

RSCFB

Rotation Speed Control
with Flexible Binding

Lukáš Hubka, Petr Školník
Steffen Gärtner, Sebastian Fleischer

Liberec 2013

RSCFB

Rotation Speed Control
with Flexible Binding

Lukáš Hubka, Petr Školník
Steffen Gärtner, Sebastian Fleischer

Liberec 2013



CONTENT

1	Test objectives	3
2	Description of the process.....	4
3	Determination of the controller for an oscillatory system	6
3.1	Experimental controller setting	6
3.2	Synthesis of the root locus.....	6
3.2.1	Notes on root locus.....	6
3.2.2	Importance of root locus in the design of control loops	7
3.2.3	Use of Matlab program package 'SISOTOOL'	8
3.2.4	Implementation of the root locus method	11
3.2.5	RL tuning example – PI controller	11
3.2.6	RL tuning example – PID controller	17
4	Experimental procedure.....	19
4.1	Determination of the structure of the loop to be modeled	19
4.2	Determination of the steady state behavior	20
4.3	Experimental identification	21
4.3.1	Identification measurement.....	21
4.3.2	Transfer function parameters estimation	23
4.4	PID control.....	25
4.5	Investigation of the real control	28
4.6	Root Locus application to PID tuning	28
5	Literature	30
	Appendix A – Steady state measurement VI	31

1 TEST OBJECTIVES

The main goal of the task is to control the underdamped system (Fig. 1). The whole task consists of several steps.

- 1) Analysis of the controlled system.
 - a. Determinations of the advanced control system, finding of the steady state behavior of the process.
 - b. Determinations of the transfer function of the advanced control system based on the experimental identification.
- 2) Creating of a closed loop model using the LabView.
- 3) Determinations of controller setting values to the model
 - a. by experimental testing,
 - b. with the root locus method.
- 4) Investigation of the closed loop with the controller setting values detected
 - a. on the model,
 - b. on the real system.



Fig. 1: Picture of the controlled process

2 DESCRIPTION OF THE PROCESS

The laboratory experiment consists of a speed control amplifier (V), a controlled system (M + EK + TG), a converter and a PC (Fig. 2). Simulation and optimization is provided with appropriate software (Matlab and/or LabView [4, 5, 6, 7]) for data processing. The PC is connected to the system via an I/O data acquisition board. The measurements on the real-time system are performed by the I/O card with LabView add-in toolkit. It is possible to use a LabView program to develop and simulate the behavior of various types of controllers (PID controller, fuzzy controller, etc.) directly on the PC. The identification is carried out using a LabView function from the Control Design/Identification Toolkit.

The controlled system is a dynamic system (Fig. 2 – marked rectangle). This dynamic system consists of the DC motor M, which is connected by a flexible binding EK with the tachogenerator¹ TG. The DC motor is excited by the controlled voltage u_M . The output signal is the voltage of the tachogenerator u_{TG} . The control variable is the speed of the speedometer generator n , which is unfortunately not measured directly, but is proportional to the output voltage from the tachogenerator.

The functional diagram of the entire control system with all amplifiers, all transformers, the power supply, the I/O card and the PC is shown in Fig. 2. The PC input is the normalized digital (feedback) signal $y(kT)$. The normalized digital control variable $u(kT)$ is the output from the PC. The output can be signal from the controller (in a case of the closed loop test) or output signal from a generator jump (in a case of some process studies).

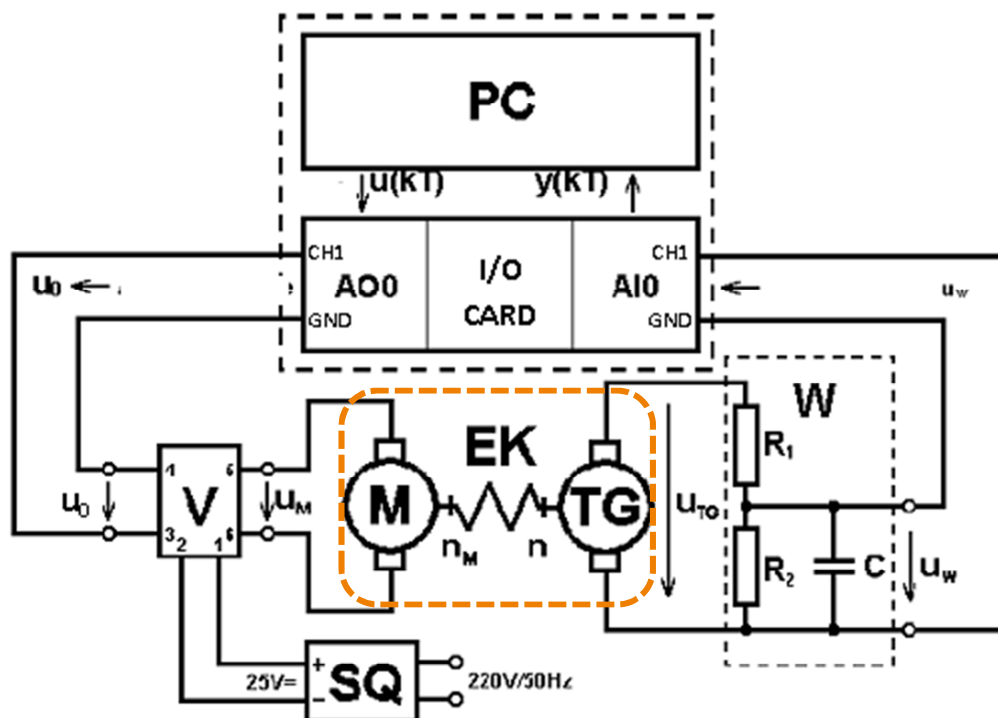


Fig. 2: Functional schema of the process and the connection with the PC

¹ The tachogenerator is build up as a DC motor which is operating in the generator mode. The output voltage is directly proportional to the rotation speed.

Tab. 1: Device description

Symbol	Description
M	DC motor with permanent excitation, type P2TV369, 24 V, 2000 rpm
TG	tachogenerator, type P2TV369, 24 V, 2000 rpm
EK	elastic coupling
V	amplifier and power actuator
W	converter (voltage divider) and low pass filter
SQ	power source, ZPA Decin, 25 V/10 A
R1	electrical resistance of the voltage divider, 33 k Ω
R2	electrical resistance of the voltage divider, 8 k Ω
C	filter capacitor
I/O	I/O card in the PC
A00	analog output of the I/O card
A10	analog input of the I/O card

Tab. 2: Signal description

Symbol	Description	Value
$u(kT)$	normalized digital manipulating variable in the PC	0 .. 10 V
u_o	voltage from the analog output of the I/O card	0 .. 5(10) V
u_M	output voltage of the amplifier	0 .. 24 V
u_{TG}	output voltage of the tachogenerator	0 .. 24 V
u_w	output voltage of the converter	0 .. 4.68 V
$y(kT)$	normalized digitized feedback signal (Matlab)	0 .. 10 V
n_M	engine speed	0 .. 2000 rpm
n	variable speed of the tachometer generator	0 .. 2000 rpm

The input channel of the I/O card converts the output of the converter u_w to a value $y(kT)$ in the ranging from 0 V to 10 V. The output channel of the I/O card generates the control voltage for the DC motor. The output channel is driven by a value $u(kT)$ in the range 0 V to 10 V from the PC and outputs a voltage u in the range of 0 V to 10 V.

3 DETERMINATION OF THE CONTROLLER FOR AN OSCILLATORY SYSTEM

A number of methods and rules are now available to the control engineers for the determination of the controller setting values. Many of them are popular, such as the rules according to Ziegler-Nichols, the optimum module, the synthesis of the root locus method, a setting by Samal or parameter optimization using a quality criterion. Not all methods are applicable to the group of oscillatory systems.

Two suitable methods for oscillatory systems are

- 1) manual setting,
- 2) synthesis by root locus method.

3.1 EXPERIMENTAL CONTROLLER SETTING

The simplest method for the given controlled process is the setting by hand. This process takes place at a controller with the transfer function by the following procedure.

- 1) **Models of the closed loop or a real control loop are available.**
- 2) **I and D components are set to zero.**
- 3) **The P component is increased until the required speed is achieved.**
- 4) **The steady state error is eliminated with the increase of I component.**
- 5) **Using the D component is set the attenuation of the closed control loop.**

This simple procedure can be achieved by a logical, successive procedure to achieve good results.

3.2 SYNTHESIS OF THE ROOT LOCUS

The method for determining the movement of the poles in the s -plane of the closed control loop in response to a variable parameter, such as the amplification factor is known as the root locus method. This is a classical method for the analysis and synthesis of linear control loops.

3.2.1 Notes on root locus

Surprising properties of the closed loop are derived from the known properties of the open control loop in this method. The overall transfer function of a linear open loop has the form

$$F_o(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (1)$$

and can be converted into the following form

$$F_o(s) = k \frac{(s - s_{b1})(s - s_{b2}) \dots (s - s_{bm})}{(s - s_{a1})(s - s_{a2}) \dots (s - s_{an})} = k \frac{Z_o(s)}{N_o(s)}, \quad k > 0, \quad (2)$$

where s_{bi} are zeros (roots) and s_{aj} are poles (may be complex values) of the open loop and k is a parameter.

The open loop is described by the following transfer function (see Fig. 3)

$$F_o(s) = R(s) \cdot F_s(s) = \text{Gain} \cdot R_0(s) \cdot F_s(s) = \text{Gain} \cdot \frac{Z(s)}{N(s)}, \quad (3)$$

where $F_o(s)$ is the transfer function of the open loop, $F_s(s)$ is the transfer function of the advanced control system, $R(s)$ is the transfer function of the controller, $R_0(s)$ is the transfer function of the basic structure of the controller with poles and zeros, Gain is overall gain of the controller (affects all parts of the controller), corresponds to the parameter k in (2), $N(s)$ is the denominator polynomial of the transfer function $F_o(s)$ and $Z(s)$ is the numerator polynomial of the transfer function $F_o(s)$.

The poles and zeros of the open loop transfer function are built from the poles and zeros of the system and from the poles and zeros of the controller.



Fig. 3: Open loop (left), closed loop (right)

The transfer function of the closed loop behavior (see Fig. 3) is

$$F_{cl}(s) = \frac{F_o(s)}{1 + F_o(s)} = \frac{\text{Gain} \cdot R_0(s) \cdot F_s(s)}{1 + \text{Gain} \cdot R_0(s) \cdot F_s(s)} = \frac{\text{Gain} \cdot Z(s)}{N(s) + \text{Gain} \cdot Z(s)}. \quad (4)$$

The poles of this function are thus the solutions of the (characteristic) equation

$$N(s) + \text{Gain} \cdot Z(s) = 0. \quad (5)$$

The zeros are determined from the numerator by following equation

$$\text{Gain} \cdot R_0(s) \cdot F_s(s) = \text{Gain} \cdot Z(s) = 0. \quad (6)$$

The locus (in the complex plane) of all solutions (roots) of equation (5) in response to a real parameter $\text{Gain} \in \langle 0; \infty \rangle$ is referred as root locus (RL). The number of solutions is equal to the number of individual paths (branches) in the RL representation. A function value exists on each path for a given $\text{Gain} = k_i$. Further accounts of the RL and construction of the RL are contained in the literature [1, 2].

3.2.2 Importance of root locus in the design of control loops

The behavior of a control loop is determined by the position of the poles and zeros of the transfer function of the closed loop (4) in the complex plane. As an example, the stability criterion (Hurwitz) says: 'All zeros of the denominator (poles) must be on the left from imaginary axes'. These and other statements about the behavior of the closed loop on the basis of the RL can be made as a function of the gain. The location of the zeros and poles of the controller and the RL are shown for a closed loop assembled from a given path (transfer function) and the type of the controller (P, PID, etc.) which is determined by the number of zeros and poles of the control transfer function. The RL shows the location of the zeros and poles of the circuit and the controller in the transfer function as a function of the parameter $\text{Gain} = k$. For a favorable control action is aimed a positions of the poles and zeros. E.g.: For an oscillatory system the zeros of the controller should be as close as possible to the complex axis.

The structure of the controller is defined by the poles and zeros of the transfer function $R_0(s)$ in the synthesis. The desired dynamics of the closed control loop is set by changing the overall gain of the controller gain. With a deliberate, gradual approach can be good values for find the controller's pole's and zero's positions. Here, the position of the poles and zeros vary the controller.

The controller setting values are calculated from the found values of the poles and zeros and the controller gain for the desired control behavior. Controller setting values in each component of the controller are multiplied for the determination of the gain. A description of this process with an application in Matlab can be found in [5]. This experiment is prepared to represent and work with the RL. The Matlab / Simulink software package 'SISOTOOL' is available [5].

3.2.3 Use of Matlab program package 'SISOTOOL'

The transfer function of the extended controlled system and the structure of the controller (e.g. PI, PID, PD) are necessary for the synthesis of the control loop. The structure of the controller is specified by the number of poles and zeros of the transfer function of the controller $R_0(s)$.

For demonstration purposes, the transfer function $F_S(s) = \frac{1}{s^2+2s+1}$ is presented as LTI model in Workspace. To create the LTI model in the Matlab command window use the command

```
sys = tf ([1], [1 2 1])
```

Starting the program 'SISOTOOL' and loading of the system model

Type the command

```
sisotool
```

in the Matlab command window and start the application by pressing 'Enter'. The window 'Control and Estimation Tools Manager' (Fig. 4) and 'SISO Design for SISO Design Task' (Fig. 5) are opened. The window 'Control and Estimation Tools Manager' includes several tabs. The most important are tabs 'Architecture' and 'Compensator Editor'. It is possible to edit the control loop, their structure and their part in the 'Architecture' tab. The current control loop structure is formed by these LTI models:

- controller C,
- system G,
- measuring device H
- and reference pre-filter F.

The initial setting of the control structure loop is suitable for our purpose and it is not recommended to change it.

The next step is to set the system transfer function in this tool. The window 'System Data' (Fig. 6) is opened by clicking on the 'System Data' in the 'Control and Estimation Tools Manager', tab 'Architecture'. To get the new window 'Import Model' (Fig. 6) and also to get the possibility to define, resp. to transport the system transfer function from the workspace click the 'Browse' button. After that the actual workspace (only existing LTI models) is displayed and the right system transfer function can be marked. Finally click 'Import' to take the transfer function over.

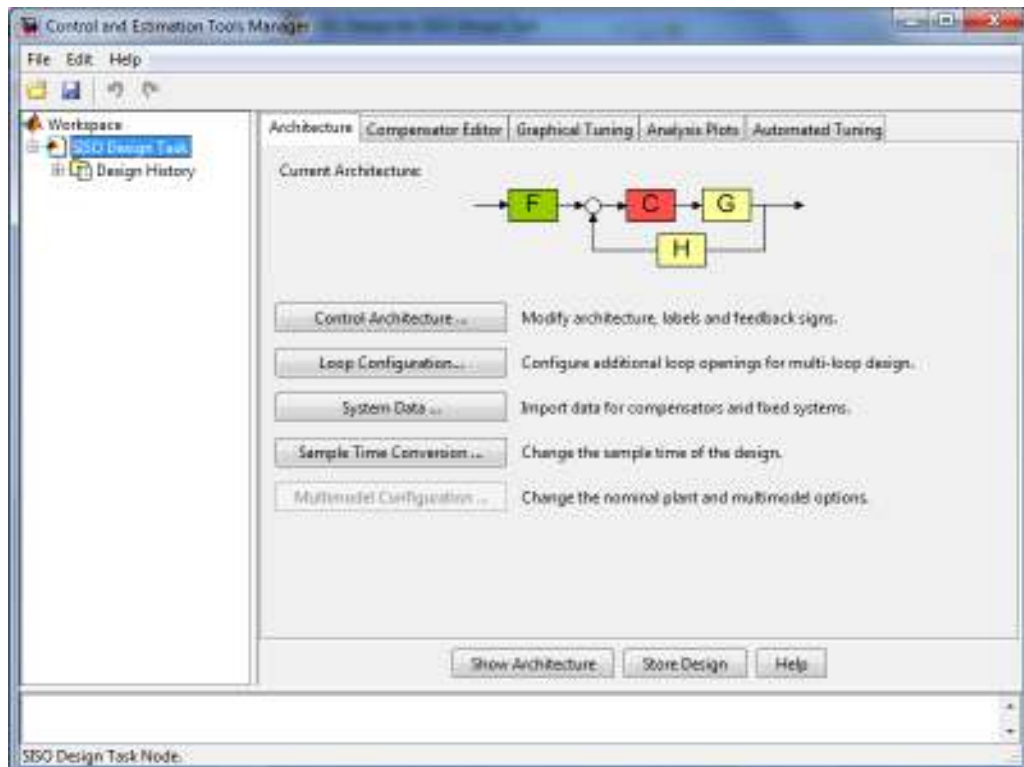


Fig. 4: 'Control and Estimation Tools Manager' window

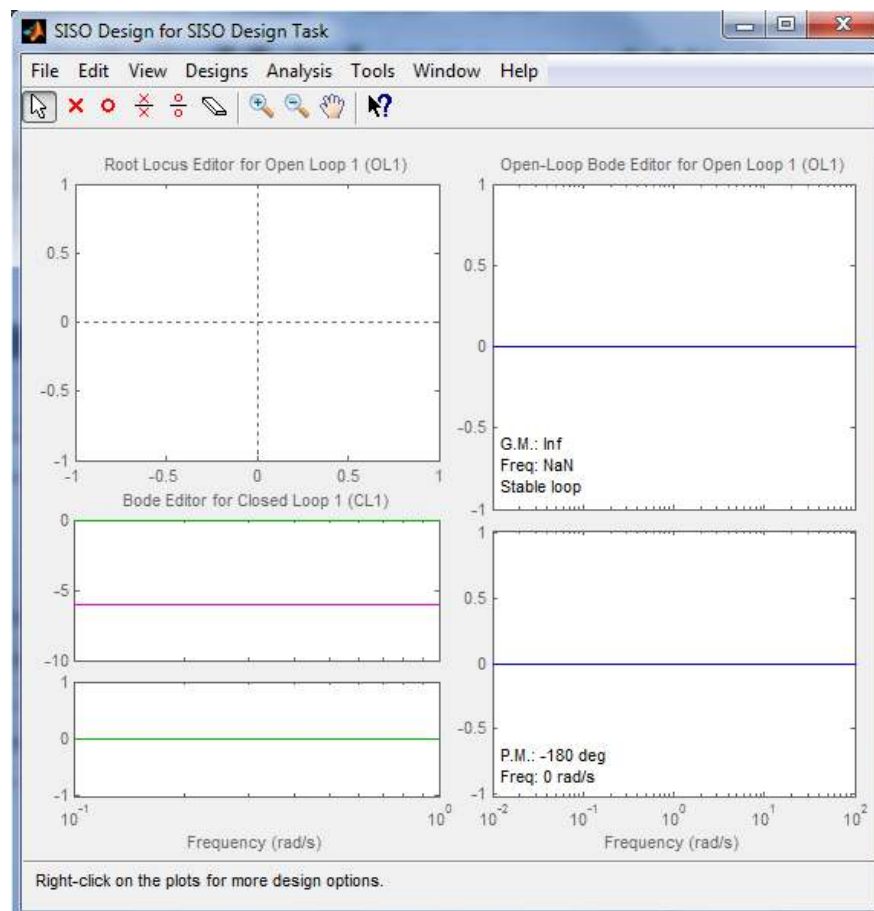


Fig. 5: 'SISO Design for SISO Design Task' window

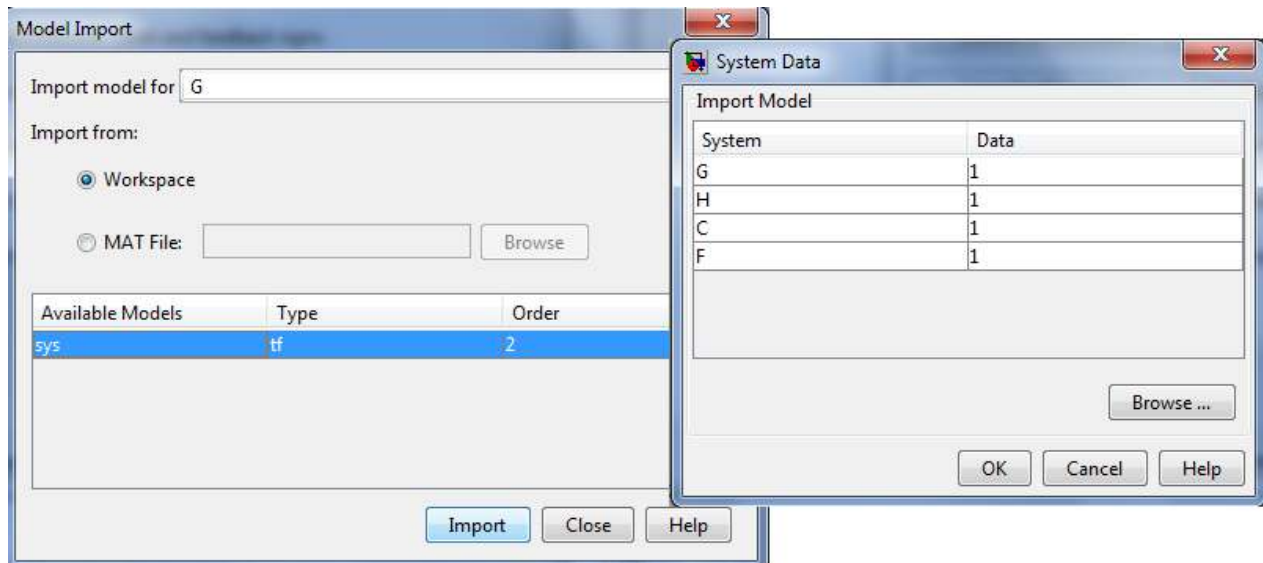


Fig. 6: 'Model Import' window (left) and main 'System Data' choosing window

To summarize, the process of importing the model transfer function from the workspace into the 'SISOTOOL' has these steps:

- Mark the row with 'G' in 'System Data' window and click to 'Browse'.
- Choose the 'G' from the pull-down menu 'Import model for' (initial selection).
- Click to the offered model (in our example 'sys').
- The import is performed by clicking the 'Import' button.
- Finally the 'Model Import' window can be closed by clicking to 'Close' and the 'System Data' window can be closed too by clicking to 'OK'.

The effect of the transport of your transfer function into the tool is noticeable in the graphical window 'SISO Design for SISO Design Task'. There are now trajectories of the root locus and the Bode characteristics of open and closed loop of the imported model (Fig. 7).

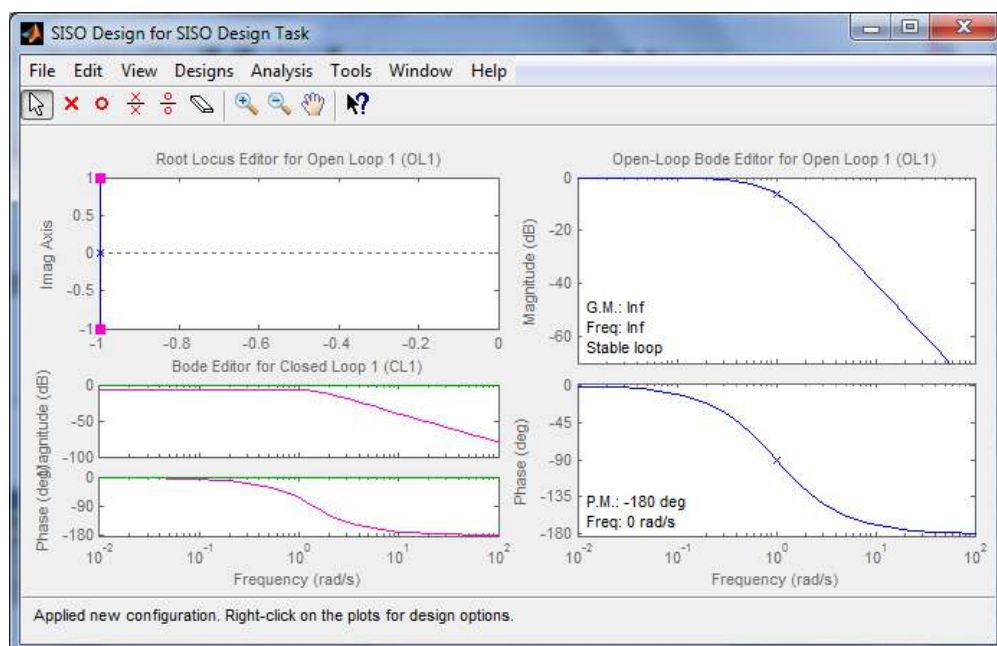


Fig. 7: The window with root locus after the system transfer function importing

3.2.4 Implementation of the root locus method

Once the system model is transported the controller setting values will be determined in the foregoing clause for a PID controller by the following procedure:

- a) **P controller:** The I and D components of the controller remain zero, no zeros and no poles are added. The gain 'Gain' is set so that the closed loop has the required speed (settling time TS).
- b) **PD controller:** The D component is adjusted by inserting a (real) zero in the transfer function of the controller. The location of the zero is varied until the desired damping of the closed control loop is achieved. Finally the gain can be changed.
- c) **PI controller:** The steady state error $e_w(\infty) = \lim_{t \rightarrow \infty} e_w(t) = 0$ and the integration character will be achieved by adding a (real) pole (integrator).
- d) **PID controller:** To create a PID controller, the integrator and two zeros has to be placed into the controller structure. It is necessary to choose if zeros will be real values or complex pair.

Generally, the zero of the PI controller or two zeros of the PID controller can be placed near the poles of the controlled system, which are closest to the imaginary axis.

3.2.5 RL tuning example – PI controller

The PI controller is created as follows:

- The clicking on the tab 'Compensator Editor' in the 'Control and Estimation Tools Manager' the window 'Compensator Editor' is opened (Fig. 8).
- Click the right mouse button inside the 'Dynamics' field and a pull-down menu is opened. Click on 'Add Pole / Zero' to open the second pull-down menu (Fig. 8). This menu is a list of possible selectable controller components (real or complex poles, real or complex zeros, integrator, lead, lag or notch structure).
- Click to the 'Integrator' to add the I part of the controller (the pole in the zero).
- The PI controller has a zero. To complete the PI structure repeat once more the right click inside the 'Dynamics' field and add real zero ('Add Pole / Zero' > 'Real Zero').
- The final structure after taking the integrator and the zero is representing by the controller transfer function $C = Gain \cdot \frac{(1+s)}{s}$ in the 'Compensator' field (Fig. 9).
- The actual position of zero can be manually change by editing the location of the zero in the 'Edit Selected Dynamics' field after marked the zero. The better idea is to change the position of the zero in the graphical tool by drag & drop (described further).

Only the 'Root Locus' can be used form possible forms of representation in the 'SISO Design for SISO Design Task' (Fig. 7) to final controller tuning. This is done by changing the setting in the 'Control and Estimation Tools Manager' in the 'Graphical Tuning' tab. Select 'Root Locus' for the 'Plot 1' (initial setting) and all other plots set to 'None' (Fig. 10). The effect is immediately noticeable on the 'SISO Design for SISO Design Task' window (click to 'Show Design Plot'), where only root locus is displayed (Fig. 11).

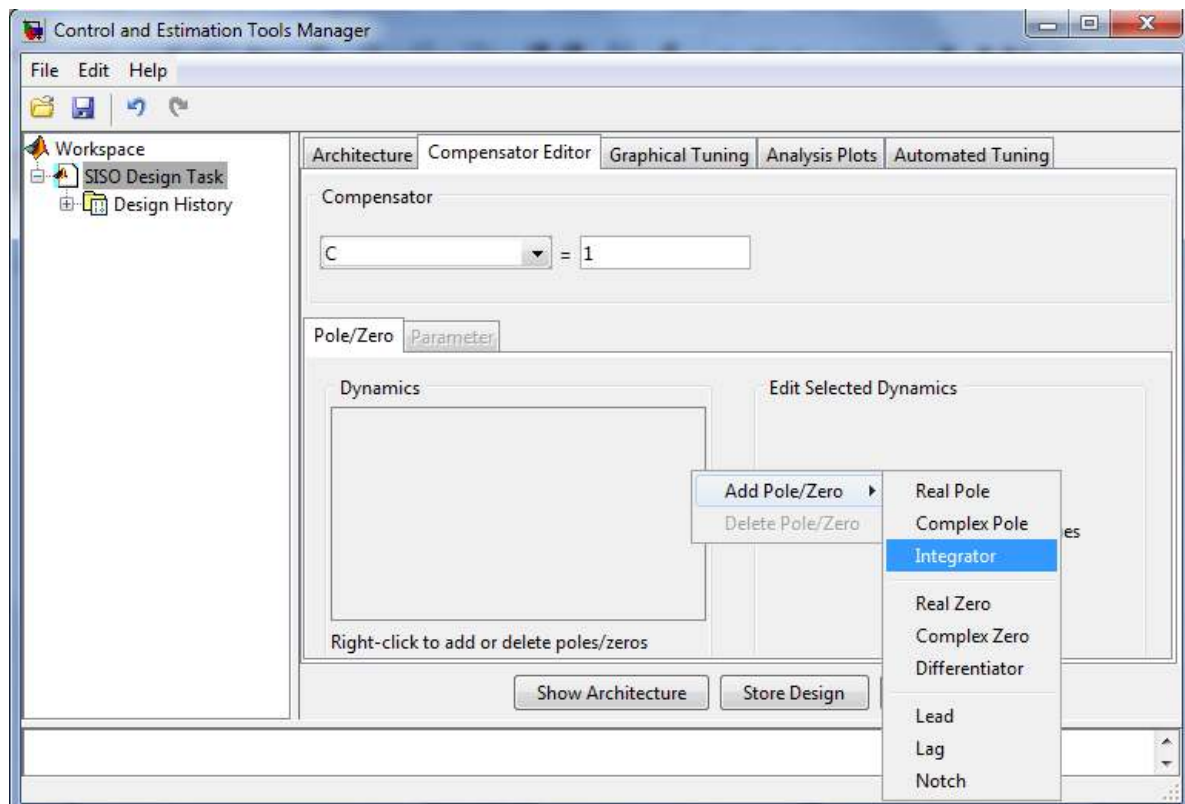


Fig. 8: 'Compensator Editor' window

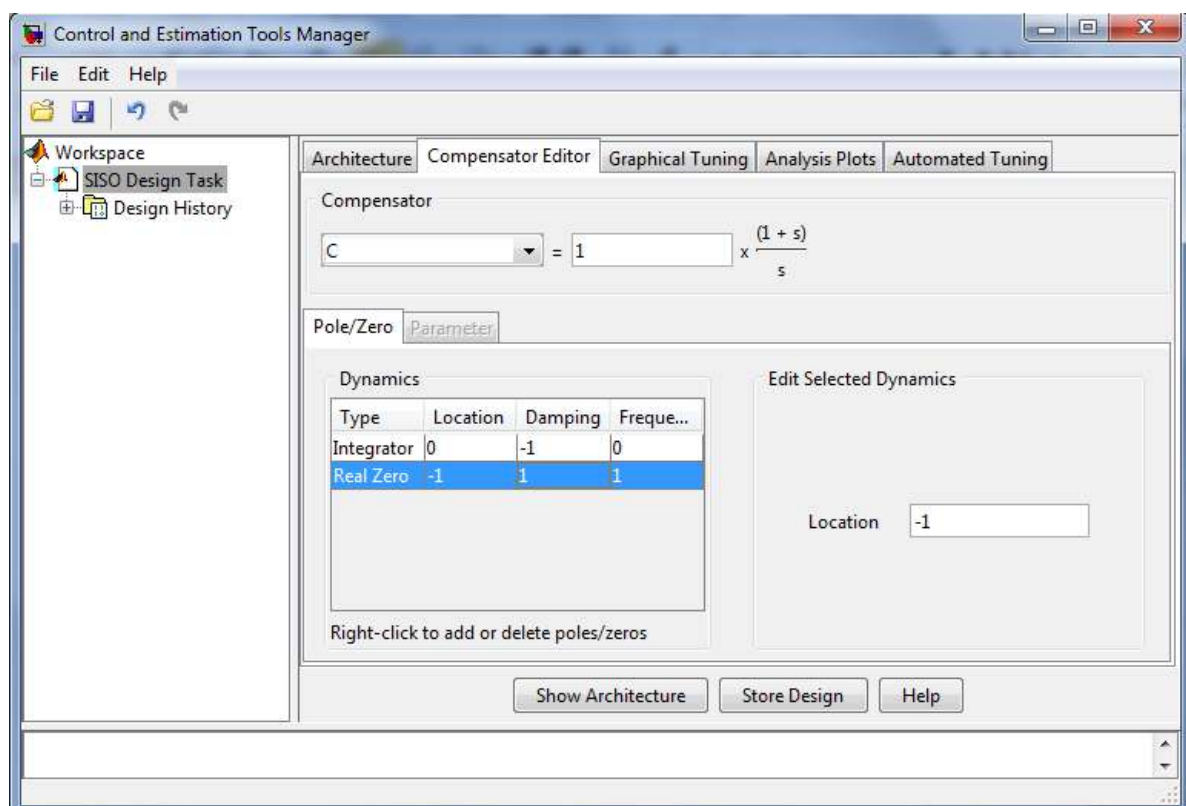


Fig. 9: The 'Compensator Editor' window with PI controller initial structure and setting

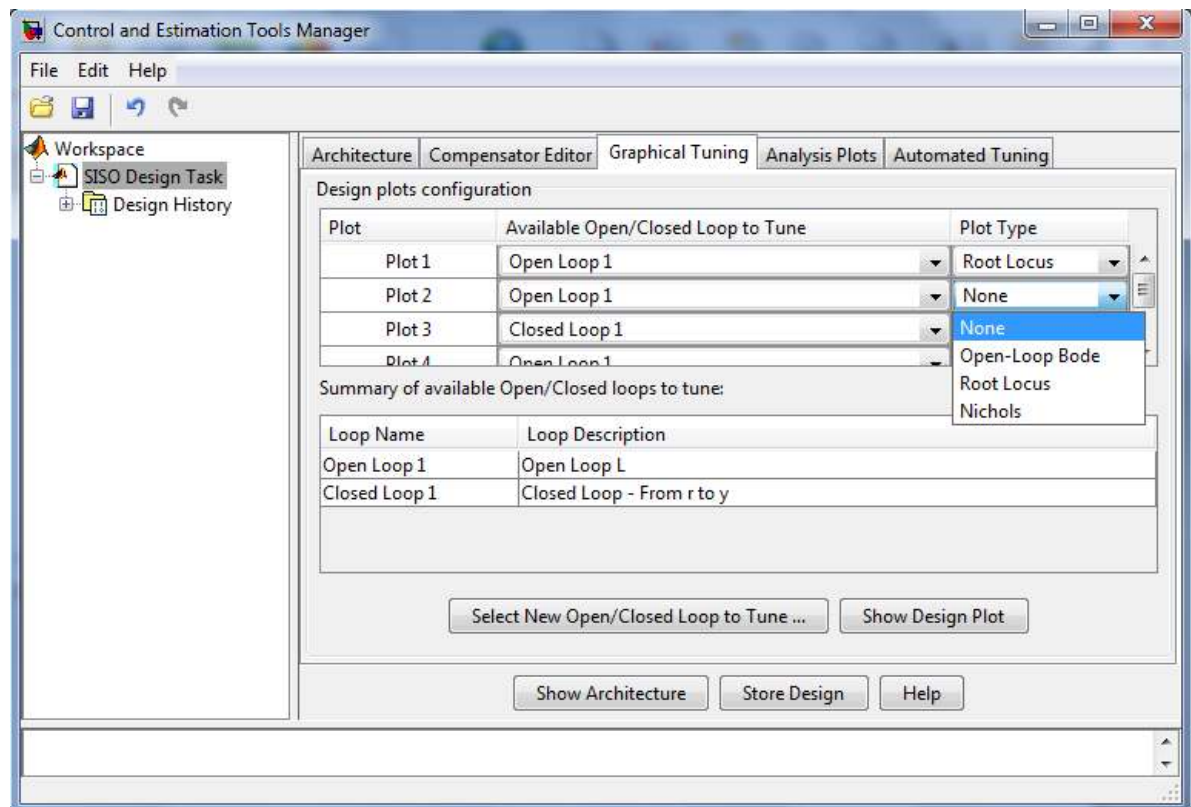


Fig. 10: 'Control and Estimation Tools Manager' window, the 'Graphical Tuning' tab

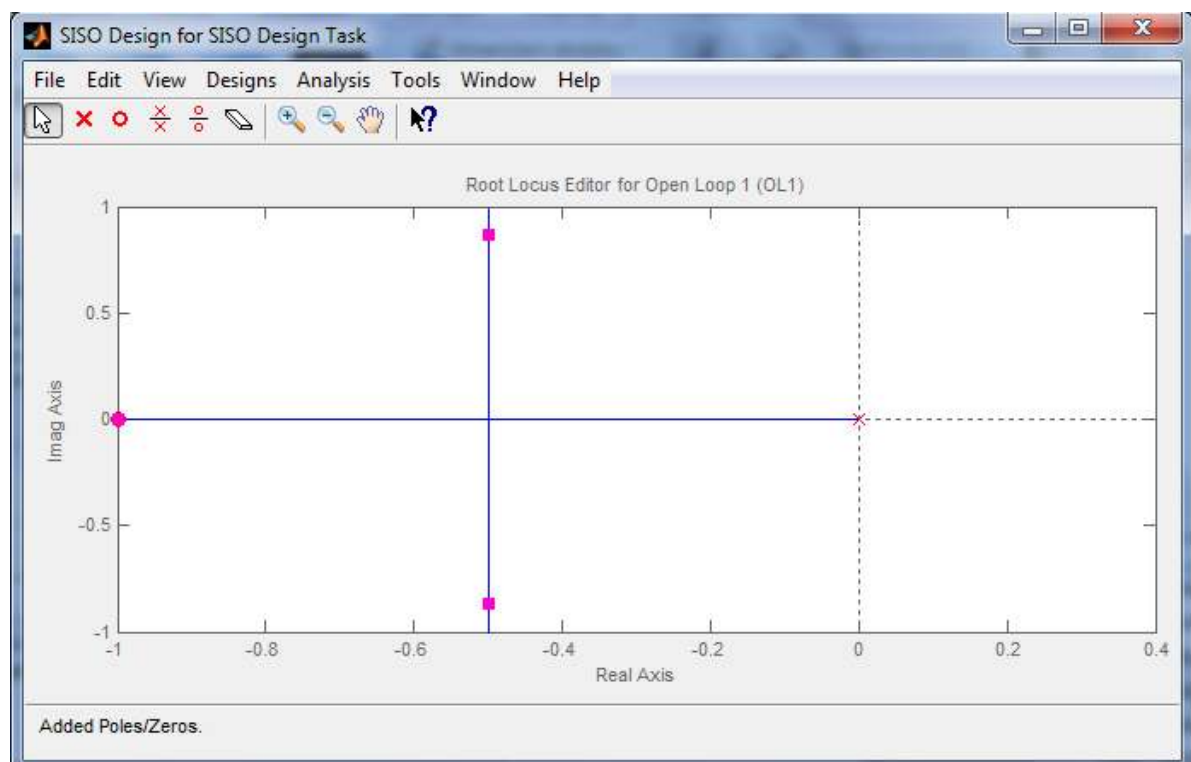


Fig. 11: Window with root locus

This illustration (Fig. 11) shows the complex plane with the poles, zeros and paths of the RL for the used transfer function. The actual zeros and poles of the controller are marked by red circles and cross. The paths of the closed loop system poles are shown as blue lines. The actual gain is

illustrated by red filled square. The values of poles and zeros can be changed by dragging the mouse. The gain can be changed by the same way. The actual numerical representation of the controller is displayed in the window 'Control and Estimation Tools Manager' in the tab 'Compensator editor' under 'C'.

The dynamic behavior of the actual control loop (controller setting) is to best to verify with the help of the step response. For that it is possible to select from the main menu of the 'SISO Design for SISO Design Task' the tab 'Analysis' and from the pull-down menu 'Response to Step Command' (Fig. 12). A new window 'LTI Viewer for SISO Design Task' is opened (Fig. 13). The initial choice of plotted lines is an output response $y(t)$ from the controlled process to step change on the reference (blue) and appropriate action value $u(t)$ which realize this output. It is possible to edit or change this option by two ways. The immediate way is through the right click on the graph (Fig. 13). The menu is opened and a different type of plot or different response (under 'Systems') can be chosen. The same possibility is in the main 'Control and Estimation Tools Manager' window under the tab 'Analysis Plots'.

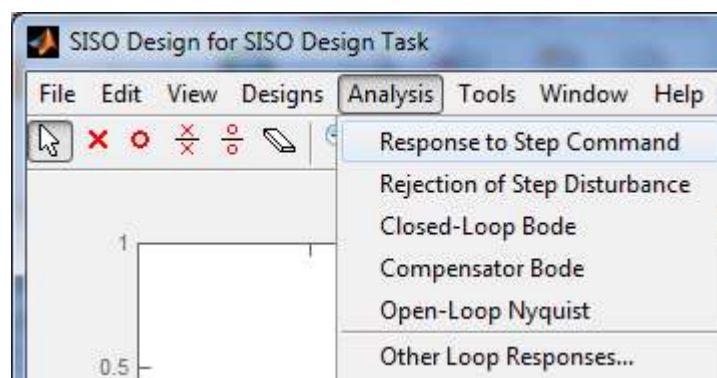


Fig. 12: Selection of the step response

The controller parameters can be changed by drag the red circles (zeros) and red filled squares (gain) as was written. There is still a big question, where is the optimal position for zeros and gain of the controller and what is the effect of the actual controller setting to the step response. Firstly, immediately after the zeros or gain position change the 'LTI Viewer' is actualized (real-time update). It means that you can drag with zeros and gain till you obtain e.g. the perfect step response. But better is to use some sophisticated approach to the whole tuning process. The common requirements to the controlled process are defined by maximal overshoot, settling time, rise time, etc. This tool serves you the possibility to restrict the whole real-complex axes system with lines, which correspond to these common requirements. The same possibility you have in the 'LTI Viewer' and in the 'SISO Design'. Restriction in 'SISO Design' are mainly use directly for the tuning purpose, the restriction in the 'LTI Viewer' are mainly use to verify the result in the time.

Let say that in our example the PI controller should be optimized for maximum overshoot $\%P < 10\%$ and a maximum settling time $T_s < 6\text{ s}$. The right click inside the white part of the graph opens the local menu where you should choose the 'Design Requirements' > 'New' (Fig. 14). The new window is opened. There you can choose the type of the requirement and the desired value for this parameter. The effect of the restriction is represented in the root locus area by a light yellow space. If the actual position is outside of this light yellow space the requirement is successfully fulfilled.

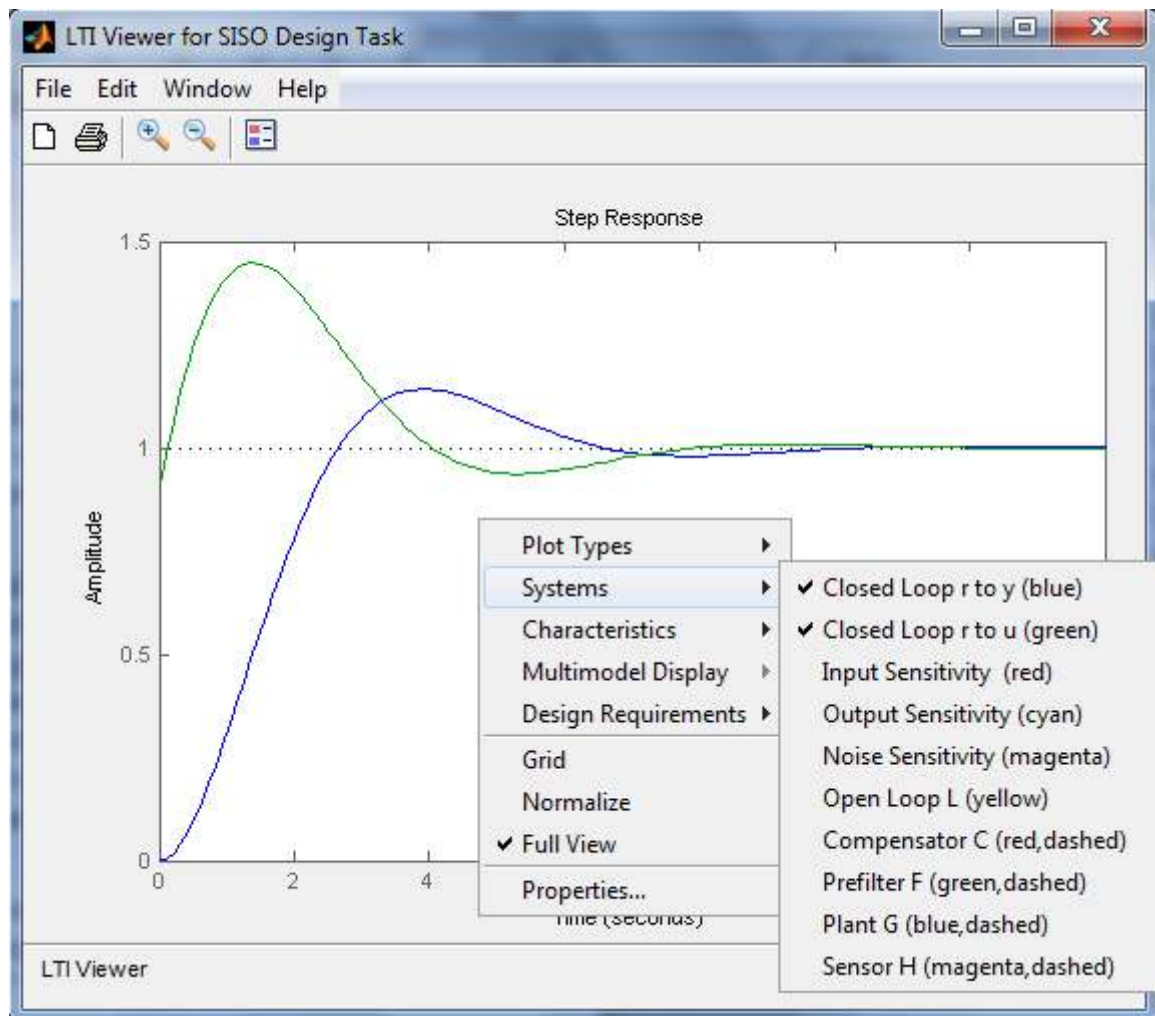


Fig. 13: LTI Viewer with step response

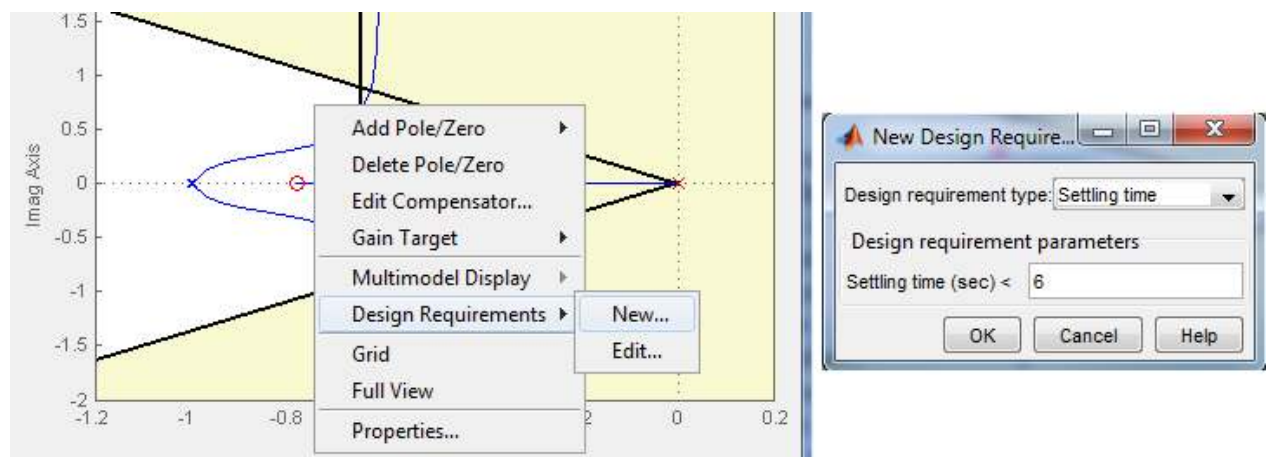


Fig. 14: Design requirements setting

The 'LTI Viewer' serves you the possibility to design requirements. Over that you can choose 'Characteristics' from the pull-down menu and the tool automatically find the values for the overshoot (peak response), settling time or rise time (Fig. 15).

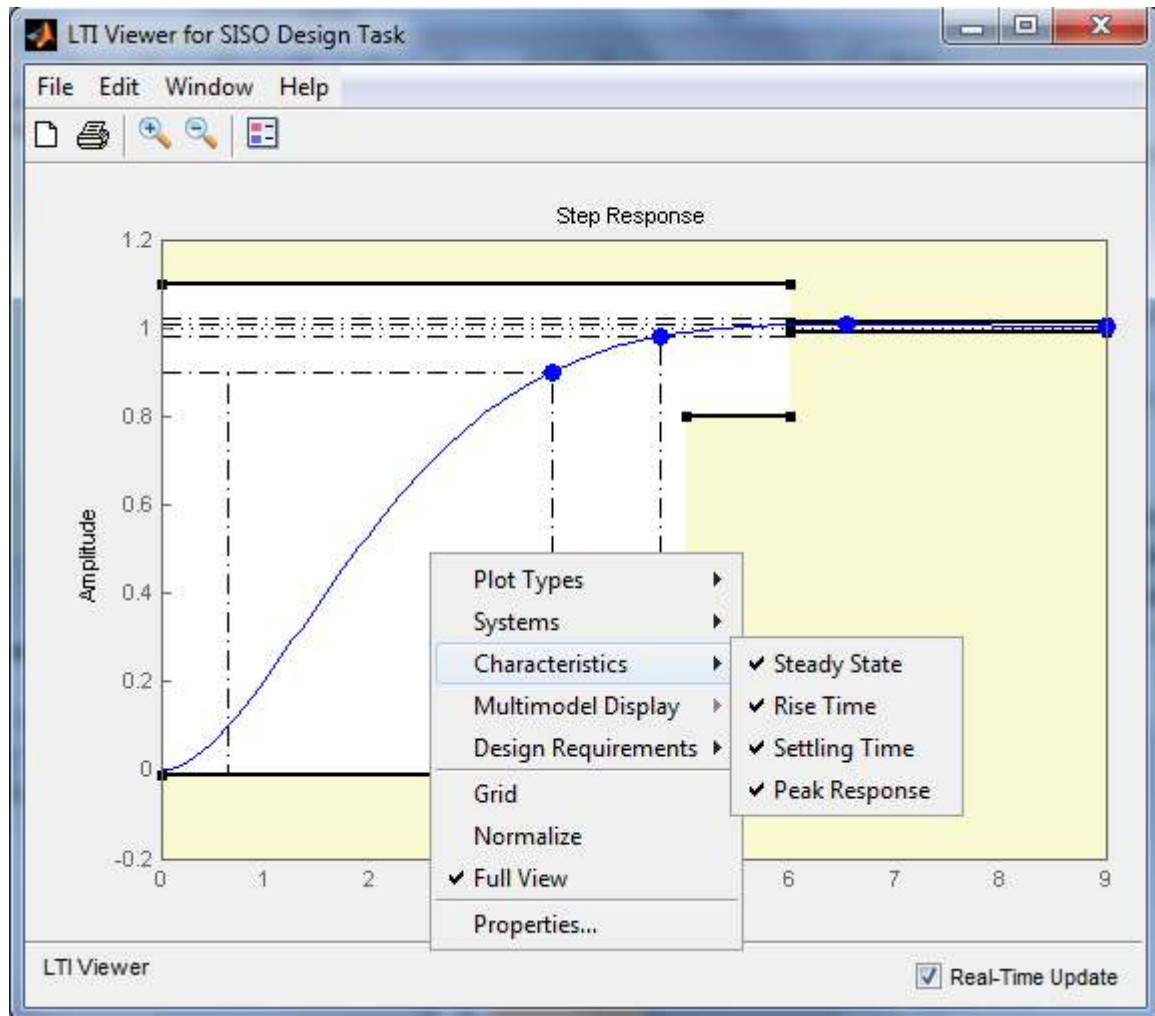


Fig. 15: 'LTI Viewer' menu with different 'Characteristics'. The restriction on design is presented.

The final task is 'only' shift with the position of zero and change the actual gain so the red filled squares (closed loop poles) are inside the white area. One of possible solutions is shown on the Fig. 16 where are 'SISO Design' with root locus and 'LTI Viewer' with step response. The actual controller parameters can be found in the main 'Control and Estimation Tools Manager' window under the tab 'Compensator Editor' (Fig. 16). If it is for the final application necessary, the controller should be rearranged to the desired form. In our example the optimal PI controller setting is $R(s) = 0.49 \cdot \frac{1+1.3s}{s} = 0.64 + \frac{0.49}{s} = 0.64 \cdot \left(1 + \frac{1}{1.3s}\right)$.

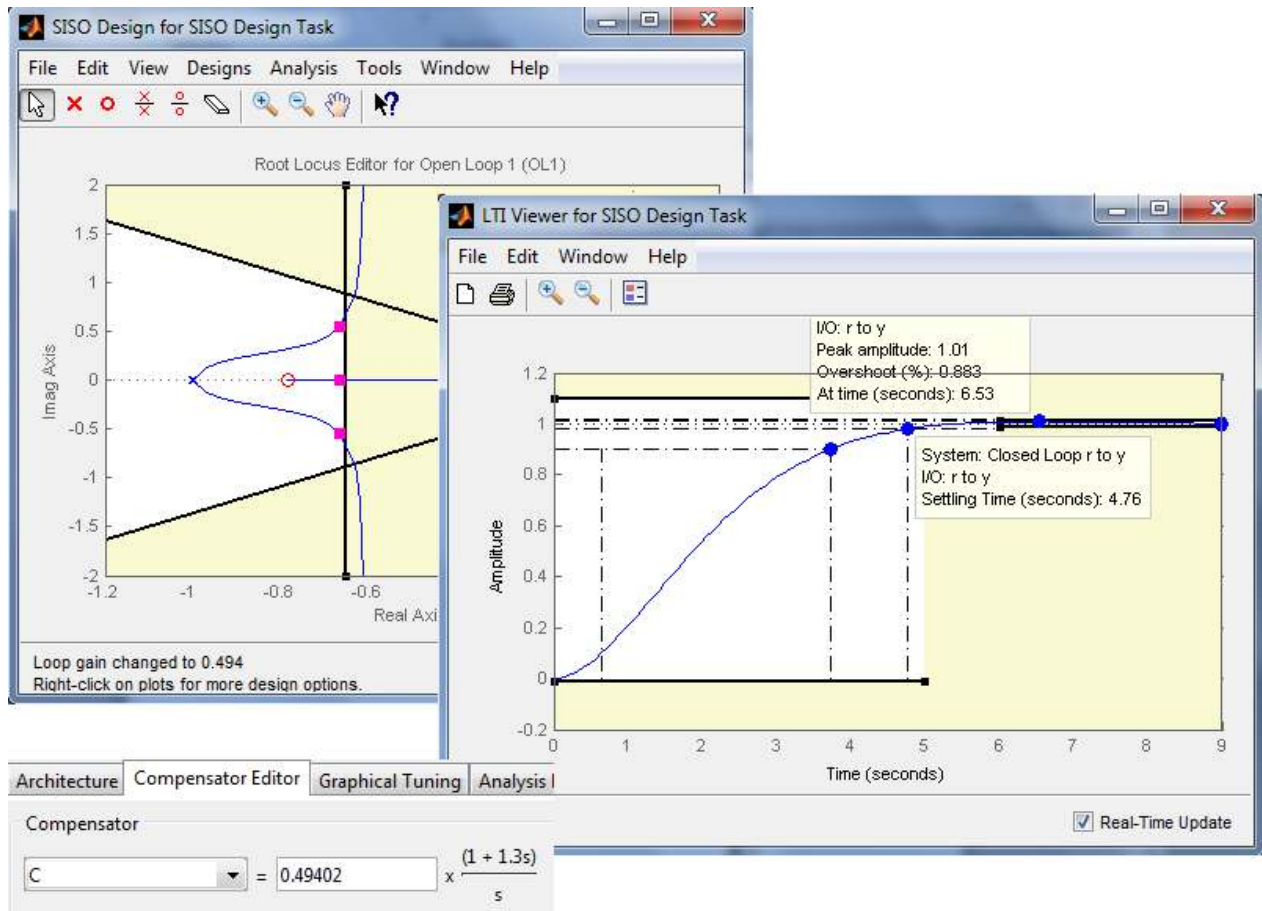


Fig. 16: Final controller tuning – root locus, step response, controller parameters

3.2.6 RL tuning example – PID controller

The system which has to be controlled by PID controller in this example has an oscillatory character and a transfer function $F_S(s) = \frac{0.6}{0.01s^3 + 0.3s^2 + 0.4s + 1}$. The poles are -28.7 ; $-0.6356 + 1.7541i$ and $-0.6356 - 1.7541i$. The PID controller consists of an integrator and two zeros. Since the poles of the system transfer function are complex, controller's complex zeros are generated. The right structure is in the 'SISOTOOL' generated by the similar way as in the previous example 'Add Pole / Zero' > 'Complex Zero'. The optimal strategy is to bring complex zeros of the controller to the neighborhood of the complex poles of the system transfer function and varies mainly the gain (Fig. 17). This method leads to the following parameters of PID controller $R(s) = 5.98 \cdot \frac{(1 + 0.38s + (0.5s)^2)}{s} = 2.27 + \frac{5.98}{s} + 1.5s$. The closed loop response has small overshoot (under 3%) and the settling time under 2.6 s. The real system will have a little different response, because the derivative part cannot be ideal in the real controller as in the model.

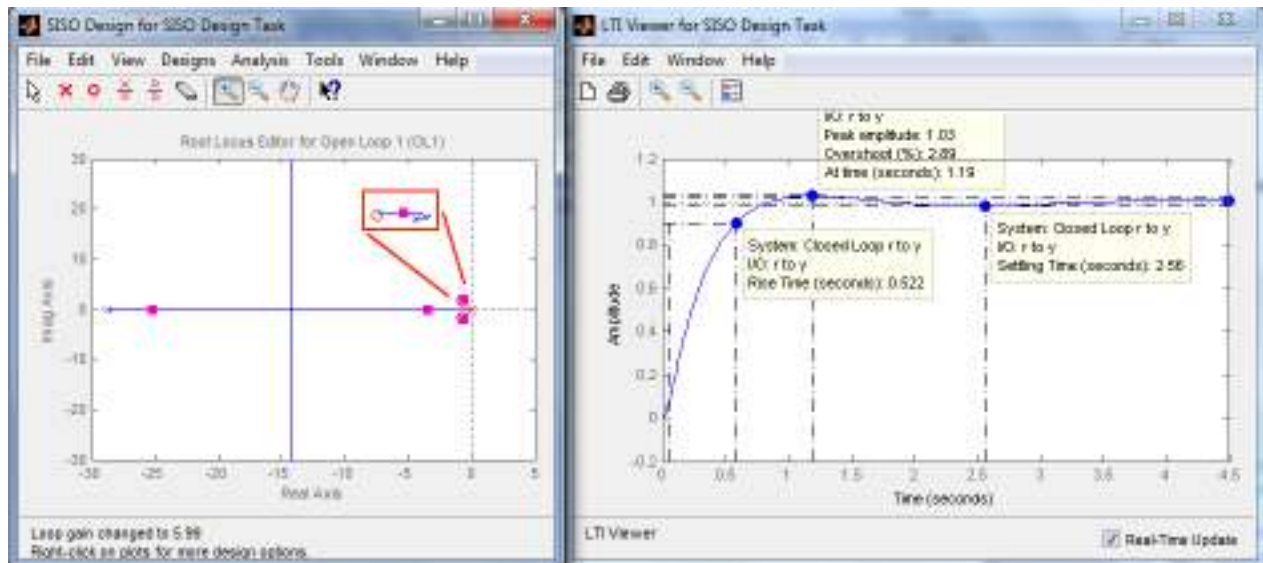


Fig. 17: Root locus with optimized PID controller (left) and step response (right)

4 EXPERIMENTAL PROCEDURE

- 1) According to the experimental objectives, the static and dynamic analysis should be performed. The static characteristic (steady state behavior) of the extended controlled system is measured and displayed. The coefficients for the selected structure of the transfer function of the controlled system are determined using the experimental identification.
- 2) The determination of a PID controller setting values. Comparison of behavior of the closed loop model with the real control loop.

4.1 DETERMINATION OF THE STRUCTURE OF THE LOOP TO BE MODELED

The decision, which part of the modeled system or where is the loop, must be realized before the measurement. It determines which signals have to be measured. A control loop is in accordance with DIN 19 221 for the following five major components (Fig. 18):

- control system,
- control element,
- measuring equipment,
- formation of management and task variable.

The signals in the loop are in line with the international designation from the DIN Standard 19 221 [12].

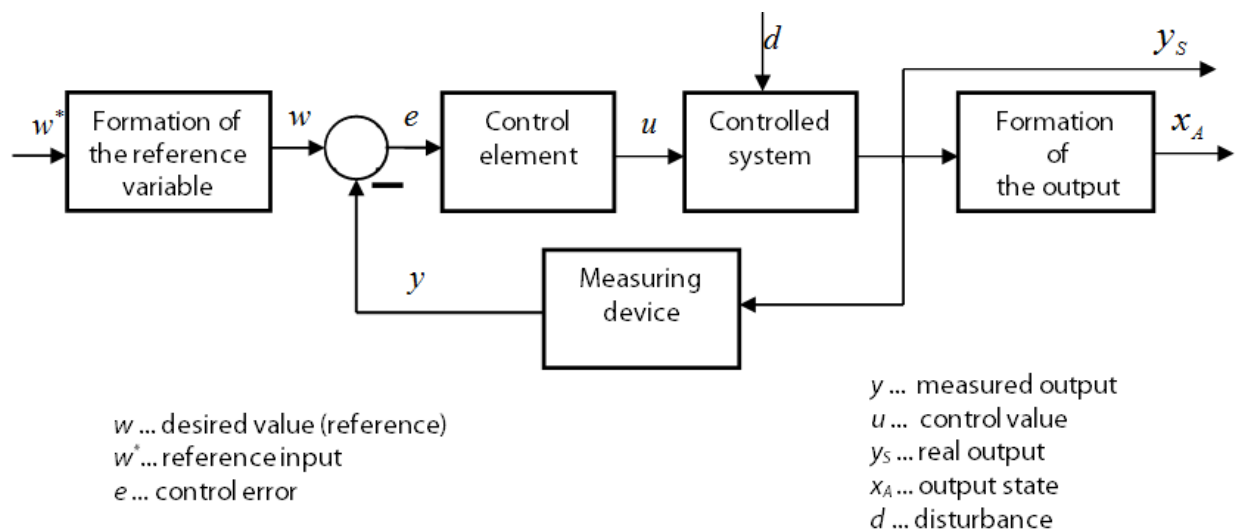


Fig. 18: Closed loop

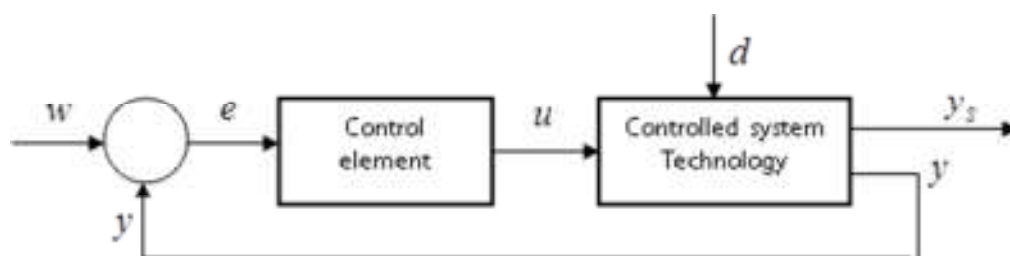


Fig. 19: Simplified block diagram of a control loop

The simplified block diagram of a control loop, which is realized with only two blocks, is shown in Fig. 19. Here, the controlled system, measuring device, transducer, actuator, etc., are combined [2] in the block 'Controlled system – Technology' which is often marked as 'Advanced control system'.

Task 1: Preparation and introduction

- Create the functional block diagram (like on Fig. 2) of a system in detail. Here the input and output signals are indicated. Generally, the transmission behavior of the individual transmission links through the transfer functions $F(s)$ or the transition functions $h(t)$ to be described.
- Determine the control system, advanced control system, measuring element, actuator, control variable, feedback variable and reference variable [11].
- Determine the range of the manipulated variable $u \in \{?\}$, and the range of the process output $y \in \{?\}$. Indicate the physical units.



4.2 DETERMINATION OF THE STEADY STATE BEHAVIOR

The determination of the steady state behavior (static characteristic) of the extended controlled system requires that at the input of the extended controlled system is applied a constant input signal and that the output signal has reached the steady state. The LabView program is prepared to realize this task. The input signal $u(t)$ is generated by a slider or can be manually entered in the textbox.

Task 2: Determination of the steady state behavior of the extended controlled system

Find the numerical and graphical representation of the steady state behavior – the relation between steady manipulated variable $u(\infty)$ and steady process output variable $y(\infty)$. Determine the linear range of advanced control system and choose the working point.

Measurement experiment description:

- Use prepared LabView program to fulfill this task.
- Manually change the actual value of the manipulated variable in the range $u = 0 \text{ V}$ to $u = 10 \text{ V}$ with maximal increment $\Delta u = 1.0 \text{ V}$.
- After you reach the maximal value on the manipulated variable continue with the measurement in opposite direction (descending).
- Always wait for the steady on the process output and fill the table with pairs of steady inputs, outputs!
- Determine the linear range of advanced control system.
- Choose a working point and region of operation.
- Try to discover the right physical values of input and output values in the static characteristic.

Discuss the results.



4.3 EXPERIMENTAL IDENTIFICATION

A mathematical model is used for simulations, system analysis, optimal control settings, etc. Such a model can be found on the basis of physical laws. Another way is the experimental identification. The experimental identification of the advanced control system is used for modeling in this case. The experimental identification can be divided into three steps:

- 1) Design and implementation of identification measurements.
- 2) Evaluations of the measurements with the aim to find a transfer function in the defined form, so called parameter estimation.

$$F(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}. \quad (7)$$

- 3) Verification of the computed mathematical model.

4.3.1 Identification measurement

The process of the identification measurement concludes from several step changes on the manipulated variable. The identification measurement has to be made directly on the system (no controller, open loop). The manipulated and output process values should stay inside the predefined linear area (working area) and cannot reach any limits. The standard way is to start with a manipulated variable jumping to the working point, holding the value till the output reach a steady state and then starting a sequence of several jumps around the working point. The new jump can be realized after the output value reaches the steady state. Finally, it is recommended to end with a jump back to the working point.

The functional characteristics of the manipulated variable jumps and the system responses are recorded in real time. Therefore, the LabView program has to be used. The input jumping sequence has to be generated by some Numeric Control block (Slide, Knob, etc.).

Task 3: Investigation of the real advanced control system – identification measurement

Create the LabView program to realize the identification measurement around selected working point. Crucial blocks are prepared in the sample file; you have to extend them to the functional one.

Use and hold these default parameters for the identification measurement:

- the sampling period $T = 0.05 \text{ s}$
- the measurement cycle time $T_Z \in \langle 10, 15 \rangle \text{ s}$
- the number of cycles $n_Z \geq 4$

The program main functions are:

- a) Realize the change of the manipulated variable to the working point.
- b) Realize the manipulated variable changes during the measurement to be able to carry out the whole identification measurement process.
- c) Use DAQ Navi Assistant (Fig. 20) to communicate with I/O card and timed loop (Fig. 21, Fig. 22) to hold real time.
- d) Automatically switch off the system after the measurement (set the output to 0).
- e) Use some graph to allow monitoring actual manipulating and process output variable values. Both lines/values have to be inside one graph.
- f) Ensure saving of the measurement into some text file.

- g) Print the graph of the measurement.

Measurement experiment description:

- Select a working point from steady state behavior and realize first jump into this point.
- Use a change of control variable $\Delta u = \pm 1 \text{ V}$ or $\Delta u = \pm 2 \text{ V}$ from the working point to design the manipulated variable course. Do not remember to wait till steady and hold T_z .
- Monitor $u(t)$, $y(t)$ during the measurement.

Discuss the result.

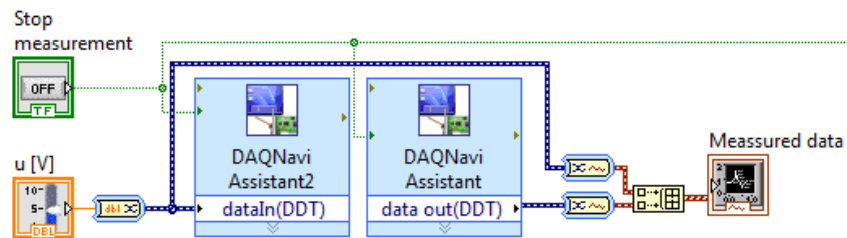


Fig. 20: Communication with I/O card

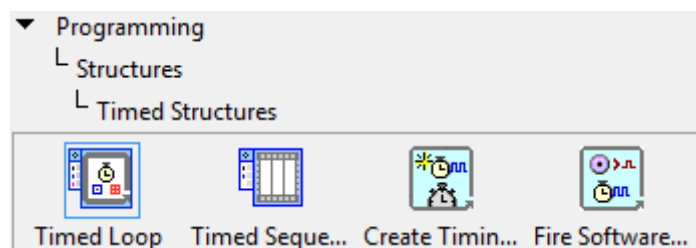


Fig. 21: Timed Loop structure

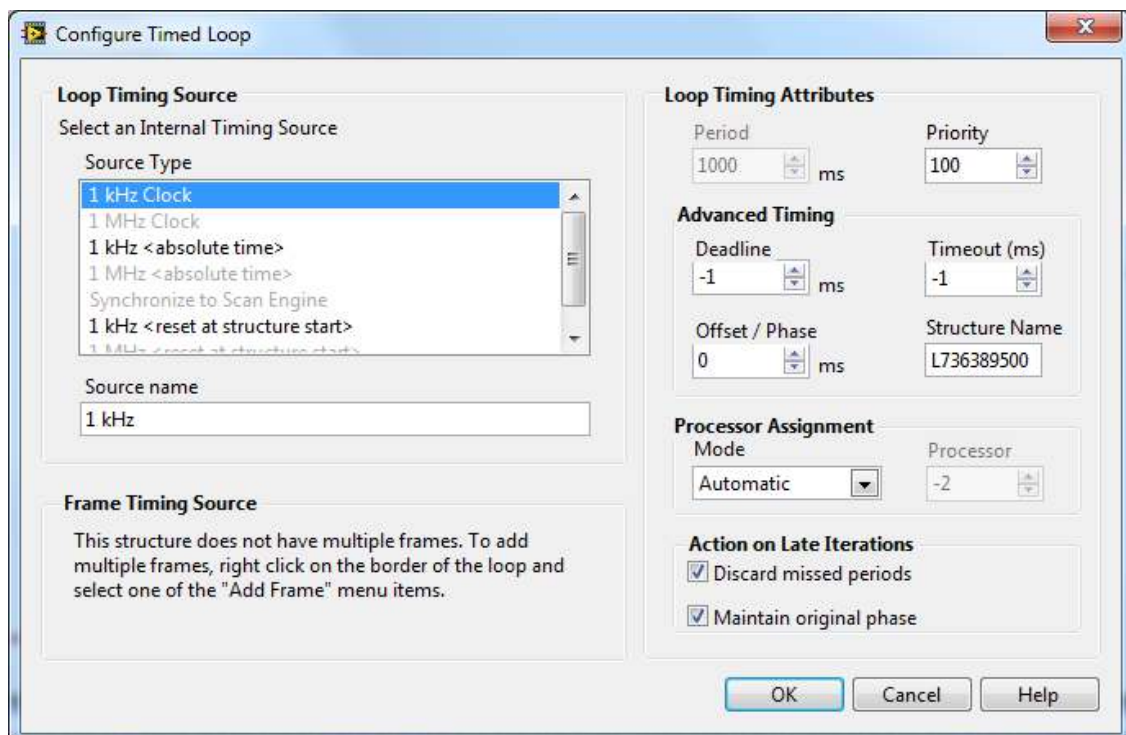


Fig. 22: Timed Loop Configuration window

4.3.2 Transfer function parameters estimation

The goal of the identification is to select the right transfer function structure and to determine the parameters of this transfer function from the dynamic behavior over the defined range. The transfer function will have, in our case, one of the following structures

$$F_3(s) = \frac{Y(s)}{U(s)} = \frac{K}{(T_1s + 1)(a_2s^2 + a_1s + 1)} \quad (8)$$

$$F_{4a}(s) = \frac{Y(s)}{U(s)} = \frac{K}{(a_4s^2 + a_3s + 1)(a_2s^2 + a_1s + 1)} \quad (9)$$

$$F_{4b}(s) = \frac{Y(s)}{U(s)} = \frac{K}{(T_1s + 1)(T_2s + 1)(a_2s^2 + a_1s + 1)} \quad (10)$$

where a_1, a_2, a_3, a_4 are general coefficients, T_1, T_2 are time constants and K is gain. Term $(a_2s^2 + a_1s + 1)$ and $(a_4s^2 + a_3s + 1)$ are describing the oscillating part/complex poles. This term can be also written as

$$a_2s^2 + a_1s + 1 = \left(\frac{s}{\omega_n}\right)^2 + 2\xi\frac{s}{\omega_n} + 1 \quad (11)$$

where ω_n is the natural frequency of oscillation, $\omega_n = \frac{1}{a_2}$ and $\xi \in (0,1)$ is the damping factor, $\xi = \frac{a_1}{2a_2}$.

Because the initial conditions of the transfer function must be zero by definition, the first step is to shift the measurement according to the working point. The measurement data are converted using the following formula

$$\begin{aligned} \Delta y(k) &= y(k) - y_0, \\ \Delta u(k) &= u(k) - u_0 \end{aligned} \quad (12)$$

where

- $y(k)$ is the measured process variable,
- $u(k)$ is the measured manipulated variable,
- y_0, u_0 are working points of the manipulated and process variable,
- $\Delta y(k), \Delta u(k)$ are data, which are entering into the transfer function parameters estimation process

Task 4: Determination of the transfer function $F(s)$

Build LabView program which allow you to find the transfer function of the system and find the right transfer function structure together with all parameters. Display the response of the optimal transfer function to the manipulated variable from your measurement and verity it with the measured process output variable.

The program main functions are:

- a) The first step (to the working point) in the measurement has to be cleared from the data (Fig. 23).
- b) Remaining measured points has to be shifted according to the selected working point via the equation (12).

- c) Use 'TF Model' block (Fig. 24) from Parametric Model Estimation toolkit to find the parameters of transfer function.
- d) Use 'Simulation' block (Fig. 25) from Model Validation toolkit to obtain the response of the transfer function for your signal.

Notes for the evaluation:

- a) The gain factor K can be determined directly from the steady states² of the process variable $y(kT)$ and the manipulated variable jumps $u(kT)$.
- b) Watch mainly the correspondence in the oscillation; the optimal transfer function should hold the same oscillation frequency/frequencies and the same damping.

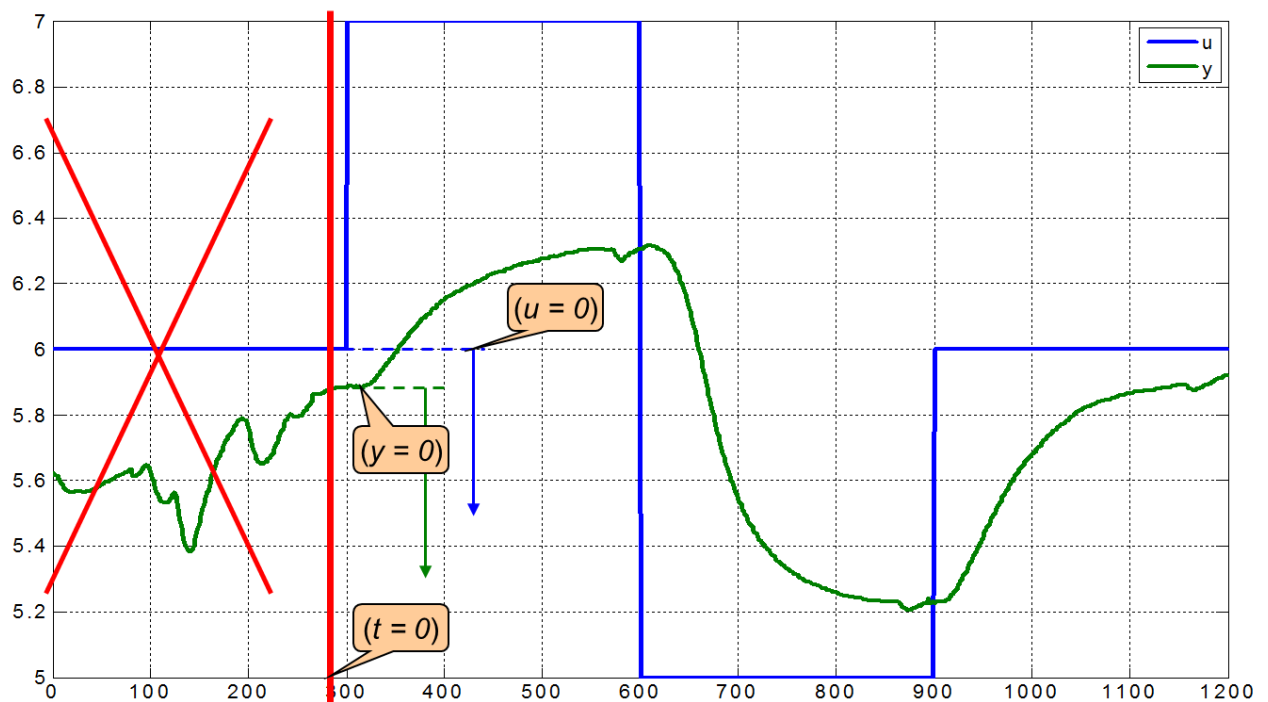


Fig. 23: General data preprocessing before parameter estimation

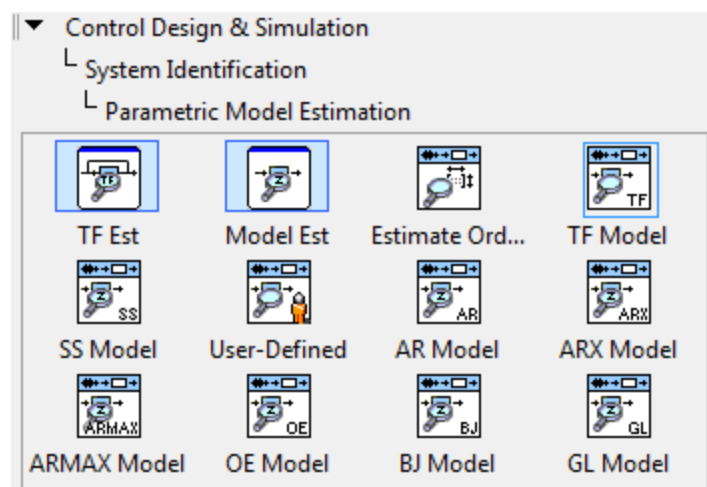


Fig. 24: Functions in LabView to find the system description

² The gain K can be also search with the optimization algorithm.

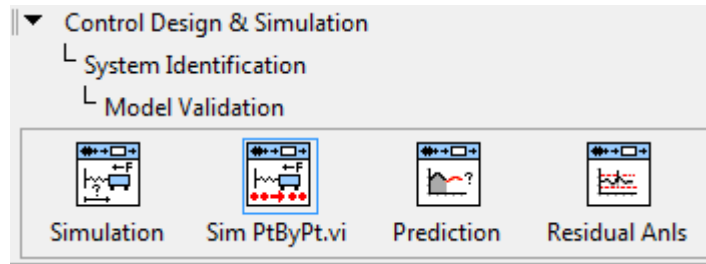


Fig. 25: Simulation block in the Model Validation

4.4 PID CONTROL

A PID controller can be described by the following equations

$$u_{par}(t) = K_R \left[e(t) + \frac{1}{T_n} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right], \quad (13)$$

$$u_{ac}(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}, \quad (14)$$

where $e(t) = w(t) - y(t)$, K_P is the proportional gain, K_R is the gain, $K_I = K_P/T_n$ is the integral coefficient in s^{-1} , $K_D = K_P T_D$ is the derivative coefficient in s , T_n is the integral time constant in s , T_D is the derivative time constant in s , $e(t)$ is the control error, $y(t)$ is the process variable, $w(t)$ is the reference and τ is the time parameter of integration. The equation (13) is describing the PID controller in a parallel form, the equation (14) in so called academic form.

To implement a PID controller in some software you require the algorithm to sample the input signals and discretize the integral and derivative action. The simplest form, also included inside PID block in LabView, is described by the following equations.

$$u_p(k) = P e(k)$$

$$u_i(k) = P \frac{\Delta T}{I} \frac{e(k) + e(k-1)}{2} + u_i(k-1) \quad (15)$$

$$u_D(k) = -P \frac{D}{\Delta T} (y(k) - y(k-1))$$

$$u_{real}(k) = u_p(k) + u_i(k) + u_D(k) \quad (16)$$

where k is the sample number at time $k \cdot t$, dT is the sampling time, $u_p(k)$ is the proportional action in the sample k , $u_i(k)$ is the integral action in the sample k and $u_D(k)$ is the derivative action in the sample k . The equation (16), see Fig. 26, is the discrete transformation of the equation (13).

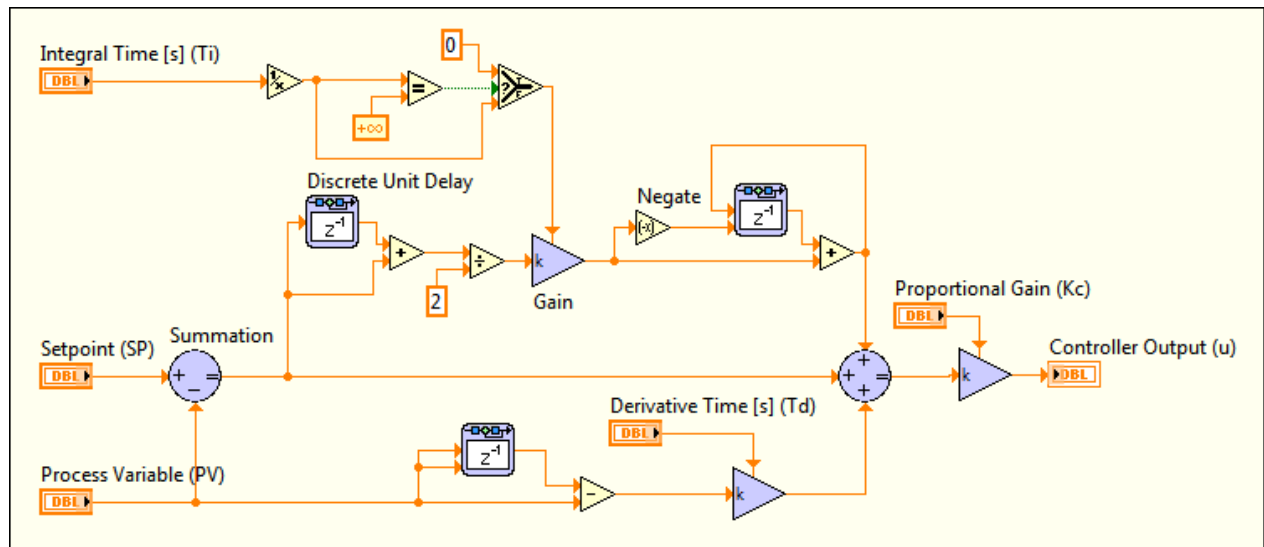


Fig. 26: Discrete realization of the PID controller

The simplest criterion of quality of the closed loop control is to assess the output signal from the closed loop $y(t)$ with the output from the forward line $y^M(t)$, whose input is the same step shaped signal – the reference (see Fig. 27). The signal $u^M(t)$ must be so large that the steady state value of $y^M(t)$ corresponds to $w(t)$. This is ensured through the 'Gain'. The gain is equal to the inversion of the system transfer function gain ($1/K$), so the whole forward line gain is equal to 1. To evaluate the effect of the controller, the step responses of the closed control loop and the straightforward control should be compared.

Another possibility is to specify some requirements in time to the closed loop step response and/or some requirements in frequencies to the closed loop frequency response. You can also want to parameterize some root locus properties. All these specifications are included inside the LabView. There are two options to make the design of the controller interactively. The first option is to use the block 'SISO Controller' from the Simulation library, the second option is to use the block 'CD Interactive Control Design' from the Control Design library. The differences are mainly in the implementation not in using (the GUIs of both block are the same).

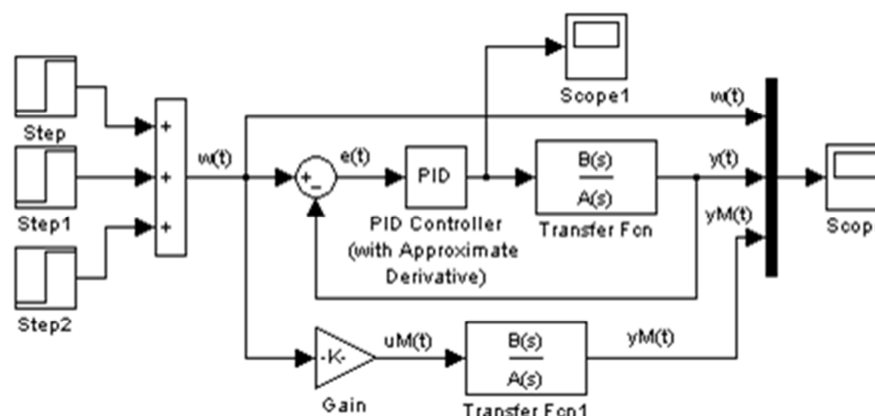
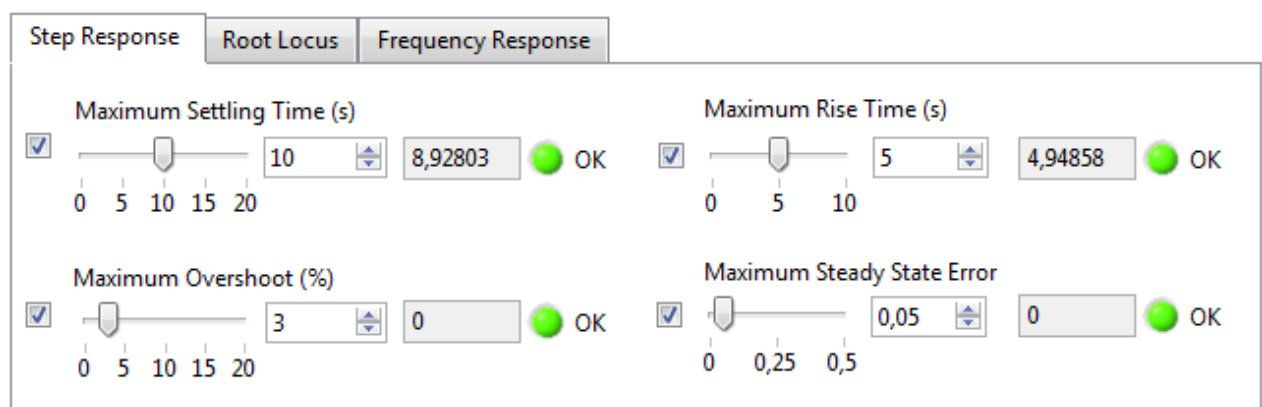
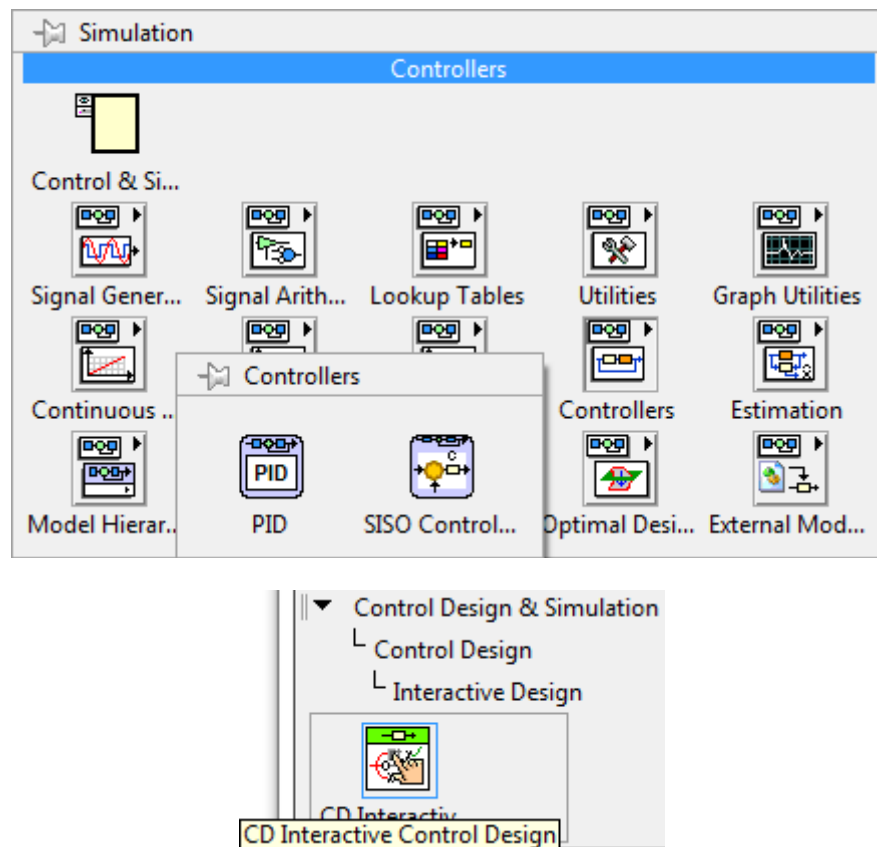


Fig. 27: Closed loop structure for controller setting



Task 5: Model control with PID controller in LabView – manual adjustment

The model of the real system is generally determined by the transfer function $F(s) = \frac{B(s)}{A(s)}$.

Construct a LabView program which allows you manually find suitable parameters of PID controller and monitor the response. Find the gain for which the output $y_M(t)$ is equal to the set point $w(t)$ in the steady state. **Write down the optimal PID controller's parameters which allow you to control the system with defined requirements.**

Closed loop step response requirements:

- The maximal settling time is 10 seconds.
- The maximal overshoot is under 3 %.
- The maximal rise time is below 5 seconds.
- The steady state error is under 0.05.

Notes for the evaluation:

- The function of the set point value $w(t)$ should be formed from the sequence of jumps or as a separate (unit) step.
- Describe the process and draw the results of $y(t)$, $w(t)$ in one graph.
- Do not forget to watch the manipulated variable through the control process. Draw it in a graph. For the real system it is necessary to add limits for the manipulated variable.
- Discuss the results.

**4.5 INVESTIGATION OF THE REAL CONTROL**

The real control of the process is realized in the LabView. The process is connected to the PC and the controller is realized inside the LabView. The communication is made through the I/O card and DAQ Navi Assistant in LabView.

Task 6: Control of the real plant from LabView

Examine the behavior of the closed loop with the real process with the determined controller setting from manual tuning. Use prepared LabView program to test the controller function. The functional profile of the reference value $w(t)$ should be formed by the sequence of jumps.

Notes for the evaluation:

- Compare results from real process control with the studies on the model. Compare both important variables: manipulated variable $u(t)$ and process output variable $y(t)$.
- Discuss results.

**4.6 ROOT LOCUS APPLICATION TO PID TUNING**

The Root Locus technique is one of the more sophisticated methods to tune PID controller. One from the main advantages is the immediately information about the stability and the allowed change on the gain to hold the stability. Matlab brings you useful tool to help with root locus controller design and tuning. The toolbox requires defining the PID controller in so called zero-pole-gain structure (Eq. (17) or Eq. (18)).

$$R_1(s) = k \frac{(s + s_{z1})(s + s_{z2})}{s} \quad (17)$$

$$R_2(s) = k \frac{s^2 + 2s\zeta\omega_0 + \omega_0^2}{s} \quad (18)$$

where k is the controller gain, $-s_{z1}$, $-s_{z2}$ are real zeros and $-\zeta\omega_0 \pm j\omega_0\sqrt{1-\zeta^2}$ is complex pair of zeros.

Task 7 (optional): Model control with PID controller in Matlab – root locus adjustment

The model of the real system is generally determined by the transfer function $F(s) = \frac{B(s)}{A(s)}$. **Use a Matlab program which allows you find suitable parameters of PID controller by the root locus. Monitor the response.**

Closed loop step response requirements:

- The maximal settling time is 10 seconds.
- The maximal overshoot is under 3 %.
- The maximal rise time is below 5 seconds.
- The steady state error is under 0.05.

Notes for the evaluation:

- a) Construct the structure of PID controller inside the Matlab toolbox.
- b) Find parameters of PID controller and write them for the parallel or academic form of PID controller. Change the positions of zeros and the value of gain to reach the desired behavior.
- c) Describe the process and draw the results of $y(t)$, $w(t)$ in one graph.
- d) Do not forget to watch the manipulated variable through the control process. Draw it in a graph. For the real system it is necessary to add limits for the manipulated variable.
- e) Discuss the results.

**Task 8 (optional): Control of the real plant from LabView/Matlab**

Examine the behavior of the closed loop with the real process with the determined controller setting from root locus. Use prepared LabView/Matlab program to test the controller function. The functional profile of the reference value $w(t)$ should be formed by the sequence of jumps.

Notes for the evaluation:

- a) Compare results from real process control with the studies on the model. If some significant differences are presented, describe them and discuss their causes!



5 LITERATURE

- [1] FÖLLINGER, O.: Regelungstechnik. Heidelberg, Hüthig, 1992, 7. Auflage.
- [2] UNBEHAUEN, H.: Regelungstechnik I. Vieweg & Sohn mbH, Braunschweig/Wiesbaden, 1992.
- [3] REINISCH, K.: Kybernetische Grundlagen und Beschreibung kontinuierlicher Systeme. VEB Verlag Technik, Berlin, 1974.
- [4] GRACE, A.; LAUB, J.A.; LITTLE, J.N.; THOMPSON, C.M.: Control System Toolbox. For Use with Matlab. User's Guide. The Math Works, Inc. 1995.
- [5] BODE, H.: Matlab in der Regelungstechnik. Verlag Teubner, Stuttgart, Leipzig 1998.
- [6] HOFFMANN, J.: Matlab und Simulink. Beispielorientierte Einführung in die Simulation dynamischer Systeme. Addison-Wesley Longman Verlag, 1998.
- [8] HAMPEL, R.: Begleitmaterial zur Vorlesung Prozeßautomatisierung, Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz, FB Elektrotechnik, FG Meßtechnik/ Prozeßautomatisierungstechnik, 1997.
- [9] GOCHT, U.; HAMPEL, R.: Arbeitsmaterialien für Vorlesung Prozeßautomatisierung, Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz, FB Elektrotechnik, FG Meßtechnik/Prozeßautomatisierungstechnik, 1998.
- [10] NELDER, J. A.; MEAD, R.: A Simplex Method for Function Minimization. Computer Journal, vol. 7, p. 308-313.
- [11] Deutsche Norm DIN 19221 Regelungstechnik und Steuerungstechnik, Formelzeichen DK 62-52/-53: 003.62 Mai 1993.

APPENDIX A – STEADY STATE MEASUREMENT VI

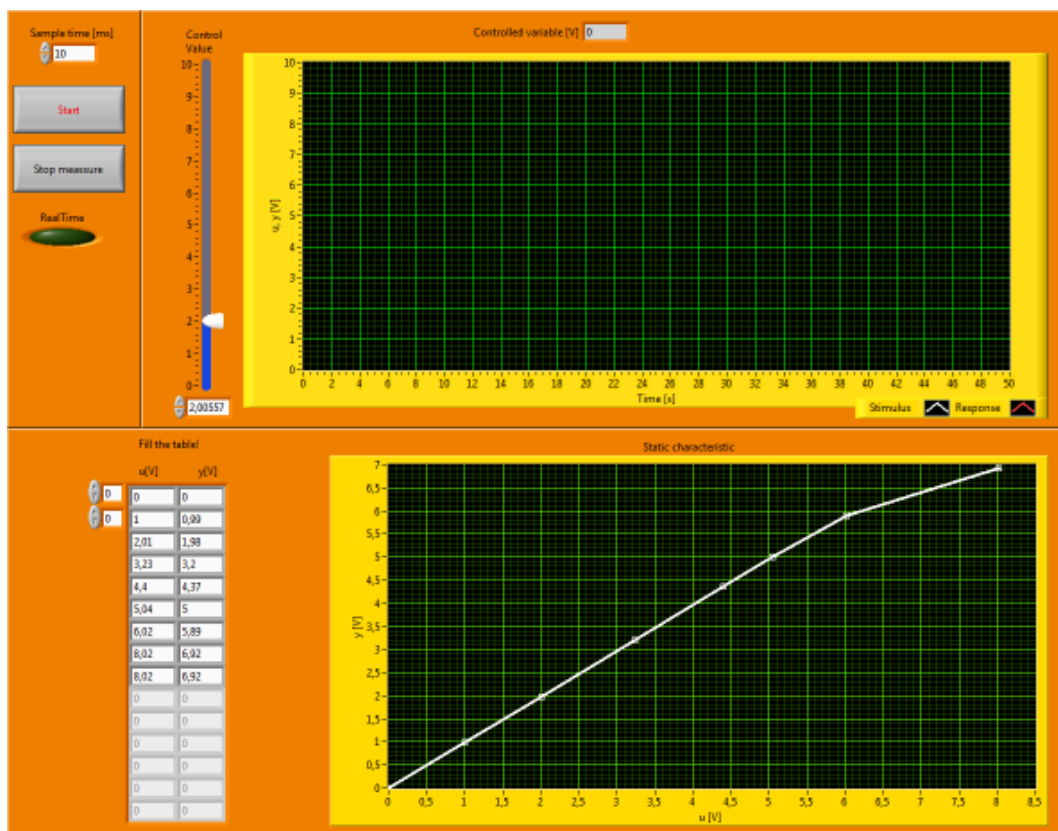


Fig. 28: Front Panel of steady state measurement

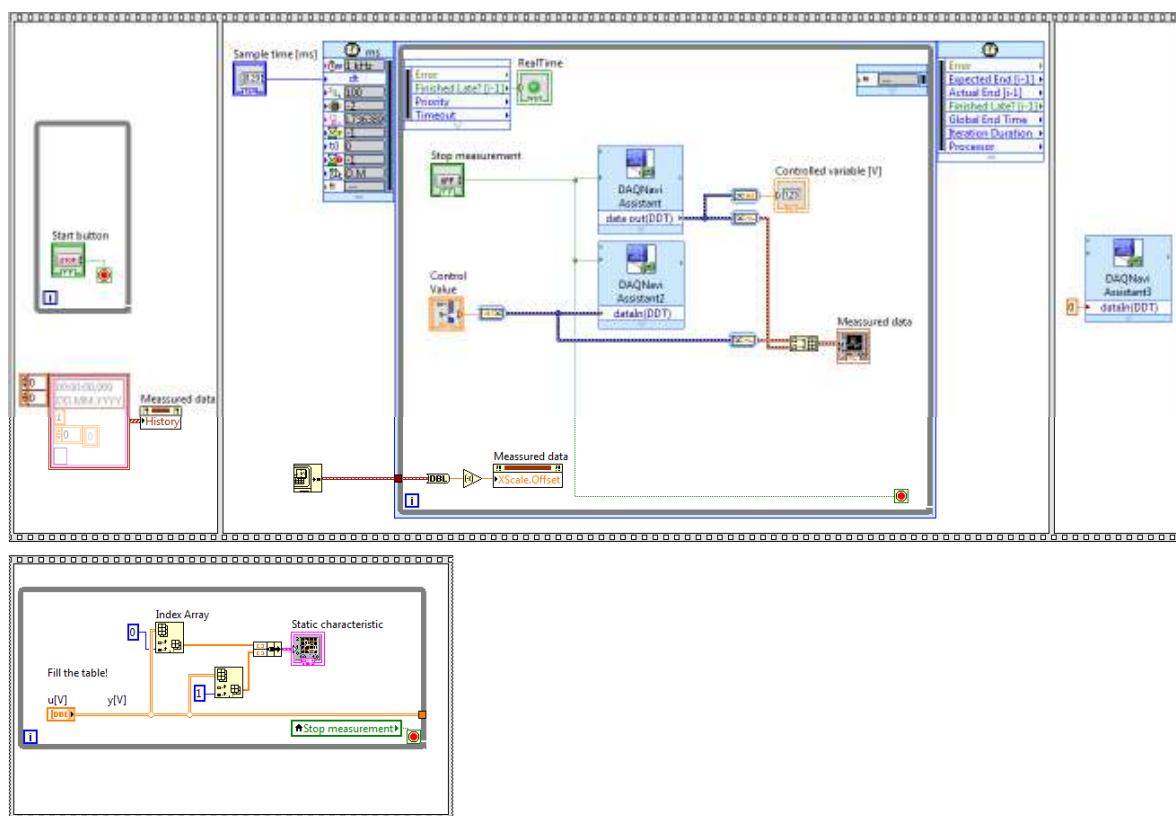


Fig. 29: Block Diagram of steady state measurement