An Adaptive Pole Placement Controller Based on Pole-Zero Parameterization

Wittenmark, Björn; Evans, R. J.

1987

AN ADAPTIVE POLE PLACEMENT CONTROLLER BASED ON POLE-ZERO PARAMETERIZATION

B. Wittenmark*

R.J. Evans†

ABSTRACT

The pole placement design principle provides a good way of determining a closed loop system with desired properties. The method has been used in adaptive contexts for many years. A drawback of the method is, however, that the polynomial formulation may lead to robustness problems because of the sensitivity with respect to the coefficients of the polynomials. Further, unless all process zeros are cancelled a set of linear equations has to be solved at each step in time. In this paper an attempt is made to avoid these two problems. The sensitivity of the polynomial formulation is avoided by parameterizing the model in the poles and zeros, which are estimated on-line. With the plant expressed in this form it is possible to parameterize the controller such that the solution of the Diophantine equation corresponds to the solution of a triangular system of equations. The controller parameters can thus be computed recursively without extensive computations.

Keywords: Adaptive control, Model parameterization, Sensitivity

* Department of Automatic Control, Lund Institute of Technology
Box 118, S-22100 Lund Sweden (for correspondence).

† Department of Electrical and Computer Engineering,
University of Newcastle, N.S.W. 2308, Australia.

1. INTRODUCTION

Many different schemes for adaptive control have been suggested in the literature. Two good survey papers are Åström (1983) and Seborg et al (1986). Most adaptive controllers can be regarded as a parameter estimator combined with a design procedure. The parameters are updated regularily, mostly at each sampling interval. The design problem is also resolved as soon as new parameters are obtained. To simplify computation, the models are sometimes reparameterized such that the new model contains the regulator parameters explicity. Such approaches are called direct adaptive control schemes. The advantage is a simplification of the design step. The disadvantage of these methods is that they are usually restricted to systems with a stable inverse i.e. minimum phase systems, because the zeros of the open loop system are cancelled in the design. See for instance Goodwin and Mayne (1987) for a discussion of continuous time model reference adaptive control. The pole placement design principle is an indirect method based on the solution of a Diophantine equation, see Åström and Wittenmark (1984). The solution of the Diophantine equation is essentially the same as solving a set of linear equations. The numerical properties of the calculations are highly dependent on the order of the system and the location of the open loop poles and zeros. If there are multiple poles or zeros close to the unit circle (in the discrete time case) then the solution is sensitive to changes in the estimated parameters, which leads to robustness problems.

In this paper an attempt is made to avoid the sensitivity problem and the computational burden of general pole placement design. The approach is to introduce a new parameterization of the model of the process. The model is parameterized into poles and zeros instead of a shift form model. This implies that the estimated model is no longer linear in the parameters, which up till now has been regarded as an almost essential property. The pole-zero parameterization will of course complicate the estimation part of the algorithm, but it is argued in the paper that there will be a total gain because of the simplification in the design step. With the plant parameterized in the poles and zeros it is possible to make a parameterization of the controller such that the solution of the Diophantine equation corresponds to the solution of a linear triangular system of equations. The controller parameters can thus be computed recursively without extensive computations.

The paper is organized in the following way. Section 2 contains the formulation of the problem. The estimation of poles and zeros is discussed in Section 3. The design problem for the new parameterization is solved in Section 4. Finally some properties of the new adaptive algorithm are discussed in Section 5.

Acknowledgements

The report was written during the first author's visit to Australia. The visit has been possible through support by Australian Research Grant Scheme, Swedish Board of Technical Development (STU), Troedssons Forsningsfond and Rydins Stiftelse. The support is gratefully acknowledged.

2. PROBLEM FORMULATION

A new adaptive design procedure will be presented for discrete time single-input single-output systems described by the input-output model

$$y(t) = \frac{B(q)}{A(q)} u(t-d) + \frac{C(q)}{A(q)} e(t) = \frac{Z(q)}{P(q)} u(t-d) + \frac{C(q)}{P(q)} e(t) \qquad (2.1)$$

where y and u are the output and input respectively. A, B, Z, and P are polynomials in the forward shift operator q, i.e.

$$A(q) = q^n + a_1 q^{n-1} + \ldots + a_n$$

$$B(q) = b_0 q^n + b_1 q^{n-1} + \ldots + b_n$$

$$Z(q) = z_0 (q-z_1)(q-z_2)\ldots(q-z_n)$$

$$P(q) = (q-p_1)(q-p_2)\ldots(q-p_n)$$

Further $\{e(t)\}$ is a sequence of white noise random variables. We will refer to $B(q)/A(q)$ as the shift model and $Z(q)/P(q)$ as the pole-zero model or the factorized model. It is assumed that $B(q)$ $(Z(q))$ and $A(q)$ $(P(q))$ do not have any common factors. Furthermore, $d \geq 1$ and $C(q)$ is assumed to have all its roots inside the unit circle.

Remark 2.1.

In the model (2.1) it is assumed that the $A(q)$ and $B(q)$ polynomials have the same degree and that there is a time delay of $d \geq 1$ sample intervals. This is no loss of generality since an n^{th} order continuous time system with a time delay that is not a multiple of the sampling period will give rise to a sampled data system of this form, Åström and Wittenmark (1984). If the delay is a multiple of the sampling interval then the degree of the B-polynomial will be one less the degree of the A-polynomial. For simplicity, we will use the structure defined above since the other case can simply be incorporated in the proposed estimation algorithm and in the design.

□

Remark 2.2.

It is also possible to formulate and solve the adaptive design problems discussed in this paper using δ-operators. The δ-operator is discussed for instance in Goodwin et. al. (1986).

□

Remark 2.3.

The shift form B/A has up until now been the dominating parameterization, since the model (2.1) is then linear in the parameters.

□

The output of the plant (2.1) is measured at each step of time and the control signal $u(t)$ is determined. It is assumed that the controller is of the form

$$R(q)\, u(t) = -S(q)y(t) + T(q)y_r(t) \tag{2.2}$$

where y_r is the reference signal. The polynomial R is assumed to be monic.

The specifications for the closed loop system are given in terms of a desired closed loop model with the pulse transfer operator

$$G_m(q) = \frac{B_m(q)}{A_m(q)} \tag{2.3}$$

from the reference value y_r to the output y. The closed loop system using (2.2) in (2.1) is

$$y(t) = \frac{BT}{q^d AR + BS} y_r(t) + \frac{CR}{q^d AR + BS} e(t) \tag{2.4}$$

The pole placement problem is solved using the Diophantine equation

$$q^d A(q)R(q) + B(q)S(q) = A_o(q)A_m(q) = A_c(q) \tag{2.5}$$

or

$$q^d P(q)R(q) + Z(q)S(q) = A_o(q)A_m(q) = A_c(q) \tag{2.6}$$

and

$$T(q) = B_m(q)A_o(q)\cdot t_0 \tag{2.7}$$

where $A_o(q)$ corresponds to observer dynamics that are cancelled in the pulse transfer operator from y_r to y. The parameter t_0 is chosen such that correct gain is obtained in (2.4). Details concerning the pole placement design method are found in Åström and Wittenmark (1984).

In the adaptive case the system (2.1) is assumed to be unknown. The parameters are estimated based on the measured outputs and the applied inputs. The estimated parameters are then used instead of the true ones in the design procedure, i.e. the certainty equivalence principle is used. The control problem is defined by the Diophantine equation (2.5) or (2.6) and the controller (2.3). The desired closed loop characteristic polynomial A_c can be given in factorized form (preferable) or in shift form.

The adaptive control problem discussed in this paper thus consists of two parts:

- Estimation of the parameters in the polynomials A(q) and B(q) or P(q) and Z(q).
- Solution of the Diophantine equation (2.5) or (2.6).

Most adaptive pole placement controllers have previously used the shift form B(q)/A(q). In the special case when all the process zeros are cancelled, it is possible to make a direct adaptive controller, see for instance Åström and Wittenmark (1980). This requires, however, that the system has all its zeros inside the unit circle, i.e. that the system is minimum phase. Cancellation of all process zeros reduces the Diophantine equation to a triangular system of equations, which is easy to solve. The approach taken in this paper is to use the pole-zero parameterization, Z(q)/P(q). This will always make it possible to convert the general pole placement problem into the solution of a triangular system of equations, thus reducing the computations when controlling general nonminimum phase systems. The pole-zero parameterization also avoids the sensitivity problems associated with the shift form.

3. PARAMETER ESTIMATION

In this section we will discuss the estimation of parameters in factorized form. This has the disadvantage when compared to the shift form that it is not linear in the parameters. Estimation of systems in factorized form has been discussed in the adaptive filtering literature, see for instance Jackson and Wood (1978) and Orfanidis and Vail (1986). In these papers usually only all zero models are considered. Another approach is suggested in Dasgupta et. al. (1983, 1985). Their approach is designed for partially known systems, but can also be used for estimation of the parameters in factorized form. The disadvantage of this approach however, is an increase in the number of estimated parameters.

The approach taken in this paper is to base the estimation of the poles and zeros on the Recursive Prediction Error Method (RPEM) described in Ljung and Söderström (1983). The idea is to convert the input-output model (2.1) into a state space model with unknown parameters. The loss function has, however, several minima depending on the ordering of the poles and the zeros. Fortunately all local minima are equally valid for us since we are only interested in the values of the poles and zeros and not on their connection with our fictious state space model. The RPEM has the advantage that it can be shown, under certain conditions, to converge to a local minimum of the loss function.

Consider the model (2.1). In Ljung and Söderström (1983) it is shown how the system can be written in the innovations form

$$\begin{aligned} \hat{x}(t+1,\theta) &= F(\theta)\hat{x}(t,\theta) + G(\theta)u(t) + K(\theta)v(t) \\ y(t) &= H(\theta)\hat{x}(t,\theta) + v(t) \end{aligned} \qquad (3.1)$$

where $v(t)$ corresponds to the prediction error or innovation

$y(t) - \hat{y}(t|\theta)$ and $K(\theta)$ corresponds to the gain in the steady state Kalman filter. Notice that the model (3.1) is parameterized in the gain, $K(\theta)$, directly instead of an indirect parameterization through noise covariances. The order of the state space representation (3.1) is n+d.

The parameters in (3.1) can be estimated using the following algorithm (Ljung and Söderström (1983) p.127).

$$\epsilon(t) = y(t) - \hat{y}(t) \tag{3.2a}$$

$$L(t) = P(t-1)\psi(t)[\psi(t)^T P(t-1)\psi(t)+\lambda(t)]^{-1} \tag{3.2b}$$

$$P(t) = [P(t-1)-L(t)[\psi(t)^T P(t-1)\psi(t)+\lambda(t)]^{-1}L^T(t)]/\lambda(t) \tag{3.2c}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t)\epsilon(t) \tag{3.2d}$$

$$\hat{x}(t+1) = F(\hat{\theta}(t))\hat{x}(t) + G(\hat{\theta}(t))u(t) + K(\hat{\theta}(t))\epsilon(t) \tag{3.2e}$$

$$\hat{y}(t+1) = H(\hat{\theta}(t))\hat{x}(t+1) \tag{3.2f}$$

$$W(t+1) = [F(\hat{\theta}(t))-K(\hat{\theta}(t))H(\hat{\theta}(t))]W(t)+\bar{M}(\hat{\theta}(t),\hat{x}(t),u(t),\epsilon(t)) - K(\hat{\theta}(t))D(\hat{\theta}(t),\hat{x}(t)) \tag{3.2g}$$

$$\psi(t+1) = W^T(t+1)H(\hat{\theta}(t))^T+D^T(\hat{\theta}(t),\hat{x}(t+1)) \tag{3.2h}$$

$$\bar{M}(\hat{\theta}(t),\hat{x}(t),\ m(t),\epsilon(t)) = \frac{\partial}{\partial\theta}\Big[F(\theta)x+G(\theta)u+K(\theta)\epsilon\Big]_{\theta=\hat{\theta}} \tag{3.2i}$$

$$D(\hat{\theta},x) = \frac{\partial}{\partial\theta}\Big[H(\theta)x\Big]_{\theta=\hat{\theta}} \tag{3.2j}$$

The forgetting factor $\lambda(t)$ may be either a constant $0 < \lambda \leq 1$ or can be chosen to be time-varying

$$\lambda(t) = \lambda_0\lambda(t-1) + (1-\lambda_0) \tag{3.3}$$

If $\lambda(t)$ is chosen according to (3.3) then $\lambda(t)$ will start at $\lambda(0)$ and approach 1 with a time constant λ_0.

Finally the updating in (3.2d) must be made such that $F(\hat{\theta}) - K(\hat{\theta})H(\hat{\theta})$ has all eigenvalues strictly inside the unit circle. This projection can be done simply by decreasing the length of the correction $L(t)\epsilon(t)$ if the updated estimates corresponds to an unstable system.

The next step is to transform the factorized input output model into state space form and to compute the partial derivatives in (3.2i, j). Since we do not know if the poles and zeros are complex or real, it is necessary to rewrite (2.1) into the form

$$\frac{Z(q)}{P(q)} = z_0 \cdot \frac{q^2+z_{11}q+z_{12}}{q^2+p_{11}q+p_{12}} \cdots \cdot \frac{q^2+z_{m1}q+z_{m2}}{q^2+p_{m1}q+p_{m2}} \tag{3.4}$$

where $m=n/2$ if n is even and $m=(n+1)/2$ if n odd. The parameters z_{m2} and p_{m2} will be zero if n is odd. Each of the second order blocks have the form shown in Figure 3.1 and can be written in the state space form

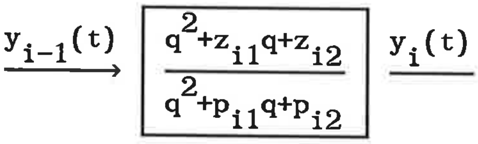

Figure 3.1 Block diagram for the i^{th} block of (3.4)

$$\begin{bmatrix} x_{i1}(t+1) \\ x_{i2}(t+1) \end{bmatrix} = \begin{bmatrix} -p_{i1} & -p_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i1}(t) \\ x_{i2}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} y_{i-1}(t)$$

$$y_i(t) = [z_{i1}-p_{i1} \quad z_{i2}-p_{i2}] \begin{bmatrix} x_{i1}(t) \\ x_{i2}(t) \end{bmatrix} + y_{i-1}(t) \tag{3.5}$$

with $y_0(t) = u(t-d)$ and $y_m(t) = y(t)$. The system (3.4) can be written as

$$x(t+1) = \begin{bmatrix} -p_{11} & -p_{12} & 0 & 0 & & 0 \\ 1 & 0 & 0 & 0 & & \\ z_{11}-p_{11} & z_{12}-p_{12} & -p_{21} & -p_{22} & & \\ 0 & 0 & 1 & 0 & & \\ \vdots & & & & & \\ \vdots & & & & & \\ \vdots & & & & & \\ z_{11}-p_{11} & z_{12}-p_{12} & z_{21}-p_{21} & z_{22}-p_{22} & -p_{m1} & -p_{m2} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} z(t) + \begin{bmatrix} z_0 \\ 0 \\ z_0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ z_0 \\ 0 \end{bmatrix} u(t-d) \tag{3.6}$$

$$y(t) = [z_{11}-p_{11} \;\; z_{12}-p_{12} \cdots z_{m1}-p_{m1} \;\; z_{mz}-p_{mz}]\, x + z_0\, u(t-d)$$

where

$$x(t)^T = [x_{11}(t) \;\; x_{12}(t) \;\; x_{21}(t) \;\; x_{22}(t) \;\; \ldots \;\; x_{m1}(t) \;\; x_{m2}(t)]$$

To obtain a complete state space representation we also need to have a small subsystem that implements the d step delay in the control signal. This part will, however, not contain any of the parameters z_{ij} or p_{ij}. The system (3.6) is now of the form (3.1) and the partial derivatives in (3.2i,j) are easily computed. The algorithm (3.2) can thus be used to estimate the parameters z_{ij} and p_{ij} $i=1,\ldots,m, j=1,2$ and z_0. The poles and zeros can then easily be computed.

The algorithm (3.2) also estimates the elements of the K-vector. The characteristic polynomial of $F(\hat{\theta})-K(\hat{\theta})H(\hat{\theta})$ is an estimate of the noise polynomial C in (2.1).

The ordering of the second order blocks in (3.4) can of course not be determined by the RPEM algorithm. This implies that the loss function will have several minima, each corresponding to one permutation of the blocks. Since we are only interested in the poles and zeros it is unimportant to which minima the algorithm converges.

4. FACTORIZED FORM OF THE DIOPHANTINE EQUATION

In this section we will discuss the solution of the Diophantine equation (2.6), where $P(q)$ and $Z(q)$ are given in factorized form. In Section 2 it was assumed that P and Z do not have any common factors. To begin with we also assume that there are no multiple poles or zeros. (That case will be treated later in the paper). Further, we have the condition that $\deg P = \deg Z$. It will be assumed in the following that $B_m = B$. The generalization to arbitrary B_m is trivial. A casual feedback cannot reduce the time delay in the system. This implies that $A_c(q)$ must contain the factor q^d, which implies that the factor also must be present in S. The equation (2.6) is now reduced to

$$P(q)R(q) + Z(q)S'(q) = A_c'(q) \qquad (4.1)$$

where

$$\begin{aligned} \deg R &= n+d-1 \\ \deg S &= n-1 \\ \deg A_c' &= 2n-1 \end{aligned} \qquad (4.2)$$

The unknown polynomials R and S' have $2n+d-1$ unknown coefficients. We will now discuss some methods of obtaining these polynomials.

The standard way to solve (4.1) is to compute the shift form of P and Z and equate the coefficients for different powers of q. This leads to a set of $2n+d-1$ linear equations.

Introduce the notation

$$R(q) = \bar{R}(q) + R'(q) \qquad (4.3)$$

where

$$\begin{aligned} \bar{R}(q) &= q^n(q^{d-1} + \bar{r}_1 q^{d-2} + \ldots + \bar{r}_{d-1}) \\ R'(q) &= r_0' q^{n-1} + r_1' q^{n-2} + \ldots + r_{n-1}' \end{aligned}$$

Considering the degrees of the polynomials in (4.1) it is easily seen that the coefficients in the $\bar{R}$ polynomial can be computed recursively. (Use for instance long division). We are thus left with 2n linear equations in the polynomial form.

$$P(q)R'(q) + Z(q)S'(q) = A_c'(q) - P(q)\bar{R}(q) \qquad (4.4)$$

This polynomial equation can be solved for R' and S' provided the P and Z (or equivalently the A and B) polynomials do not have any common factors.

In the second approach for solving the Diophantine equation we will use the fact that the model is given in factorized form.

If we evaluate (4.4) for $q=p_i$ and $q=z_i$ we get two sets of n linear equations

$$Z(p_i)\, S'(p_i) = A_c'(p_i) \qquad i=1,\ldots,n \qquad (4.5)$$

$$P(z_i)R'(z_i) = A_c'(z_i)-P(z_i)\bar{R}(z_i) \qquad i=1,\ldots,n \qquad (4.6)$$

Remark 4.1.

For simplicity we assumed that the poles and zeros are real. If there are complex poles or zeros we have to evaluate both the real and complex parts of the equations. □

This will decrease the complexity of the computations by a factor of 4 since the number of operations in the solution of a set of linear equations is proportional to m^3 where m is the number of equations.

The new way to further reduce the computations and to solve (4.4) that is proposed in this paper is to reparameterize R' and S' as

$$R'(q) = \sum_{i=0}^{n-1} r_i'' g_i(q) \qquad (4.7)$$

$$S'(q) = \sum_{i=0}^{n-1} s_i'' f_i(q) \qquad (4.8)$$

where

$$g_i(q) = \begin{cases} 1 & i=0 \\ \prod_{j=1}^{i}(q-z_j) & 1 \le i \le n-1 \end{cases} \tag{4.9}$$

$$f_i(q) = \begin{cases} 1 & i=0 \\ \prod_{j=1}^{i}(q-p_i) & 1 \le i \le n-1 \end{cases} \tag{4.10}$$

The n polynomials g_i (f_i) are linearly independent and are called the Newton coefficients of the set z_i (p_i). This representation is always possible, see for instance Blum (1972, p.339). The polynomials g_i and f_i have the properties

$$g_i(z_k) = \begin{cases} 0 & 1 \le k \le i \\ \prod_{j=1}^{i} (z_k-z_j) & i+1 \le k \le n-1 \end{cases} \tag{4.11}$$

$$f_i(p_k) = \begin{cases} 0 & 1 \le k \le i \\ \prod_{j=1}^{i} (p_k-p_j) & i+1 \le k \le n-1 \end{cases} \tag{4.12}$$

Now consider (4.5) with the parameterization defined by (4.8) and (4.10). Because of the property (4.12) we get a triangular system of equations.

$$\begin{aligned}
&s''_0 && = A'_c(p_1) \\
&s''_0 + s''_1 f_1(p_2) && = A'_c(p_2) \\
&s''_0 + s''_1 f_1(p_3) + s''_2 f_2(p_3) && = A'_c(p_3) \\
&\vdots && \\
&s''_0 + \quad \ldots \quad \ldots \quad s''_{n-1} f_{n-1}(p_n) && = A'_c(p_n)
\end{aligned}$$

A similar triangular system of equations is obtained using (4.6), (4.7), (4.9) and (4.11).

The solution of (4.4) is now reduced to two triangular n^{th} order system of equations. This implies that all of the coefficients s''_i, r''_i, $i=1,\ldots n$ and $\bar{r}_i$, $i=1, \ldots, d-1$ can be obtained recursively. This will considerably simplify the computations. There are two problems that still have to be resolved: multiple poles or zeros and complex poles and zeros.

Multiple poles or zeros.

Assume that a pole p_i has multiplicity m_i. This implies that m_i equations in (4.5) are replaced by

$$\begin{aligned} &Z(p_i)S'(p_i) = A_c'(p_i) \\ &\frac{d}{dq}\left[Z(q)S'(q)\right]_{q=p_i} = \frac{d}{dq}\left[A_c'(q)\right]_{q=p_i} \\ &\frac{d^{m_i}}{dq^{m_i}}\left[Z(q)S'(q)\right]_{q=p_i} = \frac{d^{m_i}}{dq^{m_i}}\left[A_c'(q)\right]_{q=p_i} \end{aligned} \qquad (4.13)$$

The first equation will make it possible to evaluate s''_{i-1}. The second equation can be used to get s''_i etc. The procedure is best illustrated by an example.

Example 4.1.

Assume that

$$Z(q) = (q-z_1)(q-z_2)(q-z_3)$$

and that P has a pole of multiplicity 3 in p_1. Then

$$f_0 = 1$$

$$f_1 = q-p_1.$$

$$f_2 = (q-p_1)^2$$

and

$$S'(q) = s_0'' + s_1''f_1(q) + s_2''f_2(q)$$

The equations (4.13) are then

$$Z(p_1)s_0'' = A_c'(p_1)$$

$$Z'(p_1)\ s_0'' + Z(p_1)\ s_1'' = \frac{d}{dq}\left[A_c'(q)\right]_{q=p_1}$$

$$Z''(p_1)\ s_0'' + 2Z'(p_1)s_1'' + Z(p_1)\cdot 2s_2'' = \frac{d^2}{dq^2}\left[A_c'(q)\right]_{q=p_1}$$

where Z' and Z" are the first and second derivative of Z. Further we have used the fact that

$$\frac{d}{dq}\ f_2(q) = 2f_1(q)$$

From these three equations s_0'', s_1'' and s_2'' can be obtained recursively.

□

Multiple poles will thus not cause any problems in principle. It should, however, be remarked that multiple and almost multiple poles or zeros must be tested for and specially treated in the implementation of the algorithm.

Complex poles and zeros.

Complex poles and zeros do not cause any problems since (4.5) and (4.6) can be treated as equations defined in the complex plane. The coefficients are complex however, and filtering of complex signals must be included in the implementation of the controller. (The net result must, however, become a real signal). One way to circumvent this problem is to have another representation for S' and R' for the complex poles and zeros respectively. The methodology is most easily explained by example.

Example 4.2.

Assume that

$$P(q) = (q-p_1)(q-p)(q-\bar{p})$$

where p_1 is a real pole and p and $\bar{p}$ are a pair of complex conjugate poles. The S' polynomial has three coefficients. Assume that

$$\begin{aligned} S'(q) &= s_0 + s_1(q-\bar{p}) + s_2(q-\bar{p})(q-p) \\ &= s_0 - s_1\bar{p} + s_1 q + s_2(q^2-(p+\bar{p})q + \bar{p}p) \end{aligned}$$

We now get the following equations

$$Z(p_1)[s_0 + s_1(p_1-\bar{p}) + s_2(p_1-\bar{p})(p_1-p)] = A_c'(p_1)$$

$$Z(p)[s_0 + s_1(p-\bar{p})] = A_c'(p)$$

$$Z(\bar{p})\ s_0 = A_c'(\bar{p})$$

Introduce the notation

$$\frac{A_c'(p)}{Z(p)} = \alpha + \beta i$$

$$\frac{A_c'(\bar{p})}{Z(\bar{p})} = \alpha - \beta i$$

$$\frac{A_c'(p_1)}{Z(p_1)} = \gamma$$

$$p = a + bi$$

This gives

$$s_0 = \frac{A_c'(\bar{p})}{Z(\bar{p})} = \alpha - \beta i$$

$$s_1 = \frac{A_c'(p) - s_0 Z(p)}{Z(p)(p-\bar{p})} = \frac{\beta}{b}$$

$$s_2 = \frac{\gamma - s_0 - s_1(p_1-a+ib)}{(p_1-\bar{p})(p_1-p)} = \frac{\gamma - \alpha - \beta(p_1-a)/b}{p_1^2 - 2p_1 a + a^2+b^2}$$

It is only s_0 that is complex, but we use it in the combination

$$s_0 - s_1\bar{p} = \alpha - a\beta/b$$

This implies that all the signals in the filtering are real.

□

The example can easily be generalized. For every pair of complex poles or zeros we have to solve a similar problem to that in the example in order to determine two new controller coefficients. The only computation that involves complex computations is the evaluation of the polynomials for a complex argument. It is straightforward to write a subroutine for these computations.

Realization of the control law.

The controller (2.2) can be implemented as shown in Figure 4.1. The polynomials T^*, etc. are the backward shift operator representations of the respective polynomials. The blocks for S'^* and R'^* can be implemented as shown in Figure 4.2.

This implies that the poles and zeros can be used in the solution of the Diophantine equation and in the implementation of the control law. It is thus possible to avoid the sensitivity problems associated with the shift form representation.

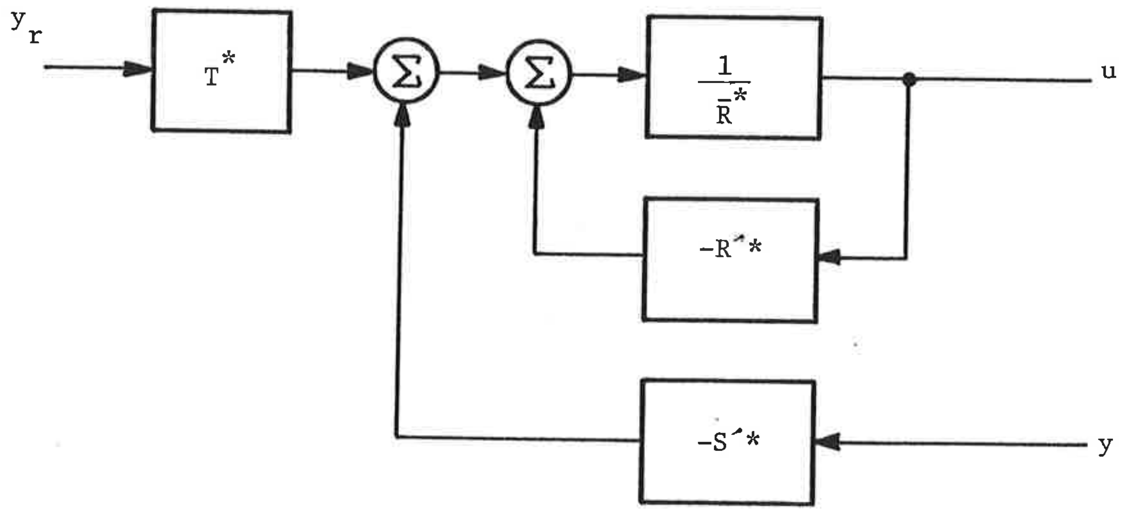

Figure 4.1. Imlementation of the controller (2.2)

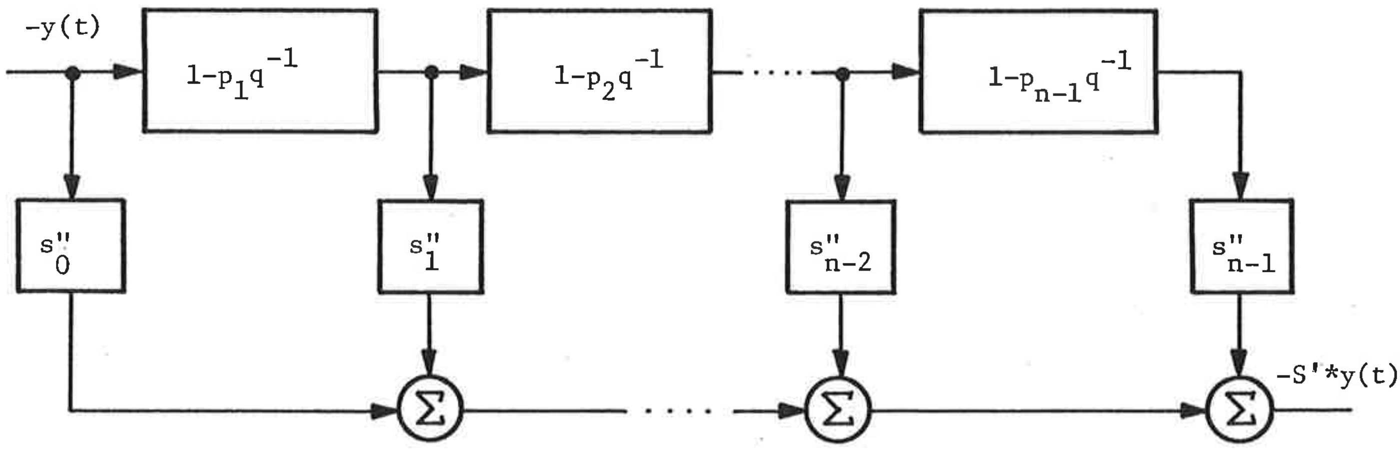

Figure 4.2. Implementation of $-S'^*y(t)$ based on the parameterization (4.8) and (4.10).

5. DISCUSSION AND CONCLUSIONS.

In this paper we have discussed and investigated new algorithms for general adaptive pole placement. The algorithms are based on a pole zero parameterization of the process to be controlled. It is shown that the new parameterization can be used to simplify the computations when solving the design problem. The Diophantine equation (4.1) is reduced to a triangular set of linear equations. The parameters of the controller can then be found recursively. It is also shown that the new parameterization can be used for implementing the controller. This makes it possible to avoid calculation of the shift form of the process model on the controller and consequently to avoid the sensitivity problems associated with the shift form. The advantages with the proposed parameterization are obtained by abandoning the linear in the parameters concept.

The estimation procedure discussed in Section 3 is not very much more complex than the conventional extended least squares method that can be used for the shift form of (2.1). The RPEM will converge to a local minimum of the loss function. There exist several local minima which corresponds (at least in the noise free case) to permutations of poles and/or zeros. It is, however, irrelevant to which of these minima the estimator converges.

There are still many unanswered questions when using the proposed algorithm adaptivley. Convergence and stability properties and the numerical robustness issues are under investigation.

6. REFERENCES.

Åström K.J. (1983): Theory and applications of adaptive control - A survey. *Automatica*, 19, pp.471-486.

Åström K.J. and B. Wittenmark (1980): Self-tuning controllers based on pole-zero placement. *Proc. IEEE*, Part D, 128, pp.120-130.

Åström K.J. and B. Wittenmark (1984) : Computer controlled systems. Prentice Hall, Englewood Cliffs, N.J.

Blum E.K. (1972): Numerical Analysis and Computation: Theory and Practice, Addison-Wesley, Reading, MA.

Dasgupta S., B.D. Anderson, and R.J. Kaye (1983): Robust identification of partially known systems. *Proc. 22nd IEEE Conf. on Decision and Control*, San Antonio, TX.

Dasgupta S., B.D. Anderson, and R.J. Kaye (1985): Identification of physical parameters in structured systems. Technical Report, Department of Systems Engineering, Australian National University, Canberra.

Goodwin G.C., R.L. Leal, D.Q. Mayne and R.H. Middleton (1986): Rapproachment between continuous and discrete model reference adaptive control. *Automatica*, 22, pp.199-207.

Goodwin G.C. and D.Q. Mayne (1987): A parameter estimation perspective of continuous time model reference adaptive control, *Automatica*, 23, to appear.

Jackson L.B. and S.L. Wood (1978): Linear prediction in cascade form. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-26, pp.518-528.

Ljung L. and T. Söderström (1983): Theory and Practice of Recursive Identification. MIT Press, Cambridge, MA.

Orfanidis S.J. and L.M. Vail (1986): Zero-tracking adaptive filters. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-34, pp.1566-1571.

Seborg D.E., T.F. Edgar and S.L. Shah (1986): Adaptive control strategies for process control : A survey. AIChE Journal, 32, pp.881-913.