FRTN15 Predictive Control
Laboratory experiment 1
Auto-tuning and Gain Scheduling

Department of Automatic Control
Lund Institute of Technology

1. Introduction

In this laboratory experiment the topics auto-tuning and gain scheduling are studied. Both these facilities are included in the commercial PID controller ECA600 from ABB Satt AB. Instead of a physical plant the MATLAB/Simulink environment extended with real-time facilities is used to simulate a number of plants on a PC running the Linux operating system. The PC is also used for plotting and for analysis of the behavior of the closed-loop system.

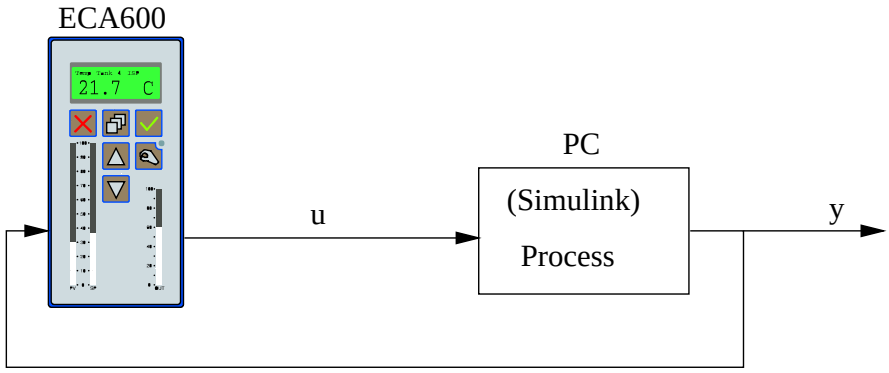

Figure 1 The laboratory setup.

1.1 Login

Log onto the lab machines with your student account. Then, download the files needed for this lab from the course homepage.

1.2 Preparation

Study the instructions for the laboratory experiment given here. Additional reading is found in chapters 8 and 9 of [Åström and Wittenmark(2008)] and chapter 7 of [Khalil(2002)].

In the next section short descriptions of the simulation software and the ECA600 controller are given. To get you familiar with the ECA600 a demonstration will be given at the start of the laboratory experiment. It is advisable to spend a few minutes at the start of the lab to go through the section on how to operate the controller.

1.3 The processes

The MATLAB/Simulink environment will be used to simulate two different processes.

Process 1: A second order system with both poles at $s = -0.5$.

$$G_1(s) = \frac{1}{(1+2s)^2}$$

Process 2: The same as Process 1 but with a time delay of 4 seconds.

$$G_2(s) = \frac{1}{(1+2s)^2} e^{-4s}$$

1.4 Design evaluation

To evaluate the control performance of the closed-loop system the following measures will be computed.

T_s Solution time. The time it takes for the process output to get and stay within 5% from the set-point after a unit set-point change.

IAE_{step} The integrated absolute error after a unit set-point change

$$IAE_{step} = \int_0^\infty |r(t) - y(t)| dt$$

where $r(t)$ is the set-point value. IAE is the total area between the process output and the set-point value. A small value indicates that the step response is fast and well damped. For non unit steps IAE is obtained by normalizing the integral with respect to the step amplitude.

IAE_{load} The integrated absolute error after a unit load disturbance step. Same definition as above except that the exciting signal is a unit load disturbance step.

2. The ECA600 controller

In **ECA600** a serial form PID-controller is used

$$G_{PID} = K\left(1 + \frac{1}{sT_I}\right)\left(1 + \frac{sT_D}{1 + sT_d/N}\right)$$

This parameterization differs from the normal text book parallel form and therefore the PID-parameters of the two forms are not equivalent. It is, however, not difficult to get expressions for transforming between the two forms. In the ECA600 controller, the derivative part is low pass filtered using $N = 8$.

The controller parameters K, T_I and T_D can be tuned manually or automatically. The automatic tuning is performed using a relay feedback experiment. Gain scheduling is obtained by using a number of PID parameter sets. Each of these sets can be tuned automatically. The currently used set of parameters is then governed by the value of a scheduling variable chosen by the user.

In many industrial applications a smooth response without overshoot is preferred to a fast response with less damping. ECA600 takes this into account in the controller design. Because of this the auto-tuning will in some cases give a closed-loop system with a well damped but rather slow response.

2.1 Using the Controller

All controls and indicators are located on the front panels of the controller as shown in Figure 2.

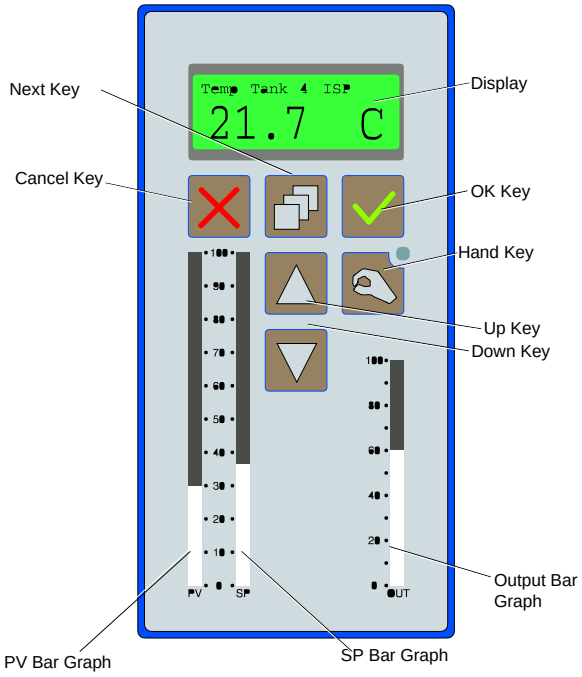

Figure 2 Front Panel Controls and Indicators.

2.2 Display

In operator mode the display shows two lines of text. The bottom row always shows the value of the currently selected signal (the process value as default PV). The top line of the display summarizes the present status of the controller or shows the name of the parameter currently presented. You can also display the following status parameters:

- Internal set-point value (ISP).
- External set-point value (ESP).
- Output signal value (OUT).
- Internal/External set-point switch status (ISP/ESP).
- Information messages.

2.3 Control keys

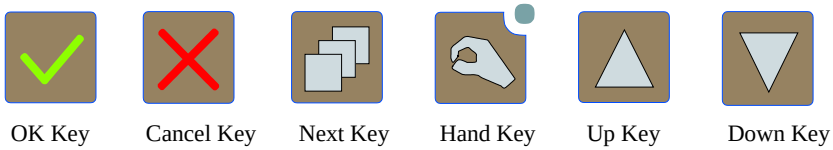

Figure 3 Control Keys.

OK key Press the OK key to descend to the next lower level in the tree structure, to save a change in a parameter, or to follow instructions on the display. In operator mode this key is primarily used to start the autotuner and to acknowledge errors and warnings.

Cancel key Press the Cancel key to ascend to next higher level in the tree structure without making any changes to the displayed parameter. In operator mode this key is primarily used to force the display to revert to the showing of default value.

Next key In operator mode press the Next key to display the value of the next parameter.

To enter the configuration menu (e.g. to change the PID parameters), press the Next Key long.

Up and Down keys In operator mode press the Up/Down key to increase/decrease the numerical value of the selected parameter. If you press and hold the Up or Down key the speed of change of the displayed value/selection increases with time.

Hand key Press the Hand key to toggle between auto and manual mode. A red light-emitting diode adjacent to the key is lit when the controller is in manual mode. In manual mode, the control signal itself is set by the operator, i.e. no automatic control is involved. The PID controller is only active in auto mode. Ensure that you are running in auto mode when you want to use the stored controller!

2.4 Operator mode

In operator mode the unit is used to control a process. In operator mode you can:

- Read the process value (PV).
- Change the set-point (SP).
- Switch between manual and automatic control.
- Change the output signal (OUT).
- Toggle between an internal set-point (ISP) and an external set point (ESP).
- Auto-tune.

2.5 Manual control of the output signal

1. Set the unit to operate in manual mode (see Hand key).
2. Press the Up/Down key to increase/decrease the output signal

2.6 Selecting internal/external set-point (ISP/ESP)

1. Press the Next key long to switch to the configuration menu until the top line of the display shows Set Point Switch. The currently selected mode is displayed on the lower part of the display.
2. Press the Up key or the Down key to toggle selection.
3. When ISP is displayed, the controller uses it’s own setpoint. When ESP is displayed, the controller uses a setpoint from an external source. This could for example be the output of another controller, as in a cascade arrangement. In this lab, the external setpoint comes from Simulink.

2.7 Change the set-point (SP)

1. Check the top line of the display to see if ISP or ESP is selected. If ESP is selected, select ISP before proceeding otherwise pressing the Up/Down keys will have no effect.
2. Press the Next key until the ISP value is displayed.
3. Press the Up/Down key to increase/decrease value as required. (The output will be displayed. Wait until it settles, and continue altering the ISP until it reaches its desired steady state value.)

2.8 Auto-tuning

The autotune function must be enabled under the configuration menu under the menu

Control → Parameters → Tuner → Mode

When the autotuner is set to "On", it can be manually initiated in the operation mode.

1. Select internal set point (ISP) for the controller.
2. Put the simulation into run mode.
3. Put the unit in manual mode.
4. Choose an internal set-point (e.g. 30 %).
5. Start the simulation.
6. Manually change the controller output using the up/down keys, until process variable (PV) and set-point are equal.
7. Press the Next key several times until the display shows "To Tune $\surd$"
8. After pressing the OK key, the unit will start its tuning program and switch to auto mode.

Notes

- Before activating tuning, ensure that the difference between the set-point and the process value is small, otherwise the tuning may fail.
- The tuning process can be interrupted by pressing the Hand key. In this case no changes are made to the controller's parameters.
- If you change process you must reset the autotuner under the menu Control → Parameters → Tuner → Reset
- Dead-time design. This is a special PID controller design where the autotuner adapts the gain and integration time for dead-time processes. You can change to this under the menu

  Control → Parameters → Tuner → ControllerDynamics
  The default setting is normal.

2.9 Changing PID parameters

There are three sets of PID parameters. The three sets are identical unless gain scheduling is enabled. The PID-parameters can be set manually under the menu

Control → Parameters → PID Parameters

2.10 Gain scheduling

When gain scheduling is enabled the controller uses different PID parameters according to the value of a user selected reference signal, *GainSchedulingSignal*. Three different settings for PID-parameters can be used, within three ranges defined by the user defined parameters *Limit1* and *Limit2*. When the *GainSchedulingSignal* passes a limit, the controller switches to a new set of PID parameters.

Range	Scheduling variable (x)	Parameter set	K	T_I	T_D
Lower	$0\% < x <$ *Limit1*	PID1	K_1	T_{I1}	T_{D1}
Middle	*Limit1*$< x <$*Limit2*	PID2	K_2	T_{I2}	T_{D2}
Upper	*Limit2*$< x < 100\%$	PID3	K_3	T_{I3}	T_{D3}

The three sets of PID-parameters can be inserted into the parameter table manually or via autotuning.

Under

Control → Parameters → Gain Scheduling

you find menus for enabling gain scheduling and for setting the above gain scheduling parameters. You also have to choose a gain scheduling signal. The signal *AI3[Value]* is the set-point signal and the signal *OC1[ControlSignal]* is the control signal.

3. Experimental setup

The lab assistent will have set up the wiring and software of the ECA600 prior to the lab. If you experience unforseen behavior, don't hesitate to ask the assistent to verify the setup.

3.1 Bias

There can be bias on signals between the controller and the computer. Therefore you have to calibrate the system prior to conducting experiments.

- The ECA600 has to be in manual mode with an internal setpoint.
- The simulation has to be set into run mode.
- Start the simulation.
- Set the internal setpoint of the controller to 50%.
- Manually change the output of the controller so that the Process variable (PV), i.e. the output of the process to be controlled, is also at 50% on the screen of the ECA600 controller.
- Double-click the "Set Bias" box to calibrate.

Don't hesitate asking the lab assistent if you are unsure how to do it.

4. The Matlab/Simulink Environment

Instead of a physical plant, Matlab/Simulink with real-time enhancements is used to simulate a number of plants.

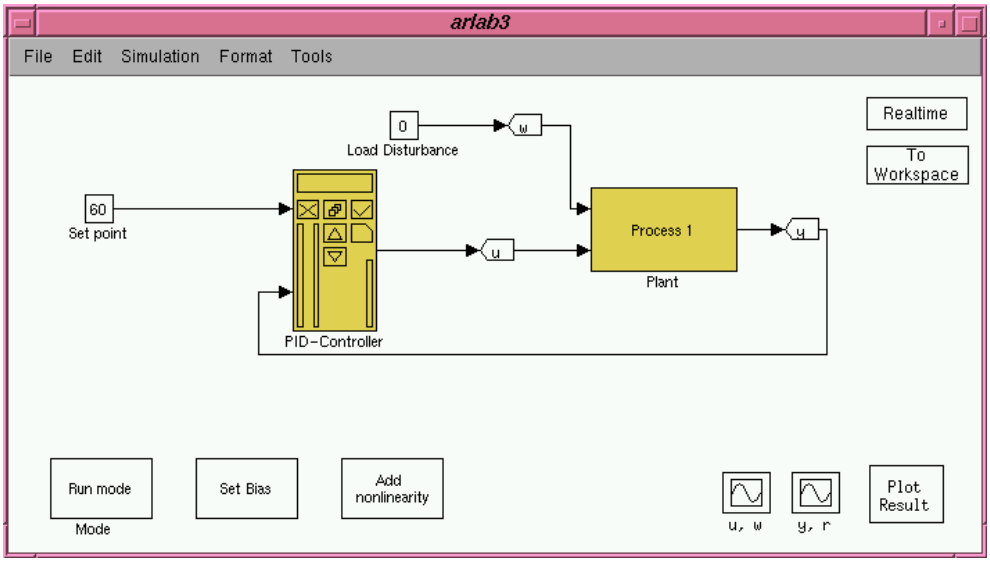

Figure 4 Simulink model `arlab3.mdl`.

Start Matlab by typing the command `VERSION=R2008a matlab` into a terminal window. Navigate to the folder that you saved the lab files in. The Simulink model used in this lab is then the following:

4.1 Simulink model `arlab3.mdl`

The Simulink model in Figure 4 implements a number of plants and a discrete-time PID controller. The model is opened from the Matlab command prompt by entering the model name (without the file extension):

```
>> arlab3
```

The following changes can be made in the block diagram:

- Double-click the plant block to switch between Process 1 and Process 2.
- Double-click the PID controller block to switch between the ECA-controller and the internal controller Reg.
- Double-click the green **Add nonlinearity** / **Remove nonlinearity** box to add/remove a nonlinear function at the process input.

Before you start the lab, make sure that Process 1 is selected and that you have no nonlinearity. Also make sure you have removed the bias as described above in section 3.1 before you start.

Choose execution mode Double click the mode box to switch between run and eval mode.

Run mode Normal simulation.

Eval mode Evaluate the closed-loop response to a set-point step followed by a load disturbance step. Observe that the ECA600 must be set to external set-point (ESP). The set-point step and the load disturbance step are applied when a steady state process output has been obtained. During evaluation IAE is computed both for the response of the set-point step and the load disturbance step. The solution time T_S as well as IAE_{step} and IAE_{load} are displayed in the result plot.

Start/Stop simulation Choose `Simulation | Start` (or press ctrl-t) to start and stop the execution. In eval mode the simulation stops when a steady state process output has been obtained after the load distrubance step. Some signals can be viewed in realtime during execution. Double click the grey box with the corresponding signal label.

Changing set-point Double click the grey Set-point box and enter the desired set-point value.

Apply load disturbance The load disturbance is best applied during simulation, when steady state has been obtained.

Double click the grey Load Disturbance box and enter the value of the load disturbance.

Plot results Click the grey Plot Result box. A figure window with the results of the last controller execution simulation is opened. Plot legends can be turned on and off with the `View | Legends` menu command. It is a good idea to keep old plot windows for later comparisons.

4.2 A note on real-time performance

The ECA600-Controller is sampled periodically and the control signal is applied to the process immediately with no time delay after each sampling instant. These are ideal *timing constraints*. In practice this is hard to achieve. Failure to meet the real-time constraints may lead to severe performance deterioration. In the Matlab/Simulink environment the Graphical User Interface (GUI) under some circumstances consumes so much computational power that the timing constraints are violated. This often is the case the first few seconds of a realtime simulation, or if the user invokes operations such as moving, opening or closing windows in the GUI. The Simulink models includes a measure of the timing constraints violations in the *jitter* signal. When the jitter is close to zero the timing is close to ideal. When the jitter is equal to one there is a delay of one sampling period at the sample instants. Be careful with working with the GUI when a simulation is running. Use keyboard shortcuts instead of menu commands whenever possible.

5. Manual tuning

Exercise 1 Find a good controller for the process G_1 by manually tuning the PID-parameters. For a set-point step the closed-loop system should be well damped and not too slow. A practical way to do this is to start by tuning a P or PD-controller using Ziegler-Nichols closed-loop method. Then include the integral part by decreasing T_I until the damping starts to decrease. Small changes can now be made in K and T_D to counteract the loss in damping due to the integral part.

In a practical situation the response to load disturbances are important. Both the set-point step response and the load disturbance response can be evaluated when running the simulation in **Eval** mode. Write down the obtained controller parameters as well as the result obtained from evaluating the design. Don't forget to use the external set-point (ESP) in ECA600 together with **Eval**.

Process	K	T_i	T_d	T_S	IAE_{step}	IAE_{load}
$G_1(s)$						

6. Auto-tuning

Exercise 2 Use auto-tuning to find a controller for G_1. Check the reproducibility of the scheme by tuning more than once. Write down the obtained controller parameters as well as the result from the design evaluation.

Process	K	T_i	T_d	T_S	IAE_{step}	IAE_{load}
$G_1(s)$						

Exercise 3 The auto-tuner might not find the best PID controller for the process. Try to find a better controller for G_1 than that obtained by auto-tuning. A natural way to do this is to use the tuned parameters as the starting point.

Process	K	T_i	T_d	T_S	IAE_{step}	IAE_{load}
$G_1(s)$						

Exercise 4 Now use auto-tuning to find a controller for the second process G_2. When ECA600 is used on a new process the auto-tuner should be reset (see Operator's manual). The reason for this is that ECA600 at the first few tunings collects and stores information about noise level and appropriate relay amplitude. This information is used to make the tunings that follow in a shorter time period.

Process	K	T_i	T_d	T_S	IAE_{step}	IAE_{load}
$G_2(s)$						

Exercise 5 We know that G_2 has a time delay. To improve the performance for processes known to have a time delay the auto-tune dynamics can be set to *dead time*. When changing the auto-tune dynamics it is not necessary to perform a new auto-tuning provided one has been made earlier. Instead the controller parameters will be automatically changed in accordance with the new auto-tune dynamics. Choose *dead time* and do a new evaluation of the performance.

Process	K	T_i	T_d	T_S	IAE_{step}	IAE_{load}
$G_2(s)$						

Exercise 6 An internal PID controller has been designed for process G_2. Using the process model, the PID parameters have been calculated to minimize the influence from the load disturbance. Evaluate the internal PID controller Reg designed for $G_2(s)$ and compare with the response of the auto-tuned closed-loop system.

Process	T_S	IAE_{step}	IAE_{load}
$G_2(s)$			

7. Gain scheduling

For processes with important nonlinearities and process variations it may be difficult to achieve satisfactory performance with a linear controller. In many cases it is known how the process dynamics change with the operating conditions. It may then be possible to apply a technique called gain scheduling. The idea is to change the controller parameters in a predetermined way according to an auxiliary variable, closely correlated to the change in the process behavior. If applicable, this is often a good way to achieve good performance with a simple controller.

To investigate the possibilities with gain scheduling a nonlinearity, shown in Figure 5, is placed at the process input. This can for instance correspond to a nonlinear valve. In Simulink use the switch **Add nonlinearity** to add the nonlinearity.

First the behavior of the closed-loop system for the process G_2 with a nonlinearity is studied without using gain scheduling.

Exercise 7 As before use the *dead time* auto-tune dynamics to get the best performance. Tune the controller automatically at the process value 10%. Look at step responses and load disturbances around the process values 10% and 70%. Observe that you shouldn't use **Eval** mode in this exercise. Make a new tuning at the process value 70%. Again look at the behavior around the same levels. Write down your observations.

Auto-tuning at PV=10%

Stationary point (PV)	Behavior
10%	
70%	

Auto-tuning at PV=70%

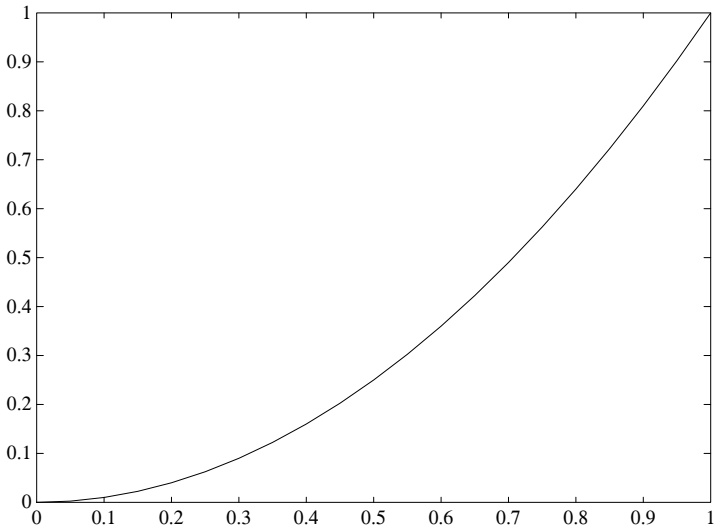

Figure 5 Characteristics of the nonlinear function at the process input

Stationary point (PV)	Behavior
10%	
70%	

What can be concluded from this?

Exercise 8 Now use gain scheduling to get better behavior over the whole working range. What is the best choice of scheduling variable? Which limits are suitable for the different parameter sets?

Tune the different parameter sets automatically and compare the obtained controller parameters. Write the resulting PID parameters down in the table below after tuning for each set.

Parameter set	K	T_i	T_d
PID1			
PID2			
PID3			

Is the closed-loop performance improved by using gain scheduling?

8. Case study

As a final task, you should now apply your knowledge of autotuning and gain scheduling to tune a controller for a “real” process.

Many pharmaceuticals and industrially used enzymes are today produced using microorganisms cultivated in a so called bioreactor. In aerobic processes it is important to keep an appropriate level of dissolved oxygen in the reactor. The consumption of oxygen is mainly determined by the nutrient feed rate and variations in this rate can be seen as disturbances. Oxygen is supplied by sparging of air and the supply can be varied by changing the stirrer speed in the reactor. Below, we will consider control of dissolved oxygen when the stirrer speed is used as control signal.

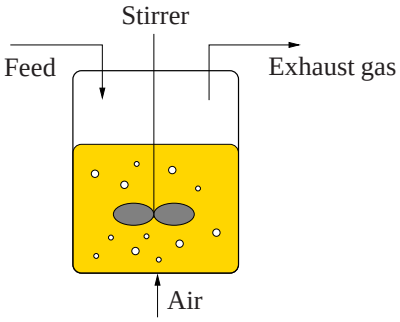

Figure 6 Sketch of a stirred bioreactor.

The difficulty with this process is that the process dynamics change significantly with the operating condition. This may cause tuning problems when controllers with fixed parameters are used. Typical observations are stability problems for low oxygen uptake rates, that is, low stirrer speeds, and sluggish control at high oxygen uptake rates.

The Simulink model `bioreact.mdl`, contains a model of the dissolved oxygen dynamics in a bioreactor where the time constants have been scaled to be 5 times faster than a real process. The process output is the dissolved oxygen tension (0–100%) and the input is the stirrer speed which take on values between 300 to 1200 rpm. These values correspond to 0–100% in the ECA600. There is also a load disturbance OUR representing the oxygen uptake rate.

Exercise 9 Your task is now to use the ECA600 controller for control of dissolved oxygen in the bioreactor. The setpoint in dissolved oxygen should be 30% and the requirements are that the solution time T_s should be less than 7 s throughout the operating range ($15 < OUR < 90$). It is also important that the closed-loop system is well damped.

9. References

[Åström and Wittenmark(2008)] K.J. Åström and Björn Wittenmark. *Adaptive Control*. Dover Publications, Mineola, N.Y., 2008. Reprint of book previously published by Addison-Wesley.

[Khalil(2002)] Hassan K Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

Document history:
Created by Per-Olof KÃd'llén
Revised August 1999 by Johan Alfvén, Mats ÃĚkesson, and Tore HÃd'gglund
Revised July 2004 by Brad Schofield
Revised August 2009 by Kristian Soltesz and Per-Ola Larsson