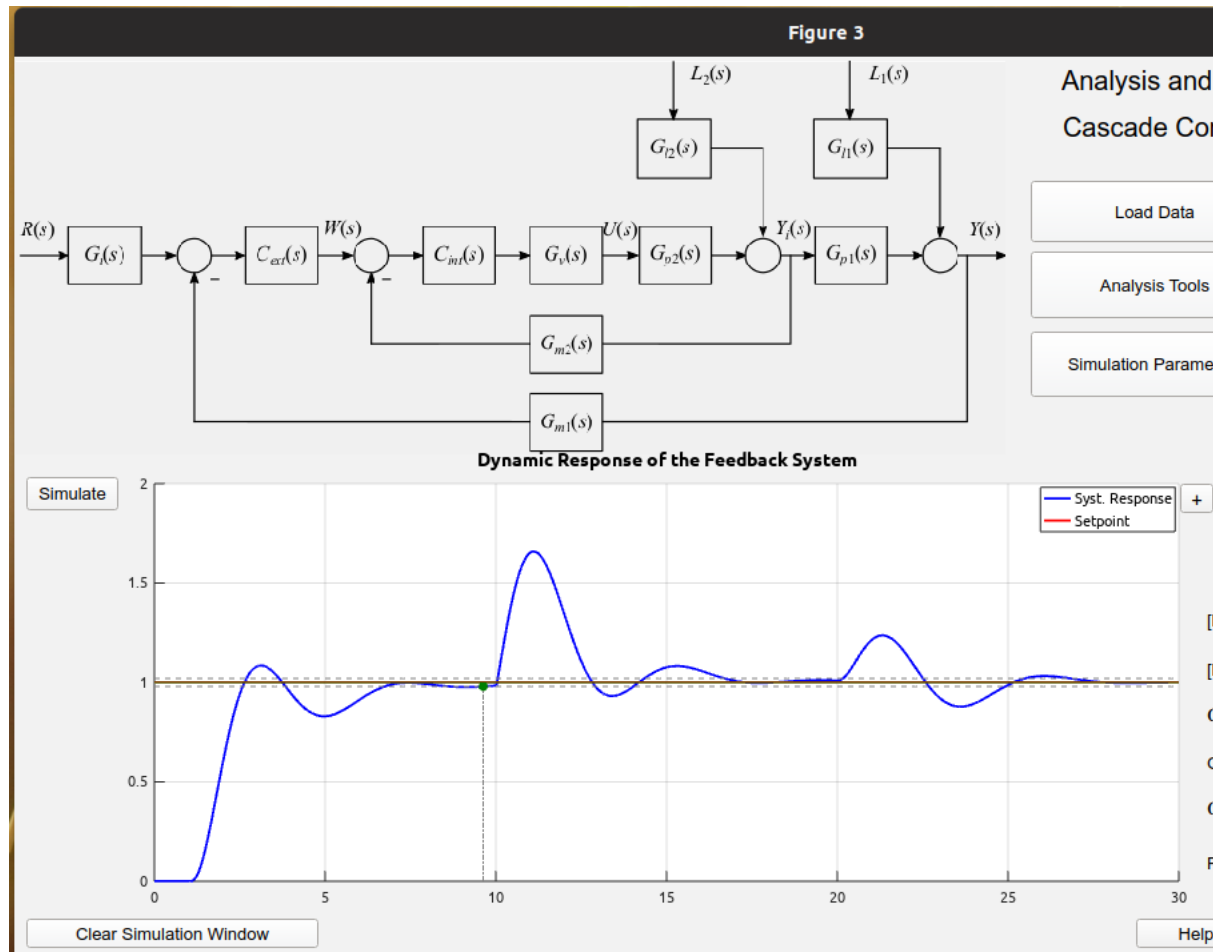


Figure 5.9: Simulation of cascade control example.

## 5.4 Conclusions

## **Chapter 6**

# **Additional Tools for LTI System Control**

This chapter includes additional tools for LTI system control. These are, Optimal Tuning of PID Controller, among others that they are in development.

### **6.1 Optimal Setting of PID Controller**

### **6.2 Developed Additional Tools**

#### **6.2.1 PID Optimal Controller Setting**

Fig. 6.1 show the developed main window for tuning PID controller based on mixed criteria. For that, the minimization of ITAE, maximum overshoot and sensitivity function was chosen.

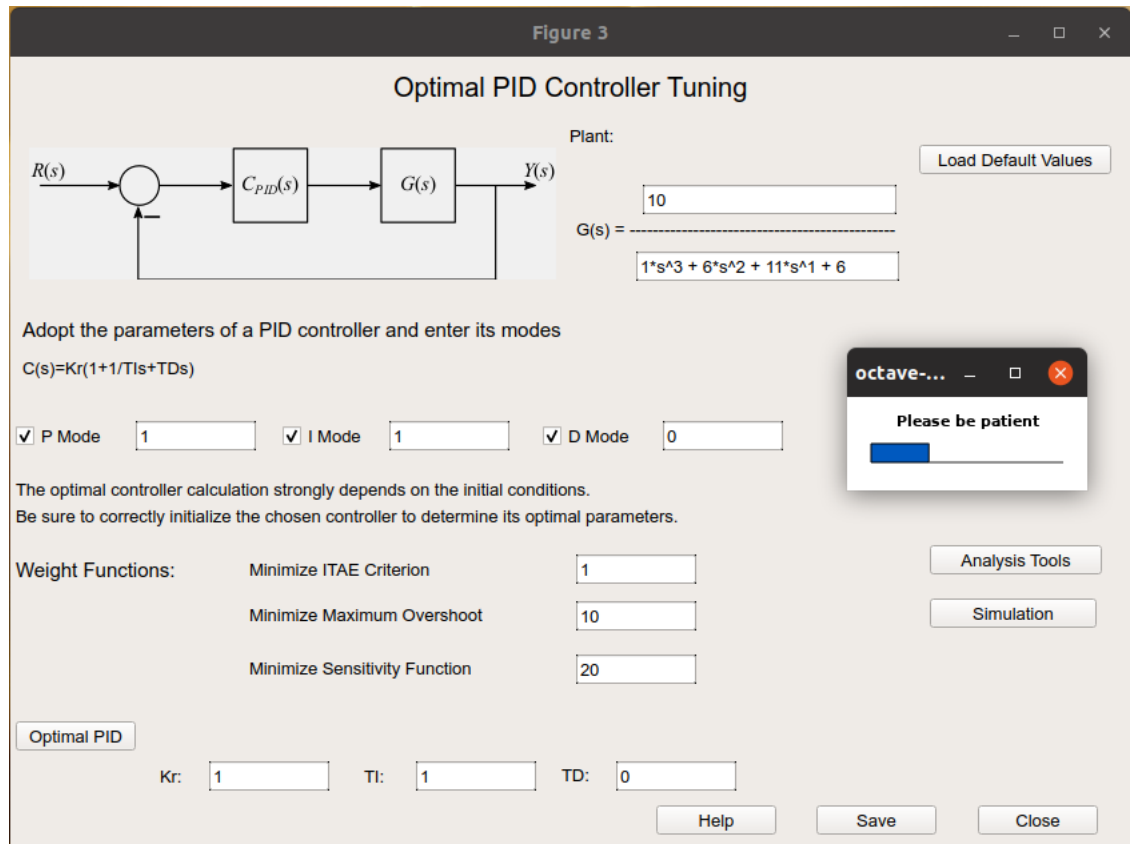


Figure 6.1: Main window for tuning optimal PID controllers.

Figure 6.2 shows simulations and results obtained in this example.

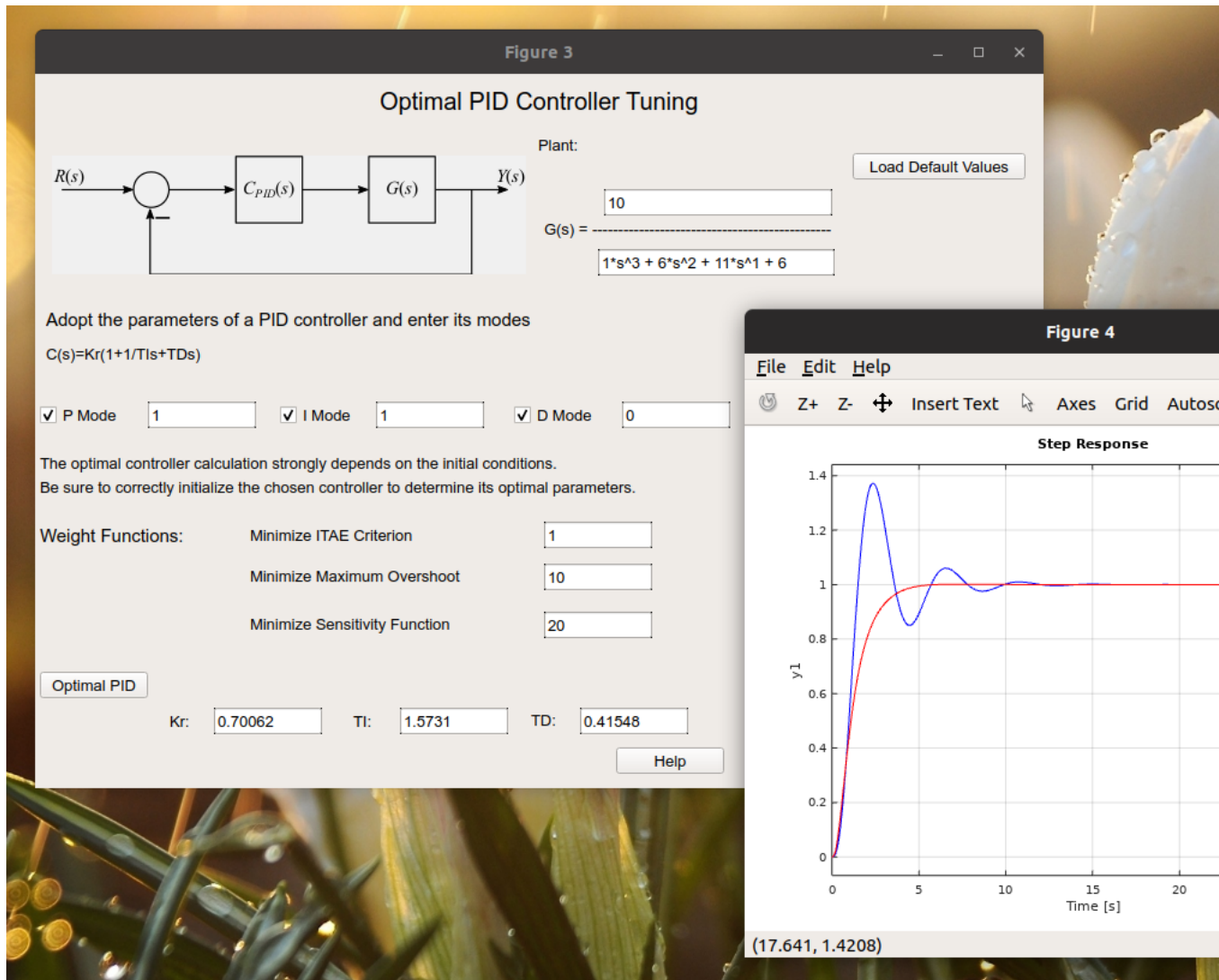


Figure 6.2: Comparative simulation between two PID controllers. With color blue it is indicated the simulation with PID controller taken as reference while with color red it corresponds to simulation of optimal PID.

# Index

- estabilidad
  - criterio de Routh-Hurwitz, 14
- ganancia estática, 7
- respuesta
  - al escalón, 6
  - al impulso, 4
  - impulsional o impulsiva, 4
  - indicial, 6

# Bibliography

- [1] E. J. Adam. *Instrumentación y Control de Procesos. Notas de Clase*. Ediciones UNL, ISBN 978-987-749-122-7, 3rd edition, 2018.