Exercises Unit 4. Graphics User Interface

Working period: Seventh and Eighth weeks
Due date: 21 April 2013

Submit only one file **`name_E4.pdf`** containing the solution to the following exercises. Include the Matlab code of the *.m file, a snapshot of the *.fig file in the GUIDE utility, and any other comment you think is necessary (for instance, property tables). Show the correct performance of your GUI printing the resulting screens for the parameter combinations.

1. GUI for spectral analysis

Remark: It is not necessary to submit the following two exercises because their code will be included in the Exercise 3.

Exercise 1. Time domain representation. Generate and plot a cosine signal **`v`** with amplitude **`A=0.5`** and frequency **`freq=5`** (Hz) in a time axis **`t`** of **`N=201`** samples, with first sample **`t0=0`** and last sample **`tf=1`** (functions **`linspace`**, **`cos`**, **`plot`**). Check that the obtained plot is the same than the first one in Fig. 1. Repeat the exercise for a square signal (function **`square`**) and for a saw tooth signal (function **`sawtooth`**).

Exercise 2. Frequency domain representation. Consider the cosine signal of the previous exercise.

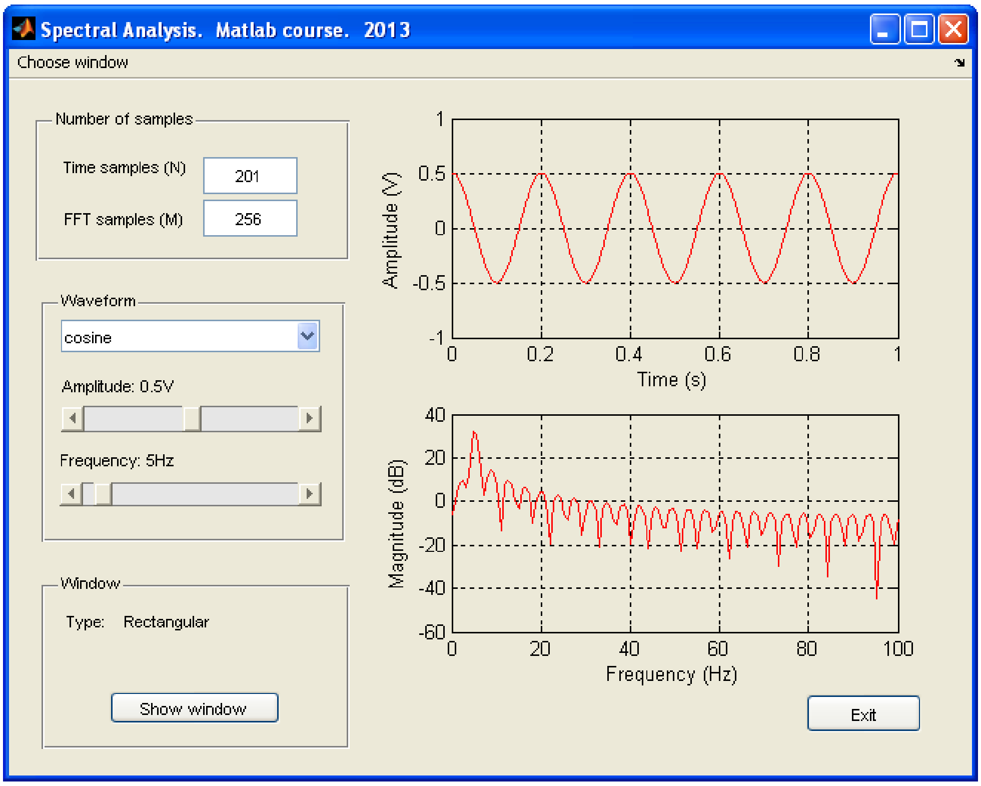
1) Compute the sampling period **`Ts=t(2)`**, and the sampling frequency (Hz), **`fs=1/Ts`**.

2) Apply a rectangular window to the signal **`v`**: To do so, generate a **`N`** samples window using **`boxcar`** and perform the dot product between **`v`** and the window. (Note: For other window types you can use the functions **`triang`**, **`hanning`**, **`hamming`**, **`chebwin`**, **`kaiser`**,...)

3) Compute the **`M=256`** points FFT (Fast Fourier Transform) of the windowed cosine (function **`fft`**). Notice that the **`M`** value is power of 2. Store the result in **`V`**. Take the first **`M/2`** samples of the **`V`** modulus and convert them to dB (functions **`:`**, **`abs`**, **`log10`**).

4) Generate a frequencies vector **`f`** of **`M/2`** points with first value **`0`** and last value **`fs/2`** (function **`linspace`**). Plot the magnitude of **`V`** (dB) at **`f`** values. Check that the obtained plot is the same than the second one in Fig. 1.

Exercise 3. Spectral Analysis. The aim is to develop an application for the spectral analysis of different waveforms. The final application must look like the figure below:



Spectral Analysis. Matlab course. 2013

Time samples (N) 201

FFT samples (M) 256

Waveform: cosine

Amplitude: 0.5V

Frequency: 5Hz

Type: Rectangular

Fig. 1. Spectral Analysis application

Performance required:

General:

- The main figure must be re-sizeable in a proportional way. Explain how to do it.
- The color of the figure, panels, axes, static texts, etc. must be the same as it is shown in Fig. 1.

Figure name:
Put your name and date in the figure name.

Axes:
There are two axes.

- *Time domain axes object*: It shows the windowed waveforms to be analyzed. The abscissas axis always range from 0 to 1s, and the ordinates axis ranges from -1V to 1V.
- *Frequency domain axes object*: It shows the spectrum of the signal shown in the time axes. The ordinates always range from -60db to 40dB and the abscissas range from 0 to fs/2 (fs being the sampling frequency).

If the “Show window” button is clicked on, the window signal (rectangular, triangular,…) must be represented (in time domain and frequency domain) instead the waveform (cosine, square,…).

“Number of samples” panel:

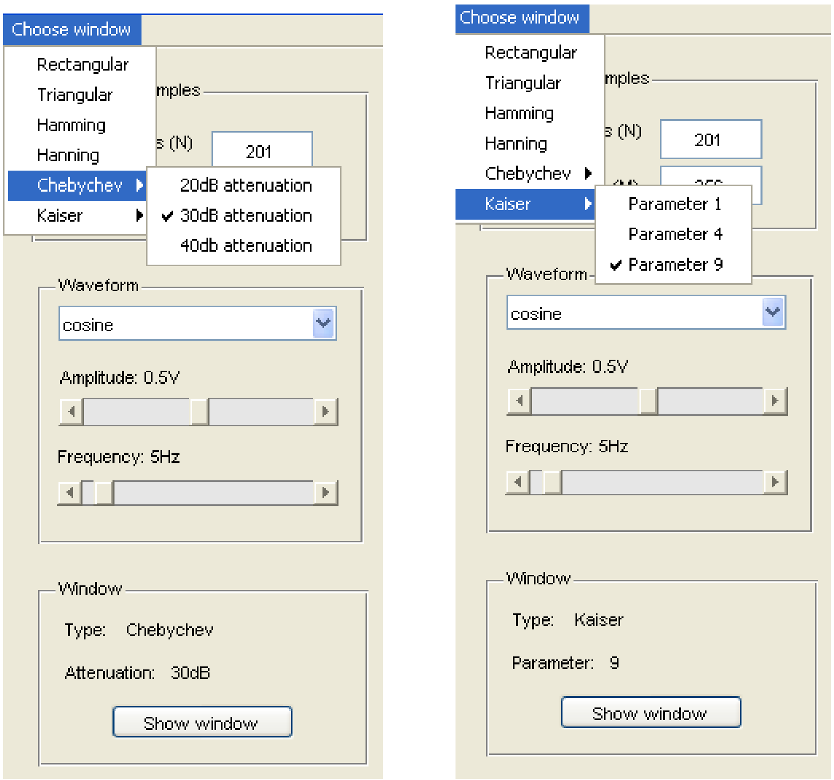
- The number of points N for the waveform must be introduced by means an edit text control. The default value is 201.
- The number of points M for the FFT can be entered by means an edit control. The default value is 256.
- Any change in N or M parameters must be immediately reflected in the two plots.

“Waveform” panel:

- The pop-up menu includes the following options: `cosine`, `square signal` and `saw tooth`.
- The sliders allow modifying the waveform amplitude (0 to 1) and frequency (0 to 100Hz). The selected values must appear in the static text objects above the corresponding sliders.
- Any change in the waveform parameters (type, amplitude or frequency) must be immediately reflected in the two plots.

“Window” panel:

- The name of the selected window signal must appear in this panel.
- If the selected window is of type “Chebychev” or “Kaiser”, another line with additional information (attenuation or parameter) must appear. See Fig. 2. If the window is any other, this additional line must be not visible.



201
cosine
Amplitude: 0.5V
Frequency: 5Hz
Type: Chebychev
Attenuation: 30dB

201
cosine
Amplitude: 0.5V
Frequency: 5Hz
Type: Kaiser
Parameter: 9

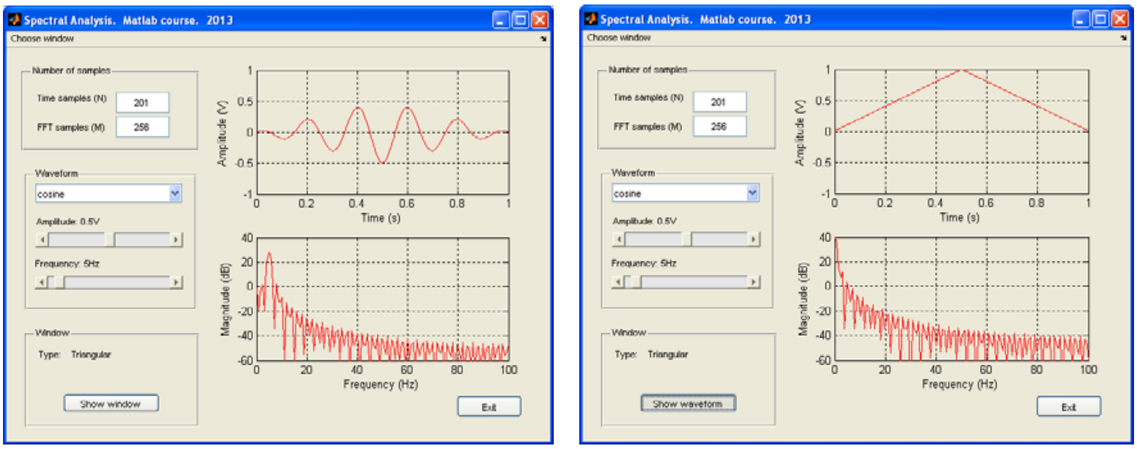
Fig. 2. Selection of the window signal

Menu bar:

- The menu bar must present the option “Choose Window” allowing the selection of the following windows: rectangular, triangular, Hanning, Hamming, Chebychev and Kaiser.
- In the case of Chebychev and Kaiser a submenu must appear. The Chebychev window can present an attenuation factor of 20dB, 30dB or 40dB. And the Kaiser window may be designed with parameter equal to 1, 4 or 9. See Fig. 2.
- Any change in the window type must be immediately reflected in the two plots.

Show window toggle button:

- When this button is activated, the axes must show the waveform and spectrum of the selected window, and the String property in the button must be “Show waveform”. See Fig. 3.



Time samples (N) 201

FFT samples (M) 256

Waveform: cosine

Amplitude: 0.5V

Frequency: 5Hz

Window Type: Triangular

Fig. 3. Show window/Show waveform toggle button

Exit push button

- This pushbutton closes the GUI

The developed application must implement all the required options. Moreover, the execution of the application must present no errors (such as grids that randomly appear and disappear, undesired axes changes, unexpected execution termination, …)

Exercise 4. Checking the GUI. To show that the GUI is correctly performing, snapshot the main figure of the GUI in the cases listed below and include them in the file name_E4.pdf.

1) Cosine, *A*=0.5V, f=40Hz, *N*=201, triangular window, *M*=256.
2) Cosine, *A*=0.5V, f=40Hz, *N*=201, Chebychev window with attenuation 30dB, *M*=1024.
3) Square, *A*=0.8V, f=100Hz, *N*=1001, rectangular window, *M*=256.
4) Sawtooth, *A*=0.25V, *f*=100Hz, *N*=501, Kaiser window with parameter 9, *M*=256. Option “Show window” activated.
5) Sawtooth, *A*=0.25V, *f*=100Hz, *N*=501, Kaiser window with parameter 9, *M*=256. Option “Show waveform” activated.
6) Cosine, *A*=1V, *f*=50Hz, *N*=501, Hanning window, *M*=1024.

2. GUI for the control of an inverted pendulum in a cart

The plant: The plant to be controlled is the inverted pendulum in a cart shown in Fig. 1. When the pendulum and/or the cart are disturbed, the system must return to its equilibrium position which is given by the values $\theta = 0^{\circ}$ and x = 0m. The disturbances acting over the plant will be modeled as initial conditions and the control signal will be the external force u acting over the cart.

<u>Data</u>: M=2kg, m=0.1kg, g=9.81m/s^2, l=2m.

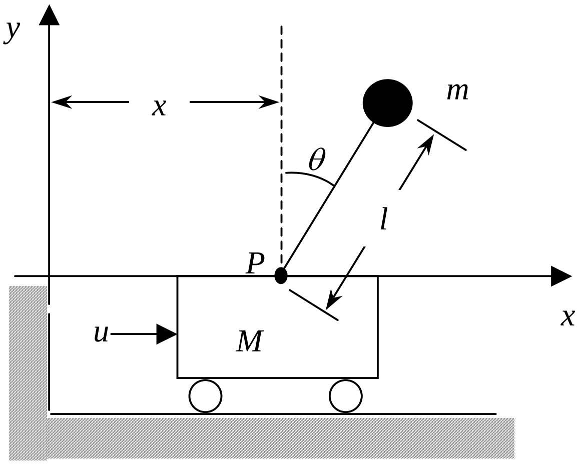


Fig. 4. Inverted pendulum and cart

The plant can be characterized in the state space by a fourth order model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{d}u$$

where the input signal is u, the output signals are $y_1 = \theta$ and $y_2 = x$ respectively, and the state vector is $\mathbf{x} = \begin{pmatrix} \theta & \dot{\theta} & x & \dot{x} \end{pmatrix}^T$, where $\dot{\theta} = \dfrac{d\theta}{dt}$, $\dot{x} = \dfrac{dx}{dt}$. The resulting state matrices are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-mg}{M} & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ \frac{-1}{Ml} \\ 0 \\ \frac{1}{M} \end{pmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{d} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Control configuration: We will use the control configuration shown in Fig. 2. For the sake of simplicity we will focus only in the inner loop (the one that compensates the pendulum inclination). The controller is a Proportional-Integral-Derivative (PID) law

$$C(s) = k_c \left(1 + \frac{1}{T_i s} + T_d s\right)$$

And, as a first selection we will take the following values: $k_c = 110, T_i = 1.1s, T_d = 0.2727s$.

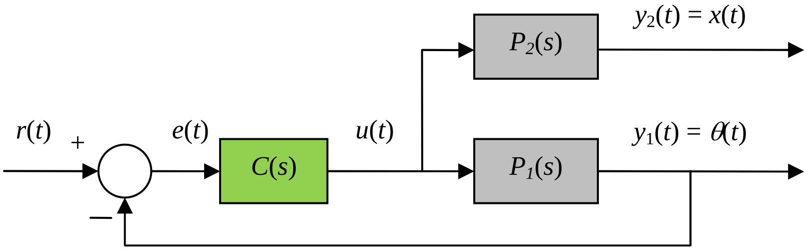


Fig. 5. Series compensation

Exercise 5. Preliminaires. The code that solves the proposed case study is available in the `inverted_pendulum.m` file.

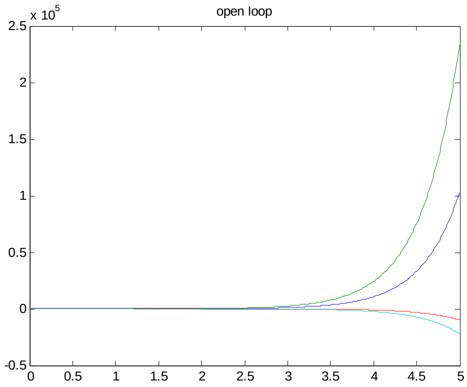
Execute this file and verify that the results are the following:

Figure (a) shows the dynamical behaviour of the 4 states. Clearly, with no control (i.e. in open loop), the system is unstable.

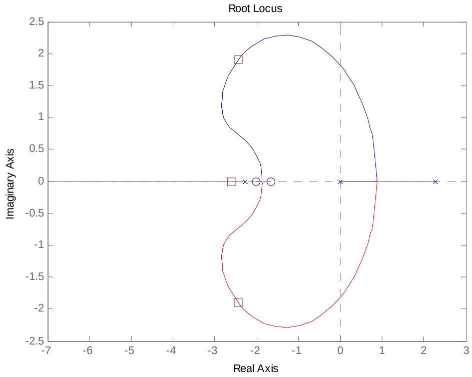
Figure (b) shows the root locus in the complex plane (function **rlocus** or **rlocusplot**). The Ti, Td values establish the position of the loop zeros. In the plot we can see which the closed loop poles are for values of kc ranging from 0 to infinity. In particular, the red squares show the poles for kc=110. These poles are the following:

```
system_poles =
  -2.4483 + 1.9004i
  -2.4483 - 1.9004i
  -2.6026
```

Since all the poles are located in the left half plane (LHP) of the complex plane, the resulting system is stable



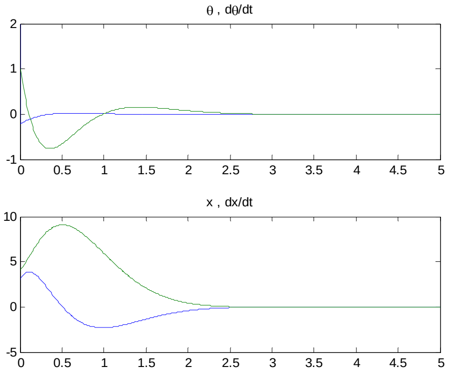
(a)



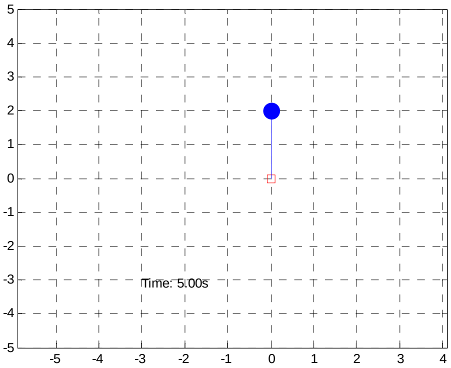
(b)

Figure (c) shows the time evolution of the four states. As expected, they all finish at the zero value (equilibrium point).

Finally, Figure (d) simulates the behaviour of the system by means of an animation effect.

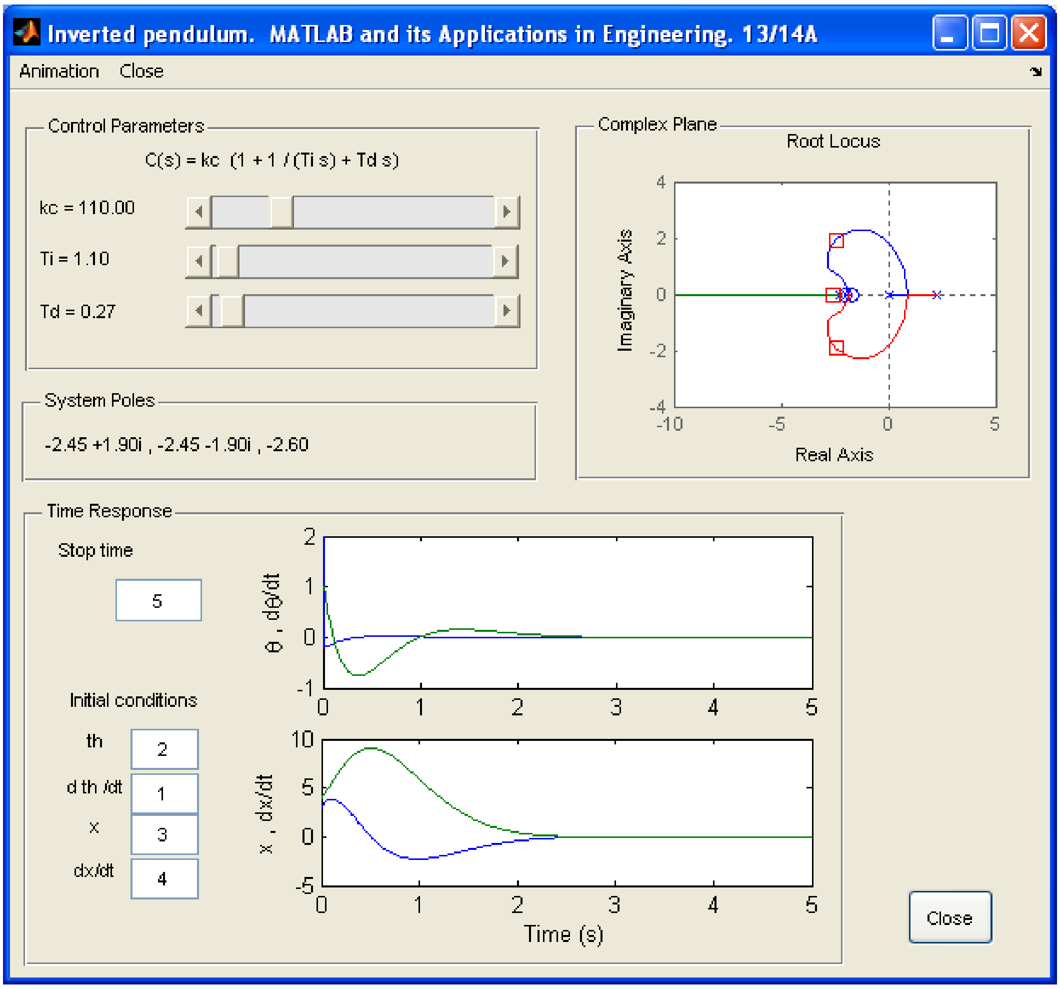


(c)



(d)

Exercise 6. Graphics user interface. The objective of the practice set is to develop a GUI to study the effect in the system behavior of different selections of the values kc, Ti, Td. The GUI must look like the one in Fig. 3. The internal instructions must be the ones in the **`inverted_pendulum.m`** file (the GUI serves only as a way to change the PID parameters and see what happens to the system poles and time response).



Inverted pendulum. MATLAB and its Applications in Engineering. 13/14A

Control Parameters

C(s) = kc (1 + 1 / (Ti s) + Td s)

kc = 110.00

Ti = 1.10

Td = 0.27

System Poles

-2.45 +1.90i , -2.45 -1.90i , -2.60

Time Response

Stop time: 5

Initial conditions

th: 2

d th /dt: 1

x: 3

dx/dt: 4

Fig. 3. GUI to be developed

The GUI must present the following features:

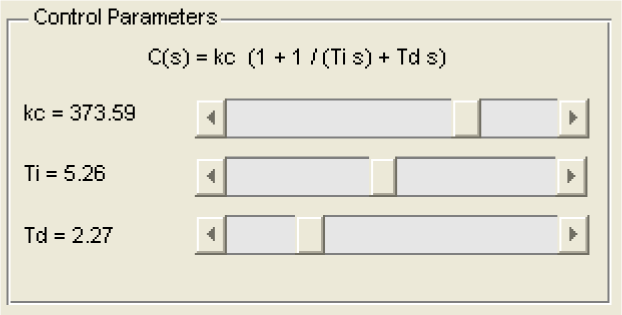
General:

- The main figure must be proportionally re-sizeable.
- The color of the static texts must be the same than the in figure and panels.
- Put your name and date in the main figure ("Name" property).
- If you click on the "Close" button or select the "Close" option in the menu bar, all the figures (main and animation) must be closed.
- When the GUI is started, the initial values must be the ones shown in Fig. 3.

"Control Parameters" panel:

- The parameters' values are changed by means of the sliders. The kc value ranges from 1 to 500, Ti ranges from 1 to 10, and Td ranges from 0 to 10.

- The value must be immediately indicated in the corresponding static text with a precision of 2 decimal digits. Also, the effect must be immediately reflected in the remaining panels.



Control Parameters

C(s) = kc (1 + 1 / (Ti s) + Td s)

kc = 373.59

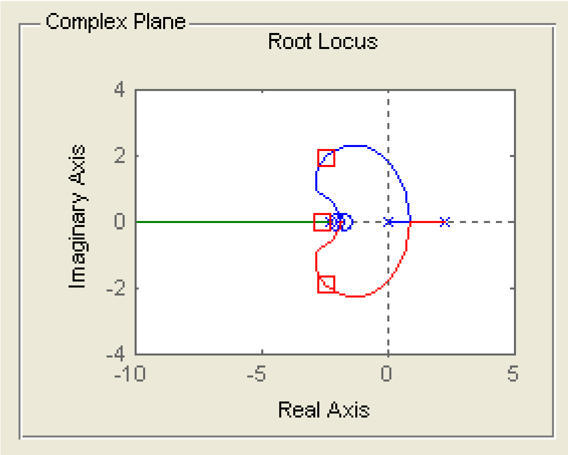
Ti = 5.26

Td = 2.27

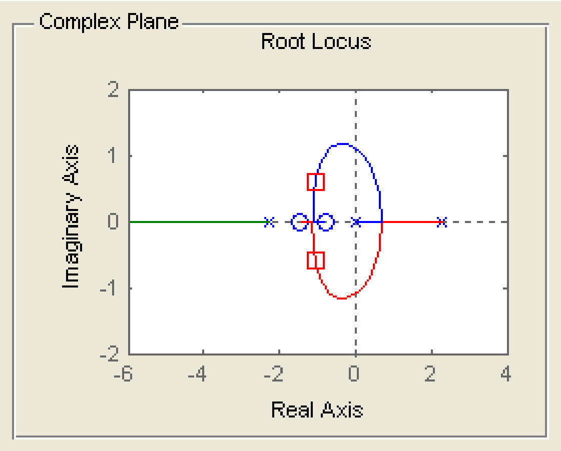
“Complex Plane” panel:

Any change in the PID parameters must be reflected in this panel. The root locus must be represented and, over it, red squares must indicate which the system poles are for the current value of kc

kc=110, Ti=1.1, Td=0.27

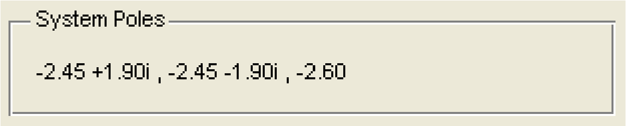


kc=107.45, Ti=1.90, Td=0.43



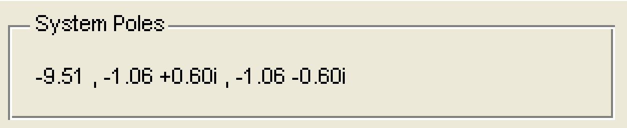
“System Poles” panel:

Any change in the PID parameters must be reflected in this panel. The system poles are separated by commas and with a 2.decimal digit precision.



System Poles

-2.45 +1.90i , -2.45 -1.90i , -2.60

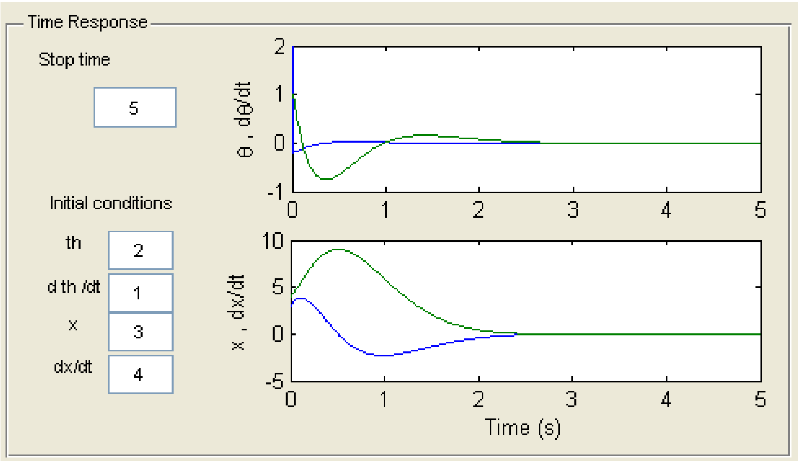


System Poles

-9.51 , -1.06 +0.60i , -1.06 -0.60i

“Time Response” panel:

It shows the time evolution of the four states. Any change in the “Stop time” or “Initial conditions” must be immediately reflected in the plots.



Time Response

Stop time: 5

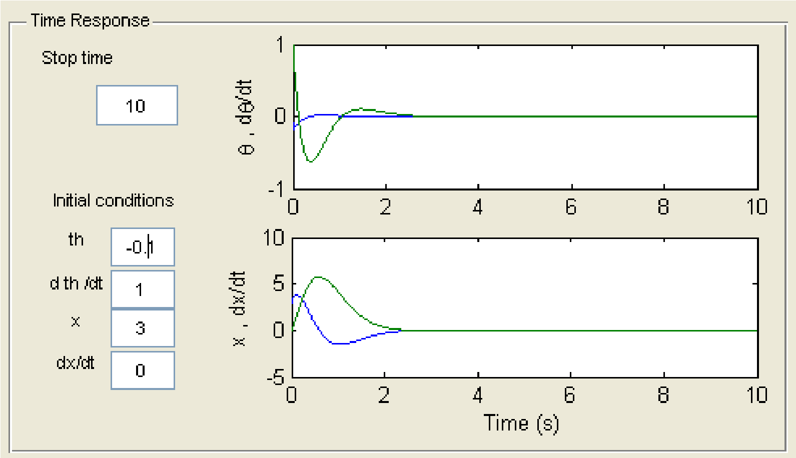
Initial conditions

th: 2

d th /dt: 1

x: 3

dx/dt: 4



Time Response

Stop time: 10

Initial conditions

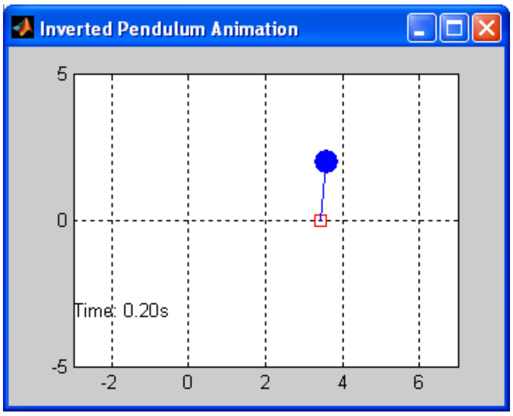
th: -0.1

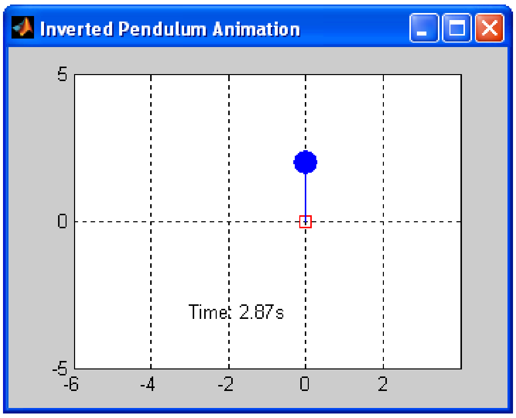
d th /dt: 1

x: 3

dx/dt: 0

Animation figure:
When the menu item “Animation” is selected, a new figure must appear showing the animation effect of Exercise 1.





Other features:
Optionally, you can improve anything you want of the proposed GUI.

Exercise 7. Results. To show that the GUI is correctly performing, snapshot the main figure and the animation figure in several cases (stable, unstable, fast or slow poles, different initial conditions and stop times…) and include them in the pdf file.