A Linear Quadratic Gaussian Self-Tuner

Åström, Karl Johan; Zhao-Ying, Z.

Published in:
Ricerche di Automatica

1982

K.J. ASTROM and Z. ZHAO-YING

A LINEAR QUADRATIC GAUSSIAN SELF-TUNER

Estratto da: *Ricerche di Automatica*
Vol. 13, No 1, October 1982

KLIM ROMA 1985

Centro Stampa Klim - Roma via del Boschetto, 107
Direttore responsabile per la stampa KLIAMAKIS ANASTASIO

RICERCHE DI AUTOMATICA
Vol. 13, No. 1, October 1982
Printed in Italy

A LINEAR QUADRATIC GAUSSIAN SELF-TUNER

K.J. ÅSTRÖM and Z. ZHAO-YING

Abstract. The paper describes a self-tuning regulator for single-input single-output systems based on linear quadratic gaussian (LQG) design and recursive estimation. The design problem is solved using spectral factorization and solution of a linear polynomial equation. The parameter estimation is based on extended least squares. The regulator has been implemented on a micro computer DEC LSI 11/03. The implementation admits interactive experimentation with operator communication via an ordinary terminal. All programming is done in the Pascal language. Applications to an analog computer simulation of ship steering and control of a laboratory process for concentration control are given.

1. Introduction.

Self-tuning regulators are based on a very simple heuristic idea. A design problem is first solved under the assumption that the model of the system and its environment is known. When the parameters are not known they are replaced by estimates obtained from a recursive parameter estimator. A self-tuning regulator where the underlying design scheme is based on linear quadratic gaussian (LQG) control theory is developed in this paper. An advantage of this formulation is that the performance of the control system can be characterized by a few parameters. In the single-input single-output case there is in fact only one parameter, the weighting factor between penalty on the control signal and the error. Another advantage is that the LQG theory is not restricted to any particular class of systems. It can

---

Manuscript received October 16, 1982.
The Authors are with the Department of Automatic Control, Lund Institute of Technology, S-220 07 Lund, Sweden.

thus easily be applied to non-minimum phase systems as well as to systems with variable time delays.

A self-tuner based on LQG was first proposed by Åström (1974). The solution was based on an iterative solution of the steady state Riccati equation. This idea was further elaborated by Åström and Wittenmark (1975), Peterka and Åström (1976), Gustavsson (1980), Belanger (1981), Zhao-ying and Åström (1981).

In this paper a different idea is used. The solution is given in terms of spectral factorization and a linear polynomial equation. This approach has several advantages. It is easy to incorporate constraints on the loop gain at high and low frequencies to ensure robustness against both low frequency disturbances and unmodeled high frequency dynamics. The sampling period can also be adjusted easily. The algorithm can also be modified to give a pole placement self-tuner.

The paper is organized as follows. A review of the LQG design for systems with known parameters are given in section 2. A brief summary of the recursive parameter estimation used is given in section 3. Section 4 deals with practical problem related to implementation and coding. Results of experiments performed with the self-tuner are presented in section 5 and section 6.

2. LQG design for systems with known parameters.

The design procedure for systems with known parameters is discussed in this section. Consider a single-input single-output system described by the model

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{2.1}$$

where q is the forward shift operator.

Let the criterion be to minimize

$$E \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} [y^2(k) + \rho u^2(k)] \tag{2.2}$$

The optimal feedback law is given by

$$R(q)u(t) = C(q)u_c(t) - S(q)y(t) \tag{2.3}$$

where the polynomials R and S are determined by a two step procedure.

The characteristic polynomial of the optimal closed loop system is first obtained as a solution to the spectral factorization problem

$$P(z)P(z^{-1}) = \rho A(q)A(z^{-1}) + B(z)B(z^{-1}) \tag{2.4}$$

where ρ is the weighting factor in the criterion (2.2). The polynomials R and S are then obtained from the solution of the linear polynomial equation.

$$A(z)R(z) + B(z)S(z) = P(z)C(z) \tag{2.5}$$

See Åström (1979) and Zhao-ying and Åström (1981b).

Sensitivity constraints

There are normally two types of sensitivity constraints on a control system. The loop gain BS/AR should be sufficiently high at low frequencies to make sure that low frequency disturbances are rejected. The loop gain should also be falling of sufficiently fast at high frequency to reduce the effects of uncertainties in the high frequency dynamics of the process.

A high loop gain at low frequencies will be obtained automatically if the model (2.1) includes sufficient low frequency disturbances. When the model (2.1) is estimated there is, however, no guarantee that the loop gain will always be sufficiently high. One possibility to guarantee a high loop gain at low frequencies is to require that the polynomial R has one or more zeroes at $z=1$ i.e. to make sure that the regulator has integral action.

Similar constraints can be introduced in order to make sure that the loop gain falls off sufficiently fast at high frequencies. The cross over frequency can be determined and a constraint which guarantees that the loop gain decreases sufficiently fast can be added. A considerable attenuation of high frequencies is obtained automatically in sampled data systems because of the antialiasing filler.

The sampling period can also be determined automatically based on a calculation of the cross over frequency.

Spectral Factorization

The spectral factorization problem is solved iteratively by the method proposed by Wilson (1969) and Vostry (1975) which uses the recursion

$$P_{j+1} = \frac{1}{2}[P_j + X_j] \tag{2.6}$$

where X_j is given by

$$P_j(z)X_j(z^{-1})+X_j(z)P_j(z^{-1}) = \rho A(z)A(z^{-1})+B(z)B(z^{-1}) \tag{2.7}$$

If the iteration is started with a stable polynomial the sequence $\{P_j\}$ will converge to the desired solution. See Wilson (1969). In practice only a few iterations are required. One to three iterations are used in the program. The polynomial P from a previous step is used as the first iterate.

Equation (2.7) is solved using the complex integrals

$$I(\ell) = \frac{1}{2\pi i} \oint \frac{\rho A(z)A(z^{-1})+B(z)B(z^{-1})}{P_j(z)P_j(z^{-1})} \frac{dz}{z^{\ell+1}} \tag{2.8}$$

Introducing

$$X = [x_o,\ldots,x_k]^T$$

$$\tilde{P}_j = \begin{bmatrix} p_{o,j} & & & 0 \\ p_{1,j} & 2p_{o,j} & & \\ & & \cdot & \\ & & & \cdot \\ p_{k,j} & 2p_{k-1,j} & \cdots & 2p_{o,j} \end{bmatrix}$$

and

$$I = [I(0),\ldots,I(k)]^T$$

the solution can be written as

$$X = \tilde{P}_j I \tag{2.9}$$

Formula $I(\ell)$ has two versions, one is accurate and another is a reduced version. See Zhao-ying and Åström (1981a).

The Linear Polynomial Equation

The diophantine equation (2.5) has a general solution:

$$\begin{aligned} R(z) &= E(z)P(z)C(z) + G(z)V(z) \\ S(z) &= F(z)P(z)C(z) + H(z)V(z) \end{aligned} \tag{2.10}$$

where E,F,G and H come from a linear transformation

$$\begin{bmatrix} E & F \\ G & H \end{bmatrix} \begin{bmatrix} A & 1 & 0 \\ B & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & E & F \\ 0 & G & H \end{bmatrix} \tag{2.11}$$

with the relationships

$$\begin{aligned} A(z)E(z) + B(z)F(z) &= 1 \\ A(z)G(z) + B(z)H(z) &= 0 \end{aligned} \tag{2.12}$$

which is solved by the Euclidean algorithm. See Kucera (1979). If a common factor exists in A(z) and B(z), say $B_o(z)$, the linear transformation gives

$$\begin{bmatrix} E_1 & F_1 \\ G_1 & H_1 \end{bmatrix} \begin{bmatrix} A_1B_o & 1 & 0 \\ B_1B_o & 0 & 1 \end{bmatrix} = \begin{bmatrix} B_o & E_1 & F_1 \\ 0 & G_1 & H_1 \end{bmatrix}$$

E_1, F_1, G_1 and H_1 are the solution to (2.12) after cancelling the common factor.

Pole Placement

It is clear from the description of the design algorithm that a pole placement self-tuner is obtained as a by-product, simply by specifying the polynomial P instead of determining it from spectral factorization.

3. Parameter estimation.

The estimation is an extended least squares algorithm, Panuska (1969) applied to the model (2.1). To describe the algorithm introduce the notations

$$\hat{\theta} = [a_1,\dots,a_n,b_1,\dots,b_m,c_1,\dots c_\ell]^T$$

$$\phi(t) = [-y(t-1),\dots,-y(t-n),u(t-1),\dots,u(t-m),$$
$$\varepsilon(t-1),\dots,\varepsilon(t-\ell)]^T$$

and

$$\varepsilon(t+1) = y(t+1) - \phi(t+1)^T\theta(t).$$

The extended least squares algorithm is given by

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1)\varepsilon(t+1)$$

$$K(t+1) = \frac{P(t)\phi(t+1)}{1+\phi(t+1)^T P(t)\phi(t+1)}$$

$$P(t+1) = \frac{1}{\lambda}\left[P(t) - \frac{P(t)\phi(t+1)\phi(t+1)^T P(t)}{\lambda+\phi(t+1)^T P(t)\phi(t+1)}\right] \tag{3.1}$$

where the factor λ is the forgetting factor introduced to discount past data when performing the estimation. In actual implementation a square root algorithm based on the U-D is preferrable. See Bierman (1977).

To avoid problem of levels,the estimation may be based on the difference model.

$$A(z^{-1})\nabla y(t) = B(z^{-1})\nabla u(t) + C(z^{-1})\nabla\varepsilon(t) \tag{3.2}$$

where ∇ is a difference operator. The parameters estimates are still given by (3.1) provided that the vector ϕ in (3.1) is replaced by

$$\phi(t) = [-\nabla y(t-1),\dots,-\nabla y(t-n),\nabla u(t-1),\dots,\nabla u(t-m),\nabla\varepsilon(t-1),\dots,$$
$$\nabla\varepsilon(t-\ell)]^T.$$

The parameters estimates are then given by (3.1).

4. Implementation.

All programs required for the self-tuner are written in Pascal for DEC LSI 11/03. The program has two major procedures, a foreground program which realizes the adaptive control algorithm and a background program for operator communication. The programs are scheduled using a simple real time schedular, Mattsson (1978). A brief description of the program is given here.

The foreground program performs the recursive parameter estimation and the regulator design. The background program initializes the algorithm. It also allows operator communication based on a simple command driven interaction.

The following commands are available:

HELP	give available commands
STOP	stop the self-tuner
RUN	start the self-tuner
DISP	display parameters
PAR	change parameters
STORE	store reference and control signals
RESULT	store parameters of regulator and estimator

Size of Program and Code

The source code is about 1400 lines of Pascal. This includes comments and declarations. Some additional details are given in Table 1. The total size of the compiled code is about 40 kbytes. Examples of execution times are given in Table 2. This table is based on the assumption that three iterations are required for the spectral factorization. In the coding flexibility and readability has been emphasized rather than compactness and computational speed.

It is of interest to compare this implementation of the LQG self-tuner with others. An implementation based on the solution of Riccati equation was given in Åström (1974). In this implementation there was no operator communication. The pure foreground code compiled to about 8 kbytes on the PDP-15. The program was transferred to PDP 11/03 by Gustavsson (1980). Operator communication was also added in this implementation. The source code for the program was about 1400 lines of Fortran code. Half of them are comments. The compiled code required about 40 kbytes. Of these about 8 kbytes was required for the pure foreground.

It thus appears that implementation based on Riccati equations and polynomial manipulations require about the same amount of code. The minimum size of a dedicated implementation with no operator communication is about 8 kbytes.

Table 1 - Size of source code including comments and declarations.

Program	Number of lines
Foreground	770
Estimation	37
Spectral factorization	144
Linear polynomial equation	275
Background	604

Table 2 - Execution times for the adaptive LQG controller with deg A = deg B = deg C = n

n	execution time [s]
1	0.18
2	0.28
3	0.50
4	0.80
5	1.22
6	1.78
7	2.50
8	3.38

5. Ship steering.

Ship steering is one of the few problems where there is a quadratic loss function given from physics. Let ψ be the heading and δ the rudder angle. The added resistance due to steering ΔR can approximatively be expressed as

$$\frac{\Delta R}{R} = k \frac{1}{T} \int_0^T \left\{ \left[\psi(t) - \psi_{ref} \right]^2 + \rho\delta^2(t) \right\} dt. \tag{5.1}$$

See Norrbin (1972). The ship steering dynamics can be described as a third order system with the transfer function

$$\frac{L\{\psi\}}{L\{\delta\}} = k \frac{s + b}{s(s+a)(s+c)} \tag{5.2}$$

Disturbances due to wind and waves may be represented as filtered high frequency noise and narrow band or periodic signals. There is also measurement noise. Experimental verification of such models are given in Åström and Källström (1976) and Källström and Åström (1981).

A model for ship steering was simulated on an analog computer. The parameters chosen were $a = 0.37$, $b = 1.4$ and $c = 2.34$. These values are representative for a large tanker. The numbers correspond to a normalized model where the time it takes to travel one ship length is the time unit. Wind disturbances were simulated as filtered white noise. Waves were simulated as a triangular wave. To simplify the interpretation of the results the disturbances were introduced as equivalent rudder motions. Under ideal situations the rudder motions should thus be similar to the disturbances. The sampling period was chosen as 0.5 time units. The parameter ρ has the value 0.14 and the forgetting factor $\lambda = 0.98$. Representative values of the parameters are

$$A(z) = z(z^2 - 1.75z + 0.61)$$

$$B(z) = 0.12z + 0.07.$$

The polynomial P is then

$$P(z) = z(z^2 - 0.90z + 0.26).$$

The regulator which minimizes the loss function (5.1) subject to the constraint that it has integral action is given by

$$R(z) = (z - 1)(z^3 + 0.81z^2 - 1.02z - 0.81)$$

$$S(z) = 18.9z^4 - 22.7z^3 + 6.8z^2.$$

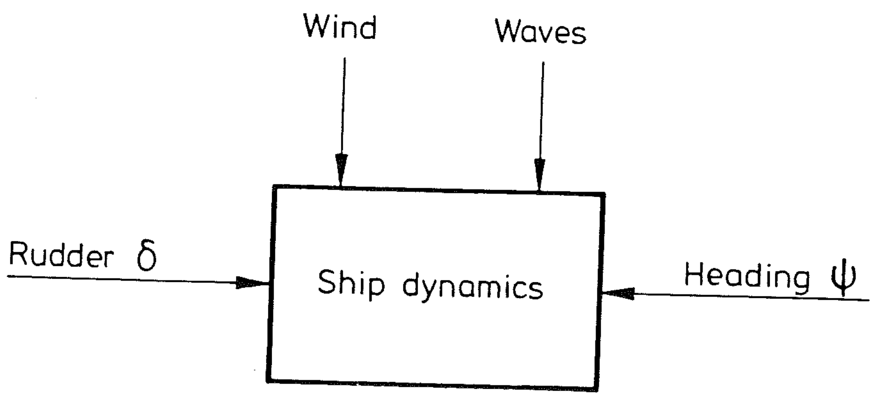

Fig. 1. Block diagram for simulation of ship steering.

The closed loop bandwidth corresponds to a period of approximatively 12 time units.

Overview of the results

Many different configurations were tried in the simulations. A systematic investigation of LQG self-tuners having different model orders and different sampling periods was done. For a second order model there is an analytic solution to the LQG problem as was shown by Åström (1981). This analytic solution admits simplifications of the calculations and the code. The performance of this simple algorithm was compared to self-tuners based on more complex models. It was found that the improvements from more complex models were barely noticable in the recorder traces of system inputs and outputs.

Low sensitivity to low frequency disturbances may be achieved by increasing the model order to allow modeling of low frequency disturbances. The desired property can also be obtained by choosing a regulator structure with integral control as was discussed in section 2. In this particular case it was advantageous to force integral action. The parameter estimates of the high order model converge slowly. The loop gain obtained at low frequencies will also vary with the disturbance level. The estimates of the low order model were stable. They also converged quite rapidly.

The performance of the system is illustrated in Fig. 2 which shows the response of the system to wave disturbances having different periods. It is seen from the figure that the heading errors

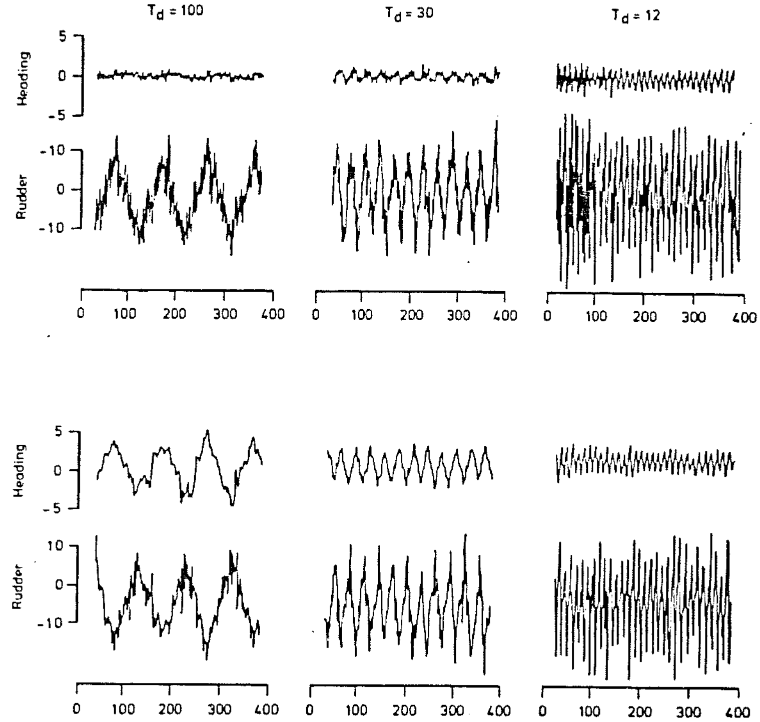

Fig. 2. Results of simulation of LQG self-tuners for a ship steering problem.

increase with decreasing period of the disturbance. Notice that the characteristic period of the closed loop system corresponds to a period of about 12 time units. Fig. 2 also shows the same results for an algorithm without enforced integral action. The performance of this system is almost the same for rapid disturbances. The performance for slow disturbances is, however, not so good. The performance improves after some time. When the disturbances have the period 100 it may however take over 1000 steps before the performance is as good as for the system with forced integral action.

6. Concentration control.

The LQG self-tuner has also been applied to concentration control using a laboratory process. The process is shown in Fig. 3. Fresh water flows through a mixing chamber where it is mixed with a concentrated salt solution. The flow rate of the salt solution is controlled by a peristaltic pump. Via a selector valve the flow is then sent through a short tube, a long tube or a stirred tank. The concentration at the outlet of either vessel is measured using a conductivity cell. The outlet flow may also be recirculated to the input. The amount of recirculation can be adjusted. The control variable is the speed of the peristaltic pump. The controlled variable is the concentration at the outlet. The dynamics of the process depends on which vessel is selected. The dynamics varies with the flow rate. The time-delay and the time-constants are inversely proportional to the flow rate. The process gain is directly proportional to the concentration of the salt solution and inversely proportional to the flow. Due to the peristaltic action there will also be rapid fluctuations in the flow.

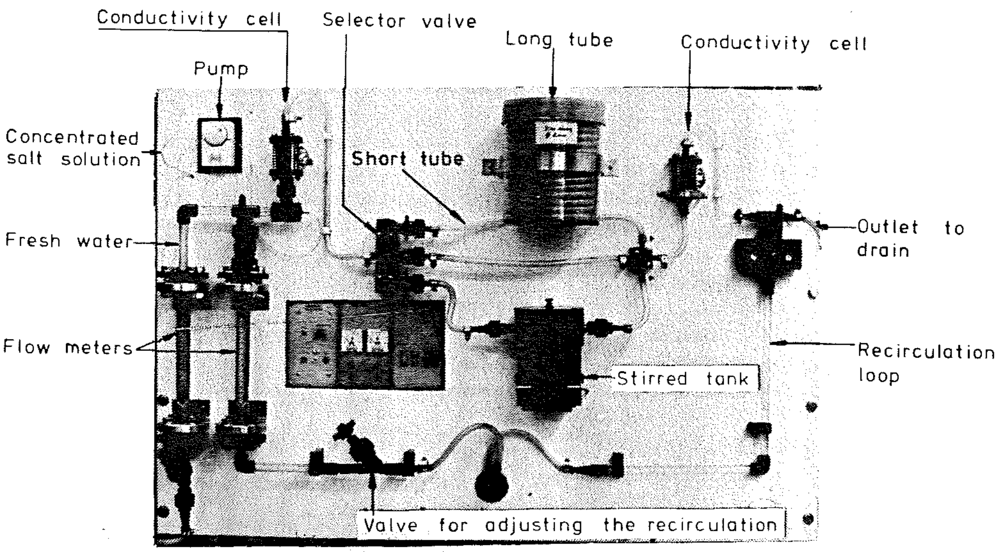

Fig. 3. The laboratory process.

Overview of the results

Many different configurations were tried in the simulations. A systematic investigation of LQG self-tuners having different model orders and different sampling periods was done. For a second order model there is an analytic solution to the LQG problem as was shown by Åström (1981). This analytic solution admits simplifications of the calculations and the code. The performance of this simple algorithm was compared to self-tuners based on more complex models. It was found that the improvements from more complex models were barely noticable in the recorder traces of system inputs and outputs.

Low sensitivity to low frequency disturbances may be achieved by increasing the model order to allow modeling of low frequency disturbances. The desired property can also be obtained by choosing a regulator structure with integral control as was discussed in section 2. In this particular case it was advantageous to force integral action. The parameter estimates of the high order model converge slowly. The loop gain obtained at low frequencies will also vary with the disturbance level. The estimates of the low order model were stable. They also converged quite rapidly.

The performance of the system is illustrated in Fig. 2 which shows the response of the system to wave disturbances having different periods. It is seen from the figure that the heading errors

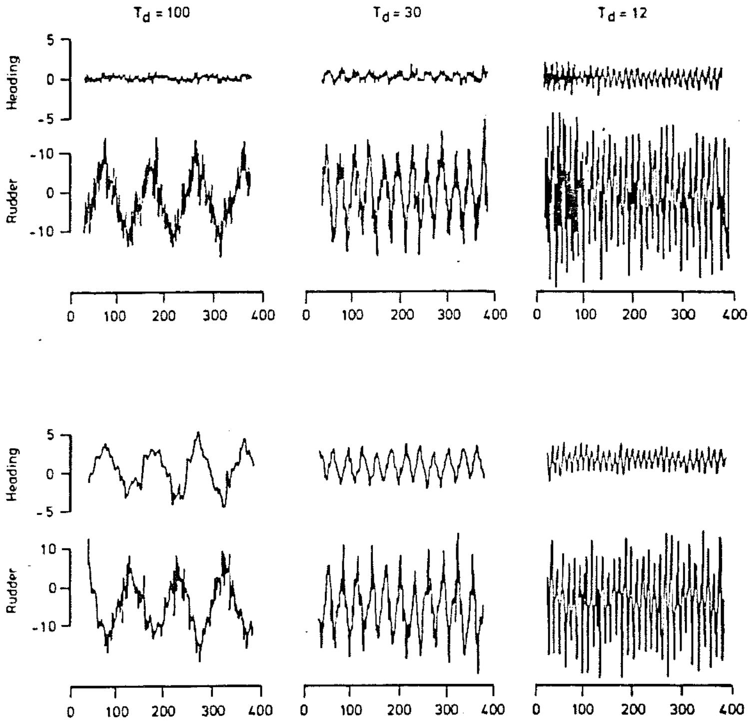

Fig. 2. Results of simulation of LQG self-tuners for a ship steering problem.

In our experiments the long pipe was used. Impulse responses at different flow rates are shown in Fig. 4. The figure shows clearly that there is a substantial variation of the dynamics with the flow rate. Also notice the response obtained with recirculation.

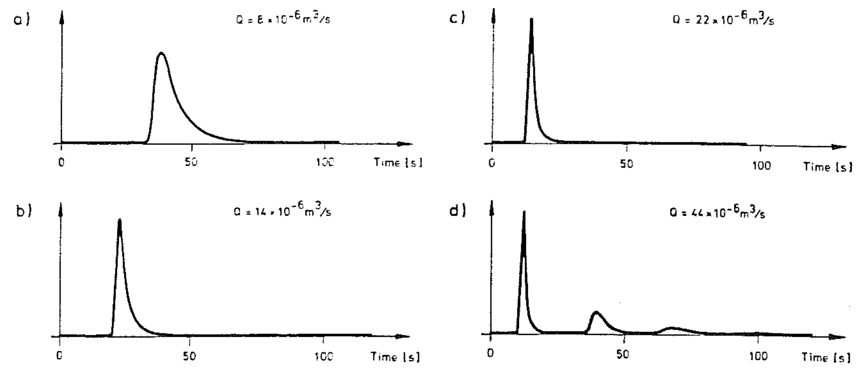

Fig. 4. Impulse responses of the process with the long pipe for different flow rates.

The control problem

The control problem we tried to solve was to design a robust adaptive controller which could handle the variations in dynamics shown in Fig. 4 and additional gain variations due to changes in the concentration of the salt solution. The identification had to be based upon set point changes and the normal process disturbances. Looking at Fig. 4 it is clear that the variations in the time delay is a major difficulty.These variations may be captured by a model of type (2.1), provided that the sampling period and the degrees of the polynomials are chosen appropriately. After some experimentation it was found that a model where the polynomial A has one coefficient the polynomial B has three or four coefficients could be estimated using the normal signals. Such a model might be expected because the dynamics may be approximated by a time delay and a first order lag. Sampling of such a model gives the discrete time system

$$y(t)+ay(t-1) = b_k u(t-k)+b_{k+1} u(t-k-1)$$

where the sampling period is chosen as the time unit and the integer k is such that the time-delay is between kh and kh+h. With the variations in the time-delay shown in Fig. 4 it is not possible to choose k and h so that all cases will be covered. Extra b- parameters are therefore introduced. These extra parameters may also contribute to modeling of higher order dynamics. Experimentation showed that the uncertainties of the estimates increased considerably if more parameters were estimated. Three b- parameters were chosen as a compromize and the following model structure was then used.

$$y(t)+ay(t-1) = b_1u(t-1)+b_2u(t-2)+b_3u(t-3)+e(t) \quad (5.3)$$

With a sampling period of 15 s it is then possible to model systems with time delays up to 30 s which is adequate for the variations shown in Fig. 4. Some experiments were made in order to determine a suitable value of ρ. The value $\rho = 5$ gives a reasonable compromise between response time and magnitudes of control actions.

Signal conditioning

There are high frequency variations in the measured signals because of pressure fluctuations introduced by the peristaltic pump. It is therefore very important to filter the signals before they are sampled so that there is no appreciable signal transmission above the Nyquist frequency. If this is not done the high frequency disturbances will appear as low frequency disturbances because of aliasing. The effect of filtering the output signal before it is sampled is illustrated in Fig. 5. The figure shows the measured output signal before the filter and the control signal. The self-tuner is started with all parameters equal to zero except the first b-coefficient. Notice the dramatic improvement after 40 sampling periods when a first order low pass filter with a time-constant corresponding to 5 sampling periods is introduced before the A-D converter.

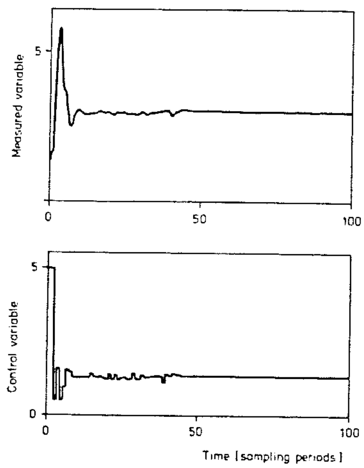

Fig. 5. Illustrates the importance of prefiltering. The sampling period is 15 s. The flow is 12×10^{-6} m^3/s. The model used in the self-tuner has the structure (5.3).

Performance of regulators with constant parameters

It will first be shown that regulators with constant parameters will not work if there are large flow changes. This is illustrated in Fig. 6. The flow is first set to 14×10^{-6} m^3/s and the self-tuner is run for about 30 sampling periods. The estimated model obtained then has the pulse transfer function

$$H(z) = \frac{0.40z + 0.14}{z^2(z - 0.52)}$$

This correspond to a first order system with a time constant of 13 s and a time delay of 17 s. The regulator parameters are then fixed and the flow is changed. It is seen from the Fig. 6 that the regulator behaves well when the flow is increased to 22×10^{-6} m^3/s. When the flow is decreased to 10×10^{-6} m^3/s the damping decreases however and the control loop becomes unstable when the flow is decreased to 8×10^{-6} m^3/s. The results are quite natural because the time-delay and the time constants increase with decreasing flow. When the flow is sufficiently small the time delay is so large that the system becomes unstable.

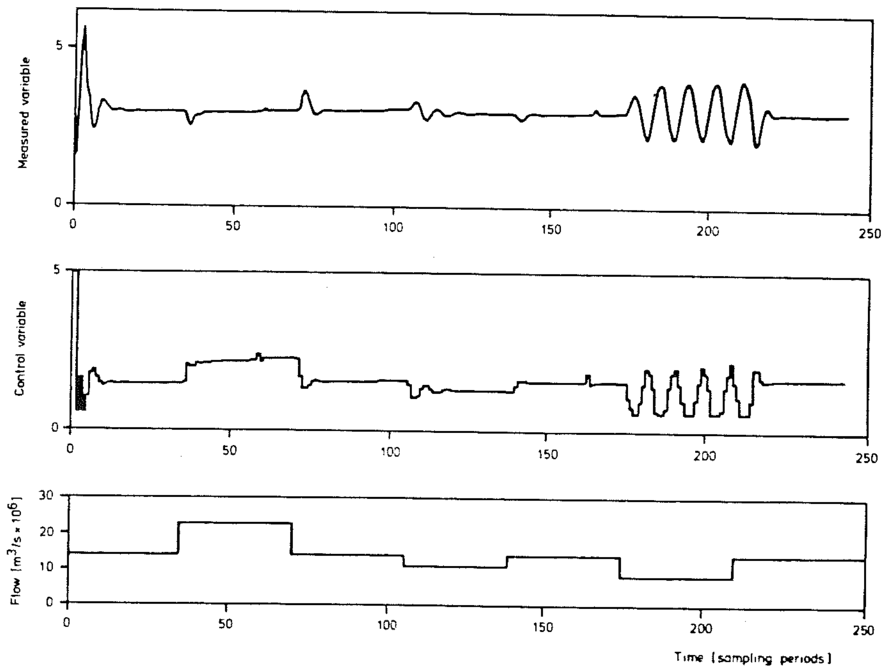

Fig. 6. Results of experiments with varying flow. A self-tuner is used during the first 30 sampling periods when the flow is 14×10^{-6} m^3/s. The regulator parameters are then fixed and the flow is changed.

Results of a similar experiment are shown in Fig. 7. In this case the process is however initialized with a flow of 8×10^{-6} m^3/s. The self-tuner is used for the first 35 sampling periods. The estimated model obtained has the transfer function

$$H(z) = -\frac{0.28}{z^2(z - 0.52)}$$

which corresponds to a first order system with time constant of 23 s and a time delay of 30 s. Fig. 7 shows that the closed loop becomes unstable when the flow is increased to 14×10^{-6} m^3/s.

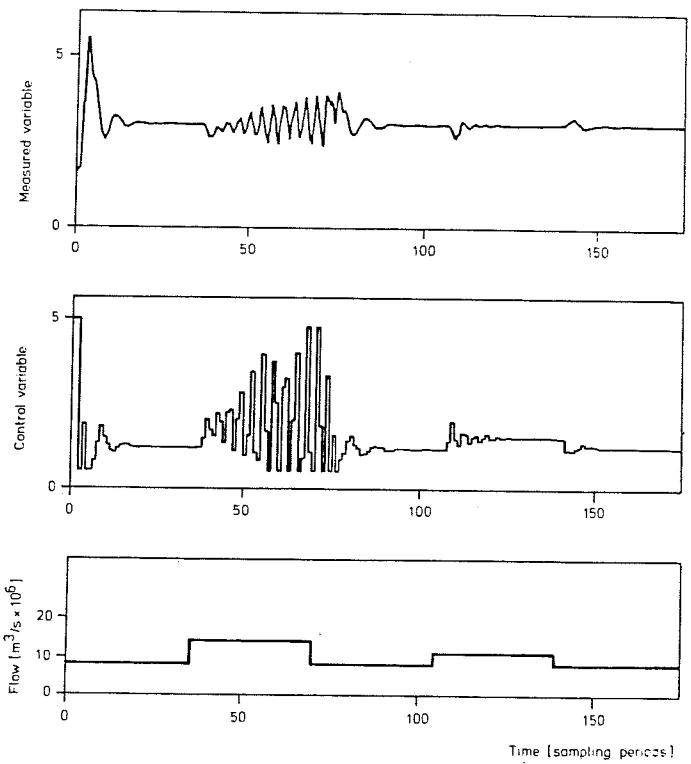

Fig. 7. Results of experiments with varying flow. A self-tuner is used during the first 35 sampling periods, when the flow is $8 \times 10^{-6} m^3/s$. The regulator parameters are then kept constant and the flow is changed.

Adaptive Control

The experiments with regulators having a fixed gain indicate that there are difficulties in finding a regulator with constant parameters which works well over a range of flows from 8×10^{-6} m^3/s to $22 \times 10^{-6} m^3/s$. It is actually possible to find a regulator which will stabilize the system over the whole range of flows provided that the gains are chosen very low. Such a regulator will, however, have a poor performance.

Results from experiments with an LQG self-tuner under the same flow changes as in Fig. 6 are shown in Fig. 8. The figure shows clearly that the self-tuner can easily cope with the parameter variations. The parameters used in the self-tuner are $\rho = 5$ and $\lambda = 0.98$. The parameter estimates are based on differences of the input and the output signals. A comparison with Fig. 6 shows that the self-tuner has considerably better performance than a constant gain regulator. It

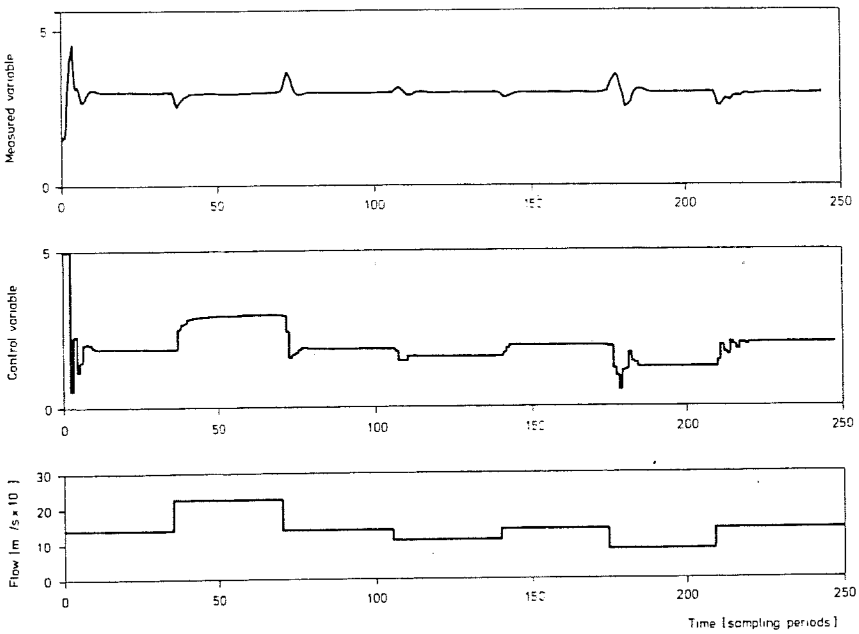

Fig. 8. Results of experiments with an LQG self-tuner when the flow varies. The changes in the flow are similar to those in Fig. 6.

is of course possible to make such a self-tuner unstable by decreasing the flow rate so much that the time delay is larger than 45 s. The model (5.3) with a sampling period of 15 s is then no longer adequate. The properties of the LQG self-tuner are further illustrated in Fig.9 which shows how it responds to set point changes, flow changes, pulse disturbances and introduction of recirculation. The self-tuner is initialized with all parameters zero except b_1. The set point is changed from 1.4 to 2 at time 35. A load disturbance is introduced at time 105 by injecting a salt solution with high concentration for 10 s. A sequence of flow changes are then introduced. The flow is increased by 30% at time 140. It is reduced to its normal value at time 150. The flow is decreased by 30% at time 210. At time 245 it is returned to its normal value again. At time 280 the recirculation pump is started. The results in Fig. 9 show that the self-tuner works very well in all cases.

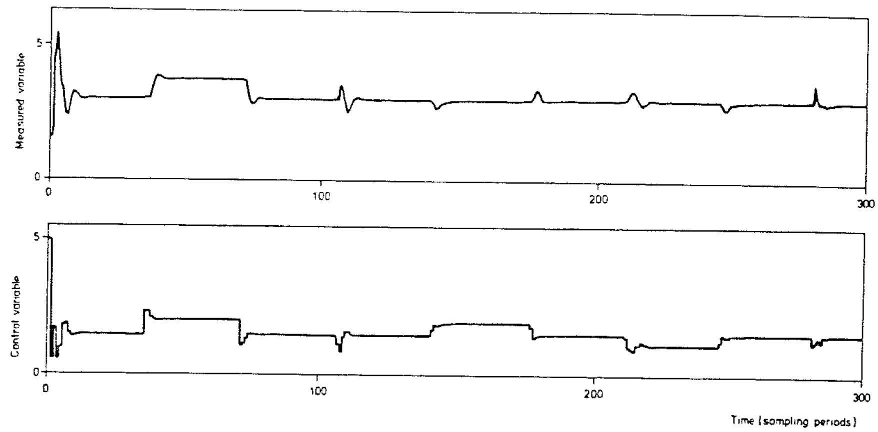

Fig. 9. Experiments with an LQG self-tuner with set point changes, load disturbances, flow changes and recirculation.

REFERENCES

P. BELANGER (1981): *On type 1 systems and the Clarke-Gawthrop method.* Preprints Yale Workshop on Adaptive Control.

G.J. BIERMAN (1977): *Factorization Methods for discrete sequential estimation.* Academic Press, New York.

I. GUSTAVSSON (1980): *Implementation of a Self-tuning Regulator for Non-minimum Phase Systems on PDP 11/03.* Report. CODEN: LUTFD2/(TFRT-7209)/1-093/(1980).

V. KUCERA (1979): *Discrete Linear Control. The Polynomial Equation Approach.* Academica Prague.

C.G. KÄLLSTRÖM and K.J. ÅSTRÖM (1981): *Experiences of System Identification Applied to Ship Steering.* Automatica 17 187-198.

S.E. MATTSSON (1978): *A Simple Real-time Scheduler.* Report. CODEN: LUTFD2/(TFRT-7156)/1-010/(1978).

N.H. NORRBIN (1972): *On the added resistance due to steering on a straight course.* Proc. 13th Towing Tank Conference,Berlin/ Hamburg.

V. PANUSKA (1979): *An Adaptive Recursive Least Squares Identification Algorithm.* Proc. 1969. IEEE Symposium on Adaptive Processes. Pennsylvania State University.

Z. VOSTRY (1975): *New algortihm for polynomial spectral factorization with quadratic convergence*. Kybernetica (Prague) 11 415-422.

G.T. WILSON (1969): *Factorization at the covariance generating function of a pure noising average process*. SIAM J. Numerical Analysis 6 1-7.

B. WITTENMARK, P. HAGANDER and I. GUSTAVSSON (1980): *STUPID Implementation of a self-tuning PID-controller*. CODEN: LUTFD2/(TFRT-7201)/1-030/(1980).

Z. ZHAO-YING and K.J. ÅSTRÖM (1981a): *Polynomial Factorization Using Recursive Formulas for Complex Intergrals*. Report. CODEN: LUTFD2/(TFRT-7223)/1-030/(1981).

Z. ZHAO-YING and K.J. ÅSTRÖM (1981b): *Regulator Synthesis Based on Polynomial Manipulation*. Report. CODEN: LUTFD2/(TFRT-7225)/1-044/(1981).

K.J. ÅSTRÖM (1974): *A Self-tuning Regulator for Non-minimum Phase Systems*. Report. CODEN: LUTFD2/(TFRT-3113)/1-048/(1974).

K.J. ÅSTRÖM and K.G. KÄLLSTRÖM (1976): *Application of ship steering dynamics*. Automatica 12 9-22.

K.J. ÅSTRÖM (1979): *Algebraic System theory as a tool for regulator design*. Acta Polytechnica Scandinavica 31 52-65.

K.J. ÅSTRÖM (1980a): *Self-tuning regulator-design principles and applications*. In Narendra and Monopoli (1980).

K.J. ÅSTRÖM (1980b): *Design of fixed gain and adaptive ship steering autopilots based on the Nomoto Model*. Proc. Symposium on Ship Steering Automatic Control. Genova June 1980, pp. 225-243.

K.J. ÅSTRÖM (1980c): *Ship Steering - A Test Example for Robust Regulator Design*. CODEN: LUTFD2/(TFRT-7222)/1-009/(1981).

K.J. ÅSTRÖM (1981): *Theory and applications of adaptive control*. Proc. 8th IFAC World Congress.