Self-Tuning PID-Controllers Based on Pole Placement

Wittenmark, Björn

1979

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Wittenmark, B. (1979). *Self-Tuning PID-Controllers Based on Pole Placement*. (Technical Reports TFRT-7179). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7179)/1-037/(1979)

SELF-TUNING PID-CONTROLLERS BASED ON POLE PLACEMENT

B. WITTENMARK

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
OCTOBER 1979

Dokumentutgivare
Lund Institute of Technology
Dept of Automatic Control

Handläggare
B Wittenmark

Författare
B Wittenmark

Dokumentnamn
REPORT

Dokumentbeteckning
LUTFD2/(TFRT-7179)/1-037/(1979)

Utgivningsdatum
October 1979

Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel
Self-tuning PID-controllers based on pole placement

Referat (sammandrag)
The paper describes an algorithm for automatic tuning of discrete time PID-controllers. The algorithm consists of two parts. Firstly a second order model of the process is estimated recursively using the method of least squares. Secondly the parameters of the controller are chosen such that the estimated closed loop system is given predetermined poles.

PID-controllers can be implemented and parameterized in many different ways. Different versions are discussed and compared. The PID-controllers are also compared with a general pole placement algorithm. The behaviour of the algorithms is illustrated in simulated examples.

Referat skrivet av
Author

Förslag till ytterligare nyckelord
44T0

Klassifikationssystem och -klass(er)
50T0

Indextermer (ange källa)
52T0

Omfång	Övriga bibliografiska uppgifter
37 pages	56T2
Språk	
English	

Sekretessuppgifter	ISSN	ISBN
60T0	60T4	60T6

Dokumentet kan erhållas från
Department of Automatic Control
Lund Institute of Technology
Box 725, S-220 07 Lund 7, Sweden

Mottagarens uppgifter
62T4

Pris
66T0

DOKUMENTDATABLAD enligt SIS 62 10 12

SIS-DB 1

1. INTRODUCTION

One of the advantages of the wellknown PID-controller (sometimes also called the three mode controller) is that it is a sufficiently flexible controller for many applications. The controller usually has three parameters to be determined. The parameters are in general manually tuned with the process in closed loop. In the literature there exist many schemes for tuning of continuous and discrete time PID-controllers, see for instance Ziegler and Nichols (1942), von Kessler (1958), Lopez, Miller, Smith and Murrill (1967), Smith (1972), and Auslander, Takahashi and Tomizaka (1978).

The tuning procedure is, however, in many cases nontrivial and it can be difficult to find values of the parameters which give a desired closed loop behaviour. Automatic tuning of the controllers is therefore of great interest. Automatic tuning of PID-controllers is discussed e.g. in Davies (1966), Rumold and Speth (1968), Pendelbury, Stedmon and West (1971), and Schilling (1976). The suggested methods are mainly based on optimization theory. The parameters of the controller are chosen in order to minimize a loss function. The drawback of this approach is that the optimization routine only gets one function evaluation at each transient, usually a step in the reference value. The tuning procedure will thus be time consuming.

In this paper the self-tuning approach is used. This means that a discrete time model of the process is estimated recursively. A design method is then applied using the estimated model as if it is the true one i.e. the certainty equivalence hypothesis is used. Self-tuning algorithms have been designed for regulator and servo problems, see Åström, Borisson, Ljung and Wittenmark (1977) and Åström, Westerberg and Wittenmark (1978). The algorithms for the regulator problem are mainly based on minimum variance control. The algorithms for the servo problem use pole placement. These self-tuning algorithms are, however, not exactly of the same structure as conventional PID-controllers. Usually they contain more parameters than the PID-controllers.

The aim of this paper is to start with the conventional PID-structure and use a pole placement algorithm. It will thus be possible to make a direct interpretation in terms of the proportional, reset and derivative parts of the controller. Also two less common structures of PID-controllers are discussed. The paper can be seen as a bridge between conventional PID-structures and more general controllers. The PID-controllers are a subset of the general controllers based on observers and state feedback.

Discrete time versions of PID-controllers are discussed in Section 2. The self-tuning PID-algorithm is described in Section 3. The self-tuning PID-controller also contains parameters which have to be selected. Section 4 contains a discussion of these parameters. In Section 5 the algorithm is compared with a more complex pole placement algorithm described in Åström, Westerberg and Wittenmark (1978). Simulated examples in Section 6 show the advantages and disadvantages of the proposed controller. The applicability of the different self-tuning algorithms is discussed in Section 7. References are found in Section 8.

2. SAMPLED DATA PID-CONTROLLERS

There are many different ways to implement PID-controllers. Different structures have different properties with respect to the behaviour of the closed loop system. Also the possibility to tune the controller is influenced by the chosen structure. We will start with the most common "textbook" structure of a PID-controller. The corresponding sampled data controller will be derived and the relations between the continuous time and the discrete time parameters will be discussed. Different versions of sampled data PID-controllers are then discussed and three structures are chosen for further analysis.

The most common PID-controller in textbooks is defined by the following equations:

$$U(s) = G(s)(Y_r(s) - Y(s))$$

with

$$G(s) = K\left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \alpha T_D s}\right) \qquad (2.1)$$

where $U(s)$, $Y_r(s)$ and $Y(s)$ are the Laplacetransforms of the controller output, the reference signal, and the process output respectively.

The three terms in (2.1) correspond to the proportional (P), the rest or the integral (I), and the derivative (D) parts respectively. The controller contains four parameters (K, T_I. T_D, and α). The parameter α is, however, usually predetermined by the manufacturer and can not be changed by the user. Common values of α are in the interval 0.1 - 0.3.

A discrete time PID-controller can be obtained by making a difference approximation of (2.1). The way chosen here is, however, to derive the sampled data form of (2.1).

The sampled data form of (2.1) can easily be derived. The PI and D parts will be treated separately. The PI-part has the transfer function

$$G_1(s) = K(1 + 1/T_I s).$$

The pulse transfer operator when G_1 is sampled with the sampling time T_0 is given by

$$H_1(q^{-1}) = K\left(1 + \frac{T_0}{T_I}\frac{q^{-1}}{1-q^{-1}}\right) = \frac{K + K(T_0/T_I - 1)\, q^{-1}}{1-q^{-1}} =$$

$$= \frac{\alpha_0 + \alpha_1 q^{-1}}{1-q^{-1}} \triangleq \frac{A(q^{-1})}{I(q^{-1})},$$

where q^{-1} is the backward shift operator. Notice that the sampling will introduce a dealy in the integrator part, i.e. the factor q^{-1}. This delay is sometimes removed when the regulator is implemented. This will, however, only change the interpretation of the A-polynomial.

To get a good approximation of the derivative it is assumed that the signal is linear, i.e. the derivative is constant, between the sampling points. The D-part has the transfer function

$$G_2(s) = K\frac{T_D s}{1 + \alpha T_D s} = \frac{K}{\alpha}\left(1 - \frac{1}{1 + \alpha T_D s}\right).$$

The sampled data pulse transfer operator is

$$H_2(q^{-1}) = \frac{K\,T_D\,(1 - e^{-T_0/\alpha T_D})(1-q^{-1})}{1 - e^{-T_0/\alpha T_D}\,q^{-1}} = \frac{\beta I(q^{-1})}{1 + \gamma q^{-1}} = \frac{B(q^{-1})}{C(q^{-1})}.$$

The sampled data parameters can be expressed in the continuous time parameters and vice versa:

$\alpha_0 = K$	$K = \alpha_0$
$\alpha_1 = K(T_0/T_I - 1)$	$T_I = T_0\alpha_0/(\alpha_0 + \alpha_1)$
$\beta = KT_D(1 - \exp(-T_0/\alpha T_D))/T_0$	$T_D = \beta T_0/(K(1+\gamma))$
$\gamma = -\exp(-T_0/\alpha T_D)$	$\alpha = -K(1+\gamma)/(\beta \ln(-\gamma))$

Notice that the sampled data controller has a continuous time counterpart only if $-1 < \gamma < 0$. When tuning in sampled data PID-controller it can be useful to allow that $|\gamma| < 1$.

The sampled data PID-controller corresponding to (2.1) has the form

$$u(t) = \left(\frac{A(q^{-1})}{I(q^{-1})} + \frac{B(q^{-1})}{C(q^{-1})} \right) (y_r(t) - y(t)), \tag{2.2}$$

where $y(t)$ is the process output, $y_r(t)$ the reference value and $u(t)$ is the controller output. Assume that the system to be controlled is given by

$$y(t) = \frac{q^{-k-1} B(q^{-1})}{A(q^{-1})} u(t), \tag{2.3}$$

where A and B are polynomials in the backward shift operator of order n and n-1 respectively. The closed loop system can be represented by the block diagram in Fig. 2.1, where

$$T(q^{-1}) = AC + BI$$

$$S(q^{-1}) = AC + BI$$

$$R(q^{-1}) = IC.$$

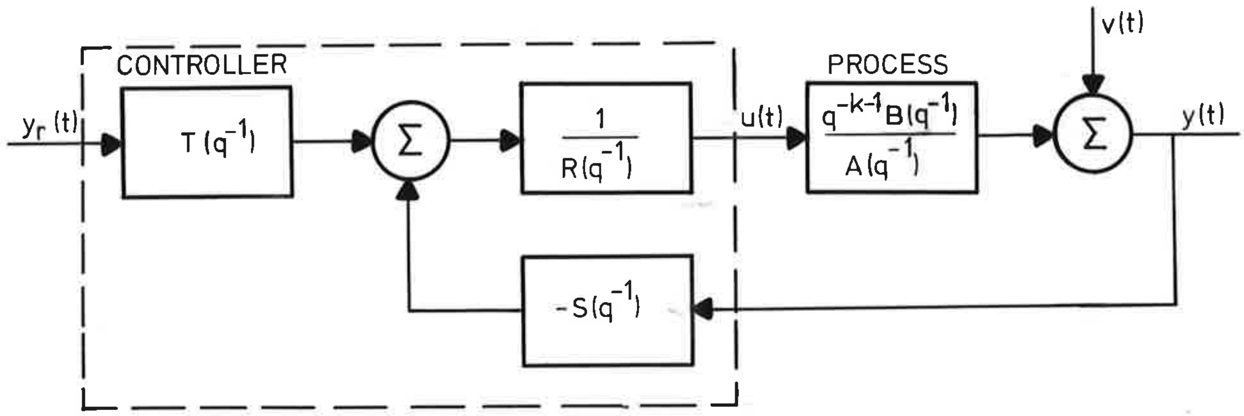

Fig. 2.1 - A general structure for a sampled data controller. The variables y_r, u, v, and y are reference value, control signal, disturbance, and output respectively.

The closed loop system is given by

$$y(t) = \frac{q^{-k-1}BT}{AR + q^{-k-1}BS} y_r(t) + \frac{AR}{AR + q^{-k-1}BS} v(t), \qquad (2.4)$$

where v(t) is a disturbance.

The controller given by (2.2) is only one way to get a controller with three mode actions. Fig. 2.2 shows some other structures which also correspond to PID-controllers. All controllers in Fig. 2.2 can be written in the form

$$R(q^{-1})\ u(t) = T(q^{-1})\ y_r(t) - S(q^{-1})\ y(t). \qquad (2.5)$$

The expressions of the polynomials is different for the different controller. The polynomials T and S are shown in Table 2.1 while $R = IC$ for all structures. All controllers have four parameters to determine. The S-polynomial is of second order

$$S(q^{-1}) = s_0 + s_1 q^{-1} + s_2 q^{-2}.$$

The S- and C-polynomials can be chosen arbitrarily. The parameterization can either be as in Fig. 2.2 or the

Version	$S(q^{-1})$	$T(q^{-1})$
PID 0	$AC + BI$	$AC + BI = S$
PID 1	$AC + BI$	AC
PID 2	$C\alpha_1 + I(\alpha_0 C + B)$	$\alpha_1 C = C \cdot S(1)/C(1)$
PID 3	$\alpha_1 + I(\alpha_0 C + B)$	$\alpha_1 = S(1)$
PID 4	$A(1 + \delta q^{-1})$	$A(1 + \delta q^{-1}) = S$
PID 5	$AC + (1 + \delta q^{-1})\ I$	AC

Table 2.1 - The S and T polynomials of (2.4) for the different PID-controller structures in Fig. 2.1.

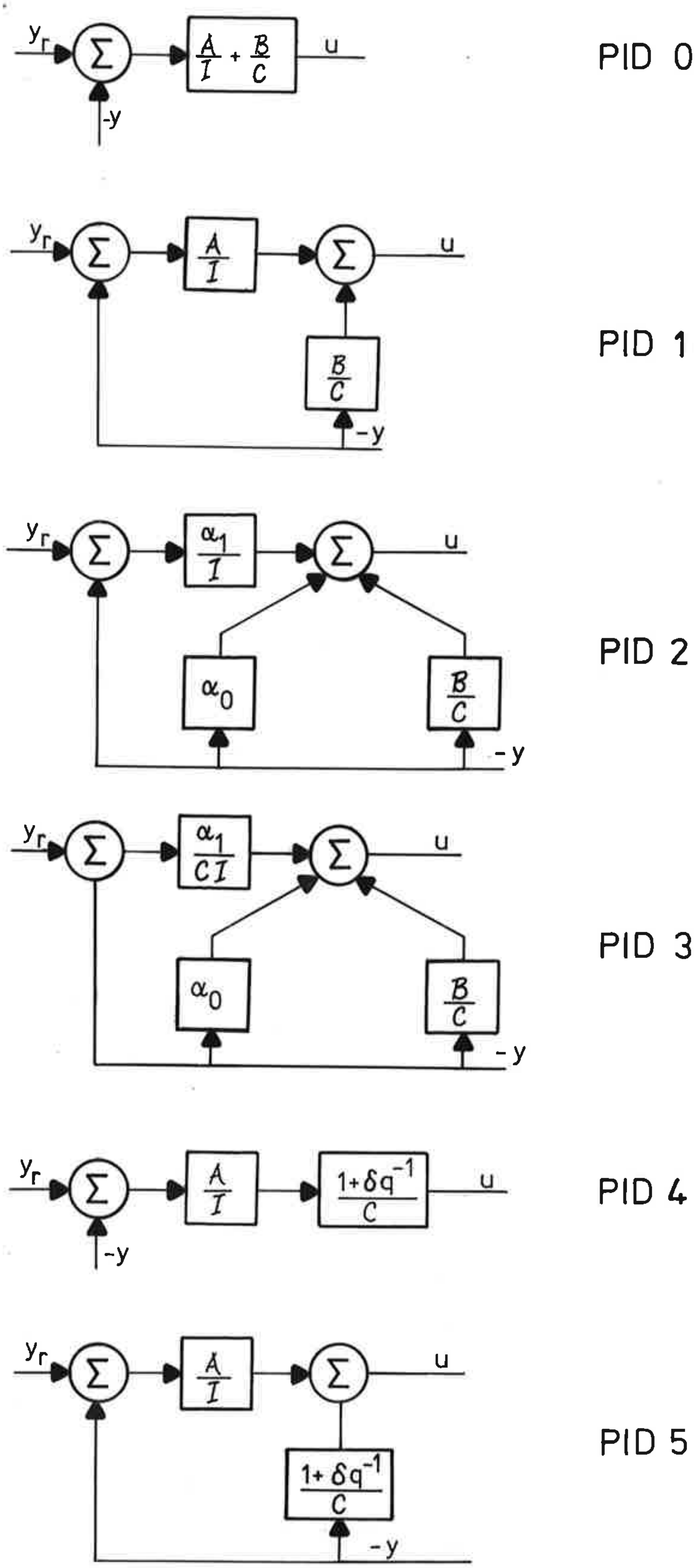

Fig. 2.2 - Six different structures for PID-controllers.

parameters s_0, s_1, s_2, and γ are chosen as the free parameters. The T-polynomial introduces zeros in the closed loop system, see (2.4). This polynomial can not be chosen arbitrarily but is a function of the four chosen parameters.

PID 0 corresponds to (2.2). In PID 1 there is no differentiation of the reference signal. This is one of the most common ways to implement a digital PID-controller. PID 2 and PID 3 are two less wellknown ways to make controllers with three mode action, see Auslander, Takahashi and Tomizuka (1978). The term $1/I$ in PID 2 and PID 3 can be interpreted as an integrator without any delay. The integrator part will thus use the error as input while the proportional and the derivative parts both use -y as input. PID 4 can be interpreted as a PI-controller in cascade with a lead filter. PID 5 is the corresponding controller with no differentiation of the reference signal. It is sometimes claimed that the continuous time versions of PID 4 or PID 5 are easier to tune than the continuous time versions of PID 0 or PID 1.

It is possible to get exactly the same characteristic polynomial of the closed loop system in all cases by choosing the four parameters in the different controllers. The characteristic polynomial is of order $n+k+2$. The zeros of the closed loop system introduced by the T-polynomial may, however, be different for the different controllers. All controllers, except PID 3, will introduce one or two extra zeros compared with the open loop system. The four parameters of the controller will then influence both the poles and the zeros of the closed loop system. It may thus be difficult to find values of the parameters which give the system desired closed loop behaviour. Rules of thumb corresponding to the method of Ziegler and Nichols for tuning PID 0 and PID 1 can be found for instance in Takahashi, Chan and Auslander (1971), and Chin, Corripio and Smith (1973).

A more general controller with integral action can be obtained by using the structure in Fig. 2.1 with general T, S, and R polynomials. The integral action will be obtained if R contains the factor $I(q^{-1}) = 1-q^{-1}$ and if $T(1) = S(1)$. By choosing R and S of sufficiently high order the closed loop system may be given desired poles. The polynomial T can now be more freely chosen. This type of pole placement algorithms are discussed in Åström, Westerberg and Wittenmark (1978). Continuous time controllers of the same structure are discussed in Åström (1976).

In the following we will give an algorithm for automatic tuning of the parameters of PID 1, PID 2, and PID 3. The reason to look at PID 1 is that it is a common way to implement digital PID-controllers. PID 2 and PID 3 are less known but are in some situations better to use than PID 1. The reason is that one and no zero respectively are introduced while PID 1 introduces two zeros in the closed loop transfer function.

3. SELF-TUNING PID-CONTROLLERS

Self-tuning PID-controllers based on the structures PID 1, PID 2, and PID 3 in Fig. 2.2 are discussed in this section. The self-tuning controllers are based on pole placement. It is assumed that the tuning is done in order to solve the servo problem. I.e. the transfer function from y_r to y will be studied. It is possible to write one algorithm that covers all the three cases.

Assume that the process (2.3) is of second order and that there is no time delay in the system, i.e. $n = 2$ and $k = 0$. The process is then defined by the polynomials

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2}$$

and

$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2}.$$

The closed loop system (2.4) is then of fourth order. The PID-controllers PID 1 - 3 have four parameters each. It would then be possible to choose the parameters such that the closed loop system get prescribed poles. The following equation should be satisfied

$$AR + q^{-1}BS = A\mathcal{I}\mathcal{C} + q^{-1}BS = P', \tag{3.1}$$

where the expressions for S are given in Table 2.1 and $P'(q^{-1})$ is a prespecified desired characteristic polynomial. The unknown parameters of the polynomials $\mathcal{C}$ and S can be uniquely determined from (3.1) if $A\mathcal{I}$ and B has no common factors and if $B \neq 0$, see Åström (1976). The parameters in $\mathcal{A}$ and $\mathcal{B}$ can then be determined from the identities:

$$\begin{aligned} &\text{PID 1:} \quad S = \mathcal{A}\mathcal{C} + \mathcal{B}\mathcal{I} \\ &\text{PID 2:} \quad S = \mathcal{C}\alpha_1 + \mathcal{I}(\alpha_0 \mathcal{C} + \mathcal{B}) \\ &\text{PID 3:} \quad S = \alpha_1 + \mathcal{I}(\alpha_0 \mathcal{C} + \mathcal{B}). \end{aligned} \tag{3.2}$$

It is easy to show that the parameters α_0, α_1 and β can be determined if $\gamma \neq -1$. The condition $\gamma = -1$ implies that B/C is a constant, i.e. there is no D-part in the controller. For a second order system it is thus in most cases possible to place the poles of the closed loop system arbitrarily.

Two zeros ($-\gamma$ and $-\alpha_1/\alpha_0$) will be introduced by the PID 1 controller and one zero by PID 2 ($-\gamma$), see (2.4) and Table 2.1. In order to eliminate the effect of the two introduced zeros we may place poles close to the zeros, provided that the poles are placed within the unit circle. The remaining poles of P' can be chosen in order to give the closed loop system a desired response. These poles will be called the control poles. The desired characteristic polynomial could then be written as

$$P'(q^{-1}) = D(q^{-1})\ (1 + k_1\gamma q^{-1})\ (1 + k_2\alpha_1/\alpha_0 q^{-1}) \qquad (3.3)$$

where k_1 and k_2 are chosen in order to eliminate the effect of the introduced zeros. Equation (3.1) is, however, non-linear in the parameters of the controller if P' is chosen as in (3.3).

One way to obtain the parameters is to use an interative algorithm. This can be done in the following way: Introduce the polynomial $P = (1-pq^{-1})$ where p is assumed to be given. Let the desired characteristic polynomial be

$$DP(1 + k_1\gamma q^{-1}) = (1-k_1)\ DP + k_1 DPC. \qquad (3.4)$$

Equation (3.1) can then be written as

$$(AI - k_1 DP)\ C + q^{-1}BS = (1-k_1)\ DP. \qquad (3.5)$$

As P is assumed to be known, Eq. (3.5) can be used to find C and S. The expressions in (3.2) is then used to determine A and B. The value of p can now be compared to the computed value of $-k_2\alpha_1/\alpha_0$. The value of p can now be updated for instance by using

$$p(t+1) = \lambda_1\ p(t) - (1-\lambda_1)\ k_2\ \alpha_1(t)\ /\ \alpha_0(t). \qquad (3.6)$$

If the scheme converges it will converge to a solution such that (3.3) is fulfilled, i.e. $p = -k_2\alpha_1/\alpha_0$. If PID 2 is used p can be arbitrarily fixed. This is done by using $\lambda_1 = 1$ in (3.6). When PID 3 is used there is no need to eliminate any zero and there are four poles that can be arbitrarily placed.

The design scheme described above assumes that the second order model (2.3) is known. In practice the process is unknown and has to be estimated. The estimation can be done from input-output data using for instance the recursive least squares method or a recursive maximum likelihood method. The parameter estimates are then updated at each sampling time.

An algorithm for a self-tuning PID-controller can be described by the following steps. Steps 1 - 3 are repeated at each sampling time.

A self-tuning PID-controller algorithm

Data: The D-polynomial, k_1, k_2, and λ_1 are assumed to be given.

Step 0: (Initialization) Give initial values to the parameter estimator and to p in (3.6).

Step 1: (Estimation) Update the parameters in a second order model of the process using for instance the method of least squares.

Step 2: (Parameter determination) Solve (3.5) for C and S and solve (3.2) for A and B. Update p using (3.6).

Step 3: (Control) Determine the control signal using (2.5) with the parameters obtained in Step 2.

The control law may be implemented as a state space representation of (2.5). A minimal representation of (2.5) can be written as the second order system, compare Andersson and Åström (1978):

$$\begin{cases} x(t+1) = \begin{pmatrix} 1-\gamma & 1 \\ \gamma & 0 \end{pmatrix} x(t) + \begin{pmatrix} t_1 - t_0(\gamma-1) \\ t_2 + t_0\gamma \end{pmatrix} y_r(t) - \begin{pmatrix} s_1 - s_0(\gamma-1) \\ s_2 + s_0\gamma \end{pmatrix} y(t) \\ u(t) = x_1(t) + t_0\, y_r(t) - s_0\, y(t) \end{cases} \quad (3.7)$$

Notice that (3.2) does not need to be solved if PID 2 or PID 3 and the controller implementation (3.7) are used. The T polynomial can be obtained directly from S and C, as indicated in Table 2.1.

It is possible to obtain a self-tuning PI-controller in an analogous way. The estimated model now has to be of first order. The closed loop system will be of second order and the poles can be placed by choosing the two parameters in the controller. As before different structures of the controller may be used. Structures corresponding to PID 1, PID 2 (or PID 3) will be obtained by letting $B = 0$ and $C = 1$ in Fig. 2.2. In the first case one zero will be introduced. The effect of the zero may be eliminated using a dipole parameter. In the second case no zero will be introduced and the two degrees of freedom can be used to place two poles arbitrarily.

The self-tuning algorithm contains the following parameters that have to be determined in advance:

- The "dipole parameters", k_1 and k_2
- The control poles of the closed loop system $D(q^{-1}) = 1 + d_1 q^{-1} + d_2 q^{-2}$
- The convergence rate parameter, λ_1
- The initial values of the estimator
- The sampling time, T_0

It is found that these parameters are easier and less crucial to determine than to directly determine the controller parameters. The choice of the parameters is discussed in Section 4.

The process to be controlled, (2.3), may not be of second order. By using a recursive parameter estimator the process will be approximated by a second order model. If the process is not too complex the second order model will catch the main features of the process. There may be problems if the process (2.3) contains time delays, i.e. $k \neq 0$. The algorithm can, however, be used as long as the estimated B-polynomial is not identically zero. If the process has long timedelays it may still be possible to find a second order model. The model will, however, become a worse and worse approximation of the true system as the timedelay is increased.

4. CHOICE OF THE PARAMETERS IN THE ALGORITHM

As mentioned in Section 3 there are several parameters to choose in the self-tuning PID-algorithm. These parameters are, however, easier to choose than to directly determine the parameters of the controller. The parameters are d_1, d_2, k_1, k_2, λ_1, and the initial values in the estimator. When using the method of least squares the parameters are the initial values of the parameters and of the covariance matrix. Usually it is also convenient to include an exponential discounting factor in the estimator. The discounting factor makes it possible to follow time-varying parameters. Finally the sampling time, T_0, also will influence the behaviour of the closed loop system.

The choice of the parameters in the least squares estimator used in an adaptive controller is discussed in Wittenmark (1973). The conclusions are that the least squares estimator is quite insensitive to the choice of the initial values. The rest of the section will be devoted to discuss the parameters in the controller part.

The dipole parameters

The dipole parameters, k_1 and k_2, are introduced in order to eliminate the effect of the zeros introduced by the polynomials A and C. The zeros will not influence the output if $k_1 = k_2 = 1$. Values close to one will give dipoles which usually not change the closed loop behaviour very much.

By analysing a first order system it is easy to understand how the dipole will change the step response. Consider the first order system

$$y(t) = \frac{1-a}{1-b} \cdot \frac{1-bq^{-1}}{1-aq^{-1}} u(t) = \frac{1-a}{1-b}\left(1+\frac{(a-b)q^{-1}}{1-aq^{-1}}\right) u(t). \tag{4.1}$$

The steady state gain is equal to one. Depending on the values of a and b the step response will look quite different. Step responses for four different pole-zero patterns are shown in Fig. 4.1. Cases a) and b) correspond to lag networks and cases c) and d) to lead networks. In order to avoid too large overshoots it is desirable to choose pole--zero patterns corresponding to lag networks. Simulations of the self-tuning PID-controllers have shown that the parameter γ often will be positive. Also depending on the system to be controlled and on the specifications it may happen that $|\gamma| > 1$. In order to keep down the oscillations it is thus desirable to have a quite small value in k_1. The second introduced zero, $-\alpha_1/\alpha_0$, is often positive and less than one. The parameter k_2 can then be chosen close to one. In order to eliminate a disturbance the value of k_2 should be small, see Example 6.1 in Section 6.

The control poles

The control poles are specified through the polynomial $D(q^{-1}) = 1 + d_1q^{-1} + d_2q^{-2}$. The specifications for a control loop are often given as rise and solution times and overshoot. These specifications can often be transformed into specifications of damping and resonance frequency of a second order continuous time system. Assume that the second order system

$$G(s) = \frac{\omega_0^{\,2}}{s^2 + 2\xi\omega_0 s + \omega_0^{\,2}} \tag{4.2}$$

is given. The rise time for this system is:

$$T_r = \omega_0^{-1} \exp(\phi / \tan \phi)$$

where $\cos \phi = \xi$. The 5 % solution time is

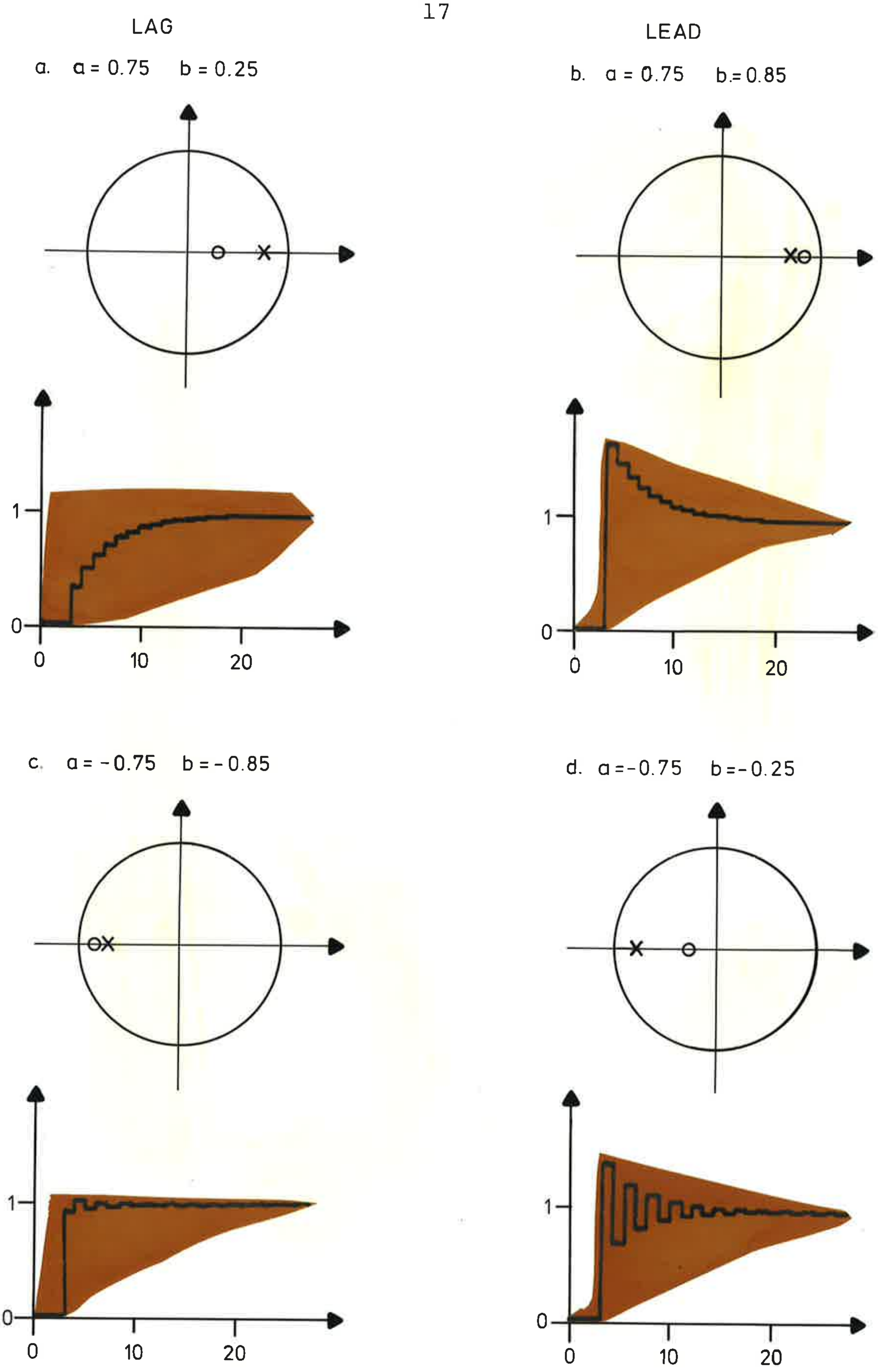

Fig. 4.1 - Step responses of the system (4.1) for different pole-zero patterns. The step is applied at t = 2.

$$T_s \approx 3/(\omega_0 \xi) \qquad 0 \leq \xi < 0.9.$$

The overshoot is

$$M = \exp(-\pi\xi / \sqrt{1-\xi^2}) \tag{4.3}$$

From these expressions it is possible to determine ξ and ω_0 which make it possible to fulfill the specifications.

Sampling of (4.2) gives a discrete time characteristic polynomial which is a function of ξ, ω_0, and the sampling time, T_0. The discrete time system corresponding to (4.2) has the characteristic polynomial

$$1 + d_1 q^{-1} + d_2 q^{-2}$$

with

$$d_1 = -2\, e^{-\xi\omega_0 T_0} \cos(\omega_0 T_0 \sqrt{1-\xi^2})$$

$$d_2 = e^{-2\xi\omega_0 T_0}.$$

By choosing ω_0 and ξ it is possible to determine $D(q^{-1})$. It is, however, not only D that determines the closed loop behaviour, but also the B-polynomial and the dipoles, compare (2.4). It may then be necessary to modify ω_0 and ξ in order to be able to satisfy the specifications of the closed loop system. In many cases it is necessary to increase the damping compared with the value obtained from (4.3).

The convergence rate parameter

The convergence rate parameter, λ_1, (only used in PID 1) determines how fast the pole p will approach $-k_2\alpha_1/\alpha_0$. Equation (3.6) is a first order system with the pole λ_1. The influence of the initial value will be decreased by a factor 10 after n steps where

$$n = \frac{\ln 0.1}{\ln \lambda_1} \approx \frac{2}{1-\lambda_1}.$$

Suitable values of λ_1 can be in the interval 0.4 - 0.7. It has not yet been possible to analyse when the iterative scheme converges. In some simulated example the scheme has diverged.

Sampling time

The sampling time will of course influence the behaviour of the closed loop system. In general the sampling time is quite short in conventional discrete time PID-controllers. One reason is that the parameters of the continuous time controller then can be used directly. Also the sampling time must be quite short in order to get a good approximation of the derivative of the output.

When using the pole placement controller described in this paper there is quite a large freedom to choose the sampling time. One should try to use as long sampling time as possible. As a rule of thumb one should allow about 3 - 10 samples at each transient of the closed loop system.

5. COMPARISON WITH A GENERAL POLE PLACEMENT ALGORITHM

General self-tuning pole placement algorithms are described in Åström, Westerberg and Wittenmark (1978). The controllers have the same structure as in (2.5) and in Fig. 2.1. This controller structure can be interpreted as a polynomial representation of an observer and a feedback from the observed states, see Åström (1976). The polynomials are then allowed to be of higher degree than if the self-tuning PID-controller is used. Further the T polynomial is not dependent of the R and S polynomials as in Table 2.1. The T polynomial is instead chosen as $T = A_0 B_m^+$, where A_0 is a stable polynomial corresponding to the desired observer poles. B_m^+ is used to introduce new zeros in the closed loop system. The R and S polynomials are obtained from the identity

$$AR + q^{-k-1} BS = PB_1 A_0 \tag{5.1}$$

where B_1 are the open loop zeros that are going to be cancelled. P is the desired closed loop characteristic polynomial. Notice that A_0 is cancelled in the transfer function from the reference value to the output. The closed loop system is

$$y(t) = \frac{q^{-k-1} TB}{AR + q^{-k-1} BS} y_r(t) = \frac{q^{-k-1} B_m^+ B_2}{P} y_r(t),$$

where $B_2 = B/B_1$ are the open loop zeros that also are closed loop zeros. Correct steady state gain can be obtained by normalizing the polynomials. The steady state gain may, however, not be correct until the estimation has converged. An integrator can be introduced by replacing R by RI in (5.1). Correct steady state value will then be obtained if $T(1) = S(1)$.

The general pole placement algorithms have the advantage that all the poles of the system can be arbitrarily placed and no extra zeros need to be introduced. It is thus easier to predict the behaviour of the closed loop system than if the self-tuning PID-controller, especially PID 1, is used. Further the process can be allowed to have long time delays. The general algorithm is more complex, more parameters have to be estimated and a higher order system of equations has to be solved. In many cases it is, however, possible to use a low order model in the estimation and the determination of the control law. Some examples and analysis are found in Åström (1978).

The self-tuning PID-controller presented here can be regarded as a link to the more general controllers. PID 3 is the same as the general controller with an estimated second order model and with the observer poles in the origin. Further an integrator but no extra zeros are introduced. The PID-structure is in many cases sufficient to use and the self-tuning PID-controller can be used to tune already installed digital PID-controllers. If starting from scratch it could, however, be better to use the general algorithm based on identification of a low order model. This is especially true if the process contains time delays. The properties of the two algorithms are compared in Section 6.

6. SIMULATED EXAMPLES

The properties of the self-tuning PID-controller have been investigated through simulations. A program for simulation of different types of adaptive controllers has been used, Gustavsson (1978). This program is based on the simulation package SIMNON, Elmqvist (1975). Continuous time processes have been controlled by the sampled data PID-controller. Three examples of different complexity are given. If not specified it is assumed that the initial estimates of a_1 and a_2 are zero and those of b_1 and b_2 are 0.1, the initial covariance matrix $P(0) = 100 \times I$ (I is the unit matrix) and the exponential forgetting factor $\lambda = 1$.

Example 6.1

The process is

$$G(s) = \frac{1}{(1+s)(1+0.5\,s)}. \tag{6.1}$$

This process fits the assumptions given in Section 3 for the self-tuning PID-controller and all closed loop poles can be arbitrarily placed. Assume that the specifications are that the solution time is about 4 seconds and that the damping is about 0.7. A sampling time of $T_0 = 0.5$ was used in the simulations. Figure 6.1 shows the output, the reference signal, the control signal and the pole p, see (3.6), when the controller structure PID 1 was used. The initial value of the pole p was 0, further $k_1 = 0$ and $k_2 = 1$. The control signal was limited to ±3. Already after the second step change the closed loop system behaved satisfactory. Fig. 6.2 shows the result when PID 3 was used with $k_1 = 0$. PID 2 gave an almost identical result and is not shown. All three structures gave similar results but it is easier to use PID 2 and PID 3 since the parameter k_2 does not need to be chosen.

The effect of the dipole parameters k_1 and k_2 were investigated for PID 1 using the true parameter values of the process,

$$a_1(t) = -0.9744 \qquad a_2(t) = 0.2231$$
$$b_1(t) = 0.1548 \qquad b_2(t) = 0.0939.$$

These values are the values of the parameters in the process when (6.1) is sampled with $T_0 = 0.5$.

Table 6.1 shows k_1, p, α_0, α_1, β, and γ when $k_2 = 1$. The corresponding step responses are very similar as long as k_1 is smaller than about 1. The controller is more sensitive to changes in the parameter k_2. Fig. 6.3 shows the step responses for different values of k_2. The parameter k_1 was 0.

The explicit self-tuning algorithm, STURP2, from Åström, Westerberg and Wittenmark (1978) gave almost identical results as PID 3.

The effect of a disturbance at the input of the system was also investigated. Fig. 6.4 shows the output when the reference value is a square wave with amplitude ±1 and the disturbance is a square wave of amplitude ±0.2. The regulator PID 3 was used and it gave better results than PID 1 and PID 2. There is, however, a problem with the disturbance since it will also influence the estimation part of the regulator. If the disturbance is large compared with the reference value then the closed loop system may behave badly. This can be avoided by also identifying the disturbance.

The regulator is quite sensitive to time delays in the process. It was possible to control the system fairly well as long as the time delay was one or two samples. One way to overcome the effect of the time delay was to increase the sampling time.

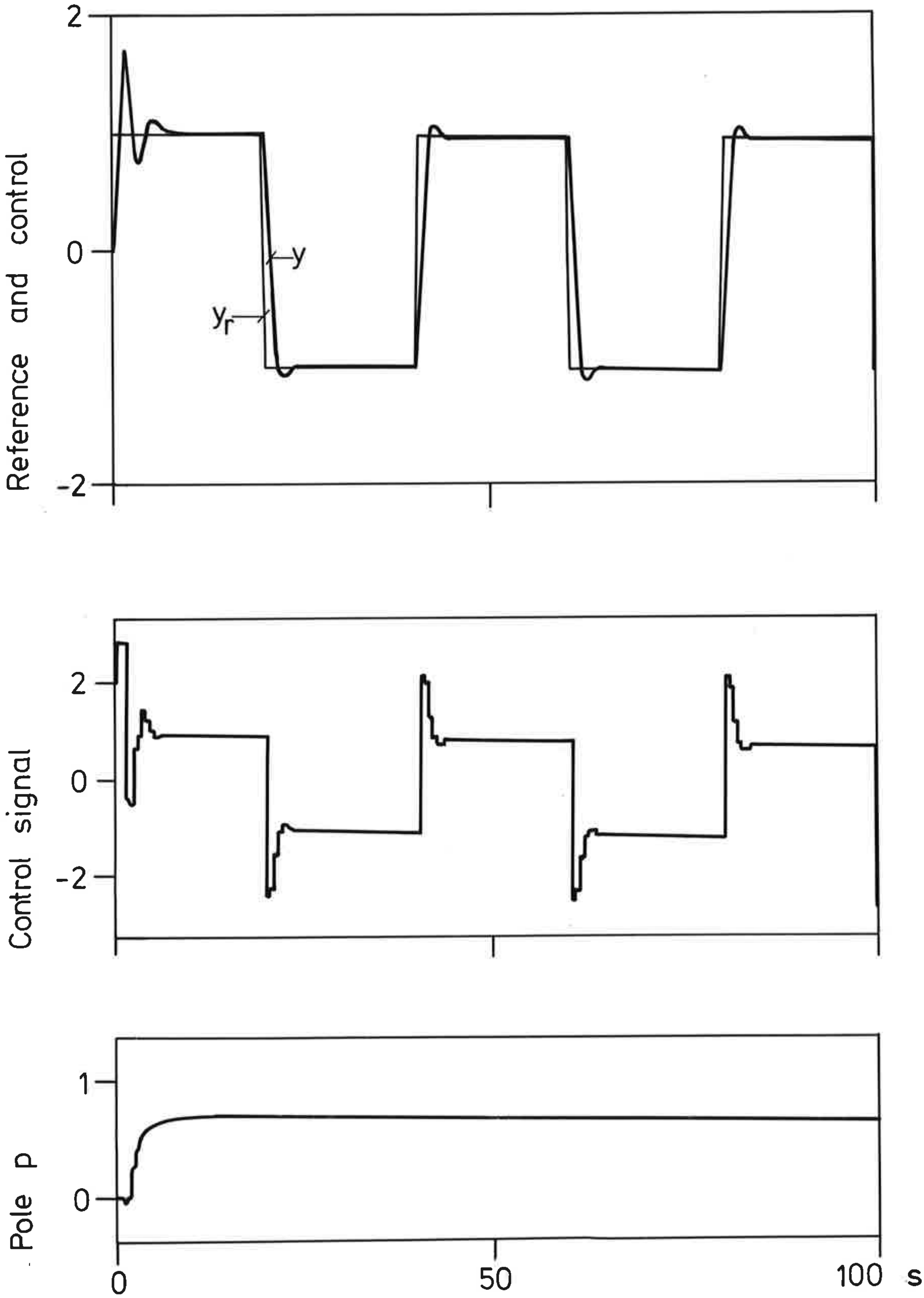

Fig. 6.1 - Output y, reference signal y_r, input u, and pole p when the process (6.1) is controlled with the self-tuning PID-controller PID 1.

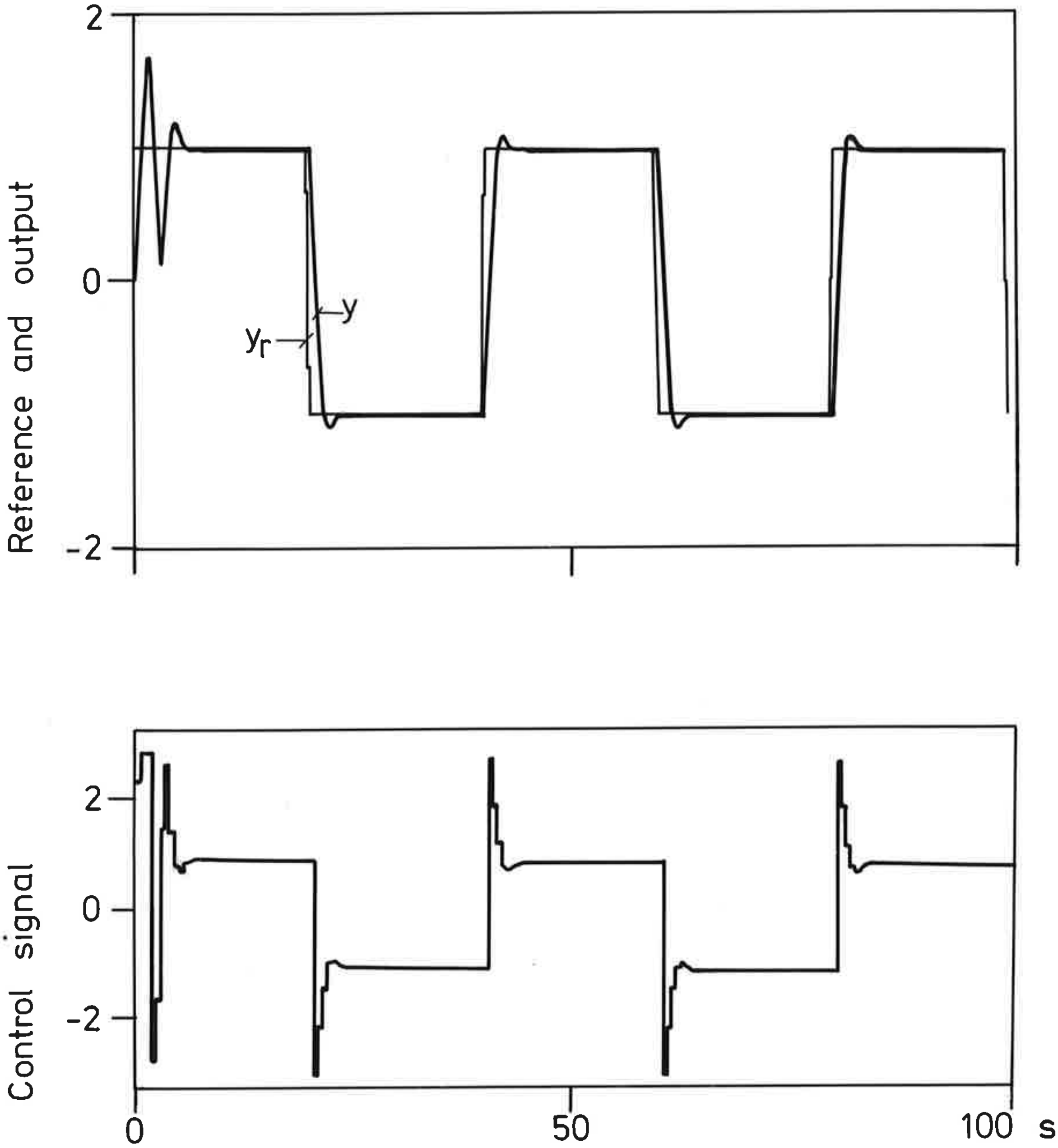

Fig. 6.2 - Output y, reference value y_r, and input u when the process (6.1) is controlled with the self-tuning PID-controller PID 3.

k_1	p	α_0	α_1	β_1	γ
0.98	0.73	1.98	-1.45	1.11	0.57
0.9	0.73	1.93	-1.41	0.96	0.46
0.5	0.72	1.80	-1.30	0.67	0.24
0	0.71	1.73	-1.23	0.56	0.16
-1	0.71	1.67	-1.19	0.49	0.09
-2	0.91	1.64	-1.16	0.46	0.06

Table 6.1 - The effect of the dipole parameter k_1 on the regulator parameters when $k_2 = 1$ and when PID 1 is used.

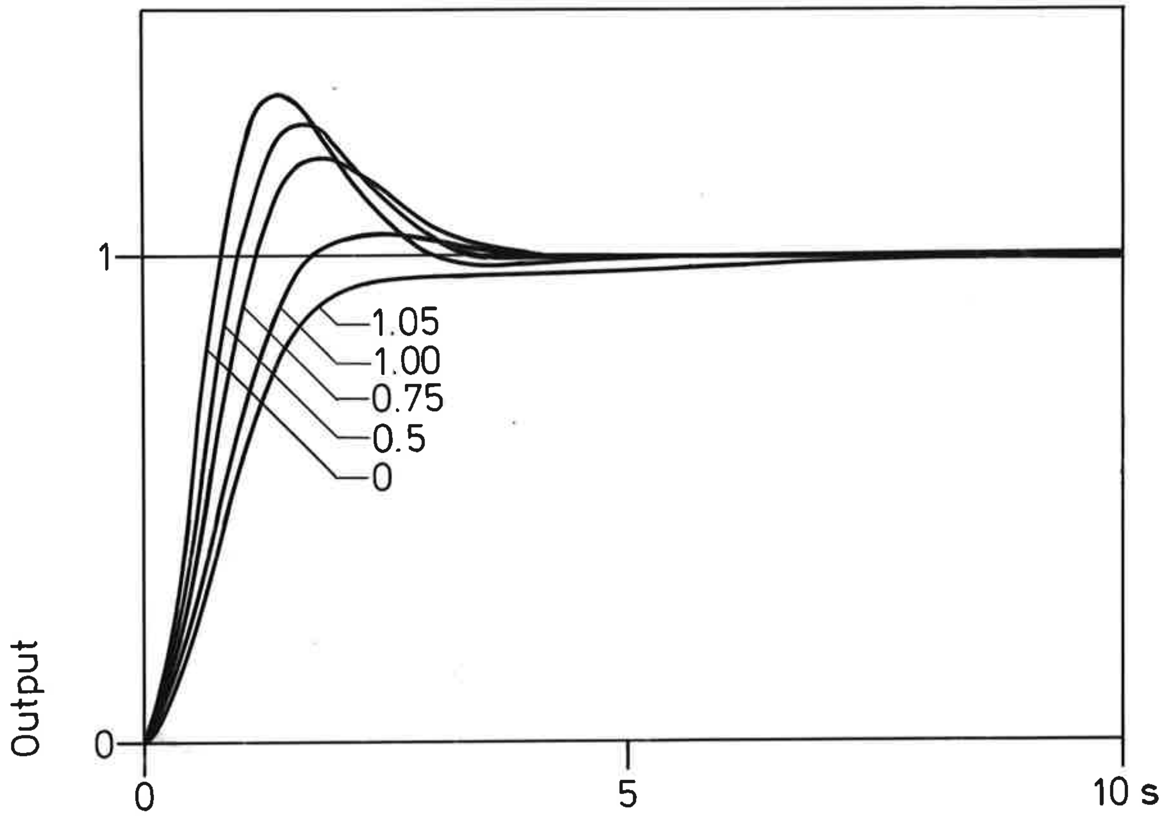

Fig. 6.3 - The effect of the dipole parameter k_2 when the process (6.1) is controlled using true parameter estimates when $k_1 = 0$ and with the controller structure PID 1.

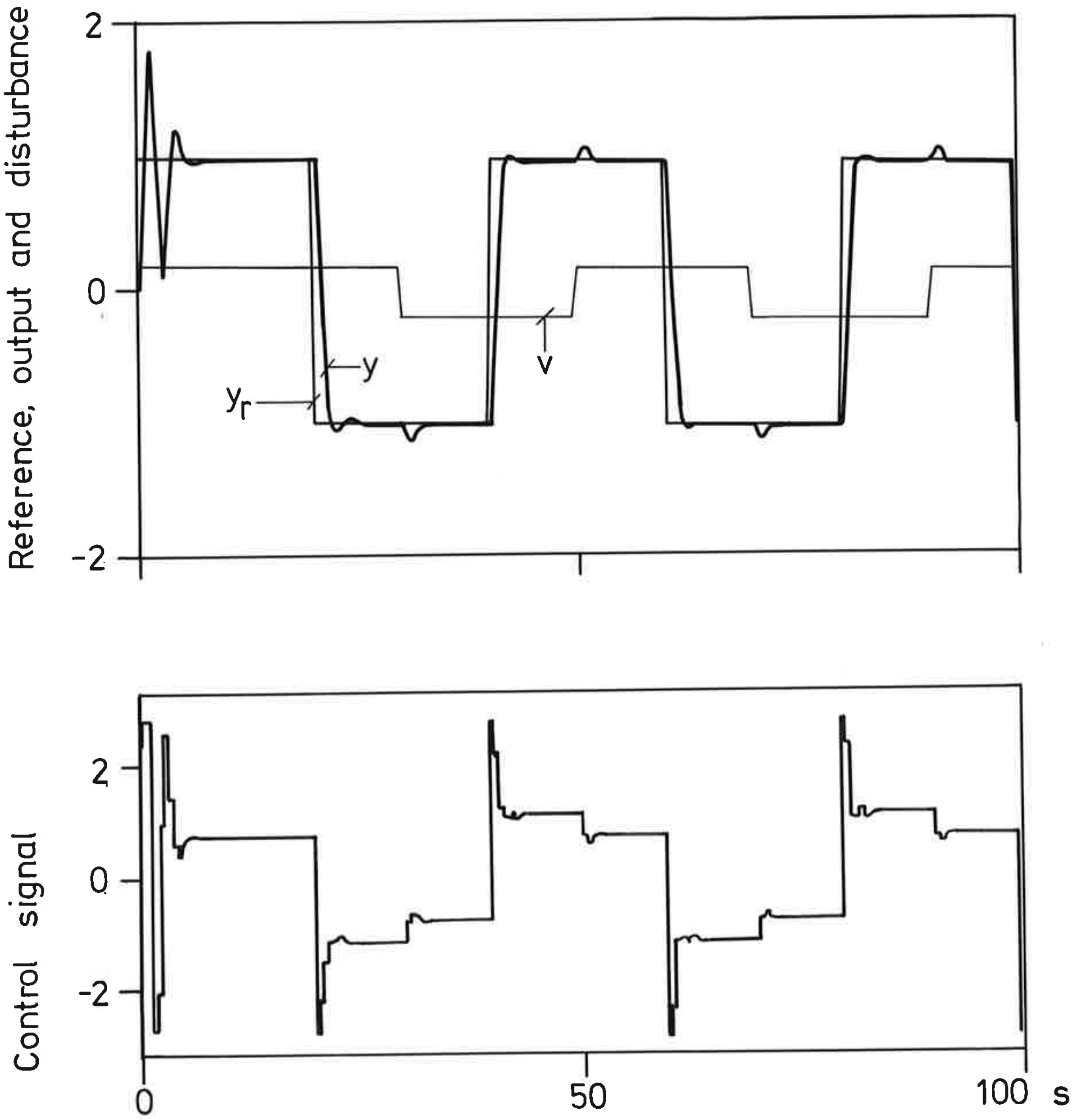

Fig. 6.4 - The output y, the reference value y_r, the disturbance v, and the control signal u, when the process (6.1) is controlled with PID 3.

Example 6.2

A third time constant of 5 seconds was introduced in (6.1). The transfer function now is:

$$G(s) = \frac{1}{(1+s)(1+0.5\,s)(1+5s)} \,. \qquad (6.2)$$

The specifications are changed to a solution time of 10 seconds and it is assumed that the closed loop system should

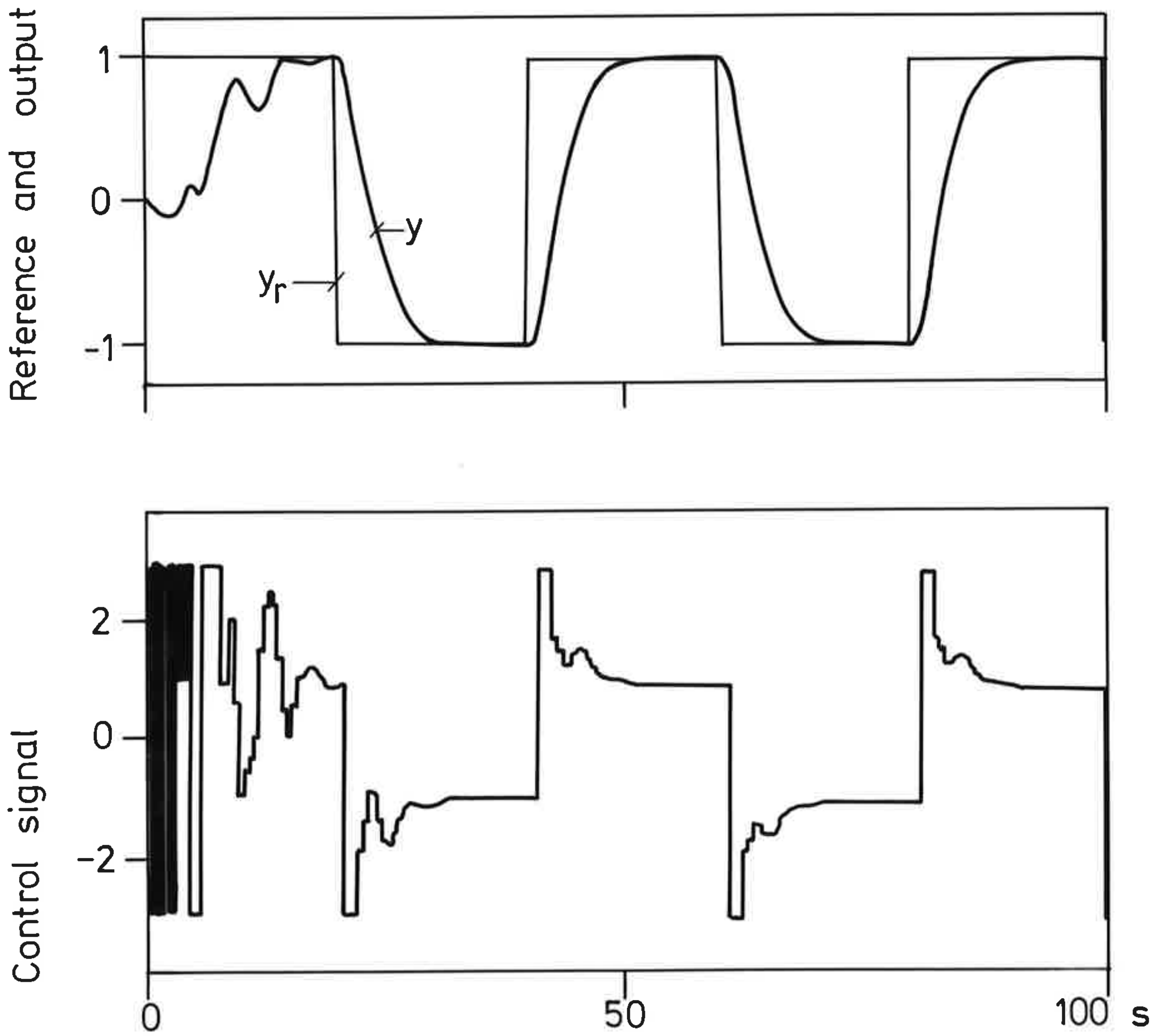

Fig. 6.5 - The output y, the reference signal y_r, and the control signal u when the process (6.2) is controlled with PID 1.

have an exponential response with none or very small overshoot. The sampling time is still assumed to be $T_0 = 0.5$ seconds.

PID 1, PID 2, and PID 3 gave similar results. Figure 6.5 shows the output, the reference value, and the control signal when PID 1 is used. The system does not fulfill the assumption that the controlled system is of second order. Despite of this it is possible to get good control based on a second order model. The sampling time $T_0 = 0.5$ is sufficiently small to reveal all the time constants of the system.

The control signal will become oscillatory if the sampling time is descreased for instance to $T_0 = 0.25$ while a good response can be obtained even with a sampling time of $T_0 = 2$.

Similar results as in Fig. 6.5 were obtained when STURP2 was used with a third or second order model in the estimation. The STURP2 algorithm is, however, less sensitive to time delays, known or unknown, in the process.

Example 6.3

Figure 6.6 shows a block diagram of control system for speed control of a dc motor. The model is a fairly realistic linear model for a mid-size motor. All blocks in the diagram have unit gain. This implies that all signals can be regarded as deviations in procent from a nominal working point. The specifications are a solution time of 300 ms after a step response in the reference value of the speed, n_r. The damping should be about 0.7. Due to the long solution time compared with the time constants the sampling time was chosen to T_0 = 50 ms.

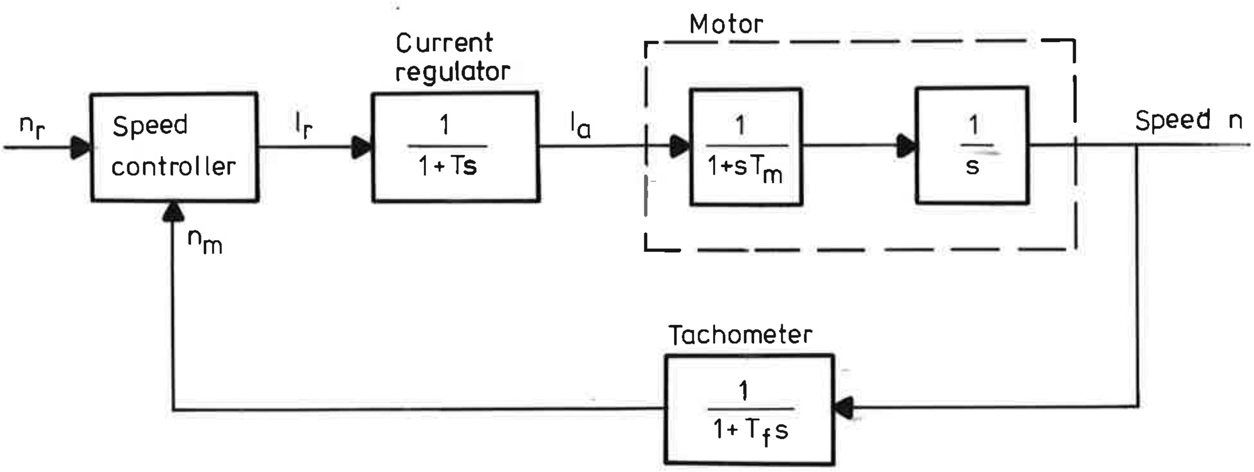

Fig. 6.6 - Block diagram for speed control of a dc motor. The time constants in the model are $T = 10$ ms, $T_m = 5$ ms, and $T_f = 20$ ms.

In this example PID 3 gave a much better response than PID 1. This was due to the fact that it was not possible to eliminate the introduced zeros since one was outside the unit circle. This resulted in a large overshoot when PID 1 was used.

Figure 6.7 shows the behaviour of the closed loop system when the controller PID 3 was used. As in the other examples the closed loop system had a good behaviour after a couple of transients.

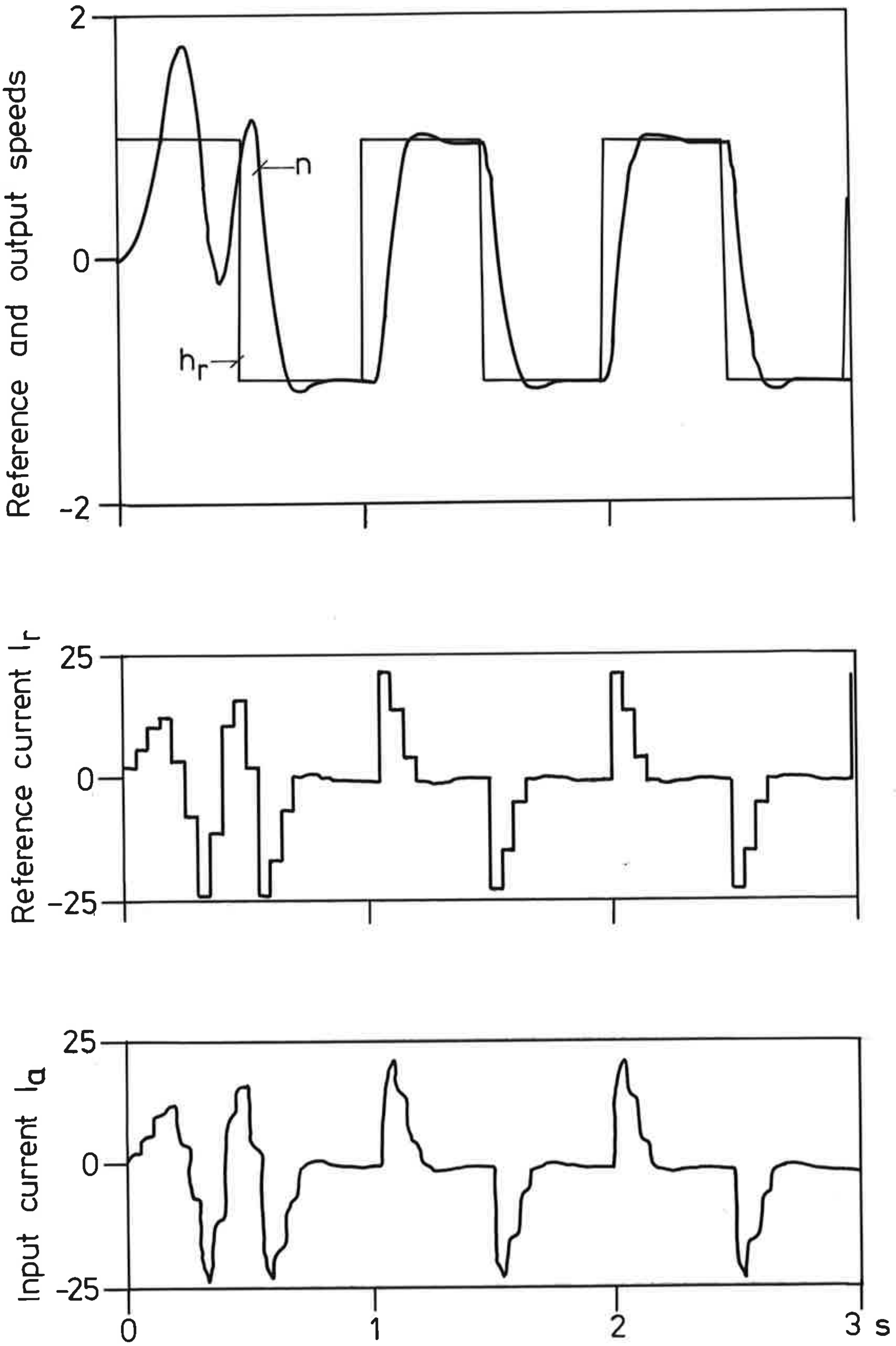

Fig. 6.7 - The speed n, the reference value n_r, the reference current I_r, and the output of the current regulator I_a, when the self-tuning PID-controller PID 3 was used to control the process in Fig. 6.6.

7. DISCUSSION

The paper has described a way to make automatic tuning of PID-controllers. The structure PID 1 is the structure that is used in many applications while PID 2 and PID 3 are two other structures with PID-features. A standard PID-controller has three parameters that usually are manually tuned with the system in closed loop. In this paper a fourth parameter, the filter constant in the D-part, has also been used as a free parameter used in the tuning. Using four parameters it is possible to arbitrarily place the poles of the closed loop system provided the process is of second order and does not include any time-delay. This has been used in the self-tuning algorithms to give the closed loop system a desired behaviour. The structure PID 1 has the disadvantage that two zeros are introduced which may give undesired overshoots in the step responses. PID 2 and PID 3 do not have this drawback and are thus easier to use.

The self-tuning PID-controllers do also contain parameters which have to be chosen. These parameters are, however, believed to be easier to choose. The self-tuning algorithms often have a very good performance after only a couple of transients. Also it is possible to control systems which do not fulfill the assumptions concerning order and time delay.

However, the self-tuning PID-controllers will not make any miracles. It can only behave as a well tuned PID-controller. It is thus only possible to use the self-tuning PID-controller on the same type of processes as the conventional PID-controllers can be used on. The advantage is that the manual tuning is eliminated. The self-tuning PID-controller can be seen as a bridge between conventional PID-controllers and more general pole placement adaptive controllers.

8. ACKNOWLEDGEMENT

This work has been stimulated through many discussions with collegues at the Department of Automatic Control. Especially I want to thank Leif Andersson, Carl Fredrik Mannerfelt, Ivar Gustavsson, Per Hagander, and Karl Johan Åström.

9. REFERENCES

Andersson L, Åström K J (1978): An interactive MISO regulator. Dept of Automatic Control, Lund Institute of Technology, Sweden, CODEN: LUTFD2/(TFRT-7154)/1-034/(1978).

Åström K J (1976): Reglerteori. Almqwist & Wiksell.

Åström K J (1978): Self-tuning control of a fixed bed chemical reactor system. Dept of Automatic Control, Lund Institute of Technology, Sweden, CODEN: LUTFD2/(TFRT-3151)/1-066/(1978).

Åström K J, Borisson U, Ljung L, Wittenmark B (1977): Theory and applications of self-tuning regulators. Automatica 13, 457-476.

Åström K J, Westerberg B, Wittenmark B (1978): Self-tuning controllers based on pole-placement design. Dept of Automatic Control, Lund Institute of Technology, Sweden, CODEN: LUTFD2/(TFRT-3148)/1-052/(1978).

Auslander D M, Takahashi Y, Tomizaka M (1978): Direct digital process control: Practice and algorithms for microprocessor applications. Proc IEEE 66, 199-208.

Chin K C, Corripio A B, Smith C L (1973): Digital control algorithms: Part I Dahlin algorithm, Part II Kalman algorithms, Part III Tunig PI- and PID-controllers. Instruments and Control Systems, Oct 57-59, Nov 55-58, Dec 41-43.

Davies W D T (1966): A self-optimizing proportional plus integral controller. Control 10, 594-598.

Elmqvist H (1975): SIMNON - User's manual. Dept of Automatic Control, Lund Institute of Technology, Sweden. Report TFRT-3091.

Gustavsson I (1978): User's guide for a program package for simulation of self-tuning regulators. Dept of Automatic Control, Lund Institute of Technology, Sweden, CODEN: LUTFD2/(TFRT-7149)/1-076/(1978).

Kessler von C (1958): Das Symmetrische Optimum, Teil I, II. Regelungstechnik 6, 395-400, 432-436.

Lopez A M, Miller J A, Smith C L, Murrill P W (1967): Tuning controllers with error integral criteria. Instrumentation Technology 14, No 11, 57-62.