Contents

1 Introduction **1**

1.1 The importance of control 1

1.2 Software tools for analysis and design of control systems . . . 5

1.3 A short history of control 6

2 Introduction to feedback control **11**

2.1 Introduction 11

2.2 Terminology. Formulation of the control problem 11

2.3 Solutions to the control problem 14

2.3.1 Introduction 14

2.3.2 Control using a constant control signal 14

2.3.3 Control using error-based control signal (feedback control) 16

2.4 Examples of control systems. Documentation with P&I diagram and block diagram 19

2.5 Function blocks in the control loop 22

2.6 Controller functions 27

2.6.1 Introduction 27

2.6.2 Tuning the nominal control signal 28

2.6.3 On/off controller 30

2.6.4 Overview: The PID controller 33

2.6.5 P controller 34

2.6.6 PI controller 37

2.6.7 PID controller 40

2.6.8 Positive or negative controller gain? 51

2.7 Practical problems: Control kicks, windup, and noise 54

2.7.1 Reducing P- and D-kick caused by setpoint changes . 54

2.7.2 Integrator anti wind-up 59

2.7.3 Measurement noise. Signal variance 62

2.8 Performance index of control systems 71

2.9 Selecting P, PI, PD, or PID? 73

2.10 Reduction of control error by process changes 74

2.11 Control loop stability 75

3 Control equipment 79

3.1 Process controllers 79

3.2 PLCs and similar equipment 83

3.3 SCADA systems and DSC systems 84

3.3.1 SCADA systems 84

3.3.2 DCS 86

3.4 Embedded controllers in motors etc. 86

4 Experimental tuning of PID controllers 89

4.1 Introduction 89

4.2 A criterion for controller tuning 90

4.3 The P-I-D method 91

4.4 Ziegler-Nichols' closed loop method 94

4.5 Åstrøm-Hägglund's On/off method 98

4.6 Ziegler-Nichols' open loop method 103

4.6.1 Ziegler-Nichols' open loop method used experimentally 103

4.6.2 Ziegler-Nichols' open loop method with transfer function models 105

4.7 Consequences of adjusting controller parameters 110

4.7.1 Introduction 110

4.7.2 Increasing K_p 110

4.7.3 Reducing T_i 111

4.7.4 Increasing T_d 111

4.8 Auto-tuning 112

4.9 PID tuning when process dynamics varies 117

4.9.1 Introduction 117

4.9.2 Gain scheduling PID controller 118

4.9.3 Adjusting PID parameters from process model 125

4.9.4 Adaptive controller 128

5 Discrete-time PID controller 131

5.1 Introduction 131

5.2 Computer based control loop 132

5.3 Development of discrete-time PID control function 133

5.3.1 PID control function on absolute form 133

5.3.2 Incremental PID control function 138

5.4 How the sampling interval influences loop stability and tuning 139

6 Analysis of feedback control systems 145

6.1 Introduction 145

6.2 About using simulators 146

6.3 Setpoint tracking and disturbance compensation 147

6.3.1 Introduction 147

6.3.2 Analysis based on differential equation models 148

6.3.3 Transfer function based analysis of setpoint tracking and disturbance compensation 151

6.3.4 Frequency response analysis of setpoint tracking and disturbance compensation 161

6.4 Stability analysis of feedback systems 170

6.4.1 Introduction 170

6.4.2 Review of stability properties 170

6.4.3 Nyquist's stability criterion 173

6.4.4 Stability analysis in a Bode diagram 182

6.4.5 Stability margins and robustness 184

7 Transfer function based PID tuning 189

7.1 Introduction 189

7.2 Controller tuning from specified characteristic polynomial . . 192

7.2.1 Introduction 192

7.2.2 Tuning a controller for an integrator process 192

7.2.3 Tuning a controller for a first order process 201

7.3 Controller design using the direct method 205

7.4 Dead-time compensator (Smith predictor) 208

7.5 Skogestad's method 211

7.5.1 Introduction 211

7.5.2 Skogestad's tuning formulas 213

7.5.3 Skogestad's method with faster disturbance compensation 216

7.5.4 Skogestad's method for processes without time delay . 218

7.6 Controllers with two degrees of freedom 220

8 Frequency response based PID tuning 223

8.1 Introduction 223

8.2 Useful facts about frequency response of a control loop 224

8.2.1 Frequency response of the loop transfer function . . . 224

8.2.2 Frequency response of PID and PI controller 225

8.3 Ziegler-Nichols' frequency response method 229

8.3.1 Introduction 229

8.3.2 Tuning procedure 230

8.3.3 Comparing PID and PI tuning in frequency domain . 233

8.4 Frequency response based adjustment of PID parameters . . . 236

8.4.1 Introduction 236

8.4.2 Adjusting K_p 238

8.4.3 Adjusting T_i 239

8.4.4 Adjusting T_d 241

8.4.5 Summary 243

9 Various control methods and control structures 245

9.1 Feedforward control 245

9.1.1 Introduction 245

9.1.2 Designing feedforward control from differential equation models 247

9.1.3 Designing feedforward control from transfer function models 250

9.2 Cascade control 253

9.3 Ratio control and quality and product flow control 261

9.3.1 Ratio control 261

9.3.2 Quality and production rate control 262

9.4 Split-range control 263

9.5 Control of product flow and mass balance in a plant 264

9.6 Multivariable control 267

9.6.1 Introduction 267

9.6.2 Single loop control with PID controllers 268

9.6.3 Single loop PID control combined with decoupling . . 272

9.6.4 Model-based predictive control 274

A Codes and symbols used in Process & Instrumentation Diagrams 279

A.1 Letter codes 279

A.2 Instrumentation symbols used in P&IDs 279

B The Laplace transform 285

B.1 Properties of the Laplace transform 285

B.2 Transform pairs 286

Preface

This book gives an introduction to PID control of dynamic systems. The PID controller (PID = Proportional Integral Derivative) is the dominating (most frequently used) controller function in industry. This book can be used as a text-book in control courses in B.Sc. studies and in M.Sc. studies. It may also serve as a reference for engineers working in the industry.

The book describes the theory, but does not (except in a few cases) describe computer tools for analysis and design. However, lots of supplementary material are available from the homepage of the book on **http://techteach.no**. This material is in the form of documents which describes how analysis, simulation, and design of dynamic systems can be performed in MATLAB, Octave[1], SIMULINK, and LabVIEW. From this homepage there is also a link to KYBSIM (http://techteach.no/kybsim) which is a library of freely available simulators. Many of these simulators are used in this text book.

To benefit from all parts of the book, you must be familiar with systems theory of continuous-time dynamic systems – specifically basic mathematical modeling, differential equations, transfer functions, block diagrams, first and second order systems and frequency response.[2]

The theoretical tools for analysis and design described in this book is for continuous-time feedback control systems. The theoretical tools for analysis and design of discrete-time (sampled) feedback systems are quite similar to tools for continuous-time systems, and they are described in documents available for free on http://techteach.no.

[1] Octave is a free mathematical tool, quite similar to MATLAB, with lots of in-built function categories, like the toolboxes in MATLAB. Octave is available from http://www.octave.org.

[2] These topics are included in the textbook **Dynamic systems – modelling, analysis and simulation** by F. Haugen, Tapir Academic Publisher, 2004. (Information on http://techteach.no.)

A textbook covering advanced control topics building on the present book will be available during 2004. (Information is given on http://techteach.no.)

The book focuses on topics which I have found practically important. I have tried to describe the material in a simple and understandable way. I will appreciate suggestions and comments about both the presentation in the book and the choice of topics (e-mail to finn@techteach.no).

A comment about mathematical notation used in the book: Given a function of time, say $f(t)$. Taking the Laplace transform of $f(t)$ yields, say $F(s)$. Different symbols are used since they are different functions. However, because it is very convenient to do it, I have chosen to use the same symbol for both the time function and the corresponding Laplace transform in this book. So I write $f(s)$ for the Laplace transform of $f(t)$. It is my experience that this style of notation does not cause problems or misunderstandings.

The book is written with the text formatting program Scientific Word. LabVIEW, MATLAB, and SIMULINK are used as computer-based tools for analysis and simulation. Most simulations are performed with LabVIEW.

An exercise book with solutions is available during 2004 (information will be given on http://techteach.no).

A few words about my background: I have a M.Sc. degree (1985) in Engineering cybernetics from the Norwegian Institute of Technology. I have been doing teaching, writing, programming, and consulting since then. I have now a teaching position at the Telemark University College. I also work in my one-man company TechTeach.

I want to thank my family for giving me good working conditions while writing this book.

FinnHaugen

Skien, Norway, August 2004

Chapter 1

Introduction

1.1 The importance of control

Control engineering is a fascinating and important field. In short, control engineering is the methods and techniques used in technical systems having the ability of automatically correcting its own behaviour so that specifications for this behaviour are satisfied. The following process variables are typical objects of control:

- Level or weight (mass)
- Pressure
- Temperature
- Flow
- pH
- Speed
- Position

Due to control engineering, a supply ship will stay at or close to a specified position without anchor; a painting robot paints accurately and smoothly on a car body; The temperature and the composition in a chemical reactor will follow the specifications defined to give an optimal production; A turbine generator produces AC voltage of the specified frequency of 50 Hertz; The pen of an X-Y-plotter draws (follows) a varying voltage signal with great precision; The tool of a rotational cutting machine cuts the work-pieces with high precision; The emission of ammonia from a fertilizer producing factory is kept within limits established by law; The pH value and the composition of Nitrogen, Phosphate and Potassium in the fertilizer which is sent to the market lies between certain quality limits. And many more examples can be given.

Control engineering may be of crucial importance for the following applications:

- **Product quality**: A product will have acceptable quality only if the difference between certain process variables and their setpoint values – this difference is called the *control error* – are kept less than specified values. Proper use of control engineering may be necessary to achieve a sufficiently small control error, see Figure 1.1.

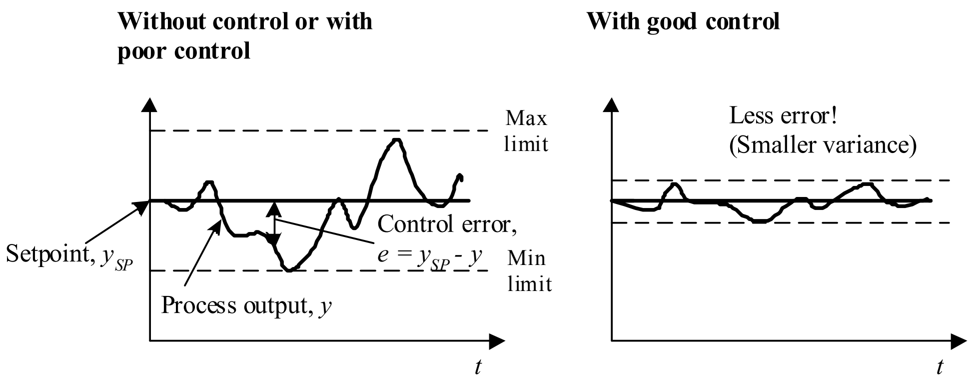

Figure 1.1: Good control reduces the control error

  One example: In fertilizers the pH value and the composition of Nitrogen, Phosphate and Potassium are factors which express the quality of the fertilizer (for example, too low pH value is not good for the soil). Therefore the pH value and the compositions must be controlled.

- **Production economy**: The production economy will be deteriorated if part of the products has unacceptable quality so that it can not be sold. Good control may maintain the good product quality, and hence, contribute to good production economy. Further, by good control it may be possible to tighten the limits of the quality so that a higher price may be taken for the product!

- **Security**: To guarantee the security both for humans and equipment, it may be required to keep variables like pressure, temperature, level, and others within certain limits– that is, these variables must be controlled. Some examples:
  - An aircraft with an autopilot (an autopilot is a positional control system).
  - A chemical reactor where pressure and temperature must be controlled.
- **Environmental care**: The amount of poisons to be emitted from a factory is regulated through laws and directions. The application of control engineering may help to keep the limits. Some examples:
  - In a wood chip tank in a paper factory, hydrogen sulfate gas from the cookery is used to preheat the wood chip. If the chip level in the tank is too low, too much (stinking) gas is emitted to the atmosphere, causing pollution. With level control the level is kept close to a desired value (set-point) at which only a small amount of gas is expired.
  - In the so-called washing tower nitric acid is added to the intermediate product to neutralize exhaust gases from the production. This is accomplished by controlling the pH value of the product by means of a pH control system. Hence, the pH control system ensures that the amount of expired ammonia is between specified limits.
  - Automatically controlled spray painting robots avoid humans working in dangerous areas. See Figure 1.2.

Figure 1.2: Spray painting robot (IRB580, ABB)

- **Comfort**:

  - The automatic positional control which is performed by the autopilot of an aircraft to keep a steady course contributes to the comfort of the journey.
  - Automatic control of indoor temperature may give better comfort.

- **Feasibility**: Numerous technical systems could not work or would even not be possible without the use of control engineering. Some examples:

  - An exothermal reactor operating in an unstable (but optimal) operating point
  - Launching a space vessel (the course is stabilized)
  - A dynamic positioning system holds a ship at a given position without an anchor despite the influence of waves, wind and current on the ship. The heart of a dynamic positioning system is the positional control system which controls the thrusters which are capable of moving the ship in all directions. See Figure 1.3.

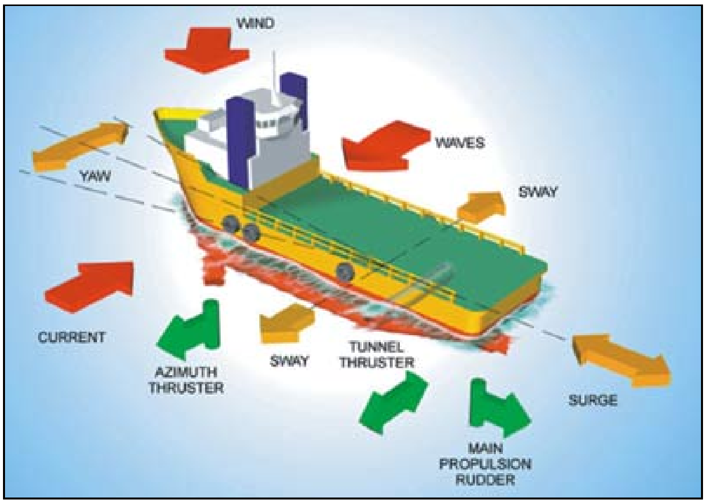

Figure 1.3: A dynamic positioning system holds a ship at a given position without an anchor despite the influence of waves, wind and current on the ship (Kongsberg Simrad, Norway)

- **Automation**: Due to automatic control the operators can perform various tasks in stead of continuously controlling the process, for example perform maintenance or just resting.

1.2 Software tools for analysis and design of control systems

Some typical tasks for software tools for analysis and design of control systems are:

- **Analysis of control systems:**
  - Calculating poles and eigenvalues to observe dynamic properties and stability properties.
  - Calculating frequency response to observe dynamic properties in the term of bandwidth and stability properties.
  - Simulating control systems to observe
    * dynamic properties,
    * control system robustness against noise and parameter variations,
    * implications of nonlinear elements in the control loop, as saturation, hysteresis, etc.
- **Design of control systems:**
  - Calculation of controller parameters on basis of a mathematical model of the control system from specifications to time response, frequency response or stability.
  - Tuning controller parameters by applying an experimental method in a simulator.
  - Trying out various control system structures and control methods on a simulator.

MATLAB[1] with Control System Toolbox [8] and SIMULINK covers the above items. Octave[2], which is a freely available MATLAB-like computer tool for numeric analysis and visualization, includes a set of functions which are similar to the functions of Control Toolbox in MATLAB. LabVIEW[3] with Control Design Toolkit, PID Control Toolkit and Simulation Module also supports the items above. In addition, LabVIEW has powerful tools for developing graphical user interfaces, and has comprehensive I/O-support (Input/Output) to physical processes. (On the

[1] Produced by The MathWorks
[2] http://www.octave.org
[3] Produced by National Instruments

homepage of this book you can find documents and other files which describes using MATLAB, SIMULINK and LabVIEW to such analysis and design, including simulation.)

Computer tools as described above assumes that a mathematical model of the process to be controlled, is available. To develop a precise physics based models for industrial processes is a demanding task. Months of work may be required, except for the most simple processes, as the wood-chip tank described in Example 2.3 (page 19). However, tools are available, as MATLAB's System Identification Toolbox and LabVIEW's System Identification Toolkit, for development of input-output-models in the form of transfer functions or state-space models from experimental data, and these models can be used for analysis and design, as described above. Note that for teaching and training testing (trial) purposes simplified models can be very useful.

There are commercially available simulator for processes, including control systems, based on precise models of processes as heat exchangers, reactors, and columns. The instrumentation diagram of the process constitutes the user interface. Examples of such simulators are Hysys [4] and ASSETT[5].

1.3 A short history of control

Back in 2000 B.C. the Babylonians constructed automatic watering systems based on level control. The old Greeks constructed level control systems for water clocks and oil lamps. The weight control system shown in Figure 1.4 seems to be an automatic bartender.

In the fifteenth and sixteenth century there were made temperature control systems for incubators (heating boxes for eggs), pressure control systems for boilers, and position control systems for wind mills.

In 1788 James Watt constructed a speed control system for a steam engine, see Figure 1.5. Watt's speed control system was based on feedback from measured rotational speed to the opening of the steam valve via a centrifugal controller, which works as follows: The larger the speed, the smaller the valve opening (and steam supply), and vice versa. In this way the speed was held at or near a constant set-point value, despite the disturbances as variations in the steam pressure and changing load torques acting on the engine shaft. Watt's speed control system is regarded as the

[4] Produced by AspenTech
[5] Produced by Kongsberg Simrad

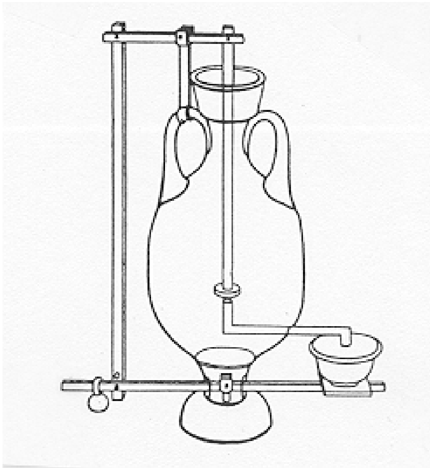

Figure 1.4: A weight control system from the Antics. An automatic bartender? [12]

first industrial application of control engineering.

Watt's control system was not based on any accurate mathematical analysis, but on experiments and trial-and-error. In 1868 James C. Maxwell made a mathematical analysis of the speed control system, and this analysis may be regarded as the stating point of the theoretical methods for analysis and design of control systems.

The field of control engineering and control theory has had an enormous development since 1930. Mechanical and/or pneumatic controllers were developed for the process industry. The first controllers has proportional action only, and later integral and derivative action was implemented. The controller was typically a physical unit mounted on the control valve. There was lack of good methods for tuning the controller parameters. However, this problem was solved by Ziegler and Nichols [20] around 1940. Their controller tuning methods remains among the very best methods available today, and their two methods are described in this book. Their work increased the availability of control engineering in the process industry. Mr. Ziegler was also involved in the first commercial PID controller (Fulscope 100 produced by Taylor Instruments & Co. at the end of the nineteen-thirties).

The big steps, or the new directions, in the control theory have typically been initiated by practical problems which had to be solved. One example is the development of feedback electronic amplifiers with Bell Telephone Lab. in USA in the thirties which led to the *frequency response* methods

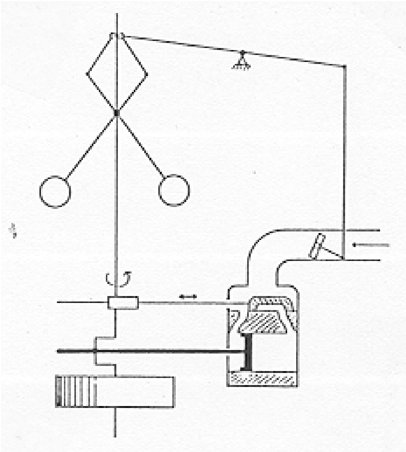

Figure 1.5: Principal diagram of James Watt's speed control system. (Based on [25].)

for analysis and design of feedback amplifiers and feedback control systems. One other example is the development of control systems for radar systems and artillery under The Second World War. The development of the space technology in the Soviet Union and the USA in the fifties and the sixties raised problems which were attempted to be solved by *optimal control* which is formulated by using *state-space methods*. (A state-space model is a set of first order differential equations describing the system.) In an optimal control system there is an optimal balance between the "amount" of control power used and the control error. The optimal solution minimizes a certain optimal criterion.

The development of auto-pilots required *adaptive controllers* as there was a need for control systems which adapted to the varying dynamics of the aeroplane during the flight. The first adaptive controllers were *gain scheduling* controllers, in which the PID parameters are found from a table-lookup in a table or schedule of precalculated PID parameter values. In the 1980's the first generic commercial adaptive PID controllers were introduced. In these adaptive controllers a process model is estimated continuously, and the PID parameters are automatically adjusted from this model.

In the late 1980's and the 1990's there were much interest in *fuzzy control*. Fuzzy control is available in several commercial controllers. The theoretical basis stems from the fuzzy-logic developed by Lotfi Zadeh around 1965. Fuzzy control is particularly suited for processes where the knowledge about how to control it is in the form of an empirically developed set of rules.

From the mid 1980's *model-based predictive control* or MPC has been in the focus of the research of control methods. Several vendors now offer MPC-modules, and MPC has been applied in various industries. MPC is based on a mathematical process model, which can be in the form of a transfer function model or a step-response model or a state-space model. The models used in MPC include the physical limits of the process to be controlled. The MPC algorithm calculates a future sequence of the control variable from an criterion which typically is a criterion containing quadratic terms of the control variable and the control error. From this sequence the first element is used to actually control the process. The MPC algorithm is executed regularly with a fixed time-step. MPC has proven to give good control of difficult processes, as nonlinear multivariable processes with dead-time. We may say that MPC is the next most important control method in the industry today (next to PID control).

Chapter 2

Introduction to feedback control

2.1 Introduction

I this chapter the control problem is defined, and the principle of feedback is introduced as the most important solution to the control problem. Furthermore, standard industrial controller functions based on feedback are described. These are versions of the PID controller. The on/off controller is also introduced. Many practical aspects of the PID controlled are described. The chapter also shows how control systems can be documented in process and instrumentation diagrams – P&I-diagrams or just P&IDs – and block diagrams.

2.2 Terminology. Formulation of the control problem

Control engineering solves a *control problem*. We will soon formulate it, but first we need to define the terminology which will be used.

Figure 2.1 shows a general *block diagram* of the process, which can be of material, mechanical, thermal or electrical type (concrete examples follows soon). Below are definitions of the quantities shown in Figure 2.1.

- **The process** is the physical system which is to be controlled.

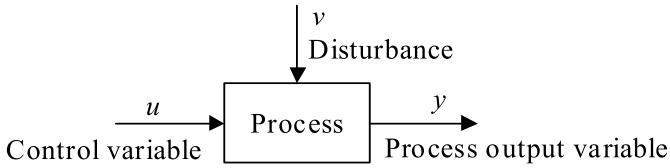

Figure 2.1: Block diagram representation of a process with input and output variables

Included in the process is the actuator, which is the equipment with which (the rest of) the process is controlled.

- **The control variable or the manipulating variable** is the variable which the controller uses to control or manipulate the process. In this book u is used as a general symbol of the control variable. In commercial equipment you may see the symbol MV (manipulating variable).

- **The process output variable** is the variable to be controlled so that it becomes equal to or sufficiently close to the setpoint. In this book y is used as a general symbol of the process output variable. In commercial control equipment PV (process variable or process value) may be used as a symbol.

  The process output variable is not necessarily a physical output from the process! One example: In a heat exchanger where the temperature of the product flow is to be controlled, the temperature – and *not* the product flow – is the process output variable.

- **The disturbance** is a non–controlled input variable to the process which affects the process output variable. This influence is undesirable, and the controller will adjust the control variable to compensate for the influence. In this book v is used as a general symbol for the disturbance.

Above, the control variable, the process output variable and the disturbance were assumed to be scalar values. In general, however, there may be several of these variables, most typically: More than one disturbance.

To formulate the control problem, we need a few more definitions:

- **The setpoint or the reference** is the desired or specified value of the process output variable. The general symbol y_{SP} will be used in this book.

- **The control error** is the difference between the setpoint and the process output variable:

$$e = y_{SP} - y \tag{2.1}$$

Now let us formulate:

The control problem:

Adjust the control variable u so that the control error e is within acceptable limits.

"Within acceptable limits" typically means that *the steady-state or static control error*, e_s, is zero:

$$e_s = \lim_{t\to\infty} e(t) = 0 \tag{2.2}$$

The static control error is the error when all variables have (converged to) constant values.

In practical control systems there are random noise and disturbances acting on various parts of the system, causing the control error to fluctuate randomly around its mean value, see Figure 2.2. The requirement $e_s = 0$

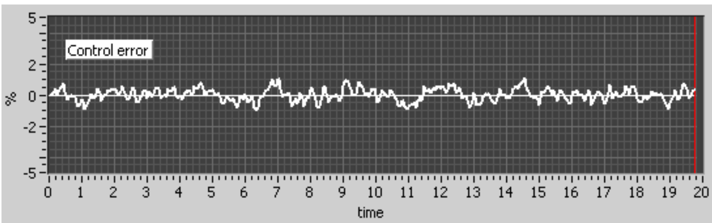

Figure 2.2: In practical control systems the control error fluctuates more or less randomly around its mean value.

then must be interpreted as zero mean value of e. In most of the examples of control systems described in this book, the system is assumed to be noise-free. However, consequences of random measurement noise in control systems is treated in Section 2.7.3.

The present value of the process output variable y defines the *operating point* of the process, for example water level of 3.2 meters or product temperature of 150°C. If y is equal to y_{SP}, we say that the process is in the specified operating point. Usually a specific operating point is steady-state which means that all process variables have constant values. If necessary, the control variables and the disturbances can also be included in the specification of an operating point.

2.3 Solutions to the control problem

2.3.1 Introduction

The control problem is about finding the value of the control variable u so that the control error e becomes sufficiently small. Two ways to try to solve the control problem are as follows:

- Using a *constant control signal*, independent of the present value of the control error.
- Using a *control signal which is continuously adjusted as a function of the control error.*

These two solutions are described in more detail in the following sections.

There is actually a third way to control a process: By continuously calculating the control signal from a mathematical model of the process to be controlled. This control method is however not easy to use in practice since an accurate model expressing the dynamics of the process is not easily available for most processes. The method is called *feedforward control*, and it is described in Section 9.1.

2.3.2 Control using a constant control signal

Using a constant control signal is the simplest way to control a process. Figure 2.3 shows a block diagram of the process controlled by a constant control signal

$$u = u_0 = \text{constant} \tag{2.3}$$

The constant control signal u_0 can be tuned in two ways:

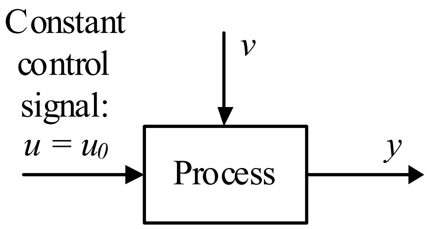

Figure 2.3: Controlling the process with a constant control signal, $u = u_0$.

- **Experimentally**: u_0 is adjusted until we observe that the process output variable y (or its measurement value) is approximately equal to the setpoint y_{SP} in steady-state, and then u_0 is fixed at this value.
- **Calculated from a mathematical process model**: This approach eliminates possibly expensive or time consuming experiments on the physical process, but a mathematical process model is required. The procedure is the same as for finding the nominal control variable used in feedback control, which is described – with a concrete example – in Section 2.6.2.

If there are no changes in the setpoint or in the disturbance, using a constant control variable is an acceptable solution. But if the setpoint or the disturbance varies – a common situation in real control systems – the control error may be too large. A better, but more complicated solution is feedback control which is described in Section 2.3.3.

Example 2.1 ***Controlling a process with constant control signal***

Figure 2.4 shows typical responses for a simulated process controlled by a constant control signal.[1] The control variable u has a constant value, $u_0 = 50$. (The unit of this value is not important here, but it may be percent.) Initially, the setpoint is $y_{SP} = 50$ and the disturbance is $v_0 = 0$. The setpoint y_{SP} is changed as a step from 50 to 70 (so the amplitude is 20) at $t = 5$, and the disturbance v is changed as a step from 0 to -20 (amplitude -20) at $t = 15$. Figure 2.4 shows a *steady-state control error different from zero* (more specifically 20) after the step in y_{SP}, and the error increases to 40 after the step in v.

[End of Example 2.1]

[1] The simulated process is a first order system with time delay.

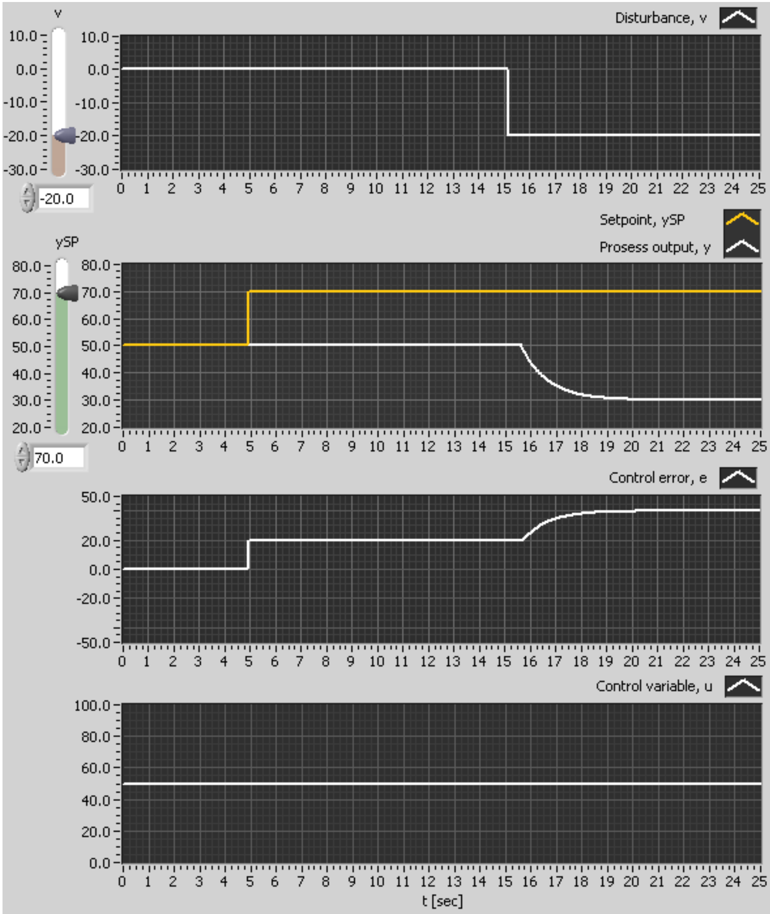

Figure 2.4: Responses in a simulated process controlled by a constant control signal. The control system is excited by a setpoint step and a disturbance step.

The solution of using a constant control variable is sometimes denoted open loop control, since it can be regarded an alternative to closed loop control which is described in Section 2.3.3). The solution can also be regarded as static feedforward, cf. Chapter 9.1.

2.3.3 Control using error-based control signal (feedback control)

The problem with control with constant control signal, cf. Section 2.3.2, is that there is no adjustment of the control variable if there are changes in the setpoint or in the disturbance. Consequently the control error can be different from zero and maybe too large. We should expect better control, that is, smaller control error, if the control signal is calculated continuously *as a function of the control error*. Since the error $e = y_{SP} - y$, y must be *measured*. Figure 2.5 illustrates this solution, which is error-driven control or *feedback control* since there is a connection from the process output variable y back to the control variable (the process input) u. The loop which consists of process, sensor and controller is called the *control loop*.

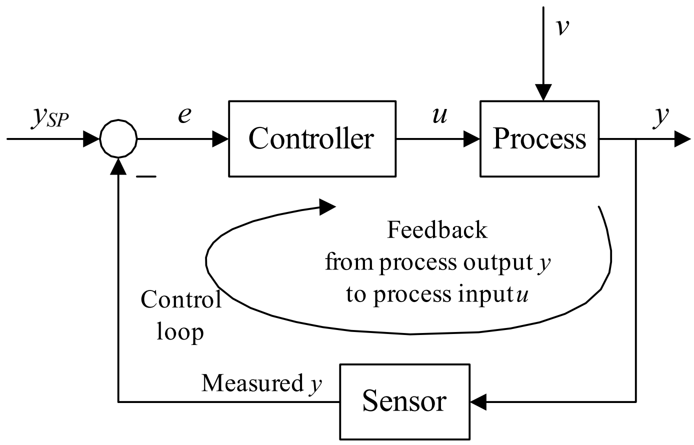

Figure 2.5: Feedback control

The calculation of u takes place in *the controller*. The term controller here means *controller function*, which usually is implemented in a computer program in the control equipment. We will use the term controller for both the control function and the physical equipment in which the control function is implemented.

Example 2.2 ***Controlling a process with error-based control signal***

Figure 2.6 shows typical responses in a feedback control system for a simulated process (the process is the same as used in the simulation shown in Figure 2.4).[2] The control variable u has initially value 50, the setpoint has value 50, and the disturbance has value 0. The control system is excited by a step in the setpoint y_{SP} from 50 to 70 (amplitude 20) at $t = 5$, and with a step in the disturbance v from 0 to -20 (amplitude -20) at $t = 15$. Figure 2.6 shows responses in the control variable u and in the process output variable y. We see that the control variable changes value when the control error e changes value (from zero), which takes place after

[2] The simulated process is a first order system with time delay. The controller is a PI-controller.

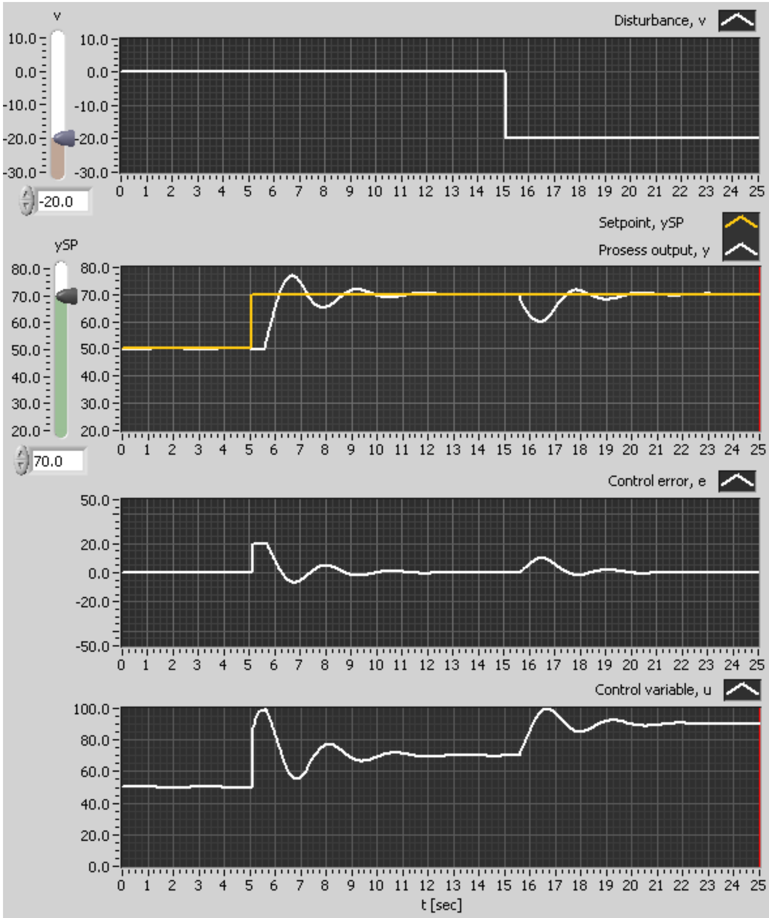

Figure 2.6: Typical responses in a feedback control system, where the control variable is continuously calculated as a function of the control error. The control system is excited by a setpoint step and a disturbance step.

the steps in y_{SP} and v. In this simulated control system the static control error, e_s, becomes zero both after the steps in y_{SP} and after the step in v. This is a *large improvement* compared to using a constant control variable, cf. Example 2.1 and Figure 2.4.

[End of Example 2.2]

A control system which is capable of getting static control error for any constant setpoint value and any constant disturbance value is said to give *perfect static control*.

2.4 Examples of control systems. Documentation with P&I diagram and block diagram

Sections 2.3.2 and 2.3.3 indicated that the control error becomes smaller with feedback control (error-based control) than with a constant control variable. Therefore, feedback control is the main control principle. In the following several examples of feedback control systems are described. These are control system for a wood-chip tank, a heated liquid tank, a motor, and a shower. The control systems, except the latter, are described or documented in two ways:

- **Process and instrumentation diagram or P&I diagrams** which is a common way to document control systems in the industry. This diagram contains easily recognizable drawings and symbols of the process to be controlled, together with symbols for the controllers and the sensors and the signals in the control system. Appendix A gives a small overview over some of the most frequently used symbols in P&I diagrams. There are international standards for instrumentation diagram, but you must expect that company standards are used.
- **Block diagram**, which are useful in principal and conceptual descriptions of control systems.

Example 2.3 ***Level control of a wood-chip tank***

Figure 2.7 shows a P&I diagram and a block diagram of a level control system for a wood-chip tank with feed screw and conveyor belt (which moves with constant speed).[3] Chip is consumed via a outlet screw in the bottom of the tank. This outflow is a disturbance to the control system. The chip level h shall be controlled to be equal or approximately equal to a given level setpoint h_{SP}.[4] LT (Level Transmitter) represents the level sensor. (The levels sensor is based on ultrasound: The level is calculated from the reflection time for a sound signal emitted from a transmitter to a

[3] Such a tank is in the beginning of the process string in the paper mass factory Södra Cell Tofte in Norway.

[4] A few words about the need for a level control system for this chip tank: Hydrogene sulphate gas from the cookery is used to preheat the wood chip. If the chip level in the tank is too low, too much (stinking) gas is emitted to the athmosphere, causing pollution. With level control the level is kept close to a desired value (set-point) at which only a small amount of gas is expired. The level must not be too high, either, to avoid overflow and reduced preheating temperature increase.

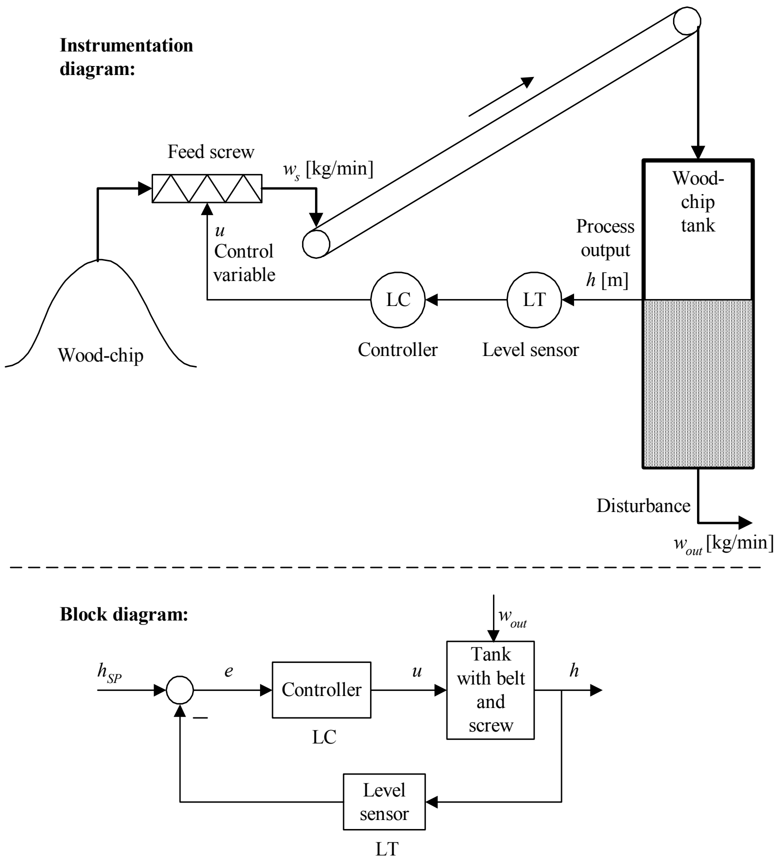

Figure 2.7: P&I diagram and block diagram of a level control system for a wood-chip tank

receiver.) LC is the Level Controller. The setpoint is usually not shown explicitly in an P&I diagram (it is included in the LC block). The controller controls the chip level by manipulating the (rotational speed of the) feed screw.

[End of Example 2.3]

Example 2.4 ***Temperature control of heated liquid tank***

Figure 2.8 shows a P&I diagram and a block diagram of a temperature control system for a heated liquid tank with continuous inlet and outlet flow. This process can represent a heat exchanger or a heated reactor in a process line. The temperature T is to be controlled. The temperature setpoint is T_{SP}. Important disturbances of the temperature control system are the inflow temperature T_{in} and the environmental temperature T_e. The controller controls the temperature by manipulating the power via the control signal u to the power amplifier.

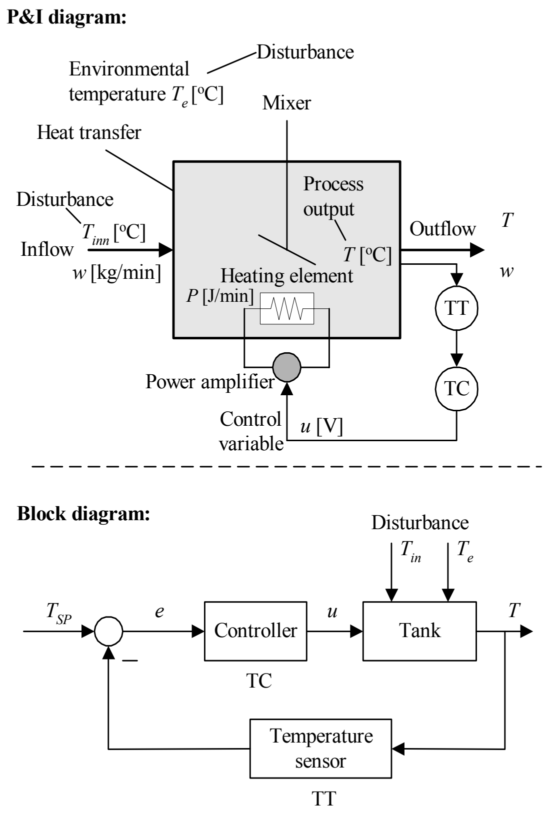

Figure 2.8: P&I diagram and a block diagram for a temperature control system for a heated liquid tank

[End of Example 2.4]

Example 2.5 ***Speed control of motor***

Figure 2.9 shows a P&I diagram and a block diagram of a speed control system for a motor (electrical or hydraulic). The load can be a tool or a conveyor belt. The rotational speed n is to be controlled. The speed setpoint is n_{SP}. The motor is influenced by a load torque T_L, which is a disturbance on speed control system. The controller controls the rotational speed by manipulating the power supplied to the motor.

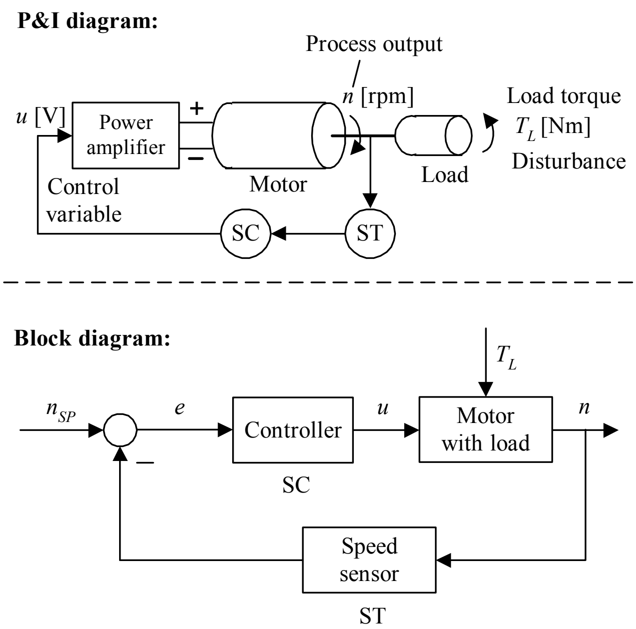

Figure 2.9: P&I diagram and a block diagram of a speed control system for a motor. SC is speed controller. ST (Speed Transmitter) is speed transmitter.

[End of Example 2.5]

Example 2.6 ***Control of shower water temperature***

When we adjust the position of the crane in the shower on the basis of the hand measured temperature to obtain a pleasant water temperature, we actually implement feedback temperature control. The hand is the sensor, the brain is the controller, and the nerve signal which via the (other) hand controls the crane position, is the control variable. Figure 2.10 shows the process and a block diagram of the temperature control system.

[End of Example 2.6]

2.5 Function blocks in the control loop

We will now take a closer look at the function blocks in a control loop. The level control system for the wood-chip tank, cf. Example 2.3, will be used as a concrete example. Figure 2.11 shows a detailed block diagram of

The process:

Block diagram:

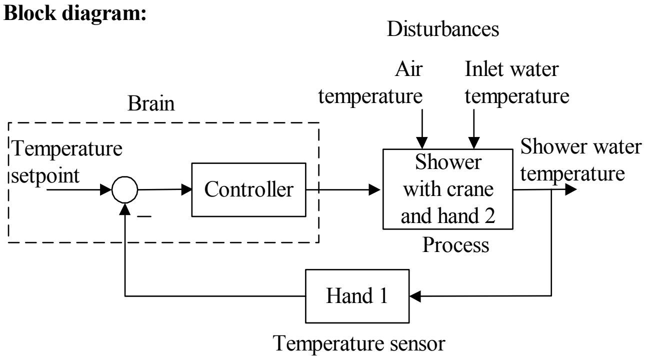

Figure 2.10: Feedback control of shower water temperature

the level control system. The block diagram contains a switch between *automatic mode* and *manual mode* of the controller:

- **Automatic mode:** The controller calculates the control signal using the control function (typically a PID control function).
- **Manual mode**: A human operator may adjust the control variable directly on the equipment, with the control (PID) function being inactive. The process is controlled by the manually adjusted control variable signal u_0 – *the nominal control value* . Typically, u_0 can not be adjusted in automatic mode. Its value is however included in the control signal in switching from manual to automatic mode to ensure bumpless transfer between the control modes.

  The operator sets the controller into manual mode during controller tuning or maintenance (you can then just not turn off the controller, otherwise e.g. the reactor would stop).

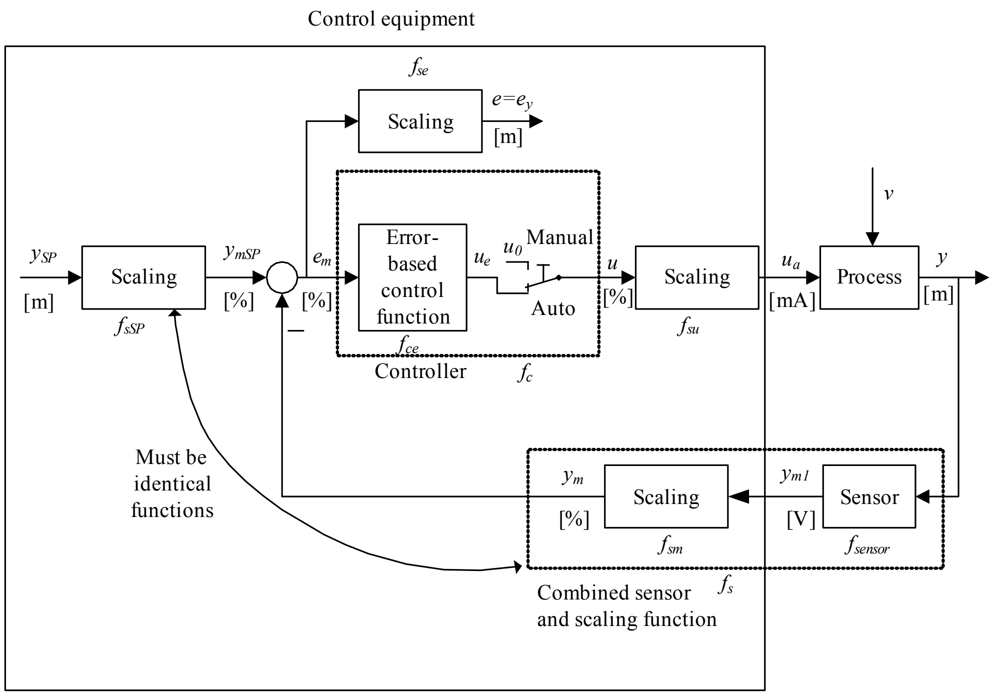

Figure 2.11: Detailed block diagram of the level control system

On commercial control equipment the operator typically can switch the controller between automatic and manual mode e.g. via a physical button or a menu on a screen.

The block diagram in Figure 2.11 contains scaling blocks for calculating the signals in proper units. Let us look at some of the blocks.

- **Sensor or measurement function**: The relation between the process output variable y and the physical sensor signal y_{m_1} can be expressed by the *sensor or measurement function* f_{sensor}:

  $$y_{m_1} = f_{sensor}(y) \tag{2.4}$$

  The sensor function is in most cases linear and can then be written on the form:

  $$y_{m_1} = \underbrace{K_{m_1}(y - y_0) + y_{m_{1_0}}}_{f_{sensor}(y)} \tag{2.5}$$

  where K_{m_1} is the measurement gain and y_0 is the value of the process signal y which gives the measurement value $y_{m_{1_0}}$. Figure 2.12 illustrates (2.5).

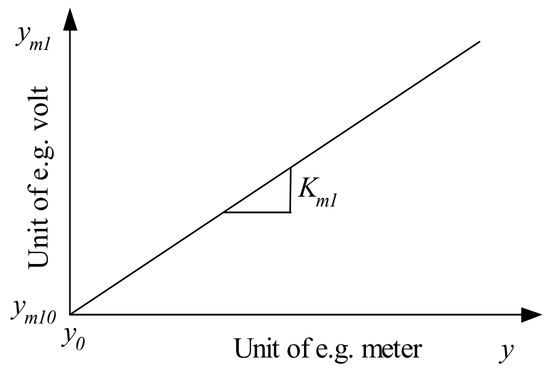

Figure 2.12: The sensor or measurement function $y_{m_1} = K_{m_1}(y - y_0) + y_{m_{1_0}}$

Usually the physical measurement signal y_m is a voltage signal or a current signal. For industrial applications a number of standard signal ranges of measurement signals are defined. Common standard ranges are 0 — 5V, −10 — +10V and 4 — 20mA.

Example (the wood-chip tank): Assume that the measurement signal range is 0 — 5V, and that this range corresponds to 0 — 15m (linearly). Thus,

$$y_0 = 0\text{m},\ y_{m_{1_0}} = 0\text{V},\ K_{m_1} = \frac{5 - 0\text{m}}{15 - 0\text{V}} \approx 0.33\text{V/m} \tag{2.6}$$

- **Scaling of measurement signal**: It is quite usual that the measurement signal is in unit % and that the %-value is used in the control function. The measurement signal y_{m1} in unit V is scaled to a corresponding signal y_m in unit % by the scaling function f_{sy}. With a linear f_{sy},

$$y_m = \underbrace{K_{sm}(y_{m_1} - y_{m_{1_0}}) + y_{m_0}}_{f_{sm}} \tag{2.7}$$

Commercial control equipment contains functions for scaling. During configuration of the controller, the user typically gives information about the minimum and the maximum values of the physical sensor signal, e.g. 4mA and 20mA, and the minimum and the maximum values of the scaled measurement signal, e.g. 0% and 100%. From this information the controller automatically configures the scaling function (2.7).

Example (wood-chip tank): Assume that the 0 — 5V measurement signal range corresponds to the range 0 — 100%. Then,

$$y_{m1_0} = 0\text{V};\ y_{m_0} = 0\%;\ K_{sy} = \frac{100 - 0\%}{5 - 0\text{V}} = 20\%/\text{V} \tag{2.8}$$

- **Combined scaling and sensor/measurement function**: The function f_s in Figure 2.11 is the combined function of (2.4) and (2.7). If both these functions are linear, the combined function is on the form

$$y_m = f_s(y) = K_m(y - y_0) + y_{m_0} \tag{2.9}$$

  Example (wood-chip tank): The combined scaling and measurement function from level in meter to level measurement signal in percent can be found from (2.4) and (2.7). However, in this case it can more easily be set up from the information that $0 - 15$m corresponds to $0 - 100\%$. Thus,

$$y_0 = 0\text{m},\ y_{m_0} = 0\%,\ K_m = \frac{100 - 0\%}{15 - 0\text{m}} = 6.67\%/\text{m} \tag{2.10}$$

- **Setpoint scaling**: The setpoint and the measurement signal must of course be represented with the same unit, otherwise subtracting the measurement signal from the setpoint is meaningless. The setpoint must be scaled using a scaling function which is equal to the combined scaling function for the measurement signal, (2.9). Thus,

$$y_{m_{SP}} = \underbrace{K_m(y_{SP} - \overbrace{y_{SP_0}}^{=y_{m_0}}) + y_{m_{SP_0}}}_{f_{sSP} = f_s} \tag{2.11}$$

- **The controller** calculates the control variable according to the control function (control functions are described in Section 2.6).

- **Scaling the control variable**: If the controller calculates the control variable u in % (which is quite usual), then a scaling of the %-value to the physical unit used by the actuator, is necessary. A linear scaling function is

$$u_a = \underbrace{K_{su}(u - u_{p_0}) + u_{a_0}}_{f_{su}} \tag{2.12}$$

  Example (wood-chip tank): Assume that the control variable in the range $0 - 100\%$ is to be scaled to the range $4 - 20$mA. In this case

$$u_{p_0} = 0\%,\ u_{a_0} = 4\text{mA},\ K_{su} = \frac{20 - 4\text{mA}}{100 - 0\%} = 0.16\text{mA}/\% \tag{2.13}$$

- **Scaling the control error**: To scale the control error e_m from measurement unit (typically %) to the unit used of the process output variable y, the following scaling function can be used:

$$e_y = \underbrace{K_{se}}_{f_{sy}} = \frac{1}{K_m} e_m \tag{2.14}$$

where K_m is the same as in (2.9).

Example (wood-chip tank): Scaling of the control error in % to meter is realized by

$$K_{se} = \frac{15 - 0\text{m}}{100 - 0\%} = 0.15\text{m}/\% \tag{2.15}$$

2.6 Controller functions

2.6.1 Introduction

Figure 2.11 shows where the control function – usually denoted simply "controller" – is placed in a control loop. The most common control functions are the following:

- On/off controller
- P controller (proportional)
- PD controller (proportional-derivative)
- PI controller (proportional-integral)
- PID controller (proportional-integral-derivative)

The P-, PD- and PI controllers can be derived from the PID controller. The PI- and the PID controller is by far the most commonly used in the industry since they gives best control: Zero static control error is achieved. The derivative term creates problems due to amplification of high frequent measurement noise, so that the control variable would become noisy, and therefore the D-term is not used in many practical cases. Thus, the PI controller is probably most frequently used.

The On/off controller is particularly easy to implement using an electrical or a mechanical on/off-element (as in a thermostat) or by simple expressions in a control program. However, the On/off controller gives a somewhat imprecise control because there will be sustained oscillations in the control loop. The On/off controller may be used for tuning of a PID controller, cf. Section 4.5.

The control functions which are described in detail in the following, are all on the following form:

$$u = u_0 + u_e \tag{2.16}$$

where u_0 is the nominal (manually adjusted) control variable and u_e is a function of the control error e, cf. Figure 2.11. (2.16) is illustrated in Figure 2.13. u_0 is the control signal required to keep the process in or close

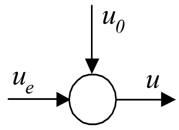

Figure 2.13: The control variable (or signal) is calculated as $u = u_0 + u_e$.

to the nominal (specified) operation point when the controller is in manual mode. u_0 can be adjusted by the operator while the controller is in manual mode, but is typically fixed while the controller is in automatic mode. u_0 may be used by the control function as an initial value of the control variable u when the controller is switched from manual to automatic mode. The term u_e in (2.16) represents the feedback term or the error based term, which gives compensation for changes in the setpoint or in the disturbances.

Tuning u_0 is described in Section 2.6.2. Various control functions producing (calculating) the feedback term u_e are described in Sections 2.6.3 – 2.6.7.

2.6.2 Tuning the nominal control signal

The nominal control signal u_0 can be tuned in two ways:

- **Experimentally**: u_0 is adjusted until we observe that the process output variable y (or, more correctly: its measurement value) is approximately equal to the setpoint y_{SP} in steady-state, and then u_0 is fixed at this value.

  Example: Figure 2.7 shows a feedback control system for a wood-chip tank. The nominal control signal is adjusted until it is observed that the level is constant and approximately equal to the level setpoint.

- **Calculated from a mathematical process model**: This approach eliminates possibly expensive experiments on the physical process, but it is required that a mathematical model exists. The procedure is as follows: u_0 *is calculated as the steady-state or static solution* u *of the equations which constitute the static process model in which the setpoint* y_{SP} *is substituted for the process output variable* y. The static process model is derived from a dynamic model in the form of differential equations by setting all time-derivatives equal to zero and neglecting any time-dependencies, as time delays.

Example 2.7 ***Nominal control signal for wood-chip tank***

A level control system for a wood-chip tank is described in Example 2.3 (page 19). We will now calculate the nominal control signal u_0 for the following operation point: The level h is equal to the level setpoint h_{SP}, and the chip outflow w_{out} has a constant (static) value of w_{out_s}.

We need a mathematical model. Assume the following: h [m] is the level. A [m^2] is the cross sectional area. ρ [kg/m^3] is the chip density. ρAh [kg] is the mass of chip in the tank. w_{in} [kg/min] is the chip inflow from the belt. w_s [kg/min] is the chip inflow to the belt from the screw. w_{out} [kg/min] is the chip outflow from the outlet in the bottom of the tank. K_s [(kg/min)/%] is the screw gain. τ [min] is the time delay or transport delay on the conveyor belt, which runs with constant speed. u [%] is the control variable.

Mass balance of the chip in the tank yields

$$\frac{d\left[\rho Ah(t)\right]}{dt} \equiv \rho A\dot{h}(t) = w_{in}(t) - w_{out}(t) \qquad (2.17)$$

$$= w_s(t-\tau) - w_{out}(t) \qquad (2.18)$$

$$= K_s u(t-\tau) - w_{out}(t) \qquad (2.19)$$

or, rearranged,

$$\dot{h}(t) = \frac{1}{\rho A}\left[K_s u(t-\tau) - w_{out}(t)\right] \qquad (2.20)$$

A *static* process model can be found from the dynamic the model (2.20) by setting the time-derivative equal to zero and neglecting the time delay:

$$\rho A\dot{h}_s(t) = 0 = \underbrace{K_s u_0}_{w_{in_s}} - w_{out_s} \qquad (2.21)$$

or[5]

$$u_0 = \frac{w_{out_s}}{K_s} \quad [\%] \tag{2.22}$$

This control signal will keep the chip level at a constant value, but at which value? Actually, at any level, since (2.22) does not contain information of the level – it just ensures that the inflow and the outflow are equal. For this process – and for many other tank processes - it is insufficient to use a fixed control signal to control the level. The control variable must contain an error-based term or feedback term to obtain control of the level, as explained in the following sections.

[End of Example 2.7]

2.6.3 On/off controller

The On/off controller calculates the control variable u according to (2.16), which is repeated here:

$$u = u_0 + u_e \tag{2.23}$$

The nominal control signal u_0 can be calculated as explained in Chapter 2.6.2. With the On-off controller the error-based or feedback term u_e is calculated as a function of the control error e as follows:

$$u_e = \left\{ \begin{array}{c} A \text{ for } e \geq 0 \\ -A \text{ for } e < 0 \end{array} \right\} \tag{2.24}$$

where A is the *amplitude*. The control error e has same unit as for the setpoint and the process measurement in the expression $e = y_{SP} - y$, typically %, or V, mA, m, ^{o}C or some other. Figure 2.14 illustrates u_e and u.

If your controller does not implement an On/off controller, you can get one by using a P controller with a very large (ideally: infinite) value of the controller gain K_p. The P controller is described in detail in Section 2.6.5.

Maybe you recognize the On/off controller from the room temperature control at home? The thermostat is actually a combined temperature sensor and an On/off controller.

Unfortunately, with an On/off controller all signals in the control loop will oscillate continuously (unless some saturation limit is reached). These oscillations comes *automatically*. This can be explained as follows: Assume that the control error e is positive. Then the control signal u equals

[5] I guess you could have come up with this relation without deriving the full dynamic model...

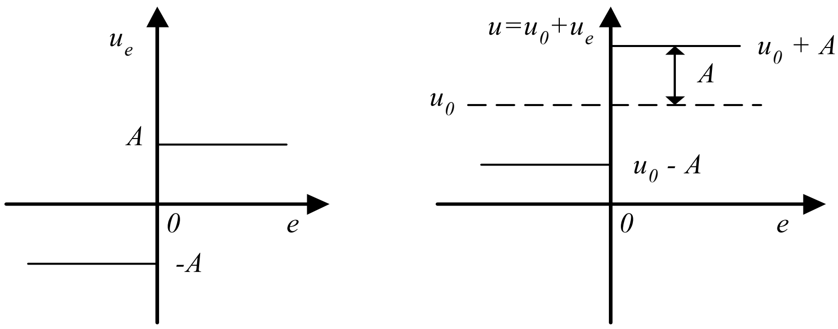

Figure 2.14: The total control signal u and the error-based term u_e in the On/off-controller

$u_0 + A$, which causes the process output variable y and consequently the process measurement y_m to increase. When y_m has increased so much that it becomes greater that the setpoint, the error changes sign, and the control signal becomes $u_0 - A$, which causes the process output variable to decrease, and then the error eventually is positive and the control variable becomes $u_0 + A$, and so on. Thus, there are oscillations.

The oscillations in u are in the form of a square wave. If the actuator is a mechanical device, e.g. a feed screw or a valve, the stepwise movements may cause wear. But if the actuator is an electronic device, e.g. an electronically controlled heating element, there will not be any wear problems. The oscillations in the process output variable y becomes sinusoidal for most processes, but for processes having only integrator dynamics (as a tank), the oscillations are triangular, as we will see in Example 2.8 (below).

Example 2.8 ***On/off-control of chip level of a wood-chip tank***

Figure 2.15 shows the front panel of a simulator for the wood-chip tank.[6] (The front panel also shows parameters of a PID controller, but this controller is not used in this example.) Here is some information about the simulation: The amplitude A is 20%. The initial level is 10 m. The setpoint is initially 10 m and is increased to 12 m at approximately $t = 60$ min. The chip outflow w_{out} (the disturbance) is initially 1500 kg/min and is increased to 1800 kg/min at approx. 120 min. The nominal control

[6] The simulator, which is implemented in LabVIEW, is based on a numeric solution of the differential equation (2.20) which expresses the mass balance of the tank.

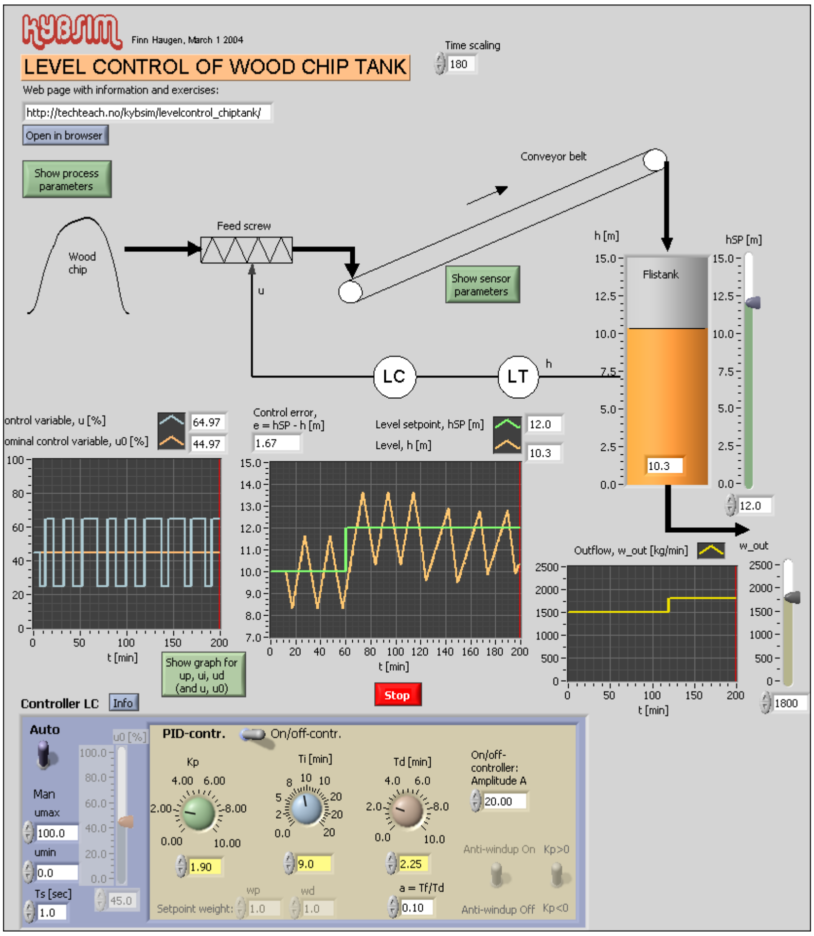

Figure 2.15: Example 2.8: Level control of the wood-chip tank with an On/off-controller. (The front panel shows also PID-parameters, but they are irrelevant in this simulation.)

signal u_0 is 45%, which is calculates according to (2.22) where $w_{out_s} = 1500$ kg/min and $K_s = 33.36$ (kg/min)/%.

The simulation shows the following:

- The control variable oscillates as a square wave. It is symmetric about the nominal control value u_0 (45%). The oscillations become asymmetric if u_0 is no longer correctly tuned (after $t = 120$ min.).
- The oscillations in the level are triangular (not sinusoidal as they may be for many other processes), which is due to the integrator dynamic of the tank (the time-integral of a piecewise constant inflow

is a piecewise ramp).

- The level oscillates about the setpoint with a mean value equal to the setpoint as long as the nominal control signal u_0 has correct value.
- The level oscillates about the setpoint with a mean value which is *not* equal to the setpoint and with an asymmetric form when u_0 has an incorrect value, which is the case after the disturbance was changed from 1500 til 1800 kg/min.

[End of Example 2.8]

In the following sections you will see that the controller can perform far better (without oscillations and, for some of the controllers, with zero static control error) if the error-based term u_e in the control variable (2.16) is calculated using a "softer" and more dynamic function than the abrupt On/off function.

2.6.4 Overview: The PID controller

In the following sections, the P, PI and PID controllers are described. Typically, if you need the only a P- or PI control function, these are achieved as special cases of the PID controller. Therefore, although the PID controller is described in detail later, in Section 2.6.7, it is proper to present the (ideal) PID controller now:

$$u = u_0 + \underbrace{K_p e}_{u_p} + \underbrace{\frac{K_p}{T_i}\int_0^t e\, d\tau}_{u_i} + \underbrace{K_p T_d \frac{de}{dt}}_{u_d} \tag{2.25}$$

The controller parameters are as follows: K_p is the proportional gain. T_i [s] or [min] is the integral time. T_d [s] or [min] is the derivative time. Furthermore, u_0 is the nominal value of the control variable. u_p is the P-term. u_i is the I-term. u_d is the D-term.

From the PID controller (2.25), the P controller and the PI controller can be found from the PID controller as follows:

- A P controller is achieved by setting $T_i = \infty$ (or to a very large value) and $T_d = 0$.[7]

[7] In some commercial controllers you can set T_i to 0 (zero), which is a code expressing that the integral term is de-activated.

- A PI controller is achieved by setting $T_d = 0$.

2.6.5 P controller

The *P controller* (proportional) calculates the control variable according to (2.16) as follows:

$$u = u_0 + \underbrace{K_p e}_{u_p} \tag{2.26}$$

where u_p is the *P-term*. The nominal control variable term u_0 can be found experimentally or by model-based calculations as explained in Section 2.6.2. K_p is the *proportional gain*. Figure 2.16 illustrates the controller function (2.26).

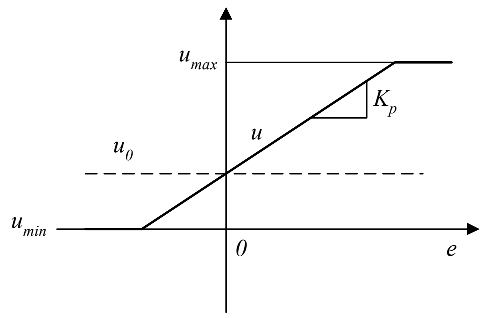

Figure 2.16: P controller as given by (2.26)

In some commercial controllers the *proportional band* P_B, is used in stead of the proportional gain. The proportional band is given by

$$P_B = \frac{100\%}{K_p} \tag{2.27}$$

where K_p is the gain, which here is assumed to be dimensionless. (It will be dimensionless if the control error e and the control variable u have the same unit, typically percent). It is typical that P_B has a value in the range of $10\% \leq P_B \leq 500\%$, which corresponds to K_p being in the range of $0.2 \leq K_p \leq 10$. It is important to note that P_B is inverse proportional to K_p. Thus, a small P_B corresponds to a large K_p, and vice versa.

What does *proportional band* actually mean? One explanation is that P_B is the size of the control error interval Δe (or the size of the measurement signal interval) which gives a control signal interval Δu equal to 100%: From (2.26) we see that $\Delta e = \Delta u/K_p = 100\%/K_p = P_B$.

How does the P controller *work*? Let us look at the wood-chip tank, cf. Example 2.3 (page 19). The P controller changes the control signal proportionally to the error. Assume that the level is less than the setpoint. Then, the control error e is positive, and consequently the controller calculates a control signal change $K_p e$ which is positive, which again increases the chip inflow, so that that the level increases and the error is reduced.

Although the P controller increases the control signal if the control error increases, it will in practice not achieve zero error: Assume that the nominal control value u_0 *does not have a correct value*, so that e is different from zero (the process not in the specified operation point). In this situation, the P controller can not bring e to zero, since if it could, e would be zero, and e was assumed to be different from zero. In other words: *As long as u_0 does not have the correct value (and this is, strictly, always the case), the static control error is different from zero with a P controller.*

The static control error which exists with a P controller, can be reduced by increasing the controller gain K_p, since an increased K_p gives more control variable adjustment, $K_p e$, for a given error e, and this again gives less error. The drawback of increasing K_p is that the control loop gets *reduced stability*, and if K_p becomes too large, the control loop becomes unstable. The stability of control loops is discussed further in Section 2.11 and in Chapter 6.4.

Example 2.9 ***P control of the level of a wood-chip tank***

Figure 2.17 shows simulated responses for the level control system for wood-chip tank described in Example 2.3 (page 19). The front panel of the simulator is as shown in Figure 2.15 (page 32). The controller gain, tuned with the Ziegler-Nichols' closed loop tuning method, cf. Section 4, is

$$K_p = 1.55 \tag{2.28}$$

The initial level is 10m. The setpoint is initially 10m and is increased to 12m at approx. $t = 10$min. The chip outflow w_{out} (which is a disturbance to the control system) is initially 1500kg/min and is increased to 1800kg/min after approx. 60min. The nominal control signal u_0 is 45%, which is calculated from (2.22) where $w_{out_s} = 1500$kg/min and $K_s = 33.36$ (kg/min)/%. The simulation shows the following:

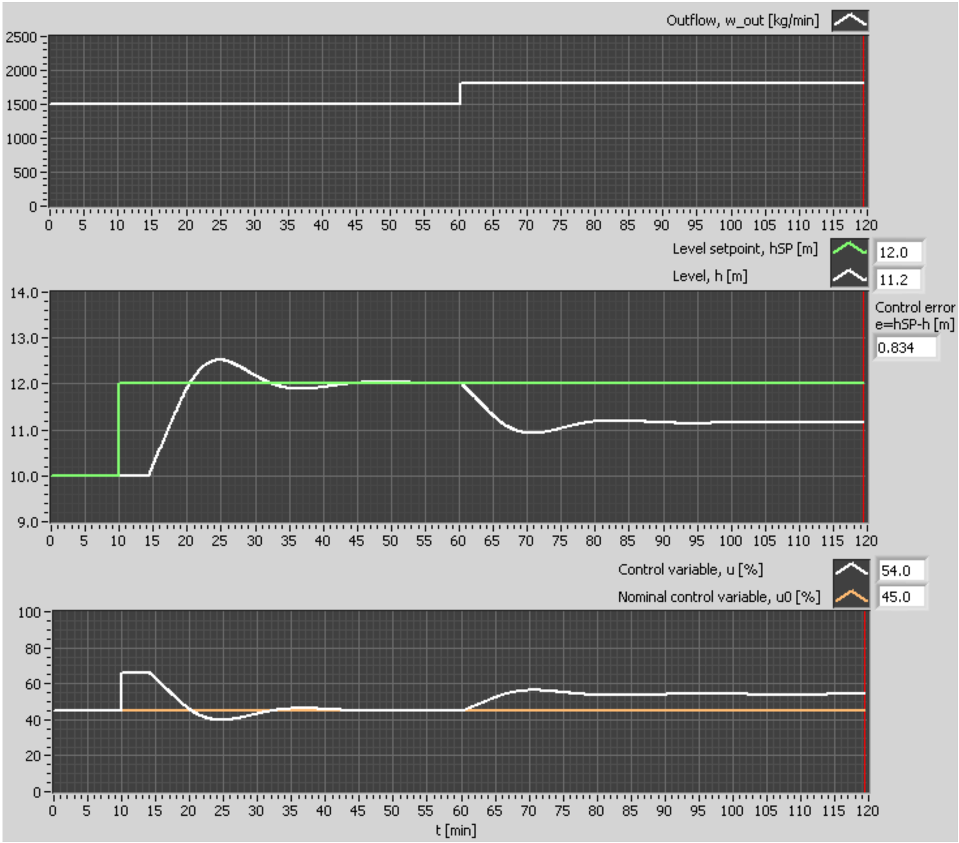

Figure 2.17: Example 2.9: Level control of the wood-chip tank with a P controller. (The front panel of the simulator is as shown in Figure 2.15 on page 2.15.)

- Tracking properties: The steady-state error e_s is zero only as long as the nominal control variable value u_0 has a correct value, which is the case for $t < 60$min.
- Compensation properties: e_s is *different from zero* due to the step in the disturbance w_{out} which comes at $t = 60$min. In Figure 2.15 e_s has value 0.83 at $t = 120$min which is the time when the simulation is stopped. However, if the simulations were run longer (the responses has not converged completely at $t = 120$s) we would have seen that the error actually settles at

$$e_s = 0.87\text{m} \tag{2.29}$$

Finally, what happens if we *increase* K_p? We should expect that e_s is reduced since increasing K_p in the compensation term $u_p = K_p e$ forces e to become smaller. Figure 2.18 shows simulated responses with $K_p = 2.6$. The *steady-state control error will be reduced* from 0.87m to 0.52m, and *the stability of the control loop is reduced.*

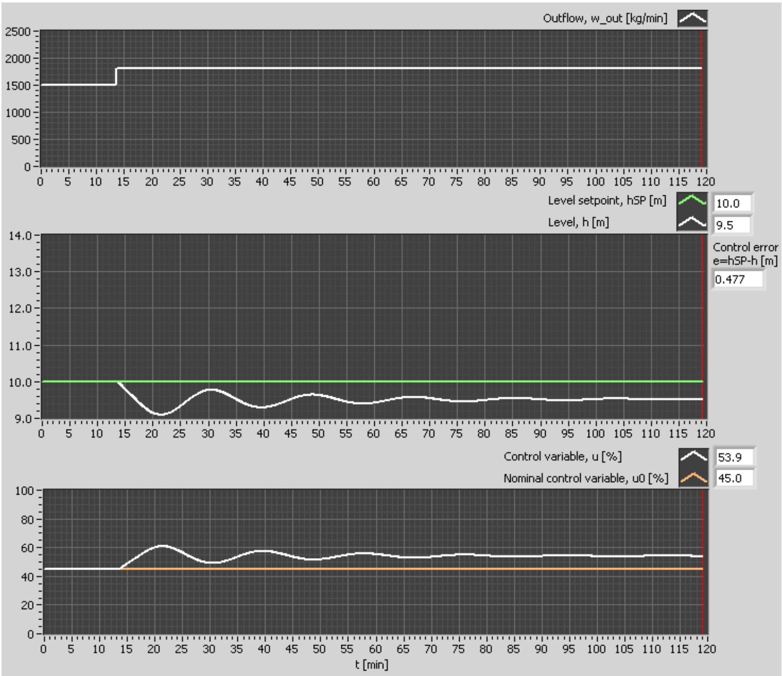

Figure 2.18: Example 2.9: Level control of the wood-chip tank with a P controller with an increased gain of $K_p = 2.6$. (The front panel of the simulator is as shown in Figure 2.15.)

[End of Example 2.9]

2.6.6 PI controller

Example 2.9 demonstrated a problem with the P controller: The steady-state control error e_s becomes different from zero when the nominal control signal u_0 does not have correct value. In practice, u_0 does never have a completely correct value since there are always unknown disturbances acting on the process to be controlled. If the P controller is substituted by a PI controller, $e_s = 0$ can be achieved, for any value of u_0!

In the PI controller (proportional + integral) the control variable is calculated as

$$u = u_0 + \overbrace{\underbrace{K_p e}_{u_p} + \underbrace{\frac{K_p}{T_i}\int_0^t e\, d\tau}_{u_i}}^{u_e} \tag{2.30}$$

Here, u_i is the integral term, K_p is the proportional gain. T_i [s] or [min] is the integral time (also denoted the reset time). In some commercial controllers the fraction K_p/T_i is represented by the *integral gain* K_i:

$$K_i = \frac{K_p}{T_i} \tag{2.31}$$

In some controllers the value of $1/T_i$ is used in stead of the value of T_i. The unit of $1/T_i$ is *repeats per minute*. For example, 5 repeats per minute means that T_i is equal to $1/5 = 0.2$min. The background of the term repeats per minute is as follows: Assume that the control error e is constant, say E. The P-term has value $u_p = K_p E$. During a time interval of 1 minute the I-term equals $\frac{K_p}{T_i}\int_0^1 E\, d\tau = K_p E \cdot 1[\text{min}]/T_i = u_p \cdot 1/T_i$. Thus, the I-term has repeated the P-term $1/T_i$ times.

How does the PI controller work? The integral term is essential. u_i as calculated as *the time integral* of the control error e from as initial time, say $t = 0$, to the present point of time, thus the integral is being calculated continuously. Let us think about the level control of the wood-chip tank: Assume that e initially is greater than zero (the level is then less than the level setpoint). As long as e is positive, u_i and therefore the total control variable u will get steadily increasing value, since the time integral of a positive value increases with time. The increasing u gives an increasing wood-chip inflow. Consequently, the chip level in the tank increases. Due to this level increase the control error eventually becomes less positive. The increase of the integral term (the inflow) continues until the error has become zero. The conclusion is that the *integral term ensures zero steady-state control error*. The zero steady-state control error is achieved even if the nominal control signal u_0 has an incorrect value (even if the value is zero).

The potential of achieving zero static control error is why the PI controller, besides the PID controller which also contains an integral term, is the most commonly used control function in the industry.

Example 2.10 ***PI control of chip level of a wood-chip tank***

Figure 2.19 shows simulated responses for the level control system for the wood-chip tank described in Example 2.3 (page 19). The front panel of the

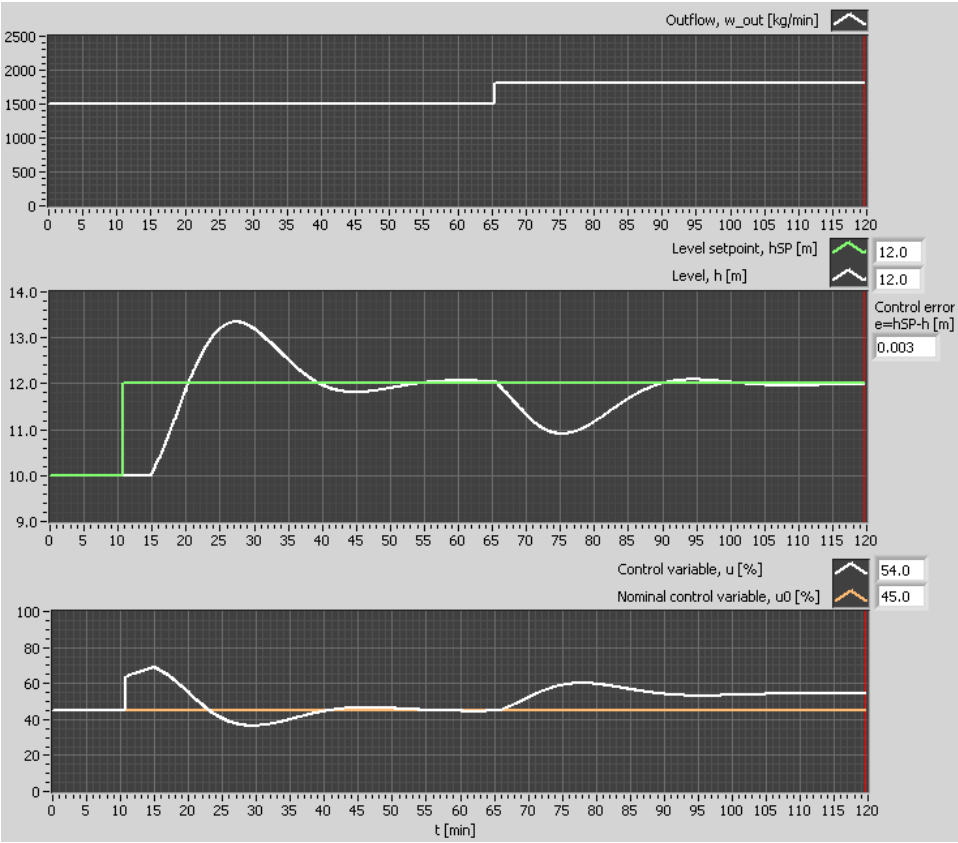

Figure 2.19: Example 2.10: Level control of the wood-chip tank with a P-controller. (The front panel of the simulator is as shown in Figure 2.15 (page 2.3).)

simulator is as shown in Figure 2.15. The controller parameters have values

$$K_p = 1.40;\ T_i = 900\text{s} = 15.0\text{min} \tag{2.32}$$

(found using the Ziegler-Nichols' closed loop-method, cf. Section 4.4). The simulation shows that the steady-state control error is zero both due to a step in the level setpoint and due to a step in the disturbance (chip outflow).

[End of Example 2.10]

2.6.7 PID controller

We may be quite content with the PI controller since it gives zero steady-state control error. But in some cases it would be desirable to have *faster control* than with the PI controller. This can be achieved by including a term in the control variable that is *proportional to the time derivative* or the rate of change of the error e. Then we have the *PID controller* (proportional + integral + derivative)

$$u = u_0 + \overbrace{\underbrace{K_p e}_{u_p} + \underbrace{\frac{K_p}{T_i}\int_0^t e\,d\tau}_{u_i} + \underbrace{K_p T_d \frac{de}{dt}}_{u_d}}^{u_e} \tag{2.33}$$

u_d is the derivative term. K_p is the proportional gain. T_i [s] or [min] is the integral time . T_d [s] or [min] is the derivative time. In some commercial controllers the product $K_p T_d$ is represented by the *derivative gain* K_d:

$$K_d = K_p T_d \tag{2.34}$$

The derivative term of the PID controller works as follows: Assume that the control error e is increasing. Then the time derivative de/dt is positive, and the derivative contributes with a positive value to the total control signal u. This will in general give *faster control*.

All commercial controllers implements a PID controller. But none of them implements (2.33)! It is namely an *ideal PID controller*, and its D-term must be modified, otherwise the controller will not work properly in practical applications (we will return to this practical aspect later in this section).

Example 2.11 ***PID control of chip level of a wood-chip tank***

Figure 2.20 shows simulated responses for the level control system for wood-chip tank described in Example 2.3. The PID parameters have values

$$K_p = 1.86;\ T_i = 540\text{s} = 9.0\text{min};\ T_d = 135\text{s} = 2.25\text{min} \tag{2.35}$$

(found using the Ziegler-Nichols' closed loop-method, cf. Section 4.4). The simulation shows the following:

- The steady-state or static control error is zero both after a step in the level setpoint and after a step in the disturbance (chip outflow).

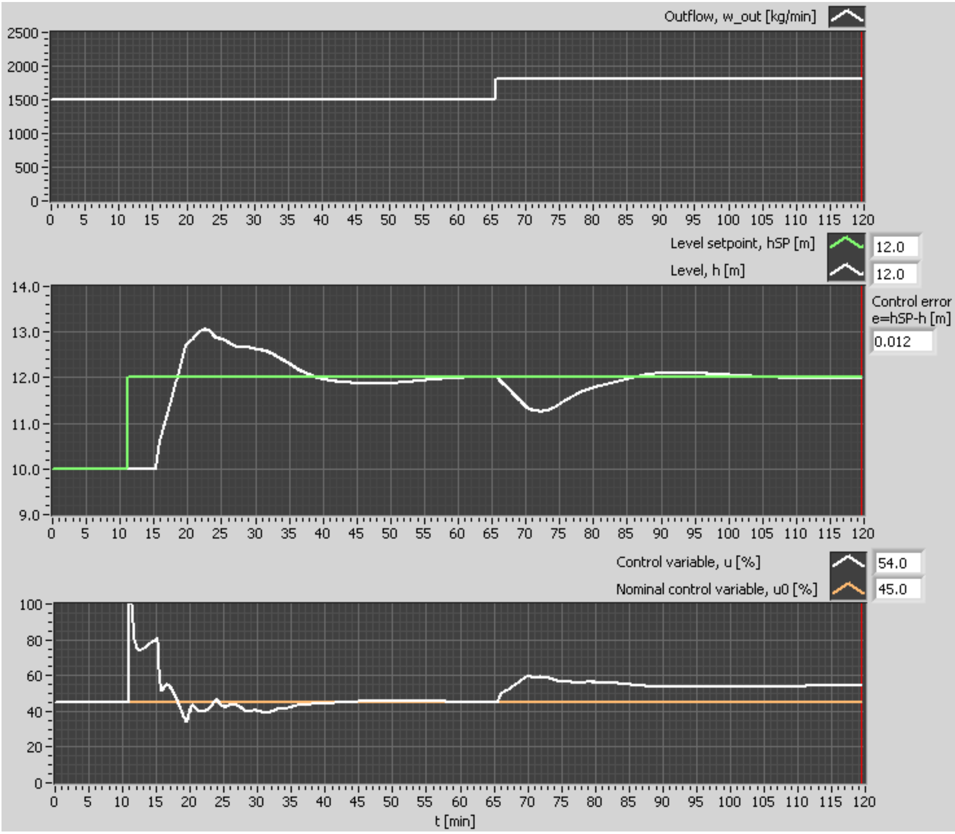

Figure 2.20: Example 2.11: Level control of the wood-chip tank with a P-controller. (The front panel of the simulator is as shown in Figure 2.15.)

- The setpoint tracking is faster than with a PI controller, cf. Example 2.10. Also, the compensation of the step change of the disturbance (outflow) is faster than with a PI controller.

[End of Example 2.11]

Behaviour of the PID controller terms

Do you want to see how each of the terms of a PID controller function works? Figure 2.21 shows the time-responses due to a step in the outflow w_{out} (the disturbance) from the initial value of 1500kg/min to 1800kg/min. The controller parameters have values as in Example 2.11. The nominal control signal u_0 has value 45% which gives an inflow that is equal to the initial value of w_{out}. We can observe the following:

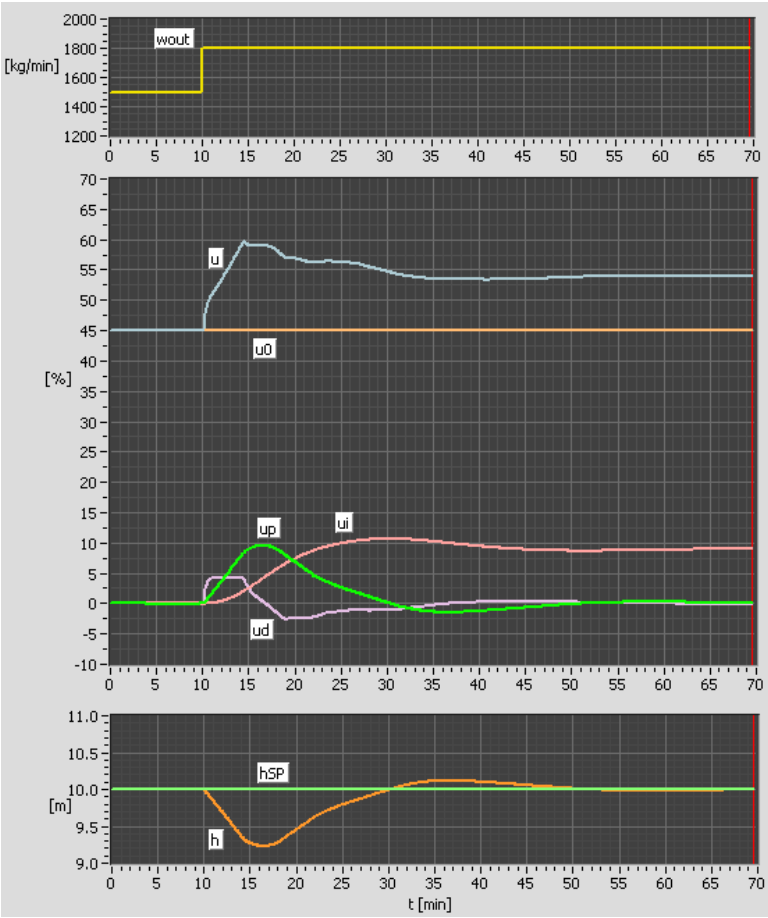

Figure 2.21: Behaviour of the various terms of the PID-controller terms after a step in the outflow w_{out} (disturbance)

- The D-term u_d reacts abruptly and its value converges towards zero (the time-derivative of a constant error is constant).

- The I-term is relatively sluggish. Its value changes as long as the control error is different from zero. u_i goes to a (new) constant value after the step in w_{out}. The change in u_i constitutes the compensation for the incorrect value in u_0 after the change in w_{out} (the disturbance).

- The P-term u_p is quicker than the I-term, but more sluggish than the D-term, and its value goes to zero since $K_p e$ goes to zero when e goes to zero.

Measurement noise and lowpass filter in the D-term

There is a potential problem using the PID controller: It may give a very unsteady high frequent control signal due to noise in the process measurement – and such noise is always present, more or less. See Figure 2.22. The measurement noise can stem from electronic noise sources or

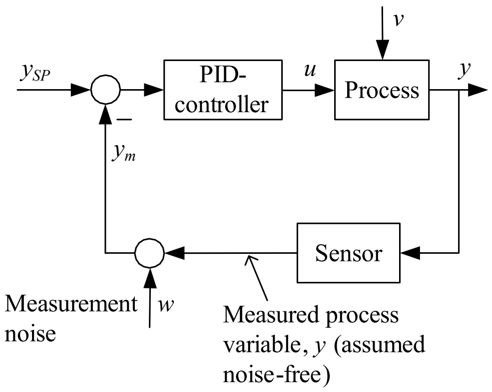

Figure 2.22: Measurement noise in the control loop

from the measurement principle, e.g. ultrasound based level measurement of a liquid surface.

The unsteady control signal is due to the differentiation of the control error e in the D-term. The control error consist of the following terms:

$$e = y_{SP} - y_m = y_{SP} - (y + w) \tag{2.36}$$

where y_{SP} is the setpoint, y_m is the process measurement, y is the (noise-free) process variable, and w is the measurement noise. The D-term becomes

$$\begin{aligned} u_d &= K_p T_d \frac{d(y_{SP} - y_m)}{dt} && (2.37) \\ &= K_p T_d \frac{d\left[y_{SP} - (y + w)\right]}{dt} && (2.38) \\ &= \underbrace{K_p T_d \frac{d(y_{SP} - y)}{dt}}_{\frac{de}{dt}} - K_p T_d \frac{dw}{dt} && (2.39) \end{aligned}$$

The term dw/dt is the time-derivative of the noise and it is a term in the control variable. If the noise w is high frequent, its time-derivative (rate of change) may get very large values, and the control variable u may be very unsteady. We can see this by assuming that w is sinusoidal:

$$w(t) = W\sin(\omega t) \tag{2.40}$$

From this we get

$$\frac{dw}{dt} = \underbrace{\omega W}_{A_w}\cos(\omega t) \tag{2.41}$$

If the frequency ω is large, the amplitude $A_w = \omega W$ of the time-derivative dw/dt may be large, and consequently the term $-K_pT_d dw/dt$ in the D-term u_d may get a large amplitude.

How can we reduce the problem of the time-differentiation of the measurement noise? If we can not reduce or remove the noise itself, we can *lowpass filter* the control error used in the D-term before it is differentiated. This is a standard solution used in commercial controllers. Let us use the symbol e_f for the filtered error. The modified PID controller is then

$$u = u_0 + \underbrace{K_p e}_{u_p} + \underbrace{\frac{K_p}{T_i}\int_0^t e\,d\tau}_{u_i} + \underbrace{K_p T_d \frac{de_f}{dt}}_{u_d} \tag{2.42}$$

The filter is typically a first order lowpass filter. It is convenient to represent the filter with its Laplace transfer function. The relation between e_f and e is then[8]

$$e_f(s) = \frac{1}{T_f s + 1} e(s) \tag{2.43}$$

where T_f is the filter time constant which usually is expressed as a fraction of the derivative time T_d:

$$T_f = aT_d \tag{2.44}$$

a is a constant which typically is chosen between 0.05 and 0.2. If no special requirements exist, we can set $a = 0.1$.

Figure 2.23 shows simulations of a control system (not the wood-chip tank this time). The setpoint y_{SP} and the process measurement y_m are shown in one diagram, and the control variable u is shown in the other diagram. The controller is a PID controller where K_p and T_i have constant values. The setpoint is constant. The measurement contains random measurement noise w (uniformly distributed between $\pm 0.2\%$). The simulation shows three situations:

- Between $t = 120$ and 140s: No D-term, that is, the controller is a PI controller ($T_d = 0$ in the PID controller). The simulation shows

[8] Although it is not mathematically correct, it is convenient to use the same symbol for the time function and the Laplace transform of the function.

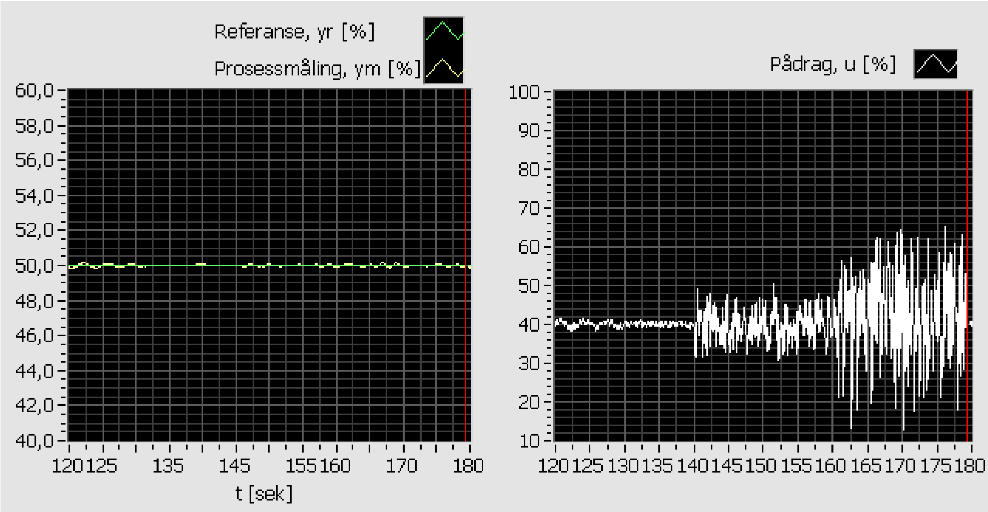

Figure 2.23: Simulation of a PID-control system with measurement noise for three different situations, cf. the text

naturally enough some noise in the control signal u. The noise propagates to the control variable mainly via the P-term, but also somewhat via the I-term.

- Between $t = 140$ and 160s: Ordinary PID controller with lowpass filter with $a = 0.1$. The noise gives a larger response in the control variable than with the PI controller due to the noise sensitivity in the D-term. This demonstrates that *the PID controller gives more noisy control signal than PI controller.*

- Between $t = 160$ and 180s: PID controller with an (approximately) ideal D-term, that is, the lowpass filter in the D-term is (approximately) removed. The response of the noise in the control signal is very noisy . This demonstrates that *the lowpass filter in the D-term is important for attenuating the response of the measurement noise in the control variable.*

If the measurement noise has a mean value m_w different from zero, there will be a steady-state control error different from zero, since m_w will appear as an addition to the setpoint. The PID controller ensures that the process output variable y will track this false setpoint (containing the m_w term).

Above, the solution to the measurement noise was to lowpass filter the control error used in the D-term. If this does not give enough filtering, we can try to use a separate measurement filter acting on the measurement signal. This is described in Section 2.7.3.

Block diagram of the PID controller

Figure 2.24 shows a block diagram of the PID controller given by (2.42).

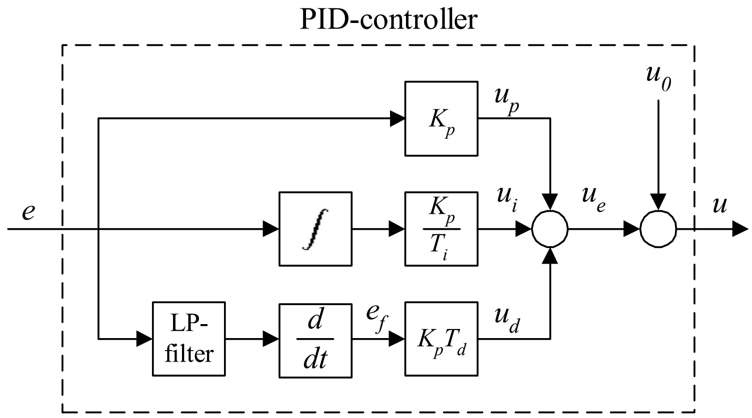

Figure 2.24: Block diagram of the PID-controller (2.42)

Transfer function of the PID controller

In some situations it is useful to represent the PID controller (2.42) by its transfer function. This is the case in frequency response analysis control systems, analytical calculation of time-responses using the Laplace transform, and simulations when it is sufficient to use a compact, linear controller model.

It is quite easy to find the controller transfer function $H_c(s)$ from input e to output u by taking the Laplace transform of (2.42) combined with (2.43) and neglecting u_0 (the controller transfer function is independent of u_0):

$$
\begin{aligned}
u(s) &= K_p e(s) + \frac{K_p}{T_i}\frac{1}{s}e(s) + K_p T_d s e_f(s) && (2.45)\\
&= \underbrace{\left[K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{T_f s + 1}\right]}_{H_c(s)} e(s) && (2.46)\\
&= \underbrace{\frac{K_p T_i \left(T_f + T_d\right) s^2 + K_p \left(T_i + T_f\right) s + K_p}{T_i T_f s^2 + T_i s}}_{H_c(s)} e(s) && (2.47)
\end{aligned}
$$

PID controller on serial form

The PID controller given by (2.33) is said to be on *parallel form* since a block diagram of the controller function shows the P, I and D term in parallel paths, cf. Figure 2.24. There is also a *serial form.* In most cases it is not an important difference between the parallel form and the serial form. The the serial form consists of a PD controller (which is a PID controller with the I-term removed) in series with a PI controller, and with the gain of the PD and the PI controllers combined in the common gain K_p:

$$
\begin{aligned}
u(s) &= \underbrace{K_{p_i}\left(1 + \frac{1}{T_i s}\right)}_{\text{PI}} \cdot \underbrace{K_{p_d}\frac{T_d s + 1}{T_f s + 1}}_{\text{PD}} e(s) && (2.48)\\
&= K_p \frac{\left(T_i s + 1\right)\left(T_d s + 1\right)}{T_i s \left(T_f s + 1\right)} e(s) && (2.49)\\
&= \underbrace{\frac{K_p T_i T_d s^2 + K_p \left(T_i + T_d\right) s + K_p}{T_i T_f s^2 + T_i s}}_{H_s(s)} e(s) && (2.50)
\end{aligned}
$$

A few comments:

- The serial form is more practical than the parallel form in frequency response based controller design, cf. Chapter 8, due to the factorized form of (2.49).
- The parallel form is more general since it can have complex zeros in its transfer function (the serial form can only have real zeros).
- The parallel form is somewhat easier to express in the time domain as a differential/integral equation) and to realize as a practical discrete-time algorithm.

- According to [24] the serial form is more frequently used in modern commercial controllers – probably because the serial form behaves similar to the first industrial PID controllers which were pneumatic and was used in the 1930s.[9]

Transformation from serial to parallel form

You can perform a transformation from serial form to parallel form (the reverse transformation is less used). One reason for performing such a transformation is that you used tuning methods which assumes the serial form, while the controller you use, actually implements the parallel form. The transformation can be executed as follows[24] (it is based on comparing coefficients between the ideal PID controller functions, that is, with T_f set to 0): Given the parameters K_{p_s}, T_{i_s} and T_{d_s} of the serial form PID controller. The corresponding parameters, K_{p_p}, T_{i_p}, T_{d_p} and T_{f_p} of a parallel PID controller having approximately the same behaviour as the serial form, is achieved with the following transformations:

$$K_{p_p} = K_{p_s}\left(1 + \frac{T_{d_s}}{T_{i_s}}\right) \tag{2.51}$$

$$T_{i_p} = T_{i_s}\left(1 + \frac{T_{d_s}}{T_{i_s}}\right) \tag{2.52}$$

$$T_{d_p} = T_{d_s}\frac{1}{1 + \frac{T_{d_s}}{T_{i_s}}} \tag{2.53}$$

In addition, the time constant of the lowpass filter in the derivative term can be calculated by

$$T_{f_p} = aT_{d_p} \tag{2.54}$$

where a typically is 0.1. For P and PI controllers the serial and the parallel forms are identical (since T_{d_s} is 0).

From (2.51)-(2.53) we see that the transformations are functions of the ratio T_{d_s}/T_{i_s}. The less T_{d_s}/T_{i_s}, the less importance of the transformations. In the Ziegler-Nichols' tuning methods, cf. Chapter 4,

$$\frac{T_{d_s}}{T_{i_s}} = \frac{1}{4} \tag{2.55}$$

If this relation is used in (2.51)–(2.53),

$$K_{p_p} = 1.25K_{p_s} \tag{2.56}$$

[9] The famous Ziegler-Nichols' methods for controller tuning, cf. Chapter 4, were published in 1942 and they must have been based on pneumatic controllers approximately implementing the serial form.

$$T_{i_p} = 1.25T_{i_s} \tag{2.57}$$

$$T_{dp} = 0.8T_{d_s} \tag{2.58}$$

In this case the parameter transformations do not change the PID parameters much, and you can quite safely assume that the two PID controllers behave approximately equally, which implies that you not need to care about which PID form which is actually implemented in the controller. But if you feel a little uncertain about the different implementations, you should still consider to use the transformations. You can use simulations to check if the parallel and serial form causes any substantial difference in the behaviour of the control systems.

Example 2.12 ***Parallel and serial form of the PID controller***

In this example control systems for processes having the following transfer function model are simulated:

$$y(s) = \frac{K_u}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} u(s) \tag{2.59}$$

$$+ \frac{K_v}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} v(s) \tag{2.60}$$

(The process model is thus a second order system with time delay.) The process parameter are

$$K_u = 1;\ K_v = 2;\ T_1 = 1\text{s};\ T_2 = 1\text{s};\ \tau = 0.5\text{s}; \tag{2.61}$$

For comparison two control system are simulated simultaneously: One with a parallel PID controller and one with a serial PID controller. The process to be controlled, and the setpoint and the disturbance are identical for both control systems. The two control systems are simulated in two scenarios:

1. *Without using PID parameter transformation*: The following PID parameters (found using the Ziegler-Nichols' closed loop method) are used for both the parallel PID controller and the serial PID controller:

$$K_p = 3.6;\ T_i = 2.0\text{s};\ T_d = 0.5\text{s}; \tag{2.62}$$

   Figure 2.25 shows the simulated responses due to a setpoint step (at $t = 4$s) and a disturbance step (at $t = 20$s). The simulations shows that the responses in the two control systems are somewhat but not dramatically different. The stability is satisfactory in both systems.

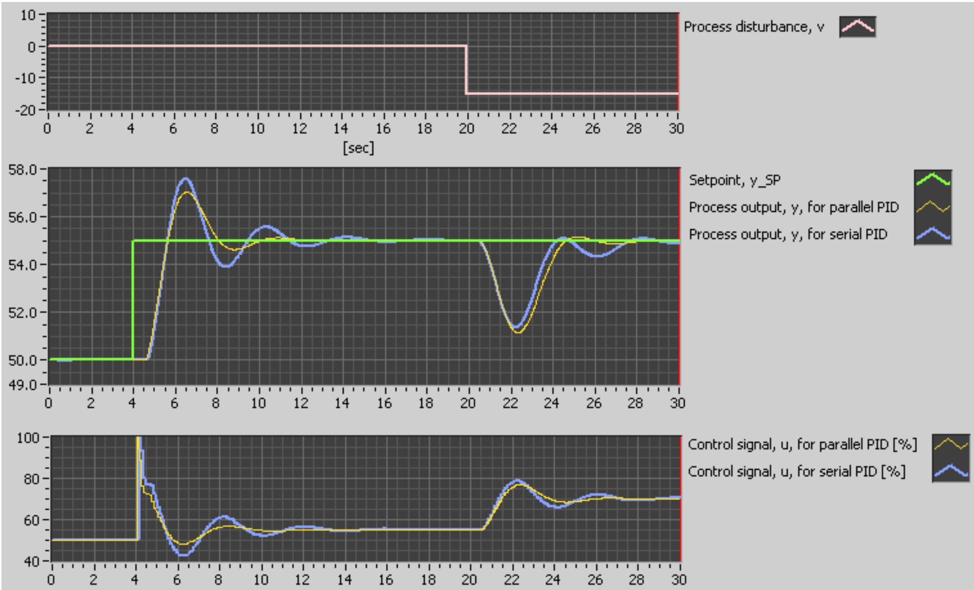

Figure 2.25: Example 2.12: Simulated responses due to a setpoint step and a disturbance step. PID parameter transformation is *not* used.

2. *Using PID parameter transformation*: The PID parameters (2.62) are still used for the serial PID controller, but for the parallel PID controller the following parameters are used:

$$K_p = 4.5;\ T_i = 2.5\text{s};\ T_d = 0.42\text{s}; \tag{2.63}$$

These parameter values are found by transforming the serial form parameters (2.62) to parallel form parameters using (2.51)-(2.53). The parallel PID controller and the serial PID controller should then have the same behaviour. Figure 2.26 shows the simulated responses due to a setpoint step and a disturbance step. The responses in the two control systems are now almost identical. The difference is probably due to simulation technicalities, and possibly due to the fact that the parameter transformations are not ideal since they do not take the derivative filter time constant T_f into account.

[End of Example 2.12]

This above example indicates that it is probably not important to distinguish between the serial PID controller and the parallel PID controller. This conclusion is true if the Ziegler-Nichols' closed loop tuning method is used, in which $T_d/T_i = 1/4$. If a different (larger) ratio between

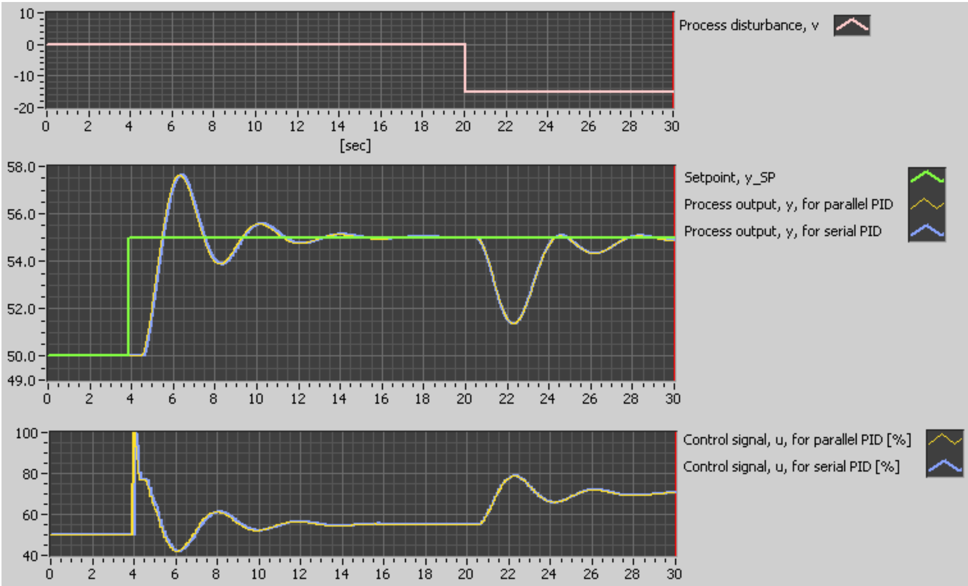

Figure 2.26: Example 2.12: Simulated responses due to a setpoint step and a disturbance step. PID parameter transformation is used.

T_i and T_d is used, the difference between the controllers may be larger and hence the PID parameter transformations more important.

2.6.8 Positive or negative controller gain?

On commercial controllers can you choose whether the controller gain K_p of the PID controller has *positive or negative value*. Let us write the PID controller (2.42) as

$$u = u_0 + \underbrace{K_{sign} \cdot K_{p_1}}_{K_p} \left(e + \frac{1}{T_i} \int_0^t e \, d\tau + T_d \frac{de_f}{dt} \right) \tag{2.64}$$

The controller gain, which is $K_p = K_{sign} K_{p_1}$ where K_{p_1} is always positive, will have positive sign with $K_{sign} = 1$ and negative sign with $K_{sign} = -1$. On commercial controllers the user typically sets the value of K_{p_1}, while the sign, here K_{sign}, is set via a parameter field or a button. The default choice is positive gain ($K_{sign} = 1$).

The consequence of choosing *wrong* sign of K_p is dramatic: The control loop becomes *unstable*. Instability implies that the variables in the control loop shows a steadily increasing amplitude, until some saturation occurs.

How do you know which controller gain sign to use? It is the sign of the *process gain* K which determines the controller gain sign, as follows:

- If the process gain K is positive, K_p shall be positive, that is $K_{sign} = 1$. The controller is in this case said to have *reverse action*, since an increase of the process output variable gives a reduction of the control signal.
- If the process gain K is negative, K_p shall be negative, that is $K_{sign} = -1$. The controller is in this case said to have *direct action*.

Above, the term "process" includes all subsystems in the control loop except the controller. Consequently, the "process" also includes the sensor.

The point is that the total gain of the loop shall be positive whatever the sign of the process is (in this context we disregard the negative gain of the subtraction between the setpoint and the process measurement). This is achieved by requiring that $K_p \cdot K_{sign}$ is positive, cf. Figure 2.27.[10]

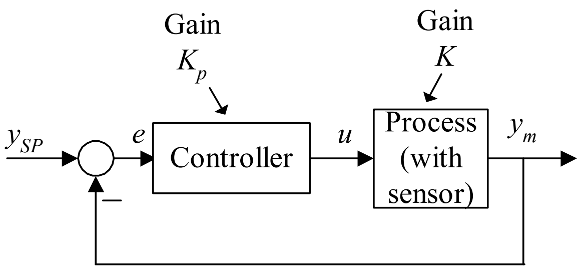

Figure 2.27: To have a stable control loop the product $K_p \cdot K_s$ must be positive.

What is the process gain, K? Simply stated, the gain of a system is the ratio of the output signal (or the rate of change of the output signal if the system has integral dynamics) to the input signal of the system. So, if the output has a positive response to a positive input, the gain is positive. Here are a few examples:

- Figure 2.28 shows a level control system for a liquid tank where the control variable *controls the outflow* of the tank. An increase of the control signal reduces the level and the level measurement (it is

[10] Note that $K_pK_s > 0$ does not ensure stability since the loop will be ustable if K_pK_s has too large positive value. But the loop is certainly unstable if $K_pK_s < 0$.

assumed that the measurement signal decreases as the level decreases). Consequently, the process (including sensor) has a negative process gain.

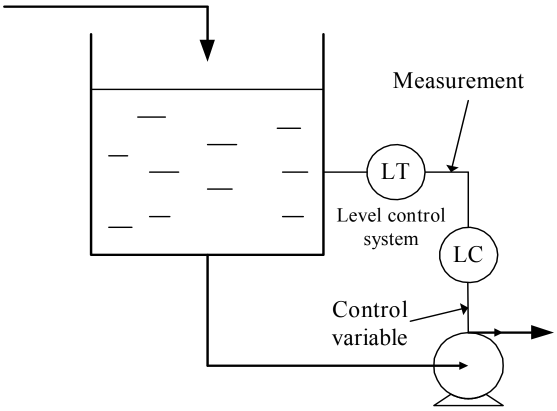

Figure 2.28: An example of a process (with sensor) with negative gain. An increase of the control signal will reduce the level.

- A heat exchanger with temperature control where the control signal *controls the supply of cooling media (e.g. cold water)* has a negative gain since an increase of the control signals increases the cooling and hence decreases the temperature (and the temperature measurement).

The process gain can of course be found from the process model. Here are a few examples:

- The transfer function model (first order with time delay)

$$y_m(s) = \frac{3}{s+1}e^{-2s}u(s) \tag{2.65}$$

  has positive gain, namely 3.

- The transfer function model (integrator with time delay)

$$y_m(s) = \frac{-2}{s}e^{-s}u(s) \tag{2.66}$$

  has negative gain, namely -2.

2.7 Practical problems: Control kicks, windup, and noise

This Section describes several important practical problems which can exist in real control loops, and how to solve these problems.

2.7.1 Reducing P- and D-kick caused by setpoint changes

Introduction

Abrupt changes of the setpoint y_{SP}, for example step changes, may cause unfortunate kicks in the control variable. The problem is connected to the P-term and the D-term of the controller function (2.42). These kicks are denoted proportional kick or P-kick and derivative kick or D-kick, respectively. Such kicks may cause mechanical actuators to move abruptly, resulting in excessive wear.

One solution to the above mentioned problem is to modify the P-term and/or the D-term of the PID controller. Another solution is to accept only smooth setpoint changes! We will study these solutions in detail in the following sections.

For an easy reference the PID controller function with setpoint weights in the D-term and the P-term is repeated here:

$$u = u_0 + \underbrace{K_p e_p}_{u_p} + \underbrace{\frac{K_p}{T_i}\int_0^t e\,d\tau}_{u_i} + \underbrace{K_p T_d \frac{de_{d_f}}{dt}}_{u_d} \tag{2.67}$$

where e_{d_f} is given by

$$e_{d_f}(s) = \frac{1}{T_f s + 1} e_d(s) \tag{2.68}$$

and

$$e_p = w_p y_{SP} - y \tag{2.69}$$

$$e_d = w_d y_{SP} - y \tag{2.70}$$

where w_p and w_d are setpoint weights in the P-term and the D-term, respectively.

Note: Do not try reducing the setpoint weight in the I-term since it will cause the static control error to become different from zero! This is because the integrand becomes zero in steady-state in a stable control system, and if the integrand of the u_i-term is not equal to the difference $y_{SP} - y$, but in stead say $w_i y_{SP} - y$, then of course $y_{SP} - y = e$ can not be equal to zero in steady-state. So, once more: Do not use reduced setpoint weighting in the I-term!

Reduction of D-kick

The derivative term of the PID controller (2.42) is

$$u_d = K_p T_d \frac{de_{d_f}}{dt} = K_p T_d \frac{d\left(w_p y_{SP} - y\right)_f}{dt} \tag{2.71}$$

where index f is for "filtered". Assume initially that $w_d = 1$, that is, no reduced setpoint weight. Due to the time derivative, an abrupt change of the setpoint y_{SP} gives an abrupt change of u_d and a corresponding change of the total control variable u in which u_d is an additive term, cf. (2.67). For example, a stepwise change of the setpoint gives an impulse in u_d since the time derivative of a step is an impulse.

To avoid such changes of u_d, the setpoint y_{SP} can be given a reduced weight in the D-term by giving w_d a value less than 1. In the case of reduced weight, it is common to set $w_d = 0$, causing the setpoint to be removed completely from the derivative term. In many commercial controllers $w_d = 0$ is a fixed factory setting.

One drawback with reduced setpoint weight is more sluggish response to varying setpoint signals. This can be unfortunate in at least the following cases

- Secondary controllers in cascade control systems, cf. Chapter 9.2.
- Servo systems (control systems for motors).

One question is: Will reduced weighting of the setpoint in the D-term influence the ability of the controller to compensate for disturbances? The answer is *no* because the compensation for disturbances takes place after the disturbance has caused a response in the process output variable y and measurement y_m, and the appearance of y_m in the D-term is independent of the setpoint weight.

Example 2.13 ***Reduced setpoint weight in the D-term***

In this example a control system for a process having the following transfer function model is simulated:

$$y(s) = \frac{K_u}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} u(s) \quad (2.72)$$

$$+ \frac{K_v}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} v(s) \quad (2.73)$$

(The process is thus a second order system with time delay.) u is the control variable, and v is the process disturbance. The process parameter are

$$K_u = 1;\ K_v = 1;\ T_1 = 2\text{s};\ T_2 = 1\text{s};\ \tau = 0.5\text{s}; \quad (2.74)$$

The PID parameters are

$$K_p = 3.6;\ T_i = 2.0\text{s};\ T_d = 0.5\text{s}; \quad (2.75)$$

(tuned with the Ziegler-Nichols' closed loop method). Two cases are simulated:

- *Full setpoint weight,* $w_d = 1$: Figure 2.29 shows simulated responses in the control system due to a setpoint and a disturbance step.

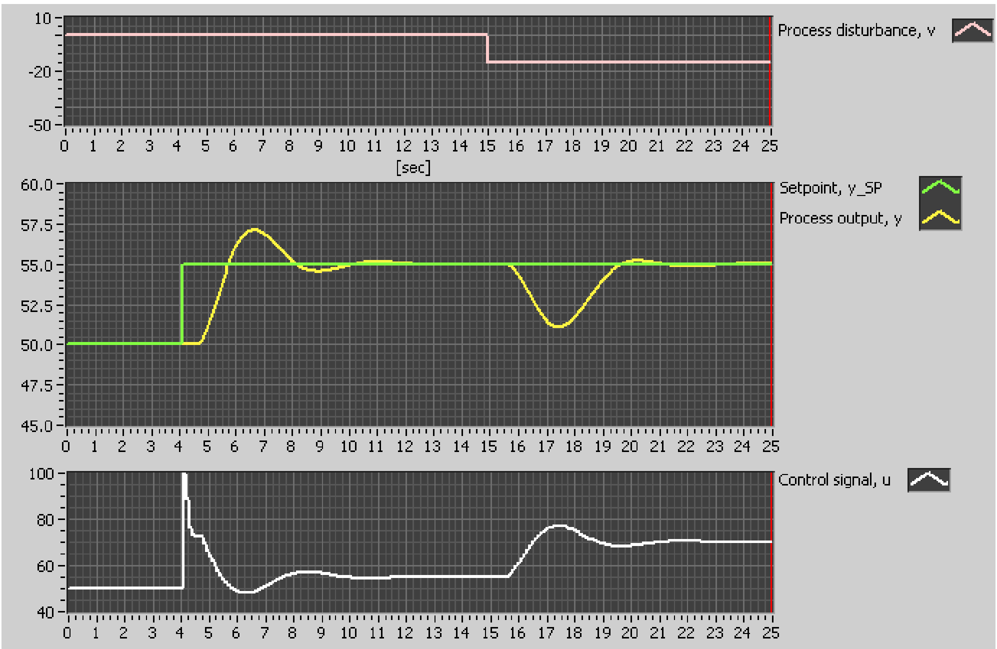

Figure 2.29: Example 2.13: Simulated responses in the control system. There is No reduction of setpoint weight, thus $w_d = 1$.

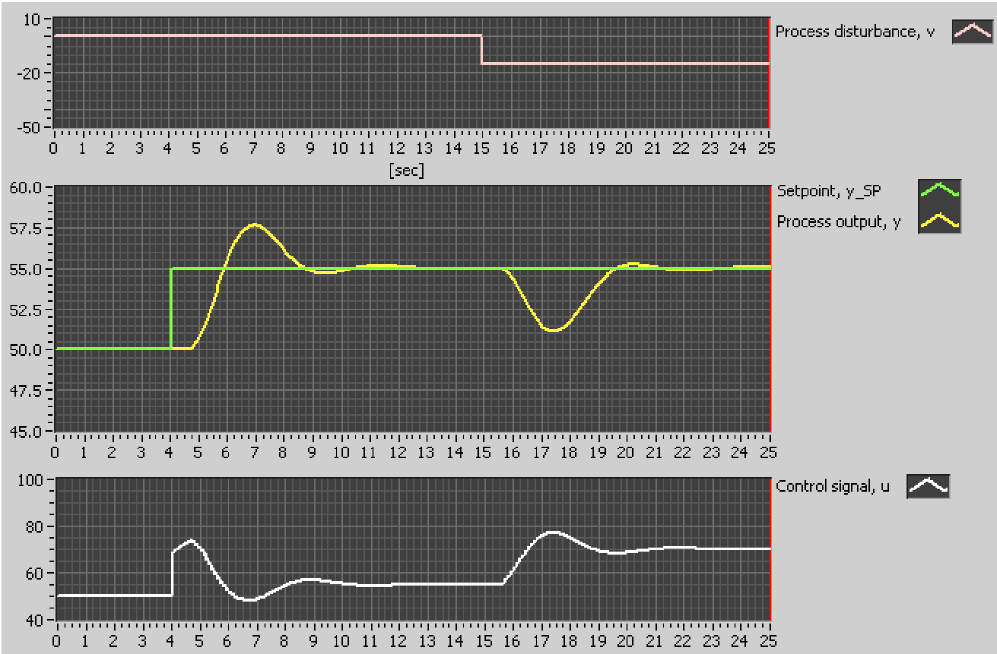

Figure 2.30: Example 2.13: Simulated responses in the control system. The setpoint is removed from the D-term with $w_d = 0$.

- *No setpoint weight,* $w_d = 0$: Figure 2.30 shows simulated responses.

Comparing the responses in Figures 2.29 and 2.30, it is clear that the control signal reacts smoother with $w_d = 0$ than with $w_d = 1$ after the setpoint step. However, there is no difference in the control signals after the disturbance step, as expected.

[End of Example 2.13]

Reduction of P-kick

The proportional term (P-term) in the PID controller (2.67) is

$$u_p = K_p e = K_p (w_p y_{SP} - y) \tag{2.76}$$

Let us assume initially that $w_p = 1$, which means there is no reduction of the setpoint weight in the P-term. If the setpoint is changed, say it is changed as a step, the P-term u_p and therefore the total control variable where u_p appears as an additive term, is changed like a step, too. This can be unfortunate for mechanical actuators, cf. the discussion in Chapter 2.7.1.

By setting w_p less than 1, the setpoint has reduced weight in the P-term, which implies less abrupt changes in u_p caused by setpoint changes. It is however common in commercial controllers to have $w_p = 1$ (which means no reduced weight), but if w_p is reduced, then $w_p = 0.3$ is suggested [24].

Note: If you are to use the Ziegler-Nichols' closed loop method to tune PID parameters, the control system will not react at all to excitations via the setpoint if $w_p = 0$. So, you should not set $w_p = 0$ during the tuning.

Smoothing the control signal by ramping the setpoint

In the previous sections you have seen that abrupt setpoint changes implies abrupt control signal changes. As explained, one way to reduce problem is to use a reduced setpoint weight in the D-term and/or in the P-term (the D-term is the most critical case due to the time differentiation). An alternative solution is to avoid sudden changes, e.g. step changes, in the setpoint. The change of the setpoint from one value to another may follow a ramp in stead of a step, see Figure 2.31. Commercial

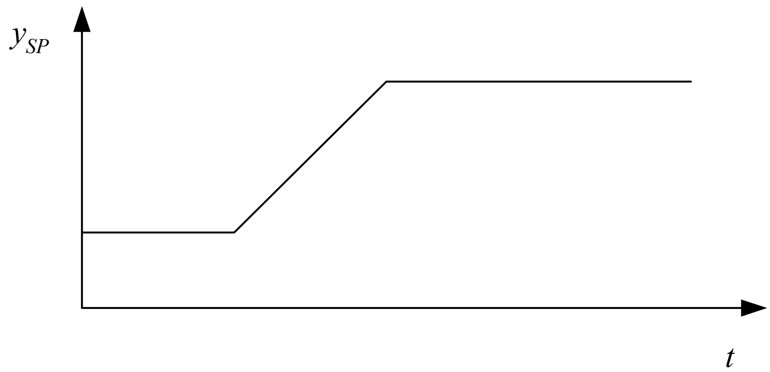

Figure 2.31: The change of the setpoint from one value to another may follow a ramp in stead of a step to avoid kicks in the control signal.

controllers typically supports setpoint ramping.

Example 2.14 ***Setpoint ramping***

Figure 2.32 shows a simulation of the same system which was simulated in Example 2.13, but now there is no reduced setpoint weight. In stead, the setpoint is changed as a ramp. Clearly the control signal varies much smoother compared to the response shown in Figure 2.29 where the setpoint was changed as step. The response after the disturbance step is of course the same since the disturbance is independent of the setpoint.

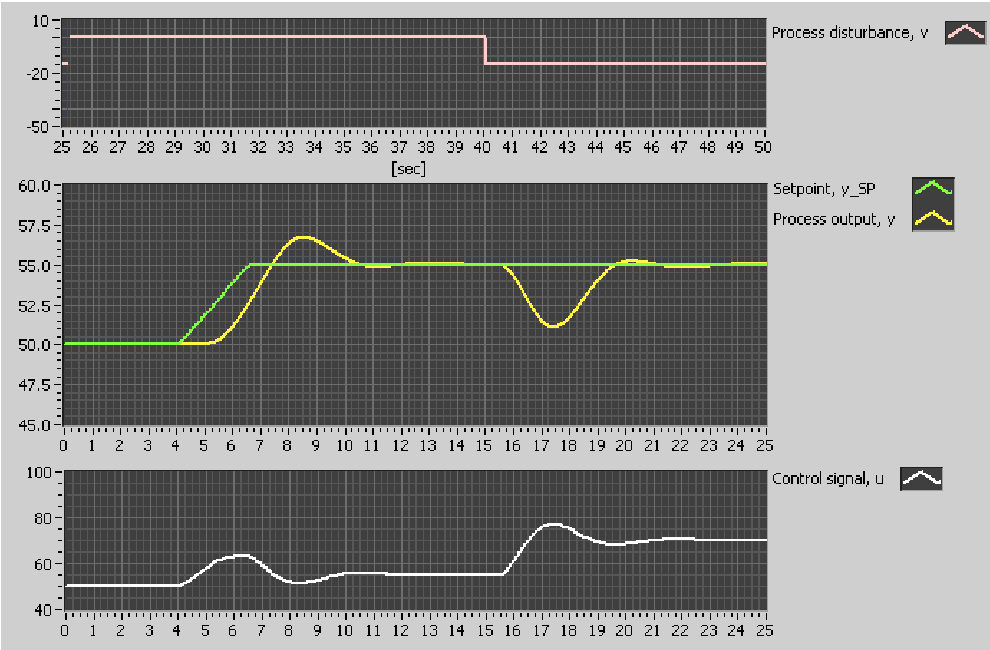

Figure 2.32: Example 2.14: Ramping the setpoint gives smoother control signal.

[End of Example 2.14]

2.7.2 Integrator anti wind-up

All actuators have saturation limits, i.e. a maximum limit and a minimum limit. For example, a power amplifier (for a heater or a motor) can not deliver an infinitely amount of power, and a valve can not have an infinitely large opening and can not be more closed than closed(!). Under normal process operation the control variable should not reach the saturation limits.

But everything is not normal all the time. Assume that in a time interval a large process disturbance acts on the process so that the process output variable is reduced. The control error then becomes large and positive, and the control variable will increase (because of the integral term of the PID controller) until the control signal limits at its maximum value, $u_{\max}$. Assume that the disturbance is so large that $u_{\max}$ is not large enough to compensate for the large disturbance. Because of this, the control error keeps large, and the integral of the control error continues to increase, which means that the calculated integral term u_i continues to increase. This is *integrator wind-up*.

Assume that the process disturbance after a while goes back to its normal value. This causes the process output variable to increase since the disturbance is reduced (the load is removed), and the error will now change sign (it becomes negative). Consequently the integral term starts to integrate downwards (its value is continuously being reduced), so that the calculated u_i is reduced, which is ok, since the smaller disturbance requires a smaller control signal. However, the problem is that is may take *a long time* until the large value of the calculated u_i is reduced (via the down-integration) to a normal (reasonable) value. During this long time the control variable is larger than what is required to compensate for the disturbance, causing the process output variable to be larger than the setpoint during this time.

To sum it up: A large and long-lasting process disturbance which forces the control variable (via the controller) to one of its saturation limits, implies a long-lasting error different from zero.

A practical PID controller must be able to cope with the possibility of integrator wind-up, that is, it must have some *anti wind-up* mechanism. You can assume that anti wind-up is implemented in commercial controllers. The principle of an anti wind-up mechanism is simple: Since the problem is that the integral term increases continuously during actuator saturation, the solution is to halt the integration when the control signal reaches either its maximum or minimum limit. An analogy of anti wind-up is to mount an overflow outlet in a liquid tank, see Figure 2.33. A tank is dynamically an integrator, so it represents here the I-term of the controller.

Note that you can not implement integrator anti-windup by just limiting the control signal, u, calculated by the PID controller. It is crucial to halt the integration of the control error.

There are several ways to implement anti wind-up in a continuous-time PID controller [24], but these are not described here. It is more likely that, if you are to implement integrator anti wind-up in a controller, it will be on a discrete-time controller, cf. Chapter 5.

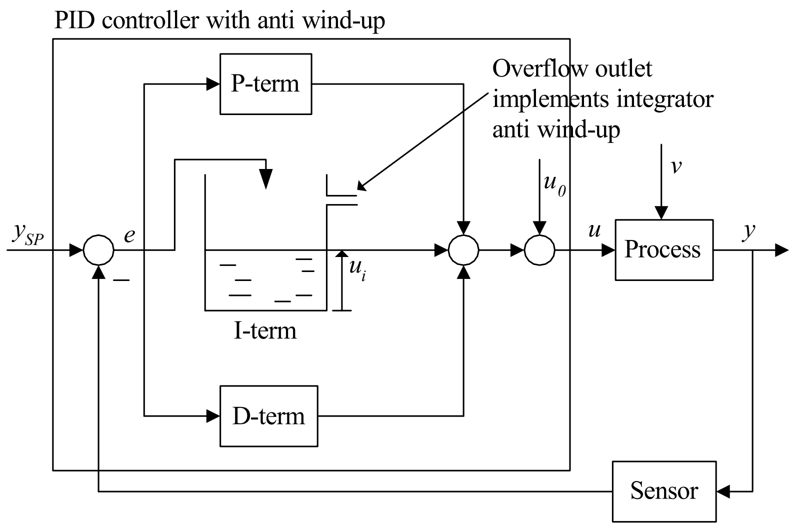

Figure 2.33: An analogy of anti wind-up: The overflow outlet limits the *integral* of the inflow (which is the volume, or the level) in a liquid tank

Example 2.15 ***Integral anti wind-up in a temperature control system***

Figure 2.34 shows the front panel of a simulator for a temperature control system for a liquid tank with continuously mass flow. The disturbance is here the inlet temperature T_{in}, which is is changed as a step from 40°C to 10°C at approx. 210min and back to 40°C at approx. 300min. The temperature setpoint T_{SP} is 70°C (constant). The parameters of the PID controller are $K_p = 6.7$, $T_i = 252\text{s} = 42\text{min}$ and $T_d = 63\text{s} = 10.5$ min (found using Ziegler-Nichols' closed loop method). The maximum value of the control variable is 100% and the minimum value is 0%. When T_{in} is reduced to 10°C, the actuator (heating element) goes into saturation (100%), trying to compensate for the (cold) disturbance. It can be shown that the control variable u should have a value of 122.5% (which corresponds to more heat power than what is available) to be able to compensate for $T_{in} = 10$°C.

Figure 2.34 shows the simulated responses in the control system *without* using integrator anti wind-up, and Figure 2.35 shows the responses *with* integrator anti wind-up. In the case of no anti wind-up, it was observed (but this is not indicated in Figure 2.34) that the integral term u_i in the PID controller reached a maximum value of approximately 2200%! The simulations clearly show that it is beneficial to use integrator anti wind-up (as the temperature returns much quicker to the setpoint after the

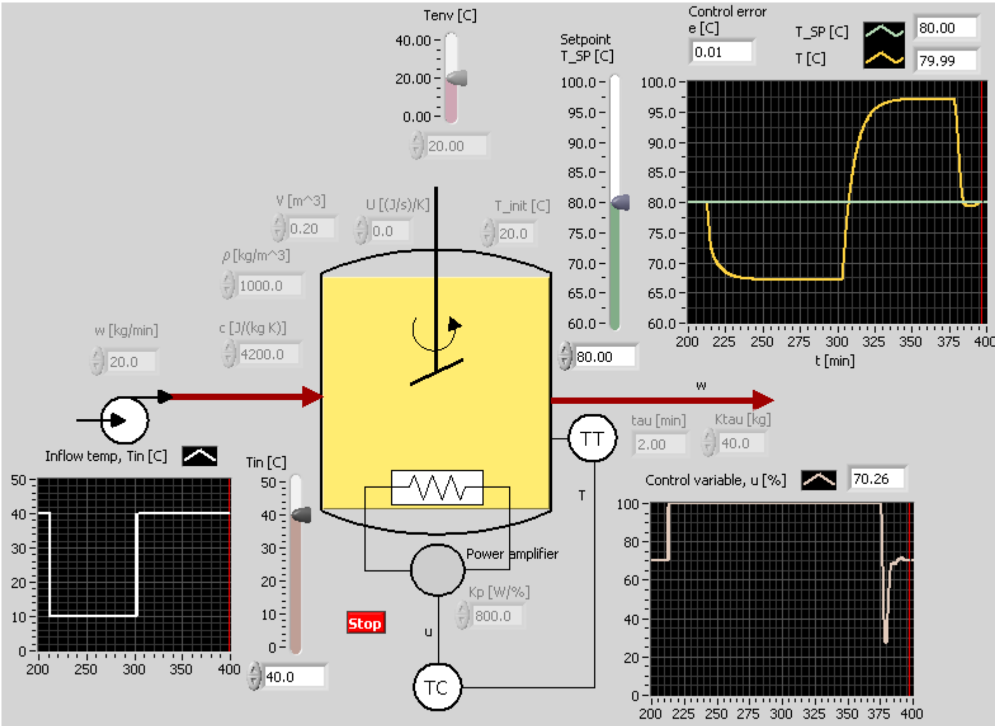

Figure 2.34: Example 2.15: Temperature control *without* anti wind-up

disturbance has changed back to its normal value).

[End of Example 2.15]

2.7.3 Measurement noise. Signal variance

Introduction

You have already seen the problems concerning measurement noise in a control loop, cf. Section2.6.7. Figure 2.22 shows where the measurement noise enters the control loop. Section2.6.7 describes a necessary modification of the derivative term in a PID controller: A lowpass filter is inserted before (in series with) the D-term to attenuate the noise before it is time-differentiated to avoid too large noise-generated responses in the control signal. But what if the lowpass filter in the D-term does not give sufficient noise filtering? Then an additional filter should be included in the feedback path, acting on the measurement signal. This filter can be

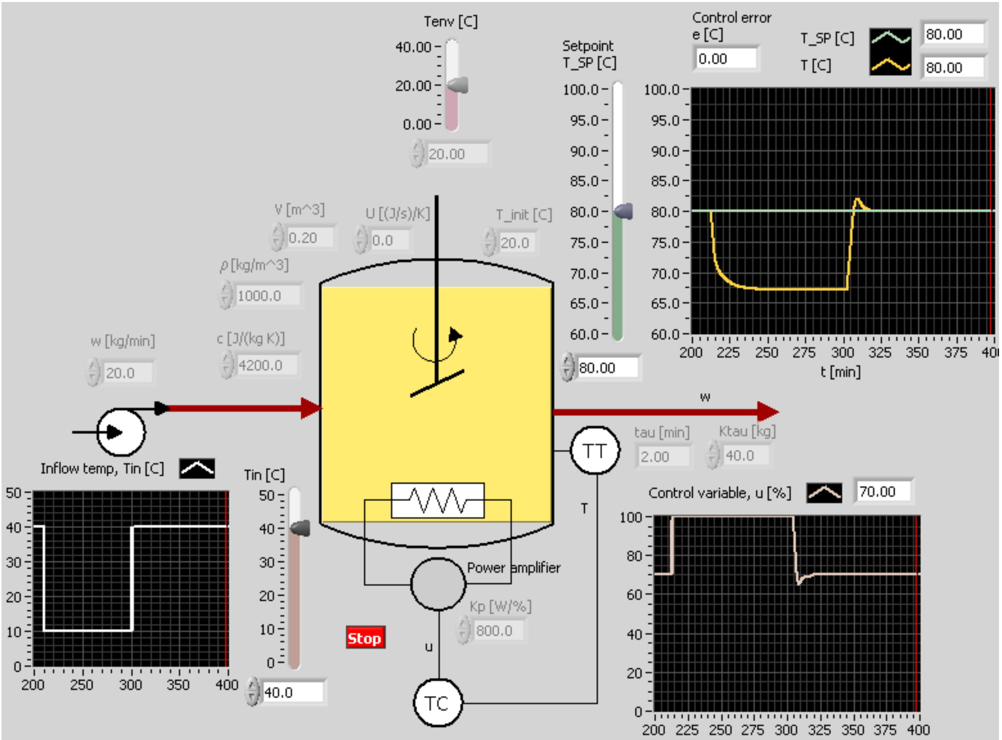

Figure 2.35: Example 2.15: Temperature control *with* anti wind-up

- either a linear dynamic lowpass filter,
- or a deadband filter.

These solutions are described in more detail below.

Calculating the variance

Measurement noise is typically a random signal. The noise propagates through the control system via the controller, causing variations in all variables in the control system. Figure 2.36 shows typical examples of a noisy process measurement and the control variable and in a simulated control system. The variances shown in the figure are calculated as explained below from the 50 most recent samples.

To express the variation of a process variable, the statistical *variance* can be calculated, alternatively the standard deviation which is the square root of the variance. The larger variance, the larger the variations. The

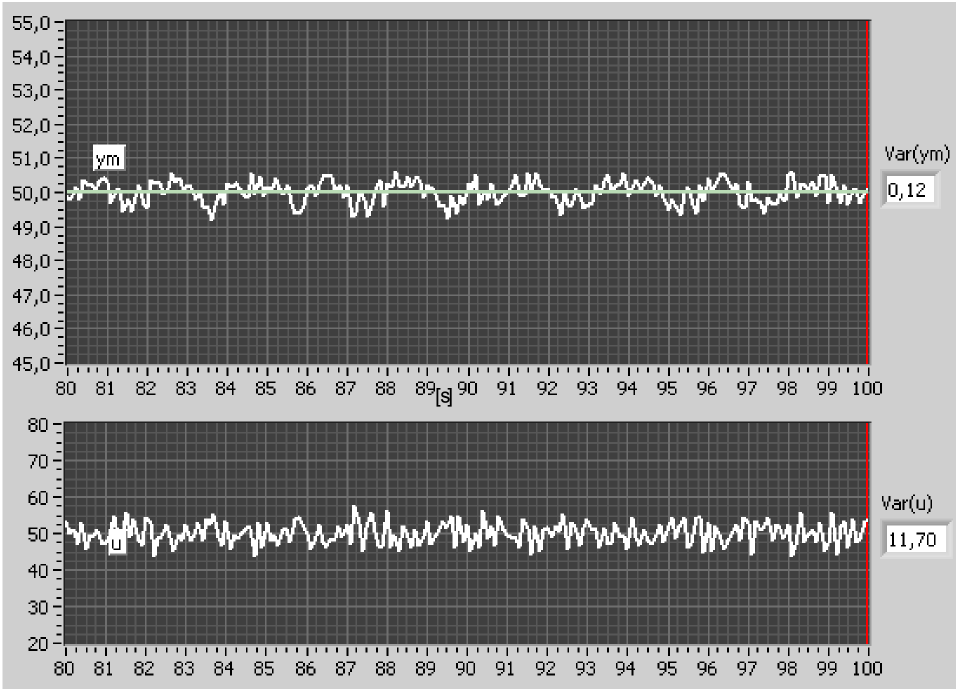

Figure 2.36: Typical examples of a noisy process measurement and the control variable

variance is the mean square deviation about the mean value[11]

$$\text{Var}(y_m) = \frac{1}{N-1}\sum_{k=1}^{N}\left[y_m(t_k) - m_{y_m}\right]^2 \tag{2.77}$$

where N is the number of samples and m_{y_m} is the mean value of y_m, which may be calculated by

$$m_{y_m} = \frac{1}{N}\sum_{k=1}^{N} y_m(t_k) \tag{2.78}$$

The numerical value of the variance is usually not particularly useful in itself, but it is useful when comparing signals.

In Example 2.16 variances will be used to express the improvements by using a lowpass filter on the process measurement signal.

[11] To obtain a so-called nonbiased estimate of the variance, you must divide by $N-1$, not by N.

Using a dynamic lowpass filter

Figure 2.37 shows a control loop having a lowpass filter acting on the measurement signal. The filter can be a discrete-time filter implemented in

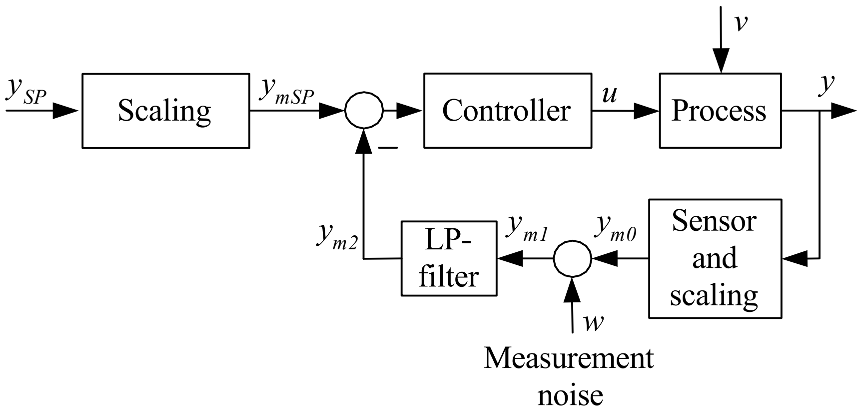

Figure 2.37: Control loop having a lowpass filter acting on the measurement signal. (LP = lowpass.)

the control equipment. It is common that control equipment have inbuilt lowpass filter functions. Alternatively, the filter can be a continuous-time lowpass filter implemented using electronic components external to the control equipment. For example a first order filter can be implemented as an RC-circuit.

Let us assume that the measurement lowpass filter is a first order filter. Such a filter has the following transfer function from filter input x_{in} to filter output x_{out}:

$$\frac{x_{out}(s)}{x_{in}(s)} = H(s) = \frac{1}{\frac{s}{\omega_b} + 1} = \frac{1}{\frac{s}{2\pi f_b} + 1} \tag{2.79}$$

The bandwidth of the lowpass filter is $\omega_b = 2\pi f_b$ where ω_b has unit rad/s and f_b has unit Hz. The bandwidth must be given a value which is smaller than the frequency of the substantial noise frequency components so that these components fall within the stopband of the filter. The bandwidth may be tuned experimentally. Figure 2.38 shows a typical amplitude gain function of first order lowpass filter. One example of a noise frequency component is shown in the figure (it is in the stopband of the filter). The bandwidth is typically defined as the frequency where amplitude gain is $1/\sqrt{2} = 0.71 \approx -3\text{dB}$.

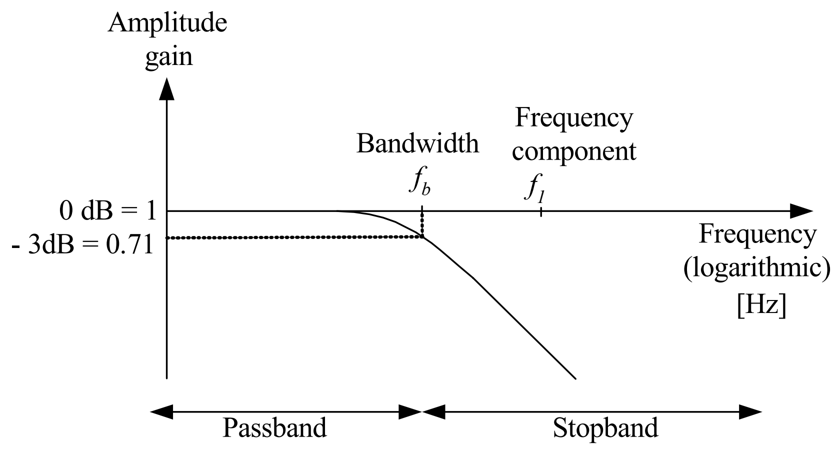

Figure 2.38: Typical amplitude gain function of a lowpass filter. One example of a noise frequency component is shown.

Example 2.16 ***Measurement noise filter in a control loop***

In this example a control system for a process having the following transfer function model is simulated:

$$y(s) = \frac{K_u}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} u(s) \tag{2.80}$$

$$+ \frac{K_v}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau s} v(s) \tag{2.81}$$

(The process is thus a second order system with time delay.) u is the control variable, and v is the process disturbance. The process parameter are

$$K_u = 1;\ K_v = 1;\ T_1 = 1\text{s};\ T_2 = 0.5\text{s};\ \tau = 0.3\text{s}; \tag{2.82}$$

The PID parameters are

$$K_p = 2.8;\ T_i = 1.2\text{s};\ T_d = 0.3\text{s}; \tag{2.83}$$

(tuned with the Ziegler-Nichols' closed loop method). Figure 2.39 shows simulated responses in the control system. The measurement noise is a random signal uniformly distributed[12] between -1 and $+1$. The lowpass filter acts on the process measurement, cf. Figure 2.37. It is switched into the loop at $t = 10$s. From Figure 2.37 we see that the filter removes noise from the process measurement and that the control variable (therefore) is less noisy. The filter is a first order lowpass filter with bandwidth 1.5Hz.

[12] which means that there is equal probability for any value between -1 and $+1$.

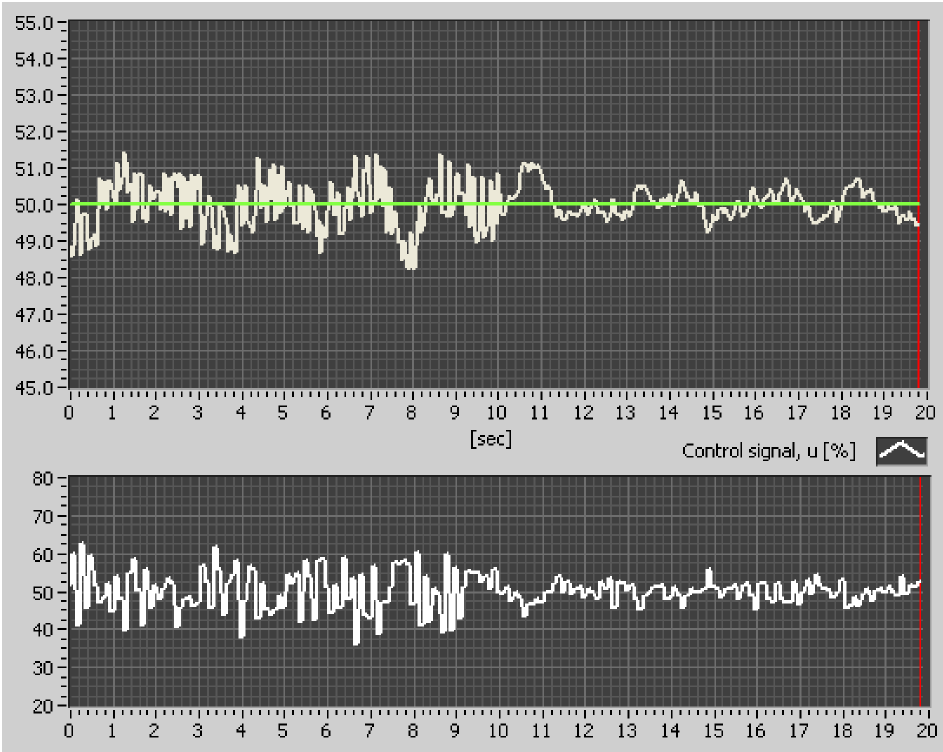

Figure 2.39: Example 2.16: Simulated responses of a control system. A lowpass filter acting on the process measurement signal is switched into the control loop at $t = 10$s.

Table 2.1 shows the variances of the control signal u and the process measurement (after the filter) without and with lowpass measurement filter. The variances are calculated from the 50 most recent signal samples. It is clear from the variances that the filter reduces the influence of the noise in the loop.

Without Filter	**With Filter**
$\text{Var}(u) = 36.4$	$\text{Var}(u) = 5.2$
$\text{Var}(y_m) = 0.33$	$\text{Var}(y_m) = 0.13$

Table 2.1: Variances of control signal u and measurement signal y_m without and with lowpass filter

[End of Example 2.16]

Including a filter in the control loop *changes the dynamic properties* of the loop! Actually, it can cause stability problems in the control loop. In most cases, the less bandwidth (i.e., more sluggish filter), the more reduction of the stability of the loop.

Example 2.17 ***Poor stability because of measurement filter***

Figure 2.40 shows simulated responses for the same control system simulated in Example 2.16. Before $t = 10$s there is no lowpass filter in the loop, while after $t = 10$s a first order lowpass filter with bandwidth 0.2Hz is switched into the loop. The filter causes the control system to have very poor stability.

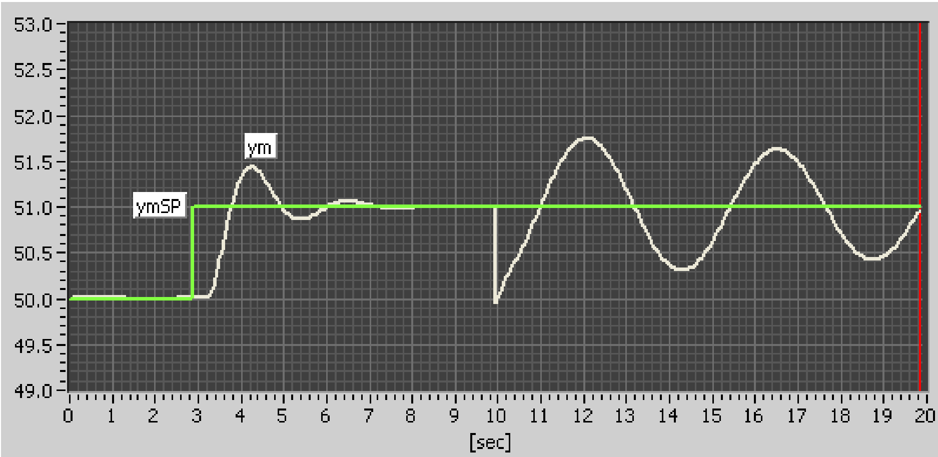

Figure 2.40: Example 2.17: A first order lowpass filter is switched into the control loop at $t = 10$s, causing the control system to have poor stability.

[End of Example 2.17]

If a measurement filter results in poor stability of the control loop – how can that problem be avoided? By *tuning (or re-tuning) the controller with the filter in the control loop.*

It is tempting to select a very small bandwidth of the measurement lowpass filter to achieve strong attenuation of the measurement noise. But in addition to attenuating noise, also frequency components in the ideal (noise-free) process output signal is attenuated. In other words: Important process information may be removed from the measurement signal. This in turn may cause the controller to calculate the control signal on basis of an erroneous control error value. One way to solve this problem, is to introduce a similar filter in series with the setpoint, as shown in Figure . This solution is equivalent to placing one filter in series with (or before) the PID controller in Figure 2.41. A setpoint filter implies that the setpoint which the controller observes, becomes more sluggish since high frequency components are attenuated.

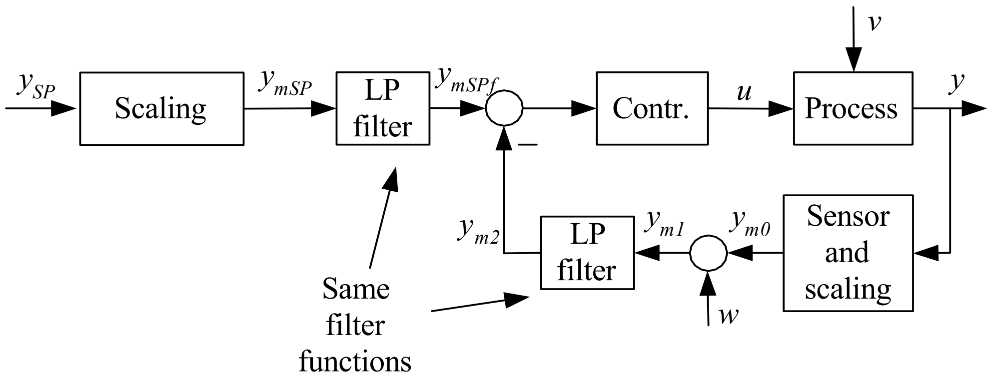

Figure 2.41: Lowpass filter acting on the setpoint

Using a deadband filter

If you know the maximum amplitude of the measurement noise, the noise can be removed from the noisy measurement signal by letting the signal pass through a deadband filter, see Figure 2.42. The output value of the

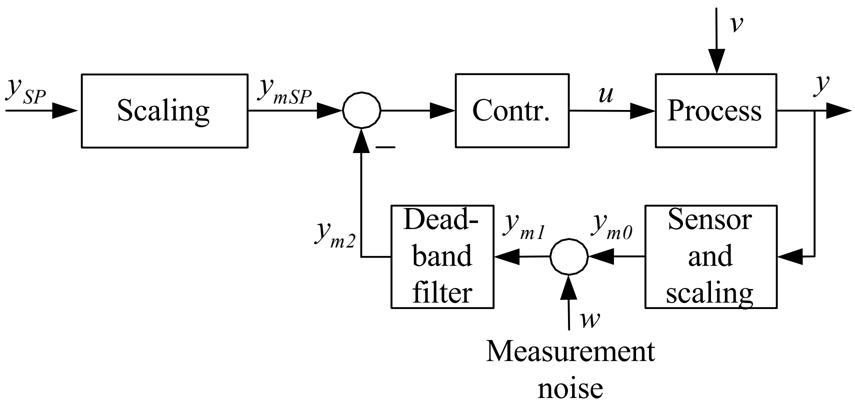

Figure 2.42: Deadband filter acting on the process measurement signal

deadband filter changes value only if the change of the input signal is larger than the deadband.

Example 2.18 ***Deadband measurement filter in the control loop***

Figure 2.43 shows a simulation of a control system with deadband measurement filter. The process and the PID controller are as described in Example 2.16. In the simulation the measurement noise is a random signal

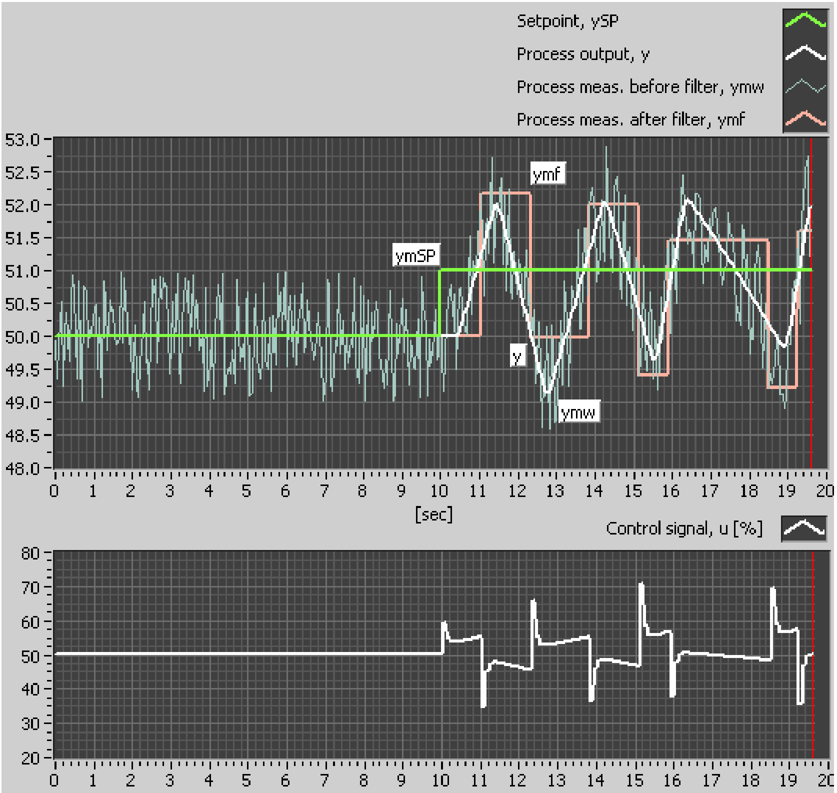

Figure 2.43: Example 2.18: Simulation of a control system with deadband measurement filter

uniformly distributed between −1% and +1%. The deadband of the filter is 2%. The simulation shows the following:

- Up to time $t = 10$s the setpoint is constant (50%). The measurement signal which is the output of the deadband filter is constant since the amplitude of the noise is smaller than the deadband. And since the measurement signal is constant, the control signal generated by the controller is constant – which is good!
- At $t = 10$s the setpoint is changed as a step (from 50% to 51%) which implies that the measurement signal due to the overshoot in

the step response changes value beyond the deadband of 2%. Thereafter the deadband filter acts similar to an on/off-element in the loop, and there are sustained oscillations in the loop – not good!

[End of Example 2.18]

You have in this Section seen two ways of filtering measurement noise:

- *Using a dynamic lowpass filter*: The dynamic filter can be easily tuned via the bandwidth. The filter influences the dynamics and hence the stability of the loop. The controller should be tuned with the filter in the loop.
- *Using a deadband filter*: This filter may give a constant measurement signal, as long as the input signal to the deadband filter does not change more than the deadband of the filter. Once the deadband is exceeded, the deadband filter may behave almost like an on/off controller, causing oscillations in the control loop.

From the above results it seems that the dynamic filter is a safer way than deadband filter to handle measurement noise.

2.8 Performance index of control systems

Assume that different control systems are to be compared, or that different controller parameter for one control system are to be compared. There are several ways to express the performance of the control systems, e.g. bandwidth and stability margins. These measures are based on a mathematical model of the control system, and they are described in Chapter 6.

Alternatively, we can use performance indices which are functions of the observed control error e. These indices does not require a mathematical model. Probably the most frequently used index is the IAE – Integral of Absolute value of control Error [15]:

$$\text{IAE} = \int_0^\infty |e| \, dt \tag{2.84}$$

The less IAE value, the better performance. The IAE value is finite only if e converges towards zero in steady-state, which in practice requires the controller to have have integral action, as in a PI controller and in a PID controller. In discrete time the IAE value can be regarded as the sum of the absolute values of the sample values of the control error, since

$$\text{IAE} = \int_0^\infty |e|\,dt \approx \sum_{k=0}^{\infty} h\,|e(t_k)| = h\sum_{k=0}^{\infty}|e(t_k)| \tag{2.85}$$

where h is the time step (time interval between each discrete point of time). h is $t_k - t_{k-1}$. k is a time index: $t_k = hk$.

We can in practical applications calculate the IAE value only over a finite time interval, say from $t = 0$ to t_k. We can derive a recursive algorithm of the IAE as follows:

$$\text{IAE}(t_k) = \int_0^{t_k} |e|\,dt \tag{2.86}$$

$$= \int_0^{t_{k-1}} |e|\,dt + \int_{t_{k-1}}^{t_k} |e|\,dt \tag{2.87}$$

$$\approx IAE(t_{k-1}) + h|e(t_k)| \tag{2.88}$$

The expression $h|e(t_k)|$ in (2.88) is an Euler backward (rectangular) approximation to the latter integral in (2.87).

Example 2.19 ***IAE for level control of chip tank***

The level control system for the wood-chip tank described in Example 2.3 (page 19). Figure 2.19 shows simulated responses with a PI controller, and Figure 2.20 shows responses with a PID controller. In both cases the IAE index is computed for the control error after the step in the outflow w_{out} (disturbance), which is from $t = 65$s to 120s. The results are as follows:

- PI controller: IAE = 6296
- PID controller: IAE = 3767

We see that the PID controller has better IAE performance than the PI controller. This is in accordance with the better compensating performance of the PID controller that we can easily see in the simulations.

[End of Example 2.19]

2.9 Selecting P, PI, PD, or PID?

In general, the PID controller is the first choice since it gives zero static control error and relatively quick control. Here are a few guidelines for choosing other controller functions than PID:

- If there is much process measurement noise the derivative term should be dropped. What remains is a PI controller.
- If the process is of first order or is a pure integrator and in addition has a time delay, the stability margins of the control system may be small if the controller has a derivative term. This means that the control system stability may become poor after small parameter changes in the process. If you want to be on the safe side, the derivative term may be left out, and a PI controller remains.
- If the process has fast dynamics compared to other processes it is connected to, the derivative term may be dropped since the control system will be quick enough with a PI or a P controller. On example is quick flow control loops which works as inner or secondary loops within more sluggish loops for temperature control or level control, as in cascade control, cf. Chapter 9.2. The increase of control speed due to the derivative term is usually not important since the inner loop in any case will be faster than the outer, primary loop, due to the fast dynamics of the inner process.
- The P controller may be a sufficiently good controller for processes containing a pure integrator, as motors where position is to be controlled, and liquid tanks where level is to be controlled when the disturbance (e.g. load force or load torque on the motor or tank inflow or outflow) is zero or small. In these cases, the integral action in the "inbuilt" integrator in the process ensures zero static control error at constant setpoint. However, a further analysis, as described in Chapter 6, should be done to see if the simple P controller is sufficient to obtain the specifications of maximum control error and quickness (bandwidth).
- The PD controller may be applied in electrical servomechanisms where the steady-state control error due to disturbances (as load torque) is sufficiently small. The D-term may increase the quickness (bandwidth) of the control loop. The D-term may cause stability problems in hydraulic servomechanisms because of the hydraulic resonance.

2.10 Reduction of control error by process changes

Earlier in this chapter we have seen how to use the control variable u to control the process so that the control error becomes sufficiently small. However, the control error, $e = y_{SP} - y$, depends not only of only the control variable, but also of the disturbance and the process itself. This implies that it may be possible to reduce the control error by change the disturbance and/or the process. This is explained in more detail below.

1. **Reducing control error by reducing or isolating the disturbance(s).** In most processes it difficult or impossible realize this point. This is because the disturbance often is closely related to the function of the process.

   Here are a few examples:

   - *Example 1*: In a level control system for the wood-chip tank, cf. Example 2.3, the chip outflow (disturbance) can not be reduced since it is the feed to the cookery downstream in the process line.
   - *Example 2*: In a temperature control system for liquid the tank, cf. Example 2.4, it is hardly possible to change the ambient temperature (disturbance). But it may be possible to change the inlet temperature (also a disturbance).
   - *Example 3*: In a motor speed control system, cf. Example 2.5, it is not realistic to be able to change the load torque (disturbance) since it is probably closely related to the function of the motor, as in a grinding machine or a conveyor belt.

2. **Reducing control error by changing the process construction**. Of course it may be impractical to change the construction of a process which is already built, but a process under planning can be changed more easily.

   A few examples:

   - *Example 1*: In a level control system for a liquid tank a wider tank will reduce the level variations (but not mass variations).
   - *Example 2*: In the level control system for the wood-chip tank, cf. Example 2.3, reducing the transport delay on the conveyor belt may give quicker control and hence smaller control error.

Imagine the inlet screw rotational speed *and* the band speed were controlled simultaneously and proportionally...[13]

- *Example 3*: In a temperature control system, better tank isolation will reduce the effects of the ambient temperature, see Figure 2.44. And an increase of the tank volume would give

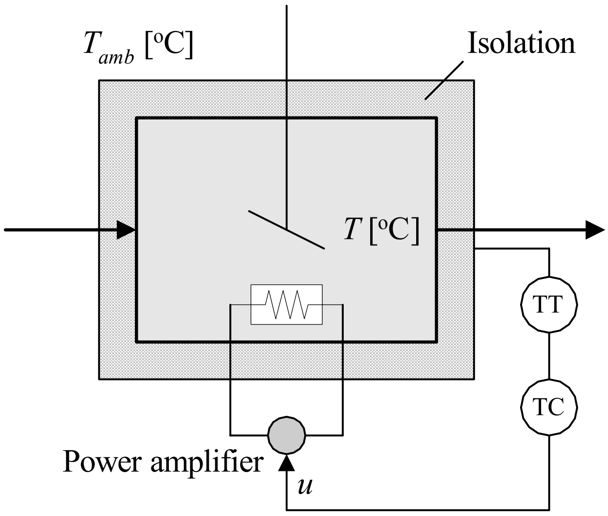

Figure 2.44: Better isolation of the tank will reduce the influence of the ambient temperature T_{amb} on the tank temperature T.

better attenuation (dynamically, but not statically) of temperature disturbances in the tank.

- *Example 4*: In a speed control system a larger motor or using a gear reduces the effects of the load torque on the motor speed.

2.11 Control loop stability

It is important to be aware that there may be stability problems in a control loop. It is an basic requirement to a control loop that it is stable. Simply stated this means that the response in any signal in control loop converges towards a finite value after a limited change (with respect to the amplitude) of the setpoint or the disturbance or any other input signal to the loop. For example, the control loop constituting the control system for the wood-chip tank in Example 2.11 is stable.

All methods for tuning controller parameters have as the main aim that

[13] Then the transport delay was eliminated.

the control loop is stable. The PID parameters used in the level control system in Example 2.11 where tuned using the Ziegler-Nichols' closed loop method, cf. Chapter 4.4. However, there is always a possibility that a feedback control system which is originally stable, *may become unstable* due to parameter changes in the loop. Instability implies that signals in the loop starts to increase in amplitude until some saturation limit is reached (for example, a valve have a limited opening).

Instability can be explained in two ways:

- The signal one place in the loop is amplified too much through the subsystems in the loop. In other words, *the loop gain is too high.* The loop gain is the product of the gains in each of the subsystems (controller, process, sensor) in the loop.
- There is *too much time delay* through the subsystems in the loop.

Chapter 6.4 describes ways to analyze control loop stability theoretically.

Example 2.20 ***Instability in the wood-chip tank level control system***

We will see that the level control system for the wood-chip tank becomes unstable if the controller gain K_p in the PID controller becomes too large, and if the transport delay related to the conveyor belt becomes too large. Figure 2.15 (page 32) shows the front panel of the simulator. The control system is initially stable for the following PID parameters:

$$K_p = 1.9;\ T_i = 540\text{s} = 9.0\text{min};\ T_d = 135\text{s} = 2.25\text{min} \tag{2.89}$$

(found by the Ziegler-Nichols' closed loop-method, cf. Section4). The time delay is 250s = 4.17min.

Figure 2.45 shows responses for $K_p = 6$, which is considerably larger than the optimal (Ziegler-Nichols) value of 1.9. The setpoint and the disturbance (chip outflow) are constant. The control error has initially the very small value of 0.0004. We see from the simulation that the control system is unstable. One explanation of the instability is that the loop gain is too large, due to the large K_p-value. The amplitude of the oscillations are limited due to the limits of the control variable (the maximum value is 100%, and the minimum value is 0%).

Figure 2.46 shows responses with the original PID values, but for an increased value of the time delay τ, namely 600s = 10min (the nominal

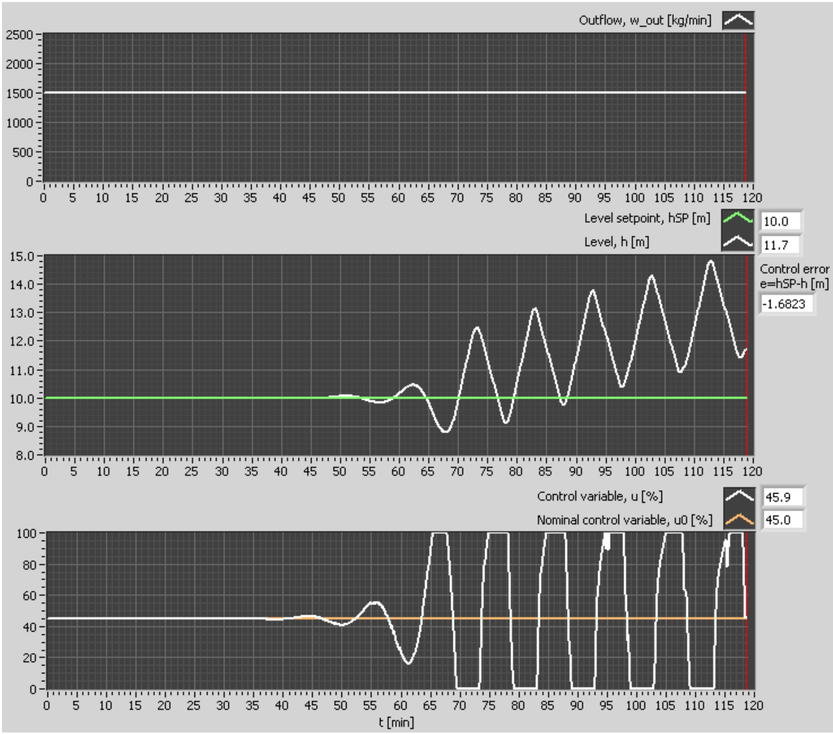

Figure 2.45: Example 2.20: Level control of the wood-chip tank with a (too) large K_p-value of 6, which causes the control system to become unstable. (The front panel of the simulator is as shown in Figure 2.15.)

value is 250sec= 4.17min). The simulation shows that the control system is unstable, which is due to the (too) large time delay in the loop.

[End of Example 2.20]

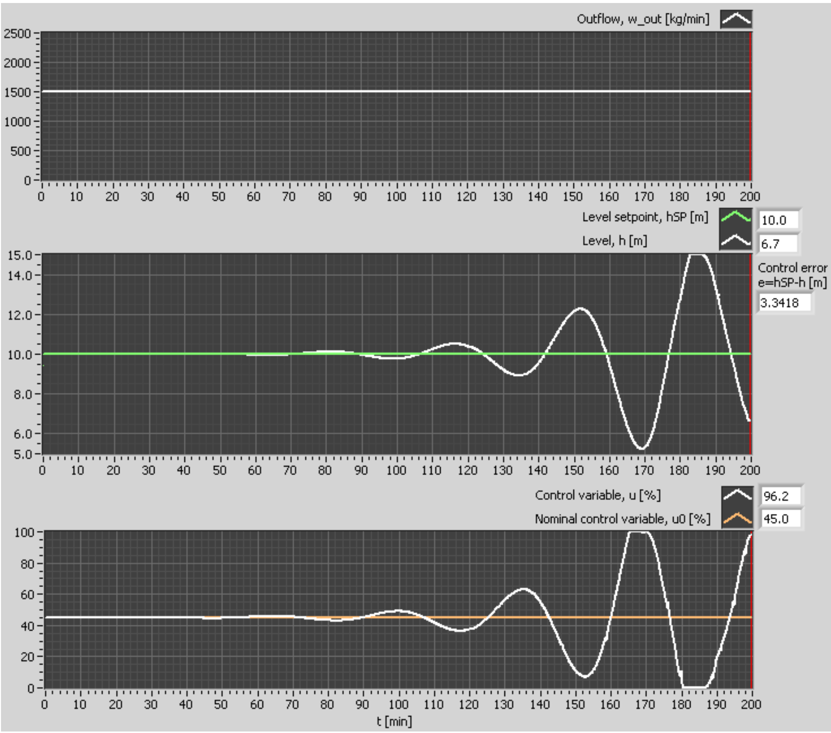

Figure 2.46: Example 2.20: Level control of the wood-chip tank with a (too) large τ-value of 9, which causes the control system to become unstable. (The front panel of the simulator is as shown in Figure 2.15.)