Design Principles for Self-Tuning Regulators

Åström, Karl Johan

1980

LUND UNIVERSITY

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

CODEN:LUTFD2/(TFRT-7203)/0-049/(1980)

DESIGN PRINCIPLES FOR SELF-TUNING REGULATORS

KARL JOHAN ÅSTRÖM

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
SEPTEMBER 1980

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 725
S 220 07 Lund 7 Sweden

Document name
REPORT

Date of issue
September 1980

Document number
CODEN:LUTFD2/(TFRT-7203)/0-049/(1980)

Author(s)
Karl Johan Åström

Supervisor

Sponsoring organization
STU contract 78-3763

Title and subtitle
Design Principles for self-tuning regulators

Abstract
A unified description of many types of self-tuners is given. Relations to design of controllers for systems with known parameters and recursive estimation methods are emphasized. The distinction between self-tuners based on identification of explicit and implicit process models are discussed as well as the relations between Self-Tuning Regulators (STR), and Model Reference Adaptive Systems (MRAS). An overview of practival problems and operational issues is given. The particular problems of integral action and estimator windup are covered in more detail.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
49

Security classification

Recipient's notes

Distribution: The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

DESIGN PRINCIPLES FOR SELF-TUNING REGULATORS

K.J. ÅSTRÖM
Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

ABSTRACT

A unified description of many types of self-tuners is given. Relations to design of controllers for systems with known parameters and recursive estimation methods are emphasized. The distinction between self-tuners based on identification of explicit and implicit process models are discussed as well as the relations between Self-Tuning Regulators (STR), and Model Reference Adaptive Systems (MRAS). An overview of practical problems and operational issues is given. The particular problems of integral action and estimator windup are covered in more detail.

1. INTRODUCTION

Adaptive control has been a challenge to control engineers for a long time. Many adaptive control schemes have been proposed. In spite of this progress in the field has been comparatively slow. One reason is that it is difficult to understand how adaptive systems work because they are inherently nonlinear. Another reason is that it has been costly and fairly complicated to implement adaptive controllers. The situation has changed drastically with the advent of microprocessors which makes implementation of adaptive controllers feasible. Recently there has also been progress in theory of adaptive control. See Ljung (1977), Egardt (1979), Goodwin et al (1978), Morse (1979) and Narendra et al (1979).

Self-tuning regulators (STR) and model reference adaptive systems (MRAS) are two popular approaches. An overview of STR is given in Section 2. It is shown that self-tuning regulators can be derived in a simple way which has a strong intuitive appeal. It is then shown by examples, how many different types of self-tuners can be generated. Relations between STR and MRAS are also discussed in Section 2. Practical aspects on self-tuners are discussed in Section 3. This includes different ways to use STR as well as abuses of self-tuners. Two particular practical problems namely how to introduce integral action and how to avoid estimator windup are discussed in Sections 4 and 5. The parametrization problem is discussed in Section 6.

2. SELF-TUNING REGULATORS

This section gives a brief description of self-tuning regulators. The discussion is limited to control of single-input single-output systems described by

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t) \qquad (2.1)$$

where u is the input, y the output and $A(q^{-1})$ and $B(q^{-1})$ polynomials in the backward shift-operator. For further details we refer to the original papers Peterka (1970), and Åström and Wittenmark (1973) and the recent review Åström (1979a), where many references are given. The principles are first discussed. A self-tuner based on classical control design is then presented as an example. The notion of explicit and implicit algorithms is also discussed.

Principles

A block diagram of self-tuning regulator is shown in Fig.1.

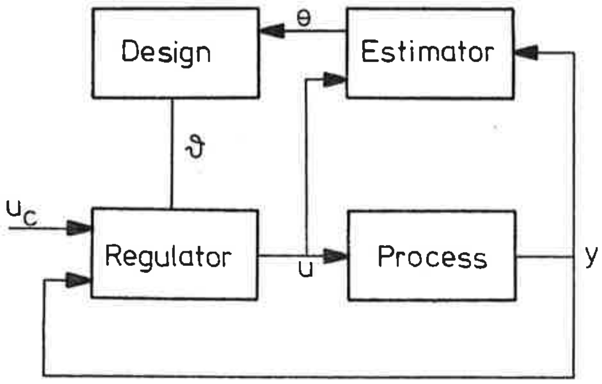


Figure 1. Schematic diagram of a self-tuning regulator

The self-tuner can be thought of as being composed of three parts, a parameter estimator, a design calculation and a regulator with adjustable parameters. The design calculation computes the parameters of the regulator from the parameters which describe the process. The parameter estimator determines the parameters which characterize the process and its environment from measurements of the process input and output.

The regulator structure shown in Fig.1 is very flexible because it allows many different combinations of design and estimation methods. So far, only a small number of the possible combinations have been explored. Intuitively it seems reasonable to choose a design method, which gives desired performance when the parameters of the process are known, and an estimation method which will work well for the particular disturbances. It turns out, however, that the structure shown in Fig.1 also has unexpected properties. The regulator shown in Fig.1 is a *certainty equivalence* control in the terminology of stochastic control theory because the fact that the parameter estimates are not exact is disregarded. It is possible to introduce modifications which also take the uncertainties of the parameter estimates into account (*cautious control*) and modifications which introduce extra *probing signals* when the parameter estimates are uncertain. The principles will be illustrated by a few simple examples.

A self-tuning servo

Consider a servoproblem. A classical formulation of the design problem is to find a regulator which gives the desired transfer function from the command signal to the output. Let the desired transfer function be

$$G_M = \frac{Q}{P} \tag{2.2}$$

A self-tuning servo which gives this transfer function is given by

ALGORITHM E1 (*Basic explicit algorithm*)

Data: The polynomials P, T_1, and Q_1 are given.

Step 1: Estimate the parameters of the model

$$Ay(t) = Bu(t) \tag{2.1}$$

by least squares.

Step 2: Factor the estimated polynomial $\hat{B}$ into $\hat{B}^+$ and $\hat{B}^-$ where all zeros of $\hat{B}^+$ are well damped and all zeros of $\hat{B}^-$ are unstable or poorly damped.

Step 3: Solve the linear equation.

$$\hat{A}R_1 + \hat{B}^-S = PT_1. \tag{2.3}$$

(Notice that there are many solutions and that a choice has to be made).

Step 4: Calculate the control variable u from

$$Ru = Tu_c - Sy \tag{2.4}$$

where $R = R_1\hat{B}^+$, and $T = T_1Q_1$.

The steps 1, 2, 3, and 4 are repeated at each sampling period. □

The algorithm is discussed in detail in Åström and Wittenmark (1979). Similar algorithms for regulation are discussed in Wellstead et al (1979). If the parameter estimates converge the closed loop transfer function will be

$$\frac{Q_1B^-}{P}$$

Notice that this is the best that can be obtained because it is not possible to cancel unstable or poorly damped process zeros.

The algorithm E1 is called an algorithm based on *estimation of process* parameters or an algorithm with *explicit identification*, because the parameters of the process model (3.1) in the standard form are estimated. Using the terminology of model reference adaptive systems the algorithm is also called *indirect*, because the parameters of the regulator are updated indirectly via estimation of the process parameters (Step 1) and the design calculations (Steps 2, 3, and 4). See Narendra, Lin and Valavani (1979).

The algorithm E1 can be simplified little in two special cases. If it is known that the process has no unstable zeros apart from a known number of time-delays it follows that $B^-(q^{-1}) = q^{-k}$. Step 2 is then not necessary. The second step in the algorithm is also avoided if all process zeros are considered as unstable or poorly damped. In that case $\hat{B}^- = \hat{B}$.

Implicit algorithms

It is possible to construct algorithms where the design calculations are avoided and the parameters of the regulator are updated directly. The basic self-tuning regulator in Åström and Wittenmark (1973) is a prototype for algorithms of this type. The idea is to rewrite the process model in such a way that the design step is trivial. By a proper choice of model structure the regulator parameters are updated directly and the design calculations are thus eliminated. Algorithms of this type are called algorithms based on *implicit identification* of a process model. In the terminology of model reference adaptive systems the corresponding algorithms are also called *direct methods* because the parameters of the regulator are updated directly.

An example of an explicit algorithm will now be given. Consider a process described by (3.1) with $B^- = q^{-k}$. Assume that it is desired to find a feedback such that the transfer function from the reference value to the output is

$$\frac{z^{-k}}{P(z^{-1})}$$

This means that all process zeros have to be cancelled. Assuming that the process model is known the design equation (2.3) becomes

$$PT_1 = AR_1 + q^{-k}S$$

Hence

$$PT_1y = AR_1y + q^{-k}Sy = q^{-k}R_1Bu + q^{-k}Sy = q^{-k}(Ru + Sy) \qquad (2.5)$$

where (2.1) is used to obtain the second equality. The process can thus be represented either by (2.1) or by (2.5). The representation (2.5) has the advantage that the polynomials R and S, which appear in the feedback law, occur explicitly. The following self-tuning control algorithm is then obtained

ALGORITHM 12 (*Implicit algorithm with all process zeros cancelled*)

Data: Given the polynomials P and T, where P is normalized such that P(1) = 1.

Step 1: Estimate the parameters of the polynomials R and S in the model

$$PT\, y = q^{-k}(Ru + Sy) \qquad (2.5)$$

by least squares.

Step 2: Calculate the control signal using

$$\hat{R}u = Tu_c - \hat{S}y, \qquad (2.6)$$

where $\hat{R}$ and $\hat{S}$ are the polynomials estimated in Step 1.

The Steps 1 and 2 are repeated at each sampling period. □

This algorithm was originally proposed in Clarke and Gawthrop (1975). Since the specifications require that all process zeros are cancelled, they must be sufficiently well damped for the algorithm to function. The algorithm will thus not work for non-minimum-phase systems. It also requires that k is known apriori. Notice that T can be interpreted as the observer polynomial.

Implicit STR and MRAS

It will now be shown that the implicit self-tuning pole-placement algorithm 2 is equivalent to a model reference adaptive system (MRAS). For this purpose it is necessary to consider some details of the algorithm. Introduce

$$\varphi(t) = [y(t-k) \; \ldots \; y(t-k-n_S) \; u(t-k) \; \ldots \; u(t-k-n_R)]^T \qquad (2.7)$$

where

$$n_S = \deg S \text{ and } n_R = \deg R.$$

In the implicit algorithm the estimated parameters are equal to the regulator parameters. Hence

$$\theta = [s_0 \ldots s_{n_S} \;\; r_0 \ldots r_{n_R}]. \qquad (2.8)$$

The residual ε can then be written as

$$\varepsilon(t) = PT\ y(t) - \hat{R}u(t-k) - \hat{S}y(t-k) = PT\ y(t) - \varphi^T(t)\theta \tag{2.9}$$

The least squares formula for updating the parameter estimates can be written as

$$\theta(t+1) = \theta(t) + P(t+1)\ \varphi(t+1)\ \varepsilon(t+1) \tag{2.10}$$

Equation (2.10) can clearly be interpreted as an adjustment rule for the regulator parameters θ. Notice that it follows from (2.9) that

$$\varphi(t) = -\ \text{grad}_\theta\ \varepsilon(t) \tag{2.11}$$

The vector φ can thus be interpreted as a sensitivity derivative, and the least squares updating formula can be written as

$$\theta(t+1) = \theta(t) - P(t+1)\ \varepsilon(t+1)\ \text{grad}_\theta\ \varepsilon(t+1) \tag{2.12}$$

This is identical to the 'MIT rule' used to design MRAS, provided that the model error is replaced by the least squares residual.

LQG self-tuners

Optimal control methods are popular design techniques. Such methods can of course also be used to generate self-tuning regulators. The idea is illustrated using a simple example. Consider a system described by

$$A(q^{-1})\ y(t) = B(q^{-1})\ u(t) + C(q^{-1})\ e(t) \tag{2.13}$$

where e is white noise. Assume that it is desired to find a control law such that the criterion

$$J = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{\infty} [y^2(t) + \rho u^2(t)] \tag{2.14}$$

is minimal. A self-tuning regulator for this problem is given below.

ALGORITHM (Explicit LQG)

Data: Given ρ and the sampling period h.

Step 1: Estimate the parameters of the model (2.12) by extended least squares or by recursive maximum likelihood.

Step 2: Determine a stable polynomial P such that

$$PP^* = \rho\hat{A}\hat{A}^* + \hat{B}\hat{B}^* \quad P(z)P(z^{-1}) = \rho\hat{A}(z)\hat{A}(z^{-1})+\hat{B}(z)\hat{B}(z^{-1}) \tag{2.15}$$

where $\hat{A}$ and $\hat{B}$ are the estimates obtained in Step 1.
Find a solution to the diophantine equation.

$$\hat{A}R + \hat{B}S = \hat{C}P \tag{2.16}$$

such that deg S = deg $\hat{B}$ + deg C − deg P

Step 3: Use the control law

$$Ru = -Sy \tag{2.17}$$

The steps 1, 2 and 3 are repeated at each sampling period □

Notice that there are many variants. Instead of performing the spectral factorization (2.15) and solving the linear equation (2.16) the feedback law (2.17) can be obtained from a Riccati equation. See Åström (1974).

3. PRACTICAL ASPECTS

Some practical aspects on simple regulators are first reviewed briefly. The corresponding problems for self-tuners are then discussed. Operational issues and abuses of self-tuners are also covered.

Simple Regulators

The basic algorithm for a PID regulator is very simple:

$$u = K[e + \frac{1}{T_I}\int^{t} e(s)\,ds + T_D\frac{de}{dt}]. \tag{3.1}$$

An implementation of this algorithm in analog or digital hardware does, however, not necessarily give a good controller. In practice it is also necessary to consider operator interface, filtering of the signals, automatic/manual transfer, bumpless parameter changes, reset windup, nonlinear output, (gap, saturation etc). How well a PID regulator works in an industrial environment depends very much upon these considerations.

Self-tuners

All things that apply to the simple regulators also apply to the self-tuners. For self-tuners there are, however, more things to be considered because the basic algorithm is more complicated than the PID algorithm. For example windup occurs in a PID regulator because the integrator in the algorithm could achieve large values if the control value saturates or if it is driven manually. In a self-tuner with a forgetting factor windup can also occur in the estimator. Some of these problems are discussed in more detail in the following sections. The self-tuning regulator can operate in many different modes like estimation only, tuning etc. The problem of operator interface is particularly important. A key problem is how the specifications are entered and how an operator should interact with the controller. There are many different possibilities ranging from the case where there are no knobs at all on the panel to fairly complicated operator interfaces. Instead of just having manual and automatic modes it maybe useful to have several automatic modes e.g. fixed gain, estimate process parameters but do not update controller parameters, estimate and update controller parameters. The self-tuning regulators which are already on the marked or which are in the process of coming out illustrate the wide range of possibilities.

Operational issues

Self-tuning regulators can be used in many different ways. Since the regulator becomes an ordinary constant gain feedback regulator if the parameter estimates are kept constant, the self-tuner can be used as a *tuner* to adjust the parameters of a control loop. In such an application the self-tuner is connected to the process and run until satisfactory performance is obtained. The self-tuner is then disconnected and the system is left with the constant parameter regulator obtained. This mode of using the self-tuner is convenient to implement in a package for direct digital control (DDC-package). The DDC-package is simply provided with a tuning routine which can be connected to an arbitrary loop in the package.

The self-tuner can also be used *to build up a gain schedule*. In such a case the system is run at different operating points and the controller parameters obtained are stored. When the process has been run at a sufficient number of operating points a table for scheduling the controller parameters can be generated by interpolating and smoothing the parameter values obtained.

The self-tuner can also be used as a truly *adaptive controller* for systems with varying parameters. In cases where rapid adaptation over widely varying operating conditions are required combinations between gain-scheduling and self-tuning can also be considered.

Abuses of self-tuners

Compared with a three-term controller the self-tuner is a sophisticated controller.

Such a controller can of course be misused. The self-tuner should of course not be used if a simpler controller will do the job. Before considering a self-tuning regulator it is therefore useful to check if a simpler regulator will work. The following list may help to decide.

PI or PID
Linear MISO (What order?)
Nonlinear
Fixed Gain
Gain Schedule
Self-tuning or Adaptive

Notice that it is not always easy to decide if a constant gain regulator will work based on the open loop characteristics of the process. Two examples illustrate the point.

Example 1

Fig. 2 shows the step responses of systems with the transfer function

$$G(s) = \frac{1}{(s+1)(s+a)} \tag{3.2}$$

for a = 0, 0.01 and 0.02.

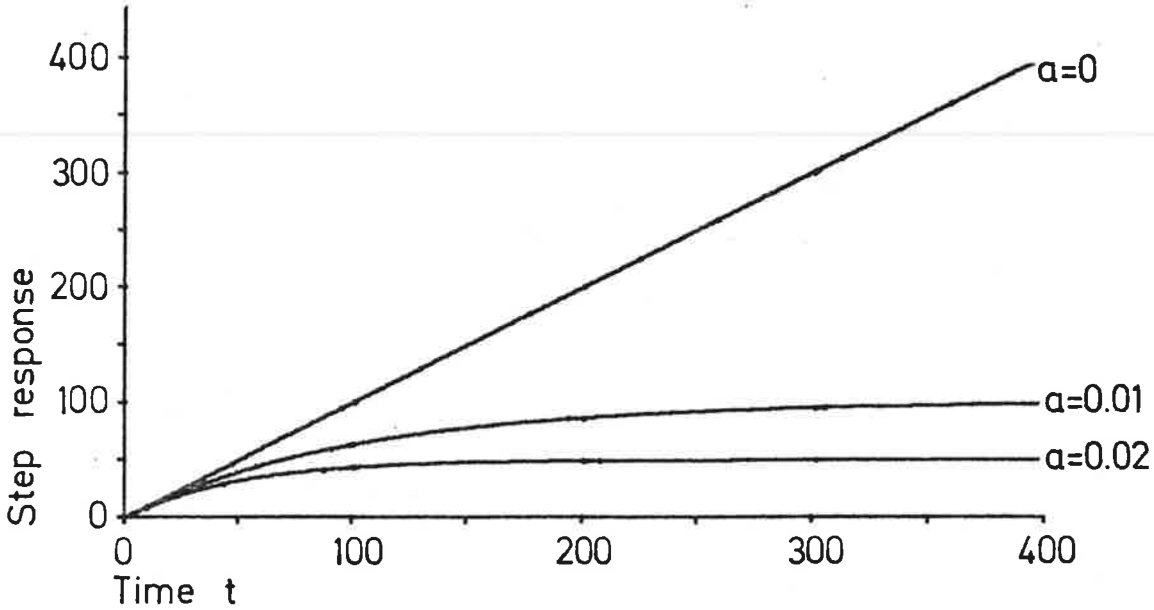


Figure 2. Step responses of open loop systems with transfer function (3.2).

The step responses of the corresponding closed loop systems obtained with the constant parameter feed-back

$$u(t) = y_r - y(t)$$

are shown in Fig.3.

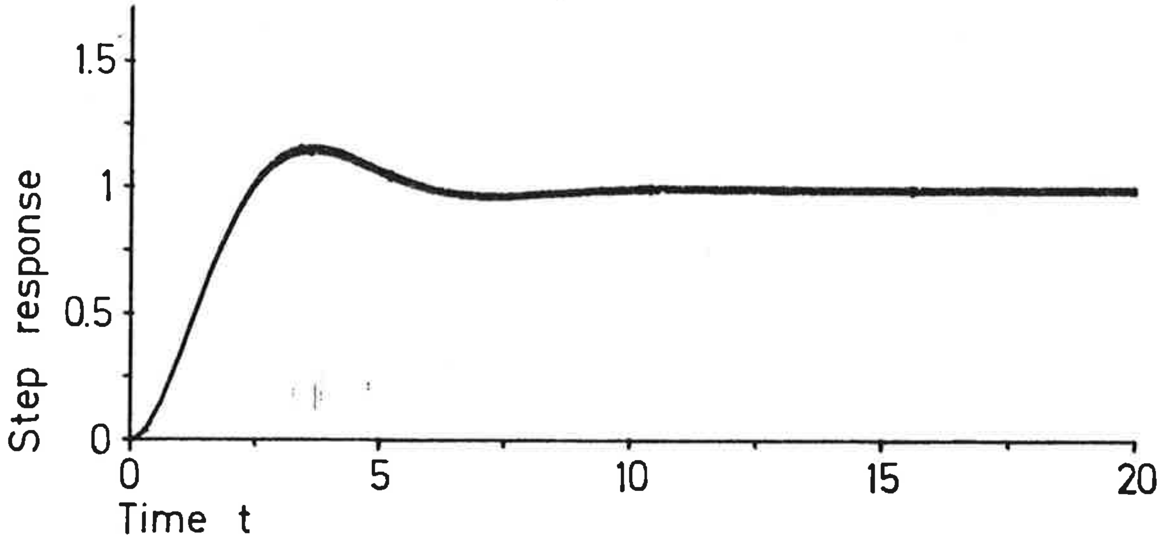


Figure 3. Unit step responses for closed loop systems

□

Example 2

Fig.4 shows the step responses of systems with the transfer function

$$G(s) = \frac{20(1-sT)}{(s+1)(s+20)(1+sT)} \tag{3.3}$$

for T = 0, 0.01, 0.02

The step responses of the corresponding closed loop systems obtained with the constant parameter feed-back law

$$u(t) = 20(y_r - y(t))$$

are shown in Fig.5.

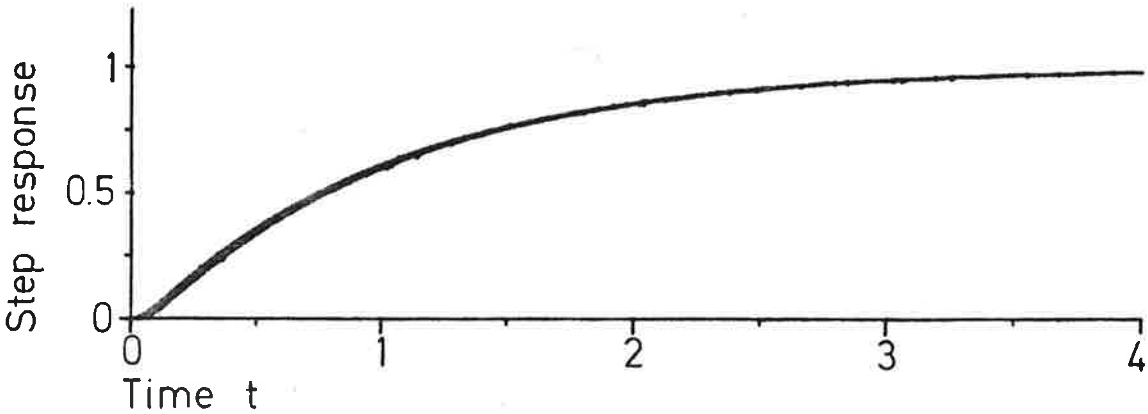


Figure 4. Unit step responses for systems with transfer function (3.3).

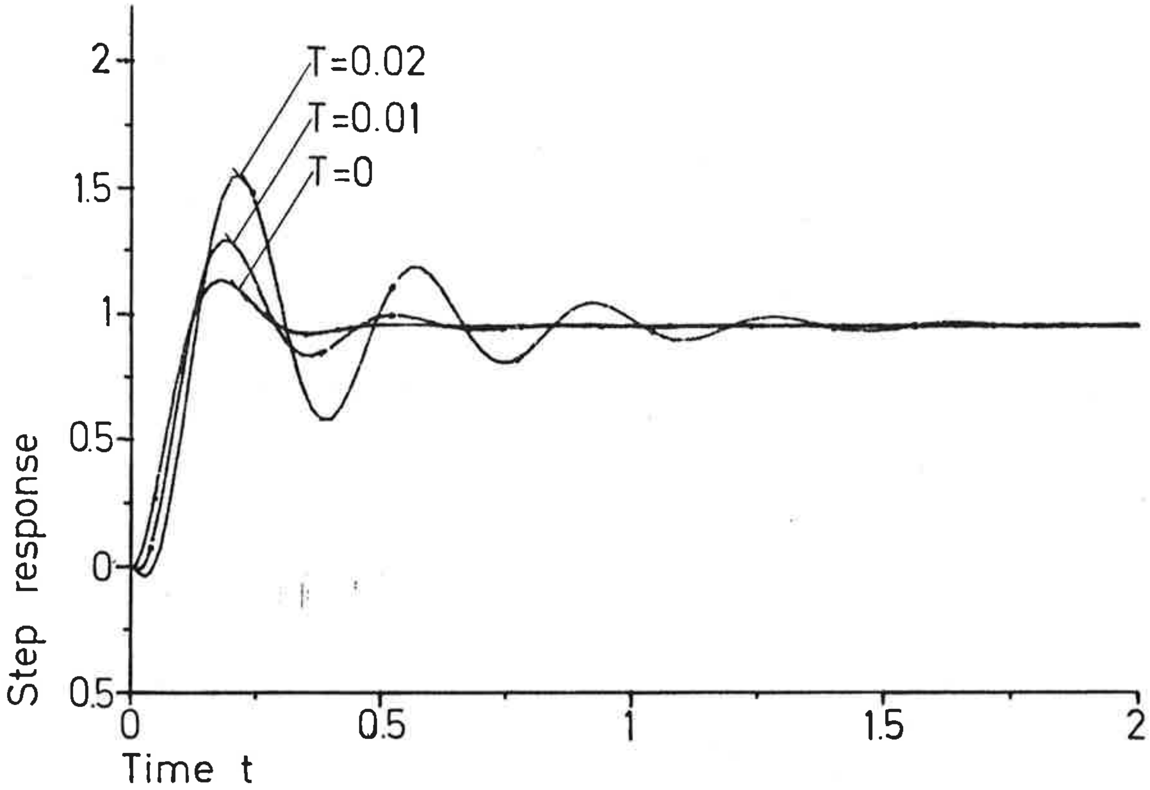


Figure 5. Unit step responses for closed loop systems with transfer function (3.3) □

When designing a self-tuning regulator it is useful to consider the particular application carefully and decide upon a design method which is suitable for the particular problem if a model for the process and its environment are known. A parameter estimation scheme which works well for the particular problem should also be chosen before the details of the design are considered.

4. INTEGRAL ACTION

The reason for introducing reset and integral action is to eliminate steady state errors in the closed loop system. Steady state errors can be generated by many different mechanisms, calibration errors, nonlinearities, load disturbances etc. Irrespective of the origin of the disturbances it has been found empirically that the errors can be eliminated simply by letting the feedback signal have a term which is proportional to the integral of the error. It is also well known that integral feedback can lead to difficulties. It destabilizes the system and may lead to oscillations with large amplitudes. Since the integral is an unstable system it may happen that the integral can assume very large values if the control signal saturates (due to nonlinearities or manual control) when there is an error. This is called *reset windup*. Special precautions have to be taken in ordinary regulators to avoid windup of the integrator. There are several ways to provide reset in self-tuning regulators. Since there is no method which is uniformly best a few different schemes will be discussed.

Automatic reset provided by the STR

Since many self-tuning regulators estimate models of the environment it can be expected that the self-tuner will attempt to model slowly drifting disturbances and compensate for them by introducing integral action automatically. This is indeed the case for many configurations. It is easy to check if a particular self-tuner has this ability simply by investigating possible stationary solutions when there is an off-set or a drifting disturbance. A typical example is given below.

EXAMPLE 3

Consider the simple implicit self-tuner discussed in Åström and Wittenmark (1973), which is based on least squares parameter estimation and minimum variance control. The self-tuner is based on the model

$$y(t+k) = Ru(t) + Sy(t)$$

The conditions for an equilibrium of the parameter estimates is that

$$\frac{1}{N}\sum_{t=1}^{N} y(t+\tau)\, y(t) = 0 \qquad \tau = k,\ldots,k + \deg S$$

$$\frac{1}{N}\sum_{t=1}^{N} y(t+\tau)\, u(t) = 0 \qquad \tau = k,\ldots, k + \deg R$$

These conditions can clearly not be satisfied unless the mean value of the output y is zero. When there is an off-set or a disturbance the parameter estimates will assume values such that $\hat{R}(1) = 0$.

□

Another example which shows that reset can also be provided automatically in explicit algorithms is given in Åström (1979b).

In many cases it is thus not necessary to make any special provisions to obtain re-set action. The self-tuner will automatically introduce reset when needed. The main drawback of such a scheme is that the response of the system to sudden variations in the load level may be slow. The problem is particularly severe if the nature of the disturbances change drastically with time. The method is also inconvenient when the STR is used as a tuner. It could easily happen that the disturbances encountered during the tuning have a small low frequency component. The regulator obtained will then not necessarily have sufficient gain at low frequencies. When integration is provided automatically it is necessary to introduce facilities to avoid reset windup i.e. to ensure that the regulator state which correspond to the integral will not grow without bounds when the output saturates. One possibility is to replace the control law (2.4) by

$$u(t) = \mathrm{sat}[Tu_c(t) - Sy(t) - (R-r_0)u(t)]/r_0 \tag{4.1}$$

where sat is a saturation function which saturates before the actuator. Another way to avoid reset windup is discussed in Andersson and Åström (1978).

Estimation of a Bias

A simple way to model the off-set errors is to replace the model (2.1) by

$$A\,y(t) = Bu(t) + b \tag{4.2}$$

where the bias term b represent the errors. With a model like (4.2) it is natural to estimate the bias b and to compensate for it. Such a scheme was proposed by Clarke and Gawthrop (1979). An advantage is that the estimation of b is simple. The drawbacks are that an extra parameter has to be estimated. The estimate $\hat{b}$ will converge slowly unless special precautions are taken. If forgetting factors are used it is useful to have separate forgetting factors for $\hat{b}$ and the other parameters. See Åström (1979b). If bias is eliminated in this way it is not possible to use the STR simply as a tuner because there will be no reset when estimation is switched off.

Forced Integral Action by Use of a Special Model Structure

One possibility to obtain reset is to choose a model structure so that the regulator designed from the model will always contain an integrator. For explicit self-tuners based on pole-placement design this can be done by using the lack of uniqueness in the equation (2.3) to impose the condition that $1 - q^{-1}$ should be a factor of R. This can always be done. For implicit self-tuners integral action can be imposed by replacing the model (2.5) by

$$PT_1\, y(t) = R\nabla u(t-k) + Sy(t-k) + b \tag{4.3}$$

where $\nabla = 1 - q^{-1}$.

The control law (2.6) is then replaced by

y(t)

$$R\nabla u(t) = Tu_c(t) - Sy(\cancel{t-k}) \tag{4.4}$$

Notice that it follows from the design procedure that T(1) = S(1). Notice also that it is useful to include estimation of the bias b although the estimate is not used when calculating the control signal.

The main advantage of this scheme is that the controller will always have integral action. If the STR is used as a tuner the regulator obtained when the tuning is switched off will always have integral action. A drawback is that there will be one additional mode in the controller. In the pole-placement design it is then an additional pole to position. This pole is not entirely trivial to choose. If it is placed at the origin the controller will have an unnecessarily high gain. The scheme also requires special tricks to avoid reset windup. Another drawback with the scheme is that the self-tuner may try to eliminate the integral action when it is not needed. The estimated polynomial S then has the factor $\nabla = (1 - q^{-1})$. This means that the regulator transfer function has an unstable mode which is cancelled, and the system will be unstable. An example where this happens is discussed in Åström and Gustavsson (1978).

Integration in Inner Loop

Steady state errors can be avoided by the scheme shown in Fig.6. The process is provided with a fixed gain feedback loop with integrating action.

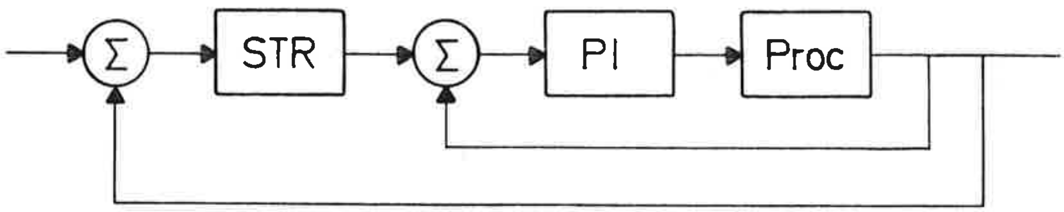


Figure 6. Block-diagram of a regulator with an integrating regulator in an inner-loop and an outer-loop with self-tuning.

The use of an inner loop was originally proposed by Wittenmark (1973). The arrangement shown in Fig.6 was applied by Dumont and Belanger (1978). One drawback of the scheme is that it may be difficult to tune the regulator in the inner loop. Another drawback is that it is not good practice to have integration in an inner loop even for systems with fixed parameters.

Integration in an Outer Loop

Another possibility to avoid steady state errors is shown in Fig.7. A self-tuner is first connected to the process. An outer loop with integral action is then introduced. Since the self-tuner makes the inner loop invariant to changes in process dynamics it is possible to have fixed gain in the outer loop.

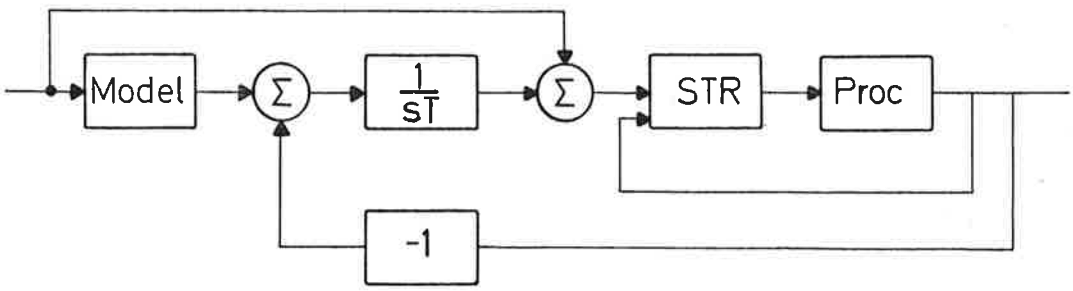


Figure 7. Block-diagram of a regulator structure with a self-tuner in an inner loop and an outer loop with integration.

The approach is particularly attractive for self-tuners whose specifications are directly related to properties of the closed loop transfer function because the outer loop gain can then be set automatically. Another advantage is that standard methods can be used to avoid reset windup. The major disadvantage is that is is not convenient to use the scheme for self-tuners whose performance are not directly related to the bandwidth of the closed loop system. In such a case the integrator gain cannot be set automatically.

5. ESTIMATOR WINDUP

The problem of windup can occur whenever there is an unstable mode in a regulator. In a self-tuning regulator there may be unstable modes associated with the parameter estimator. The problem is closely connected with the design of the estimator, and the way in which control signals are limited.

Input Saturation

There are several mechanisms which can cause instability. Consider for example the case when the actuator saturates. If no precautions are taken it could easily happen that the control signal calculated by the regulator is outside the saturation limits. The estimated process model will then have too low gain. The calculated controller gain will be too large. Saturation effects will be even more pronounced etc. This simple intuitive argument has been supported by simulations. In simple cases it can also be verified analytically. There is a simple remedy. Introduce a saturation in the controller where the limits are set tighter than the actuator saturation. e.g. as in (4.1). The parameter estimator will then have a faithful representation of the actual process variable.

Covariance Windup

Another mechanism which can cause instability will now be discussed. For this purpose the equations describing the parameter estimator are needed. They are

$$\theta(t+1) = \theta(t) + P(t+1)\,\varphi(t+1)\,\varepsilon(t+1) \tag{5.1}$$

$$P(t+1) = [P(t) - P(t)\,\varphi(t)\,R(t)\,\varphi^T(t)\,P(t)]/\lambda \tag{5.2}$$

$$R(t) = [\lambda + \varphi^T(t)\,P(t)\,\varphi(t)]^{-1} \tag{5.3}$$

Consider the equation (5.2). The negative term in the right hand side represents the reduction in uncertainty due to the last measurements. If the control signal and the output are zero the vector $P(t)\varphi(t)$ will then be zero. There will not be any changes in the parameter estimate and the negative term in the right hand side of (5.2) will be zero. The equation (5.2) then reduces to

$$P(t+1) = \frac{1}{\lambda} P(t)$$

and the matrix P will thus grow exponentially if $\lambda < 1$. If there are no changes for a long time the matrix P may thus become very large. Since P represents the gain in the parameter estimator (5.1) a change in the command signal may then lead to large changes in the parameter estimates and in the process output. The large values of the matrix P may also lead to numerical problems. The problem will occur whenever the vector $P(t)\varphi(t)$ is zero or sufficiently small over a period of time. The problem is closely associated with identifiability conditions and the selection of the forgetting factor λ.

Excitation of the Process

Identifiability depends on the input signal u and the number of estimated parameters. In typical regulation problems where the system is continuously excited by the disturbances the problem will not occur provided that the number of estimated parameters is not too high. The problem will be much severe in a typical servo problem where the major excitation comes from the command signal which may be constant for long periods of time. The situation is similar for regulation problems where the major disturbances are constant over long time periods. One possibility to ensure that the process is properly excited is to introduce perturbation signals or to use a dual control law.

The Forgetting Factor

Covariance windup is closely related to the choice of the forgetting factor λ. If $\lambda = 1$ the problem will not occur. For $\lambda = 1$ the estimator gain will, however, decrease and the estimator will be very sluggish. There are, however, several other possibilities to obtain estimators with non-decreasing gain. The matrix P could simply be chosen as a fixed matrix. This is commonly done in model reference systems. Another possibility is to replace equation (5.3)

$$P(t+1) = P(t) + P(t)\,\varphi(t)\,R(t)\,\varphi^T(t)P(t) + R_1 \qquad (5.4)$$

In this case the matrix P will grow linearly instead of exponentially when $P\varphi$ is zero. A third possibility is to replace the equation (5.3) by

$$P^{-1}(t) = [\alpha I + \sum_{k=1}^{t} \lambda^{t-k}\,\varphi(k)\varphi^T(k)]^{-1}$$

where α is a small number. This ensures that P stays bounded. The size of P is determined by α.

A fourth possibility is to simply put a bound on P e.g. by restricting it so that the trace of the matrix P is constant in each iteration. This has been proposed by Irving (1979).

A fifth possibility is to adjust the forgetting factor automatically. It can e.g. be chosen as

$$\lambda = 1 - \alpha\,\varepsilon^2/\overline{\varepsilon^2}$$

where $\overline{\varepsilon^2}$ is the mean value of ε^2 over a certain period. More complicated formula for adjusting λ have also been proposed. See Fortescue et al (1978). An automatic adjustment of λ does not guarantee that the matrix P stays bounded. The period where P has a reasonable size may, however, increase substantially.

It has also been proposed to eliminate covariance windup by stopping the updating of θ and P when $P\varphi$ or ε is sufficiently small. See Egardt (1979).

In Goodwin et al (1978) it is proposed to analyse the conditioning number of the matrix P and to switch to a stochastic approximation algorithm when the matrix P becomes poorly conditioned.

6. THE PARAMETERIZATION PROBLEM

A mathematical model can be parametrized in many different ways. The choice of parameters is important for the design of self-tuners. For example when discussing implicit and explicit algorithms for self-tuning servos in section 2 it was found that the algorithm could be simplified substantially if the model was parametrized in the regulator parameters.

Although the parametrization problem is important it has been given little attention in literature. The general tendency, both as far as MRAC and STR are concerned, is to parametrize in such a way that the estimation problem becomes simple e.g. linear in the parameters. In Åström (1979c) an example is given which shows that it may be advantageous to use other parametrizations.

The parametrization of the minimum variance self-tuner (Åström and Wittenmark (1973)) or its model reference equivalent has been given some attention. For minimum variance regulation the variable y is often chosen as the control error. Since $PT_1 = 1$ for minimum variance control the estimation model (2.5) then reduces to

$$y(t+k) = S(q^{-1})\, y(t) + R(q^{-1})\, u(t) \tag{6.1}$$

and the control law becomes

$$\hat{R}(q^{-1})\, u(t) = -\hat{S}(q^{-1})\, y(t) \tag{6.2}$$

This control law has one redundant parameter because the polynomials $\hat{R}$ and $\hat{S}$ can be multiplied by a constant without changing the control law. The redundant parameter can be eliminated by reparametrizing the estimation model (6.1) as

$$\begin{aligned} y(t+k) &= r_0[u(t) + r_1'\, u(t-1) + \dots + r_{n_R}'\, u(t-n_R)] \\ &+ s_0\, y(t) + s_1 y(t-1) + \dots + s_{n_S}\, y(t-n_R) \end{aligned} \tag{6.3}$$

The control law (6.2) then becomes

$$\begin{aligned} u(t) &= -\frac{1}{\hat{r}_0}[\hat{s}_0\, y(t) + \dots + \hat{s}_{n_S}\, y(t-n_S)] \\ &\quad - \hat{r}_1'\, u(t-1) - \dots - \hat{r}_{n_R}'\, u(t-n_R) \end{aligned} \tag{6.4}$$

It is shown in Åström and Wittenmark (1973) that the estimate $\hat{r}_0$ can be fixed apriori if

$$0.5 \leq \hat{r}_0/r_0 < \infty$$

without influencing the equilibrium condition. In Ljung (1977) it is shown that if the algorithm converges for $\hat{r}_0 = r_0$ it will still converge if (6.5) holds. The convergence rate is, however, influenced by $\hat{r}_0$. It is fastest for $\hat{r}_0 = r_0$

For minimum variance self-tuners either of the models can be used. The algorithm based on (6.4) with fixed $\hat{r}_0$ is most robust provided that apriori knowledge to choose r_0 subject to (6.5) is available. If this is not possible the parameters in (6.1) can be estimated. Identifiability is poor because of the feedback. The estimates of the parameter combinations r_i/r_0 and s_i/r_0 converge as 1/t. The estimate of r_0 converges, however, at a slower rate. Algorithms which treat r_0 in a special way are therefore also used.

7. CONCLUSIONS

The word self-tuning regulator may lead to the false conclusion that such regulators can be switched on and used blindly without any apriori considerations. This is not true. The self-tuning regulator is a fairly complex control law. A proper design involves the choices of gross features like underlying design and estimation methods and decisions on details like initialization, selection of parameters, and safeguard methods. Proper choices require insight and knowledge. There are known cases where bad choices have been disastrous.

There has recently been considerable progress in the theory of adaptive control. Stability results have been proven for simple self-tuners (implicit minimum variance and pole-placement) connected to linear systems. The theory requires assumptions which are hard to verify in a practical situation. The theory is also limited to simple self-tuners. The theory required to use self-tuners confidently is thus not available A cautious person would then perhaps be inclined not to try a self-tuner. To get some perspective it may be useful to reflect on the role of theory in similar situations. The properties of the closed loop system obtained when a PID regulator is connected to a linear system are fully understood theoretically provided that the regulator operates in the linear region. As soon as nonlinearities associated with gap, saturation and anti-windup are introduced there is, however, little theory which tell theoretically what happens. In spite of this,large systems with many interconnected PID regulators are designed, sold, commissioned and used routinely.

Based on attempts to develop suitable theory and experiences from a few applications I believe, however, that self-tuning regulators can and will be used profitably, even if all their properties are not fully understood theoretically. I hope that this paper may inspire some of you to acquire the appropriate knowledge and try some schemes. I also hope that some of you will tackle the important theoretical problems that remain.

8. ACKNOWLEDGEMENTS

This work was supported by the Swedish Board of Technical Development (STU) under contract STU 78-3763.

9. REFERENCES

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

Åström K J (1974): A self-tuning regulator for non-minimum phase systems. Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden, Report TFRT-7411.

Åström K J (1979a): Self-tuning regulators - design principles and applications. Proc Yale Workshop on Applications of Adaptive Control, Yale University.

Åström K J (1979b): Simple self-tuners I. Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN:LUTFD2/(TFRT-7184))1-052/(1979).

Åström K J (1979c): New implicit adaptive pole-zero-placement algorithms for non-minimum phase systems. Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN:LUTFD2/(TFRT-7172)/1-121/(1979).

Åström K J and Gustavsson I (1978): Analysis of a self-tuning regulator in a servo-loop. Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN:LUTFD2/(TFRT-3150)/1-058/(1978).

Åström K J and Wittenmark B (1973): On self-tuning regulators. Automatica 9, 185-199.

Åström K J and Wittenmark B (1979): Self-tuning controllers based on pole-zero placement. Submitted to IEE Proceedings.

Clarke D W and Gawthrop B A (1975): Self-tuning controller. Proc IEE 122, 929-934.

Clarke D W and Gawthrop P J (1979): Implementation and application of microprocessor based self-tuners. Preprints 5th IFAC Symposium on Identification and System Parameter Estimation, Darmstadt, September 1979, p 197-208.

Dumont G A and Bélanger P R (1978): Self-tuning control of a titanium dioxide kiln. IEEE Trans AC-23,532-537.

Egardt B (1979). Stability of adaptive controllers. Lecture Notes in Control and Information Sciences, Vol 20, Springer-Verlag, Berlin.

Fortescue T R, Kershenbaum L S and Ydstie B E (1978): Implementation of self-tuning regulators with variable forgetting factors. Report, Dept of Chemical Eng and Chemical Technology, Imperial College, London.

Goodwin G C, Ramadge P J and Caines P E (1978): Discrete time stochastic adaptive control. Report Div of Applied Science, Harward University Cambridge Mass., USA, December, 1978.

Irving E (1979): Improving power network stability with adaptive generator control - new developments. Proc Workshop on Applications of Adaptive Control, Yale University, New Haven.

Ljung L (1977): Analysis of recursive stochastic algorithms. IEEE AC-22, 551-575.

Morse A A (1979): Global stability of parameter adaptive control systems. S&IS Report 7902R, Yale University, March, 1979.

Narendra K S, Lin Y-H, and Valavani L S (1979): Stable adaptive controller design - Part II. Proof of Stability. S&IS Report 7904, Yale University, April, 1979.

Peterka V (1979: Adaptive digital regulation of noisy systems. Proc 2nd IFAC Symp on Identification and Process Parameter Estimation, Prague.

Wellstead P E, Prager D and Zanker P (1979): Pole assignment self-tuning regulator. Proc IEE, 126, 781-787.

Wittenmark B (1973b): Self-tuning regulators. Report TFRT-7321 (thesis). Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden.

DESIGN PRINCIPLES FOR SELF-TUNING REGULATORS

DESIGN PRINCIPLES FOR SELF-TUNING REGULATORS

1. INTRODUCTION
2. SELF-TUNING REGULATORS
   - PRINCIPLES
   - A SELF-TUNING SERVO
   - EXPLICIT AND IMPLICIT ALGORITHMS
   - RELATIONS TO MRAS
   - A LQG SELF-TUNER
3. PRACTICAL ASPECTS
4. INTEGRAL ACTION
5. ESTIMATOR WINDUP

DRIVING FORCES FOR ADAPTIVE CONTROL

SUCCESSFUL APPLICATIONS HAVE BEEN DEMONSTRATED

EASY TO IMPLEMENT WITH MINI- OR MICRO COMPUTERS

CONVENIENT TO USE

INCREASED REQUIREMENTS FOR CONTROL PERFORMANCE

PROGRESS IN THEORY

MANY RESEARCH PROBLEMS LEFT

A SELF-TUNING SERVO

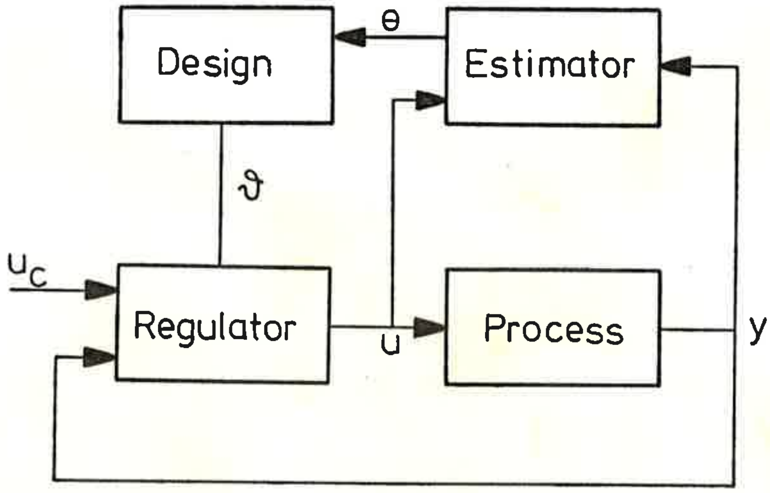
PROBLEM FORMULATION

DESIGN FOR KNOWN PARAMETERS

EXPLICIT AND IMPLICIT METHODS

RELATIONS TO MRAS

P R I N C I P L E S



MANY POSSIBILITIES

DESIGN METHODS

- MINIMUM VARIANCE
- LQG
- POLE-PLACEMENT
- PHASE- AND GAIN MARGINS

ESTIMATION METHODS

- STOCHASTIC APPROXIMATION
- RECURSIVE LEAST SQUARES
- EXTENDED LEAST SQUARES
- MULTI-STAGE LEAST SQUARES
- INSTRUMENTAL VARIABLES
- RECURSIVE MAXIMUM LIKELIHOOD

LQG SELF-TUNER

DATA: GIVEN h AND ϱ ← ONE KNOB CONTROL WEIGHT ϱ

1. ESTIMATE PARAMETERS IN THE MODEL

   $$A(q^{-1})\,y(t) = B(q^{-1})\,u(t) + C(q^{-1})\,e(t)$$

   BY ELS OR RML.

2. FIND CONTROL LAW WHICH MINIMIZES

   $$J = \lim \frac{1}{N} \sum_{0}^{N} \left[y^2(t) + \varrho u^2(t) \right]$$

   $$P(z)\,P(z^{-1}) = \varrho \hat{A}(z)\,\hat{A}(z^{-1}) + \hat{B}(z)\,\hat{B}(z^{-1})$$

   $$\hat{A}R + \hat{B}S = \hat{C}P$$

3. USE CONTROL LAW

   $$Ru = \hat{C}y_r - Sy$$

NOTE: SAMPLING PERIOD h CAN BE TUNED AUTOMATICALLY.

POLE-PLACEMENT DESIGN

DATA: PROCESS MODEL $G = B/A$

DESIRED RESPONSE $G_D = Q/P$

OBSERVER POLYNOMIAL T_1 DEG T_1 → = DEG A KALMAN; → = DEG A-1 LUENBERGER

REQUIREMENTS

1. DEG P - DEG Q > DEG A - DEG B.
2. $B = B^+B^-$, $Q = Q_1B^-$
   B^+ STABLE AND WELL DAMPED, B^- UNSTABLE OR POORLY DAMPED.
3. A AND B NO COMMON FACTORS.

DESIGN

1. CHOOSE $T = T_1Q_1$ AND SOLVE THE DIOPHANTINE EQUATION

$$AR + BS = PB^+T_1. \quad (AR_1 + B^-S = PT_1)$$

2. CONTROL LAW IS THEN

$$Ru = Ty_R - Sy. \quad (R = R_1B^+, \quad T = Q_1T)$$

ALTERNATIVE INTERPRETATION

$$u = \frac{Q}{P} \cdot \frac{A}{B} y_R - \frac{S}{R}\left[y - \frac{Q}{P} y_R\right]$$

$\frac{Q}{P}$ — DESIRED RESPONSE

$\frac{A}{B}$ — INVERSE MODEL

$\frac{Q}{P} y_R$ — DESIRED OUTPUT

POLE PLACEMENT

$$Q = Q_1 B^-, \quad B = B^+ B^-, \quad T = T_1 Q_1$$

$$AR + BS = PB^+ T_1$$

SPECIAL CASES WHICH AVOID FACTORIZATION:

CASE 1 ALL ZEROS CANCELLED

$$Q = 1 \Rightarrow Q_1 = B^- = 1, \quad B^+ = B, \quad T_1 = T$$

$$AR + BS = PTB \Rightarrow B \text{ DIV } R, \quad R = R_1 B$$

$$AR_1 + S = PT$$

$$R_1 B u = T y_R - S y$$

CASE 2 NO ZEROS CANCELLED

$$Q = B \Rightarrow Q_1 = B^+ = 1, \quad Q_2 = B, \quad T = T_1,$$

$$AR + BS = PT$$

$$Ru = T y_R - S y$$

EXPLICIT DESIGN

INDIRECT CONTROL

1. ESTIMATE MODEL $Ay = Bu$ BY LS

2. FACTOR B-POLYNOMIAL AS $B = B_1 B_2$, B_1 STABLE

3. PUT $T = T_1 Q_1$ AND SOLVE $AR + BS = PBT_1$ FOR R AND S

   (T_1: Desired observer poles; Q_1: Desired additional zeros)

4. USE CONTROL LAW $Ru = Ty_R - Sy$

NOTICE: CLOSED LOOP TRANSFER FUNCTION IS

$$\frac{Q_1 B_2}{P}$$

SPECIAL CASE
$Q_1 = 1$, $B_2 = B$

IMPLICIT ALGORITHMS

DIRECT CONTROL

EXAMPLE

$$\frac{BT}{AR+BS} = \frac{1}{P}$$

DESIGN IDENTITY: $AR + BS = PBT$

$$PTy = \frac{A}{B} Ry + Sy = Ru + Sy$$

ALGORITHM:

STEP 1: ESTIMATE PARAMETERS IN MODEL

$$PTy = Ru + Sy$$

BY LS.

STEP 2: USE CONTROL LAW

$$Ru = Ty_R - Sy$$

NOTE. THE POLYNOMIAL B IS CANCELLED

BECAUSE $R = R_1 B$

S T R & M R A S

MODEL REFERENCE:

MIT RULE:

$$\frac{d\vartheta}{dt} = - e \frac{\partial e}{\partial \vartheta}$$

SENSITIVITY DERIVATIVE

AUGMENTED ERROR

IMPLICIT POLE-PLACEMENT STR

$$\vartheta(t+1) = \vartheta(t) + P(t+1)\,\varphi(t+1)\,\varepsilon(t+1)$$

$$\varepsilon(t) = PTy(t) - Ry(t-k) - Sy(t-k) =$$

$$= PTy(t) - \varphi^T(t)$$

$$\Rightarrow \quad \varphi(t) = - \frac{\partial \varepsilon(t)}{\partial \vartheta}$$

$$\vartheta(t+1) = \vartheta(t) - P(t+1)\,\varepsilon(t+1)\,\text{GRAD}_{\vartheta}\,\varepsilon(t+1)$$

$\varepsilon(t)$ = PREDICTION ERROR