large ISSN 0280–5316
ISRN LUTFD2/TFRT--5571--SE

Performance Improvement of PI-Control Using Nonlinear Feedforward

Bernd Trageser

Department of Automatic Control
Lund Institute of Technology December 1996

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
December 1996

Document Number
ISRN LUTFD2/TFRT--5571--SE

Author(s)
Bernd Trageser

Supervisors
Karl Johan Åström
Mikael Johansson

Sponsoring organisation
ITM

Title and subtitle
Performance Improvement of PI-Control Using Nonlinear Feedforward. (Prestandaförbättring av PI-reglering med hjälp av olinjär framkoppling).

Abstract

Processes whose dynamics include large time delays have poor control performance and slow responses to set-point changes. The aim of this thesis is to improve the set-point response by augmenting standard feedback control with a nonlinear feedforward. The feedforward approximates the input-output characteristics of the plant. It can be realized as an universal function approximator such as a neural network. In this way, the feedforward directly computes the control required to obtain the desired output in steady-state. A standard PI-controller is used in the feedback loop to force the control deviation to zero, taking into account the effects of model uncertainties, load disturbances and disturbances to the controlled variable. The same method can be used to include feedforward from measurable disturbances. Methods for automatic tuning of the PI-controller and the nonlinear functions are presented. The PI-controller can be tuned by standard automatic-tuning experiments, while the nonlinear feedforward compensation is learned through a series of closed-loop step-response experiments. This gives a simple and automated way of designing nonlinear feedforward compensators. The power of the proposed method is illustrated by a number of simulations.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
57

Recipient's notes

Security classification

Contents

1. Introduction

When controlling processes whose dynamics include large time constants or time delays standard feedback control is often not sufficient to achieve good performance. Since the feedback controller does not include information on the steady-state control of the plant the control signal is adjusted by integral action. This leads to characteristic slow and oscillatory responses to set-point changes.

This report deals with a control structure that reduces these difficulties, thus increasing the control performance. Extending feedback control by feedforward control gives an approach that maintains the major advantages of feedforward and feedback control, while compensating their essential disadvantages. We exploit the static input-output relation of the process to speed up conventional feedback control. The required information is obtained by a series of closed loop experiments. The (nonlinear) input-output characteristic of the process is implemented by feedforward control. The feedforward is used to come close to the desired value. Hence the feedback controller has only to do some fine adjustments. Since this structure relies on detailed process knowledge, a substantial performance improvement for control of processes with slow dynamics is achieved. Moreover, feedforward is used to compensate measurable load disturbances. The method is restricted to stable processes.

2. Control Structure

Figure 2.1 shows the control structure, which combines feedback and feedforward control, see Åström and Hägglund [1]. It consists of a standard feedback loop, a feedforward and a reference model.

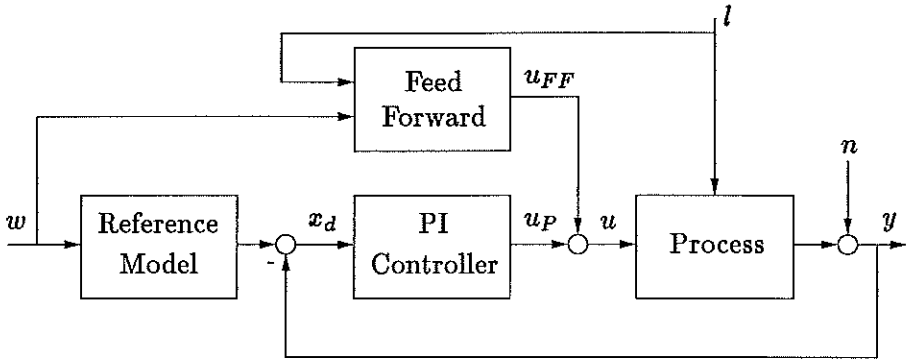

Figure 2.1 Proposed control structure

First, consider the standard control loop without the feedforward and the reference model. The feedback controller adjusts the control signal according to process disturbances and forces the output signal y to be equal to the reference w. There the proportional part of the controller takes care of the dynamics and its integral action adjusts the control signal to cancel the control deviation which may rise because of model uncertainties, load and/or output disturbances denoted by l and n in Figure 2.1.

The benefit of augmenting feedback control with the static feedforward points out with process disturbances by set-point changes. The feedforward supplies the process immediately with the steady-state control corresponding to the reference signal. Thus it takes care of the major part of the control signal for set-point changes. This results in a fast response to set-point changes. The PI-controller is then only used to adjust for model uncertainties and process disturbances. Thus the rapid response of feedforward is combined with the precision of feedback control. Another advantage of this approach is that measurable load disturbances can be compensated by feedforward before they influence the controlled variable. The extension of the feedforward with a disturbance compensation is indicated in Figure 2.1 by a second input to the feedforward.

The reference model in Figure 2.1 is used to make sure that the PI-controller does not counteract the feedforward signal. Recall that the feedforward supplies the steady-state control immediately. To reach this the control deviation has to be kept at a small value. Hence the reference model has to modify the reference to change similar to the process output.

The properties of the controller are illustrated by the following example.

EXAMPLE 2.1—DOUBLE TANK PROCESS
The nonlinear model of second order of a double tank system is given by

$$\begin{aligned} \dot{h}_1 &= \tfrac{1}{A_1}\left(Ku - a_1\sqrt{2gh_1}\right) \\ \dot{h}_2 &= \tfrac{1}{A_2}\left(a_1\sqrt{2gh_1} - a_2\sqrt{2gh_2}\right) \end{aligned} \tag{2.1}$$

Where h_1 and h_2 are the tank levels, A the cross-section area of a tank, a the cross section area of the outflow, g the acceleration of gravity and K the actuator gain of the control signal u. Additionally a dead time is considered to model a pipe-system that is connected to the out-flow of the second tank. The aim is to control the level of the second tank.

At equilibrium, all derivatives are zero. Solving the system of equations with respect to this constraint, we get the steady-state control as function of the reference level h_{2ref}.

$$u_0 = \frac{a_2}{K}\sqrt{2gh_{2ref}} \tag{2.2}$$

Figure 2.2 shows simulation results, where the proposed controller was compared with a standard feedback control. The solid line shows the response of the conventional control loop and the dashed curves the response of the combined feedforward/feedback control approach. The feedforward approximates the constraint given in equation (2.2). We see obvious that the performance is improved (Recall the dead time of $L = 10\ sec$). Notice that the response of the conventional system has overshoot. Moreover the load disturbance response is substantial improved where the first disturbance at $t = 400\ sec$ acts to the control signal and the second one at $t = 700\ sec$ to the first tank.

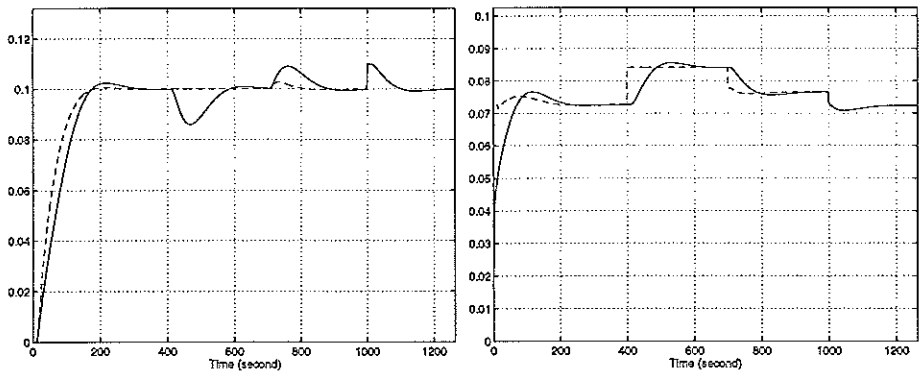

Figure 2.2 Simulation results that give a comparison between standard feedback control (solid line) and the combined feedforward/feedback control approach (dashed line). For it the output signals $y(t)$ (left) and the control signals $u(t)$ (right) are depicted

□

Finally we would like to emphasize the need to approximate the steady-state characteristics of the process. In general, a process is given by

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{aligned}$$

where f and h are nonlinear functions. To obtain the steady-state control signal u_{ss}, we consider the equilibrium state for a certain reference value and solve the equations,

$$\begin{aligned} 0 &= f(x_{ss}, u_{ss}) \\ w_{ref} = y_{ss} &= h(x_{ss}, u_{ss}) \end{aligned}$$

However a detailed process model is not always available. Thus the input-output characteristic cannot be implemented analytically. For that reason we need to find a different way to approximate the static input-output characteristic of the process. In general this relation can be an arbitrary nonlinear function. Thus we need an approach with universal approximation properties. In this thesis a neural network is used. Other approximation methods are given by lookup tables, splines or fuzzy logic. With this approach the feedforward can be trained by an on-line procedure. In this way it is also possible to accommodate changes in the process based on measured data.

3. Feedforward Control

This chapter presents the fundamentals of feedforward control. The combined feedforward/feedback control approach is explained and some feedforward applications are given.

3.1 Feedforward Control

Feedforward control is an useful complement to feedback. Some reasons for using feedforward are given in Shinskey [9].

- "The nature of feedback implies that there must be a measurable error to generate a restoring force, hence perfect control is unobtainable
- The feedback controller does not know what its output should be for any given set of conditions, and so it changes its output until measurement and set-point are in agreement. It solves the control problem by 'trial and error' , which is characteristic for the oscillatory response of a feedback loop
- Any feedback loop has a characteristic natural period. Should disturbances occur at intervals less than about three periods, it is evident that no steady-state will ever be reached"

Using a feedforward that takes into account all essential parameters it is at least theoretically possible to obtain perfect control. The reason for that is that feedforward acts immediately to cancel the disturbances before they affect the controlled variable. Feedforward is limited by the accuracy of the model and the measurements and the precision of the computations. A model error will e.q. result in an offset. This emphasizes that their quality is crucial for the design. Feedforward control is much faster than feedback control for set-point changes as well as for load disturbances.

Since the feedforward is capable to compensate disturbances caused by set-point changes or load disturbances one might ask why we do not omit the conventional PI-controller and use only the feedforward. This is outlined in the following section.

3.2 Combining Feedforward And Feedback

Though the feedforward control has several advantages compared to feedback control there are also some disadvantages. It concerns essentially about the assumed process knowledge to design a feedforward. In practical work there is not always a detailed information on the process available, at least with moderate effort. Moreover, there are often some parameters that are neglected. Thus, the feedforward is not able to make the controlled variable reach the desired value exactly. For some applications the achievable precision may of course be sufficient but in this thesis the aim is to reach an accurate control. Furthermore most of the industrially used control loops are based on PI-controllers. Thus we want to find an approach that is based on this conventional control

loop.

By combining feedback and feedforward it is possible to obtain a control structure which has most of the advantages of both methods and avoids most of their disadvantages. A block diagram of the proposed control structure is shown in Figure 3.1.

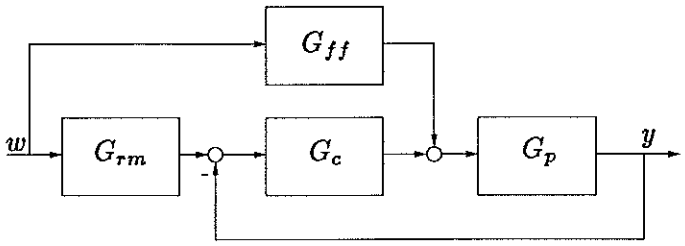

Figure 3.1 Proposed controller structure that combines feedforward and feedback

Consider a linear process, then the transfer functions of each block in Figure 2.1 can be described as given in Figure 3.1. Assume for simplicity $G_{rm} = 1$. Thus we can determine the transfer function of this system in the s-domain as

$$G(s) = \frac{Y(s)}{W(s)} = \frac{G_{ff}G_p + G_cG_p}{1 + G_cG_p}$$

Perfect control is achievable if $G(s) \equiv 1$. This is reached if the feedforward transfer function is equal to the inverse of the process or

$$G_{ff} = G_p^{-1}$$

In general this is not realizable. If we do only a steady-state consideration. Then G_p^{-1} reduces to the inverse of the process gain, and we get

$$G_{ff} = K_p^{-1}$$

To take care of the dynamics the reference model is used to reduce the controller activity during a set-point change. Thus we found another meaningful interpretation of the control approach. It confirms the basic idea to combine feedback with feedforward control.

The feedforward can also be used to compensate load disturbances and disturbances that act to inner states of the process. In the second case the feedforward signal has to pass the fast process dynamic before it can cancel the disturbance. Thus the controlled variable is already affected. It depends crucially on the process dynamics whether the feedforward compensation gives an improvement or not, because of the interaction by the feedback controller. If the interaction between feedback and feedforward leads to bad results then only the PI-controller should be used to compensate the disturbance.

3.3 Feedforward Applications

There are mainly three different types of applications where feedforward control can be used profitably.

- systems with slow dynamics where it is difficult to design high-bandwidth controllers with classical approaches
- compensation of measurable load disturbances
- systems where feedback is difficult or expensive to implement

The first two cases have already been discussed. An example of the third case is a simple temperature control for a car without external temperature sensor. There a valve is used to control the incoming air mass-flow. It keeps the incoming air mass-flow passing a heating inductance if the cabin temperature shall be increased or passing by the air mass-flow directly to the cabin if the temperature shall be decreased. The valve position is fixed for a certain temperature, regardless of disturbances like the outside temperature or an open window.

An additional application for standard feedforward is given in Ramirez [8]. He uses a feedforward for the determination of optimal set-points for supervisory computer control. For it the current measurements of the state, controls and disturbance variables as well as management directions and market conditions are fed into the feedforward which approximates a steady-state of the process.

4. Control System Design

In this chapter we give a design procedure for the proposed controller. First, a class of systems is proposed for which this approach can be used profitably. Afterwards the design of the controller which is based on an automatic tuning approach is given. Moreover, the design of the static feedforward is outlined. Finally some design methods for the reference model are given and the perturbation compensation is explained.

4.1 Appropriate Processes

The present controller structure is more appropriate for processes where conventional feedback control does not give a good response to disturbances from load or set-point changes, such as processes with slow dynamics. Nevertheless based on its easy and automated design this structure can be used for a large number of processes, even when no explicit model is available. It is also advantageous to apply the system to processes with measurable disturbances. In these cases load disturbances can be compensated very well. Figure 4.1 shows an appropriate process of first order with a nonlinear gain and a dead time. The load disturbance l_1 can be always compensated since it is canceled before affecting the controlled variable at all. It depends on the process dynamics whether there is also an improvement compensating l_2 and l_3. It may be possible to compensate l_2, since the control signal only has to pass the nonlinear gain before it can cancel the disturbance. Whether it is advantageous to compensate also l_3 by feedforward depends on the process variables T_1 and L. The most favorable situation is when $T_1 \ll L$.

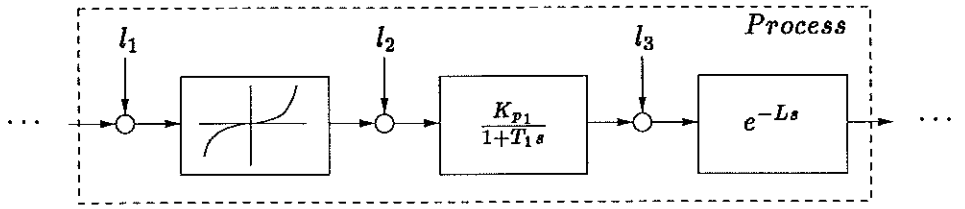

Figure 4.1 Example of an appropriate process with disturbances

Let the process transfer function in the s-domain be

$$G(s) = \frac{b_0 + b_1 s + \ldots + b_m s^m}{1 + a_1 s + \ldots + a_n s^n} e^{-Ls}$$

where we use that $n > m$. To obtain a measure at the significance of the process dynamics we introduce the normalized dead time τ. It can be computed as $\tau = L/(L+T)$. Where L denotes the dead time and T is the dominant time constant of the process. For small values of s we have approximated $T \approx a_1 - b_1/b_0$. The value of τ is between zero and one. The difficulty of control increases with τ.

4.2 Controller Design

If there is a full process model available classical methods can be applied to design the controller. Current methods are for instance pole placement or the Haalman approach (see Åström and Hägglund [1]). One can use the information on the process to tune the controller parameters as good as possible. In that case it is also possible to compute the feedforward analytically. However a full model is not always available (with moderate effort). Moreover it can be difficult to tune the controller parameters. Hence it would be advantageous to have an automatic tuning method. In this section an automatic tuning scheme for PI-controller is introduced. It is based on the frequency response method of Ziegler Nichols. Though this approach does not use any explicit process knowledge its power is quite considerable. In the following the basic idea of the automatic-tuning approach is outlined.

Relay Feedback

The relay feedback is based on the frequency response method of Ziegler-Nichols. Both methods compute the ultimate point on the Nyquist curve experimentally. This point is specified by the ultimate gain K_u and the ultimate period T_u. The Ziegler-Nichols approach increases the gain of a proportional controller until the process has reached its stability margin. Of course, the Nyquist curve of the process needs to have an intersection with the negative real axis. Hence the process has to be at least of third order or contain a dead time. Nevertheless the most real processes fulfill this constraint. If the process has reached the stability margin an oscillation with the ultimate period occurs. Moreover the amplitude of the oscillation is related to the ultimate gain. The handicap of this approach is that the amplitude of the process output can be very large.

The relay feedback avoids this problem using a relay instead of the proportional gain. Thus it is also possible to get oscillations of the process output but with a smaller amplitude since it depends on the amplitude of the relay. Moreover the oscillation state is reached very quickly. Its block diagram is shown in Figure 4.2. It is taken from Åström and Hägglund [1].

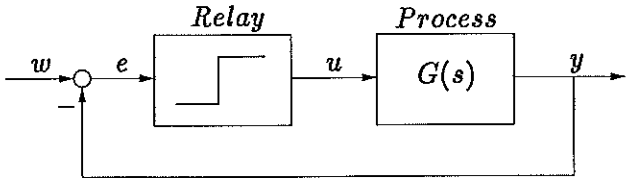

Figure 4.2 Block diagram of the relay feedback arrangement

The output and the control signal for the double tank system with dead time is shown in Figure 4.3. Since it concerns in here about a nonlinear process the ultimate parameters depend on the operating point. Hence the relay amplitude should be chosen according to this point. Nevertheless the obtained parameters may be suited also for other reference values. If the parameters give bad results then the ultimate parameters have to be determined for different operating points.

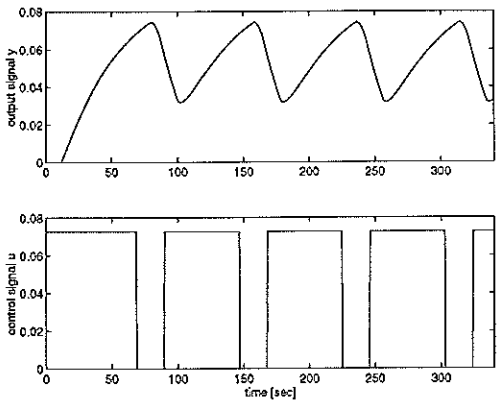

Figure 4.3 Output signal $y(t)$ and control signal $u(t)$ for a system under relay feedback

Considering only the first harmonic of the Fourier series, presumed the process has a sufficient low pass characteristic, we get a sinusoidal oscillation with the amplitude $4d/\pi$ for the control signal u. Where d is the amplitude of the relay. With the amplitude a of the output oscillation we get

$$G(w_u) = -\frac{\pi a}{4d}$$

Notice the phase shift of 180° between the output and the control signal (assume for simplicity w to be equal to zero). Exploiting further the constraint for an oscillation (gain in the loop is equal to one to maintain an oscillation which can be written as $K_u\ G(w_u) = -1$), we get for the ultimate gain

$$K_u = \frac{4d}{\pi a}$$

The ultimate period can be obtained immediately from the period of the output oscillation.

Automatic Tuning

To obtain a simple tuning rule one could apply the Ziegler-Nichols frequency response method which is based on T_u and K_u. However this rule has several handicaps such as shown in Åström and Hägglund [1].

- "Too oscillatory responses
- Different tuning rules are required for set-point response and for load disturbance response
- The rules give poor results for systems with normalized dead time close to one
- There is no tuning parameter"

The automatic tuning can cancel these drawbacks. It is based on an empirical method to find new tuning rules. For it a test batch of several representative stable processes was used. The controller parameters were determined using dominant pole design. The obtained values were normalized and plotted as a function of the gain ratio $\kappa = 1/K_u K_p$. Where K_p denotes the process gain, that can be easily determined from a steady-state consideration. Afterwards

the relations between the normalized controller parameters and the gain ratio were approximated. For it functions of the following form were used.

$$f(\kappa) \;=\; a_0\, e^{(a_1\kappa + a_2\kappa^2)} \tag{4.1}$$

Although this approach uses only three (instead of two) parameters to characterize the process it gives a substantial improvement. The method supplies good controller parameters even for processes with τ-values close to one.

Moreover the maximum sensitivity M_s is used as a tuning parameter. It is defined such as given in equation (A.1). Its value gives a measure of the robustness of the process (see Appendix A). There, a small value for M_s corresponds to a good robustness. Hence designing the tuning rules so as to minimize the maximum value of the sensitivity function in general lead to a closed loop system with better damping. Reasonable values for M_s are in the range of 1.3 to 2. Depending on this value one obtains different values for the design parameters a_0, a_1 and a_2.

With respect to the proposed structure we have chosen the value of M_s relative small to avoid overshoot because of an interaction between the feedforward and the feedback. In Table B.1 the design parameters for a maximum sensitivity of $M_s = 1.4$ are given. The controller parameters K_c, T_i and b can be computed using equation (4.1). Where b is the factor of the set-point weighting (see Appendix A). Usually this method is used to cope with the second handicap when using the Ziegler-Nichols tuning rule. Then the controller is tuned for load disturbances and the set-point weighting is used to obtain the desired set-point response.

The simple and automated controller design makes the automatic tuning to a powerful tool. The method is already used in many applications such as adaptive control.

4.3 Feedforward Design

The feedforward design consists essentially of the approximation of the input-output characteristic of the plant. Hence we need an universal approximator to get no general restrictions when approximating a function. There are several methods to approximate a static nonlinear function. They depend on the available information of the process. The methods are described in Chapter 5.

This thesis uses a neural network to approximate the input-output relation. In particular a Radial Basis Function Net is used. To approximate the function with the neural net one has to adjust the net parameters. Hence, measuring data of the process and a special training algorithm is necessary. While the data collection is described below the training procedure is outlined mainly in Appendix C.

Data Collection

A simple way to obtain training data is to measure the control signal for several values of the reference. One approach is to determine the input range

and divide it into as many parts as required to have a good approximation. Usually one needs at least 2-3 times as much measuring points as parameters. Thus we obtain an input vector that serves as input sequence for the closed loop system. The closed loop system already contains the controller which was obtained from the automatic tuning. The output vector is obtained by measuring each control signal for the corresponding input value when the process has settled. The control loop arrangement is shown in Figure 4.4. The reference and the control signals may obtained are shown in Figure 4.5.

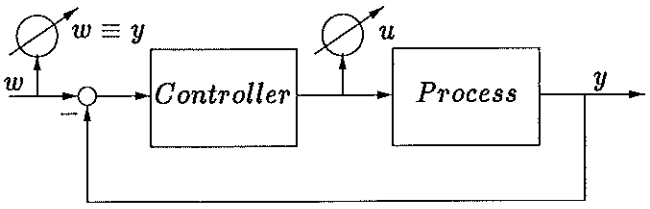

Figure 4.4 Measurement arrangement to collect training data sampling the reference signal

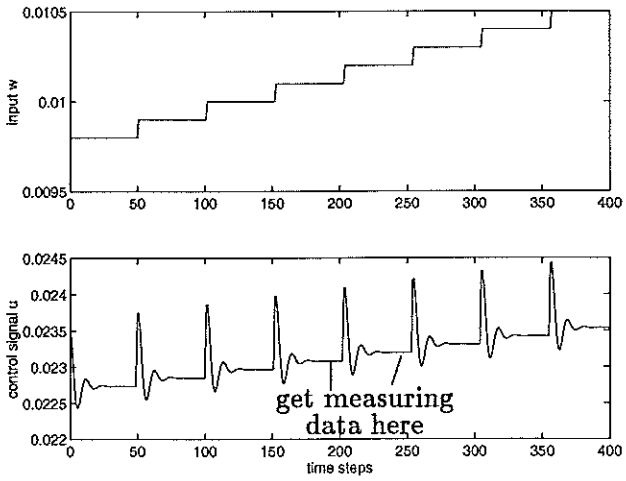

Figure 4.5 Input and output signal during the measurement phase

To obtain the steady-state characteristics, data has to be measured when the process has settled (see Figure 4.5). More than one measuring point at each input value should be taken if there is measure noise.

To improve the quality of the measuring data it is possible to choose the data points according to the gradient of the nonlinear function. This can be obtained by increasing the reference signal gradually until there is a significant change in the control signal. Then the set-point is kept constant until the process has settled. Afterwards the reference signal is increased and the procedure is repeated. The 'significant change' depends on the process. In this context it is meaningful to get the range of the control signal and to choose the value of this variable again corresponding to the number of measuring data one needs to have. Thus one gets a nonlinear distribution of the measured data corresponding to the gradient of the function, which is to be approximated. The control signal range can be determined measuring its value for the margins of the reference signal. This is a very meaningful approach since the neural

network needs more information in the areas with large gradients than in the regions with smaller gradients. The proposed approach uses also the measurement arrangement shown in Figure 4.4. The method corresponds to a closed loop sampling of the control signal. Figure 4.6 shows how the measured data looks like for the double tank system with constant step size in the control signal u. Recall that the input-output relation gain looks like equation (2.2).

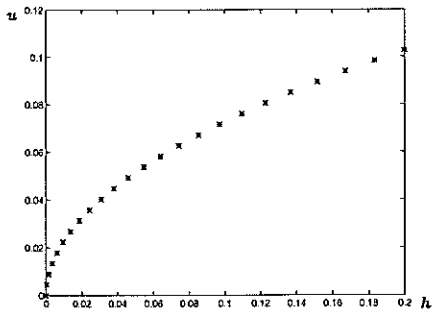

Figure 4.6 Measured data for the double tank system with constant step-size in the control signal $u(h)$. The data was obtained by the method sampling the control signal in closed loop

The input-output characteristic of the process can also be determined in a series of open loop experiments. The measurement arrangement is shown in Figure 4.7. The measuring data is obtained by applying an appropriate input sequence to the process. The sequence consists of a set of steps with increasing amplitude. The corresponding output data can be measured when the steady-state is reached. Note that the output signal is equal to the reference when the process has settled. Based on the series of set-point responses it is possible to determine the process time constant for different operating points (Recall the assumed nonlinear process). This can be meaningful to adapt the reference model for different reference values.

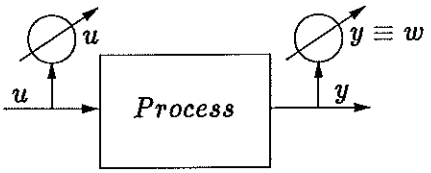

Figure 4.7 Measurement arrangement to collect training data sampling the control signal in open loop

The training of the neural network is explained in Appendix C. Nevertheless it shall be said in here that it is quite easy to adjust the feedforward to process changes or unmeasurable (load) disturbances acting to the control signal or the faster part of the process. For it the on-line learning approach as shown in Figure C.4 has to be used. With it the parameters can be re-trained using new data.

4.4 Reference Model

The reference model is used to decrease the activity of the feedback-controller during set-point changes. Since the feedforward supplies the process immediately with the correct steady-state control signal, the feedback controller

has only to do minor adjustments. Hence it is necessary to keep the control deviation at a small value after a set-point change. This is done by filtering the reference signal with a model. The modified reference rises then similar to the process output. Of course the control deviation is zero when an exact model is used. The quality of the model is crucial for the performance of the proposed control approach since otherwise the controller counteracts the 'optimal' control signal of the feedforward. Note also that the controller action due to disturbances is still the same.

A process model is required to design the reference model. We will use the following simple model, where the gain K_m has to be set equal to one to avoid set-point modification.

$$G_m(s) = \frac{K_m}{1+T_m s} e^{-L_m s} \tag{4.2}$$

Three Parameter Process

If the process transfer function is exactly of the assumed structure given in equation (4.2) we can determine T_m analytically. For it the relation between the Nyquist curve of the process and the ultimate point has to be evaluated. There one needs to calculate the (first) intersection of $K_u G_m(j\omega)$ with the negative real axis. Since K_u denotes the ultimate gain the intercept is equal to the critical point. We obtain the following equation

$$\omega T + tan(\omega L) \equiv 0$$

where $\omega = 2\pi/T_u$. With T_u obtained from the relay feedback method, one can calculate the model time constant T_m approximately to

$$T_m \approx -\frac{tan(\frac{2\pi L}{T_u})}{2\pi} T_u \tag{4.3}$$

The dead time can be determined during the phase when collecting data for the function approximation.

Process of Higher Order

If the process structure deviates from the assumed one in equation (4.2) one can use common step-response methods to get approximate model parameters. Figure 4.8 shows a step response from a system of second order, including a dead time. The dead time can be determined from the distance 0-A. Where A is the intercept of the tangent to the step-response that has the largest slope with the horizontal axes. The time constant can be obtained in two different ways. The distance A-C gives a crude estimation of T that can be improved using the distance A-B. Where C is the time instant when the tangent intersects the final value and B is the time instant when the process output has reached 63% of its final value.

4.5 Load Disturbance Compensation

The response corresponding to load disturbances is an important part. The feedforward can be used to compensate load disturbances. It depends on the

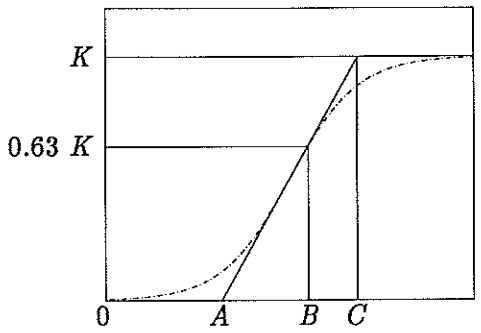

Figure 4.8 Step-response from a system of second order with a dead time

process dynamics whether it is also possible to compensate disturbances acting to inner states of the process. Disturbance compensation is illustrated by the double tank example.

EXAMPLE 4.1—DOUBLE TANK DISTURBANCE COMPENSATION
Consider a load disturbance l_1 acting as shown in Figure 4.9.

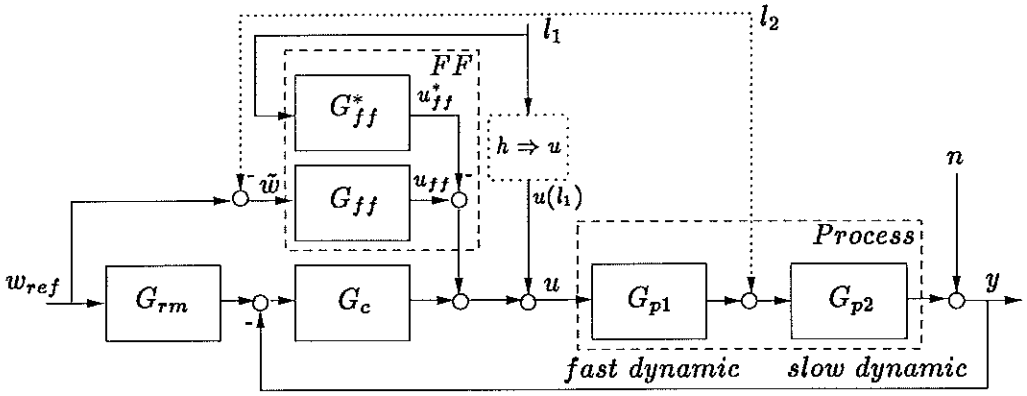

Figure 4.9 Disturbed control loop with feedforward disturbance compensation. There are different types of load disturbances considered acting also to inner states of the process.

Usually there is only indirect information on the load disturbance available. Consider for instance a third tank, whose level tends to exceed a certain limit in the emergency case. To avoid this a valve is opened and a mass-flow is led into the first tank. The available information is the measurable height of the liquid in the medium tank. Also this tank acts in a nonlinear way to the process (represented by the nonlinear converter in Figure 4.9).

Let u^*_{ff} denote the control signal of the disturbance compensation and $u(l_1)$ is the control signal generated by the disturbance. To compensate the measurable load disturbance l_1 the signal u^*_{ff} has to cancel $u(l_1)$. Hence G^*_{ff} has to approximate the relation between l_1 and $u(l_1)$. The steady-state relation between these two quantities is given by

$$u(l_1) = \frac{a_3}{K}\sqrt{2gl_1}$$

Assuming the same out-flow diameter for all tanks we see that G^*_{ff} equals the input-output relation of the process. Thus we can use the previous results

of approximating the input-output relation to implement the feedforward G_{ff}^* for load disturbance compensation. This gives a very simple and powerful compensation of load disturbances.

For compensation of the load disturbance l_2 denoted by a dashed line in Figure 4.9 we get an even simpler method. For instance assume l_2 to be an mass-flow of the medium tank into the second tank which results in a different liquid height in the second tank. Thus the load disturbance l_2 can be easily compensated modifying the reference input of G_{ff}. The new, modified input looks now like

$$\tilde{w} \;=\; w_{ref} \;-\; l_2$$

However, the compensating control signal has to pass the fast process dynamic ($T_1 \approx 9.275$) before canceling the disturbance. So that the controlled variable is already affected.

For further investigations we also introduce the load disturbance l_3 acting in between the second tank and the pipe-system. An investigation of the output error for the load disturbances l_1, l_2 and l_3 has shown that it can be forced to be equal to zero using only the feedforward compensation. For it, the fast process dynamic has to contain the process gain. So that all slower parts have a process gain equal to one. Otherwise one gets a deviation for the disturbances l_2 and l_3, which has to be compensated by the controller. Nevertheless, since this is only a steady-state consideration. It depends mainly on the process dynamics, in particular on the time constants of the process whether the achievable results give an improvement or not. We will assess the improvement by the feedforward later on by evaluation of several performance indices. □

Generally - for instance if the out-flow diameters have different values - the feedforward G_{ff}^* is different from G_{ff}. Thus one has to train also G_{ff}^* using the methods described before when training G_{ff}. For that reason one has to collect suitable measuring data. Assume that there is no feedforward compensation yet. In that case the PI-controller takes care of the load disturbance. Thus it is easy to obtain measuring data out of its steady-state control signal and the measured information on the disturbance. However, it takes some time to get sufficient measuring data to approximate the nonlinear constraint over the whole range. Usually the feedforward compensation cannot be used during that time. Hence it is meaningful to extend the on-line training with a strategy that allows us to use the feedforward compensation even if there is only few measured data. The strategy exploits the already available data to obtain an estimate for the desired output. To make the idea clear we consider Figure 4.10.

In the upper part we see a set of load disturbances. The load disturbance amplitude is normalized with the one of the reference signal. In the lower part of this figure the function to be approximated is depicted with a dashed line. Assume the first two values of the load disturbance to be already evaluated. The obtained information is given by '*'. If we consider now a new change in the load disturbance we can use the already available information. There one obtains an estimate of the compensation signal for instance by using a line between the two measured points. The obtained information is given by a '+'. If the load disturbance is compensated by the controller this estimate

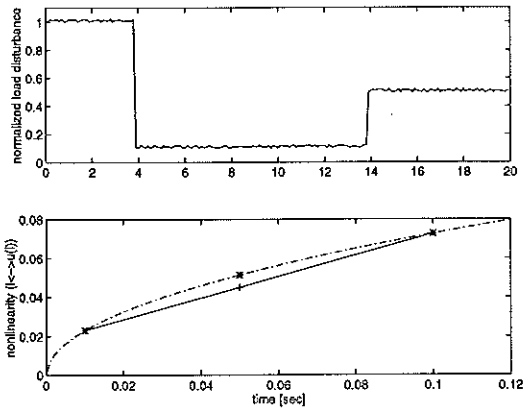

Figure 4.10 Set of load disturbances (normalized with reference value) and nonlinearity to be approximated

can be adapted to the correct value. Thus we can use the feedforward compensation much earlier. Hence the control loop responds much better since the major part of the load disturbance can be compensated before affecting the controlled variable.

This approach is also transferable to set-point changes, exceeding the range where the nonlinearity was trained. Furthermore regularization can be used to increase the robustness of identification algorithms. This approach is introduced in Appendix C.3.

5. Function Approximation

In this chapter, different methods for approximating static nonlinearities are described. Most approaches are based on the idea to approximate a desired function with a superposition of other (radial) basis functions. For the superposition the basis functions are weighted and summarized. The proposed methods distinguish essentially by the different choice of the basis functions. Starting with a lookup table that uses rectangular functions we can increase the precision using more advanced curves. Such as triangular or continuous (Gaussian) functions that are used by splines and neural networks, respectively.

5.1 Lookup Table

The measured information on the function to be approximated can be used for a lookup table wherein the input and the output data is stored. This data defines the so called base points of the function. For input values between these base points the output value is approximated by a constant. Hence this method can be considered as a superposition of rectangular functions at each value of the input data. The height of each rectangular corresponds to the value of the output data (see Figure 5.1, where '*' denotes a data pair). It is also possible to approximate the desired function between the base points with a line or a more complex curve. To achieve a sufficient accuracy the lookup table may need to store a large number of data. This is the main disadvantage of this approach.

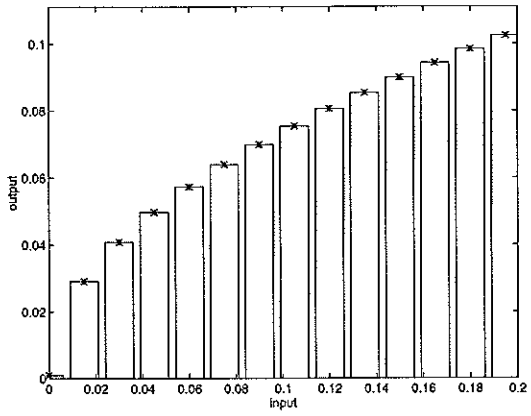

Figure 5.1 Function approximation using a lookup table

5.2 Splines

To improve the approximation between the measured data one can use splines. This approach uses a line to approximate the function between the measured data. Each measured point is represented by a function. Assume for instance a triangular that has a height equal to the output value of this data pair

and a width corresponding to the distance between two base points. If all those functions are superpositioned one gets finally an approximation of the measured function. This can be written mathematically as

$$\hat{f}(x, \underline{w}) = \sum_{i=1}^{N} w_i g_i(x) \tag{5.1}$$

where $\hat{f}$ denotes the approximation of the nonlinear function and w_i are the weights of each basis function g_i. Also in here some problems may rise for large numbers of data.

5.3 Fuzzy Logic

The basic idea to approximate a function using fuzzy logic is different from the previous one. Nevertheless the methods are similar in the end since a fuzzy logic system also uses basis functions that are merely superpositioned in a different way.

In general a fuzzy logic system uses rules to set its variables. Those rules can be obtained either from collecting process data or an experienced operator. There are several ways described in Wang [10] to design such a system based on the available process information.

Basically fuzzy logic tries to imitate the 'human decision making'. In binary logic the value of a variable can only be equal to zero or one. Hence we can define a quantity only by expressions like *v belongs to a set V* or *v belongs not to a set V*. To describe the same quantity with human expressions we need to extend the binary logic so that a variable can also have every intermediate value between zero and one. With this fuzzy approach we get for instance *v belongs a little to a set V*. To specify *a little*, so called membership functions are introduced. These functions are similar to the basis functions. Their value is restricted to a range from zero to one. Normally triangular and trapezoidal functions are used. But also more smooth functions such as Gaussian functions can be applied. Let us consider a common example.

EXAMPLE 5.1—DESCRIBING A TEMPERATURE WITH FUZZY LOGIC
In Figure 5.2 one way to describe a temperature with fuzzy variables is given. Each membership function has a value between zero and one corresponding to a certain temperature. The value of a membership function denotes the membership of the temperature. For instance the membership of the temperature of 18°C to the function 'warm' is equal to $\mu_{warm}(18) = 0.8$. The membership of this temperature to the function 'cold' is equal to $\mu_{cold}(18) = 0.2$. Thus one can evaluate the fuzzy rules which look for instance such as the following one.

$$IF\ a\ is\ 'warm'\ \ AND\ \ b\ is\ 'cold'\ \ THEN\ \ 'increase'\ c \tag{5.2}$$

Where a and b denote input variables. For instance a is equal to the room temperature and b stands for the outside temperature. *'warm'*, *'cold'* and *'increase'* are membership functions and c is the output variable. Assume c to be the heating power. □

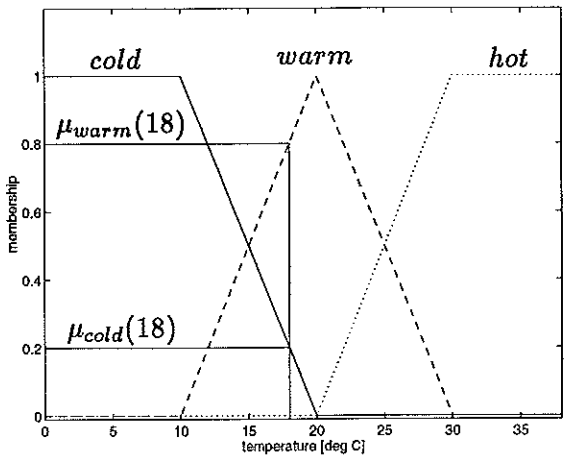

Figure 5.2 Membership functions and membership

Depending on the arithmetical operator that is used to calculate the AND in equation (5.2) one gets different results for an input vector. In equation (5.3) the most common choice in fuzzy logic is given. The operators are derived from the conventional binary logic.

$$\begin{aligned} and &: \mu_{a\ and\ b}(x) = min(\mu_a(x), \mu_b(x)) \\ or &: \mu_{a\ or\ b}(x) = max(\mu_a(x), \mu_b(x)) \\ not &: \mu_{not\ a}(x) = 1 - \mu_a(x) \end{aligned} \tag{5.3}$$

To obtain a system with 'human decision making' one has to combine several rules which uncover the whole output range. The rules that refer to the same output variable are also combined with an arithmetical operator such as given in equation (5.3). The rule in equation (5.2) can be explained as follows. The operator AND gives the condition that fits the best. This value determines how much the rule 'fires', which means how much this rule is active. Afterwards all rules are superpositioned. For it a superposition is made with the areas of all corresponding output membership functions that are cut off at the calculated value. To make the proceeding clear we consider again the temperature example.

EXAMPLE 5.2—DESCRIBING A TEMPERATURE WITH FUZZY LOGIC
Using equation (5.3) to determine the AND we get $\mu_c = \mu_{warm\ AND\ cold}(18) = min(\mu_{warm}(18), \mu_{cold}(18)) = min(0.8, 0.2) = 0.2$. This value denotes the activity of the output variable c. Thus the area of the membership function 'increase' is the one below the limit of 0.2. □

Since the output range is also described with membership functions a transformation is required to convert the fuzzy value into a crisp one. There are several ways for it. The most common method is to calculate the center of gravity of all active output functions.

A fuzzy logic system consists essentially of three parts. First, the fuzzification is used to convert all input variables into fuzzy variables. Afterwards the inference evaluates the rule base and finally the defuzzification supplies a crisp value for the output variable.

5.4 Neural Networks

A further method to approximate any static nonlinearity is to use neural networks. There are mainly two structures that can be used. Namely the Multi Layer Perceptron (MLP) and the Radial Basis Function (RBF) network. However in this thesis only the RBF-net is investigated since linear optimization methods can be applied to adjust its parameters. This is one main advantage compared to the MLP where nonlinear optimization methods have to be applied. Nonlinear optimization leads to an increased calculation effort which is not appropriate for an on-line application. Nevertheless a short description of the MLP is given in Appendix C.1 when deriving a training algorithm to adjust the parameters. Moreover the training algorithm will be used to assess the results of a linear optimization.

Also the RBF-net uses a superposition of several basis functions to approximate a desired function. For it, the initial parameters of the basis functions have to be chosen a priori. Afterwards the weights are optimized using linear optimization methods. The structure of the RBF-net is shown in Figure 5.3. The net has P inputs, M basis functions and one output. c_{pm} is the center and σ_{pm} the variance of the m-th basis function of the p-th input. w_m is the weight of the $m-th$ basis function.

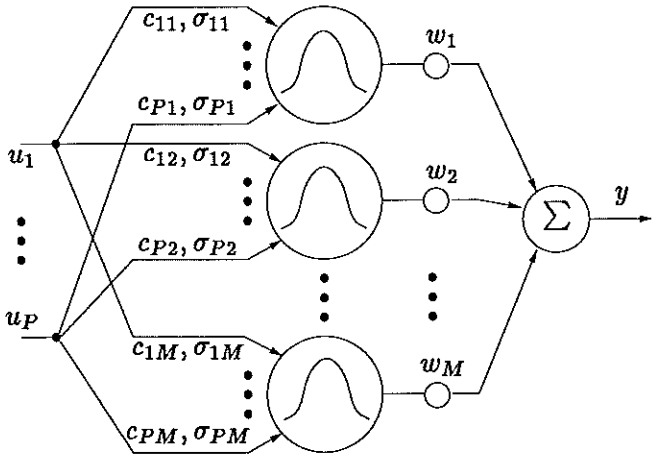

Figure 5.3 RBF-network

The transfer function of a RBF-net with P inputs and one output is given in equation (5.4). One can see the relation to splines comparing this equation to equation (5.1). With this neural network one can achieve any desired accuracy since the basis functions are smooth. Of course this depends on the number of basis functions.

$$y(\underline{u}, \underline{w}) = \sum_{m=1}^{M} w_m \, g_m(\underline{u}) \tag{5.4}$$

If a Gaussian function is assumed to be the basis function - this is the most

common choice - then equation (5.4) can be written as

$$y(\underline{u},\underline{w}) = \sum_{m=1}^{M} exp\left(-\frac{1}{2}\left(\frac{u_1-c_{1,m}}{\sigma_{1,m}}\right)^2 - \frac{1}{2}\left(\frac{u_2-c_{2,m}}{\sigma_{2,m}}\right)^2 - \ldots \right.$$
$$\left. \ldots - \frac{1}{2}\left(\frac{u_P-c_{P,m}}{\sigma_{P,m}}\right)^2\right)$$

To approximate the desired function the weights have to be optimized. For it, in general a loss function is minimized. The most common choice is to use a quadratic one since this leads to a linear optimization problem. The equation of the loss function is given below.

$$J = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5.5}$$

Where y_i denotes the measured output data and $\hat{y}_i$ stands for the output of the neural network for the $i-th$ of N measured data points. One obtains the minimum applying for instance the LS-algorithm. Its equation is given below. Where $\underline{\Theta}$ (mx1) contains the estimated weights, $\underline{\Psi}$ (Nxm) is the data matrix and $\underline{y}$ (Nx1) contains the output data.

$$\underline{\hat{\Theta}} = [\underline{\Psi}^T\ \underline{\Psi}]^{-1}\ \underline{\Psi}^T\ \underline{y} \tag{5.6}$$

To obtain good results from this equation we require sufficient information in the measured data, so that the matrix $[\underline{\Psi}^T\ \underline{\Psi}]$ is invertible. In general the process input has to fulfill some conditions to ensure a sufficient excitation of the process. This may cause problems, since the process to be identified is often (completely) unknown. In here we avoid this problem by the introduced methods to collect training data. When there is enough data available, application of equation (5.6) supplies the weights within one step. Because of its simplicity the LS-algorithm gives a meaningful approach to adjust the parameters of the function approximator (for the first time).

Since we have to measure sufficient data before we can apply equation (5.6) the LS-algorithm is not appropriate to adapt the feedforward on-line, for instance when there is only few new data available. In that case the Recursive Least Squares (RLS) algorithm such as given in equation (5.7) can be applied. This approach avoids the matrix inversion by a mathematical transformation. The parameters are determined recursively. This leads to an iterative solution.

$$\underline{\hat{\theta}}(k) = \underline{\hat{\theta}}(k-1) + \underline{P}(k)\ \underline{\Psi}(k)\ e(k) \tag{5.7}$$

$\underline{P}$ and e have to be determined in every step by the following equations

$$\begin{aligned} \underline{P}(k) &= \frac{1}{\lambda(k)}\left(\underline{P}(k-1) - \frac{\underline{P}(k-1)\ \underline{\Psi}(k)\ \underline{\Psi}^T(k)\underline{P}(k-1)}{\lambda(k)+\underline{\Psi}^T(k)\ \underline{P}(k-1)\ \underline{\Psi}(k)}\right) \\ e(k) &= y(k) - \underline{\Psi}^T(k)\ \underline{\hat{\theta}}(k-1) \end{aligned} \tag{5.8}$$

There $\lambda(k) = [0,\ldots,1]$ denotes a forgetting factor that is used to achieve faster convergence. Usually its value is set equal to $\lambda(k) = 0.8,\ldots,0.98$. Thus also past values are taken into account. However with this approach the effects

of disturbances is amplified.

There are also some nonlinear optimization methods trying to find the global minimum. If the surface of the loss function has a complex structure with many minima. Although there are several approaches in literature to find the global minimum they are not further investigated since they require a large amount of calculation effort. Hence they are not appropriate for an on-line implementation. In general a nonlinear approach is always meaningful when the measuring data already indicates a special nonlinear course.

It is also possible to combine linear and nonlinear optimization. For instance one can get some initial values for the weights using the LS. Afterwards a nonlinear optimization method can be applied to adjust also the centers and variances. To investigate nonlinear optimization methods the well known Widow-Hoff training algorithm is derived in Appendix C.1. Moreover the equations to adjust the centers and variances of the basis functions and finally a training schedule is given.

Initial Data

The initial values for the centers and the variances can be chosen essentially by three methods. The first one is to put the basis functions on a lattice uncovering the input space. The variances can be easily chosen according to the distance between the functions. This method is not applicable for large dimensions of the input vector since this causes a combinatorial explosion of the number of parameters to be optimized.

The second method builds clusters with the input data and place the basis functions on each center of the clusters. If there is only input data in a very limited range then this approach neglects few but important values at the margins. Moreover it is hard to find a good algorithm for choosing the variances. Normally the distance to the next basis function is used.

The third method to choose the initial data puts the centers of the basis functions on randomly chosen input data. This approach is meaningful if there is few input data. Otherwise the uncoverage of the input space may be bad. Also the training data has to be chosen appropriate. Otherwise this approach gives bad results. Consider for instance a periodic signal where the period is a multiple of the sampling time. Again it is difficult to find values for the variances. A starting point is also to choose them corresponding to the distance between each basis function.

Number Of Parameters

The number M of free parameters is essentially responsible for the quality that is within reach. Moreover it determines the required time for the optimization. With N equal to the number of measured data there are basically two cases distinguished:

- $M = N$ - Interpolation:
  The curve goes across all measured data. The loss function is zero.
- $M < N$ - Approximation:
  The loss function is minimized by the curve but reaches never zero be-

cause of too few degrees of freedom.

Using an interpolation is only meaningful if the measured data is exactly correct. As soon as the measured data contains a disturbance - this is always the case in practical work - an interpolation models also the noise. For that reason one get worse results than using the approximation which automatically takes an average mean. The degree of this average mean depends on the number of free parameters. The difference between interpolation and approximation is depicted in Figure 5.4. Here a noisy sinusoidal function is interpolated once with 20 radial basis functions (dashed dotted line) and once approximated with only 2 radial basis functions (solid line).

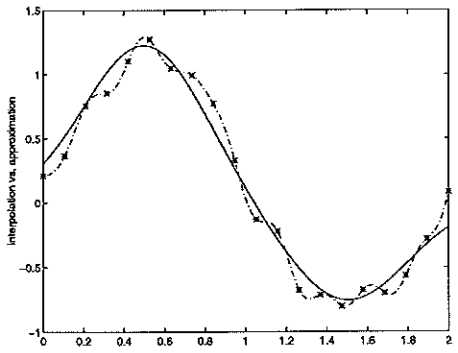

Figure 5.4 Interpolation and approximation of a noisy sinusoidal function with 20 and 2 RBF's

This problem can be described also by the well known bias variance tradeoff. The total squared error is equal to the sum of the squared bias plus the variance. Hence reducing the bias will always lead to an increased variance and the other way round. Thus an interpolation gives a large variance without bias and an approximation gives a small variance with larger bias.

We can see from Figure 5.4 that there is always a bias if one uses too few parameters (approximation). Because the function cannot be approximated exactly. If one uses too much parameters on the other hand this leads to an approximation of the noise. Out of this the question results: what is the optimal number of parameters to find a good compromise minimizing the bias and the variance. This is no trivial problem with a general solution. Hence, the number of parameters has to be chosen by a 'trial and error' proceeding.

6. Implementation

In this chapter the implementation of the proposed control structure for a typical process with slow dynamics is described. First, the process model is given. Afterwards the controller and the feedforward design is explained. Moreover the reference model is designed and the simulation results are shown.

6.1 Double Tank Process

The chosen process is a double tank system. A sketch of the plant is shown in Figure 6.1. The variable to be controlled is the level h_2 of the big tank.

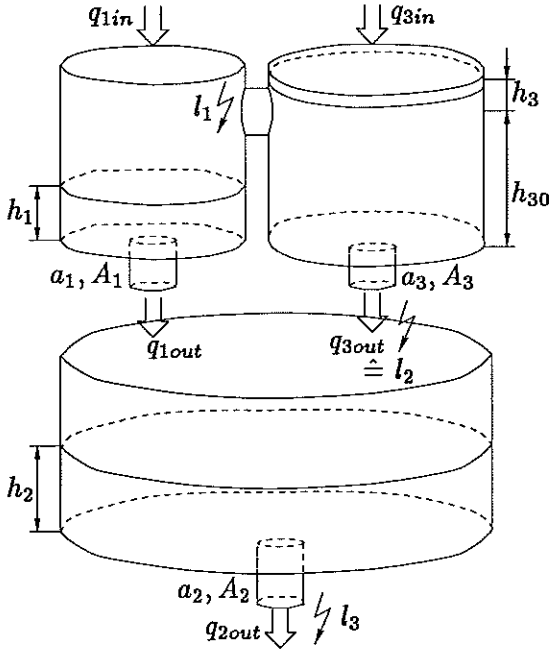

Figure 6.1 Double tank system

The present process consists of a small and a big tank. A medium tank is connected to the small one. In the emergency case the connection to the small tank is opened and the out-flowing liquid acts as load disturbance l_1. If h_3 (distance between center of out-flow and water height) still increases then also an out-flow at the bottom of the medium tank is opened. Thus the load disturbance l_2 acts to the big tank. Also we consider a load disturbance l_3 acting to the outflow of the big tank. Moreover a dead time is introduced to model a pipe system that is connected to the big tank.

The model of this nonlinear process of second order (without the medium tank and the pipe system) can be easily derived as shown in Appendix D. The obtained model equation is given below.

$$\begin{aligned} A_1 \tfrac{dh_1}{dt} &= Ku - a_1\sqrt{2gh_1} \\ A_2 \tfrac{dh_2}{dt} &= a_1\sqrt{2gh_1} - a_2\sqrt{2gh_2} \end{aligned} \tag{6.1}$$

For the steady-state the following equation is valid.

$$\begin{aligned} u_0 &= \frac{a_1}{K}\sqrt{2gh_{1ref}} \\ h_{1ref} &= (\frac{a_2}{a_1})^2 h_{2ref} \end{aligned}$$

Thus the static input-output characteristic of the process looks like

$$u_0 = \frac{a_2}{K}\sqrt{2gh_{2ref}} \tag{6.2}$$

The nonlinear relation between the level of the medium tank and its outflow $q_{3_{out}}$ is given in equation (6.3). It is clear to see the relation to equation (6.2).

$$q_{3_{out}} = a_3\sqrt{2g(h_3 + h_{30})} \tag{6.3}$$

So far we have got only a simple process of second order. To obtain a process with significant dynamics we consider also a pipe system that is connected to the big tank. We can model this part by a time delay of $L = 10\ sec$. Moreover this fulfills the constraint of an intersection of the Nyquist curve with the negative real axis. Hence we can apply the automatic tuning to design the controller.

To be able to assess the significance of the process dynamic we consider a linear model of this process. Hence we linearize the model equation (6.1) around the operating point of $h_{op} = 0.1\ m$, and obtain the following linear process model of second order

$$G_1(s) = \frac{K_p\ T_2}{(1 + T_1 s)(1 + T_2 s)} e^{-Ls} \tag{6.4}$$

Where $K_p = 0.05$, $T_1 = 9.275\ sec$, $T_2 = 55.65\ sec$ and $L = 10\ sec$. The dominant time constant of this process can be determined such as given in section 4.1. We obtain $T = T_1 + T_2 = 64.925$. Thus we can determine the normalized dead time as $\tau = 0.133$. Although this value is close to zero it is not possible to design a satisfactory controller using classical approaches.

To show the power of the combination of feedforward and feedback control especially for processes with more significant dynamic ($\tau \to 1$), we consider a second process. This linear process of first order contains a larger time delay ($L = 5T$), so that the normalized dead time becomes $\tau_2 = 0.833$. Obviously this process is even more difficult to control. The process transfer function is given below.

$$G_2(s) = \frac{1}{1 + s}\ e^{-5s} \tag{6.5}$$

6.2 Controller Design

The controller parameters for both processes were determined using the automatic tuning approach. During the relay feedback of the first process the relay

amplitude was set equal to $d = 0.0726$. This corresponds to an operating point of $h_{op} = 0.1\ m$. The reference value was set equal to $h_{ref} = 0.64 h_{op}$. Moreover a hysteresis of $hy = -0.01$ was used. The obtained controller parameters are given in Table 6.1.

Table 6.1 Controller parameters for the double tank process. The values were obtained using the automatic tuning

Parameter	K_c	T_i	b
Values	0.3489	36.4368 *sec*	1.155

For the process given in equation (6.5) the parameters were also calculated using the Haalman method. Since this approach presumes this process structure we can assess the results of the automatic tuning method. The Haalman approach was designed to give the closed loop system the transfer function

$$G_{cl}(s) = \frac{2}{3Ls} e^{-Ls}$$

where the value 2/3 was found minimizing the mean square error for a set-point change. Thus the controller transfer function looks as follows

$$G_c(s) = \frac{2T}{3L}(1 + \frac{1}{Ts})$$

The obtained parameters using both - the Haalman and the automatic tuning - methods are given in Table 6.2.

Table 6.2 Comparison of the controller parameters, obtained from the Haalman approach and the automatic tuning

Method	K_c	T_i	b
Haalman	0.1333	1 *sec*	1
Automatic Tuning	0.1336	1.7137 *sec*	3.287

We can assess this parameters considering Figure 6.2, where a comparison is made using each controller for a closed loop simulation with the process from equation (6.5). In the left part of Figure 6.2 the output signals and in its right part the control signals are shown. The step-response of the Haalman system is depicted by a solid line, while the response of the system using the automatic tuning parameters is represented by a dashed line. First we consider the responses to a set-point change. It is obvious to see the much better result of the controller designed by using automatic tuning - even if the Haalman approach assumes the exact process structure.

If we consider the sensitivity of the Haalman system we obtain - exploiting equation (A.1) - a value of $M_s \approx 1.9$. Compared to $M_s = 1.4$ for the automatic tuning system we get closer to the critical point. This explains the overshoot of almost 20% while the automatic tuning system settles immediately. However

this robust design with $M_s = 1.4$ leads to worse results when controlling load disturbances. Consider the load disturbance response in Figure 6.2. Although the system controlled by the automatic tuning controller is slightly faster than the Haalman system the error gives a larger value. We can improve the results of the automatic tuning system by designing the controller with a larger value of M_s.

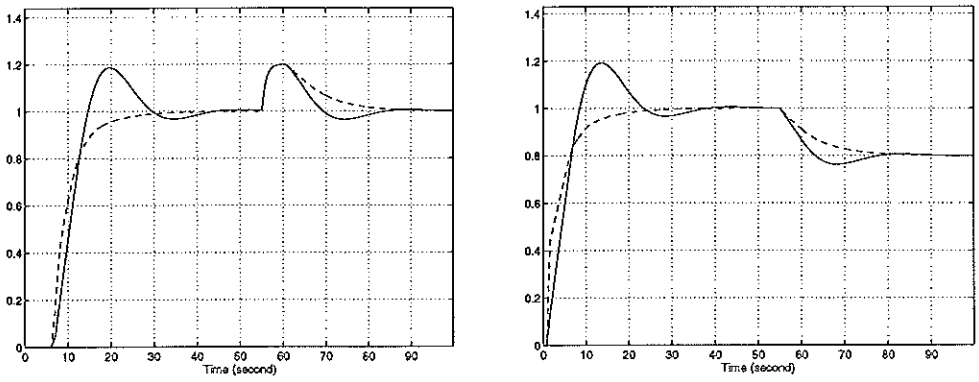

Figure 6.2 Comparison between the automatic tuning design (dashed) and the Haalman method (solid). Controlling the process given in equation (6.5) the output signal $y(t)$ (left) and control signal $u(t)$ (right) are shown

Nevertheless since we use feedforward control to compensate load disturbances the automatic tuning method is the most appropriate for the proposed structure. Hence in the following this method is used to design the controller parameters. However we will not use set-point weighting since this technique modifies the control deviation of the proportional part of the controller (see Appendix A). Thus even if we use an exact reference model the controller will be active after a set-point change. This leads to an interaction with the feedforward control signal. Hence b is set equal to one. Nevertheless there is still an substantial improvement using the automatic tuning method to design the feedback controller.

Performance Indices

To assess the obtained controller parameters and moreover the whole system more sophisticated we introduce several performance indices. Thus we get a measure of the performance improvement using the proposed control structure.

When designing a conventional controller there are mainly two objectives. Namely a good performance for set-point changes and load disturbances. One can also consider process or measurement noise but this is neglected in here. For the performance corresponding to a set-point change usually the settling time is considered. This quantity denotes the required time until the output signal stays within a certain range of the final value. In this thesis a range of 5% is used. Moreover the normalized maximum overshoot is considered. For it the maximum overshoot is normalized by the reference value. Thus we get

$$OS_{max} = \frac{y_{max} - w_{ref}}{w_{ref}} 100\%$$

Considering the response to load disturbances there are several penalty functions suggested in literature. The most common ones are given below with

IE-integrated error, IAE-integrated absolute error, ISE-integrated square error and IT^2SE-integrated time multiplied square error.

$$
\begin{aligned}
IE &= \int_0^\infty e(t)\ dt \\
IAE &= \int_0^\infty |e(t)|\ dt \\
ISE &= \int_0^\infty e(t)^2\ dt \\
IT^2SE &= \int_0^\infty (te(t))^2\ dt \\
ITAE &= \int_0^\infty t\ |e(t)|\ dt
\end{aligned}
$$

Usually the integrated error (IE) is chosen, because of its on-line capability - the integral is easy to evaluate. However since it cancels areas with positive and negative sign we use the integrated absolute error (IAE) to assess the performance improvement. This equation is also used to calculate an additional performance index for the set-point response to take into account the rise of both systems.

6.3 Feedforward Design

For the simulation a Radial Basis Function neural network was used to approximate the input-output characteristic of the process. The corresponding relation for the double tank process was already given in equation (6.2). Hence the new control law for this nonlinear process looks now such as given in equation (6.6), where the first part denotes the PI-controller signal and the second one the feedforward control signal.

$$u_{new} = K_c(1 + \frac{1}{T_i\ s}) + \frac{a_2}{K}\sqrt{2gh_{2ref}} \tag{6.6}$$

Since the RBF-net uses continuous (Gaussian) basis functions one can achieve any desired accuracy. Moreover we do not need any a priori knowledge to train its parameters since we can obtain the required information from simple closed loop measurement. Of course one can use also lookup table, splines or fuzzy logic to approximate the function.

The RBF-net has several advantages compared to other neural network structures (MLP-network). Since it contains the parameters linear we can apply linear optimization methods. Thus the approach is on-line capable. Also the local characteristic of the structure is sufficient since the range of the reference value - with respect to the investigated process - is restricted ($h = 0.2\ m$ means a full tank and $h = 0\ m$ denotes an empty one).

Of course, if we consider a linear process we get a linear input output relation. Then it is sufficient to approximate this function by a line. The line needs to have a slope equal to the inverse of the process gain. Nevertheless to consider the most general case we assume an (unknown) nonlinear process.

Neural Network Structure

To implement the new control law, given in equation (6.6), the neural network has to approximate the second term as function of the reference value h_{2ref}. For it eight basis functions were used so that the approximation has a form such as given in equation (6.7). The chosen initial values for the centers and the variances are given in Appendix B.2.

$$\hat{y}_{rbf} = \sum_{i=1}^{8} w_i \, exp\left(-\frac{1}{2}\left(\frac{u - c_i}{\sigma_i}\right)^2\right) \tag{6.7}$$

Data Collection

For collecting the training data the method sampling the control signal in closed loop was chosen. This method gives the best conditioned training data since the control signal is sampled with a constant step-size. To reach this the control signal corresponding to the reference values $h_{min} = 0\ m$ and $h_{max} = 0.2\ m$ was measured, and the input range was determined. Afterwards we divided the obtained range into 24 equidistant steps (3*number of basis functions). The reference is gradually increased until the control signal exceeds one of the obtained levels. In that case the reference is kept constant until the steady-state is reached then the control signal is measured. Thus we obtain 24 data pairs. The general manner is shown in Figure 4.4. This approach is very meaningful since it leads to more training data in the region where the nonlinearity has its largest gradient. Exactly in this region we need more information on the nonlinearity. We obtain less training data (information) for the area where the nonlinearity has less derivation. The measured data was already represented in Figure 4.6.

Adjusting The Parameters

The parameters were trained using the off-line learning approach from Figure C.3. It is also possible to use the on-line learning approach from Figure C.4 to set the parameters. The initial data of the centers were chosen with an increasing distance for larger numbers of h, and the variances according to the distance between each of the basis functions. There we have chosen some centers with negative values to deal with the different course of the nonlinearity in the region close to zero. The information on the nonlinear function can be obtained from the measured data.

To be more autonomous the centers can be chosen with every third measured data. For it one has to determine the number of required basis functions, and collect sufficient training data. If we use the above mentioned method to collect the data, then this leads also to a nonlinear distribution of the basis functions according to the gradient of the nonlinearity. The variances can be chosen corresponding to 1.5 times the distance between two neighbored basis functions. This method gives also considerable results.

The weights of the basis functions were calculated with the LS-algorithm given in equation (5.6). Afterwards also the centers and the variances were adapted with the Widrow-Hoff (WH) training algorithm. This was done to assess the results of the linear optimization by nonlinear optimization. For the nonlinear optimization also an adaptive step size η_{ad} was used to guarantee

a fast convergence of the nonlinear optimization algorithm. The weights obtained using the linear and the nonlinear optimization approach are given in Appendix B.2.

Approximation Results

The results of the approximation with the RBF-net (8 basis functions) are shown in this section. Figure 6.3 shows the approximation (left) and the approximation error (right) in relative measure.

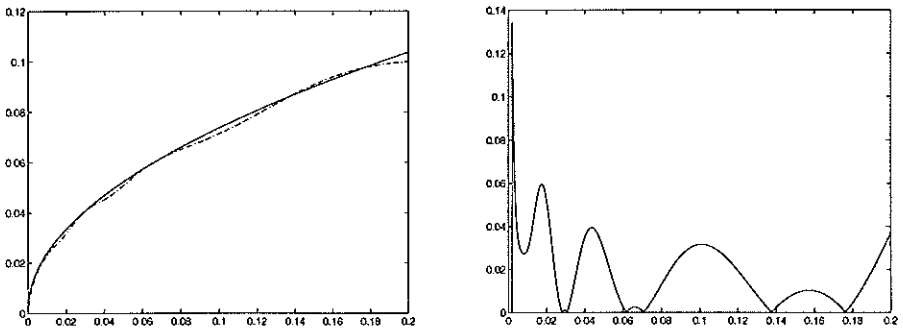

Figure 6.3 Neural net output $\hat{y}(t)$ (left) and approximation error in relative measure $e_{rel}(t)$ (right), using only linear optimization methods

If we apply only a linear optimization (LS) to adjust the weights of the neural network the approximation is already quite considerable. We get a maximal value for the error in relative measure of 6%. Note that this quantity is calculated such as given in equation (6.8), and that thus we almost get a division by zero for very small values of y. Hence we do not consider this quantity for small input values.

$$e_{rel} = \left| \frac{y - \hat{y}}{y} \right|$$

Only in the region close to the upper margin the approximation might be improved. This can be achieved by using more training data and basis functions, respectively. However this leads to more effort when collecting the training data. Moreover the feedforward is only used to come close to the desired output. On the other hand the approximation is remarkable for the region close to zero. We could expect some problems in that area because of the large gradients of the nonlinearity and the opposite shape of the Gaussian basis functions. One reason for it is the good training data and the increased number of basis functions in that region.

To assess the results of the linear optimization we use the obtained parameters as initial data for a nonlinear optimization. For it the variances, the centers and the weights were optimized corresponding to the equations derived in Appendix C.1.

We can see from Figure 6.4 and the values given in Appendix B.2 that even now (72 data points and 50 iterations) the approximation is almost not improved. This can be explained with the good choice of the initial data for

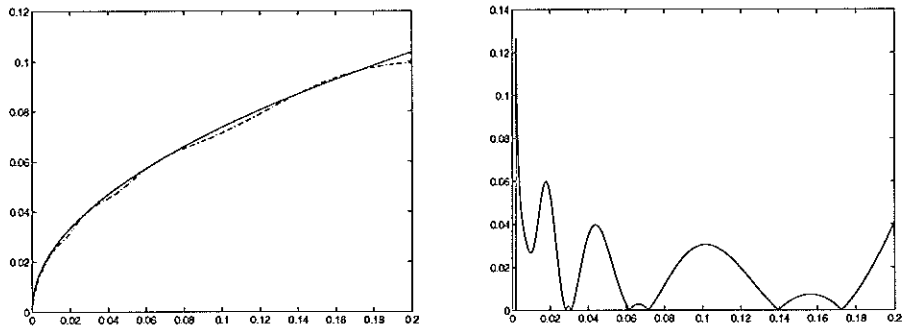

Figure 6.4 Neural net output $\hat{y}(t)$ (left) and approximation error in relative measure $e_{rel}(t)$ (right), using also nonlinear distributed optimization methods

the linear optimization. For it we have used information on the nonlinearity that was obtained by simple measurement without any a priori knowledge on the process. Hence the LS-estimation of the weights already led to good results.

It can be seen from the present investigation that linear optimization methods are absolute sufficient to train the parameters of the approximator. The necessary information we need to have to choose the initial data can be obtained from the measured training data.

To confirm the results we did a second simulation. This time we measured the training data using the method sampling the reference signal. This approach leads to constant distance between two neighbored basis functions. It turned out that eight basis functions could approximate the nonlinearity with an approximation error in relative measure of 5%. Only in the region close to zero the approximation was not really satisfying. The variances were chosen to 1.5 times the distance between two neighbored basis functions.

6.4 Reference Model

We design the three parameter reference model for the double tank process using the step-response method. We evaluate the part of the step-response that can be measured during the relay feedback. Let us consider again Figure 4.3. Since the relay does not switch for the second time until the output exceeds the reference value we can apply the method given in Section 4.4. If we determine the process gain in advance we obtain an estimate of T_m from the distance A-B and L_m from the distance 0-A, respectively (see Figure 4.8). Note that these values depend on the operating point. Hence it might be necessary to evaluate several step-responses. The operating point is defined by the amplitude of the relay. According to an operating point of $h_{op} = 0.1\ m$ the relay feedback parameters were already given in Section 6.2. The choice of the reference value ($h_{ref} = 0.64\ h_{op}$) guarantees that the controlled variable exceeds the threshold of 63% of the reference value.

The parameters of the reference model were computed as follows: $K_m = 1$, $T_m = 48\ sec$ and $L_m = 13\ sec$. Note that K_m has to be equal to one to avoid a modification of the set-point value. A comparison between the modified reference and the process response is depicted in Figure 6.5. The solid line

denotes the process response and the dashed one the modified reference signal.

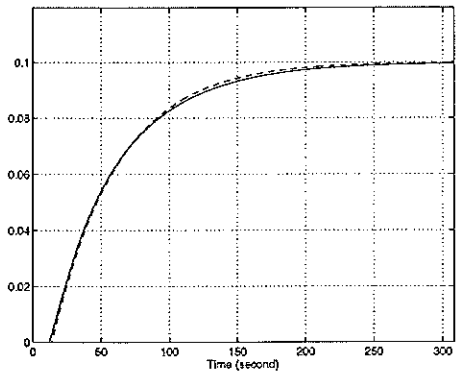

Figure 6.5 Comparison between reference model and process response corresponding to a step input of 0.1

We see that the achieved results are quite considerable. The model response is almost the same as the one of the process response. Since the reference model reduces the controller activity to set-point changes, the feedforward takes care of the major part of the control signal. Moreover, the feedforward supplies immediately the steady-state control signal. Hence the process output gives a step response of the system. To speed up the response even more we can use a simple trick. To reach this, we require an additional control signal during the transition that tends to zero the closer the response gets to the desired value. Of course one could implement this control law with a special controller but this would probably lead to a more sophisticated approach. To reach this course one can use another more simple method. The reference model response should rise faster than the response of the process. This results in a control deviation that causes the desired control signal from the PI-controller. To make the modified reference rise faster the model dead time has to be underestimated. This means that its value is set to smaller numbers than estimated. Nevertheless though Figure 6.5 shows a slightly opposite course the achieved results are considerable. This will be confirmed in Section 6.5.

6.5 Simulations

In this section the power of combining feedforward and feedback control is shown by means of several simulations. To model the double tank process we have used equation (6.1). Several block diagrams of the whole model that was used to simulate the double tank system are given in Appendix E. Moreover we have used equation (6.5) to model the second process. First, we consider the double tank system. In Figure 6.6 the levels h_2 of the big tank (left) and h_1 of the small tank (right) are shown. Figure 6.7 shows the control signal (left) and the control deviation (right), where the signals of the conventional system (solid line) are each compared to the signals of the combined approach (dashed lines). Moreover in Figure 6.8 the feedforward control signal (left) and the output disturbance (right, solid line) as well as the load disturbances l_1, l_2 and l_3 (right) are shown. They appear in the following order: at $t = 400$ *sec* the load disturbances l_1 (-30%), at $t = 700$ *sec* l_2 (20%) and at $t = 1000$ *sec*

l_3 (10%) act such as shown in Figure 6.1. Moreover the output disturbance n occurs at $t = 1300$ *sec* (10%). Figure 6.9 shows a set-point response without disturbances. It is used to assess the performance of both structures. The controller parameters obtained from the automatic tuning method are given in Table 6.1. Recall that b is set equal to one. The feedforward and the reference model were designed as outlined in Section 6.3 and Section 6.4, respectively. The load disturbance compensation was discussed in Section 4.5.

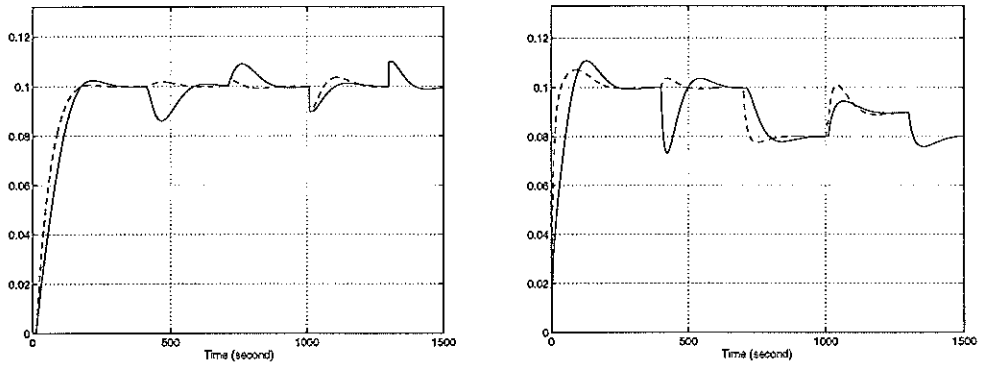

Figure 6.6 Disturbed liquid levels $h_2(t)$ of the big tank (left) and $h_1(t)$ of the small tank (right) corresponding to a set-point change. The solid line denotes a system controlled by standard feedback control and the dashed line represents a system controlled by a combination of feedforward and feedback control

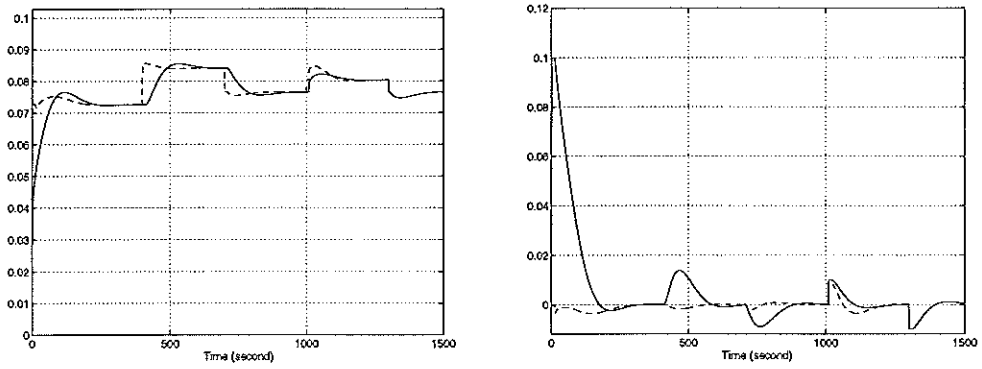

Figure 6.7 Control signals $u(t)$ (left) and control deviations $x_d(t)$ (right)

The simulation results show obviously improved performance of the extended control structure, even if the process is quite easy to control ($\tau = 0.133$), especially for load disturbance responses. First, we consider only the set-point response in Figure 6.9. We see a faster rise of the feedforward/feedback system compared to the standard feedback control. Note that the response of the feedback system already overshoot the desired value. Hence reducing the overshoot by modifying the controller parameters leads to a slower response. The benefit is more obvious if we take into account also an inner state of the process (the liquid level h_1 of the small tank). It is obvious to see the larger overshoot of this quantity. To have a real measure for the improvement we consider the performance indices that were introduced in Chapter 6.2.

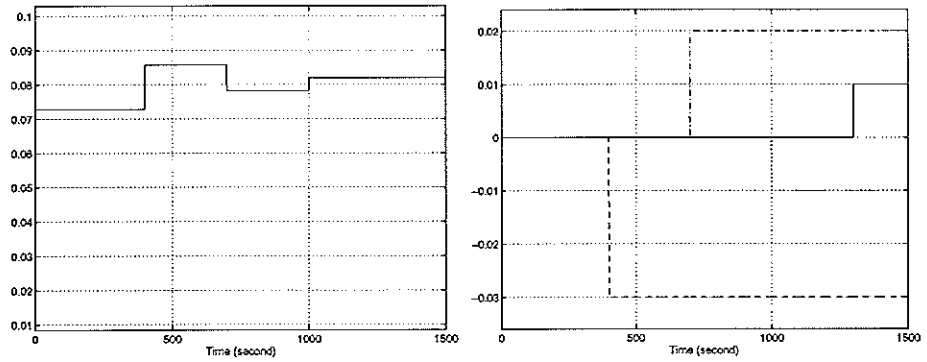

Figure 6.8 Feedforward control signal (left) and load disturbances $l1$ (dashed), l_2 (dashed dotted), l_3 (dotted, each right) and output disturbance $n(t)$ (solid, right)

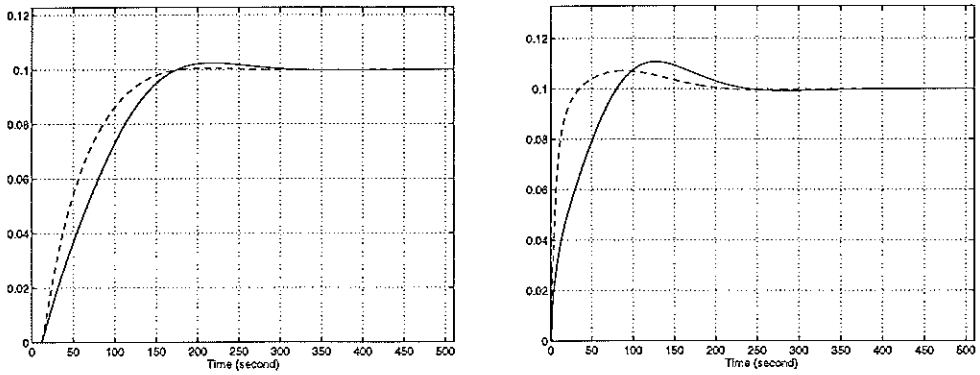

Figure 6.9 Undisturbed liquid levels $h_2(t)$ of the big tank (left) and $h_1(t)$ of the small tank (right) corresponding to a set-point change. The solid line denotes a system controlled by standard feedback control and the dashed line represents a system controlled by a combination of feedforward and feedback control

The performance indices corresponding to a set-point change were calculated using the settling time and the normalized maximum overshoot. Moreover the integrated absolute error (IAE) is considered to take into account the transition of both structures. The corresponding values are given below in Table 6.3. For all performance indices the combination of feedforward and feedback control gives better results than standard feedback control.

Table 6.3 Different set-point performance indices for standard feedback control and the combined feedforward/feedback control approach applied to the double tank process

Perf. Index	*Feedback Control (FB)*	*Combined Approach (CA)*
T_{settle}	150 sec $= 3.13T_m$	130 sec $= 2.71T_m$
$\frac{y_{max}-w_{ref}}{w_{ref}}$	2.32%	0.53%
IAE	7.40	5.59

Now we consider several load disturbance responses of both systems, where the disturbances act such as depicted in Figure 6.1. We see from Figure 6.6 to Figure 6.8 that the proposed structure can compensate the load disturbances the best that act before the slow process dynamics. In the optimal case if

there is no model error by the approximation of the input-output relation and the disturbance compensation the first load disturbance has no effect to the output signal. In this case the disturbance is compensated immediately by an accurate feedforward signal - before it affects the controlled variable. With decreasing speed of the dynamics the improvement decreases until the point where the affect of the slow process dynamics appears too much. Then the feedforward compensation does not give an improvement compared to feedback control, since there is an interaction from the feedback controller. Hence only the feedback controller should be used to compensate these (load) disturbances. We can see from Figure 6.6 that a short output disturbance is almost not affected by the feedback controller because of the slow process dynamics. The responses of both systems to the output disturbance are the same we do not use feedforward compensation and the controller parameters are the same. The proposed structure shows in total a much better performance than the conventional system.

Again we consider the performance indices to assess the performance improvement. For it we have chosen the IAE-criterion. The corresponding values of the IAE-criterion for different load disturbances are given below in Table 6.4. The IAE-criterion applied to an output disturbance of 10% gives for both systems a value of $IAE_n = 0.59$.

Table 6.4 IAE-criterion to assess the load disturbance responses of standard feedback control and the combined feedforward/feedback control approach applied to the double tank process

Load Disturbance	IAE_{FB}	IAE_{CA}
l_1	1.34	0.17
l_2	0.91	0.18
l_3	0.62	0.69

For the load disturbances l_1 and l_2 the IAE-criterion gives much smaller values for the combination of feedforward and feedback control compared to standard feedback control. It is advantageous to use the proposed structure to compensate these load disturbances. However the interaction between feedback and feedforward leads to slightly worse results when using the proposed structure to compensate the load disturbance l_3.

To show the full power of the proposed structure we consider the simulations using the second process which was given in equation (6.5). This process with a normalized dead time equal to $\tau_2 = 0.833$ was introduced since it is in general difficult to design a controller - using classical methods - for a value of the normalized dead time τ that is close to one.

In Figure 6.10 the output signal $y(t)$ (left) and the control signal $u(t)$ (right) of both systems are shown. Again we compare the proposed structure (dashed line) with a standard feedback control (solid line). In Figure 6.11 the control deviation $x_d(t)$ (left) and the feedforward control signal u_{ff} (right) are shown. Moreover in Figure 6.12 the disturbances are depicted. The load disturbance l_1 (dashed) acts to the control signal at $t = 50\ sec$ (with 30%), the

load disturbance l_2 (dashed dotted) to the output of the fast process dynamic at $t = 100$ *sec* (with -20%), and the output disturbance n acts to the process output at $t = 150$ *sec* (with 10%). Also in here the controller parameters are obtained with the automatic tuning approach. The set-point weighting is still used for the conventional control loop since this gives the best results for standard feedback control. The set-point weighting is not used for the feedforward/feedback structure since this would lead to an interaction by the feedback controller.

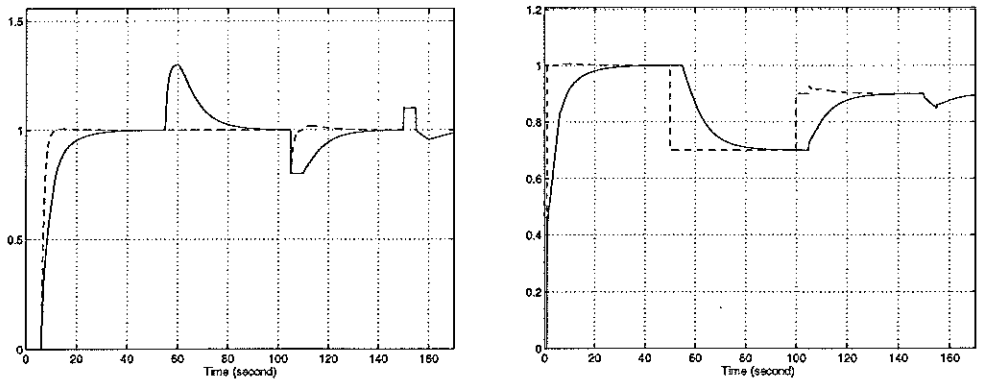

Figure 6.10 Output $y(t)$ (left) and control signal $u(t)$ (right) of the second process for a set-point change

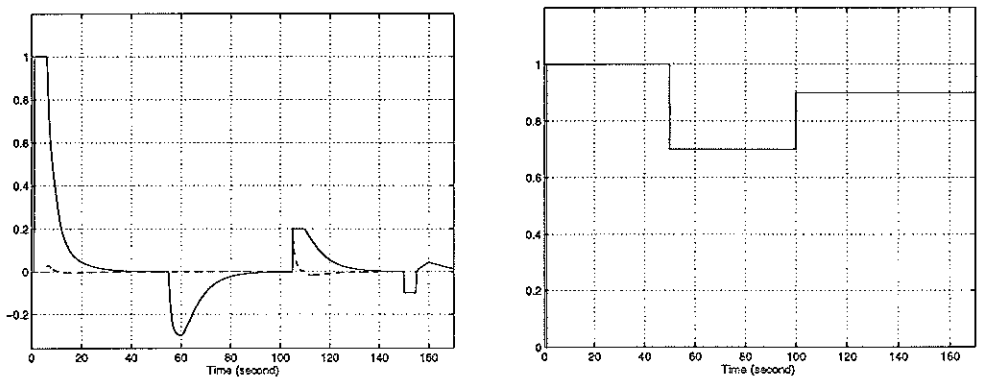

Figure 6.11 Control deviations $x_d(t)$ (left) and feedforward control signal $u_{ff}(t)$ (right)

One can see from this simulations that the advantage using the combination of feedforward and feedback control is even larger when controlling more complex processes ($\tau \rightarrow 1$). Even if the process time constant is estimated with an error of 5%, the step-response is almost perfect (Note the dead time of $L = 5$ *sec*). Also the disturbance compensation by the feedforward shows very good results for l_1 and l_2 - much better than for standard feedback control. Again we would get worse results when using the feedforward to compensate output disturbances. This is taken into account by the PI-controller.

The results are also reflected in the performance indices for the set-point response (given in Table 6.5) and the load disturbance responses (given in Table 6.6). Notice the time delay that gives a large value for the IAE-criterion applied to the set-point response. The overshoot of the combined approach is

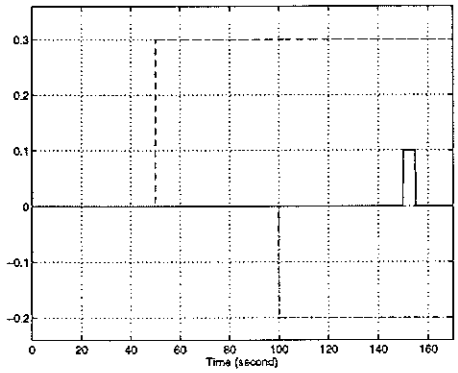

Figure 6.12 Load disturbances $l_1(t)$ (dashed) and $l_2(t)$ (dashed dotted) and output disturbance $n(t)$ (solid)

caused by the inaccurate estimation of the reference model time constant. Notice the substantial performance improvement when using the feedforward to compensate the load disturbances l_1 and l_2.

Table 6.5 Different set-point performance indices for standard feedback control and the combined feedforward/feedback control approach applied to the process given in equation (6.5)

2mm

Perf. Index	*Feedback Control (FB)*	*Combined Approach (CA)*
T_{settle}	19.3 sec $= 20.3T_m$	8.9 sec $= 9.4T_m$
$\frac{y_{max}-w_{ref}}{w_{ref}}$	0%	0.57%
IAE	8.90	6.06

Table 6.6 IAE-criterion to assess the load disturbance responses of standard feedback control and the combined feedforward/feedback control approach applied to the process given in equation (6.5)

Disturbance	IAE_{FB}	IAE_{CA}
l_1	3.84	0.00
l_2	2.56	0.41
n	0.99	0.99

7. Conclusion And Future Work

In this report a control structure that combines feedback and feedforward control was investigated. The rapid response of feedforward control is used to speed up conventional PI-control. Hence the inherent accuracy of feedback control is maintained as well as the speed of feedforward control.

An introduction to the problem of controlling processes with slow dynamics was given in Chapter 1. Chapter 2 gave a survey of the combined feedforward/feedback control approach. Chapter 3 discusses feedforward control and its properties. The design of the proposed controller was treated in Chapter 4. We have outlined that the whole system can be designed without detailed process knowledge. Automatic tuning was used to design the feedback controller. Even the information required to implement the feedforward can be obtained from simple closed loop experiments. There are several ways to approximate the static (nonlinear) process characteristic. They were discussed in Chapter 5. The precision of the feedforward can be improved by using an on-line learning structure. New process information that is obtained during operation can thus be taken into account. Implementation and the simulations for several processes are discussed in Chapter 6. Several performance indices were introduced to assess the performance improvement.

It turned out from the previous investigation that the augmentation of feedback with feedforward control leads to an improved performance for control of processes with slow dynamics. The load disturbance response can be improved substantial, since feedforward control is suitable for load disturbance compensation. This gives another interesting area where this approach can be used advantageous. Moreover, the system can be designed without a detailed process knowledge. The method can thus be used even for simple processes where a process model is not available (at least with moderate effort).

It is the simple, automated design that make this control structure a powerful tool for automatic control. Existing feedback control loops can be adapted in a very simple way. Notice that a major part of industrially used PID-controllers act only as a PI-controller. Because of the existing process noise the derivative part is switched off, since it would otherwise amplify the noise.

It was shown that the advantage of the control structure is even larger for more complex processes. There the significance of the process dynamics can be characterized by the normalized dead time $\tau = L/(L+T)$. This quantity gives a measure of the difficulty to control the process with conventional feedback control.

So far the approach is only applicable to stable processes. Hence, a direction for future work could be the extension of the proposed approach to unstable processes. The design then has to be modified since the automatic tuning was derived using a test batch of stable processes. A starting point could be to investigate the controller design methods that are used in adaptive control.

Because there is also an autonomous design required. Moreover one need to use a different method to determine the parameters of the reference model.

Since the neural network used can be trained on-line to improve the approximation the structure is appropriate to control processes containing dynamical nonlinearities. For that goal a dynamical neural network should be used. Hence this can even lead to a new controller for adaptive control systems.

A. Supplement To Automatic Tuning

In this chapter few information on the sensitivity of a process and on set-point weighting is given. For more details see Åström and Hägglund [1].

A.1 Sensitivity

The sensitivity of a process can be characterized by the M_s-value such as given in equation (A.1). M_s is equal to the inverse of the shortest distance from the Nyquist curve to the critical point (-1/0). This admits a direct interpretation as a robustness measure, because it tells how much the process can change without causing instability. Reasonable values for M_s are in the range from 1.3 to 2.

$$M_s = \max_{0 \le \omega < \infty} \left| \frac{1}{1 + G_P(i\omega)\, G_C(i\omega)} \right| = \max_{0 \le \omega < \infty} |S(iw)| \qquad \text{(A.1)}$$

An useful physical interpretation of the sensitivity function S is also given in Åström and Hägglund [1]. They assume a sinusoidal disturbance with frequency ω that enters the system in an arbitrary way. Moreover the amplitude of the open loop system is assumed to be α_0. If the system is controlled with a controller that gives the sensitivity function S, the amplitude of the controlled system is then $\alpha_0|S(i\omega)|$. Feedback thus reduces the effect of the disturbance if $|S(i\omega)| < 1$, and it amplifies a disturbance if $|S(i\omega)| > 1$.

Another common measure to describe the sensitivity of a function is given by the amplitude margin (A_m) and the phase margin (φ_m). Their equations and the relation to M_s is given as follows.

$$A_m = \frac{1}{|G_{cl}(i\omega_u)|} > \frac{M_s}{M_s - 1}$$

$$\varphi_m = \pi + arg G_{cl}(i\omega_g) > 2 arcsin \frac{1}{2M_s}$$

Where the ultimate frequency ω_u is the frequency where $arg G_{cl}(i\omega) = -\pi$ and the gain cross-over frequency ω_g is the frequency where $|G_{cl}|(i\omega) = 1$. Typical values for these quantities are: $\varphi_m = 30^\circ \ldots 60^\circ$ and $A_m = 2 \ldots 5$.

A.2 Set-point Weighting

Set-point weighting leads to a more flexible controller structure. A weighting-parameter b is introduced to treat the set-point and the process output separately. Thus the error in the proportional part becomes

$$e_p = b\, w_{ref} - y$$

Hence the new control law for a PI-controller looks like

$$u_{sp} = K \left(e_p + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right)$$

B. Parameter values

B.1 Automatic Tuning Design Parameters

In Table B.1 the automatic-tuning design parameters are given corresponding to a maximum sensitivity of $M_s = 1.4$. The ultimate gain K_u, the ultimate period T_u and the gain ratio $\kappa = 1/K_u K_p$ are used to obtain the controller parameters by using equation (4.1).

Table B.1 Automatic tuning design parameters

Parameter	a_0	a_1	a_2
K_c/K_u	0.053	2.9	-2.6
T_i/T_u	0.9	-4.4	2.7
b	1.1	-0.0061	1.8

B.2 Neural Network Parameters

In this section the neural network parameters are given that were used to approximate the input-output characteristic of the double tank process.

Linear Optimized Parameters

In equation (B.1) the data is given that was obtained by applying linear optimization. The center c_{lin} and the variances σ_{lin} were chosen a priori using information on the function to be approximated. This information was obtained from 24 measurement (3*number of basis functions). The vector w_{lin} containing the weights was obtained by applying the LS-algorithm.

$$c_{lin} = \begin{bmatrix} 0.01 \\ 0.02 \\ 0.04 \\ 0.08 \\ 0.16 \\ -0.02 \\ -0.04 \\ -0.08 \end{bmatrix}, \quad \sigma_{lin} = \begin{bmatrix} 0.00625 \\ 0.0125 \\ 0.0148148 \\ 0.05 \\ 0.32 \\ 0.0125 \\ 0.0148148 \\ 0.05 \end{bmatrix}, \quad w_{lin} = \begin{bmatrix} 0.012459199595 \\ 0.020912772147 \\ -0.003784069419 \\ -0.030846741672 \\ 0.102381373251 \\ 0.661270805064 \\ -4.825285621693 \\ -0.530263457690 \end{bmatrix} \quad \text{(B.1)}$$

Nonlinear Optimized Parameters

In Equation (B.2) the data is given that was obtained by applying nonlinear optimization. The initial data was chosen equal to the data given in equation B.1 that was obtained by applying linear optimization. For the nonlinear optimization the WH-algorithm was chosen (see Appendix C). There, 24 parameters were trained with 72 measured data (3*3*number of basis functions) and for 50 iterations.

$$c_{nl} = \begin{bmatrix} 0.01000004590159 \\ 0.01999998078040 \\ 0.04000000123127 \\ 0.08000000360773 \\ 0.16000000153626 \\ -0.01999999786226 \\ -0.04000000678425 \\ -0.07999999695637 \end{bmatrix}, \quad \sigma_{nl} = \begin{bmatrix} 0.00625006152015 \\ 0.01249999591862 \\ 0.01481482336518 \\ 0.05000003163366 \\ 0.32000000015596 \\ 0.01249999506271 \\ 0.01481479742768 \\ 0.05000001740391 \end{bmatrix},$$

$$w_{nl} = \begin{bmatrix} 0.01377088399698 \\ 0.02234542279479 \\ -0.00337519083878 \\ -0.03025778974767 \\ 0.10193358551286 \\ 0.67033844876038 \\ -4.76696641915301 \\ -0.54480698470091 \end{bmatrix} \tag{B.2}$$

C. Training The Neural Network

In this Chapter the training of the neural network is described. We derive the Widrow-Hoff training algorithm and apply the algorithm to the RBF-net. Thus we obtain the equations to adjust the parameters by a nonlinear optimization. Afterwards the training and generalization phase is explained. Moreover regularization is introduced to improve the robustness of standard parameter identification methods. Finally a training schedule is given.

C.1 Training Algorithm

In this section the Widrow-Hoff (WH) training algorithm is derived. We start with brief description of the Multi Layer Perceptron, because the WH-rule is based on this structure.

Multi Layer Perceptron (MLP) Network

In Figure C.1 a single perceptron is shown. We can see that the MLP summarizes the weighted input vector $\underline{u}$. This corresponds to a reduction of the input dimensionality. Hence the MLP is well appropriate for higher dimensional problems in opposite to the RBF-net. Notice that the RBF-net treats each input as a dimension. This leads to an enormously increased number of parameters.

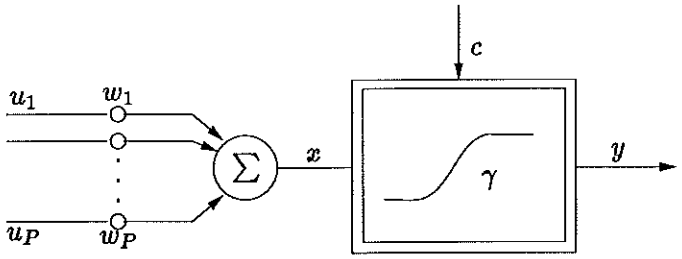

Figure C.1 Perceptron

The transfer function of a perceptron is given by the following equations.

$$x = \sum_{i=1}^{P} w_i u_i = \underline{u}^T \underline{w}$$

$$y = \gamma(x - c)$$

Where x denotes the activity of the perceptron, $\underline{u}$ is the input vector of order P, $\underline{w}$ is the weight vector also of order P, y denotes the output of the perceptron, γ is the activity function and c is the threshold level. There are several functions that can be used as activity function. Usually the sigmoidal function is chosen. It looks like depicted in Figure C.2.

Mathematically the sigmoidal function is given by the following equation

$$y = \frac{1}{1 + e^{-(x-c)}}$$

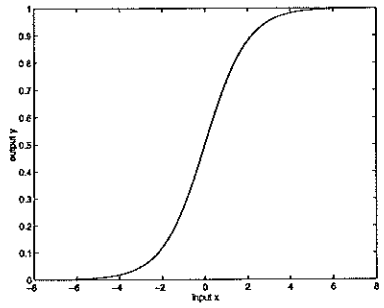

Figure C.2 Sigmoidal activity function

The threshold level c moves the range where the perceptron is most active. In Figure C.2 c is set equal to zero. We see a linear range around this operating point. Considering the right margin of the sigmoidal function we can see the global characteristic of a perceptron. This means that the perceptron gives an output value greater than zero even if there is an input value far from the operating range. This is another difference to the RBF-net that has only a local characteristic. Thus the RBF-net gives for the same input value an output value equal to zero. Caused by the nonlinear optimization that has to be applied the perceptron is only appropriate for off-line applications.

Widrow-Hoff Training Algorithm

The probably most common nonlinear optimization algorithm that is applied to a neural network is the Widrow-Hoff training algorithm. It is derived from a single perceptron. Again we use the quadratic loss function from equation (5.5). The condition for adjusting the weights w - to get the minimum of the loss function - is given by

$$w_{i_{new}} = w_{i_{old}} - \eta \frac{\partial J(w_i)}{\partial w_i} \tag{C.1}$$

Introducing the error $e(n) = y(n) - \hat{y}(n)$ and calculating the derivative of J with respect to the parameters w_i, we get

$$\begin{aligned} w_{i_{new}} &= w_{i_{old}} + \eta \sum_{n=1}^{N-1} e(n) \frac{\partial y(n)}{\partial w_i} \\ &= w_{i_{old}} + \eta \sum_{n=1}^{N-1} e(n) \dot{\gamma} \frac{\partial x(n)}{\partial w_i} \\ &= w_{i_{old}} + \eta \sum_{n=1}^{N-1} e(n) \dot{\gamma} u_i(n) \end{aligned}$$

Where $\dot{\gamma}$ is the derivate of the activity function. This approach using all measured data for calculating the change of the weights is called the accumulative approach. Using each one measured data for calculating the parameter change leads to the singular approach. The equation of this simplified algorithm can be written mathematically as

$$w_i(n) = w_i(n-1) + \eta\, e(n)\, \dot{\gamma}\, u_i(n)$$

Adaptation Equation For The RBF-Net

To obtain the adaptation equations for the parameters of the RBF-net we apply the WH-algorithm to equation (6.7). Using the singular approach we

get for the centers

$$\begin{aligned} c_{i_{new}} &= c_{i_{old}} - \eta \frac{\partial J(c_i)}{\partial c_i} \\ &= c_{i_{old}} + \eta\, e\, \frac{\partial \hat{y}}{\partial c_i} \\ &= c_{i_{old}} + \eta\, e\, \frac{\partial}{\partial c_i} \left(\sum_{i=1}^{M} w_i\, exp\left(-\frac{1}{2} \left(\frac{u-c_i}{\sigma_i} \right)^2 \right) \right) \end{aligned}$$

Thus we obtain finally the adaptation equation for the centers of each basis function.

$$c_{inew} = c_{iold} + \eta\, e\, \frac{w_i}{(\sigma_{iold})^2} (u - c_i)\, exp\left(-\frac{1}{2} \left(\frac{u - c_i}{\sigma_{iold}} \right)^2 \right) \quad \text{(C.2)}$$

Deriving the adaptation equations for the variances and the weights of each basis function in an analog way we get

$$\sigma_{inew} = \sigma_{iold} + \eta\, e\, w_i\, (u - c_i)^2\, \frac{1}{(\sigma_{iold})^3}\, exp\left(-\frac{1}{2} \left(\frac{u - {c_i}^2}{\sigma_{iold}} \right) \right) \quad \text{(C.3)}$$

$$w_{inew} = w_{iold} + \eta\, e\, exp\left(-\frac{1}{2} \left(\frac{u - c_i}{\sigma_{iold}} \right)^2 \right) \quad \text{(C.4)}$$

To ensure the convergence of the optimization algorithm we use an adaptive step-size instead of a fixed value. The strategy to adapt the step size is given in the following part of a MATLAB-file.

```
if (J(k+1) <= (1+epsilon)*J(k))
  % increase stepsize
  eta_new = alpha*eta_old
else
  % decrease stepsize
  eta_new = beta*eta_old
  % discard last step
  teta(k) = teta(k-1)
end;
```

Usually the parameters are chosen as follows: $\varepsilon = 0.04$, $\alpha = 1.04$ and $\beta = 0.7$.

C.2 Training and Generalization

In this thesis only supervised learning methods are considered where input and output data is available. The output data is used to calculate the error between the desired value and the neural network output. This quantity is then used to adjust the parameters minimizing the quadratic error. There are also some unsupervised learning methods, where only input data is required. The data can be summarized to groups by special algorithms (clustering). Nevertheless this methods are not further investigated in this report.

When adjusting the parameters (for the first time) there are different methods that can be used. First, we consider the off-line learning structure such as shown in Figure C.3. It uses the data that was measured before to adjust the parameters Θ via the error e. If we apply a linear optimization then the parameters are obtained within one step.

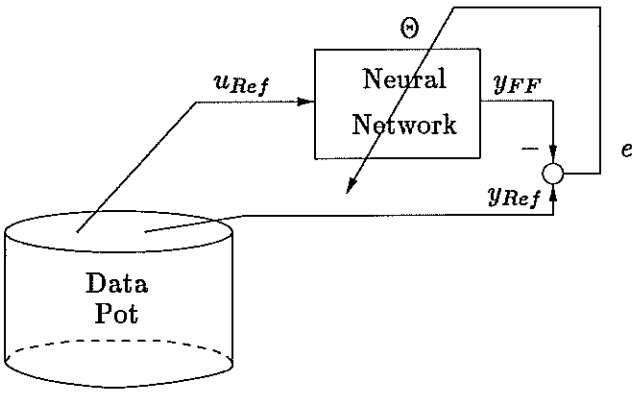

Figure C.3 Off-line learning structure

To adjust the parameters on-line one can use the structure such as shown in Figure C.4. It uses actual measured data to adjust the parameters according to the new available process information. In here this approach is not well appropriate to tune the parameters for the first time, because it takes some time before new data is available (Recall the slow dynamics). Nevertheless this structure is very meaningful to improve the approximation of the off-line structure using new measured data. Moreover it is possible to adapt the feedforward to process variations.

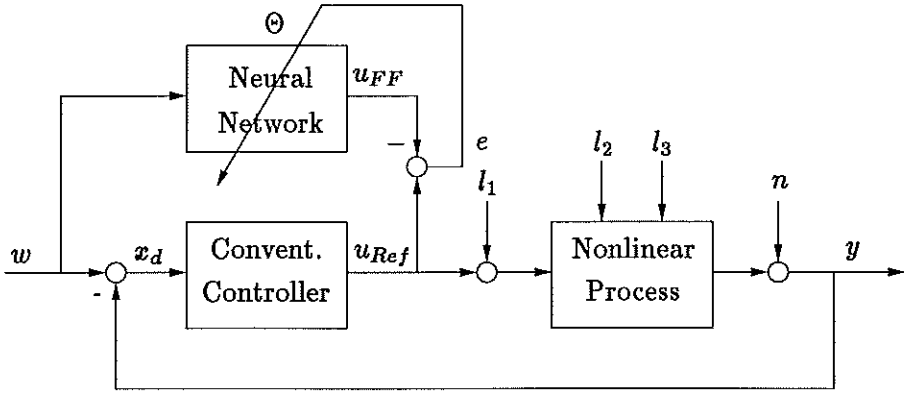

Figure C.4 On-line learning structure

The quality of the approximation that is within reach depends crucial on the measured data. Furthermore if we use a nonlinear optimization we have to avoid a memorization of the measured data. This can be reached by splitting up the whole data set into a training and a generalization data set. The training data is used to adjust the parameters of the neural network, while the generalization data verifies the approximation. We can check for a memorization by determination of the loss function for both data sets. If there is a memorization the loss function tends to zero when using the training data, but it increases if the generalization data is used. This relation is depicted in Figure C.5. The figure is taken from Ayoubi and Nelles [7].

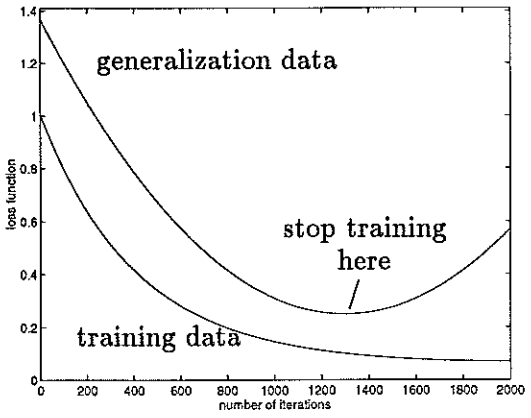

Figure C.5 Loss function for training and generalization data

We can see that the training should be terminated when the loss function - that is calculated using the generalization data - has its minimum. Then we get the best performance.

Using an approximator that stores all training data can force the loss function to zero, of course, but it will not reach optimal quality for the generalization data. It is just this effect one wants to avoid, because in this case all disturbances contained in the measured data will be approximated. This problem was already discussed in Section 5.4.

In the next section an approach is introduced that can be used if it is not possible to collect suitable training data for the whole range (lack of data in some region or bad conditioned data). Usually this leads to bad results when applying the optimization. Thus the obtained parameters are spread over a wide range.

C.3 Regularization

The regularization approach avoids this problem by attracting all available degrees of freedom to that region. Thus it is possible to improve the robustness of standard parameter identification methods. The basic idea is to introduce a penalty term in the least squares criterion that attracts any excessive degrees of freedom in the model structure towards reasonable regions of the parameter space. Excessive degrees of freedom are available if the quality criterion is not sensitive to perturbations in a certain parameter sub-space. Then the penalty term should add a significant value to the criterion for that parameter sub-space. If we assume to use the LS-algorithm to estimate the parameters the modified criterion looks like

$$J_{Reg}(\underline{\Theta}, \underline{D}_M) \;=\; J(\underline{\Theta}, \underline{D}_M) \;+\; \gamma\Omega(\underline{\Theta}) \tag{C.5}$$

Where $\underline{\Theta}$ denotes the parameter vector, $\underline{D}_M$ the measured data, $\gamma > 0$ the regularization parameter and Ω the stabilizer for the problem. The last parameter is most commonly chosen to be the Levenberg-Marquardt stabilizer.

$$\Omega_{LM} \;=\; (\underline{\Theta} - \underline{\Theta}^*)^T(\underline{\Theta} - \underline{\Theta}^*) \tag{C.6}$$

This approach attracts the parameters towards a point $\underline{\Theta}^*$ in the parameter space. If there is no a priori knowledge available this variable is usually set equal to zero. This gives still a substantial robustness improvement. Also other stabilizers like the Tikhonov stabilizer can be used. The achievable improvement using the regularization approach is quite considerable. For more details see Johansen [5].

C.4 Training Schedule

A general schedule for implementing the training phase with a combination of linear and nonlinear optimization methods is given below.

1. set initial values of centers $c_i(0)$
2. set initial values of variances $\sigma_i(0)$
3. get initial weights $w_i(0)$ from LS-algorithm
4. calculate RBF-output, error signal and training loss function
5. adapt centers c_i (with w_i and σ_i constant) according to equation (C.2)
6. adapt variances σ_i (with w_i and c_i constant) according to equation (C.3)
7. adapt weights w_i (with c_i and σ_i constant) according to equation (C.4)
8. while number of current data is less than total number of data, get new data and goto 4
9. calculate generalization loss function
10. while generalization loss function decreases adapt step size η and start new iteration
11. ...
    ⋮

Using only linear optimization to adjust the weights one can terminate the training schedule after step 3.

D. Process Model

In this chapter we derive the exact mathematical model for the double tank process using mass and energy balances. First, consider only the small tank (compare to Figure 6.1). Applying a mass balance we get

$$A_1 \frac{dh_1}{dt} = q_{1in} - q_{1out} \tag{D.1}$$

where A_1 denotes the cross-section area of the small tank, $\frac{dh_1}{dt}$ is the time derivative of the water height in the tank, q_{1in} is the input mass-flow and q_{1out} is the output mass-flow. Moreover, when applying also an energy balance (Bernoulli) to the small tank we get

$$\rho g h_1 = \frac{\rho {v_1}^2}{2} \tag{D.2}$$

where ρ is the water density, g is the acceleration of gravity, h_1 is the water height in the tank and v_1 is the velocity of the out-flowing water. For the out-flowing water-mass of the small tank we can write

$$q_{1out} = a_1 v_1 \tag{D.3}$$

where a_1 denotes the cross-section area of the out-flow of the tank. Using equation (D.2) and equation (D.3) leads to equation (D.4) for the output mass-flow q_{out}.

$$q_{1out} = a_1 \sqrt{2gh_1} \tag{D.4}$$

The input mass-flow can be described using the electrical equation of the water pump

$$q_{1in} = q_{pump} = Ku \tag{D.5}$$

where K stands for the gain of the water pump and u is the control signal. Inserting equation (D.4) and equation (D.5) into equation (D.1) we get the nonlinear model of the small tank.

$$A_1 \frac{dh}{dt} = Ku - a_1 \sqrt{2gh_1} \tag{D.6}$$

Consider now the big tank. Starting again with the mass balance we get

$$A_2 \frac{dh_2}{dt} = q_{2in} - q_{2out} \tag{D.7}$$

where $q_{2in} = q_{1out}$. Using this relation and equation (D.4) - corresponding for the big tank - we finally get the process model given in equation (6.1).

E. Simulation Model

For the simulations MATLAB-Simulink was used. In Figure E.1 the block diagram of the simulated control structure for the double tank system is shown. Figure E.2 shows the reference model and Figure E.3 the PI-controller. The feedforward is shown in Figure E.4. Moreover, Figure E.5 shows the neural network and Figure E.6 the model of one tank.

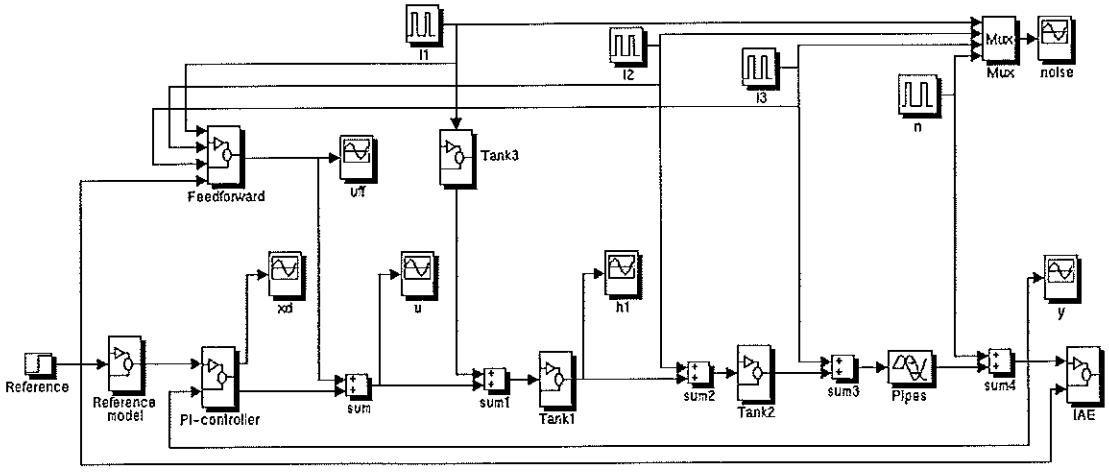

Figure E.1 Block diagram of the proposed controller for the double tank system

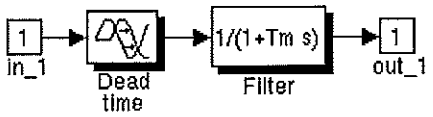

Figure E.2 Block diagram of the reference model

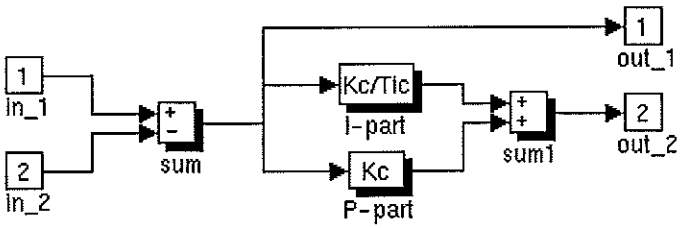

Figure E.3 Block diagram of the feedback controller

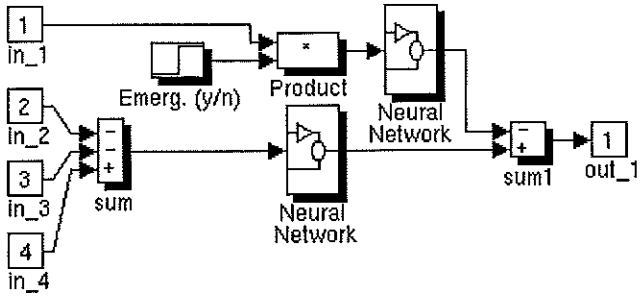

Figure E.4 Block diagram of the feedforward

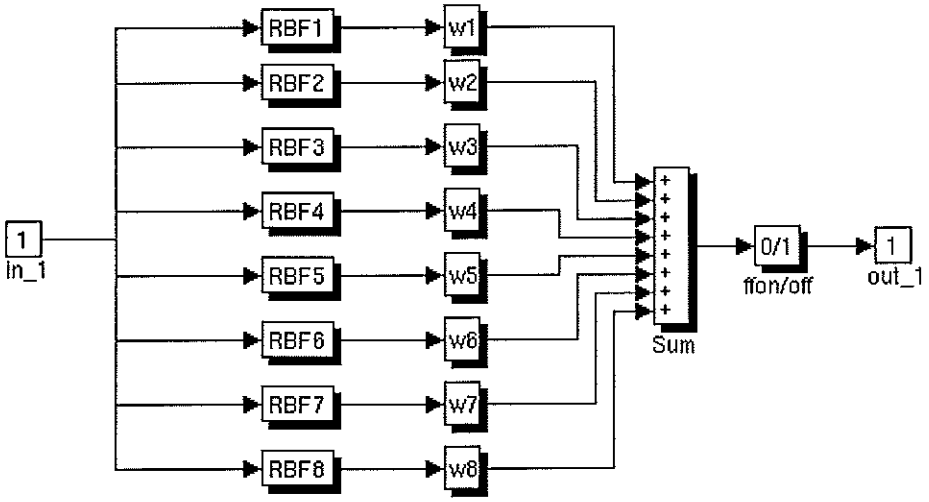

Figure E.5 Block diagram of the neural network

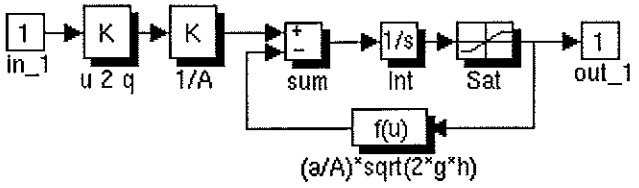

Figure E.6 Block diagram of one tank

F. Bibliography

[1] K.J. Åström and T. Hägglund. *PID-Control - Theory, Design and Tuning - 2nd Edition.* ISA-The international society of measurement and control, 1995.

[2] K.J. Åström and B. Wittenmark. *Adaptive Control.* Addison-Wesley Publ. Comp., 1989.

[3] K.J. Hunt et al. A review of advances in neural adaptive control systems. Technical report, Daimer-Benz AG, University of Glasgow, 1994.

[4] Rolf Isermann. *Identifikation dynamischer Systeme 1.* Springer Verlag, 1992.

[5] T. A. Johansen. Robust identification of takagi-sugeno-kang fuzzy models using regularization. Technical report, SINTEF Automatic Control, N-7034 Trondheim, Norway, 1996.

[6] K.-H. Lachmann. Lecture notes on adaptive control systems. Lecture notes for the graduate course 'Adaptive Regelsysteme 95/96' at Technical University of Darmstadt, Automatic Control, Control Engineering and Proces Automation.

[7] O. Nelles M. Ayoubi. Einfuehrung in die neuronalen netze zur identifikation nichtlinearer systeme - skript zur vorlesung identifikation dynamischer systeme. Lecture notes for the graduate course 'Identifikation nichtlinearer Systeme 95' at Technical University of Darmstadt, Automatic Control, Control Engineering and Proces Automation.

[8] W. F. Ramirez. *Process Control and Identification.* Academic Press, 1994.

[9] F. G. Shinskey. *Process-Control Systems Application Design Adjustment.* McGraw-Hill, New York, second edition, 1979.

[10] L.X. Wang. *Adaptive Fuzzy Control Systems and Control: Design and Stability Analysis.* Prentice Hall, International Editions, 1994.

[11] Philip D. Wasserman. *Neural Computing - Theory and Practice.* Van Nostrand, Reinhold - New York, 1989.