Katholieke Universiteit Leuven

Department of Electrical Engineering **ESAT-SISTA/TR 1997-08**

A tutorial on PID-control[1]

Katrien De Cock, Bart De Moor, Willem Minten, Wim Van Brempt, Herman Verrelst[2]

January 1997

[1]This report will be available by anonymous ftp from *gate.esat.kuleuven.ac.be* in the directory *pub/SISTA/minten/reports/pidcontrol.ps.gz*

[2]ESAT - Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Leuven (Heverlee), Belgium, tel 32/16/22 09 31, fax 32/16/22 18 55, email: `Willem.Minten@esat.kuleuven.ac.be`. This work is supported by several institutions: the Flemish Government (Concerted Research Action GOA-MIPS, the FWO (Fund for Scientific Research - Flanders: G.0292.95, G.0256.97), the FWO Research Communities (ICCoS, Advanced Numerical Methods for Mathematical Modeling), the IWT (Flemish Institute for Science and Technology in Industry: IWT/VCST project (1992-1997), IWT Action Program on Information Technology (ITA/GBO/T23) - IT-Generic Basic Research: ISIS)), the Belgian Government (Inter-university Attraction Poles: IUAP P4-02 (1997-2001) and IUAP P4-24 (1997-2001)), and the European Commission (Human Capital and Mobility Network SIMONET; SCIENCE-ERNSI SC1-CT92-0779; TMR Network ALAPEDES; Keep In Touch SYSIDENT-KIT124). The scientific responsibility is assumed by its authors.

Abstract

The problem of tuning the parameters of a PID-controller arises in many engineering applications. Often, different measures of quality must be considered simultaneously, so performance trade-offs are inevitable.
It is the aim of this seminar to give a *practical* introduction into the basic skills of PID tuning. Therefore a mathematical treatment of the subject is omitted as much as possible. The text starts by introducing basic modelling tools of the process to be controlled. Then some process properties are explained in the time and Laplace domain. Thereafter the basic performance specifications of a basic compensator control configuration are shown and some basic control laws are proposed. Some other frequently used control configurations are also explained in short. Finally the basic tuning procedures for PID-controllers are presented. First, a stabilising controller is designed, next, the controller parameters are changed to obtain the desired performance. In the near future the implementation of the PID controller in an industrial environment will be included also.

If you look for an answer on a specific question about these seminars, please visit the FAQ list: `http://www.esat.kuleuven.ac.be/~minten/PIDsem.html`

Contents

1 Introduction **4**
- 1.1 Why Control? . . . 4
- 1.2 Development of a Control System . . . 4

2 Plant Modelling **7**
- 2.1 Introduction . . . 7
- 2.2 Physical Modelling . . . 7
  - 2.2.1 An example: A hydraulic positioning tool for plastic colouring purposes 8
- 2.3 Identification . . . 15
  - 2.3.1 Introduction . . . 15
  - 2.3.2 Graphical fitting of first order processes using step-tests . . . 15
    - Tangent Method . . . 15
    - Inflection Method. . . . 16
    - 35%-85%-Method . . . 18
  - 2.3.3 Graphical Fitting of second-order models using step-tests . . . 18
    - Harriot's Method . . . 20
    - Smith's Method . . . 21
    - Model for systems with oscillatory step response . . . 23
  - 2.3.4 Model for Integrating Systems . . . 24
    - Impulse Response . . . 24
    - Step Test . . . 24

3 Plant Analysis **25**
- 3.1 Pole-Zero analysis . . . 25
- 3.2 Frequency domain analysis . . . 30
- 3.3 Time domain analysis . . . 32

4 Performance Specifications **34**
- 4.1 Closed-loop stability . . . 35
- 4.2 Frequency domain performance . . . 35
- 4.3 Time domain performance . . . 37
- 4.4 Sensitivity to Parameter Changes . . . 38
- 4.5 Conclusions . . . 39
- 4.6 Two methods to reach the goals . . . 39

5 Control Laws **40**
5.1 Introduction . . . 40
5.2 On-Off Control . . . 41
5.2.1 General Working Principle . . . 41
5.2.2 Hysteresis . . . 42
5.3 PID-Control . . . 44
5.3.1 Open Loop performance of a PID-controller : the control law . . . 44
5.3.2 Closed Loop performance of a PID-controller . . . 45
P-action . . . 46
I-action . . . 48
PI-action . . . 49
PD-action . . . 49
PID-action . . . 49
5.3.3 Technical Terms . . . 49
5.3.4 PID-Control in Processes with dead-time . . . 52
5.3.5 Anti Wind-Up . . . 55

6 Control Configurations **58**
6.1 Master-Slave Control (Cascade Control) . . . 58
6.2 Feed-forward Control . . . 59
6.3 Split-Range Control . . . 61
6.4 Ratio Control . . . 62

7 Tuning Methods **63**
7.1 Introduction . . . 63
7.2 Heuristic Tuning Methods . . . 63
7.2.1 Trial-and-Error tuning method . . . 63
7.2.2 Ziegler-Nichols Tuning Rules . . . 64
7.3 Stability Limits (Routh-Hurwitz) . . . 68
7.4 Root Locus . . . 72
7.5 Bode Plots . . . 77
7.5.1 P-control . . . 78
7.5.2 I-control . . . 78
7.5.3 D-control . . . 79
7.5.4 PI-control . . . 80
7.5.5 PD-control . . . 80
7.5.6 PID-control . . . 81
7.6 Auto-Tuners . . . 82
7.7 Optimal Control . . . 82
7.7.1 Output feedback optimum control strategies . . . 82
Classical time-domain performance measures . . . 83
Some important guidelines . . . 90
Response shape time domain performance measures . . . 92
7.7.2 State feedback optimum control strategies . . . 94

8 Seminar tasks **96**
8.1 Zitting 1 . . . 96
8.2 Zitting 2 . . . 102
8.3 Zitting 3, 4 en 5: De electromagnetische YoYo . . . 108
8.3.1 Algebraïsche beschrijving . . . 109
Niet-lineair toestandsruimtemodel . . . 109
Linearisatie . . . 110
8.3.2 Simulatie . . . 110
8.3.3 Opdracht . . . 111
8.3.4 Verslag . . . 111

Chapter 1

Introduction

1.1 Why Control?

Control-engineers face two main problems:

- *The regulator problem:* The aim is to keep the system's output as close as possible to a fixed set-point (to keep the system in an operating point), in the presence of disturbances. Such disturbances are e.g. measurement noise, additional inputs, etc.
  E.g. we want to maintain the temperature inside a room at 17 degrees.

- *The tracking problem:* The aim of tracking control is to let the output and/or the states track a desired trajectory or reference signal as close as possible.
  E.g. We desire a robot arm that follows as close as possible a certain path in space.

Generically, both problems must be solved simultaneously: we would like to track a reference trajectory/signal (including a good steady state accuracy), even if there are disturbances and process uncertainties.
The goal of this seminar is to present a method for designing control systems. The emphasis will be on control laws and control configurations that are frequently used in today's industrial applications.

1.2 Development of a Control System

A classical procedure to develop a suitable control system is shown in Figure 1.1. From this flow-diagram, it follows that different activities need to be done in a well-defined sequence. Although the specific aspects will be explained in the next sections, the various activities are shortly described here:

- *Open Loop Analysis:* what kind of system is considered?

  Initially, the system to be controlled needs to be known, at least mathematically. Since a mathematical model of the system is assumed to be available (by means of physical modelling, eye-ball methods or identification), the system can easily be analysed in the frequency domain and the time domain.

- *Performance Specifications:* how is the system required to behave?

Once the system is analysed, the desired performance must be expressed in terms of the different performance measures that are chosen. The importance of a certain performance measure depends on the type of control problem to solve. We must stress that control is not the one and only way to achieve some goals. Analysis of the problem can reveal some structural design problems in the system under consideration. Again referring to the heating problem, the inside temperature can be controlled such that it is not very sensitive for perturbations in the outside temperature. But one might as well think of putting a better thermal isolation such that the problem will not rise that easily.

- *Control Configuration:* which signal(s) are used to calculate the control signal?

  Depending on the plant, the desired performance specifications and the allowed complexity of the control system, a control configuration must be chosen. Depending on the type and the number of input signal(s) to the controller different configurations are recognised.

- *Control Law:* which algorithm is used to calculate the control signal?

  Once the control configuration is fixed, the control law must be chosen. When the designer is experienced and familiar with a specific control law, it is good practice to choose that law first!

- *Parameter Design (Tuning):* which are the parameters of the algorithm to calculate the control signal?

  Once a control law is chosen, the different control parameters must be tuned. For each control law, different quantitative techniques and/or qualitative rules are known.

- *Evaluation:* how will the controlled system behave in theory?

  As soon as the controller is tuned on a computer, the controlled system must be evaluated by means of e.g. test-simulations. When the specifications are met, the controller can be implemented. In the other case, when not all specifications are met, different actions may be undertaken, respecting the following order:

  - Change the control law, for example by adding or changing a controller parameter (e.g. from PI- to PID- or PD-control).
  - Change the control configuration.
  - Relax the design specifications.

- *Implementation* and *Verification:* how will the control system be realized? How does the controlled system behave in practice?

  The controller will be implemented and one will verify whether the system is controlled as expected.

From the foregoing analysis, one can conclude that the classical controller design process is an iterative process. In the next sections, the different steps are described in detail. Besides this classical way of designing a control system, also a new 'mechatronic' way of design is under development. The main difference with the sequential design lies in the fact that both system design and control design aspects are treated simultaneously. This is however not treated in this text.

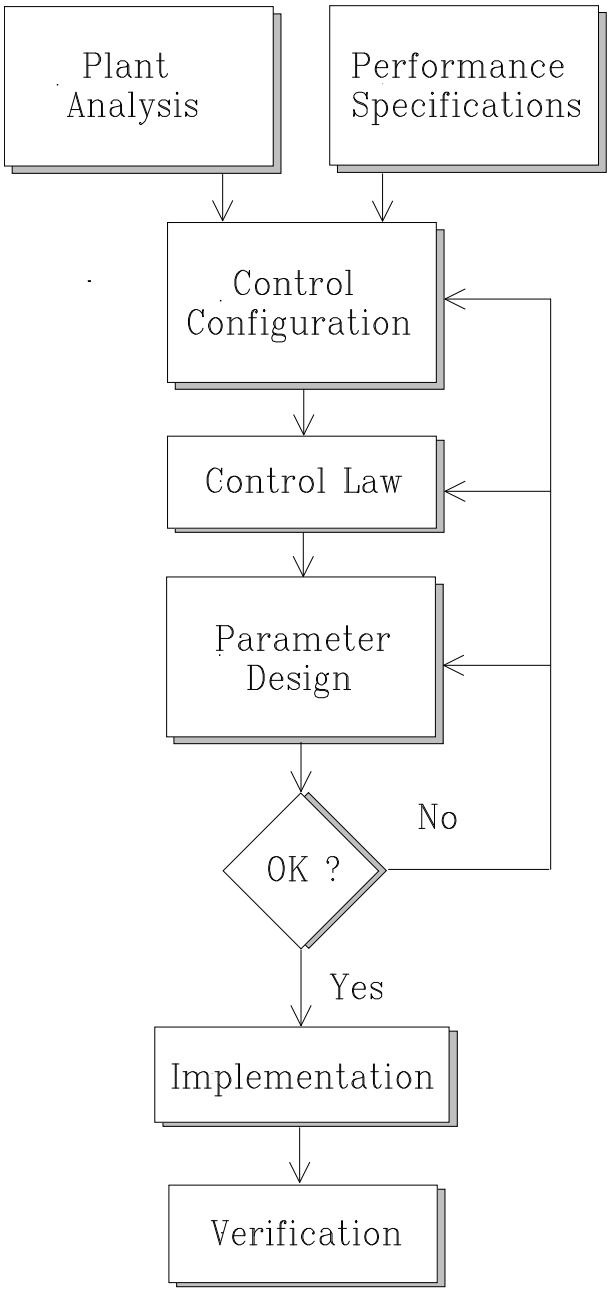


Figure 1.1: A possible procedure to develop a control system.

Chapter 2

Plant Modelling

2.1 Introduction

In order to be able to design a controller for the system under concern, we must know which system we are dealing with. Two different methods exist to obtain a model (in the form of a transfer function): physical modelling and identification.

2.2 Physical Modelling

One way to obtain a model of a given plant is to use known physical laws and relationships that describe the working principle of the system. Structural methods to derive the necessary and sufficient set of governing equations exist, but this is beyond the scope of the text. The advantage is that, *if* a complete description is obtained, it leads to a better comprehension of the system. The mathematical formulation of the system (e.g. a transfer function) can be interpreted from physical backgrounds, which has unavoidably many benefits.
In such a description one gets a lot of variables (constants) that must be measured or found somehow, like masses, frictions, dimensions... In most of the cases the resulting models are nonlinear. The existence of such nonlinear models is not a drawback from physical modelling, but is in first place a property of the physical system itself. A lot of analysis tools exist for controller design purposes, but most of them are only valid for linear models. So, in order to be able to use linear controller design, a linearisation of the nonlinear model should be done first.
Since in regulator problems the deviation from the setpoint value won't be much (that's the goal at least!), one can often assume that the behaviour in that region will be linear. Using series expansion, the nonlinear functions will then be replaced with their linear counterparts. In tracking problems, the nonlinearity may be more visible when the tracking region is large. A solution can exist in designing various controllers for some different 'operation regions'. Also nonlinear control schemes can be examined, this is not the intention of this seminar though!

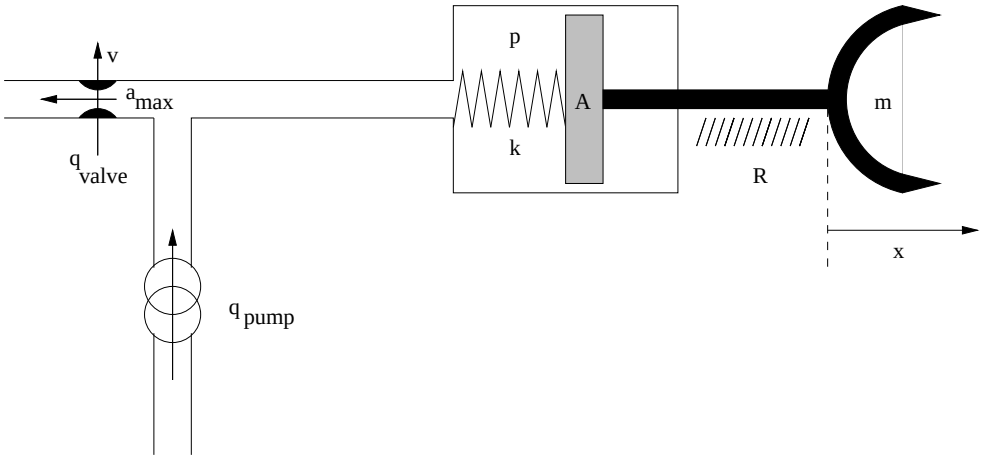


Figure 2.1: Ideal model of an Hydraulic Positioning System.

2.2.1 An example: A hydraulic positioning tool for plastic colouring purposes

In Figure 2.1 a hydraulic positioning system is shown. The idea is to control the position (distance x) of the mass m. The mass is positioned with a hydraulic cylinder (pressure p and surface A). The spring k is inserted into the cylinder in order to be able to return the cylinder when pressure decreases. A viscous friction R is present. The pressure is maintained by a hydraulic oil pump which delivers a flow q_{pump}. A valve V with percentual aperture v (maximum absolute aperture is denoted as a_{max}) is used to control the pressure and thus the position.

First look for the relevant physical equations. In the majority of the cases this is the most difficult step.

- Force balance of mass m:

$$m\ddot{x} = -R\dot{x} - kx + pA \tag{2.1}$$

- Flow balance for an isotherm reversible process is given by:

$$\sum \text{Flow}_{\text{in}} = \sum \text{Flow}_{\text{out}} + \text{accumulated_flow}$$

Hence

$$q\text{pump} = q_{\text{valve}} + A\dot{x} + \frac{V_1(x)}{\beta}\dot{p} \tag{2.2}$$

where all these (and the following) physical values are in S.I. (i.e. q $[\frac{m^3}{s}]$, and β $[\frac{kg}{m \cdot s}]$).
The second term in the right half of the equation represents the increased volume due to a change in postion x. The third term represents a mass accumulation

which is reflected by an increased pressure and isothermal compressibility index β. $V_1(x)$ is the total instantaneous volume. We assume for simplicity reasons that $V_1(x)$ can be regarded as constant in this equation (in reality $V_1 = V_o + Ax$).

- The flow in the valve can be written as:

$$q_{valve} = C_d \frac{a_{max} v}{100} \sqrt{\frac{2p}{\rho}} \tag{2.3}$$

Cd is a proportional constant of the valve, a_{max} is the maximal absolute aperture of the valve, v the percentual aperture and ρ the oil density.

- Temperature T influences the density ρ:

$$\rho(T) = \rho_o(1 - \alpha T)$$

This disturbance will not be taken into account though.

- The pump flow q_{pump} may be assumed to be constant. However it might fluctuate. These changes could be modelled as steps. We will model the pump flow as a pulsating flow. Then it is written as:

$$q_{pump} = \gamma \bar{N}_{pump} \left(1 + \delta sin\left(\frac{\bar{N}_{pump} n}{60} t\right)\right)$$

The sinusoid represents the flow variations due to the pump construction with n chambers. $\bar{N}_{pump}$ is the nominal number of revolutions per minute. γ and δ are proportional values.

Summarising:

The input variable is the percentual valve aperture v.
The variables to observe are: x, $\dot{x}$ and p.
Disturbances are q_{valve} and temperature T.

Next step is the linearisation of the equations. we notice that the equation that describes q_{valve} is a nonlinear function of p and v. We have:

$$q_{valve} = f(p, v)$$

We'll approximate this function by its first order Taylor expansion in the operation point $(\bar{q}, \bar{v}, \bar{p})$.

$$f(p, v) = f(\bar{p}, \bar{v}) + \left.\frac{\partial f}{\partial v}\right|_{\bar{p},\bar{v}} (v - \bar{v}) + \left.\frac{\partial f}{\partial p}\right|_{\bar{p},\bar{v}} (p - \bar{p})$$

One finds:

$$q_{valve} = \bar{q} + K(v - \bar{v}) + L(p - \bar{p})$$

with

$$K = \left.\frac{\partial f}{\partial v}\right|_{\bar{p},\bar{v}} = Cd\frac{a_{max}}{100}\sqrt{\frac{2\bar{p}}{\rho}} \tag{2.4}$$

$$L = \left.\frac{\partial f}{\partial p}\right|_{\bar{p},\bar{v}} = Cd\frac{a_{max}\bar{v}}{100}\sqrt{\frac{1}{2\rho\bar{p}}} \tag{2.5}$$

In the operating point $(\bar{q}_{valve}, \bar{v}, \bar{p})$ it is easier to work with deviation variables q'_{valve}, v' and p', which indicate the deviation from the operation point:

$$q'_{valve} = q_{valve} - \bar{q}_{valve}$$

$$p' = p - \bar{p}$$

$$v' = v - \bar{v}$$

In this way the equation is completely linearised:

$$q'_{valve} = Kv' + Lp' \tag{2.6}$$

Also the flow equation can be reduced since in equilibrium state

$$\dot{x} = 0, \dot{p} = 0$$

$$\bar{q}_{valve} = \bar{q}_{pump}$$

Thus:

$$q'_{pump} = q'_{valve} + A\dot{x}' + \frac{V_1}{\beta}\dot{p}' \tag{2.7}$$

The same holds for the movement equation:

$$m\ddot{x}' = -R\dot{x}' - kx' + p'A \tag{2.8}$$

since in equilibrium $k\bar{x} = \bar{p}A$.

Next step is to write the Laplace Transforms of the linearised deviation equations:

If we define: $\mathcal{L}(x'(t)) = X(s)$, $\mathcal{L}(p'(t)) = P(S)$, $\mathcal{L}(v'(t)) = V(s)$, $\mathcal{L}(q'_{valve}(t)) = Q_{valve}(s)$ and $\mathcal{L}(q'_{pump}(t)) = Q_{pump}(s)$, we obtain from Eqn.(2.6)

$$Q_{valve}(s) = KV(s) + LP(s),$$

from Eqn.(2.7)

$$Q_{pump}(s) = Q_{valve}(s) + AsX(s) + \frac{V_1}{\beta}sP(s),$$

and from Eqn.(2.8)

$$ms^2X(s) = -RsX(s) - kX(s) + P(s)A.$$

The transfer functions from input $V(s)$ and disturbance Q_{pump} towards $X(s)$ can now easily be derived:

$$H(s) = \frac{X(s)}{V(s)} = \frac{-AK}{\left(\frac{V_1}{\beta}m\right)s^3 + \left(Lm + \frac{V_1R}{\beta}\right)s^2 + \left(LR + \frac{V_1k}{\beta} + A^2\right)s + kL}$$

$$H_{dist}(s) = \frac{X(s)}{Q_{pump}(s)} = -\frac{1}{K}H(s)$$

$$X(s) = H(s)V(s) + H_{dist}(s)Q_{pump}(s)$$

We assume real values for the parameters:

- $A = 0.01m^2$
- $m = 30kg$
- $k = 40000\frac{N}{m}$
- $R = 20000\frac{N \cdot s}{m}$
- $Cd = 0.61$
- $a_{max} = \frac{\pi}{4}0.006^2m^2$
- $\rho = 900\frac{kg}{m^3}$
- $V_1 = 3l$
- $\beta = 400MPa$

- $\gamma = 0.0172 \frac{l}{min \cdot rpm}$
- $\delta = 0.15$
- $n = 15$

In steady state one may drop the derivatives, and one obtains the static equations:

$$q_{valve} = Cd \frac{a_{max} v}{100} \sqrt{\frac{2p}{\rho}}$$

$$q_{valve} = q_{pump}$$

$$kx = pA$$

For a pump with $N = 2000 rpm$ and valve opening $v = 50\%$, one calculates easily that in steady state:

- $\bar{p} = 20 bar$
- $\bar{Q}_{valve} = 34.5 \frac{l}{min}$
- $\bar{x} = 0.5 m$

If we take this steady state as operating point, K and L can be calculated:

$$K = Cd \frac{a_{max}}{100} \sqrt{\frac{2}{\rho} \bar{p}} = 1.15 \cdot 10^{-5} \frac{m^3}{s}$$

$$L = Cd \frac{a_{max} \bar{v}}{100} \sqrt{\frac{1}{2 \rho \bar{p}}} = 1.44 \cdot 10^{-10} \frac{m^3}{s \cdot Pa}$$

The linearised transfer function is now:

$$H(s) = \frac{-1.15 \cdot 10^{-7}}{2.25 \cdot 10^{-10} s^3 + 1.543 \cdot 10^{-7} s^2 + 1.0317 \cdot 10^{-4} s + 5.7482 \cdot 10^{-6}}$$

First the nonlinearised system was simulated in Simulink. This simulink diagram is shown in Figure 2.2.

A step-response of this positioning system to a change in valve opening together with the approximated linearised response is given in Figure 2.3.

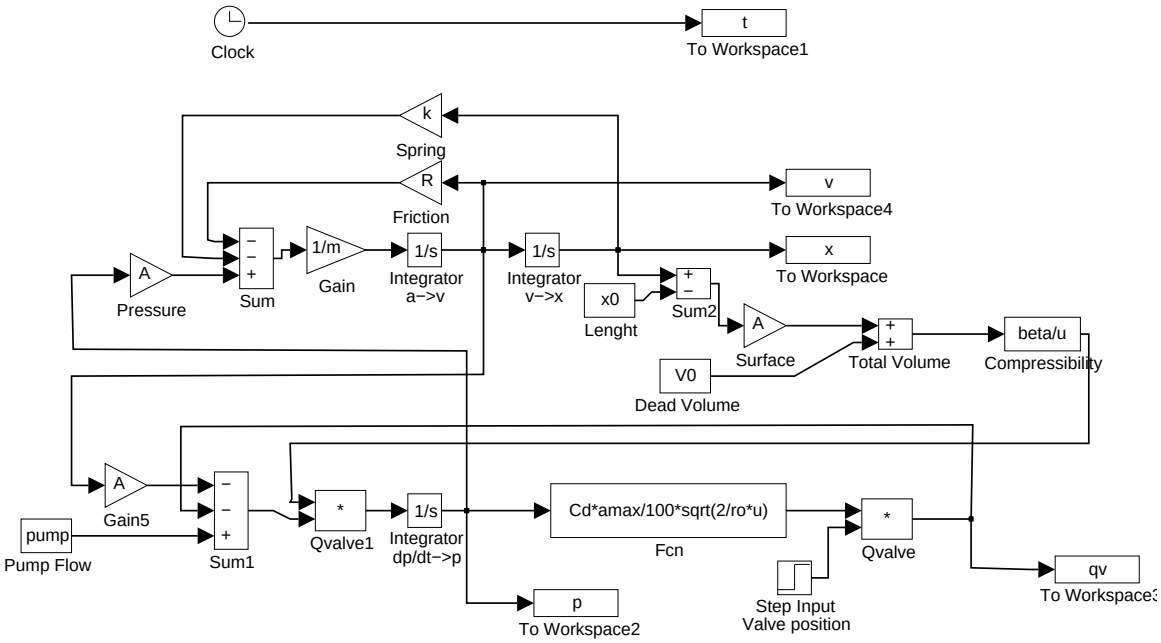


Figure 2.2: Simulink Model of Hydraulic Positioning System.

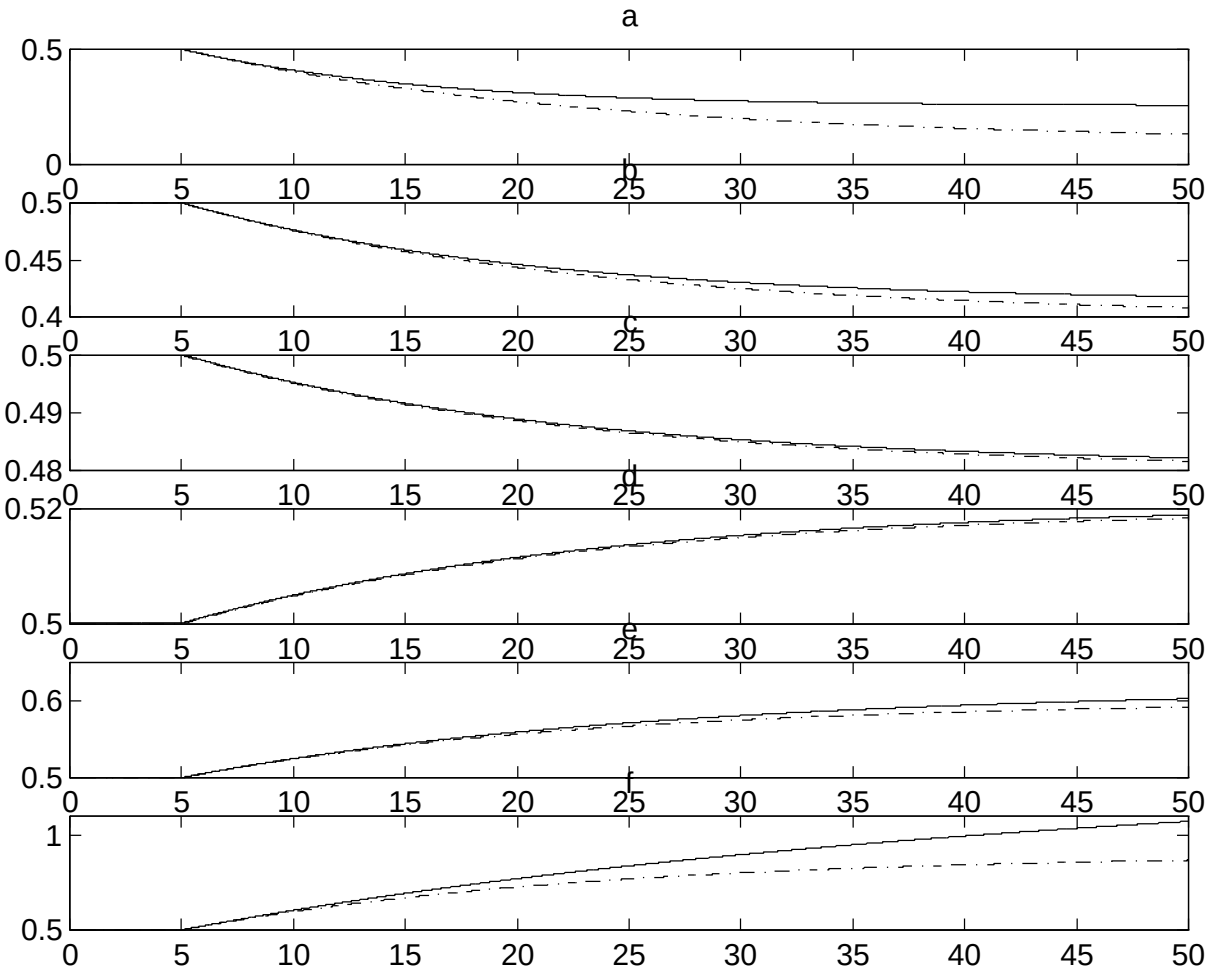


Figure 2.3: Step Response of the Hydraulic Positioning System to a change in valve position. The step takes place after 5 seconds and starts at 50% valve opening. The response is given in solid, while the linearised response is given in dashed. 0.5m is added to the linearised response, which corresponds with the operating point. Notice the deviation from the true response when the input gets big. Notice the nonlinearity in the true response. The step input is a) 70%, b) 55%, c) 51%, d) 49%, e) 45%, f) 30%.

2.3 Identification

2.3.1 Introduction

A completely different way to obtain a system model, is to use identification methods. This is called 'black box' modelling, since one is not interested anymore in all internal working mechanisms, but only in input-output relationships. In very complex systems (e.g. chemical process plants etc.) this can be the only way to obtain a somewhat decent model. The procedure is to gather input-output data sets and to try to fit a mathematical model that relates the output signal with the input signal. This method is extensively treated in the course on 'Identification and Multi-variable Systems'.
Some graphical identification methods that are quite common in industrial practices will be presented here. These so-called *eye-ball methods* are very quick methods which deliver a first-order or second-order system, based on a simple step-test. They're susceptible to noise, and will hardly ever give as good results as numerical methods. They are very handy though for having a quick idea about your system. The idea is to interpret the system response as if it were a first order or second order system. Higher order dynamics are neglected, and this can give a good result if the remaining time constants are very small compared with the two dominant ones.
The model is formed after the evaluation of a step test. Depending on the type of system, a response to a step input can differ significantly. In Figure 2.4 various typical responses are shown. Notice especially the second order under-damped response (a) and the first order or second order over-damped response (b). For a second order response, the transfer function $G(s)$ can be written as:

$$G(s) = \frac{K\omega^2 e^{-\theta s}}{s^2 + 2\zeta\omega s + \omega^2}$$

For $\zeta > 1$ one can show that the response has no overshoot; it is called an *over-damped system*. For $\zeta < 1$ the system is called *under-damped*, since overshoot will be present.

2.3.2 Graphical fitting of first order processes using step-tests

Tangent Method

If a step input is applied to a first order process, a typical output is shown in Figure 2.5. It is easily found that a first order process with gain K and time constant τ

$$G(s) = \frac{K}{\tau s + 1}$$

can be deduced from the plot. K is the ratio of the steady state value of the response to the step amplitude ($s = 0$). τ is the value of t at which the response is 63% complete. It is also the time at which the tangent of the step response in $t = 0$ intercepts the final value.

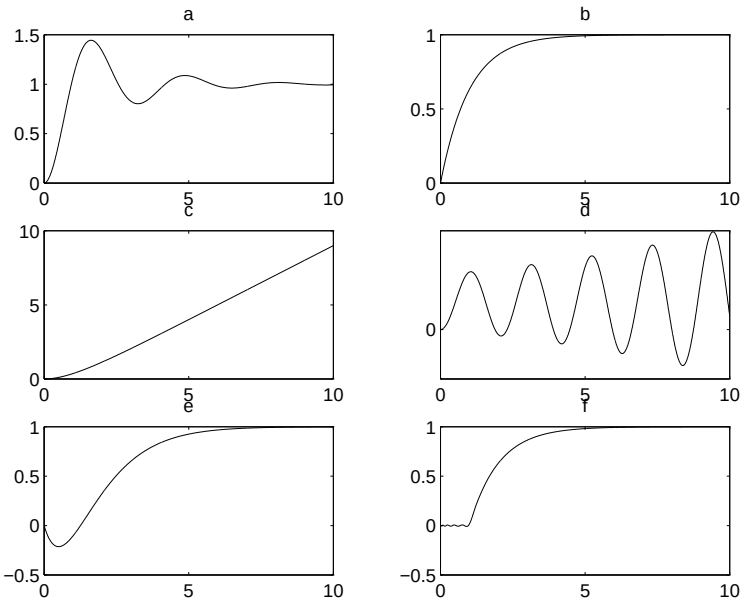


Figure 2.4: Typical Step Responses: a.oscillatory b.first order c.integrator d.unstable e.non minimum phase f.dead-time.

Inflection Method.

Very few experimental plots of the step response show purely first order behaviour because:

- The true process is neither first order nor linear.
- It is difficult to form a perfect step input. Usually process equipment cannot change from one setting into another instantaneously, but must be ramped. If the ramp time is small though compared with the system's time constant, the approximation is reasonable.
- Due to measurement noise or non-uniformities in the variables, the data will be corrupted.

To account for higher order dynamics that are neglected in the simple first-order model, a time delay θ is introduced. This extra time delay will improve the agreement between the experimental data and the obtained model. The model under concern is given by:

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}$$

This model will not only be used for first order plants with delay, but also for higher order systems. To find the delay (see also Figure 2.6):

- The gain K is the steady state value of the step response divided by the input amplitude.
- A tangent is drawn at the point of inflection (steepest slope) of the step response. The intersection of the tangent line and the horizontal axis is the time delay θ .
- Where the tangent intersects the steady-state value of the step response, the time elapsed corresponds to $t = \theta + \tau$. So time constant τ can be found.

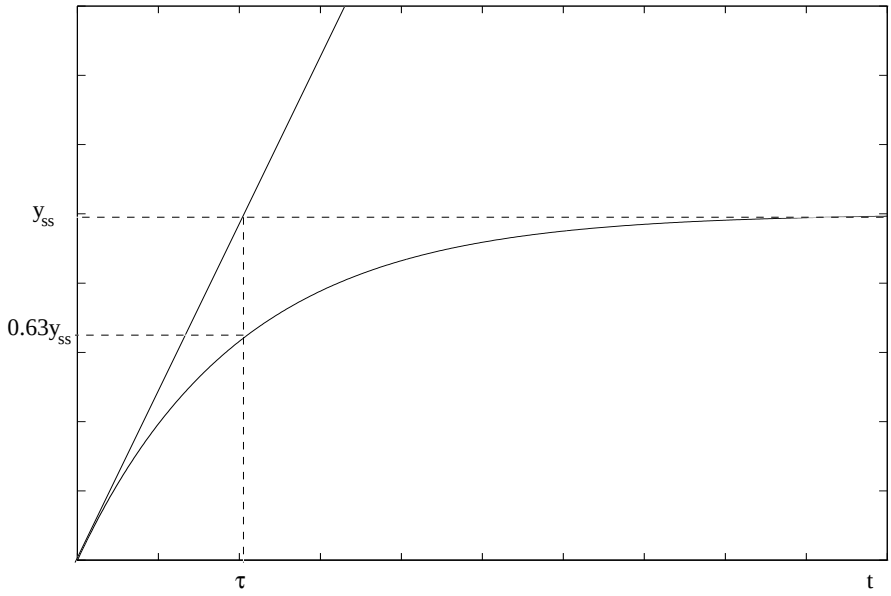


Figure 2.5: Tangent Method.

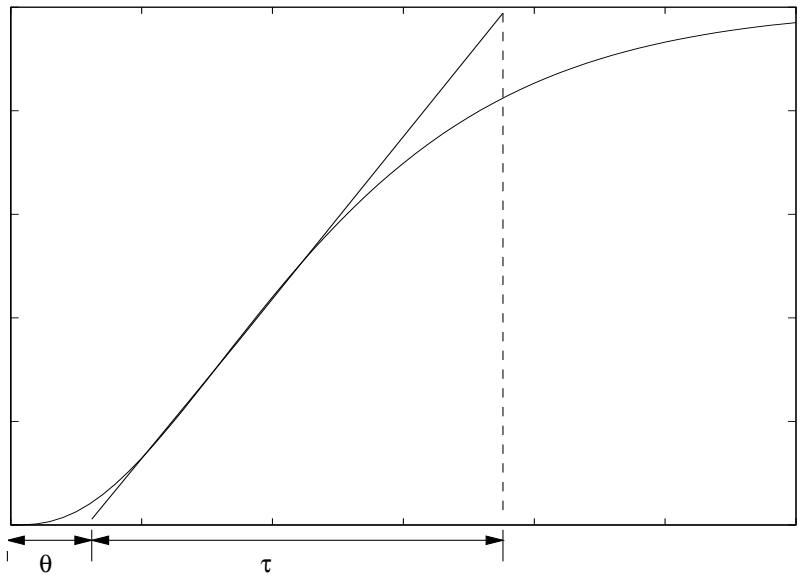


Figure 2.6: Eye Ball Method for first order process with dead-time.

35%-85%-Method

The major disadvantage of the time-delay estimation method discussed above, is the difficulty to find the point of inflection due to measurement noise or small graphs. Therefore an alternative method has been proposed that uses two time instants in order to estimate τ and θ. The time t_1 and t_2 correspond to respectively 35% and 85% response. τ and θ are then found using the following equations:

$$\begin{cases} \theta = 1.3t_1 - 0.29t_2 \\ \tau = 0.67(t_2 - t_1) \end{cases}$$

As an example, the inflection method and the 35%-85%-method are applied to the next system:

$$P_1(s) = \frac{12e^{0.1s}}{(s+1)^2(s+2)(s+3)}$$

With the inflection method, one obtains $K = 2$, $\theta = 1$ and $\tau = 3.1$, which gives the simulated response of Figure 2.7).

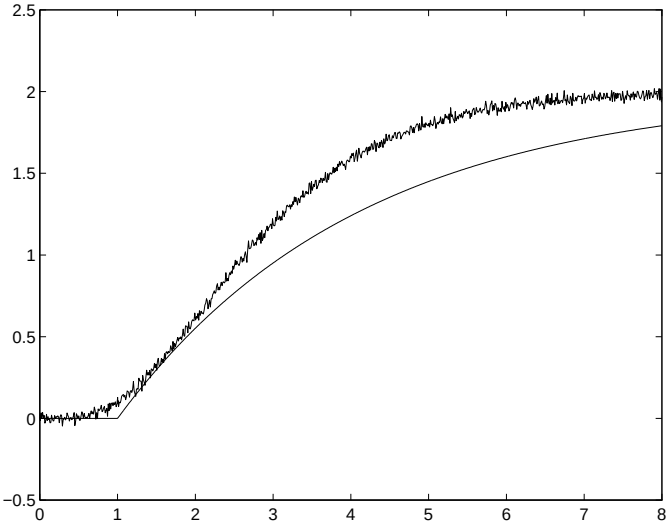


Figure 2.7: Inflection Method. The noisy curve is the measurement, the other one the step response of the identified system.

With the $35\% - 85\%$ -time response method, one gets $t_1 = 2.2$ and $t_2 = 4.3$, resulting in $\tau = 1.41$ and $\theta = 1.61$ with simulation shown in Figure 2.8.

2.3.3 Graphical Fitting of second-order models using step-tests

A better approximation to the step response can be expected using second-order models. Three methods will be treated: Harriot's, Smith's and a model for oscillatory systems. In all these methods the delay is determined by visual inspection or with the inflection method.

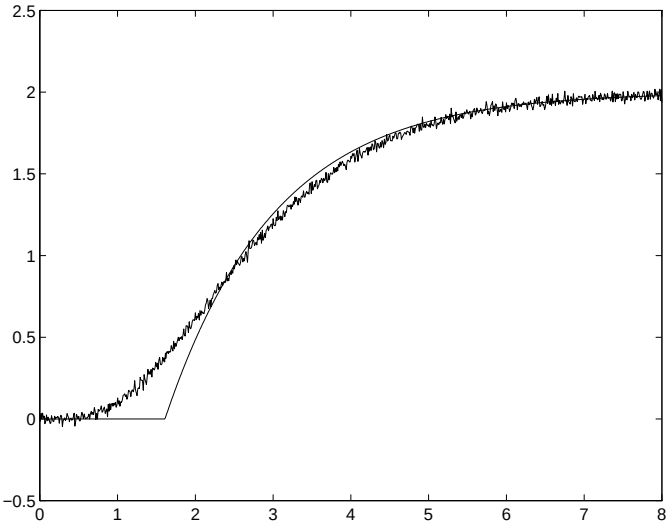


Figure 2.8: 35%-85%-time method. The noisy curve is the measurement, the other one the step response of the identified system.

Harriot's Method

In Harriot's method one tries to fit an over-damped model without numerator dynamics. The transfer function is given by:

$$G(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

In Figure 2.9 Harriot plotted the range of shapes that can occur in the over-damped case with two time constants τ_1 and τ_2. As can be seen in Figure 2.9, all curves seem to pass approximately through the 73%-of-steady-state-value at the same point. This led to a first equation:

$$\tau_1 + \tau_2 = \frac{t_{73\%}}{1.3}$$

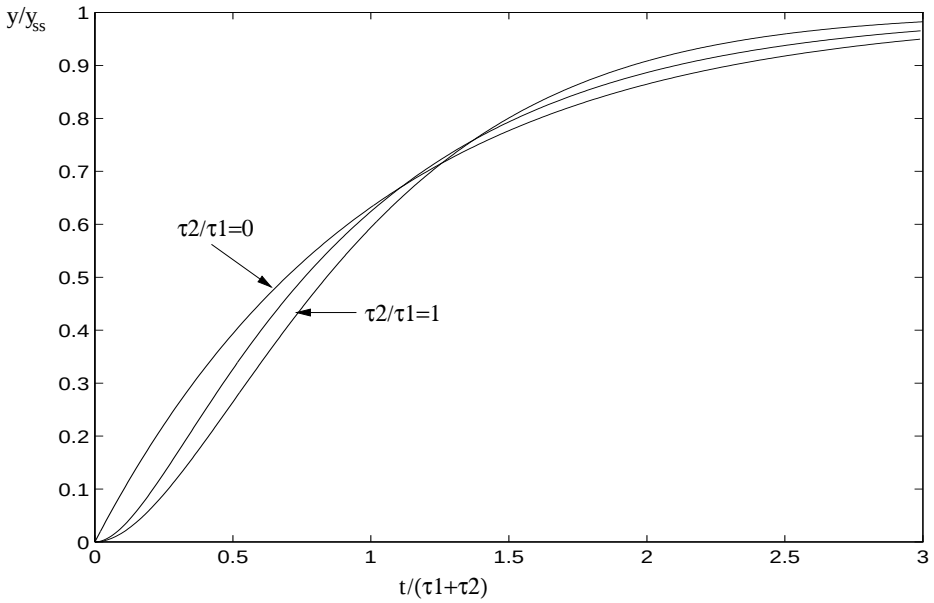


Figure 2.9: Step Response for several over-damped second-order systems

In Figure 2.9 one also notices that the deviation between the different curves is largest for $t/(\tau_1 + \tau_2) = 0.5$. Therefore Harriot plotted the output value at this point, for different values of $\tau_1/(\tau_2 + \tau_1)$. The result can be found in Figure 2.10. Based on the former equation and this Figure, one determines τ_1 and τ_2. If the response is less than 26 % or more than 39 % of the steady state value at $t/(\tau_1 + \tau_2) = 0.5$, this indicates that the process requires a higher order model, or that the process is under-damped.

Harriot's method executed on the previous example, results in: $\theta = 0.6$, $K = 2$, $t_{73\%} = 3.6 - 0.6 = 3.0$, and thus $\tau_1 + \tau_2 = 2.3$. For $t/(\tau_1 + \tau_2) = 0.5$, corresponding with $t = 1.15$. On the graph, one finds for $t = 1.15 + 0.6$ that $y = 0.5/K = 0.25$. In Figure 2.10 one finds finally that $\tau_1 = \tau_2 = 1.15$. The simulation is shown in Figure 2.11.

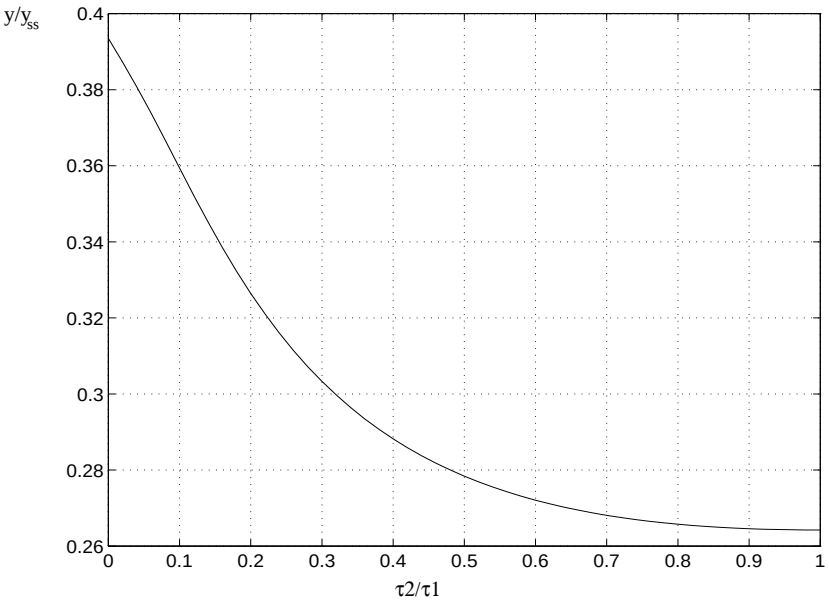


Figure 2.10: Step Response value at $t/(\tau 1 + \tau 2) = 0.5$

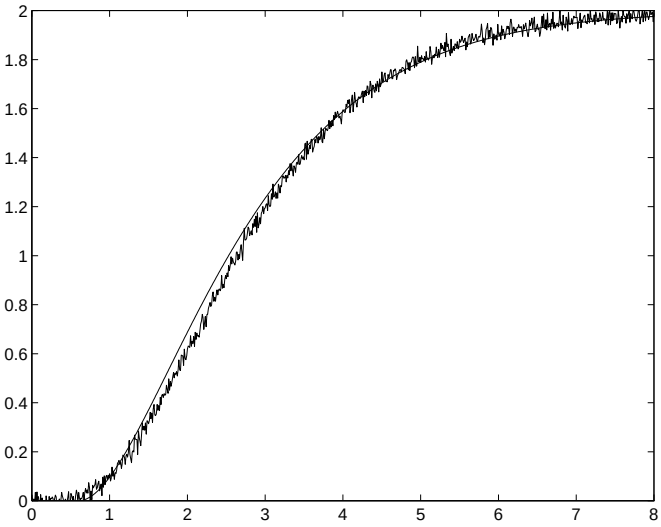


Figure 2.11: Simulation of model obtained with Harriot's method. The noisy curve is the measurement, the other one the step response of the identified system.

Smith's Method

Smith's method uses the model transfer function

$$G(s) = \frac{Ke^{-\theta s}}{\tau^2 s^2 + 2\zeta\tau s + 1}$$

that can be used for both over-damped and under-damped systems. The model makes use of two points : the 20% and the 60% of steady-state value. First the apparent dead time is removed with the inflection method or by visual inspection. Based on Figure 2.12, and the

estimation of $t_{20}\%$ and $t_{60}\%$ one finds τ and ζ. Figure 2.12 was developed by Smith.

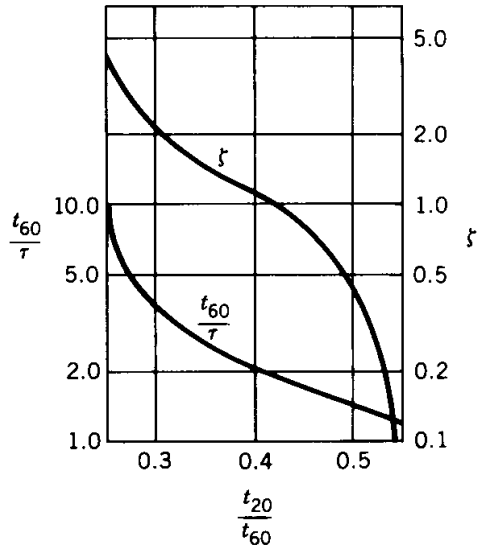


Figure 2.12: Relation between $t_{20\%}$, $t_{60\%}$ and τ, ζ according to Smith.

As a result of this method applied on the same example, one obtains Figure 2.13.

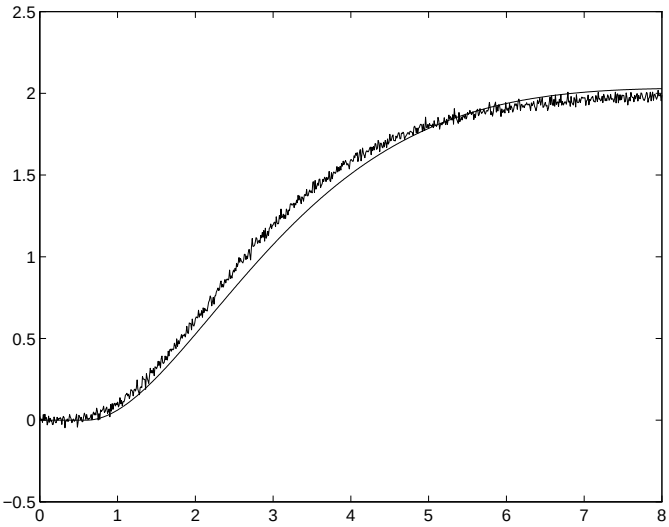


Figure 2.13: Simulation of model obtained with Smith's method. The noisy curve is the measurement, the other one the step response of the identified system.

Model for systems with oscillatory step response

For oscillatory systems the model

$$G(s) = \frac{K\omega^2 e^{-\theta s}}{s^2 + 2\zeta\omega s + \omega^2}$$

can be used. Dead time θ can be found with e.g. the inflection method and K can be found as the ratio between steady state value of the step response and step input value.

From Figure 2.14 one finds logarithmic decrement d and period T_p.

$$d = \frac{a_2}{a_1}$$

Then one calculates ζ and ω:

$$\zeta = \frac{1}{\sqrt{1 + (2\pi/logd)^2}}$$

$$\omega = \frac{2\pi}{T_p\sqrt{1-\zeta^2}}$$

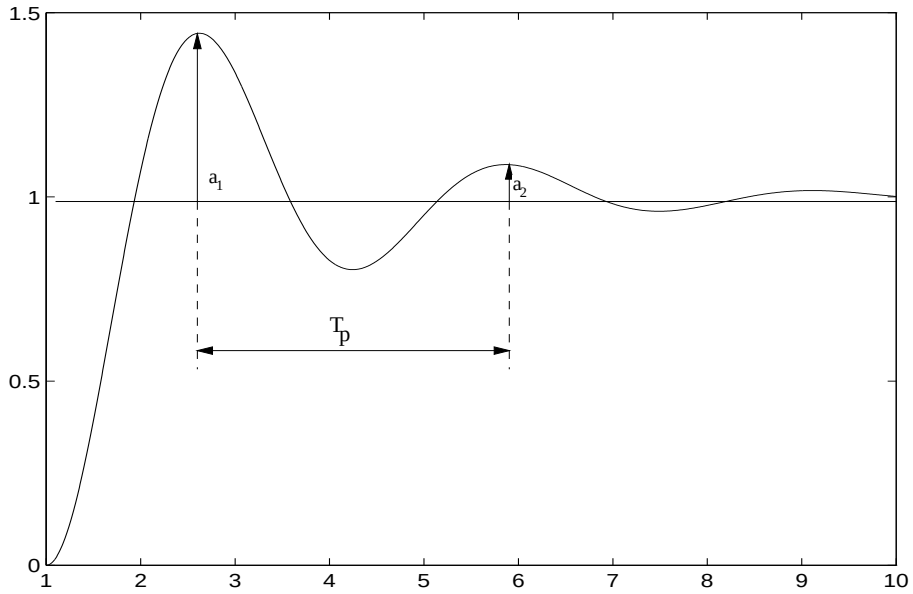


Figure 2.14: Graphical determination of parameters for plants with oscillatory step response.

2.3.4 Model for Integrating Systems

Recall that integrating systems have a transfer function with a pure integrator $1/s$ factor. The response of such a system to a step input, is a ramp. A steady-state value is never reached. Two eye-ball methods are treated.

Impulse Response

If an impulse is applied to an integrating process, a steady state is obtained though. It can be compared to applying a step to a non-integrating process. After normalising the response by dividing by the pulse area (since an ideal impulse can not be made in practice), previous methods for step tests can be applied. The transfer function of the integrating system is then obtained by multiplying with $1/s$.

Step Test

A model can be:

$$G(s) = \frac{K}{s(1+s\tau)}e^{-s\theta}$$

K and $(\theta + \tau)$ can be found as indicated in Figure 2.15. This gives two equations for 3 unknowns. The response of the given system to a step input, can be found as (by calculation of the inverse Laplace transform):

$$y(t) = K(t - \theta - \tau(1 - e^{-(t-\theta)/\tau}))$$

This is the missing equation. If one knows one y(t) value, θ and τ can be calculated.

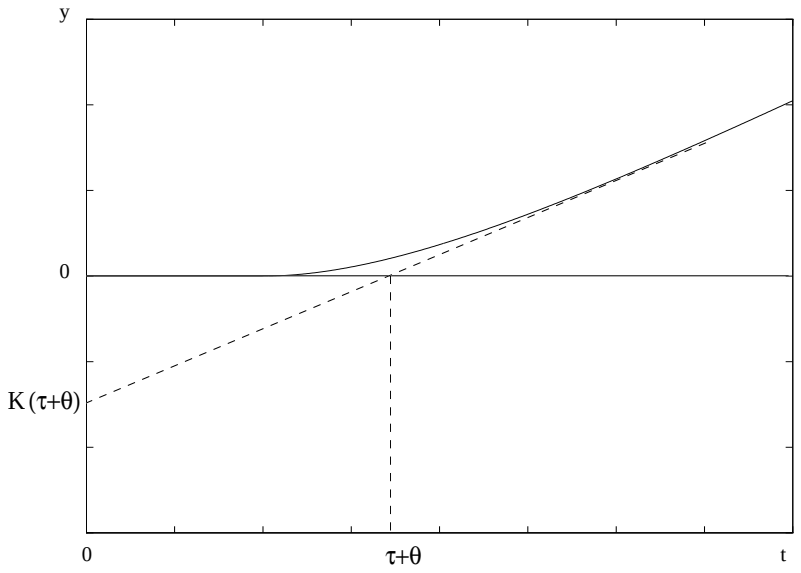


Figure 2.15: Graphical determination of the parameters for an integrating process.

Chapter 3

Plant Analysis

For control design purposes, we're mainly interested in dynamic analysis of the plant. This means that we are not only interested in the steady-state response of a plant. Also the transient response is of great importance. So, when we refer to the plant's dynamics, we are talking about the transients that occur in a process output, caused by a change in process input.

A plant is said to be static when a change in input variable results in an immediate change in output value - which however does not change anymore after that very moment, if the input also remains constant. No transient response occurs.
Nearly all processes have dynamics, but some of them can be neglected in the presence of other processes. For example, in a slow thermal process, in which a room is heated, the opening of some valves in the burner can be regarded as static since their transient response is so fast compared with the heating so that this aspect can be regarded as an immediate transition.

In classical analysis, various characteristics are used to analyse a plant. Depending on the needs, different plots are used to visualise specific properties:

1. Pole-zero plots express stability properties.
2. Nyquist and Bode plots can be used to study the influence of sinusoidal inputs. Also stability can be analysed.
3. Time responses can be used to characterise the plant's behaviour for specific input signals.

Although each plot visualises its related properties quantitatively, it also predicts various other plant properties qualitatively. In fact, the different plots can be seen as complementary information (Figure 3.1).

3.1 Pole-Zero analysis

In pole-zero analysis, the positions (in the complex plane) of the poles and the zeros of the plant's transfer function are investigated. It can be shown that many open loop and closed loop properties can be predicted qualitatively and quantitatively by analysing these poles and zeros.
The transfer function $P(s)$ of a linear system is a complex-valued, rational function $P(.)$ of

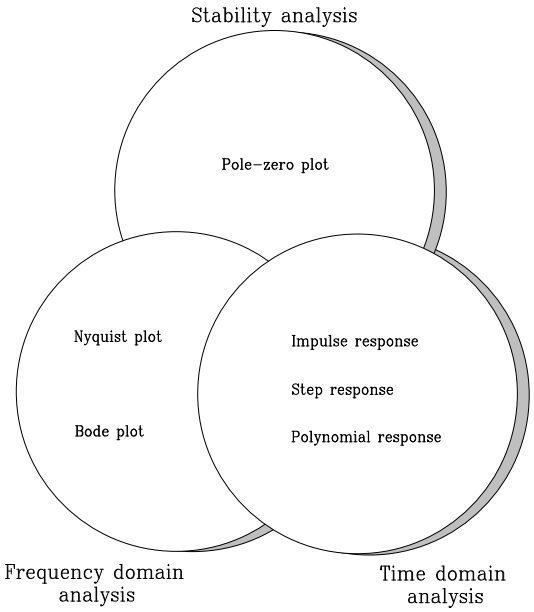


Figure 3.1: Complementary plots to visualise specific plant properties.

the complex variable s. This transfer function relates the Laplace transform $U(s)$ of the input signal $u(t)$ to the Laplace transform $Y(s)$ of the output signal $y(t)$ by:

$$Y(s) = P(s)U(s)$$

For each input signal $u(t)$, the output $y(t)$ can be calculated using the Laplace transform and its inverse.
Assume a dynamical system with one input and one output (Single Input, Single Output or SISO) of which the transfer function $P(s)$ is given by:

$$P(s) = \frac{num(s)}{den(s)}$$

The poles p_i are the roots of the polynomial $den(s)$, while the zeros z_j are the roots of the polynomial $num(s)$. Notice that the poles and zeros are complex numbers!
To allow quick analysis of some open loop and closed loop behaviour, these poles and zeros are drawn in the complex plane. By convention, the poles are indicated by $\times$-marks, the zeros by (o)-marks.
By analysing the positions of the poles, some interesting plant properties can be derived:

- Each pole-(pair) corresponds with a 'normal' mode of the plant. If one applies a Dirac-impulse to the system, the response can be written as a linear combination of the different modes. A pole p_i results in a mode $e^{p_i t}$ in the time domain. The real part of the pole corresponds with an exponential decay (increase), while the imaginary part results in oscillation: a pole-pair $p_i = \sigma_i \pm j\omega_i$ results in time domain in $e^{\sigma_i t} sin(w_i t)$.
- By analysing the mode $e^{p_i t}$, one sees that the pole with the largest real part will result in the slowest decay (increase). This pole is called a dominant pole, since the speed of

a system is limited by the slowest pole (it 'dominates' the response).
In some occasions a higher order system is reduced to a low-order system, usually a second-order system. As an approximation of the original system, one chooses the poles of the low-order system to be equal to the dominant poles of the high-order system. This is a good approximation since the slower poles determine basically the behaviour.

- The 'most right' pole (the least stable) determines the stability degree σ of the plant $P(s)$:

$$\sigma = \min_i [-\text{real}(p_i)]$$

Negative values of σ correspond to an unstable plant, while positive values correspond to a stable one. The absolute stability of a plant grows with σ. When σ decreases, the velocity of the system decreases also.

- Furthermore, the stability ratio σ_r of two stable plants $P_1(s)$ and $P_2(s)$, with respective stability degrees σ_1 and σ_2, is defined as:

$$\sigma_r = \frac{\sigma_1}{\sigma_2}$$

If σ_r is larger than one, plant $P_1(s)$ is the most stable one, if σ_r is smaller than one, plant $P_2(s)$ is the most stable one.

- Another important property of stable plants, the damping degree δ, is determined by the 'most diagonal' pole (the least damped):

$$\delta = \min_i \mid \frac{\text{real}(p_i)}{\text{imag}(p_i)} \mid$$

If δ decreases, the velocity of the system increases (positive effect), but the tendency to oscillate increases (negative effect).

But not only the poles determine the plants properties, the plants behaviour is also influenced by the position of the zeros.
If one or more zeros of the plant are in the right-half plane, the system is said to be non-minimum phase. In all other cases, the system is a minimum phase system. It can be proven that only non-minimum phase systems can give rise to undershoot.

Example 1 *Consider the following plants (normalised DC-gain):*

$$P_1(s) = \frac{6}{(s+3)(s^2+2s+2)}$$

$$P_2(s) = \frac{0.2}{(s+0.1)(s^2+2s+2)}$$

$$P_3(s) = \frac{90}{(s+3)(s^2+2s+30)}$$

$$P_4(s) = \frac{18(s+5)(s+1)}{(s+3)(s^2+2s+30)}$$

$$P_5(s) = \frac{18(s+5)(1-s)}{(s+3)(s^2+2s+30)}$$

$P_1(s)$ is a third order system. By modifying the real pole of $P_1(s)$ in $(P_2(s))$ or the complex conjugated poles of $P_1(s)$ in $(P_3(s))$, the stability degree and the damping degree are changed. However, the impulse response and the step response also change by adding zeros. Depending on the positions of the zeros in the complex plane, different properties are observed ($P_4(s)$ has only zeros in the LHP, $P_5(s)$ has a zero in the RHP).
For all plants the pole-zero plots, together with the corresponding impulse responses and step responses are for convenience shown in the Figures 3.2, 3.3, 3.4, 3.5 and 3.6. The stability degree, the stability ratio (with respect to $P_1(s)$) and the damping degree are given in table 3.1.

	$\sigma(P)$	$\sigma_r(P) = \frac{\sigma(P)}{\sigma(P_1)}$	$\delta(P)$
$P_1(s)$	1	1	1
$P_2(s)$	0.1	0.1	1
$P_3(s)$	1	1	0.186
$P_4(s)$	1	1	0.186
$P_5(s)$	1	1	0.186

Table 3.1: *Stability and damping of the proposed systems*

Notice that complex poles give rise to oscillations and that plants with a smaller damping degree oscillate 'more'. Due to the non-minimum phase character (a zero z_1 at $s = 1$), $P_5(s)$ exhibits undershoot.

△

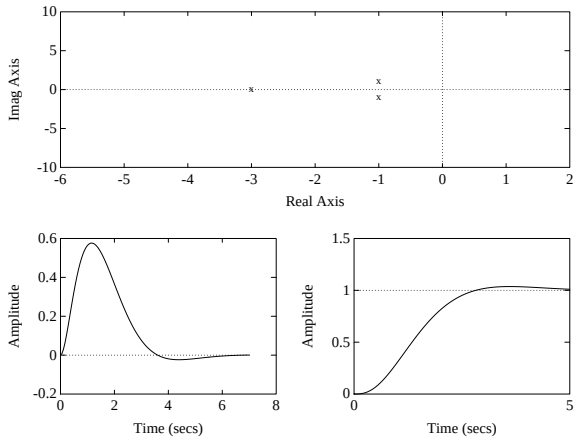


Figure 3.2: Pole-zero plot, impulse response and step response for the plant $P_1(s)$.

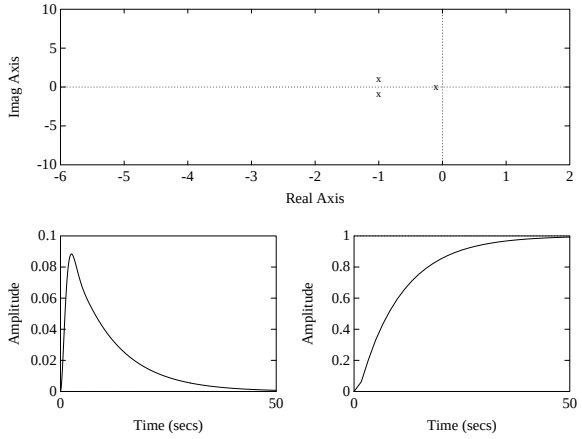


Figure 3.3: Pole-zero plot, impulse response and step response for the plant $P_2(s)$.

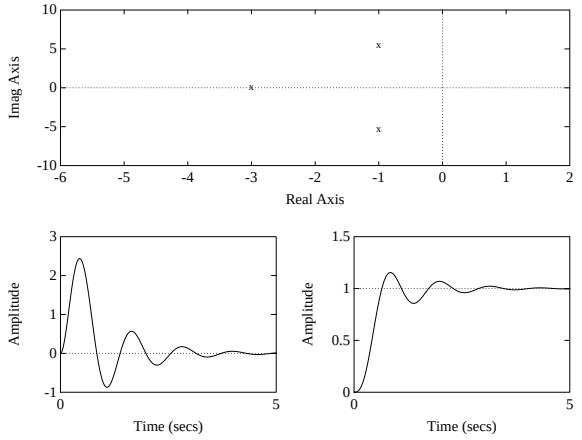


Figure 3.4: Pole-zero plot, impulse response and step response for the plant $P_3(s)$.

-

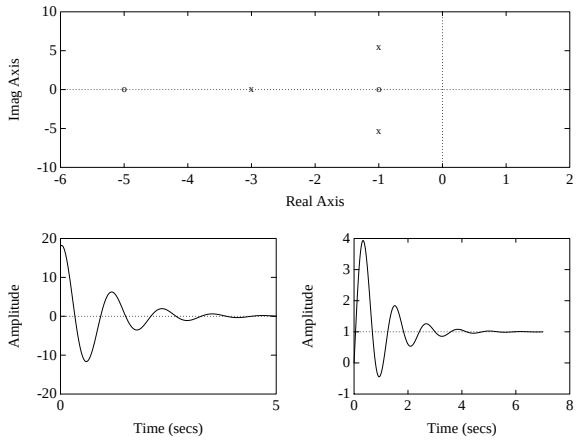


Figure 3.5: Pole-zero plot, impulse response and step response for the plant $P_4(s)$.

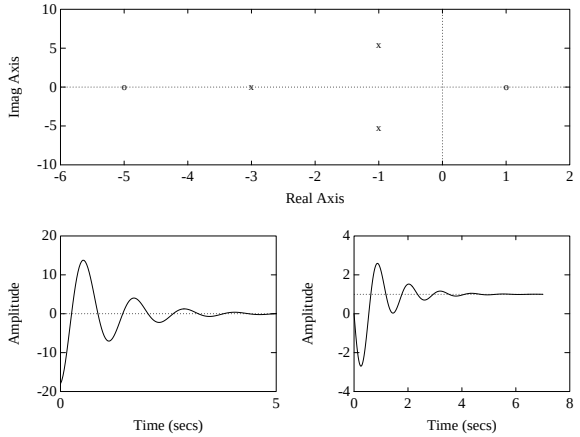


Figure 3.6: Pole-zero plot, impulse response and step response for the plant $P_5(s)$.

3.2 Frequency domain analysis

In the previous Section we saw that for each input signal $u(t)$, the output $y(t)$ can be calculated using the Laplace transform and its inverse.
An equivalent property also holds for the Fourier transform of a plant. For a very broad class of plants, the Fourier transform of the plant $P(s)$ can be calculated as:

$$P(j\omega) = P(s)|_{s=j\omega}$$

Furthermore, it is known that each input signal can be written as a sum of sinusoids (using the Fourier transform). Therefore, for linear systems, it is sufficient to analyse the plant for sinusoidal inputs.
To do the analysis for sinusoidal inputs, consider the input signal $u(t) = sin(\omega^* t)$. It can be proven that the output signal $y(t)$ becomes in steady state (i.e. after the transient behaviour):

$$y(t) = Msin(\omega^* t + \psi)$$

These calculations can be repeated for different frequencies ω^*. The Nyquist plot and the Bode plot are two plots that visualise $P(j\omega^*)$ as a function of ω^*.
The Nyquist plot shows the complex function $P_r + jP_i$ in the complex plane (with ω^* as a parameter), while the Bode plot shows the gain M and the phase shift ψ as a function of the frequency ω^*. Examples are given in Figures 3.7 and 3.8.

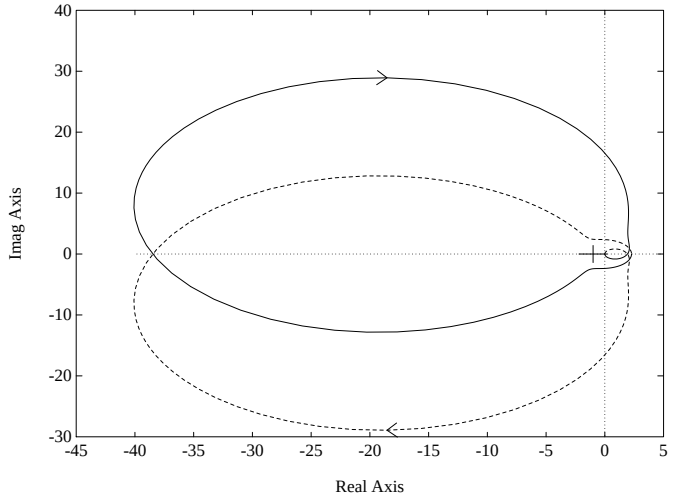


Figure 3.7: he Nyquist curve of $P(s) = \frac{1000(s-100)(s+50)(s+10)(s-2)}{(s+500)(s^2+4s+629)(s+70)(s+2)}$. The full line corresponds to positive frequencies ω, the dashed line to negative frequencies.

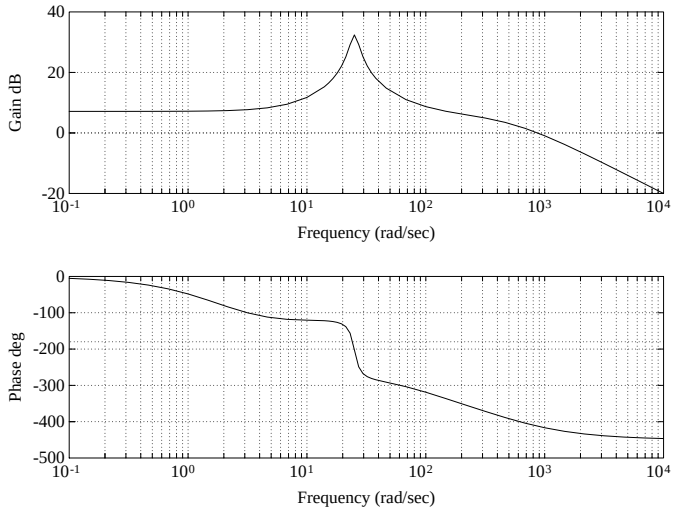


Figure 3.8: The Bode diagram of $P(s) = \frac{1000(s-100)(s+50)(s+10)(s-2)}{(s+500)(s^2+4s+629)(s+70)(s+2)}$. The gain M is given in dB (u), the phase shift ψ in degrees (b).

Notice that a Bode plot can easily be used to predict some other properties:

- A pole gives rise to a decreasing gain in the Bode plot, starting from the frequency that corresponds with the pole.

A pole in the left half plane (LHP) corresponds to a decreasing phase shift, a pole in the right half plane to an increasing phase shift.

- A zero gives rise to an increasing gain in the Bode plot, starting from the frequency that corresponds with the zero.

  A zero in the LHP corresponds to an increasing phase shift, a zero in the RHP to a decreasing phase shift.

One should be extremely careful with Bode plots of unstable systems. The Bode plot exists and will be drawn by simulation software (e.g. Matlab). It has however no real physical meaning, since the slightest transient response results in an ever increasing output signal. Theoretically one could choose initial conditions such that no transient occurs, and the bode plot then shows the steady state response. In reality the system would however become unstable due to inaccuracies, noise, disturbances etc. The question raises how one can check whether a bode plot corresponds with an unstable system. Take care: if one checks gain and phase margin, one is analysing stability of the closed loop system! To check stability of the open loop system, one can take in account the following statements:
For a plant, to be stable, it is required that there is no frequency on the Bode plot where the gain decreases and the phase shift increases!
Further, if there is a frequency where the gain and the phase shift decrease, the plant is guaranteed to be non-minimum phase (thus undershoot is possible!)

3.3 Time domain analysis

Instead of learning about the plant by analysing sinusoidal inputs, some interesting plant properties may also be observed by studying the effect of an impulse input or a step input. For a unit impulse input signal $u(t) = \delta(t)$ (Laplace transform $U(s) = 1$), the Laplace transform of the output $Y(s)$ is:

$$Y(s) = P(s)U(s) = P(s)$$

To calculate the impulse response $y(t)$, it is sufficient to take the inverse Laplace transform of $P(s)$. Notice that:

- Complex conjugated poles $p_{i,i+1} = \sigma \pm j\omega$ give rise to an output term $\exp(\sigma t)\sin(\omega t)$.
- A real pole $p_j = \sigma$ gives rise to an output term $\exp(\sigma t)$

From this, it follows that:

- Effect of $e^{\sigma t}$: a decreasing impulse response envelope requires all poles to be in the LHP, while an increasing impulse response envelope means that there is at least one pole in the RHP.
- Effect of $sin(\omega t)$: an impulse response with an oscillating behaviour (decreasing, stabilising or increasing) requires at least one pair of complex conjugated poles.

However, for many practical plants, it is impossible to apply a unit impulse, but it is easy to apply a step input. In that case, the response $y(t)$ for a unit step $u(t)$ (Laplace transform $U(s) = \frac{1}{s}$) will be observed. The Laplace transform $Y(s)$ is:

$$Y(s) = P(s)\frac{1}{s}$$

From the finite value theorem, the output value for $t \to \infty$ can be calculated:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} P(s)$$

From this, it follows that:

- The output in time domain of a plant with one or more poles at $s = 0$ (integrating terms) goes to ∞ for a step input.
- The output in time domain of a plant with one or more zeros at $s = 0$ (differentiating terms) goes to zero for a step input.
- To have no steady state error ($y(\infty) = u(\infty) = 1$), it is required that $\lim_{s\to 0} P(s) = 1$.

Chapter 4

Performance Specifications

Once the plant properties are known, the control specifications must be written down. In practice, various measures are used to indicate a control system's quality. The most frequently used measures for a classical feedback controlled system (the compensation scheme of Figure 4.1) are given below. By specifying 'desired' values for the different quality measures, the designer expresses the control specifications in an objective way. It must be stressed that in practice the specifications are not always given in the terms that will be used below. Examples can be:

- Control 'oil' temperature such that it is constant at 180 deg, and it may never exceed 190 deg.
- Control the braking force such that the car brakes from 120 kmph till stop in 50 meters without skidding.
- Control a gear box of a car, such that the car drives in a comfortable way.
- Control a paper winding process, such that it is fast, and the paper never breaks.

A first thing to do is convert these specifications in mathematical expressions. At the same time one should analyse which variables must be measured and which ones will be used for actuators on the process.

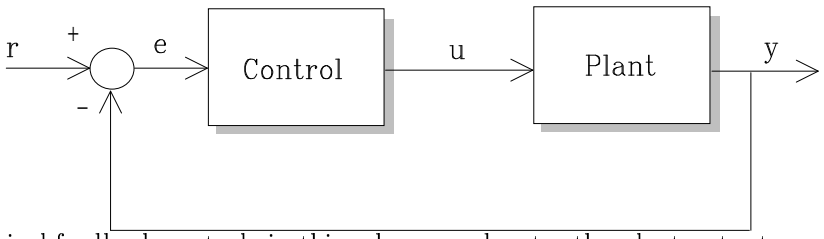


Figure 4.1: Classical feedback control: in this scheme, y denotes the plant output or sensor signal, u denotes the plant input or actuator signal, r denotes the reference or command input signal and e denotes the error signal.

The plant's transfer function is given by $P(s)$, while the transfer function of the controller is given by $C(s)$. The open loop transfer function $L(s)$ is:

$$L(s) = P(s)C(s)$$

the closed loop transfer function $H(s)$ (from r to y) is:

$$H(s) = \frac{L(s)}{1 + L(s)}$$

The transfer function that describes the dynamics between the reference input and the error, is given by:

$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)P(s)}$$

4.1 Closed-loop stability

To guarantee *absolute stability* of the closed loop system $H(s)$, the controller's parameters need to be chosen such that all poles of $H(s)$ are in the LHP. In practice, the stability degree σ of the closed-loop system is required to be positive.
On the other hand, if the loop transfer $L(s)$ itself is stable, the *relative stability* of the closed loop system can be expressed by three different measures:

- The *positive gain margin* (pgm) is the smallest increase in loop gain that results in an unstable closed loop system. A positive gain margin smaller than 0 dB means that the design yields an unstable closed loop system. In practice, a pgm of 2 dB is required in order to guarantee sufficient stability.
- The *negative gain margin* (ngm) is the smallest decrease in loop gain that results in an unstable closed loop system. A negative gain margin smaller than 0 dB means that the design yields an unstable closed loop system. In practice, a ngm of 2 dB is required in order to guarantee sufficient stability.
- The *phase margin* (pm) is the smallest decrease or increase in loop phase that results in an unstable closed loop system. A phase margin lower then zero degrees means that the design yields an unstable closed loop system. In practice, a pm of 45 degrees is required in order to guarantee sufficient stability.

Although it is sufficient to check either phase margin or gain margin, it is advisable to meet both specifications. These measures can be visualised on the Nyquist plot and the Bode plot (Figure 4.2).

4.2 Frequency domain performance

The *bandwidth* (bw) of a system is the smallest frequency at which the closed loop gain is less than -3 dB. In practice, the *cross-over frequency* ω_{co} (the frequency for which the loop gain equals 0 dB) is a good measure for the closed loop system's bandwidth (Figure 4.3).

- *Tracking properties* can be derived by observing

$$H(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)}$$

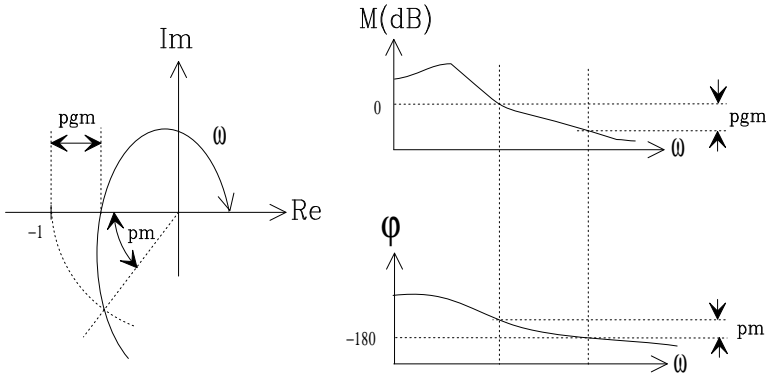


Figure 4.2: Visualising the gain and phase margin on the Nyquist plot (l) and the Bode plot (r).

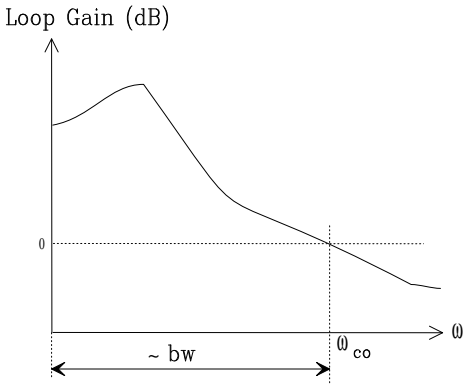


Figure 4.3: Quality measures for frequency domain performance.

For frequencies below the bandwidth, $\mid L(j\omega) \mid \gg 1$ and $1 + L(j\omega) \approx L(j\omega)$, so:

$$H(j\omega) \approx \frac{L(j\omega)}{L(j\omega)} = 1$$

thus the system is able to track these signals. For frequencies above the cross-over frequency, $\mid L(j\omega) \mid \ll 1$ and $1 + L(j\omega) \approx 1$, thus:

$$H(j\omega) \approx L(j\omega) \ll 1$$

which implies that these higher frequencies are not 'followed' anymore: the system seems to be 'insensitive' for them.

- *Output noise compensation* can be studied by observing:

$$H(j\omega) = \frac{1}{1 + L(j\omega)}$$

For frequencies below the bandwidth, $\mid L(j\omega) \mid \gg 1$ and $1 + L(j\omega) \approx L(j\omega)$, so:

$$H(j\omega) \approx \frac{1}{L(j\omega)} \ll 1$$

thus the system suppresses these noise components. For frequencies above the the cross-over frequency, $\mid L(j\omega) \mid \ll 1$ and $1 + L(j\omega) \approx 1$, thus:

$$H(j\omega) \approx 1$$

which implies that these higher frequencies are not 'suppressed' anymore.

4.3 Time domain performance

Different quality measures are used to evaluate the time-domain performance of a control system. The step response quality is expressed by the following measures (Figure 4.4):

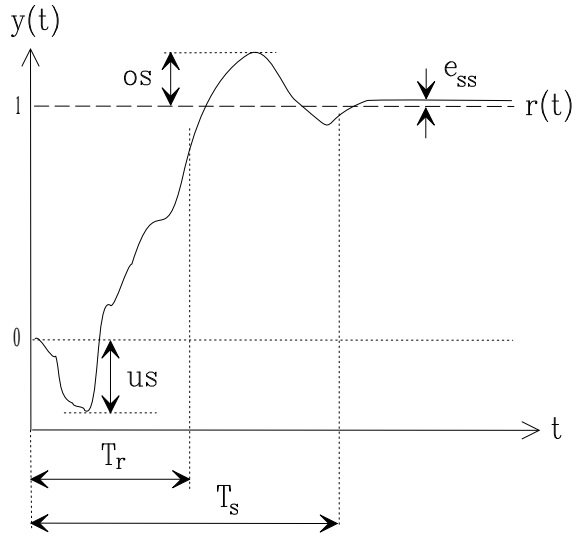


Figure 4.4: Quality measures for time domain performance, using the step response.

- The *steady state error* e_{ss} is a measure for the steady state accuracy of a system. It is the steady state difference between a reference step input and the process output. Since the relation between reference input and error is given by

$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)P(s)}$$

,

one obtains for a step input ($R(s) = 1/s$):

$$e_{ss} = e(\infty) = \lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s) = \lim_{s \to 0} s \frac{1}{s} \frac{1}{(1 + C(s)P(s))}$$

So e_{ss} is also equal to the DC sensitivity, i.e., for small changes in the open loop DC gain, it gives the percentage change in closed loop DC gain divided by the percentage change in the open loop DC gain. For example, if e_{ss} equals 0.1, a 10% change in open loop DC gain will cause a change in the closed loop DC gain of about 1%.

Notice that the e_{ss} is zero when $L(s)$ contains an integral term.

- The *steady state tracking error* $e_{ss,tr}$ is the difference in steady state between a ramp input to the system and a system's output.

  One can calculate the steady state tracking error $es_{ss,tr}$ as

$$e_{ss,tr} = e(\infty) = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = lim_{s\to 0} s\frac{1}{s^2}\frac{1}{1+C(s)P(s)} = \lim_{s\to 0}\frac{1}{sC(s)P(s)}$$

  The *velocity error constant* K_v is defined as

$$K_v = \lim_{s\to 0} sC(s)P(s) = \frac{1}{e_{ss,tr}}$$

- The *rise time* T_r is the smallest time at which the step response is at least 90% of its asymptotic value (which is the closed loop DC gain). It can be verified experimentally that this time is proportional to the inverse of the cross-over frequency ω_{co}: a system with a high cross-over frequency will also have a high *response speed*.

- The *settling time* T_s is the smallest time after which the step response stays within 5% of its asymptotic value.

- The *overshoot os* is the maximum amount by which the step response ever exceeds its asymptotic value, expressed as a percentage of the asymptotic value.

- The *undershoot us* is the maximum amount the step response dips below zero, expressed as a percentage of the asymptotic value of the step response.

- The *required control effort* is the power needed in the actuator to realize the control action that is desired. Controllers can be optimised for minimising energy ($\int u^2 dt$) (e.g. total heat generated in a motor) or for not exceeding a certain value of u (peak control). In general an actuator's price reduces dramatically when less power is needed. Imagine a crane positioning control: if a fast action is desired, speed is limited by the power of the driving motor. It is pointless to get 100N-m out of a 10mN-m motor!

4.4 Sensitivity to Parameter Changes

This topic needs to be studied separately for plant parameter changes and for controller parameter changes.

- Concerning changes in parameters of the plant, the situation is very similar to disturbance-rejection. The idea is to develop a controller that has a reasonable performance despite of changes in the plant parameters (e.g. in a time variant process) or disturbances. Such a controller is called robust. It results that, the larger the feedback loop around the offending parameter, the lower the sensitivity to changes in that parameter.

- The second aspect concerns effects of changes in the controller. Here we have control over the topology, and a design choice can be made to minimise the effects of parameter changes in $C(s)$.

4.5 Conclusions

From the foregoing analysis, the following design criteria for a closed loop control system can be derived:

- In order to guarantee absolute stability, all closed loop poles are required to be in the LHP. If the loop transfer $L(s)$ is stable, a sufficiently high phase margin (positive or negative) and gain margin guarantee a certain degree of stability (relative stability).
- To obtain a high steady-state accuracy, $L(s)$ should have a large DC-gain. This also improves robustness of the controller. By adding integrating elements, this gain becomes ∞.
- To obtain a high bandwidth (good tracking, good disturbance compensation) and a high response speed, the cross-over frequency must be high.
- In order to obtain cheaper actuators, the control effort must be minimised.

It is clear that these design criteria result in design conflicts, so performance trade-offs must be accepted.

4.6 Two methods to reach the goals

Once the performance specifications are formulated, one should think of how to realize them. Two ways exist.

The first method is of considerable importance. One should analyse first whether the plant itself can be optimised, such that some of the specifications are obtained. For example, if one wants to control fuel consumption in a heating process, one could first check whether isolation is done properly. This aspect will not be treated here, but is of considerable importance.

The second method is the design of an appropriate controller that provides necessary actuation on the process such that the goals are reached. This will be the subject of this seminar.

Chapter 5

Control Laws

5.1 Introduction

The purpose of control is to generate input signals to a process (control signals) such that relevant variables in the process behave in a desired way. In output control, only the output variables are relevant (cfr. in state control, not only the outputs but also the system's states are important).
It is clear that we shall try to obtain a desired output by introducing a well-chosen input into our system. If the temperatures drops too much inside, we will put on the heating system. On the other hand, if it gets too warm, we'll close the radiator a bit. Intuitively we relate the input to the system to the deviation of our variable temperature from the desired value. On the road you'll turn the steering wheel when you see you're leaving the road! The input into the system, also called the 'control action', will be function of the deviation error. We will open the radiator more when it is really icy then when the temperature is considerably warm. This function will be called the 'control law'. It is interesting to notice already that some systems need a control input though no set-point error occurs: we need to heat continually a house in order to maintain the same temperature. An output that equals the desired set-point, does not automatically imply a zero control action.
At this very moment we repeat the concept of open-loop and closed-loop systems.
We're talking about open loop if we're applying an input to a system and examine the output. We close the loop when we apply an input to the system in function of the output (e.g. the deviation of the output from a reference value). From the temperature example above (which is a regulator problem) it is easily understood why to close the loop: we like to apply a control action to the system that is function of the set-point error. It is also clear that it is advantageous to measure the variable you want to control - though it is not a must.
When the reference value is varied (tracking), one must take into account that the closed-loop system varies in a considerably different way than the open-loop system. Even more, this will be the goal of the feedback design: to obtain a feedback system that responds faster, in a more stable way, ... than the open loop system. It can be seen that a very slow system can be made faster by applying a very strong control action (and thus requiring more energy). In the heating problem a house can be warmed up 10 degrees in half a minute if you put enough heating power - which will of course result in giant burners and convectors!
Although the impression may be given that the only way to control a system's output, is to apply a function of the deviation error to the input, this is not the one and only truth. If

the system and the disturbances are known completely, one can calculate exactly what input is to be generated to obtain the desired output. If you dispose of a model that describes the relation between heating and temperature, you might calculate how much power you need to put into the house - without measuring the actual temperature. In this case you're only using a feed-forward block instead of a feedback controller.
In general a mixed configuration of feed-forward and feedback might exist. The goal of this seminar is to present a method for designing control systems. The emphasis will be on control laws and control configurations that are frequently used in today's industrial applications.

5.2 On-Off Control

5.2.1 General Working Principle

Maybe the most frequently used controller is the On-Off Controller. In a lot of domestic and industrial applications that are not too critical, on-off control is a possibility. These applications are processes in which some parameters are allowed to change within a certain band without damaging the quality of the process. The advantage of this type of control is the simplicity of the hardware design: the actuator will only be switched on and off. E.g. it is much cheaper to buy a valve that opens or closes, then a valve that can be opened gradually. Examples are maintaining a reference temperature at home, regulating the water level in a vessel, e.o.

We'll explain the working principle of an on-off controller on a first order system with dead-time. The working principle for higher order systems is similar.

The configuration is given in Figure 5.1. The principle is shown in Figure 5.2.

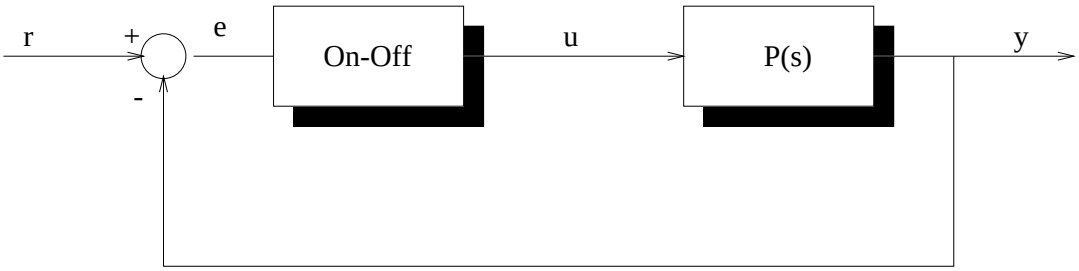


Figure 5.1: On-off control configuration

A good example is the heating of a house. Initially the temperature is at zero, and the burner u is turned on. We remark that after a dead-time θ the inside temperature starts rising. When it is detected that the temperature climbs above the set-point value r, the burner is turned off again. The temperature raises more due to the dead-time θ, where-after it starts decreasing. When it passes the r-limit, the burner is switched on again, so that after a dead-time θ temperature increases again. Due to the fact that the reference is not situated in the middle of the dynamic range, the process needs more time to warm up than to cool. This is reflected in longer on-periods than off-periods.
We notice that the temperature will oscillate between two values y_1 and y_2. With respect to the amplitude and the period of the oscillation, we can say that a smaller dead-time will result in a faster reaction of the process to the controller. Hence the amplitude and period will decrease. With a very small dead-time the amplitude variation will be negligible, but the switching frequency will be high. This can be destructive for the switching devices.

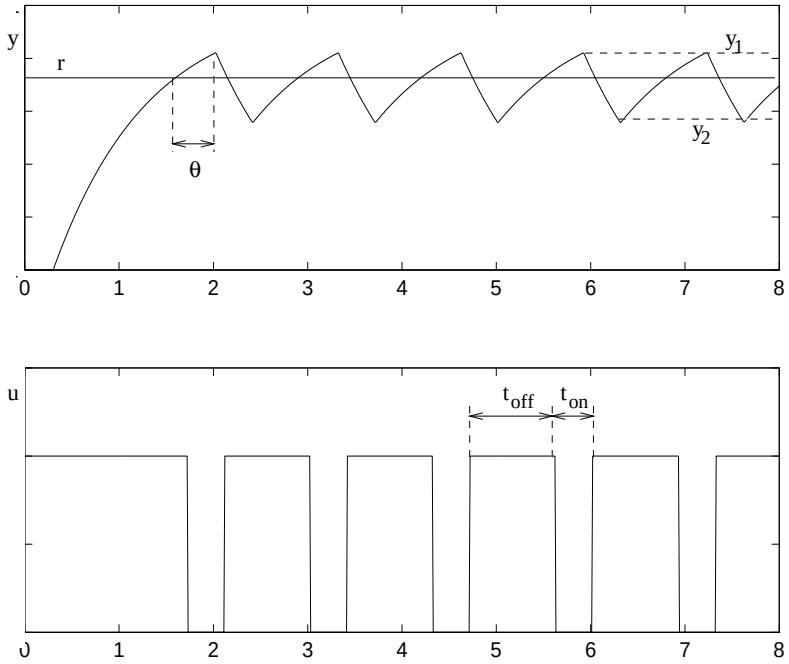


Figure 5.2: Working Principle of on-off controller

5.2.2 Hysteresis

Whenever the process value to be controlled passes the reference level, the actuator is switched. In processes with a short dead-time or in case of measurement noise on the controlled variable, the controller will be switched on and off continuously (Figure 5.3). This is often very harmful for the switching contacts and/or the actuators.

Therefore often hysteresis is introduced such that one switching point is replaced by a high and a low switching point. Only when the process value gets higher than the high switching point, the actuator is turned off. The controller only switches on when the process value decreases under the lower limit (Figure 5.4).
By these means the switching frequency can be limited, though a lower frequency will always result in a larger oscillation amplitude.

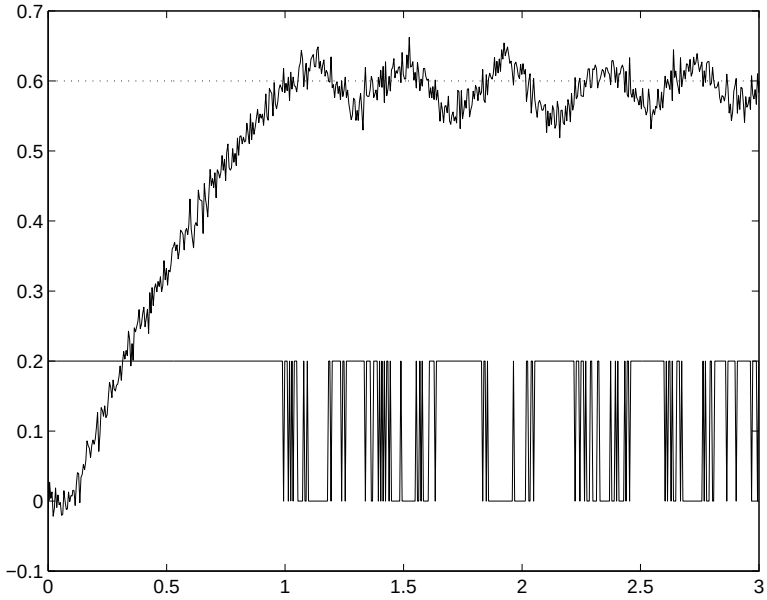


Figure 5.3: The on-off controller with a noisy process. Both process value and control action are shown.

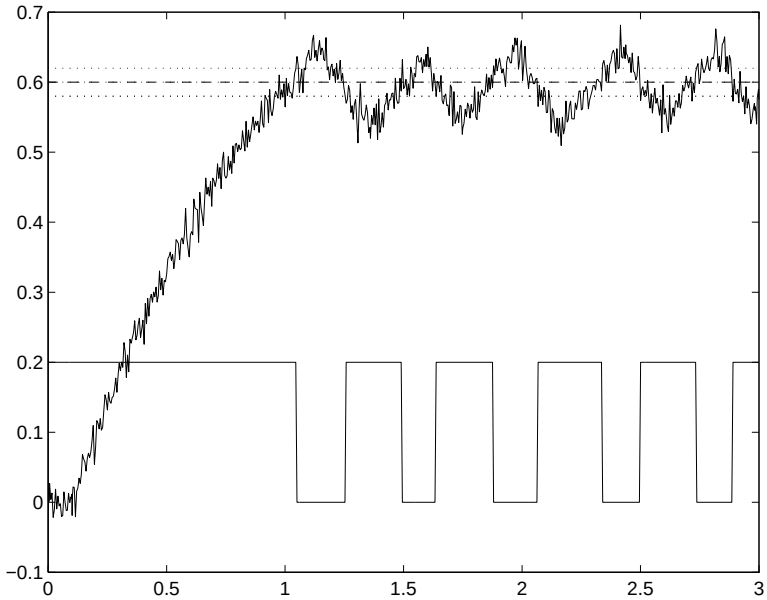


Figure 5.4: Working Principle of on-off controller with hysteresis

5.3 PID-Control

5.3.1 Open Loop performance of a PID-controller : the control law

In industrial environments, the PID-controller is still the most favourite feedback controller: a vast amount of practical experience learns that the PID-controller has a structure which is sufficiently versatile to cope in an excellent way with a multitude of process control problems. Other advantages of the PID-law are the low number of design parameters and the fact that the controller parameters can easily be related to performance measures.
The PID-controller is a linear controller. It transforms the error signal $e(t)$ into the control signal $u(t)$. Since it is a linear controller, it can be studied in the time domain and in the frequency domain.
In the time domain, the ideal control signal looks like:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{d}{dt} e(t) \right)$$

After Laplace transformation, this becomes:

$$U(s) = K_p(1 + \frac{1}{sT_i} + sT_d)E(s)$$

Basically, the controller consists of three terms, each with a typical function. The first term is proportional to the error signal, the second term is proportional to its integral and the third term is proportional to its derivative.

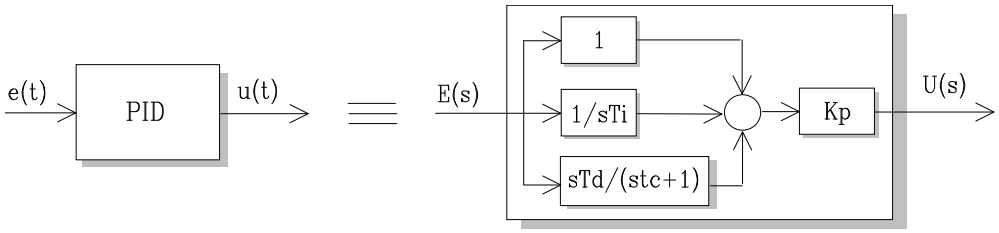


Figure 5.5: The PID-control law.

However, to be practical, this algorithm needs to be modified, especially for the implementation of the derivative action. Pure derivation is a practical nonsense, as it has unbounded amplification at high frequencies. This is physically impossible and even unwanted. Indeed, the slightest high frequency noise, superposed on the error signal, would be enormously amplified. Furthermore, the controller output at a step in the desired output would change annoyingly abrupt (in theory, an impulse is required).
'Smoothing' can be introduced by processing the error signal with a low-pass filter before taking the derivative. In practice, both operations are implemented in one filter. The transfer function of the PID-controller is then given by:

$$C(s) = K_p(1 + \frac{1}{sT_i} + \frac{sT_d}{st_c + 1})$$

where t_c must be small compared to T_d, to have a derivative action. When t_c is chosen too big, the third term becomes a proportional one, even for low frequencies. A possible PID-scheme is given in Figure 5.5.

The open loop time response of a PID controller to a step error input, is shown in Figure 5.6. The *P-action* is fast since it reacts immediately to a change in process value. A constant error results in a constant control action due to the P-term. The *I-action* causes the control action to keep on growing if the error keeps the same. Therefore the error will always decrease using a I-controller. Compared to the P-controller, the I-controller is rather slow. The *D-action* reacts very fast when an error appears. It is only active at the moment that the error changes.

Remark: In order to obtain a P-control action, a deviation must exist. Thus, if the process needs some control action to maintain a certain position, a P-controller will never obtain the desired value, since only a deviation can provide the action. It must be understood that this happens in most cases: although the desired value is obtained, one must keep on acting on the system. If we want a certain temperature at home (e.g. 20^oC), while the outside temperature is -5^oC, we need the radiators to give heat although the inside temperature is already 20^oC. Exactly, we need as much heat as heat that is lost to the outside world. So we need a non-zero constant control action to maintain our desired value.

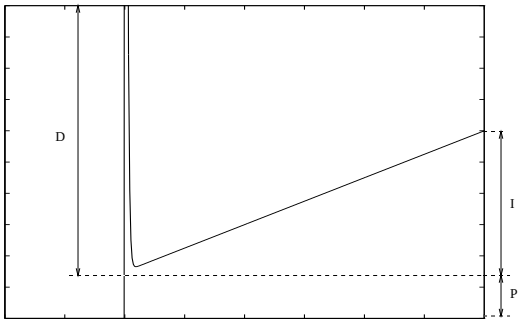


Figure 5.6: The PID-controller response to a step input in open loop. The P-action results in a constant when the error is constant. The I-action keeps on growing, while D-action is only active at the moment that the step is applied.

5.3.2 Closed Loop performance of a PID-controller

Of course the PID-controllers discussed above will always operate in closed loop, this is: connected to the system. Basically 3 parameters (excluding t_c) have to be chosen:

- *The gain* K_p
- *The integration time or reset time* T_i
- *The derivation time or lead time* T_d

We now analyse in more detail the effect of the different options.

P-action

If we assume a plant with a DC-gain equal to K and a P-controller with gain K_p, one can readily calculate that after the controller transients, the process output will not converge to a reference r. An equilibrium will be found between output and error (Figure 5.7).

At equilibrium one obtains (see also paragraph 3.3):

$$e = \frac{1}{1 + K_p K} r$$

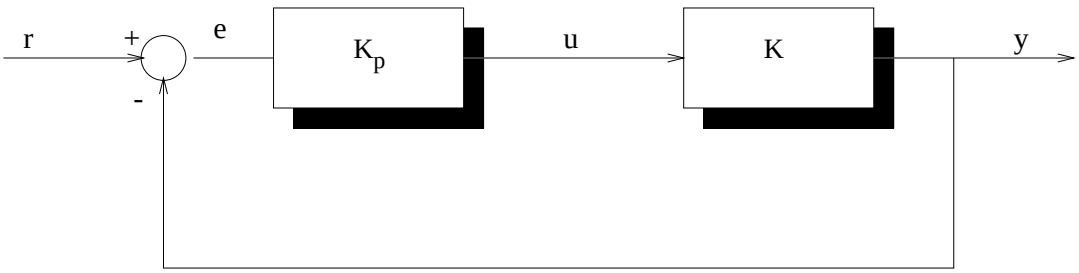


Figure 5.7: Static equilibrium between error and output with a proportional controller.

Only with an infinite DC-gain K or K_p, the error will be zero. So, a DC-offset between reference value and process output will always exist with a pure P-controller. The error can be made smaller by increasing K_p. Notice that, if the plant contains a pure integrator (a $1/s$-factor), no offset will occur since the DC-gain of the plant is ∞.

In order to understand better another effect of the P-controller, we introduce the concept 'proportional band'. Since a controller is physically limited in its output (saturation), the controller only reacts in a proportional way to its input in a certain region. That region is the proportional band. If K_p is very large, the controller almost immediately reaches the limits of the band when an error occurs. The proportional band is very small. Outside the proportional band, the output is constant and not function anymore of the error. This is shown in Figure 5.8.

$$proportional_band = \frac{u_{max} - u_{min}}{K_p}$$

We'll analyse the case when a step input is applied to a closed loop system with a small proportional band (K_p very large). The controller is almost always saturated. The controller will respond to a step reference input with a full control action. Therefore the error will decrease quickly. Only very close to the new reference value the control action will be proportional to the error. Since the input has been a full control action for almost all the time, a lot of energy has been pumped into the system. The system output will therefore not immediately slow down at the desired reference value, but will pass it right away. A too big K_p will thus result in overshoot and oscillations.

A trade-off has to be made between overshoot and DC-offset.

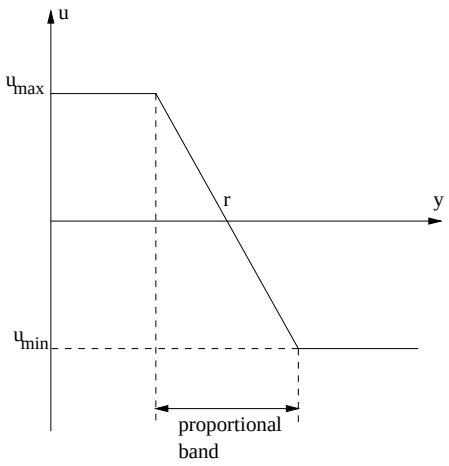


Figure 5.8: The proportional band of a controller.

In Figure 5.9 this effect is shown for a plant with transfer function $P(s)$. The controller is saturated on $(-3, +3)$. A step with unitary magnitude is applied to the closed loop function.

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

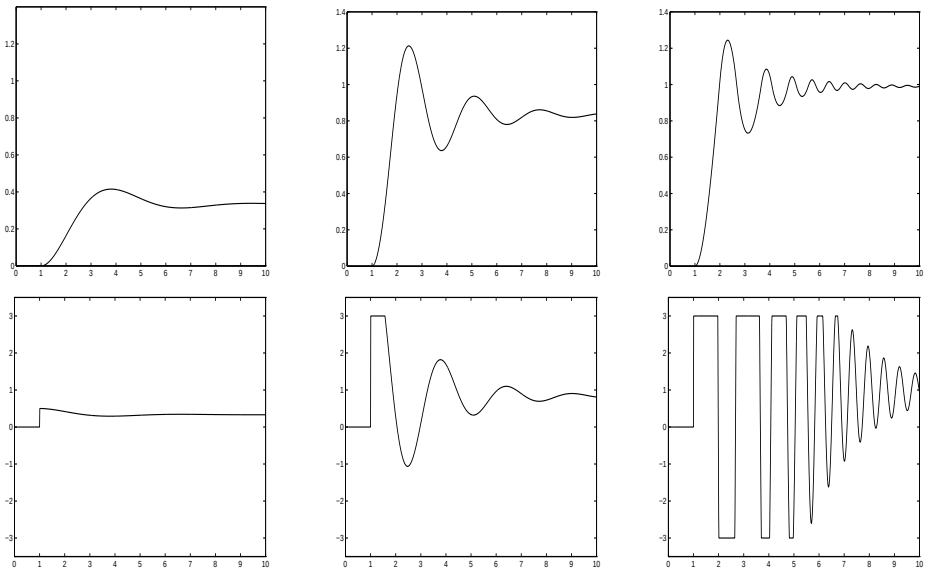


Figure 5.9: Effect of gain K_p on a response to a unit step in reference value. Above the process output is shown, below the control action. a. $K_p = 0.5$. Notice the large steady state error. b. $K_p = 5$. The steady state error is reduced, and there is some oscillation. c. $K_p = 100$. Nearly no steady state error, but too many oscillations.

I-action

As already mentioned above, the P-action alone results in a steady state error. This can be avoided by introducing an infinite DC-gain, realized by an integrator. The integrator action is thus added in order to abolish DC offset. This can be understood since an integrator can provide a non-zero output although the input error is zero. This is in contrary to the P-action which cannot provide an action if the error is zero.

Since the I-action is not changing at all when the error is zero (an integral remains constant when the input is zero), there is a real risk for overshoot. When the output value crosses the reference value, the I-action is maintained (integral!). Thus the I-action will continue in the same direction, which results in overshoot. The I-action will decrease however due to the negative error, until it is zero. From that moment on the process-value will start decreasing too (90^o phase delay between control and error signal). The resulting oscillation has a relatively large period. See also Figure 5.10.

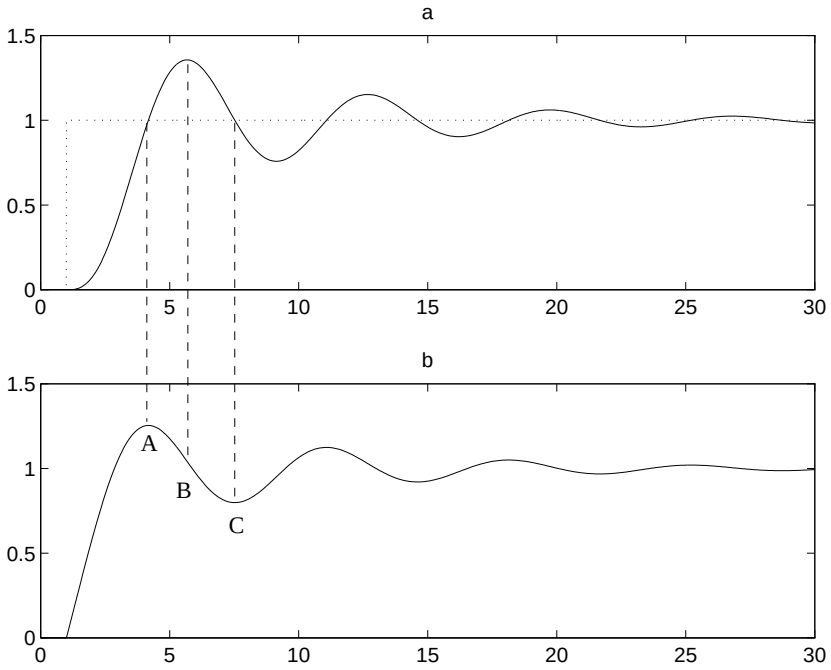


Figure 5.10: Effect of an integrator control action. a. The reference value and the process output. b. The control action. In point A the error becomes negative, nevertheless the I-action is its maximum value! In point B the error is maximally negative, the integral is changing quickly. In C the error is zero again and will become positive, thus the integral will grow again... Remark the 90^o phase difference.

Decreasing the integration time T_i slows down the response even more, resulting in a more oscillatory behaviour. It can be found that the integration time equals the time needed to obtain (in open loop, applying a step input) the same control action as a unit proportional control action. A short integration time results therefore in a strong I-action.

PI-action

Compared with the P-controller, the PI-controller eliminates the steady-state error. Compared with the I-controller, the PI-controller is faster with a smaller oscillation period.

PD-action

In contrary to the I-action, the D-action is most active when the output crosses the reference value. Since it is proportional to the derivative of the error, it can 'predict' how the error will change in the near future, and will act in that way. Therefore it has a damping effect and one can choose a larger K_p value, which will result in less offset. The meaning of derivation time (better: lead time) T_d is the following: if we apply a unit ramp in open loop to a PD controller, the P-controller will provide the same value as the D-controller, only after T_d seconds. The D-action is therefore T_d seconds faster, or 'leads' the P-action by T_d. The larger T_d, the stronger is the D-action compared to the P-one. This will be discussed later. See also Figure 5.14.

PID-action

The I-action is added to eliminate steady-state error, the P-action to make the response faster, and the D-action is added to provide damping.

5.3.3 Technical Terms

In reality, the terms P, I and D are not used. Gain, percentual proportional band, reset rate, reset time, rate time, pre-act, derivative action .... are used instead.

With respect to the P-action the quantity *gain* or *percentual proportional band* (PB) is used.

- $Gain = K_p$.
- The *percentual proportional band* is related to the proportional band. Remember that the proportional band was that interval in process values that made the P-action (output of the P-controller) go from one point of saturation to the other one (y to u). The measurement range is defined as the range of process values that can be obtained using the full control span (u to y). Notice that this is the other way round! Notice also that the proportional band need not to be equal to the measurement range. See Figure 5.11 and Figure 5.12.

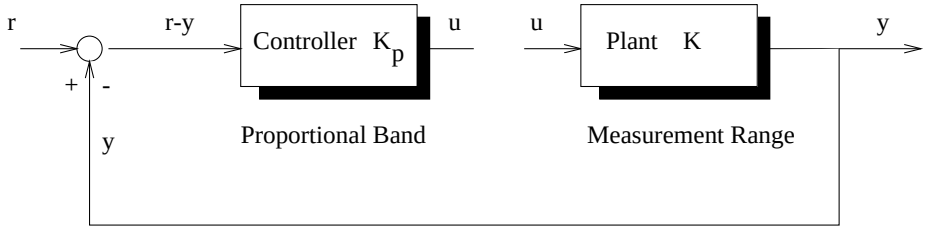


Figure 5.11: The proportional band and the measurement range.

The percentual proportional band (PB) is defined as the ratio between the proportional band and the measurement range, expressed in %. This way the sensitivity

of the process is coupled to the sensitivity of the controller. A PB of 1% means that, when the process output changes 1% of the total controllable region, the control output will change the whole range (which on its turn can move the process output 100%). PB= 1% is therefore a very strong control action. On contrary, PB= 120% is a very soft action. With process gain K, one can find that

$$PB(\%) = \frac{100}{K_p K}\%$$

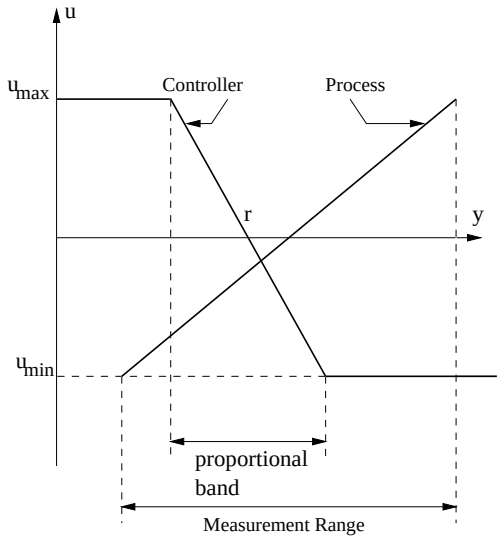


Figure 5.12: Static characteristics of process and controller.

With respect to the I-action the quantities *reset rate* and *reset time* are used.

- *Reset time* T_n compares the I-action with the P-action. It is defined as the time that the I-action needs to grow as big as the P-action, when a step is applied in open loop.
- *Reset rate* r is defined as the reciprocal value of the reset time: how many times per second/minute does the I-action repeat the P-action. One can find:

$$T_n = \frac{1}{r} = K_p T_i$$

In Figure 5.13 one observes the open loop response of a PI-controller to a unitary step input at the reference value. The P-action results in an immediate constant control action. Moreover, the control action keeps on growing due to the I-action. One notices that after one minute the control action has tripled the initial P-action. The reset rate is thus 3.

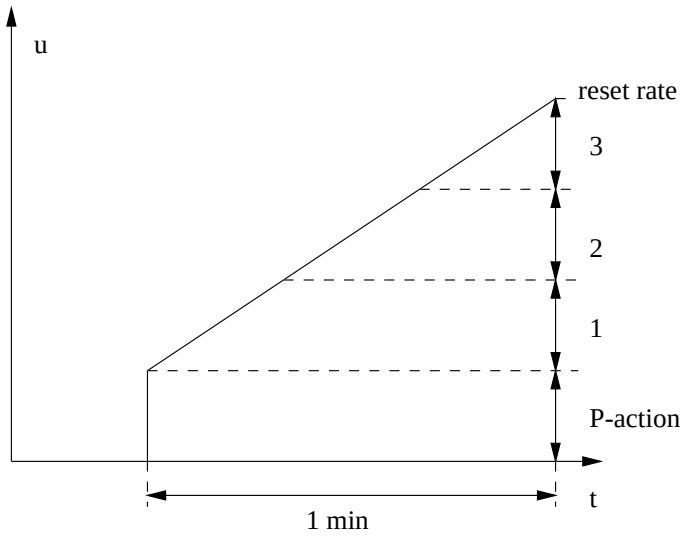


Figure 5.13: Reset Rate. The I-action repeats the P-action 3 times per minute, resulting in a reset rate of 3.

With respect to the D-action one uses *lead time, rate time and pre-act (T_v).* They all have the same meaning, defined as follows: if we apply a unit ramp in open loop to a PD controller, the P-controller will provide the same value as the D-controller, only after T_v seconds. It compares the proportional action with the derivative one. One can find:

$$T_v = T_d$$

See also Figure 5.14.

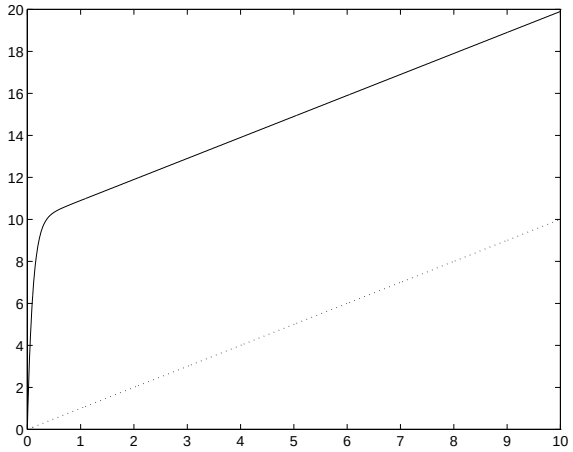


Figure 5.14: Lead Time. Dashed is the ramp input, solid is the PD controller open loop output. Notice the transient response due to the non-ideal derivative action. Lead Time is 10, since the ramp due to the P-controller needs 10 seconds to increase as much as the immediate offset due to the D-controller.

5.3.4 PID-Control in Processes with dead-time

The dead-time control problem is a very classic one. Everybody already tried to control the water temperature when taking a shower. The problem consists in the fact that when the control action is executed (when you open the tap slightly more), the process effect takes a time to feel the action. The water still has to move on in the tubes before the new temperature gets out of the shower. This problem is frequently encountered in industry, where transport delays are very common. It is not easy to tune these processes: the extra time delay results in an additional phase shift, which decreases the relative stability. More overshoot and a more oscillatory behaviour will be observed. Before implementing a controller, one must always try to optimise the design in order to minimise the delays. If a PID-controller is chosen to control the process, some rules of thumb exist that can help you to tune.

It will be supposed that the process to be controlled can be approximated by a second-order process with dead time. In a lot of cases this assumption is valid. First a pure dead time and an integrator with dead time will be analysed. The other cases will afterwards be reduced to that same form.

In the case of a pure time delay with a proportional control (see Figure 5.15), one can easily verify by hand that closed loop stability is very sensitive to the gain factor (Figure 5.16) . With small gains, the feedback system is stable but the steady state error becomes unreasonably large though.

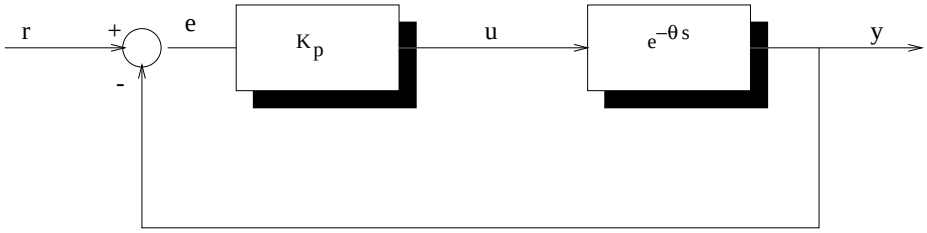


Figure 5.15: Proportional control of process with pure dead-time.

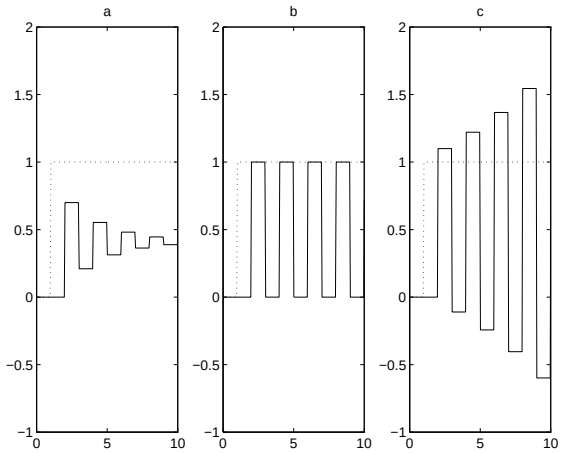


Figure 5.16: Response of pure dead-time process with varying K_p: a. $K_p = 0.7$, b. $K_p = 1$, c. $K_p = 1.1$. The output is solid, the reference input dashed.

In the case of a dead time together with an integrator , one sees that the process value becomes smoother, and no offset error is presented. See Figure 5.17 and 5.18.

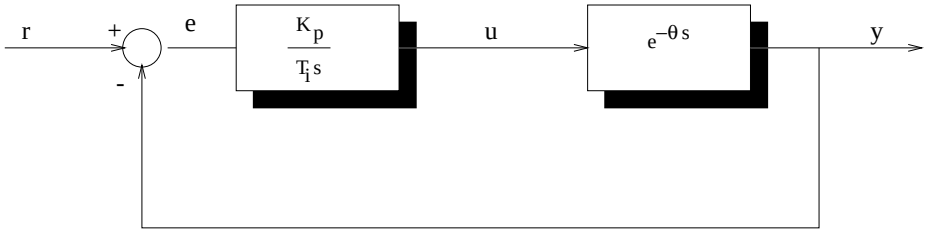


Figure 5.17: Integral control of process with pure dead-time.

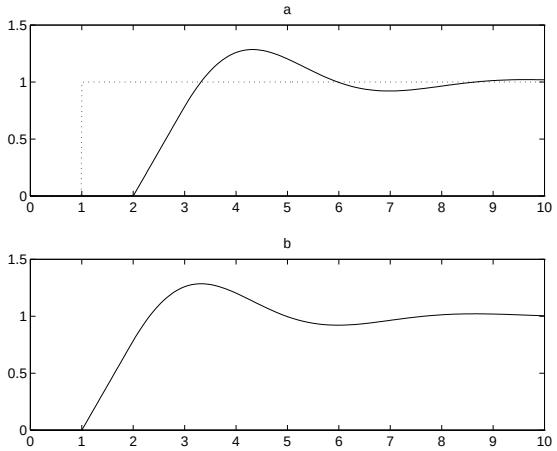


Figure 5.18: Response of pure dead-time process to integral control. a. Process output(solid) and reference value(dashed) b. Control value. Notice the 25% overshoot characteristic.

In order to obtain 25% overshoot, it can be proven that one should choose the gain K_p such that it satisfies

$$K_{p,25\%} = \frac{\pi}{4}\frac{T_i}{\theta}$$

where T_i is the integration time and θ the dead-time.
The instability region can be proven to start at K_p, being the double of the value obtained for 25% overshoot:

$$K_{p,instable} = \frac{\pi}{2}\frac{T_i}{\theta}$$

Both a first order and a second order process will be reduced to the previous case, resulting in the 25% overshoot characteristic.

A first order plant with transfer function P(s) can be transformed into the previous form with a PI-controller. If the plant is given by

$$P(s) = \frac{Ke^{-s\theta}}{1 + sT_1}$$

and the PI-controller by

$$C(s) = K_p(1 + \frac{1}{sT_i}) = \frac{K_p}{sT_i}(1 + sT_i)$$

If one chooses T_i equal to T_1, the loop transfer function reduces to a simple integrator. The gain K_p can then be chosen in order to meet the 25% goal. Notice that a first order process without dead-time never could be made unstable with a PI-controller. The dead-time introduces instability. Note however that such a pole-zero cancellation is far from optimal in practical designs, especially when system parameters are not exactly known or varying.

Also a second order system can be modified by a correctly tuned PID-controller. The plant transfer function is given by a cascade of two first order processes, with T_1 the larger time constant.

$$P(s) = \frac{Ke^{-s\theta}}{(1 + sT_1)(1 + sT_2)} = \frac{Ke^{-s\theta}}{(1 + s(T_1 + T_2) + s^2T_1T_2)}$$

The PID controller is given by the known transfer function

$$C(s) = K_p(1 + \frac{1}{sT_i} + sT_d) = \frac{K_p}{sT_i}(1 + sT_i + s^2T_iT_d)$$

If we equal

$$T_i = T_1 + T_2$$

$$T_d = \frac{T_1T_2}{T_i}$$

we obtain again the integrator with time delay. K_p must be adapted such that the 25% specification is obtained.

In general these conclusions can be simplified.

- If T_2 is much smaller than T_1, the process can be approximated by a first order system, thus PI-control is sufficient.
- If the dead-time becomes bigger than the two time constants T_1 and T_2, the process can be assumed as a pure time delay, resulting in a I-controller.
- If the largest time constant T_1 is much bigger than the dead-time, the process can be assumed without dead-time, so a P-action might do the job.

These conclusions are summarised in Figure 5.19.

Notice that these PID-tuners tune for 25% overshoot. If one is interested in other design goals, other tuning methods must be used.

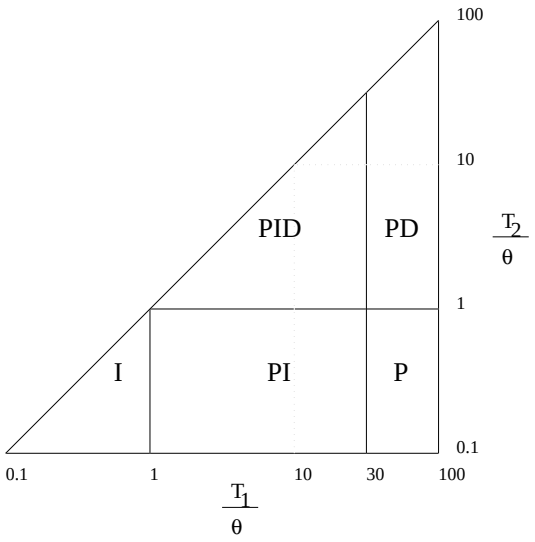


Figure 5.19: Choice of PID-controller in different dead-time processes.

5.3.5 Anti Wind-Up

The actuator is the device through which the control input is applied to the process. Since these components have a limited range (e.g. motor driving amplifiers have a minimal and maximal voltage or current output), they introduce a saturating nonlinearity into the process (Figure 5.20).
As an example, consider the PI-control of a certain process. Suppose a set-point change brings the actuator into saturation. In that case, the actuator action is smaller than the output of the PID controller desires. Consequently, the process output is varying less quickly than desired and the error signal will decrease slower than without saturation. Therefore the integrating output term will be higher and grows constantly. At the moment that the process output equals the reference set-point, the integrator output is very big. Hence at that instant, the actuator keeps saturated and, although the set-point has been reached, the actuator keeps going full speed! From that moment the error changes sign and the integrator decreases. But only from the moment that the integrator output goes beneath the actuator saturation, the actuator will start giving less than the maximum value, and the process output will start going in the other direction at last.

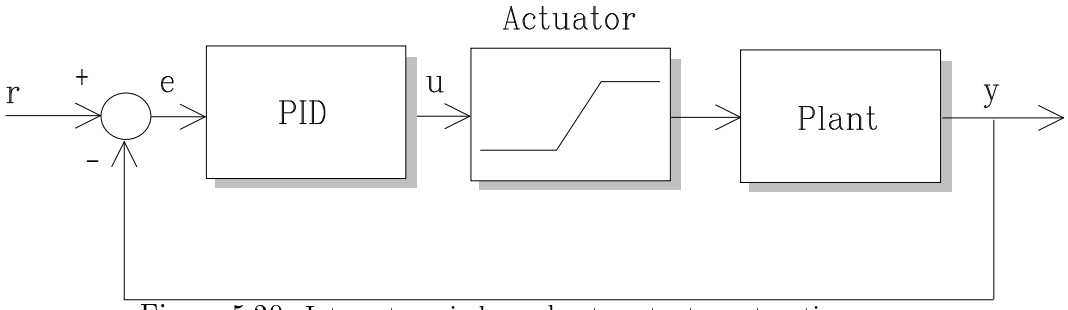


Figure 5.20: Integrator wind up, due to actuator saturation.

Summarised: the actuator will be saturated during a longer time. Further, during saturation, the feedback is actually not working: it can be replaced by a fixed-level input, which gives rise to undesirable overshoots for processes with a low stability degree. The previous effect is called *integrator wind up.*

This phenomenon can be avoided by stopping the integration when the actuator saturates. A clever way to do so is to force the output of the controller u to stay at the saturation limit u_l as long as the actuator is saturated. This can be done by implementing an extra feedback loop which forces the difference between actuator input and actuator output to be zero at all times. The integral part of the controller can be used for this purpose (Figure 5.21). As long as there is no saturation, $e_w = 0$ and the PID-controller works in the normal way. As soon as $e_w \neq 0$, the extra loop tries to bring e_w back to zero, to a rate which is inversely proportional to T_a. This may drastically reduce the saturation time and improve the step response! Be careful to take T_a not too small, if there is an active D-action ($T_d \neq 0$), as spurious peaks originating from it may drive the actuator into saturation for a short time: too quick resetting during this short peak results in an unnecessary small value for the integrator output after the peak has vanished. As a rule of thumb it is suggested to take $T_a = \sqrt{T_i T_d}$.

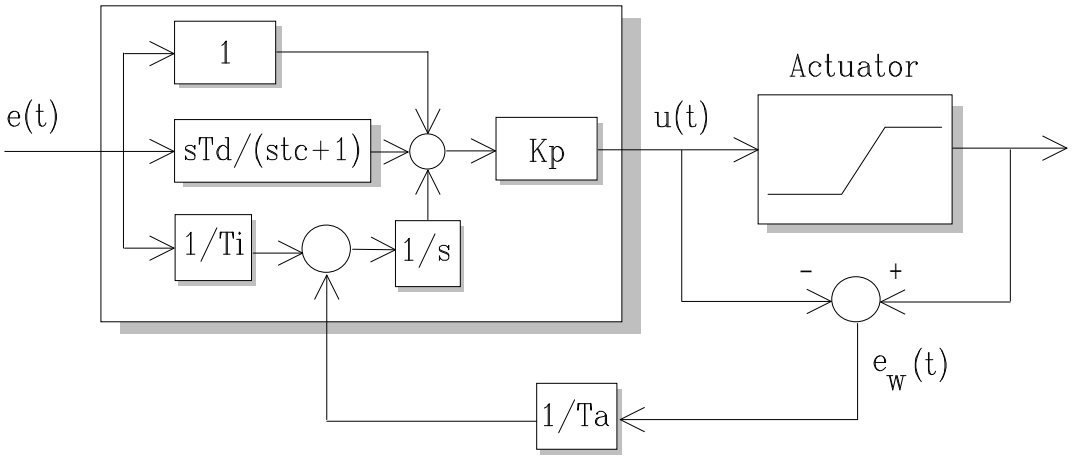


Figure 5.21: Anti-windup by introducing an extra feedback loop.

Example: the hydraulic positioning system
In Figure 5.22 the effect of anti-windup is shown for the hydraulic positioning example. *The PID-parameters are those obtained with Ziegler-Nichols, but anti-windup is added. The position settles much faster, and there's nearly no overshoot. Undershoot occurs though. This is due to the very big derivative action in the beginning that saturates the actuator. The difference between actuator input and output is thus very large, and this is fed back to the integrator. This results in an opposite control action! The anti-windup must therefore be decreased ($T_a$larger). If the step rise time would be lower, this problem could also be avoided. Take care: for a negative system (negative DC-gain) like the positioning system, the negative e_w should be fed back, if not an unstable system results.*

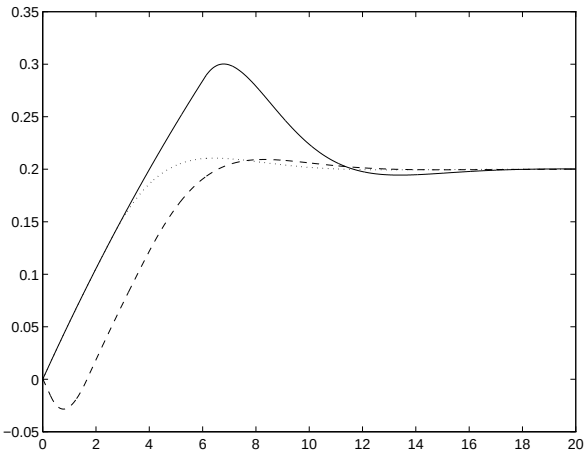


Figure 5.22: Effect of anti-wind-up on hydraulic positioning system.Solid: the response without anti-windup ($T_a = \inf$). Dashed: $T_a = 1$. Dotted $T_a = 1000$.

Chapter 6

Control Configurations

6.1 Master-Slave Control (Cascade Control)

In cases where we want to control a slow process which is disturbed by a quickly varying input variable, often cascade-control is applied.
The configuration is given in Figure 6.1.

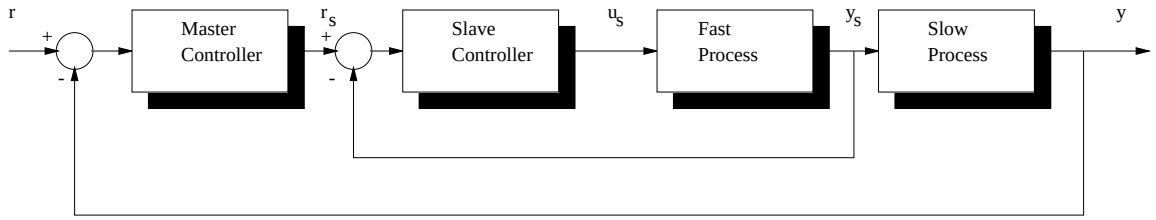


Figure 6.1: Master-slave configuration. The outer loop represents the master, the inner loop the slave.

Two control loops are distinguished: an inner loop and an outer loop. The inner loop tries to control the fast modes of the system, while the outer loop controls the parameter we are really interested in. The inner loop is also called the secondary loop or the slave loop, while the outer one is called the primary or master loop.

Example 2 *We try to keep the temperature in a vessel constant. The vessel contains an exothermic process, so the temperature is controlled by the right amount of cooling fluid. The first idea might be to use a single control loop, in which the temperature is controlled by the position of the cooling fluid input valve. This is shown in Figure 6.2.*

However, it results that due to pressure changes the input flow is not constant. So one tries first to control the valve position such that a desired flow is guaranteed (slave). The valve position is controlled by the slow temperature process (master). This is shown in Figure 6.3.
In order to tune master-slave loops, one has to take into account that:

- *One should first tune the inner loop, while the outer loop is operating. Then the outer loop is tuned.*
- *Problems can arise if both loops are equally fast. As a rule of thumb the inner loop must be at least 5 times faster than the outer one.*

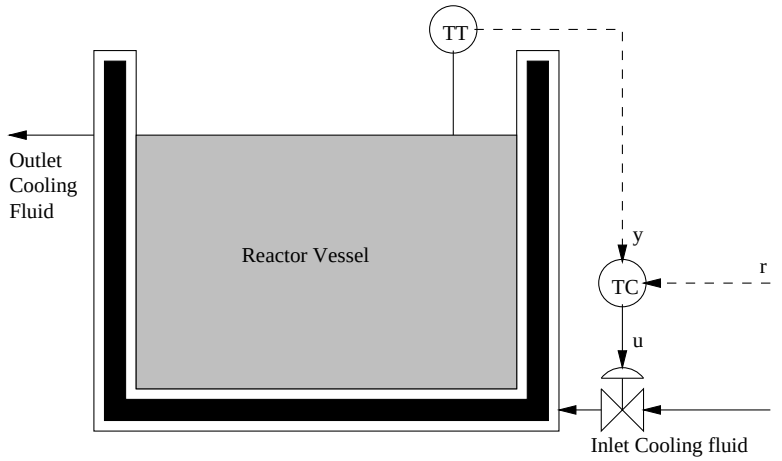


Figure 6.2: Control of vessel temperature.

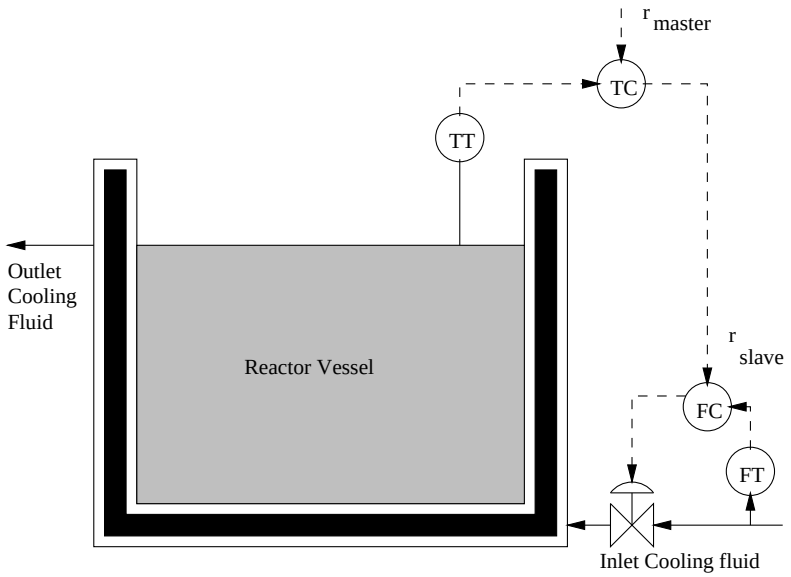


Figure 6.3: Control of vessel temperature with flow slave-control.

- *For the inner loop in most cases a D-controller will be sufficient, sometimes I-action is added to filter high-frequent noise components. For the outer loop PI-action often serves.*

△

6.2 Feed-forward Control

In feedback control, the process output must vary from the desired value in order to excite some control action. If the process is well known, another scheme of control can be used. Ideally one can calculate (if the process is completely known) the control input needed to obtain the desired output. This type of control is markedly faster, since there is no longer waited for an error to occur in order to perform a control action, but the action is undertaken

on beforehand. A distinction is made between *reference feed-forward control* and *disturbance rejection feed-forward control.*

1. In **reference tracking feed-forward control**, the output should follow a well defined path. The inverse of the plant gives us the input necessary to obtain that path, since:

$$Y(s) = P^{-1}(s)P(s)$$

This is shown in Figure 6.4.

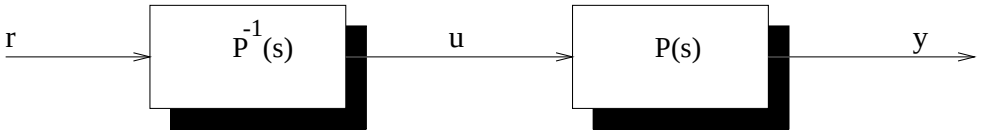


Figure 6.4: Feed-forward Control Configuration for reference tracking purposes.

One should take care when taking the inverse of the plant. Since a plant will always be causal (reflected by a higher degree in the denominator), the inverse plant will be anti-causal. Therefore, some extra poles should be introduced that limit the high frequency behaviour. This results in a deteriorated high frequency response – but that's not too dramatic since in most cases we're mainly interested in low frequency behaviour.

2. In **disturbance rejection feed-forward control**, one tries to compensate for disturbances to come. The disturbances are modelled by a transfer function $H(s)$, and are additive to the process output. If the disturbances can be measured, a feed-forward scheme shown in Figure 6.5 is able to reject them.

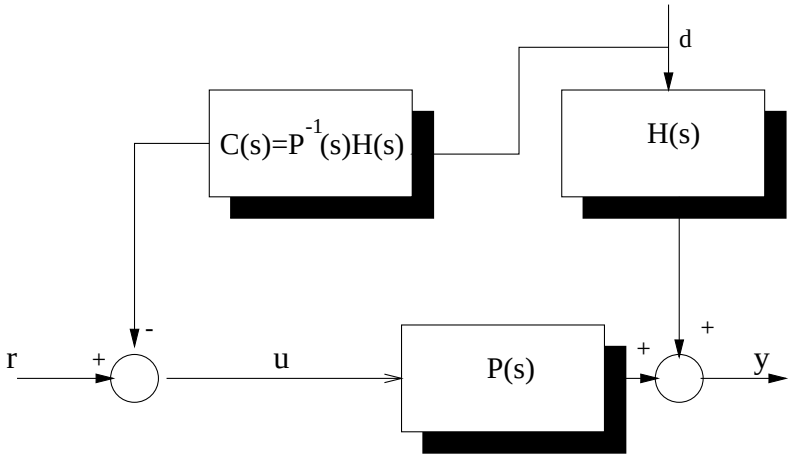


Figure 6.5: Feed-forward control for disturbance rejection purposes.

Without feed-forward control the process output would be

$$Y(s) = P(s)U(s) + H(s)D(s)$$

With disturbance rejection the output is

$$Y(s) = P(s)\Big(R(s) - P^{-1}(s)H(s)D(s)\Big) + H(s)D(s) = P(s)R(s)$$

The main disadvantage of feed-forward control is its disability to avoid steady-state errors. Ideally, this never occurs. But with the slightest error in the plant/disturbance model, the output won't exactly be what was planned. Since the error is not fed back, the controller doesn't 'know' there is an offset.

Therefore a combination of feed-forward with feedback to eliminate steady-state errors is mostly used. This is shown in Figure 6.6. Here feed-forward is used for tracking capabilities, and feedback for steady-state accuracy. In normal working operation ideally only the feed-forward would be active (inverse of the plant). If an error results however, the feedback loop will cancel it.

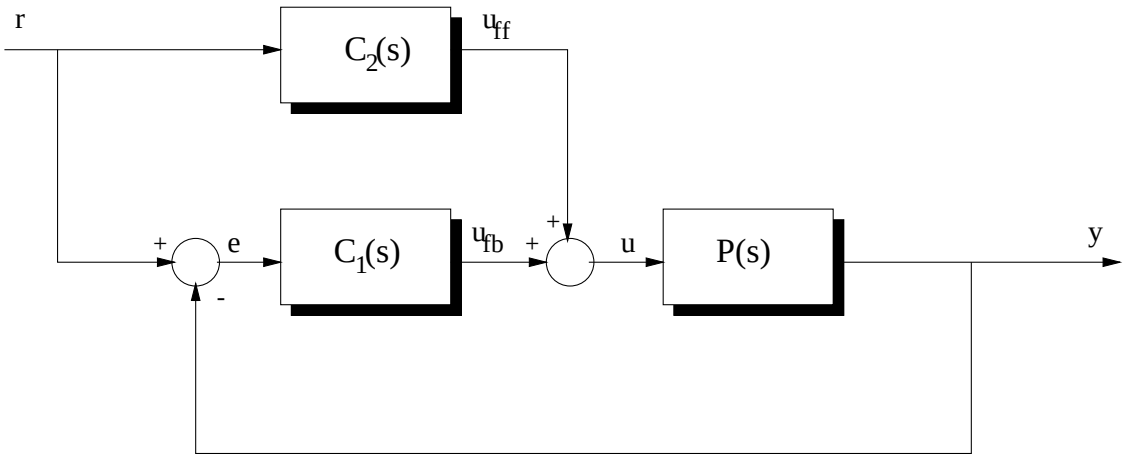


Figure 6.6: Controller with feedback and feed-forward loop.

6.3 Split-Range Control

In the cases mentioned up to now, the control action only works in one direction. If we control a liquid level by an output valve, the level can only be diminished by the control action. The level must be raised by normal working procedures, for example an ever incoming flow. If we put a heater at home, the temperature can only be raised by the heater.

We count on the heat transfer to the outside world for cooling purposes. If we want to control the temperature though for lower temperatures too (in summer e.g.), we need two devices and two controllers. The first one governs the heating process, the other one the cooling process. This type of control is called split-range control. An example is given in Figure 6.7. Notice the two slave controllers for valve positioning purposes.

The control configuration is designed such that, without control action, both of the actuators are off. For positive values the heating is proportional to the controller output and the cooler is switched off. For negative values the reverse is true.

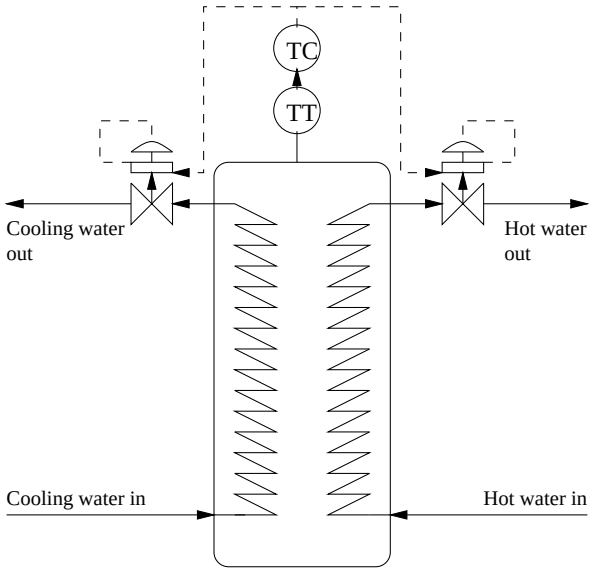


Figure 6.7: Example of split range configuration in vessel temperature control.

6.4 Ratio Control

In some applications the absolute value of incoming flows is not important, but the ratio of the two flows is the variable that matters. Examples are temperature control with a cold and a hot water inlet or concentration control problems in chemical reactors. This is generally done with a 'Ratio Station' which is in fact nothing more than a constant that, multiplied with the first variable, results in the set-point for the second variable (Feed-forward Control). The set-point that controls the ratio of the two variables is then controlled by a feed-back controller.

An example is shown in Figure 6.8.

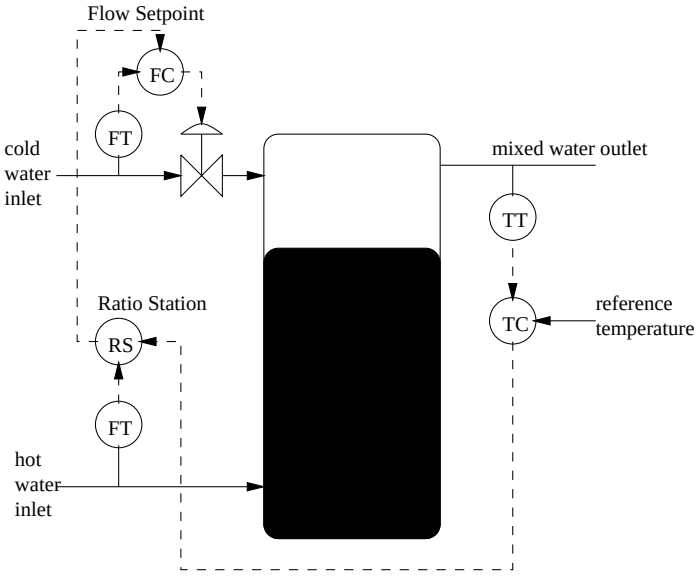


Figure 6.8: Control Strategy for controlling water temperature by Ratio Control.

Chapter 7

Tuning Methods

7.1 Introduction

Once a control configuration and a control law have been chosen, the parameters must be tuned in order to reach the specifications. The tuning methods will be applied to PID-controllers, though some of them are more general and can be used with other controllers too.

7.2 Heuristic Tuning Methods

7.2.1 Trial-and-Error tuning method

Basically one considers:

- P-action is introduced for increasing the speed of the response. With exaggerated P-action, oscillation will result.
- I-action is introduced for obtaining a desired steady-state response. Disadvantage is a more oscillating response with a longer period.
- D-action is introduced for damping purposes. Disadvantage is the fact that oscillation on a high frequency is more probable.

A trial-and-error method will now be:

1. **Proportional Action**
   Put $T_i = \infty$ and $T_d = 0$. Take a small gain K_p or a large PB (nearly 100%). Apply a small step. Damping will be very small due to the limited gain. Double the gain, damping will decrease and oscillation will increase. Keep on doubling till the oscillation exceeds your requirements.
2. **Derivative Action**
   Once the P-action is tuned, add D-action to damp the system. Increase pre-act T_d till the system is too damped. Once a good point has been found, P-action can be increased. On its turn D-action must improve the necessary damping for the new P-action. If addition of D-action makes the system response nervous and oscillatory on higher frequencies, drop D-action.

3. **Integral Action**
   First tune P and D. Decrease reset time T_i a bit. The influence of the I-action will still be limited; the output tends slowly to the set-point though. Halve reset time in order to obtain a faster set-point settling. Keep on halving, till the response gets oscillatory with a low frequency. It is possible to decrease P-action to remove the oscillatory reflex, increase pre-act, and optimise the I-action again.

Advantage of this method is the intuitive approach. It is very elaborate though and doesn't guarantee success. It turns out to be a very time consuming method when the main time constants of the plants are in order of some minutes.

7.2.2 Ziegler-Nichols Tuning Rules

All the variables of the PID must be tuned such that the overall closed-loop system response behaves like is stated in the specifications. In industry, a lot of times the famous 'Ziegler-Nichols' rules are applied, because it is not necessary to have a complete, dynamic model of the plant. Only a few, fairly simple experiments are required to find suitable controller parameters. Since these rules are empirical rules, it should be recognised that problems can occur. However, in many cases, these rules give rise to a control system with very acceptable performance trade-offs.

Transient response method (Ziegler-Nichols) To apply this method, it is required that the plant's step response has a similar shape as the one given in figure 7.1. From this step response, the time delay or lag L and the steepest slope or reaction rate $R = \frac{K}{\tau}$ can easily be derived (after a delay of L seconds, the output increases K units in τ seconds).

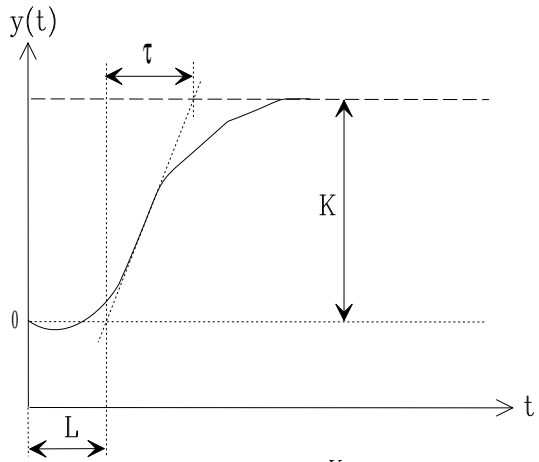


Figure 7.1: The time delay L and the reaction rate $R = \frac{K}{\tau}$ are easily calculated from a step response.

Once these parameters are computed, the controller parameters are selected from table 7.1.

Remark: If the plant's step response differs drastically from that in figure 7.1, this method is not applicable anymore, since L and R can't be measured/computed anymore.

Ziegler & Nichols	K_p	T_i	T_d
P-control	$\frac{1}{RL}$		
PI-control	$\frac{0.9}{RL}$	$3.3L$	
PID-control	$\frac{1.2}{RL}$	$2L$	$\frac{L}{2}$

Table 7.1: Ziegler-Nichols rules for standard step response

Example: the hydraulic positioning system
We apply this method now to the hydraulic positioning system discussed before. An open loop step response (in the operating point) to a 1% *valve displacement, is shown in figure 7.2. We notice a* $2cm$ *negative displacement in steady state. We find* $K = -0.02$ *and* $\tau = 22s$*, thus* $R = -9.1 \times 10^{-4}$*. Dead time* L *is zero, but this causes problems in the Ziegler-Nichols formulas. In reality, there's always a dead time, though it may be very small. We assume now a dead time* $L = 1s$*.*

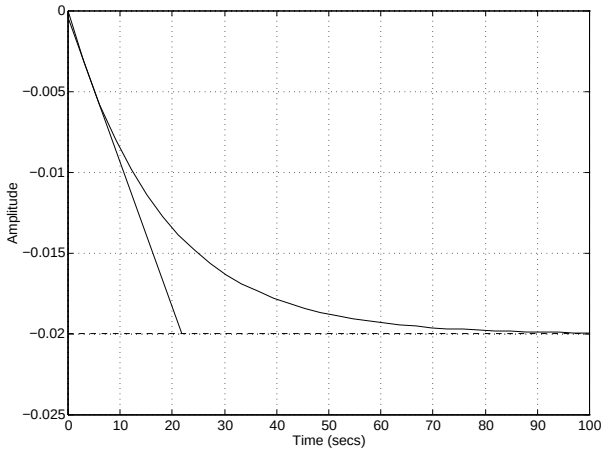


Figure 7.2: A step response to a 1% valve change.

The PID-control configuration is shown in figure 7.3. Notice the 'saturation' block, which determines the proportional band or the linear region of the controller. The controller output is limited between 0% *and* 100%*. Since the operation point is* 50%*, the saturation goes from* -50% *till* 50%*. All the inputs and outputs are given as deviation variables too. Also notice the introduction of the disturbance.*

According to the Ziegler-Nichols rules, K_p*,* T_i *and* T_d *are then calculated for a P-controller, a PI-controller and a PID-controller. As a test, the reference position is changed to* $20cm$*, and as a disturbance, a decrease of* 10 *litres/minute of pump flow is introduced after* 30 *seconds.*

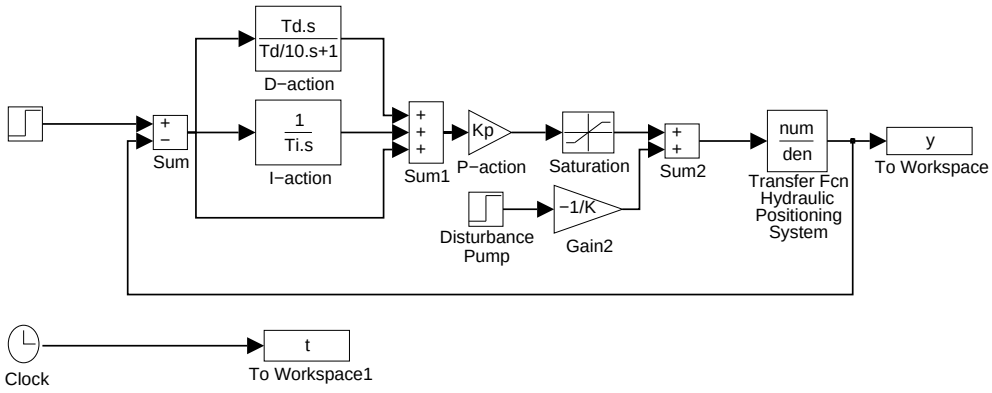


Figure 7.3: The PID control configuration for the linearised hydraulic positioning system.

1. *The results for a P-controller are shown in figure 7.4. Notice the steady state deviation and the saturation in control value u.*
2. *For a PI-controller, the Ziegler-Nichols results are shown in figure 7.5. Observe the zero-steady state error, the overshoot due to the integrator, and the excellent disturbance rejection.*
3. *For a PID-controller, the responses are shown in figure 7.6.*

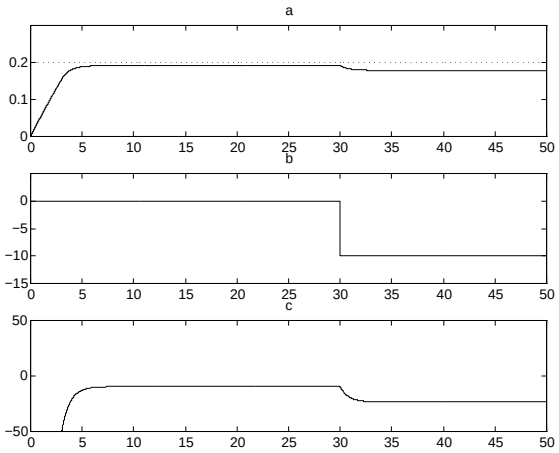


Figure 7.4: The P-controller according to Ziegler-Nichols for the linearised hydraulic positioning system. a. Reference signal (m) (dotted) and Process output (m) (solid). b. Pump flow disturbance (litres/minute) c. Control effort u (%)

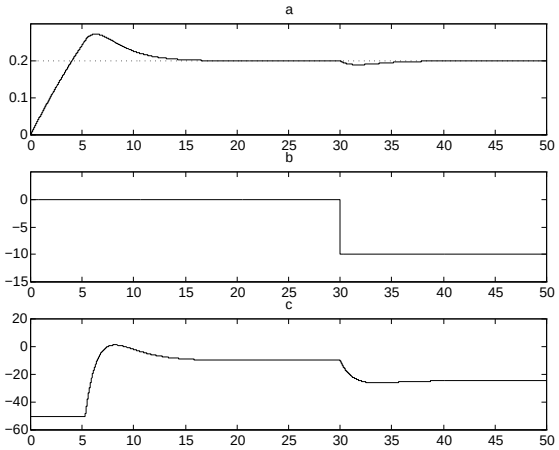


Figure 7.5: The PI-controller according to Ziegler-Nichols for the linearised hydraulic positioning system. a. Reference signal (m) (dotted) and Process output (m) (solid). b. Pump flow disturbance (litres/minute) c. Control effort u (%)

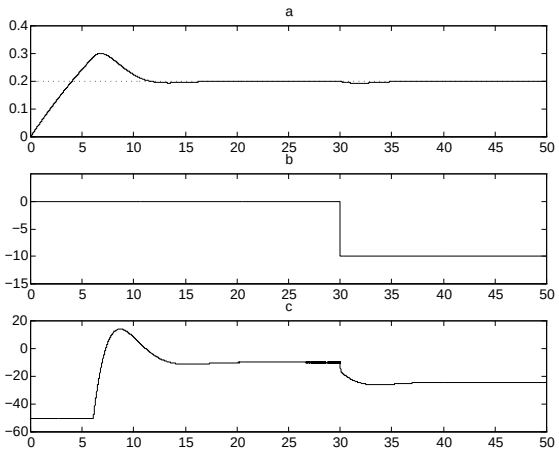


Figure 7.6: The PID-controller according to Ziegler-Nichols for the linearised hydraulic positioning system. a. Reference signal (m) (dotted) and Process output (m) (solid). b. Pump flow disturbance (litres/minute) c. Control effort u (%)

Stability limit method (Ziegler-Nichols) First, the plant is controlled using proportional control only ($T_i = \infty, T_d = 0$). Once a stable design is found (start with a very small value for the gain K_p), K_p must be increased slowly until a permanent oscillation occurs. At this point, the gain K_p and the oscillation period P_o are recorded (figure 7.7).

If they are denoted by K_o and P_o, the controller parameters are determined from table 7.2.

The hydraulic positioning example doesn't work with this method, since it turns out that it cannot be made unstable.

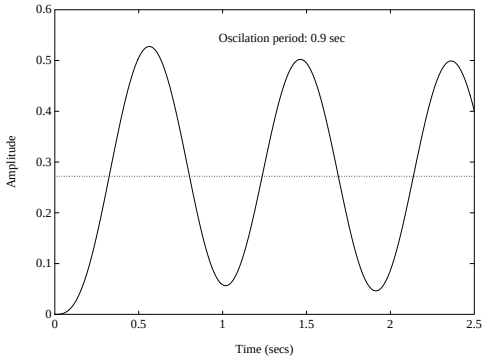


Figure 7.7: The measured oscillation period P_o for the stability limit method. The ultimate controller gain K_p, responsible for this permanent oscillation, is denoted as K_o.

Ziegler & Nichols	K_p	T_i	T_d
P-control	$0.5K_o$		
PI-control	$0.45K_o$	$\frac{P_o}{1.2}$	
PID-control	$0.6K_o$	$\frac{P_o}{2}$	$\frac{P_o}{8}$

Table 7.2: Ziegler-Nichols rules for stability limit method

Summary **Design rule:**

1. *Do some simple experiments and determine the characteristics that are required for the method you want to apply.*
2. *Use the appropriate table to calculate the controller parameters.*

7.3 Stability Limits (Routh-Hurwitz)

The Routh-Hurwitz technique is an analytical method to analyse stability of a plant. It determines whether the *characteristic polynomial* contains zeroes with positive real parts. The characteristic polynomial is defined as the denominator of the closed loop transfer function. Given a characteristic polynomial

$$a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0$$

an array (table 7.3) is constructed.

1	a_n	a_{n-2}	a_{n-4}	$\cdots$
2	a_{n-1}	a_{n-3}	a_{n-5}	$\cdots$
3	b_1	b_2	b_3	$\cdots$
4	c_1	c_2	$\cdots$	$\cdots$
$\vdots$	$\vdots$			
$n-1$	z_1			

Table 7.3: The Routh-Hurwitz Table

The first row contains the highest power of s, the third highest power of s, etc. The second row contains the second highest power of s, the fourth highest power of s, etc. Roughly a triangular structure is obtained after calculation, with only a single element in the last row.

The elements b_i, c_i, ... are calculated from the formulas

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}$$

$$b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}$$

$$\vdots$$

$$c_1 = \frac{b_1 a_{n-3} - a_{n-1}b_2}{b_1}$$

$$c_2 = \frac{b_1 a_{n-5} - a_{n-1}b_3}{b_1}$$

$$\vdots$$

Routh's Stability Criterion now states that a sufficient condition for all roots of the characteristic equation to have negative real parts, is that the elements of the first column in the array have the same sign.

By using Routh-Hurwitz it is now possible to obtain stringent conditions on some parameters, in order to have a stable closed loop system.

Example: the hydraulic positioning system

We recall that the transfer function was given by

$$P(s) = \frac{-1.15 \cdot 10^{-7}}{2.25 \cdot 10^{-10} s^3 + 1.543 \cdot 10^{-7} s^2 + 1.0317 \cdot 10^{-4} s + 5.7482 \cdot 10^{-6}}$$

we note this as:

$$P(s) = \frac{b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}$$

We want to analyse stability for the system controlled by a PI-controller:

$$C(s) = K_p \left(1 + \frac{1}{sT_i}\right)$$

The closed loop transfer function is given by

$$G(s) = \frac{C(s)P(s)}{1 + C(s)P(s)}$$

which results in

$$G(s) = \frac{b_0 K_p T_i s + b_0 K_p}{T_i a_3 s^4 + T_i a_2 s^3 + T_i a_1 s^2 + (T_i a_0 + b_0 K_p T_i) s + b_0 K_p}$$

$$G(s) = \frac{b_0 K_p s + b_0 K_p / T_i}{a_3 s^4 + a_2 s^3 + a_1 s^2 + (a_0 + b_0 K_p) s + b_0 K_p / T_i}$$

The Routh-Hurwitz table is thus given by table 7.4.

1	$2.25 \cdot 10^{-10}$	$1.0317 \cdot 10^{-4}$	$-1.15 \cdot 10^{-7} K_p / T_i$
2	$1.543 \cdot 10^{-7}$	$5.7482 \cdot 10^{-6} - 1.15 \cdot 10^{-7} K_p$	
3	c_1	c_2	
4	d_1		
5	e_1		

Table 7.4: *The Routh-Hurwitz table for the hydraulic positioning system*

The elements c_1, c_2, d_1 and e_1 are calculated:

$$c_1 = 1.032 \cdot 10^{-4} + 1.677 \cdot 10^{-10} K_p \tag{7.1}$$

$$c_2 = -1.15 \cdot 10^{-7} \frac{K_p}{T_i} \tag{7.2}$$

$$d_1 = 5.748 \cdot 10^{-6} - 1.15 \cdot 10^{-7} K_p - \frac{-1.774 \cdot 10^{-14} \frac{K_p}{T_i}}{1.032 \cdot 10^{-4} + 1.677 \cdot 10^{-10} K_p} \tag{7.3}$$

$$e_1 = c_2 \tag{7.4}$$

Since the first two elements in the first column of table 7.4 are positive, all elements in this column must be positive. This implies:

$$c_1 > 0 \overset{(7.1)}{\Longrightarrow} K_p > -6.191 \cdot 10^5 \tag{7.5}$$

$$e_1 > 0 \overset{(7.4),(7.2)}{\Longrightarrow} K_p < 0 \tag{7.6}$$

$$d_1 > 0 \overset{(7.3)}{\Longrightarrow} \begin{cases} \frac{\left(1.187 \cdot 10^{-11} + \frac{1.774 \cdot 10^{-14}}{T_i}\right) + \sqrt{\left(1.187 \cdot 10^{-11} + \frac{1.774 \cdot 10^{-14}}{T_i}\right)^2 + 4.576 \cdot 10^{-26}}}{3.857 \cdot 10^{-17}} > K_p \\ \frac{\left(1.187 \cdot 10^{-11} + \frac{1.774 \cdot 10^{-14}}{T_i}\right) - \sqrt{\left(1.187 \cdot 10^{-11} + \frac{1.774 \cdot 10^{-14}}{T_i}\right)^2 + 4.576 \cdot 10^{-26}}}{3.857 \cdot 10^{-17}} < K_p \end{cases} \tag{7.7}$$

Equations 7.5 and 7.6 limit K_p between $-6.19 \cdot 10^5$ and 0. Equation 7.7 connects T_i with K_p. The region of K_p and T_i values that result in a stable controlled system, are shown in Figure 7.8.

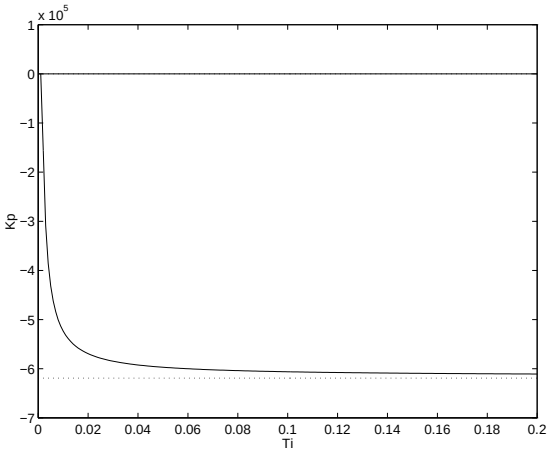


Figure 7.8: The region of K_p and T_i values that make the controlled system stable in closed loop. The region is bounded by the two solid curves. The dotted curve is the asymptote of the second curve for large T_i values (soft integral action). Observe that with a very strong integral action (T_i small), the system can be made unstable with a relative small proportional action.

△

7.4 Root Locus

Basically there are 3 concepts in Root Locus:

- The position of the closed loop transfer function poles in the s-plane determine the system's time response. Moreover, specific time behaviour like overshoot and settling time corresponds to a pole location in some well designated areas in the s-plane (for a second order system).
- If one parameter changes in the transfer function denominator, the poles will change position. Root locus method presents a way to draw the trajectory of the poles in the s-plane, when one parameter in the transfer function is varied.
- So, varying one parameter, the position of the poles changes. We will look for the parameter that makes the poles come into the required region that corresponds with the time domain specifications.

First the relation between pole position and time characteristic will be studied.
We'll start from the assumption that the system's response is basically determined by its two dominant poles (this is: approximating the system as if it were second order). This assumption is valid when

$$\mid \sigma_{dom} \mid \leq \frac{1}{10} \mid \sigma_{other} \mid,$$

in which σ_{dom} is the real part of the dominant poles and σ_{other} is the real part of the other poles.

Stability can easily be analysed, since the distance from the imaginary axis shows the stability degree. It is difficult however to translate a certain pole configuration into a phase and gain margin specification.

For a second order system $H(s)$,

$$H(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

with poles $-\omega_n\zeta \pm \omega_n\sqrt{\zeta^2 - 1}$, one finds that

- 2% Settling Time $T_s = \frac{4}{\zeta\omega_n}$.
  Since the real value of the complex pole pair is given by $-\omega_n\zeta$, the pole positions that correspond with a settling time equal or less than T_s lie in a left half-plane.
- Percent Overshoot= $100e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$.
  Since the angle α (see Figure 7.9) is given by

  $$tg(\alpha) = \frac{\sqrt{(1-\zeta^2)}}{\zeta},$$

  one observes that poles corresponding with equal overshoot, have the same α. Pole configurations with less or equal overshoot are therefore given by a cone in the s-plane.
- Oscillation frequency $\omega_d = \omega_n\sqrt{\zeta^2 - 1}$ This corresponds with the imaginary part of the poles.

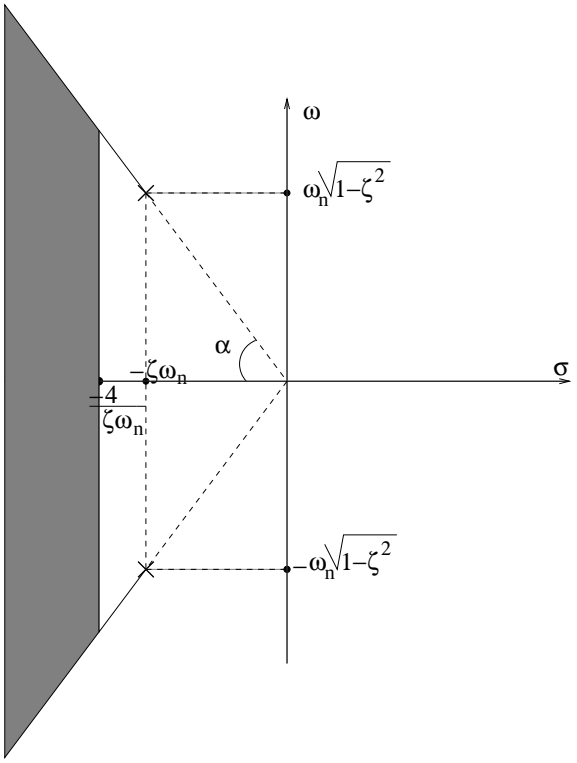


Figure 7.9: Region of overshoot and settling time

Given some constraints on settling time, overshoot and oscillation frequency one can thus draw some boundaries for pole location in the s-plane. We'll now study the influence of a varying parameter on the pole position. Consider a closed loop transfer function's denominator $d(s)$, given by

$$d(s) = \prod_{i}^{N} (s + p_i) + K \prod_{j}^{M} (s + p_j)$$

with

$$M \leq N.$$

We want to analyse the pole position for a varying K. The curve that describes the location of the poles (zeros of $d(s)$) in the s-plane in function of the variable K is called a root locus. These plots nowadays are very easily drawn with the aid of software packages. Therefore, we're not longer interested in manual plotting hints. Nevertheless, some properties are interesting when one analyses and designs a control system:

- When K varies from 0 till ∞, the poles of $d(s)$ move from p_i to p_j. If there are more p_i than p_j, the excess $N - M$ move towards infinity.
- If the excess is $N - M$, there will be $N - M$ asymptotes. Their angle ϕ with the real axis can be calculated as

$$\phi = \frac{2q + 1}{N - M}\pi$$

- The root locus will appear symmetrical with respect to the horizontal axis.
- The root locus only lies on a section of the real axis if that section has an odd number of p_i and p_j at his right hand side.

By placing extra poles and zeroes, one can thus influence the root loci.

Example: the hydraulic positioning system

Time domain specifications The design specifications for the controller are:

- No steady state error. This results in an integrator.
- No overshoot. The dominant pole(s) must lie on the real axis.
- 10 seconds settling time. The real part must be smaller than -0.4.

P-controller Although the zero-steady state error immediately implies an integral action, we'll first check the P-controller. The transfer function with numerator n(s) and denominator d(s) is given by

$$P(s) = \frac{n(s)}{d(s)}$$

With a P-controller, the closed loop transfer function becomes

$$G(s) = \frac{K_p P(s)}{1 + K_p P(s)} = \frac{K_p n(s)}{d(s) + K_p n(s)}$$

The denominator has the described form. We can therefore apply a root-locus algorithm. The result is shown in Figure 7.10a.

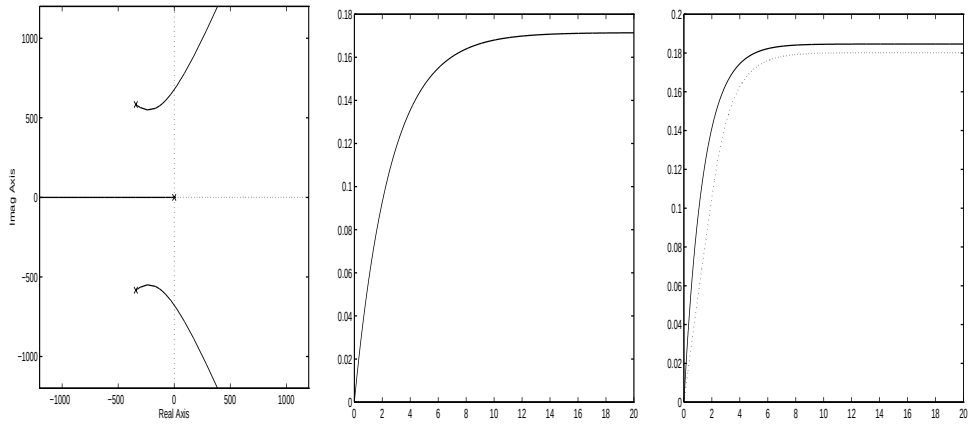


Figure 7.10: Root Locus for a P-controller. a. Root Locus diagram, b. Response for $K_p = -300$, c. Response for $K_p = -600$ (linear model : solid, with actuator saturation : dotted)

We observe 2 fast poles and one slow pole (the most right one). Notice the 3 asymptotes, according to the pole-zero excess.

- For $K_p = -300$, the root locus of the dominant pole crosses the -0.4 border (corresponding with 10 seconds settling time). The system response for $K_p = -300$ is shown in Figure 7.10b.
  Observe the steady state error (input reference step is $0.2m$).
- Doubling the gain ($K_p = -600$) results in an even more rapid response.
  The question might rise why we don't increase K_p even more.The answer is twofold.
  1. The first reason is a general remark: the real plant isn't linear but presents saturation in its actuators. A large K_p will nearly immediately result in a large control signal, and thus in saturation. It has no sense to increase K_p even more at this point.
     In Figure 7.10c the response is shown for $K_p = -600$. Dotted: the system's response with saturation included. Notice that saturation slows down the response!
  2. The second reason is that for large K_p values, the two other poles become dominant and even unstable!

PI-controller Now we'll analyse the effect of a PI-controller. In root locus diagrams unfortunately, it is impossible to draw the trajectories for two changing parameters.

- Therefore, we'll fix the K_p value on -600 and we'll vary T_i.

  The closed loop transfer function is

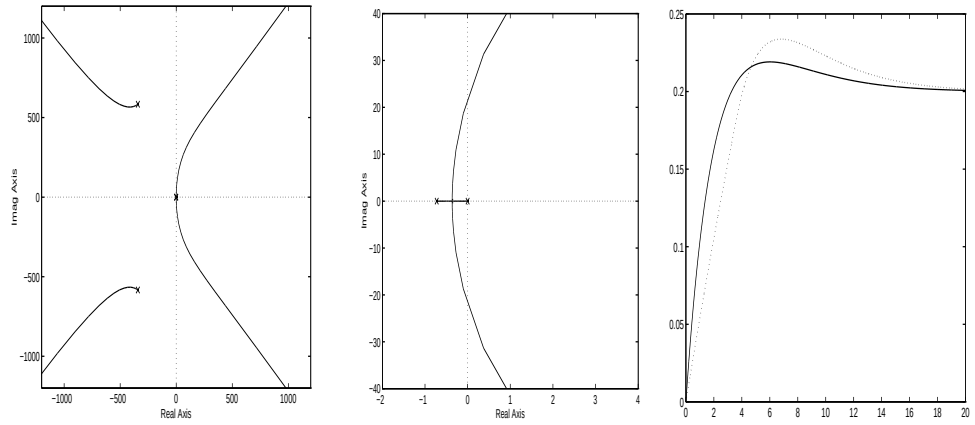
$$\begin{aligned} G(s) &= \frac{\frac{n(s)}{d(s)} K_p \left(1 + \frac{1}{T_i s}\right)}{1 + \frac{n(s)}{d(s)} K_p \left(1 + \frac{1}{T_i s}\right)} \\ G(s) &= \frac{n(s) K_p (T_i s + 1)}{n(s) K_p + T_i (d(s) + n(s) K_p s)} \end{aligned}$$

  In this way it is written in the standard root locus form, except for the fact that the first term of the denominator has a larger order than the second one. Therefore we'll work with $K_i = 1/T_i$ instead of T_i.
  The transfer function can now be written as:

$$G(s) = \frac{n(s) K_p (s + K_i)}{(d(s) + n(s) K_p s) + n(s) K_p K_i}$$

  A root locus of this transfer function for varying T_i is shown in Figure 7.11a.

  Due to the extra pole in zero, 4 asymptotes are now distinguished. The dominant region (close to the imaginary axis) is not clear though.
  In Figure 7.11b is zoomed in into that region.
  A break-away point is noticed in -0.36. Therefore it is impossible to reach the -0.4 border, since always one of the two dominant poles will be right from the -0.36 point.
  In Figure 7.11c the step response is plotted (with and without saturation) for T_i corresponding with the K_i-value for both poles in -0.36.

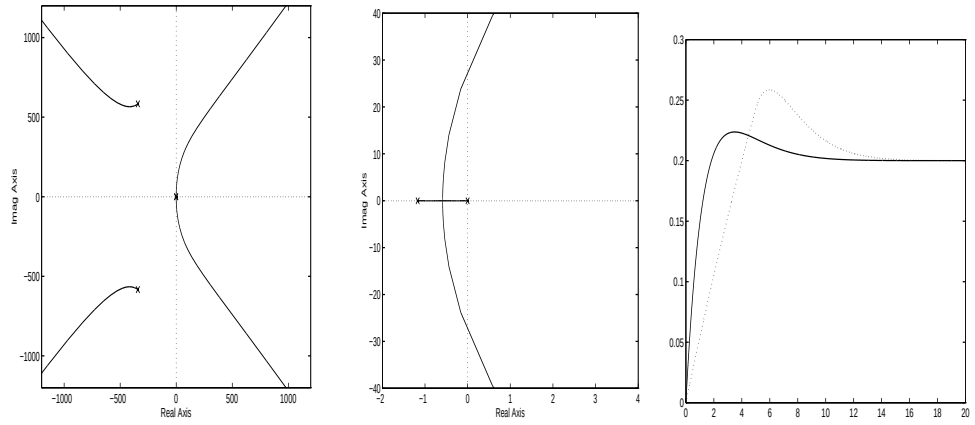


a b c

Figure 7.11: Root Locus for an I-controller. a. Root Locus diagram, b. Zoom into the dominant region, c. Response for $K_p = -600, T_i = 5$ (linear model : solid, with actuator saturation : dotted).

Notice the zero steady-state error. The overshoot that is present (although all the dominant poles are real) is completely due to the zero introduced by the PI-controller. This shows that the formulas calculating overshoot, settling time etc. are not completely right.

- With $K_p = -600$, we couldn't find a decent solution for T_i. Therefore we change the K_p value to -1000. The root locus and a zoomed-in version are shown in Figure 7.12a and 7.12b.



a b c

Figure 7.12: Root Locus for an I-controller. a. Root Locus diagram, b. Zoom into the dominant region, c. Response for $K_p = -1000,\ T_i = 3$ (linear model : solid, with actuator saturation : dotted)

- With a larger T_i-value the overshoot is cancelled nevertheless ($T_i = 15$) as shown in Figure 7.13. With saturation included, the response is worse, but with anti-windup ($T_a = 15\,000$) it gets better again.

△

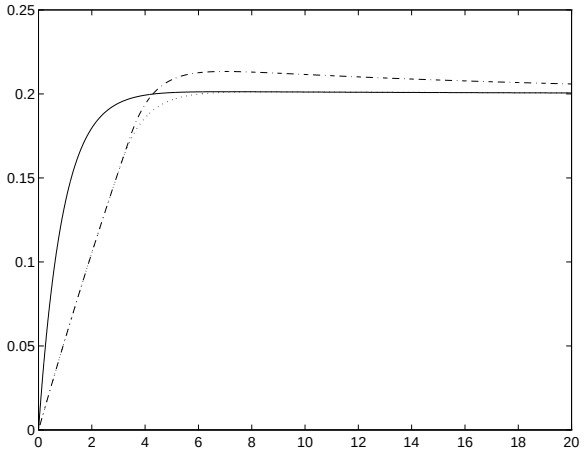


Figure 7.13: Step response for a larger T_i value (15) without saturation (solid). With saturation: dashed. With saturation and anti-windup: dotted

Note: In Matlab the next functions can be used: `rlocus, rlocfind, cloop`. Since Matlab calculates the root locus for positive gains, one has to work with the negative numerator.

7.5 Bode Plots

Bode plots are good tools to analyse robustness of a closed loop system. Robustness can be expressed as Phase and Gain margin. Take care: both indicators are measures on the open loop plant, which give information about the closed loop plant!
The *gain margin* gives us the allowable gain variation of the plant and controller before the closed loop system becomes unstable.
The *phase margin* gives the allowable phase variation (or time delay) before the closed loop system gets unstable (Figure 7.14).

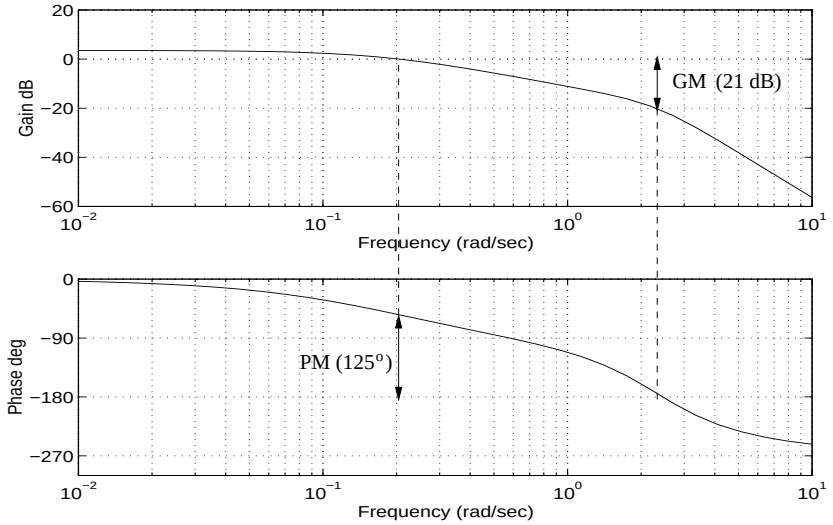


Figure 7.14: Phase Margin (PM) and Gain Margin (GM)

The impact of the different PID-actions is shown in the next Figures.

7.5.1 P-control

In case of proportional control, the controller transfer function is just a gain:

$$C(s) = K_p$$

and the control signal is proportional to the control error.
From the Bode plot of a proportional controller (Figure 7.15), it follows that the loop gain will be multiplied by K_p, while the phase shift is not influenced at all.
The purpose of proportional control is to increase the loop gain, in order to obtain a higher accuracy (DC-gain) and a large cross-over frequency. However, due to stability requirements, K_p is limited. For plants with a tendency to oscillate, one can easily verify that the rise time T_r decreases, but that the settling time T_s and the overshoot os often increase.

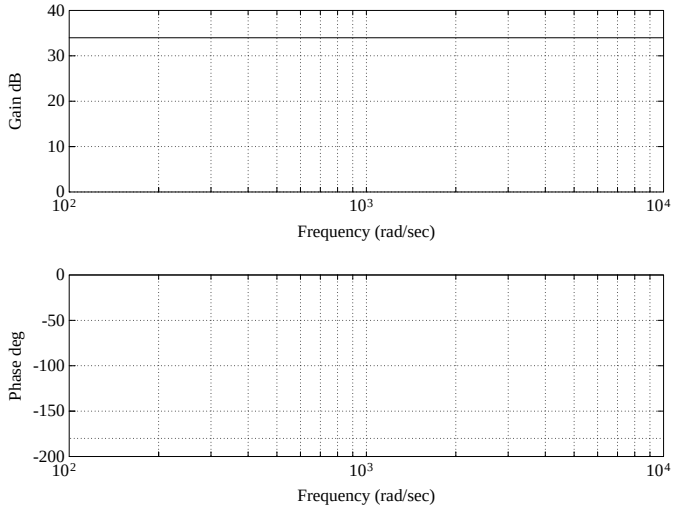


Figure 7.15: The Bode plot of a P-controller ($K_p = 50$).

7.5.2 I-control

In case of integral control, the controller transfer function is just an integrator:

$$C(s) = \frac{K_p}{sT_i}$$

and the control signal is proportional to the integral of the control error.
From the Bode plot of an integral controller (Figure 7.16), it follows that the loop gain will be multiplied by $\frac{K_p}{T_i}$ at $\omega = 1$ and that an additional phase shift of -90^o will be introduced.
The purpose of integral control is to make the loop gain at DC equal to ∞, in order to obtain a zero steady state error. However, the additional phase shift forces the designer to decrease the bandwidth, to ensure relative stability (so $\frac{K_p}{T_i}$ is limited).

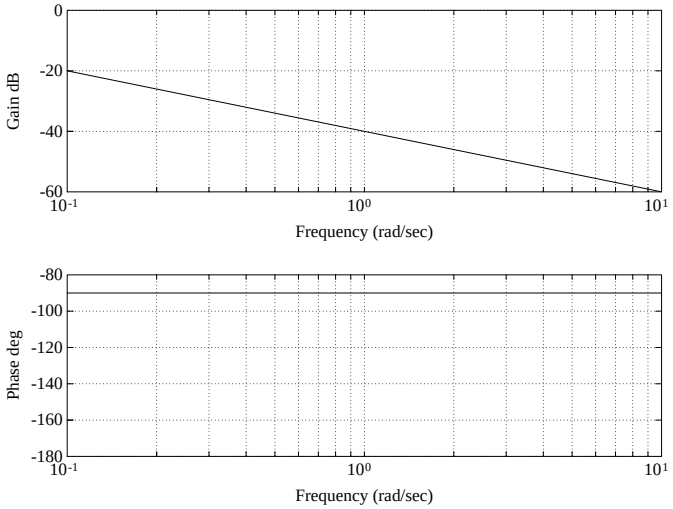


Figure 7.16: The Bode plot of a I-controller ($K_p = 1, T_i = 100$).

7.5.3 D-control

In case of proportional control, the controller transfer function is just a derivative term:

$$C(s) = sK_pT_d$$

and the control signal is proportional to the derivative of the control error.
From the Bode plot of a derivative controller (Figure 7.17), it follows that the loop gain will be multiplied by K_pT_d at $\omega = 1$ and that an additional phase shift of $+90^o$ will be introduced.

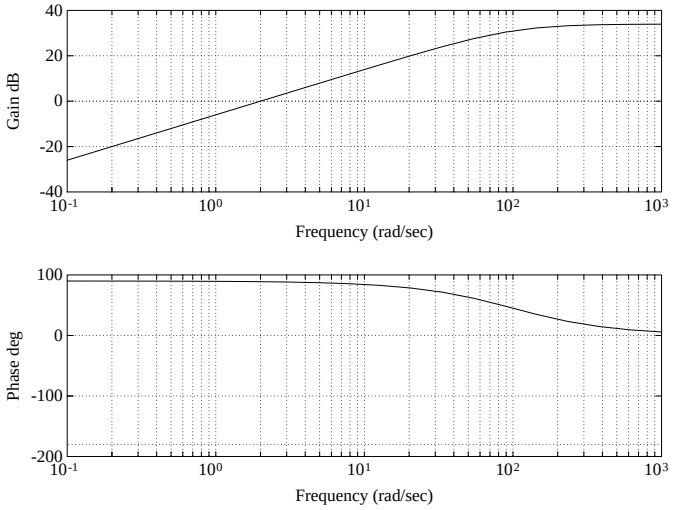


Figure 7.17: The Bode plot of a D-controller ($K_p = 1,\ T_d = 0.5,\ t_c = 0.01$). The derivative action is bounded for frequencies $\omega \geq \frac{1}{t_c} = 100$.

The purpose of derivative control is to increase the response speed (decrease T_r). However, due to the amplification at high frequencies (noise sensitivity), the derivative action must be limited. In theory, it would be sufficient to place an extra low-pass filter $\frac{1}{st_c+1}$ before the

differentiator, to 'remove' the high frequency components in the error signal. In practice, the low-pass filter and the derivative action are implemented by a single filter. In that case, the controller transfer function is:

$$C(s) = \frac{sK_pT_d}{st_c + 1}$$

7.5.4 PI-control

In PI-control, the advantages of P- and I-control are combined. P-action is added for bandwidth and speed, the I-action is added to guarantee a zero steady state error.
From the Bode plot of a PI-controller (Figure 7.18), it follows that a high loop gain is introduced for lower frequencies, while the additional phase shift vanishes for higher frequencies. The positive and the negative effects of PI-control become more outspoken by making the integration time T_i smaller: the high gain for low frequencies increases, but the phase margin reduces.

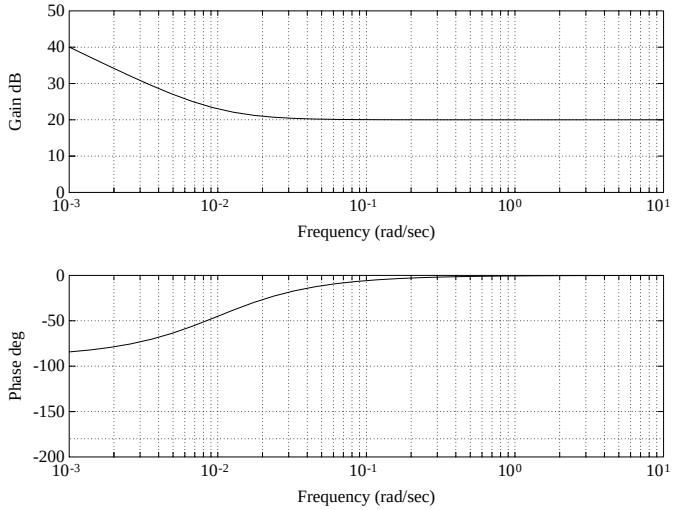


Figure 7.18: The Bode plot of a PI-controller ($K_p = 10,\ T_i = 100$). The integral action is higher for frequencies $\omega \leq \frac{1}{T_i} = 0.01$.

7.5.5 PD-control

In PD-control, the advantages of P- and D-control are combined. P-action is added for bandwidth and speed, the D-action to further decrease the rise time (while P is limited by stability requirements).
From the Bode plot of a PD-controller (Figure 7.19), it follows that a high phase lead and a high gain are introduced for higher frequencies: the speed of response increases, together with the sensitivity to noise. The positive and the negative effects of PD-control become more outspoken by making the integration time T_d larger: the phase lead for high frequencies increases, but the noise sensitivity increases also!

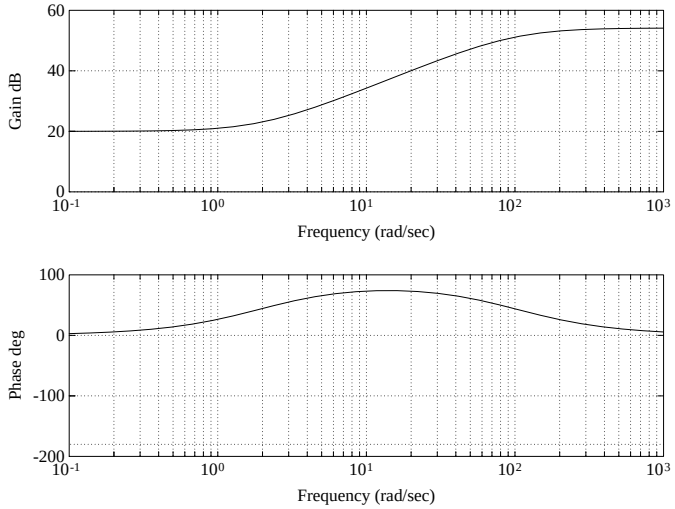


Figure 7.19: The Bode plot of a PD-controller ($K_p = 10,\ T_d = 0.5,\ t_c = 0.01$). The derivative action is bounded for frequencies $\omega \leq \frac{1}{T_d} = 2$ or $\omega \geq \frac{1}{t_c} = 100$.

7.5.6 PID-control

An attempt to combine all the previous advantages, while accepting the corresponding disadvantages, is the PID-controller, which is described in a previous section. It is easy to understand that the relative sizes of the different controller parameters determine which effects become more/less important. By analysing the Bode plot of the PID-controller (Figure 7.20), it is possible to do some predictions on the closed loop behaviour.

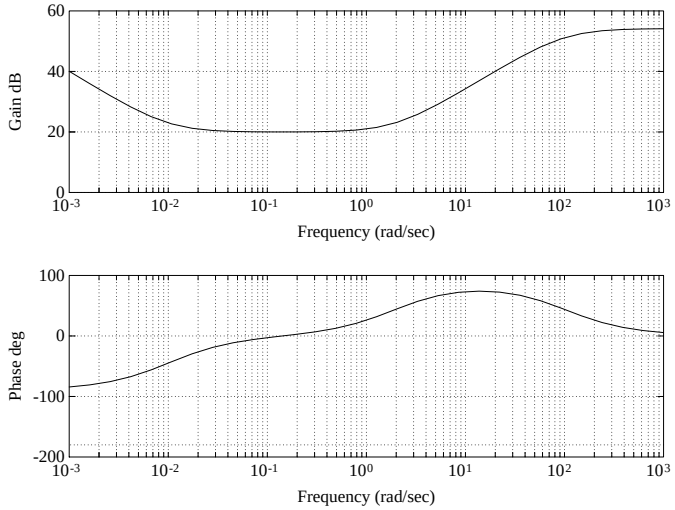


Figure 7.20: The Bode plot of a PID-controller ($K_p = 10,\ T_i = 100,\ T_d = 0.5,\ t_c = 0.01$).

7.6 Auto-Tuners

Recently controllers were developed that automatically tune their PID-values. After a fixed time (e.g. every hour) the controller is switched off, and a simple plant model is derived by e.g. step-test. Based on that model, adequate values of K_p, T_i and T_d are calculated (e.g. with the Ziegler-Nichols tuning rules). Then the controller is connected again to the system. This method is very advantageous in slightly time variant processes. A typical example of a time variant process is the winding of cloth on a reel. Since the diameter of the cloth roll is ever increasing, the moment of inertia changes too. This way the dynamics are continually varying.
When the process becomes too time-variant, these auto-tuners do not work properly. The system model is varying too fast between two 'tunes', such that the calculated PID-parameters are only valid right after the tuning moment. It must be said though that these auto-tuners do not always find a satisfactory result. Normally there is few flexibility in choosing your own tuning rules and thus your own specifications.

Instead of tuning every while, one can imagine that the identification part is done constantly (real-time), hence resulting in an adaptive control scheme. This is the area of adaptive tuners.

7.7 Optimal Control

A rather straightforward method to tune the free parameters of a controller is to use a *quantitative measure of the performance* of the controlled system. The controlled system then is called *an optimum control system*, i.e. the free closed loop controller parameters are adjusted so that the *performance measure* reaches an extreme, commonly a minimum value. Problem with such an optimum control system is that *either* the computational complexity *or* the control structure implementation is rather difficult to tackle. If one uses for example a rather simple control structure like a compensator PID controller then it turns out that a generic solution of the free parameters of the controller reveals a non convex problem (i.e. the performance measure contains local minima with regard to e.g. τ_d), hence resulting in several 'sub-optimal' or *local* solutions. On the other hand, if one uses for example a more difficult to implement state feedback instead of output feedback controller strategy, one is able to exploit state dependent performance measures, which can be defined as a convex problem. This last has inherently only one, and hence global, optimum which can be obtained algebraically (e.g. LQR related problems). This is explained short in the sequel.

7.7.1 Output feedback optimum control strategies

In this section only a basic generic procedure is described. Assume for example a nonlinear hydraulic transport system to be controlled. The nonlinear process is shown in Figure 7.21.

If linear identification strategies fail to describe this system, since for example the system does not operate in a single operating point, one can use the nonlinear description of the process to tune the free controller parameters. Improved linear controller design strategies could possibly be used, like in adaptive control, split range control, etc. But the interpretation of the design specifications (like settling time, peak overshoot, etc) is not straightforward any more.
An easy design specification of a controller, while maintaining the basic PID compensation

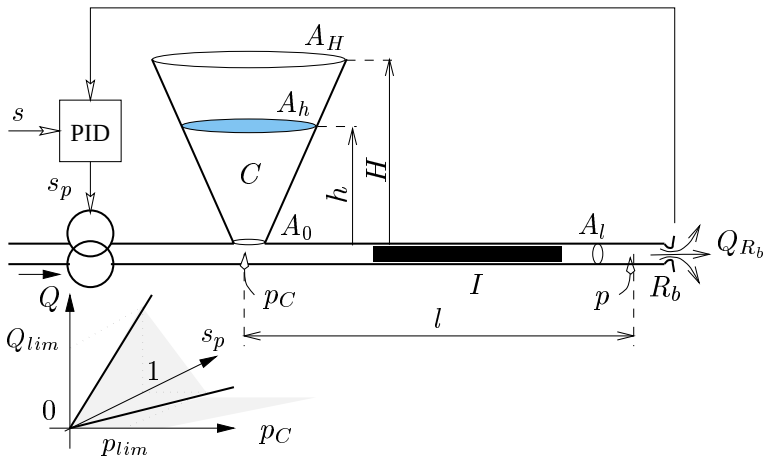


Figure 7.21: Control of a nonlinear hydraulic transport system. The control problem is a tracking one: Q_{R_b} should follow s.

control structure, is observing the measurable variable of the process, e.g. the output $y(t)$, or one related with it, e.g. $e(t) = r(t) - y(t)$. A *calculable quality measure* of the closed loop system then has the following general form

$$J = \int_0^T f(r(t), y(t), e(t), t) \cdot dt \tag{7.8}$$

This scalar quality measure J is also called the *performance index*, which can be used as a basis for comparing the performance of the controlled system. Consequently the function $f(\cdot)$ must answer to this 'norm' criterion. Due to the definition of J, this kind of optimization belongs to the time-domain optimization methods. In this sub-section two classes of quality measures in the time domain are outlined.

Classical time-domain performance measures

Below four well known performance indices are proposed:

The IAE index i.e. the integral of the absolute error:

$$J = \int_0^T |e(t)| \cdot dt$$

The ITAE index i.e. the time weighted IAE:

$$J = \int_0^T t \cdot |e(t)| \cdot dt$$

The ISE index i.e. the integral of the square error:

$$J = \int_0^T e(t)^2 \cdot dt$$

The ITSE index i.e. the time weighted ISE:

$$J = \int_0^T t \cdot e(t)^2 \cdot dt$$

The free controller parameters can be tuned by means of the following generic (constrained) optimization routine:

`minimize:` J
`subject to:` the closed loop system description

Some remarks:

1. The final value of J, and consequently also the free controller parameters, depends highly on the chosen index, the integral horizon time T, and the form of the input signal $r(t)$. Therefore it is useful to verify several 'possible good outcomes'.

2. Since in most cases the minimization problem is not convex, the outcome can be 'sub-optimal'. Therefore it is a good habit to check this 'sub-optimal' result with different optimization algorithms, start values and input shape values.

3. It can be proven that for a second order linear process the ITAE norm provides the best selectivity with respect to the free controller parameters.

4. There exist no direct connection between these performance indices and the more common used design specifications like 'settling time', 'percent overshoot', etc[1]. On the other hand, the I(T)AE and I(T)SE is often also of practical significance (energy related for example).

Some of these items will be illustrated by means of the following worked out example.

Example 3 *Consider the nonlinear hydraulic transport system, shown in Figure 7.21. The constituent relations of the basic hydraulic components are summarized below (ρ is the density of the transported liquid):*

$$\dot{p}_C = \frac{1}{C} \cdot Q_C \qquad \text{with } C = \frac{A_0 + \left(\frac{A_H - A_0}{\rho \cdot g \cdot H}\right) \cdot p_C}{\rho \cdot g} \tag{7.9}$$

$$\dot{Q}_{R_b} = \frac{1}{I} \cdot (p_C - p) \qquad \text{with } I = \frac{\rho \cdot l}{A_l} \tag{7.10}$$

$$p = R_b \cdot Q_{R_b} \qquad \text{with } R_b = \frac{\rho}{2 \cdot A_l^2} \cdot |Q_{R_b}| \text{ (Bernoulli dissipator)} \tag{7.11}$$

$$Q = f(s_p, p_C) \qquad \text{with } f(\cdot) \text{ the non-linear pump characteristic according to Figure 7.21} \tag{7.12}$$

Consequently the hydraulic inertia I is the only linear component in the circuit. The SimuLink block diagram of the closed loop system is shown in Figure 7.22.

The block diagram of Figure 7.22 (top) has been parameterized and saved as `opt_mod.m`*. Once this has been done, three other* `.m` *files should be created:*

[1] These design specifications will be recalled in the next subsubsection.

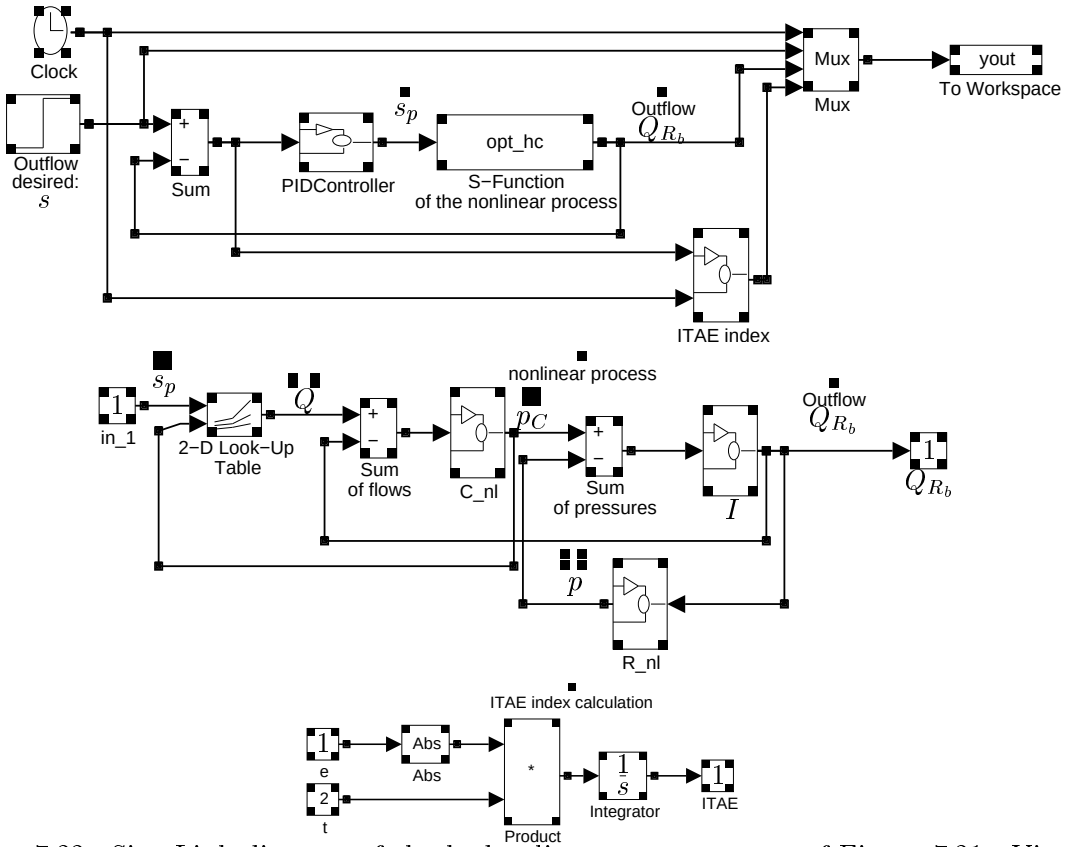


Figure 7.22: SimuLink diagram of the hydraulic transport system of Figure 7.21. View on the control configuration (top), view on the nonlinear hydraulic circuit (middle), and view on the calculation of the Performance index ITAE.

1. **The parameter file** *, e.g.* ***opt_par.m.***

The following file assigns all the fixed physical parameters in the Matlab workspace.

```
%        opt_par.m
%calculation of the physical parameters
%
%%%%constants
%physical
g=10;
rho=1E3; %water
%non ideal pump 2Dtable Q(p_C,s_p)
plim=1.5e4;
qlim=1;
%compliance
A_0=0.15;
A_H=7.4;
H=2;
%pipe
l=7;
A_l=43E-4;
%
%%%%calculation
%compliance  C = C_k0 + C_k * p_C
C_k0=A_0/(rho*g);
C_k=(A_H-A_0)/(((rho*g)^2)*H);
%pipe
I=rho*l/A_l;
%bernoulli resistance (free outlet) R_b = R_b_r * abs(Q_R_b)
R_b_r=rho/(2*A_l*A_l);
```

For convenience the step response of the open loop hydraulic process of Figure 7.22 (bottom) has been shown in Figure 7.23.

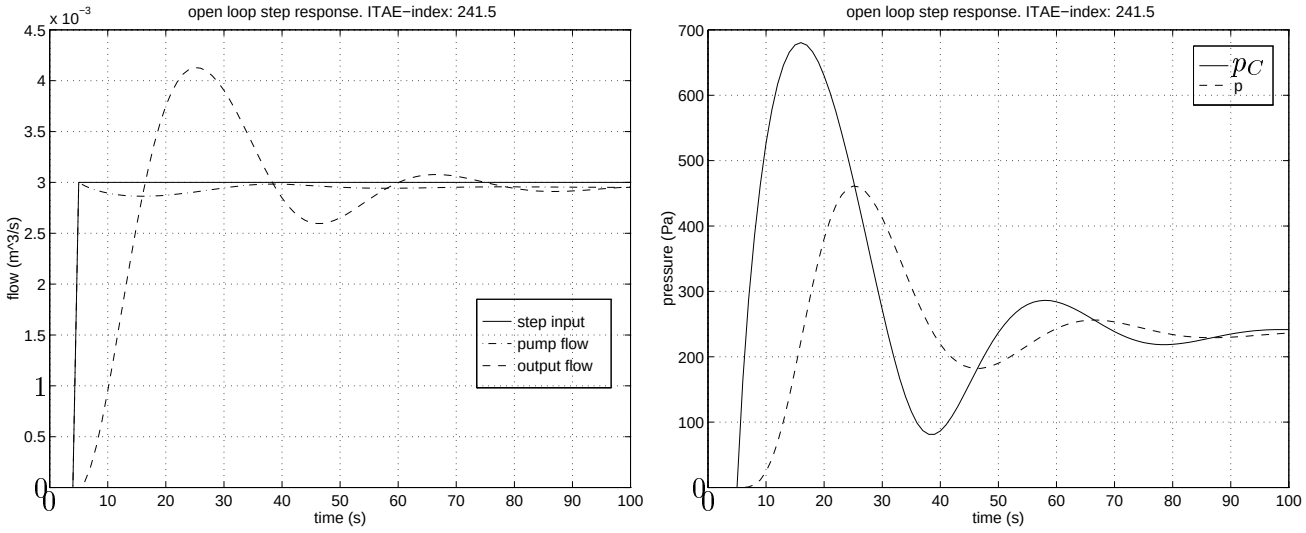


Figure 7.23: SimuLink diagram of Step response of the open loop hydraulic transport system of Figure 7.21.

2. The simulation file *, e.g.* ***opt_fcn.m.***

The following file simulates the SimuLink block diagram ***opt_mod.m****, without the necessity to enter SimuLink interactively. It also plots the time response.*

```
function [J,c]=opt_fcn(x)
%
%function opt_fcn.m simulates the simulink model (opt_mod.m)
%and plots the response
%
global K_P  T_i T_d yout
%first simulate the nonlinear SimuLink model
%with the free controller parameters K_P, T_i and T_d,
% and calculate simultaneously the Performance Index
%
%assign the PID controller
K_P=x(1);
T_i=x(2);
T_d=x(3);
%assign the simulation time horizon
tf=100;
%initializations for integration algorithm
xi=[0,0,0,0,0];  % Initial conditions (outputs at t=0) of the integrators
options(1)=1e-8; %relative error
options(2)=1e-14; %minimal step size
options(3)=0.01; %maximal step size
options(5)=3; %display all warnings
options(6)=2; %plot off
%start simulation
rk45('opt_mod',tf,xi,options);
%assign the cost function
J=yout(length(yout(:,4)),4);
%assign the constraint function c<=0 (only necessary if invoked by 'constr')
%physical constraints on the PID parameters:
c(1)=-K_P+0.05;c(2)=K_P-0.2;
c(3)=-T_i+0.3;c(4)=T_i-0.7;
c(5)=-T_d+eps;c(6)=T_d-200;
%
%eventually plot the response and the ITAE-index
plot(yout(:,1),yout(:,[2]),'-',yout(:,1),yout(:,[3]),'--')
xlabel('time (s)');
ylabel('flow (m^3/s)');
legend('desired flow','actual flow');
title(['Process with PID controller (K_P=',num2str(K_P),...
',T_i=',num2str(T_i),',T_d=',num2str(T_d),'). ITAE-index: ',num2str(J)]);
grid
hold on
drawnow
```

To check the correctness of ***opt_fcn.m*** *this file is executed with controller parameters* $K_p = 0.1$, $T_i = 0.2$, $T_d = 100$. *To do this, type at the Matlab command prompt the following:*

```
opt_par;
K_P=0.1;T_i=0.2;T_d=100;
opt_fcn([K_P T_i T_d]);
```

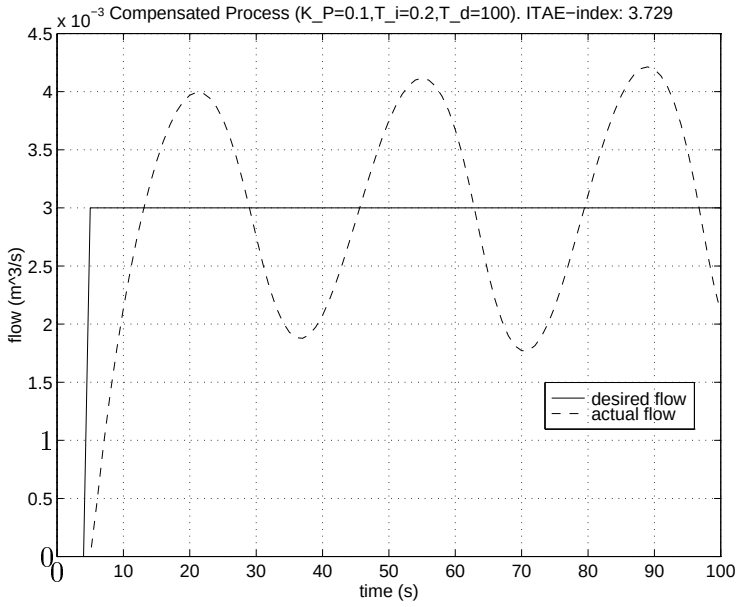


Figure 7.24: Step response of the closed loop system, Performance index calculated. Clearly the system becomes unstable under the chosen controller parameters. If the hydraulic system was linear, then an increase of only K_P or a decrease of only T_i would result in an even more unstable response.

The result is shown in Figure 7.24.

3. The optimization file *, e.g.* ***opt_go.m.***

The following file invokes the optimization (minimizing the ITAE index J) by means of changing the free parameters of the PID (K_P, T_i and T_d) within the available physical region (constrained optimization):

$$\begin{aligned} 0.05 \leq &\quad K_P &\leq 0.2 \\ 0.3 \leq &\quad T_i &\leq 0.7 \\ 0 \leq &\quad T_d &\leq 200 \end{aligned}$$

(These are the constraints $c(1)$ to $c(6)$ included in the file ***opt_fcn.m****)*

```
%      opt_go.m
% runs a constraint optimization routine
% on  the file opt_fcn.m
%
% first initialize the free parameters
K_P_init=0.1;
T_i_init=0.2;
T_d_init=100;
x0=[K_P_init,T_i_init,T_d_init];
%then initialize some optimization settings
options(1)=1;    %display intermediate results
options(2)=1e-9; % termination tolerance for x
options(14)=60;%maximum number of iterations
% the optimization routine
x=constr('opt_fcn',x0,options)
```

This file should be run as follows:

```
global K_P T_i T_d yout
opt_go
```

After 61 iterations it turns out that the optimization yields the following result:

$$\begin{aligned} K_p &= 0.2 \\ T_i &= 0.5268 \\ T_d &= 100.6 \end{aligned}$$

with ITAE index about 0.174. *If more refinement iterations are allowed (by increasing* `options(14)` *in the m-file* `opt_go.m`*) it turns out that after 143 iterations the second and sixth constraints* ($c(2)$ *and* $c(6)$*) are active:*

$$\begin{aligned} K_p &= 0.2 \\ T_i &= 0.3388 \\ T_d &= 200 \end{aligned}$$

The ITAE index becomes 0.135. *These two optimizations reveal a rather small influence of* T_d *on the ITAE index.*
The step response of the first optimized PID setting is shown in Figure 7.25. This response should be compared with Figure 7.23 (left).

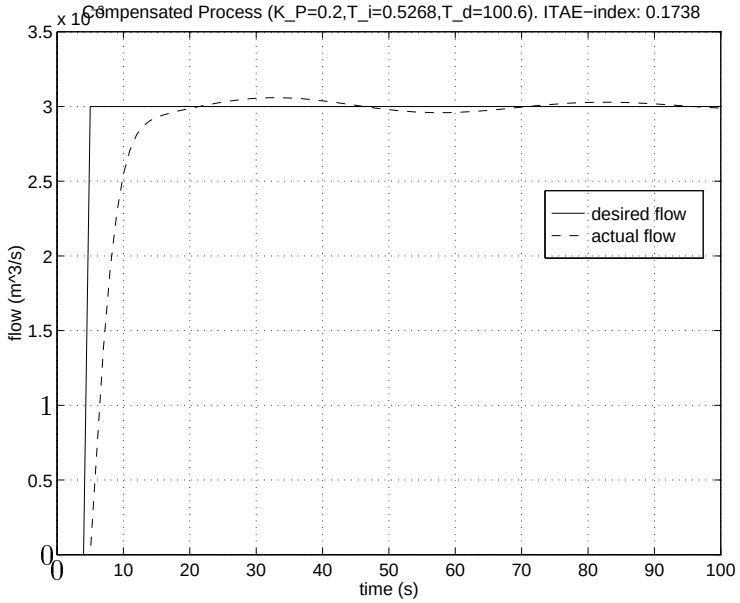


Figure 7.25: Step response of the closed loop system with optimized controller parameters. Performance index calculated.

To be complete, a view of the ITAE-index in the free controller parameter space (K_P, T_i, T_d) *is given in Figure 7.26. This Figure shows that, within the physical relevant region of the free controller parameter space, possibly only one global minimum of the ITAE index exist.* △

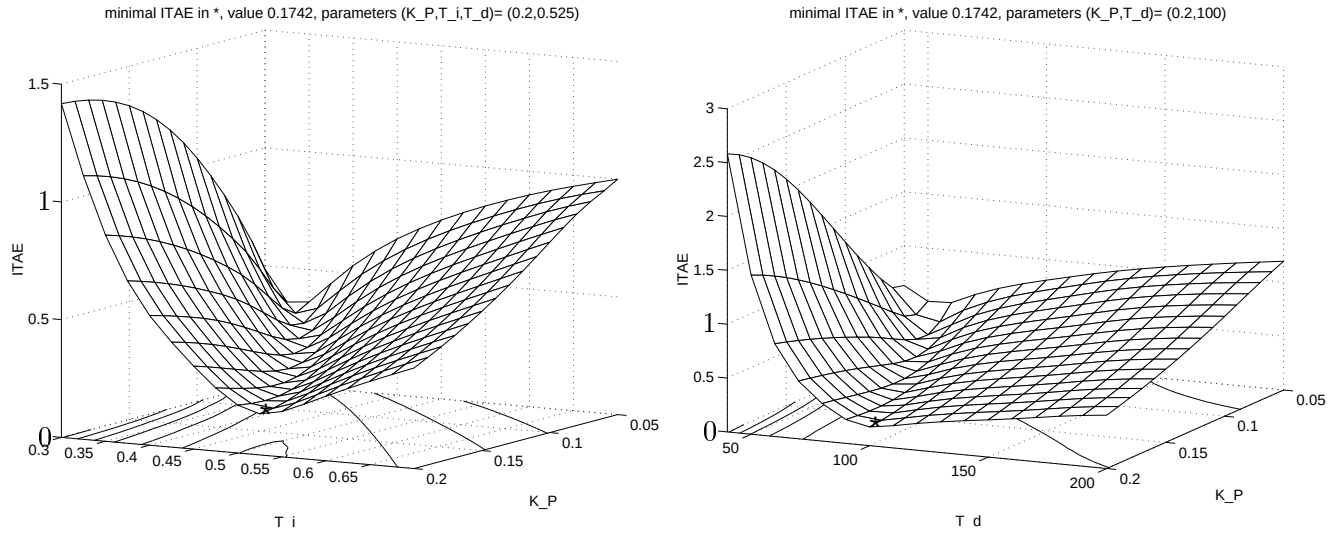


Figure 7.26: Performance index ITAE in the controller parameter space. Due to the calculation in discrete points of the parameter space the calculated optimal ITAE is an approximation. In the Figure left T_d is kept constant to 100, in the Figure right T_i is kept constant to 0.53.

Some important guidelines

Before explaining a second class of time domain performance measures, first some important guidelines are given.

1. The (sub-)optimal solution can depend highly on the initial values of the free parameters and the number of iterations. If e.g. the optimization routine in previous example is invoked with initial values $K_P = 0.1$, $T_i = 0.2$ and $T_d = 150$ the following result is achieved after 58 iteration steps:

$$\begin{aligned} K_p &= 0.2 \\ T_i &= 0.3925 \\ T_d &= 150.3 \end{aligned}$$

   The ITAE index is then about 0.152, hence again revealing a rather small influence of T_d on the ITAE index. Remark that Figure 7.26 shows that, within the physical relevant region of the free parameter space, possibly only one global minimum of the ITAE index exist. Hence the increase of the number of iterations steps will more than likely converge to the same optimal free parameters, independent of their initial values.

2. If *linear and nearly linear processes* are to be simulated it is a good habit to use in SimuLink `Linsim` as integration algorithm (accessible from `SimuLink - Options - Parameters` or from the command line with the `linsim` function).
   For *non-linear processes* `RK23` and `RK45` is more appropriate.
   In the special case of *stiff processes* (i.e. systems containing algebraic loops or having well spaced eigenvalues) the integration algorithm `Gear` should be used.
   In all these integration algorithms an adaptive integration step algorithm is included.

However it is convenient to set manually some of these boundaries. E.g. the `Minimal time step` should be chosen according to (i.e. smaller than) the fastest poles of the process to be simulated, the `Relative tolerance` should be chosen according to the desired simulation accuracy, and the `Maximal time step` should be chosen to speed-up the simulation when the system behaves smoothly.
In every case it should be verified that the obtained simulation results can be *repeated* with different integration parameter settings! Therefore it is extremely important to check possible dangerous locations in the free controller parameter space by single pre-optimization simulations of `opt_fcn.m`. As a rule of thumb large K_P together with small T_i values, or large T_d values are of interest. However, the nonlinear behavior plays in most cases also an important role. So be warned: **never trust a simulation at first glance!**

3. It is a good habit to simulate (and verify) the open loop process first, thereafter the compensated process can be observed. The starting values of the parameters of the PID compensator can be obtained by means of the already mentioned classical identification methods (eye-ball methods, exact Linearisation in a working point) and tuning methods (stability limit method for example). Notice that for stable nonlinear systems the obtained PID parameters can **de**stabilize the closed loop system. However, as long as only 'small' deviations of the input and state are allowed this should not be the case. Therefore verify that the steady state input and state of a nonlinear system and its linearized version can differ, even in the Linearisation point.

4. If one wants to use a simple PID compensator controlling a highly nonlinear process, then an optimal PID is not always the one obtained from one kind of step input profile (as is done here). However, using the same optimization strategy as is outlined in the previous guideline above one can optimize over a range of step inputs, eventually including a weighting of certain important working points. Comparing with a 'one step input' optimized PID controller, such an optimized PID controller has in general a lower performance in one specific working point, but on the other hand has a better performance in a larger range of input signals. Computer simulations show that the latter also has a higher robustness against nonlinear parameter changes. It is left for the reader to verify this statement for this example.

5. In most cases the SimuLink description of the nonlinear process is not as exact as it should be. As example the Bernoulli resistor (free outlet) of previous example is given. In some cases it is not sufficient to model this resistor as is done in Eqn. (7.11). For example, in cases where $Q_{R_b} < 0$. If this occurs, the SimuLink diagram should be modified to prevent Q_{R_b} from being negative since this is physically impossible. Similarly, the actuator signals (the pump) should be verified also. The pump input signal s_p can have physical limitations, like saturation, which limits a lot the performance of the compensator. If such saturation exist, then this should be build in the SimuLink model of the pump. Such physical limitations are not always taken into account. But since they are in fact the most dominant components influencing the *real closed loop performance*, they should be included in the model previously to any 'computer based' optimization procedure. Notice that it is *not* sufficient to verify all this with one 'after optimization' simulation of the behavior of the controlled system, since the performance measure J can be influenced largely by such forgotten factors.

Response shape time domain performance measures

It is rather straightforward to use other performance measures than the ones mentioned in subsubsection 7.7.1. Again, this is explained by example:

Example 4 *Observe previous Example 3. Figure 7.27 shows the contours in which a step response should be shaped.*

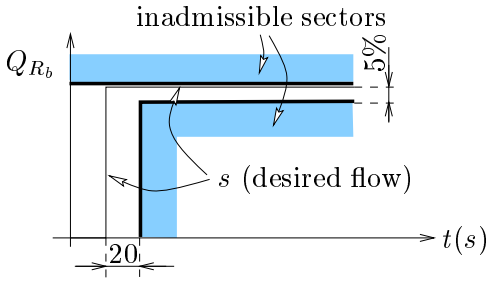


Figure 7.27: shape design.

This response shape corresponds with the following simultaneous time domain performance measures:

$$\begin{aligned} os &= 0 \\ e_{ss} &< 5\% \\ T_s &< 20sec \end{aligned}$$

For a step input s at time t_{step} this can be formulated mathematically as follows:

$$\begin{aligned} \forall t > t_{step} &: Q_{R_b} < s \\ \forall t > (t_{step} + 20sec) &: Q_{R_b} > 0.95 \cdot s \end{aligned}$$

These inequalities can be interpreted as soft bounds *or* hard bounds. *Soft bounds can be violated, hard bounds can not. In this example a soft bound implementation is given. Therefore the SimuLink model can be modified slightly, the ITAE index has to be replaced with another one. This is shown in Figure 7.28.*

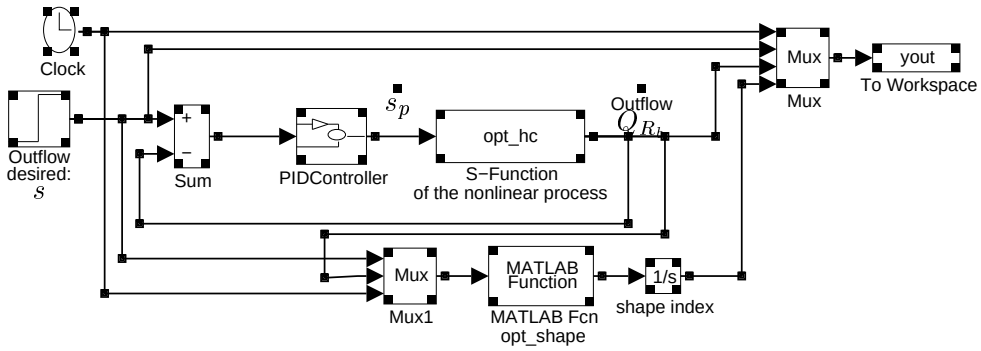


Figure 7.28: SimuLink diagram of the hydraulic transport system of Figure 7.21 with shape index calculation. View on the control configuration.

The matlab function `opt_shape.m` *is given below.*

```
function [y]= opt_shape(u)
%input u(1) should be s (desired flow)
%      u(2)           Q_R_b (output flow)
%      u(3)           time t
%calculates the instantaneous sum of os, ess and Ts
%no weighting will be used between os, ess and Ts
%it is assumed that the step input s occurs at t=5 sec

%
%calculation of weighted os (os should be 0)
os=0;
if (u(3)>5)&(u(2)>u(1))
  os=u(2)-u(1);
end
%simultaneous calculation of instantaneous ess (ess should be lees than 5%)
%and Ts (Ts should be less than 20 sec)
ess=0;
if (u(3)>25) %Ts specification (t_step=5sec)
  s_pct=0.95*u(1);
  if u(2)<s_pct %ess specification
    ess=s_pct-u(2);
  end
end
%calculate the weighted sum with relevant weights
y=100*os+u(3)*ess;
%summation over time is done in SimuLink diagram
```

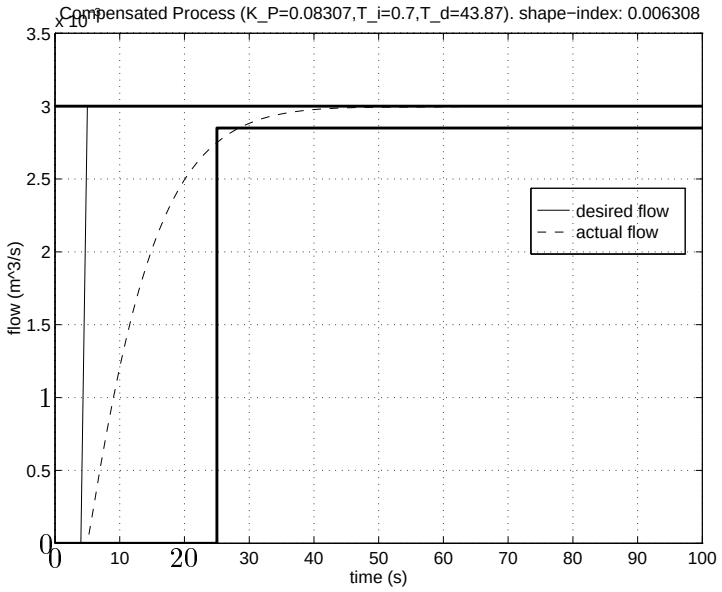


Figure 7.29: Step response of the closed loop system after 150 optimization steps. Shape index calculated. Due to the soft bounds the T_s-constraint is slightly violated. This error can be reduced by increasing the number of iteration steps, increasing the appropriate weight of the T_s-constraint (e.g. by modifying the penultimate line of `opt_shape.m`), or by making a hard constraint of the T_s-constraint (e.g. by managing the T_s as has been done with the physical constraints of the controller parameters, cfr. `opt_fcn.m`). This reponse should be compared with the ones depicted in Figures 7.23 (left) and 7.25.

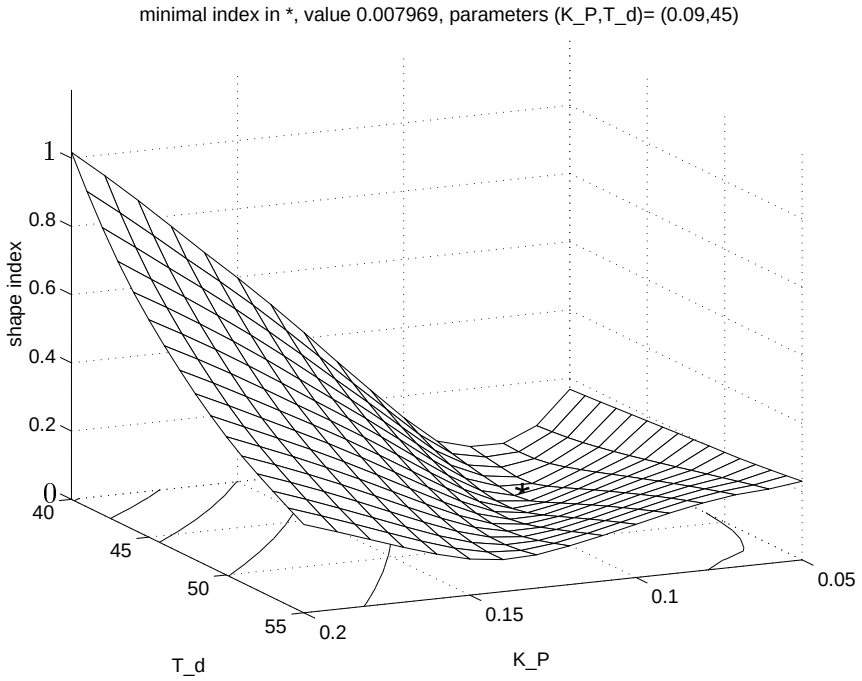


Figure 7.30: Shape index in a part of the free controller parameter space, $T_i = 0.7$.

The result of the optimization (after 150 iterations) is depicted in Figure 7.29. For convenience, in Figure 7.30 the shape index is visualized in a part of the achievable free controller parameter space. △

It should be clear that the same strategy can be used for linear systems. In that case the SimuLink diagram has to be replaced by the transfer function or state space description of the system. The problem formulation however remains in most cases non-convex. Hence such exhaustive brute-force computer simulation effort should be compared with the more classical tuning rules.

7.7.2 State feedback optimum control strategies

If a linear (or linearized) system is sufficient to describe the process behavior, then the *matrix state space description* of the process can be derived:

$$\dot{x} = A \cdot x + B \cdot u \tag{7.13}$$

with x the state vector of the process, u the process input vector, and A and B the system and input matrix. In fact, with a controlled system the process input u can be the controller output. If the state variables (or a part of it) can be measured, then the controller output can be defined with the following set of algebraic relations:

$$u = F \cdot x \tag{7.14}$$

with F the static feedback matrix (F contains no dynamics). It is easy to verify that the evolution of the state x depends on the feedback matrix F by substituting u (Eqn.(7.14)) in

Eqn.(7.13)

$$\dot{x} = \underbrace{(A + B \cdot F)}_{\tilde{A}} \cdot x$$

The control objective can be formulated with the minimization of a rather straightforward performance measure, the ISE index. But instead of using the error $e = r - y$, one now uses the state vector $x = [x_1 x_2 \cdots x_n]^t$. By using this state vector the control structure becomes much more difficult than in the output feedback control strategy. In some way one is forced to measure the states, hence more sensors and other control hardware is needed. But, on the other hand, it turns out that the computational complexity to tune such a PID is far more easy.
If the system contains only one state x_1, then the performance index is

$$J = \int_0^{t_f} x_1(t)^2 \cdot dt$$

And in general, if the system contains n states,

$$\begin{aligned} J &= \int_0^{t_f} \sum_{i=1}^{n} x_i(t)^2 \cdot dt \\ &= \int_0^{t_f} x^t(t) \cdot x(t) \cdot dt \end{aligned}$$

The relative importance of a certain state component with respect to another one can be taken into account with a so-called weighting matrix P. Besides the state, also the controller output is a measurable quantity, and hence can be included (in a weighted form) in the performance index.

$$J = \int_0^{t_f} (x(t)^t \cdot P \cdot x(t) + u(t)^t \cdot Q \cdot u(t)) \cdot dt \tag{7.15}$$

Such a problem is generally called a *Linear Quadratic Regulator* (LQR) related problem. *Linear* because the process is assumed to be linear (Eqn.(7.13)), *quadratic* because the performance index to be optimized is quadratic (Eqn.(7.15)), and *regulator* since the control problem is a regulator problem ($r = 0$, i.e. tracking is not subject of interest). Because the optimization is convex of nature (quadratic), the optimum is a *global* one.
It can be shown that if $t_f \to \infty$ this optimum control problem can be solved *algebraically* by means of a Riccati equation. This however is beyond the scope of this text, but will be exploited extensively in the next year.

A critical note: With regard to this introductory part on tuning a PID controller by means of straightforward optimization, one should be convinced that a *trade-off* between the simplicity of the control structure implementation (necessary hardware, etc.) and the computational difficulty of the tuning step should be made. This possibly also reveals why academia are more familiar with the latter (obtaining nice mathematics), while industry relies more on exploiting the former (obtaining easy implementations).

Chapter 8

Seminar tasks

8.1 Zitting 1

Oefening 1

$G(s)$ is de transferfunctie van een SISO systeem van 5de orde.

$$G(s) = \frac{-25086}{80} \frac{s^3 + 23s^2 + 56s - 80}{s^5 + 35s^4 + 419s^3 + 2531s^2 + 13048s + 25086}$$

pzmap

- de polen van $G(s)$ zijn: .....
  de nullen van $G(s)$ zijn: .....
  Is het systeem stabiel ? .....
  Waarom ? .....

step

- Teken de stapresponsie van het systeem.
  Bepaal volgende kenmerken:

zoom, ginput

  - stijgtijd $T_r =$ .....
  - inslingertijd $T_s =$ .....
  - doorschot $os =$ .....
  - onderschot $us =$ .....
    Verklaring (nullen): .....

- statische fout $e_{ss} =$ .....

  Volgens het eindwaarde-theorema: $e_{ss} =$ .....

  ... .

- de oscillatie-frequentie $f =$ .....

  Verklaring (polen): .....

bode
nyquist

- Teken Bode-plot en Nyquist-plot van het gegeven systeem.

  winst-marge $gm =$ .....

  faze-marge $pm =$ .....

  Wat kan je hieruit afleiden ? .....

  ... .

tf2ss

- Zoek een toestands-ruimtemodel dat het systeem beschrijft.

  aantal toestanden = ...

  $A =$

  $B =$

  $C =$

  $D =$

eig

De eigenwaarden van A zijn: .....

Verband met transferfunctie: .....

canon

- Bereken de modaal-canonieke vorm.

  Wat is er typisch aan dit toestands-ruimtemodel ? .....

  ... .

Oefening 2: Grafische identificatie

zp2tf, step

- Teken de stapresponsie van het systeem, beschreven door volgende polen en nullen:

  nullen: -20

  polen: $-16 \pm 6i$
  $-7 \pm 3i$
  $-1 \pm 0.2i$
  -4

- Bepaal m.b.v. een grafische identificatiemethode een benaderend model van lagere orde.

$H(s) = \ldots\ldots\ldots\

Oefening 3: Aan-uit regeling

simulink Start *simulink* in matlab.

File ↪ Open... Open het bestand /users/sista1/decock/seminarie/matlab/HUIS.m

Er verschijnt een model voor een huis (geïnteresseerden vinden meer informatie hierover in /software/matlab/toolbox/simulink/simdemos/thermdat.m).
Het model heeft twee ingangen:

- de warmtestroom afkomstig van het verwarmingstoestel (Qdot)

en

- de buitentemperatuur.

Het verwarmingselement kan enkel 'aan' of 'uit' staan. De ingang hiervan is dus **1** (aan) of **0** (uit).

De buitentemperatuur varieert sinusoïdaal (periode = 24 uur) met amplitude A rond een gemiddelde T_m.

De initiële temperatuur in het huis is T_{in}.

De variabelen A, T_m en T_{in} kunnen ingesteld worden in het matlab-venster.

Gevraagd:

Sources ↪Constant
Sinks ↪yout
Linear ↪Sum
Nonlinear ↪Sign, Relay
Connections ↪Mux

1. Ontwerp in simulink een aan/uit-regelaar die de temperatuur in het huis op 21° C houdt. Doe dit
   (a) zonder hysteresis
   (b) met hysteresis
   Varieer de variabelen en de hysteresis-waarden en beschrijf kort uw bevindingen.
2. Voeg witte ruis toe aan de temperatuursmeting. Welke aanpassingen moeten er gebeuren ?

Simulation ↪Parameters... ↪↪Gear

Save de bekomen oplossing en mail die .m-file samen met wat uitleg (o.a. instelling van variabelen) naar 'decock@esat.kuleuven.ac.be'.

Oefening 4: Stabiliteit

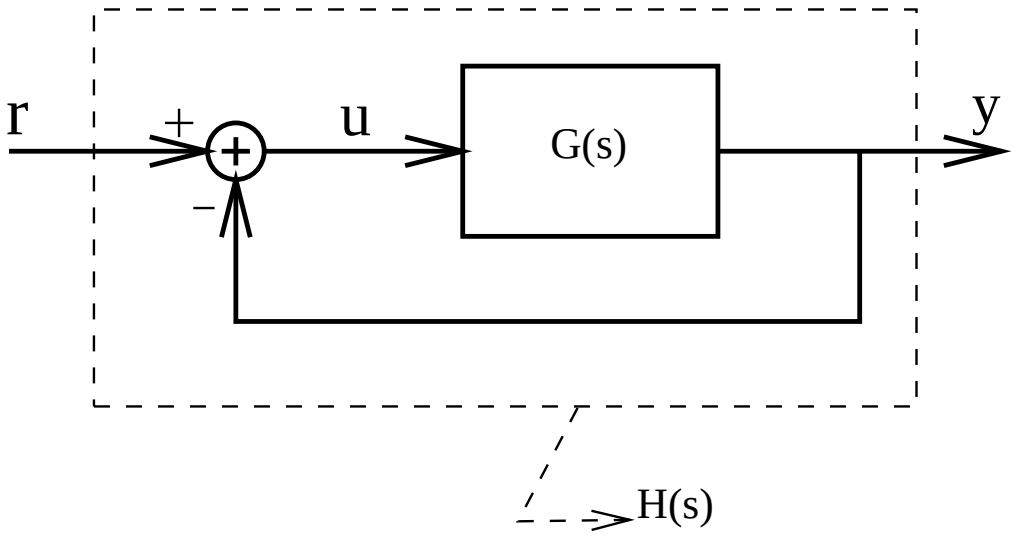


Figure 8.1:

In figuur 8.1 is $G(s)$ het systeem uit oefening 1. De uitgang y wordt (negatief) teruggekoppeld en een referentie-ingang r (gewenste uitgang) wordt aangelegd.

- Bereken de transferfunctie van het gesloten-lus-systeem in functie van de open-lus transferfunctie:

  $H(s) =$ .....

- Ga de stabiliteit van het gesloten-lus-systeem $H(s)$ na.

  ... .

  ... .

  ... .

  ... .

  Welke resultaten uit oefening 1 voorspelden dit ?

  ... .

  ... .

- Een versterkingsfactor K wordt toegevoegd (zie figuur 8.2).

  Tussen welke grenzen mag K liggen opdat $H(s)$ stabiel zou zijn ?

  ... .

bode
nyquist
rlocus
rlocfind

. . . . . .

. . . . . .

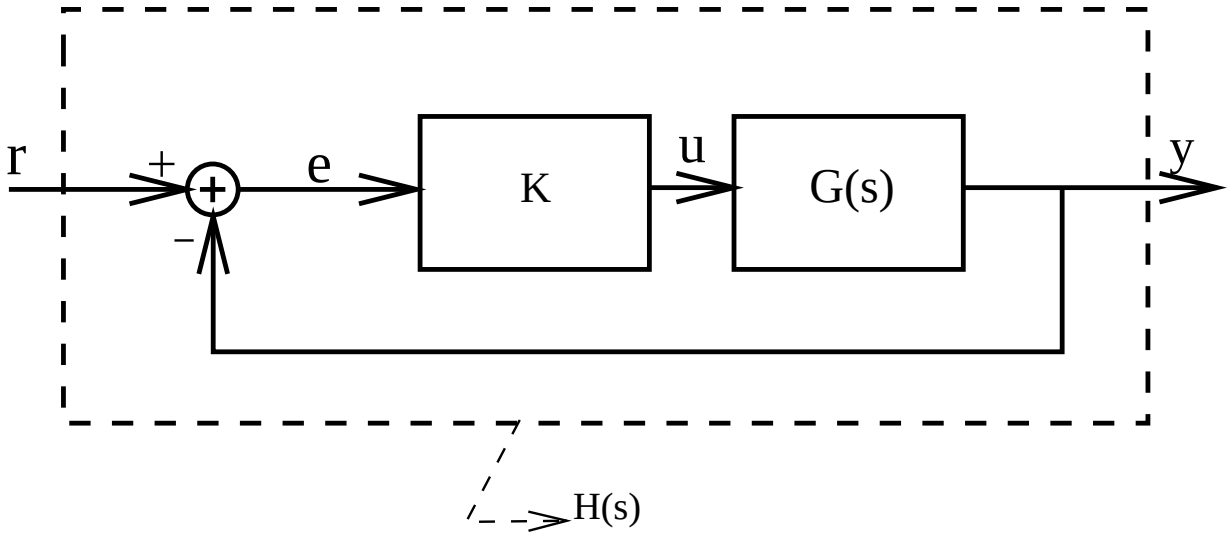


Figure 8.2:

8.2 Zitting 2

Oefening 1

Een systeem $G(s)$ heeft geen nullen en volgende polen:

-2
-2.5
$-13-3i$
$-13+3i$

- Ontwerp volgens de 'stability limit method' van Ziegler-Nichols drie regelaars (de versterking K_o die permanente oscillatie veroorzaakt ligt tussen 7000 en 8000):

  - P-regelaar
  - PI-regelaar
  - PID-regelaar.

  – $K_o =$ .....

  – $P_o =$ .....

  Vul de verschillende waarden voor de variabelen in:

Ziegler & Nichols	K_p	T_i	T_d
P-regelaar			
PI-regelaar			
PID-regelaar			

- Simuleer het geregelde systeem in simulink, waarbij een **stap**-ingang moet gevolgd worden.
- Vul in:

Sources
↪**Step Input**

Sinks
↪**yout**

Linear
↪**Sum**
↪**Transfer Fcn**
↪**Zero-Pole**
↪**Gain**

1. Vergelijking P – PI
   (a) Een belangrijke verbetering van het PI-geregelde t.o.v. het
   P-geregelde systeem is: .....
   ... .
   Meet het performantiekenmerk (T_r, T_s, os, us of e_{ss}) dat
   hiermee overeenstemt.
   - P-geregeld: .....
   - PI-geregeld: .....

   (b) Welke nadelen heeft de integrerende term ?
   ... .
   ... .
   ... .

2. Vergelijking PI – PID
   Welke invloed heeft de D-term op de inslingertijd ?
   ... .
   - $T_{s,PI} =$ .....
   - $T_{s,PID} =$ .....

   Verklaring: .....
   ... .

3. Waarom is de P-actie (K_p) bij de PID-regelaar krachtiger dan bij de P-regelaar ?
   ... .
   ... .

- Controleer of het geregelde systeem bestand is tegen stoorsignalen.
  Veronderstel dat er zich na 20 sec een stap-vormige verstoring (grootte = 50) voordoet, gesitueerd tussen regelaar en proces.

Sources
↪Step
Input

De maximale afwijking van de gewenste waarde is:

- PI-regeling: .....
- PID-regeling: .....

Bij het ongeregelde systeem is deze afwijking: .....

Oefening 2

- We benaderen het systeem uit oefening 1 door een 2de orde systeem. Daarvoor behouden we de dominante polen en passen de versterking aan (zodat de stapresponsie van het 2de orde systeem dezelfde y_{ss} heeft als die van het 4de orde systeem).
  - dominante polen: .....
  - nieuwe versterkingsfactor $k_n =$ .....

- Ga met behulp van Routh-Hurwitz na aan welke voorwaarden een PI-regelaar moet voldoen om samen met het 2de orde proces een stabiel gesloten-lus systeem te bekomen.

  Het proces: $P(s) = \frac{k_n}{a_2 s^2 + a_1 s + a_0}$

zp2tf

  - $a_2 =$ .....
  - $a_1 =$ .....
  - $a_0 =$ .....

  De regelaar: $C(s) = K_p(1 + \frac{1}{T_i s})$

  Gesloten lus: $$G(s) = \frac{C(s)P(s)}{1+C(s)P(s)} = \frac{k_n K_p s + k_n \frac{K_p}{T_i}}{a_2 s^3 + a_1 s^2 + (a_0 + k_n K_p)s + k_n \frac{K_p}{T_i}}$$

– Vul de Routh-Hurwitz-tabel in:

1				
2				
3				
4				
5				
6				

– De voorwaarden waaraan K_p en T_i moeten voldoen:

... .

... .

– Teken het gebied waarin goede K_p's en T_i's liggen.

– Controle:

· Simuleer het systeem voor een koppel (K_{p1}, T_{i1}) waarvoor (volgens Routh-Hurwitz) het systeem juist wel stabiel is.

$K_{p1} =$ .....

$T_{i1} =$ .....

· Simuleer het systeem voor een koppel (K_{p2}, T_{i2}) waarvoor het systeem juist onstabiel wordt.

$K_{p2} =$ .....

$T_{i2} =$ .....

- Ga na of de versterking K_p en de integratietijd T_i uit oefening 1 (PI-regelaar volgens Ziegler-Nichols) voldoen aan de voorwaarden voor het 2de orde systeem.

... .

Nonlinear ↪Transport Delay

- Voeg een dode tijd $\theta = 2$ sec toe aan het proces.

– Simuleer het systeem met dode tijd en de Ziegler-Nichols-parameters (referentie = stap).

– Wat merk je ? .....

... .

– Zoek een PID-regelaar die ervoor zorgt dat de stapresponsie van het gesloten-lus systeem maximaal 5% doorschot vertoont en een kleine stijgtijd heeft.

$K_p =$ .....

$T_i =$ .....

$T_d =$ .....

stijgtijd $T_r =$ .....

8.3 Zitting 3, 4 en 5: De electromagnetische YoYo

Het systeem waarop de regelopdracht moet uitgevoerd worden, is de elektromagnetische jojo (Fig. 8.3). Een spanning V aangelegd aan een spoel, wekt daarin een stroom I op, die op zijn beurt een magnetisch veld B in de spoel induceert. De veldlijnen lopen door een massa m van magnetisch materiaal, die zich in een mechanisch geleidende plastic buis bevindt.

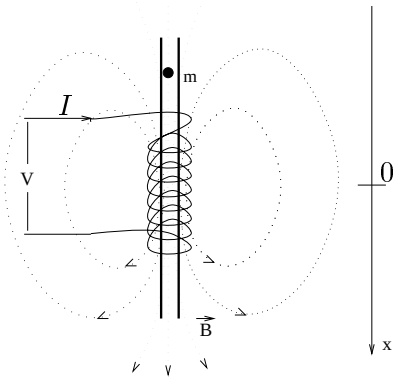


Figure 8.3: Elektromagnetisch systeem

Door de veldflux wordt een kracht $f_{e\to m}$ uitgeoefend op de massa, naast een gravitationele en wrijvende kracht (G en f_{fr}). Het vrijmaken van de massa wordt getoond in Fig. 8.4.
Een beweging van de massa induceert echter op zijn beurt een tegen-motorische spanning $e_{m\to e}$ in de spoel. Het elektrische gedeelte wordt schematisch weergegeven in Fig. 8.4. Bemerk dat naast de zelfinduktantie L, de spoel ook een weerstand R heeft.

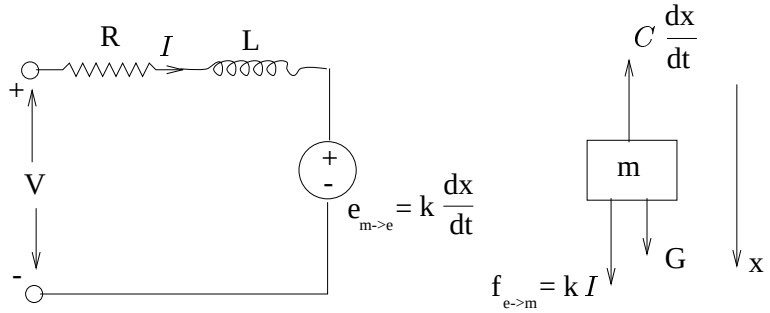


Figure 8.4: Schematische weergave van elektrisch en mechanisch gedeelte

De grootte van zowel de elektromagnetische kracht $f_{e\to m}$ als de tegen-motorische spanning $e_{m\to e}$, zijn evenredig met de grootte van de respectieve koppeling $k_{e\to m}$ en $k_{m\to e}$tussen elektrisch en mechanisch gedeelte. We beschouwen de koppeling symmetrisch te zijn tussen de twee gedeelten (zelfde koppel-waarde aan de beide zijden of $k_{e\to m} = k_{m\to e} = k$), maar nemen de waarde niet-constant. We weten dat de verstrooiïngsverschijnselen belangrijker worden naarmate de massa zich verder van het centrum van de spoel bevindt. Daarenboven is het aantal veldlijnen dat door de massa loopt ook afhankelijk van de aangelegde stroom. We modelleren deze beide effecten door voor de koppeling de volgende formule te nemen:

$$k \;=\; \alpha I^2 e^{-\frac{x^2}{r^2}} \;, \tag{8.1}$$

waarin α een empirisch bepaalde evenredigheidfactor is en τ een maat is voor de regio waarbinnen de verstrooiïngsverschijnselen klein zijn (cfr. spreiding in statistische zin). Deze koppelingskarakteristiek is weergegeven in Fig. 8.5.

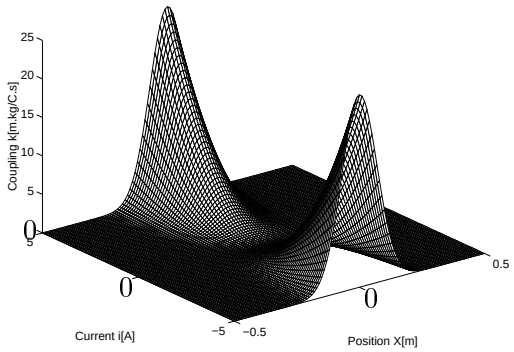


Figure 8.5: Koppeling k in functie van stroom I en afstand x.

De bedoeling van de opdracht is nu de positie x van de massa m te regelen door te spelen met de aangelegde spanning V.

8.3.1 Algebraïsche beschrijving

Niet-lineair toestandsruimtemodel

Het vrijmaken van de massa en het uitschrijven van de spanningsverdeling over de elektrische kring geeft:

$$\begin{cases} ma &= f_{e\to m} + f_{fr} + G \\ V &= e_R + e_L + e_{m\to e} \end{cases} \tag{8.2}$$

of

$$\begin{cases} m\frac{d^2x}{dt^2} &= kI - C\frac{dx}{dt} + mg \\ V &= RI + L\frac{dI}{dt} + k\frac{dx}{dt} \end{cases} \tag{8.3}$$

Door het invoeren van de toestanden $x_1 = x$, $x_2 = \frac{dx}{dt}$ en $x_3 = I$ en het invullen van de formule voor k, kunnen deze vergelijkingen geschreven worden als

$$\begin{cases} \frac{dx_1}{dt} &= F_1(x_1,x_2,x_3,V) &= x_2 \\ \frac{dx_2}{dt} &= F_2(x_1,x_2,x_3,V) &= \alpha x_2^2 e^{-\frac{x_1^2}{\tau^2}} x_3 - Cx_2 + mg \\ \frac{dx_3}{dt} &= F_3(x_1,x_2,x_3,V) &= -\frac{R}{L}x_3 - \frac{\alpha}{L}x_2^2 e^{-\frac{x_1^2}{\tau^2}} x_2 + \frac{V}{L} \end{cases} \tag{8.4}$$

We zien dat het systeem kan beschreven worden als een derde orde, niet-lineair toestandsruimtemodel, met toestandsvector $\overline{x} = [x_1; x_2; x_3]$.

Linearisatie

We herhalen even de belangrijkste stappen die moeten uitgevoerd worden om tot een gelineariseerd model te komen.

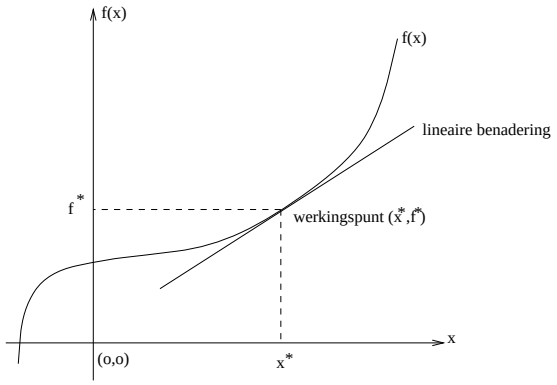


Figure 8.6: Linearisatie van een functie rond een werkingspunt.

1. **Bepalen van het werkingspunt**
   Als we spreken van een linearisatie, dan gaat het over een linearisatie rond een bepaald werkingspunt $\overline{x}^\star = [x^\star, y^\star, \dots]$

2. **Taylor-expansie rond evenwichtspunt berekenen**
   Rond het werkingspunt gaan we nu de niet-lineaire functie benaderen door haar Taylor-expansie, afgebroken na de eerste orde termen:

$$f(x, y, \dots) = f(x^\star, y^\star, \dots) + \left.\frac{\partial f}{\partial x}\right|_{(x^\star, y^\star, \dots)} (x - x^\star) + \left.\frac{\partial f}{\partial y}\right|_{(x^\star, y^\star, \dots)} (y - y^\star) + \dots + O^2$$

   Let erop dat de (partiële) afgeleiden natuurlijk geëvalueerd moeten worden in het werkingspunt $\overline{x}^\star$.

3. **Coördinaten transformeren naar werkingspunt** Via een coördinatentransformatie $\overline{x}^{'} = \overline{x} - \overline{x}^\star$ en door $f^{'}(\overline{x}) = f(\overline{x}) - f(\overline{x}^\star)$ te stellen, zetten we het systeem om in kleinsignaal (AC) vorm

$$f^{'}(x^{'}, y^{'}, \dots) = \left.\frac{\partial f}{\partial x}\right|_{(x^\star, y^\star, \dots)} x^{'} + \left.\frac{\partial f}{\partial y}\right|_{(x^\star, y^\star, \dots)} y^{'}) + \dots$$

8.3.2 Simulatie

Let op de verschillen tussen simulatie van het reële systeem of van het gelineariseerde model. Het ene werk in de absolute toestandsruimte, het andere is een kleinsignaal systeem. Bij het vergelijken moet dus rekening gehouden worden met de DC-offsets. Ga dit steeds zorgvuldig na. Controleer bvb. of de initiële waarden telkens goed gedefiniëerd zijn.
Afhankelijk van de gegeven parameters van de opdracht, moeten ook de simulatie-parameters (zoals maximale tijdstap of tolerantie) aangepast worden. Voor meer uitleg hierover: zie de Appendix in de begeleidende tekst.

Ga in de simulatie ook stapsgewijs tewerk. Probeer eerst in *open loop* consistente en repeteerbare resultaten te bekomen en schakel dan pas over op *closed loop* simulaties. Probeer ook telkens de resultaten fysisch te interpreteren. Het heeft bvb. bij een spoel van slechts $10cm$ groot geen zin uitwijkingen van tientallen meters te vragen. Bovendien heeft de linearisatie bij zo'n grote uitwijkingen toch geen betekenis meer ... .

8.3.3 Opdracht

Voor gegeven parameters (R, L, C, m, g, α, τ) en instelspanning v_s,

1. analyseer het gedrag van het reële systeem, probeer de betekenis van de parameter(waarden) te achterhalen,
2. bepaal de evenwichtspunten en lineariseer rond deze punten,
3. onderzoek in hoeverre deze linearisatie een goede benadering is,
4. ontwerp een PID-regelaar die rond **het stabiele evenwichtspunt** van het systeem een goede **tracking performantie** heeft op **stapvormige referentiesignalen**,
5. maak hierbij eventueel gebruik van verschillende methodes en vergelijk hun respectieve resultaten, zowel op reëel systeem als gelineariseerd model,
6. probeer zelf uit te maken wat de verschillende trade-offs zijn in het ontwerp. Onderzoek bv. **het werkingsgebied** van de regelaar op het reële systeem en probeer dit **zo groot mogelijk** te maken.

8.3.4 Verslag

Het verslag mag maximum 10 blz. lang zijn, figuren inbegrepen. Blokschema's (Simulink) en/of m-routines (Matlab) die essentieel zijn voor het begrijpen van het verslag, kunnen via E-Mail (*herman.verrelst@esat.kuleuven.ac.be*) ingeleverd worden.