ON-LINE PROCESS SIMULATION TECHNIQUES IN INDUSTRIAL CONTROL

ON-LINE PROCESS SIMULATION TECHNIQUES IN INDUSTRIAL CONTROL

Including Parameter Identification and Estimation Techniques

Proceedings of the Eleventh Annual
Advanced Control Conference
Sponsored by
The Purdue Laboratory for Applied
Industrial Control, Purdue University
and
CONTROL ENGINEERING

E.J. KOMPASS and T.J. WILLIAMS, Editors

West Lafayette, Indiana
September 30 - October 2, 1985

CONTROL
ENGINEERING

Foreword

The Eleventh Annual Advanced Control Conference sponsored by CONTROL ENGINEERING Magazine and the Purdue Laboratory for Applied Industrial Control was held at Purdue University, September 30-October 2, 1985. The general objective of these conferences is to further the art of control and instrumentation systems engineering through a presentation of a combination of basic tutorial lectures and a number of short papers describing specific applications of advanced control techniques.

Each year the conference has been limited to a closely defined area in the broad subject of advanced control. The first ten conferences dealt with adaptive and nonlinear control techniques, multivariable control systems, hierarchical and distributed control, the conservation of energy in manufacturing processes through the application of industrial control, on-line optimization techniques, man-machine interfaces for industrial control, computer software for industrial control, on-line production scheduling and plant-wide control, learning systems and pattern recognition (applying artificial intelligence to industrial control) and a review of the 25 year history of computer control systems with a forecast of expected future developments.

This publication, "On-Line Process Simulation Techniques in Industrial Control," was preprinted for distribution to attendees of the Eleventh Annual Advanced Control Conference, and contains the complete text of all the papers presented at the conference.

The editors acknowledge with gratitude the major contributions of Mrs. Sharon K. Whitlock, Administrative Assistant for the Purdue Laboratory for Applied Industrial Control and Mr. Henry M. Morris, Technical Editor of CONTROL ENGINEERING Magazine for the arrangement of the Eleventh Annual Advanced Control Conference and the production of this volume of papers.

Dr. Theodore J. Williams
Director, Purdue Laboratory for
Applied Industrial Control

Mr. Edward J. Kompass
Editor, Control Engineering

I. Tutorial Papers

TABLE OF CONTENTS

I. Tutorial Papers

II. Application Papers

RECURSIVE TECHNIQUES FOR IDENTIFYING DYNAMIC SYSTEMS

Lennart Ljung
Division of Automatic Control
Dept of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Abstract

Building models of systems on-line, while the data is being recorded, is becoming an important and useful tool for process control and process monitoring. Adaptive control and failure detection are typical applications. In this contribution we display the basic ideas, algorithms and techniques for such recursive identification. The emphasis is on kinship to off-line identification and on the role of a predictor function for developing the algorithms.

親戚关系

1 INTRODUCTION

System identification concerns the problem of building mathematical models of dynamical systems based on input-output measurements. Recursive or on-line techniques for system identification perform this model building in real time, and update the model continuously, so as to provide information about current system properties. In this contribution we shall give a brief state-of-the art account of the theory and the algorithms for such recursive identification.

Systems and models

The notion of systems plays an important role in modern science and technology. Many problems in various fields are solved in a systems oriented framework. Subjects like control theory, communication theory and operations research tell us how to determine suitable regulators, filters, decision rules etc. Such theory assumes that a model is available of the system in question. The applicability of the theory is thus critically dependent on the availability of good models.

How does one construct good models of a given system? This question about the interface between the real world and the world of mathematics thus becomes crucial. The general answer is that we have to study the system experimentally and make some inference from the observations. In practice there are two main routes. One is to split up the system, figuratively speaking, into subsystems, whose properties are well understood from previous experience. This basically means that we rely upon "laws of Nature" and other well established relationships, that have their roots in earlier empirical work. These subsystems are then joined together mathematically, and a model of the whole system is obtained. This route is known as modelling, and does not necessarily involve any experimentation on the actual system. When a model is required of a yet unconstructed system (such as a projected aircraft) this is the only possible approach.

The other route is based on experimentation. Input and output signals from the system are recorded and are then subjected to data analysis in order to infer a model of the system. This route is known as identification. It is often advantageous to try to combine the approaches of modelling and identification in order to maximise the information obtained from identification experiments and to make the data analysis as sensible as possible.

The use of models

Models are required for many problems in process industry and elsewhere. The design of a control system is based on models of the system's behaviour. Signal processing of raw data obtained from a collection of sensors should be derived from a model of the relationships between the measured signals. Predictions of yield and product quality go back to models of the production facility. Models thus form an indispensable ingredient in many decisions related to the operation of a process. It should be said though, that such models need not be formalized in mathematical terms - many decisions are no doubt based on "intuitive" or "mental" models of the process behaviour.

On-line models

In many cases it is necessary, or useful, to have a system model available on-line, while the system is in operation. The need for such an on-line model construction typically arises since a model is required in order to take some decision about the system. For example, a control design based on a continuously updated model is known as adaptive control, (See Åström (1983) for a recent survey) which is a promising approach to handle time-varying plants.

Another important application is monitoring and failure detection. In complex technological systems it has become increasingly important to monitor the system's behaviour by intelligent processing of measured signals. By studying models of the system's current properties it is possible to detect sudden changes due to failures of various times, as well as continuous deterioration due to wear etc.

The on-line computation of the model must also be done in such a way, that the processing of the measurements from one sample can, for sure, be completed during one sampling interval. Otherwise the model-building cannot keep up with the information flow.

Identification techniques that comply with this requirement will here be called Recursive identification methods, since the measured input-output data are processed recursively (sequentially) as they become available. Other commonly used terms for such techniques are on-line or real-time identification, adaptive parameter estimation or sequential parameter estimation.

Paper outline

The present contribution aims at displaying basic principles and techniques for recursive identification, as well as quoting their basic analytical properties. The presentation follows the framework of Ljung and Söderström (1983). Basic features are that we stress the close relationship of adaptive, recursive identification to off-line identification methods, and that we use the predictor function as a basic model concept. Section 2 deals with the model concept, while off-line identification is discussed in Section 3.The recursive techniques are outlined in Section 4. Implementation and practical issues are dealt with in Sections 5 and 6.

2 TIME-DOMAIN MODELS OF DYNAMICAL SYSTEMS

Time-domain Models of Dynamical Systems

Describing dynamical systems in the time domain allows a considerable amount of freedom. Usually, differential (partial or ordinary) equations are used to describe the relationships between inputs and outputs. In discrete-time (sampled-data systems) difference equations are used instead. The question of how to describe properties of various disturbance signals also allows for several different possibilities. Here, we shall list a few typical choices, confining ourselves to the case of linear, discrete time models. The basic notions carry over to nonlinear models also, however.

The word "model" is sometimes used ambiguously. It may mean a particular description (with numerical values) of a given system. It may also refer to a description with several coefficients or parameters that are not fixed. In the latter case, it is more appropriate to talk about a model set: a set of models that is obtained as the parameters range over a certain domain.

Linear difference equations. Let the relationship between the input sequence $\{u(t)\}$ and the output sequence $\{y(t)\}$ be described by

$$y(t)+a_1y(t-1)+....+a_ny(t-n)=b_1u(t-1)+....+b_mu(t-m) \tag{2.1}$$

Here the coefficients a_i and b_i are adjustable parameters. (A multivariable description would be quite analogous, with a_i and b_i as matrices.) We shall generally denote the adjustable parameters by a vector Θ:

$$\Theta = (a_1...a_n\ b_1...b_m)^T. \tag{2.2}$$

If we introduce the vector of lagged inputs and outputs

$$\varphi(t)=\big(y(t-1)...-y(t-n)u(t-1)...u(t-m)\big)^T, \tag{2.3}$$

then equation (1) can be rewritten in the more compact form

$$y(t) = \varphi^T(t)\Theta \tag{2.4}$$

In (2.1) or (2.4), the relationship between inputs and outputs is assumed to be exact. This may not be realistic in a number of cases. Then we may add a term v(t) to (2.1) or (2.4):

$$y(t) = \varphi^T(t)\Theta + v(t) \tag{2.5}$$

that accounts for various noise sources and disturbances that affect the system, as well as for model inaccuracies. This term can be further modelled, typically by describing it as a stochastic process with certain properties. The simplest model of that kind is to assume $\{v(t)\}$ to be white noise, i.e. a sequence of independent random variables with zero mean values. However, many other possibilities exist. Among the most common models is the following one.

ARMAX models. If the term $\{v(t)\}$ in (2.5) is described as a moving average (MA) of white noise $\{e(t)\}$ we have a model

$$y(t)+a_1y(t-1)+...+a_ny(t-n)=b_1u(t-1)+...+$$
$$+b_mu(t-m)+e(t)+c_1e(t-1)+...+c_ne(t-n) \tag{2.6}$$

Such a model is known as an ARMAX model.

Output error models. Instead of adding the disturbance v(t) to the equation as in (2.5), it can be added as an output measurement error:

$$y(t) = x(t) + v(t) \tag{2.7a}$$

$$x(t)+f_1x(t-1)+...+f_nx(t-n)=b_1u(t-1)+...+b_mu(t-m) \tag{2.7b}$$

Such models are often called output error models. The "noise-free output" x(t) is here not available for measurement, but given (2.7b) it can be reconstructed from the input. We denote by $x(t,\Theta)$ the noise free output that is constructed using the model parameters

$$\Theta = (f_1...f_n\ b_1\ ...b_m)^T, \tag{2.8}$$

i.e.

$$x(t,\Theta)+f_1x(t-1,\Theta)+...+f_nx(t-n,\Theta)=b_1u(t-1)+...+b_mu(t-m).$$

With

$$\varphi(t,\Theta)=\big(-x(t-1,\Theta)...-x(t-n,\Theta)u(t-1)...u(t-m)\big)^T, \tag{2.9}$$

(2.7) can be rewritten as

$$y(t) = \Theta^T\varphi(t,\Theta) + v(t) \tag{2.10}$$

Notice the formal similarity to (2.5) but the important computational difference! Also in output error models the character of the additive noise v(t) can be further modelled.

State-space models. A common way of describing stochastic, dynamical systems is to use state-space models. Then the relationship between input and output is described by

$$\begin{aligned} x(t+1) &= F(\Theta)x(t) + G(\Theta)u(t) + w(t) \\ y(t) &= H(\Theta)x(t) + e(t), \end{aligned} \tag{2.11}$$

where the noise sequences w and e are assumed to be independent at different time instants and have certain covariance matrices. Unknown, adjustable parameters Θ may enter the matrix elements in F, G and H in an arbitrary manner. These may, e.g., correspond to canonical parametrizations (canonical forms) or to physical parameters in a time-continuous state space description that has been sampled to yield (2.11).

A general linear model Transfer function Model

With the shift operator q, qu(t)=u(t+1) a general linear model can be written

$$y(t) = G(q,\Theta)u(t) + H(q,\Theta)e(t) \tag{2.12}$$

where

$$G(q,\Theta) = \sum_{k=1}^{\infty} g(k,\Theta)q^{-k}$$

$$H(q,\Theta) = 1 + \sum_{k=1}^{\infty} h(k,\Theta)q^{-k}$$

and $\{e(t)\}$ is assumed white noise.

Models and Predictors

The list of potential models and model sets can be made long. For our purposes it is useful to extract the basic features of a models, so as to allow for a treatment of model sets in general. First we introduce the following notation:

$M(\Theta)$: a particular model, corresponding to the parameter value Θ

M: a set of models: $M=\{M(\Theta)\,|\,\Theta\in D_M \subset R^d\}$

Z^t: the set of measured input-output data up to time t:

$$Z^t=\{u(1),y(1),u(2),y(2)....u(t),y(t)\}$$

Similarity, u^t and y^t denote the input sequence and the output sequence, respectively, up to time t.

The various models that can be used for dynamical systems all represent different ways of thinking and representing relationships between measured signals. They have one feature in common, though. They all provide a rule for computing the next output or a prediction (or "guess") of the next output, given previous observations. This rule is, at time t, a function from Z^{t-1} to the space where y(t) takes its values (R^p in general). It will also be parametrized in terms of the model parameter Θ. We shall use the notation

$$\hat{y}(t|\Theta) = g_M(\Theta;t,Z^{t-1}) \tag{2.13}$$

for this mapping. The actual form of (2.13) will of course depend on the underlying model. For the linear difference equation (2.1)=(2.4) we will have

$$\hat{y}(t|\Theta) = \Theta^T\varphi(t) \tag{2.14}$$

The same prediction or guess of the output y(t) will be used for the model (2.5) with disturbances, in case $\{v(t)\}$ is considered as "unpredictable" (like white noise). For the state space model (2.11) the predictor function is given by the Kalman filter. Then g_M is a linear function of past data.

For the ARMAX-model (2.6) a natural predictor is computed as

$$\hat{y}(t|\Theta)+c_1\hat{y}(t-1|\Theta)+...+c_n\hat{y}(t-n|\Theta) =$$

$$= (c_1-a_1)y(t-1)+...+(c_n-a_n)y(t-n)+b_1u(t-1) + ...+b_mu(t-m) \tag{2.15}$$

Notice that this can be rewritten as

$$\hat{y}(t|\Theta) = \Theta^T\varphi(t,\Theta) \tag{2.16a}$$

$$\Theta=(a_1....a_n\; b_1....b_m\; c_1....c_n)^T \tag{2.16b}$$

$$\varphi(t,\Theta) =$$

$$\left(-y(t-1)...-y(t-n)u(t-1)...u(t-m)\varepsilon(t-1,\Theta)...\varepsilon(t-n,\Theta)\right)^T \tag{2.16c}$$

$$\varepsilon(t,\Theta) = y(t) - \hat{y}(t|\Theta). \tag{2.16d}$$

For the model (2.7)=(2.11) a natural predictor is also given by (2.16a) with Θ and $\varphi(t,\Theta)$ defined by (2.8)-(2.10) Notice that in this case the prediction is formed from past inputs only. We then have, formally

$$\hat{y}(t|\Theta) = g_M(\Theta;t,u^{t-1}). \tag{2.17}$$

Such a model we call an "output error model" or a "simulation model".

Similarly the model (2.12) gives the predictor

$$\hat{y}(t|\Theta)=H^{-1}(q,\Theta)G(q,\Theta)u(t)+\left(1-H^{-1}(q,\Theta)\right)y(t) \tag{2.18}$$

Notice that the function $g_M(\Theta;t,\bullet)$ in (2.13) is a deterministic function from the observations Z^{t-1} to the predicted output. All stochastic assumptions involved in the model descriptions (e.g. white noises, covariances matrices, Gaussianness) have only served as vehicles or "alibis" to arrive at the predictor function. 不在场证据 Nonlinear models also fit into the formulation (2.13). Then g simply is a nonlinear function of Z^{t-1}.

The prediction $\hat{y}(t|\Theta)$ is computed from Z^{t-1} at time t-1. At time t the output y(t) is received. We can then evaluate how good the prediction was by computing

$$\varepsilon(t,\Theta) = y(t) - \hat{y}(t|\Theta). \tag{2.19}$$

We shall call $\varepsilon(t,\Theta)$ the prediction error at time t, corresponding to model $M(\Theta)$. This term will be the generic name for general model sets. Depending on the character of the particular model set, other names like, e.g. the (generalized) equation error may be used. For a simulation model (2.17) it is customary to call the corresponding prediction error (2.19) the output error.

3 OFF-LINE PREDICTION ERROR IDENTIFICATION METHODS

We shall now discuss how to select particular members in the model structure that describe observed input-output data as well as possible. Let the data up to time N be denoted by Z^N

What we seek is thus a mapping from Z^N to $\Theta\in D_M$ that selects the "best" $\Theta(Z^N)=\hat{\Theta}_N$. We shall discuss three basic such mappings, corresponding to different identification methods.

Given the general predictor model (2.13), or in the case of (2.12), (2.18), it is natural to evaluate the prediction error or

$$\varepsilon(t,\theta)=y(t)-\hat{y}(t|\theta)=H^{-1}(q,\theta)\big(y(t)-G(q,\theta)u(t)\big) \qquad (3.1)$$

We shall describe the following general methods for determining $\hat{\theta}_N$:

$$\hat{\theta}_N = \arg\min_{\theta\in D_{\mathcal{M}}} \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\theta) \qquad (3.2)$$

Here "arg min" denotes the minimizing argument.

The method (3.2) is a very natural approach to the identification problem: to select that member in the model set that gives the smallest prediction error when applied to the data record. We could of course also have used a more general norm than the quadratic one

$$\hat{\theta}_N= \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N}\ell\big(\varepsilon(t,\theta)\big) \qquad (3.3)$$

but we shall in this treatment confine ourselves to (3.2).

When applied to linear regression models (2.14) we have

$$\hat{\theta}_N= \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N}(y(t)-\varphi^T(t)\theta)^2=$$

$$= \frac{1}{N}\sum_{t=1}^{N}[\varphi(t)\varphi^T(t)]^{-1} \quad \frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t) \qquad (3.4)$$

yielding the celebrated least squares method. It is also well known how the method (3.3) contains the maximum likelihood method with the choice

$$\ell(\cdot)=-\log f_e(\cdot) \qquad (3.5)$$

where $f_e(\cdot)$ is the probability density function for the innovations. See Ljung (1978) or Åström (1980) for a further discussion on prediction error methods. An early reference is Åström and Bohlin (1965). The least squares method is treated in detail in e.g., Åström and Eykhoff (1971) and Hsia (1977).

Numerical techniques

Let us denote

$$V_N(\theta,Z^N)= \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\theta) \qquad (3.6)$$

To find $\hat{\theta}_N$ we use a numerical algorithm to minimize V_N. Except in the case (3.4), where V_N is quadratic in θ, the minimization has to be performed by iterative techniques. With "i" as the iteration number, we proceed as follows.

$$\hat{\theta}_N^{(i)}=\hat{\theta}_N^{(i-1)}-\mu_N^{(i)}[R_N^{(i)}]^{-1}V_N'(\hat{\theta}_N^{(i-1)},Z^N) \qquad (3.7)$$

Here V_N' is the gradient of the criterion,

$$V_N'(\theta,z^N)=- \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)\varepsilon(t,\theta) \qquad (3.8)$$

where ψ is defined as

$$\psi(t,\theta)= \frac{d}{d\theta}\hat{y}(t|\theta) \qquad (3.9)$$

R_N is a matrix that possibly modifies the search direction from the gradient one. Typical choices are:

$$R_N^{(i)}=I \text{ or } R_N^{(i)}= |V_N''(\hat{\theta}_N^{(i-1)},Z^N|^2\cdot I \qquad (3.10)$$

(gradient or steepest descent, un-normalized and normalized)

$$R_N^{(i)}= \frac{1}{N}\sum_{1}^{N}\psi(t,\hat{\theta}_N^{(i-1)})\psi^T(t,\hat{\theta}_N^{(i-1)}) \quad \text{(Gauss-Newton)} \qquad (3.11)$$

The scalar μ is chosen so as to guarantee a decrease in the criterion value. See Dennis and Schnabel (1983), Ch 10, for a readable and authoritative discussion of nonlinear least squares techniques for (3.6).

Gradients of the prediction

For specific model sets the iterative scheme (3.10), (3.15) needs only be complemented with expressions for computation of $\hat{y}(t|\theta)$ and $\psi(t,\theta)$ for any given value of θ. We shall develop such expressions now.

First consider the a linear regresssion (2.14). Clearly, we have

$$\psi(t,\theta)=\varphi(t) \qquad (3.12)$$

and it is easy to verify that with $\mu=1$, (3.7), (3.11) gives the solution (3.4) after just one iteration regardless of $\hat{\theta}_N^{(0)}$ in this case.

Next, consider a first order ARMAX model (compare (2.6) and (2.15)):

$$\varepsilon(t,\theta)+c\ \varepsilon(t-1,\theta)=y(t)+a\ y(t-1)-b\ u(t-1) \qquad (3.13)$$

$$\psi(t,\theta)= \frac{d}{d\theta}\hat{y}(t|\theta)= - \frac{d}{d\theta}\varepsilon(t,\theta)=\begin{pmatrix} \frac{d}{da}\ \varepsilon(t,\theta) \\ \frac{d}{db}\ \varepsilon(t,\theta) \\ \frac{d}{dc}\ \varepsilon(t,\theta) \end{pmatrix} \qquad (3.14)$$

By straightforward differentiation of (3.13) we obtain

$$\frac{\partial}{\partial a}\ \varepsilon(t,\theta)+c\ \frac{\partial}{\partial a}\ \varepsilon(t-1,\theta)=y(t-1)$$

$$\frac{\partial}{\partial b}\ \varepsilon(t,\theta)+c\ \frac{\partial}{\partial b}\ \varepsilon(t-1,\theta)=-u(t-1)$$

$$\frac{\partial}{\partial c}\ \varepsilon(t,\theta)+c\ \frac{\partial}{\partial c}\ \varepsilon(t-1,\theta)+\varepsilon(t-1,\theta)=0$$

or, with the notation $\varphi(t,\theta)$ defined by (2.16)

$$\psi(t,\theta)+c\psi(t-1,\theta)=\varphi(t,\theta) \qquad (3.15)$$

It is easy to see that this extends to general ARMAX models as

$$\psi(t,\theta)= \frac{1}{C(q)}\ \varphi(t,\theta) \qquad (3.16)$$

Similarly the output error model, (2.7) - (2.10) gives

$$\psi(t,\theta) = \frac{1}{F(q)}\ \varphi(t,\theta) \qquad (3.17)$$

with $\varphi(t,\theta)$ defined by (2.9).

Asymptotic Properties

The asymptotic distribution for the estimates defined by (3.2) can be determined under quite general conditions, and correspond to classical results for the maximum likelihood estimates for the case of independent observations. See, e.g. Ljung (1978) and Ljung and Caines (1979) for results corresponding to the framework used here. The techniques are based on nonstandard versions of the law of large numbers and the central limit theorem, and are somewhat algebraically involved due to the general model structure. The result is that, under weak regularity conditions we have

$$\hat{\Theta}_N \to \Theta^* = \arg\min \bar{V}(\Theta) \text{ w.p.1 as } N\to\infty \tag{3.18a}$$

$$\bar{V}(\Theta) = \lim_{N\to\infty} E V_N(\Theta, Z^N) \tag{3.18b}$$

$$\sqrt{N}(\hat{\Theta}_N - \Theta^*) \in AsN(0,P) \tag{3.19a}$$

$$P = Q^{-1} H\, Q^{-1} \tag{3.19b}$$

$$Q = \frac{d^2}{d\Theta^2}\bar{V}(\Theta)\Big|_{\Theta=\Theta^*} \tag{3.19c}$$

$$H = \lim_{N\to\infty} N \cdot E\left\{[V_N'(\Theta^*,Z^N)]^T V_N'(\Theta^*,Z^N)\right\} \tag{3.19d}$$

Here prime denotes differentiation w.r.t. Θ and (3.19a) means that the random variable on the left converges in distribution to the normal distribution with zero mean and covariance matrix P.

Notice that (3.18)-(3.19) hold even if a true description of the system is not available in the model set. If this indeed is the case so that there exists a (unique) Θ_0 in D_M such that $\varepsilon(t,\Theta_0)=e(t)$ is white noise with variance λ, then

$$\Theta^* = \Theta_0$$

and

$$P = \lambda[E\psi(t,\Theta_0)\psi^T(t,\Theta_0)]^{-1} \tag{3.20}$$

When the general criterion (3.3) is used, (3.20) takes the form

$$P = \kappa(\ell)[E\psi(t,\Theta_0)\psi^T(t,\Theta_0)]^{-1} \tag{3.21a}$$

$$\kappa(\ell) = \frac{E[\ell'(e(t))]^2}{[E\ell''(e(t))]^2} \tag{3.21b}$$

See, e.g. Ljung and Söderström (1983).

Often our prime interest is in the transfer functions of the model (2.12)

$$\begin{aligned} \hat{G}_N(q) &= G(q,\hat{\Theta}_N) \\ \hat{H}_N(q) &= H(q,\hat{\Theta}_N) \end{aligned} \tag{3.22}$$

rather than in the parameter Θ itself. Suppose that the true system can be described by

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \tag{3.23}$$

with $E\, e^2(t) = \lambda$

Then, there is a simple expression for the variance of (3.22), that is asymptotic in the model order n. We assume that the model structure (2.12) is employed with a black-box parametrization of order n: Then

$$\text{Cov}\begin{pmatrix} \hat{G}_N(e^{i\omega}) \\ \hat{H}_N(e^{i\omega}) \end{pmatrix} \sim \frac{n}{N}\Phi_v(\omega)\begin{bmatrix} \Phi_u(\omega) & \Phi_{ue}(\omega) \\ \Phi_{eu}(\omega) & \lambda \end{bmatrix}^{-1} \tag{3.24}$$

for large N and n. Here $\Phi_v(\omega)$ is the noise spectrum $\Phi_v(\omega) = \lambda|H_0(e^{i\omega})|^2$,

while $\Phi_u(\omega)$ is the input spectrum and $\Phi_{ue}(\omega)$ is the cross spectrum between input u and innovations e. See Ljung (1985) for a derivation and discussion of these results.

4 RECURSIVE PARAMETER ESTIMATION TECHNIQUES

By a recursive estimation method we shall mean an algorithm that updates Θ in order to take new information into account, using a finite, fixed amount of calculations and memory locations. Conceptually, we may write

$$S(t) = F\big(t, S(t-1), y(t), u(t)\big) \tag{4.1a}$$

$$\hat{\Theta}(t) = H\big(S(t)\big) \tag{4.1b}$$

Here, S(t) is a finite and fixed dimensional auxiliary vector ("an information state"), that condenses the information contained in the increasing data record Z^t. Usually, as we shall see, the algorithm will take more specific forms than (4.1).

Now, several different approaches to the construction of such recursive algorithms can be taken. Some common ones are

- o The Bayesian approach: The model parameter Θ is considered to be a random vector, which of course is correlated with the data process Z^t. Approximations of the a posteriori probability density for Θ can then be computed, and $\Theta(t)$ can be taken as e.g., the mean of this posterior distribution.

- o The stochastic approximation approach: A measure of fit between the data and the model is chosen, e.g., the variance of the difference of true output and model output. This criterion can then be minimized using ideas from stochastic approximation, like the Robbins and Monro (1951) algorithm.

- o Observers, Model-reference techniques, Pseudolinear regressions: The model error can be correlated with part of the data in order to make it as small as possible.

In this contribution, though, we shall concentrate on a fourth approach, viz to derive recursive algorithms from off-line identification ideas.

Recursive prediction error algorithms

Let us start by trying to cast the prediction error methods of Section 3 into a recursive form. To prepare for time-varying systems, we shall consider a weighted sum instead of (3.2):

$$V_t(\Theta, Z^t) = \sum_{k=1}^{t} \beta(t,k)\varepsilon^2(k,\Theta) \tag{4.2}$$

The weighting profile $\{\beta(t,k)\}$ is selected so that adequate weight is assigned to the different prediction errors. If, for example, the system is time varying, it is reasonable to pay less attention to old data, i.e. to let $\beta(t,k)$ be an increasing function of k. We shall work with the following structure for $\beta(t,k)$:

Let $\lambda(j)$ 0=1,...,t be a given sequence of scalars and define

$$\gamma(t)=\Big[\sum_{k=1}^{t}\prod_{j=k+1}^{t}\lambda(j)\Big]^{-1} \tag{4.3}$$

and

$$\beta(t,k)=\gamma(t)\cdot\prod_{j=k+1}^{t}\lambda(j);\ \beta(t,t)=\gamma(t) \tag{4.4}$$

Notice that a constant $\lambda(j)\equiv\lambda$ gives an exponential forgetting profile

$$\beta(t,k)=\frac{1-\lambda^{t}}{1-\lambda}\cdot\lambda^{t-k} \tag{4.5}$$

and that, by construction

$$\sum_{k=1}^{t}\beta(t,k)=1$$

For the gradient of $V_t(\Theta,Z^t)$ we have

$$\begin{aligned}V_t'(\Theta,Z^t) &= -\sum_{k=1}^{t}\beta(t,k)\psi(k,\Theta)\varepsilon(k,\Theta) =\\ &= \Big[\lambda(t)\,\frac{\gamma(t)}{\gamma(t-1)}V_{t-1}'(\Theta,Z^{t-1})-\gamma(t)\psi(t,\Theta)\varepsilon(t,\Theta)\Big] =\\ &=V_{t-1}'(\Theta,Z^{t-1})+\gamma(t)\big[-\psi(t,\Theta)\varepsilon(t,\Theta)-V_{t-1}'(\Theta,Z^{t-1})\big]\end{aligned} \tag{4.6}$$

For the prediction error approach we developed the general search algorithm (3.7):

$$\hat\Theta_t^{(i)}=\hat\Theta_t^{(i-1)}-\mu_t^{(i)}\big[R_t^{(i)}\big]^{-1}V_t'\big(\hat\Theta_t^{(i-1)},Z^t\big) \tag{4.7}$$

Here the subscript "t" denotes that the estimate is based on t data, i.e. Z^t. The superscript "(i)" denotes the i:th iteration of the minimization procedure.

Suppose now that for each iteration i, we also collect one more data point. This would give an algorithm

$$\hat\Theta_t^{(t)}=\hat\Theta_{t-1}^{(t-1)}-\mu_t^{(t)}\big[R_t^{(t)}\big]^{-1}V_t'\big(\hat\Theta_{t-1}^{(t-1)},Z^t\big) \tag{4.8}$$

For easier notation, we introduce

$$\hat\Theta(t)=\hat\Theta_t^{(t)},\qquad R(t)=R_t^{(t)} \tag{4.9}$$

We now make the (bold) approximation that $\hat\Theta(t-1)$ actually minimized $V_{t-1}(\Theta,Z^t)$, so that

$$V_{t-1}'\big(\hat\Theta(t-1),Z^{t-1}\big)=0. \tag{4.10}$$

Then, we have, from (4.6)

$$V_t'\big(\hat\Theta(t-1),Z^t\big)=-\gamma(t)\psi\big(t,\hat\Theta(t-1)\big)\varepsilon\big(t,\hat\Theta(t-1)\big) \tag{4.11}$$

With this approximation (and taking $\mu(t)=1$) we thus arrive at the algorithm

$$\hat\Theta(t)=\hat\Theta(t-1)+\gamma(t)R^{-1}(t)\psi\big(t,\hat\Theta(t-1)\big)\varepsilon\big(t,\hat\Theta(t-1)\big) \tag{4.12}$$

The choice of R(t) could be a gradient scheme (cf (3.10))

$$R(t)=I \tag{4.13}$$

or a normalized gradient

$$R(t)=\big|\psi\big(t,\hat\Theta(t-1)\big)\big|^{2}\cdot I \tag{4.14a}$$

or

$$R(t)=\sum_{k=1}^{t}\beta(t,k)\big|\psi\big(k,\hat\Theta(k-1)\big)\big|^{2}\cdot I \tag{4.14b}$$

It could also be a Gauss-Newton version (see (3.11))

$$R(t)=\sum_{k=1}^{t}\beta(t,k)\psi\big(k,\hat\Theta(k-1)\big)\psi^{T}\big(k,\hat\Theta(k-1)\big)$$

which analogously to (4.6) can be written

$$R(t)=R(t-1)+\gamma(t)\big[\psi\big(t,\hat\Theta(t-1)\big)\psi^{T}\big(t,\hat\Theta(t-1)\big)-R(t-1)\big] \tag{4.15}$$

Now, is the algorithm (4.12) a recursive one? To answer this question we consider first the linear regression case

$$\hat y(t|\Theta)=\varphi^{T}(t)\Theta.$$

Then $\psi(t)=\varphi(t)$ and (4.12), (4.15) is indeed the well-known recursive least squares algorithm (RLS).

$$\hat\Theta(t)=\hat\Theta(t-1)+\gamma(t)R^{-1}(t)\varepsilon(t) \tag{4.16a}$$

$$\varepsilon(t)=y(t)-\varphi^{T}(t)\hat\Theta(t-1) \tag{4.16b}$$

$$R(t)=R(t-1)+\gamma(t)\big[\varphi(t)\varphi^{T}(t)-R(t-1)\big] \tag{4.16c}$$

For some numerical aspects on the implementation of (4.16), see the next section.

Now, normally, it is not the case that (4.12) is recursive. Let us consider the following case.

<u>Example 4.1</u> Consider the first order ARMAX model

$$y(t)+ay(t-1)=bu(t-1)\ e(t)+ce(t-1) \tag{4.17}$$

Here $\Theta^{T}=(a\ b\ c)$. The prediction $\hat y(t|\Theta)$ is computed according to

$$\hat y(t|\Theta)+c\,\hat y(t-1|\Theta)=(c-a)y(t-1)+bu(t-1) \tag{4.18}$$

The gradient $\psi(t,\Theta)$ is determined by (3.19)

$$\psi(t,\Theta)+c\,\psi(t-1,\Theta)=\begin{pmatrix}-y(t-1)\\ u(t-1)\\ \varepsilon(t-1,\Theta)\end{pmatrix} \tag{4.19}$$

where, as usual,

$$\varepsilon(t,\Theta)=y(t)-\hat y(t|\Theta). \tag{4.20}$$

We see from (4.18) that for any given Θ, $\hat y(t|\Theta)$ is the output of a linear filter with input y(k), u(k), $k\leq t-1$. This filter has an infinite impulse response (the impulse response is $h(k)=(c-a)(-c)^{k}$). Therefore, in order to compute $\hat y(t|\hat\Theta(t-1))$ and $\varepsilon\big(t,\hat\Theta(t-1)\big)$ which we need in (4.12), we have to use the whole data record y^{t-1}. An algorithm that uses $\varepsilon\big((t,\hat\Theta(t-1)\big)$ can thus not be recursive.

We therefore need an approximation of $\varepsilon(t,\hat{\theta}(t-1))$ that can be computed recursively. A natural idea is to use (4.18) with the current estimate all the time, that is, if we let $\hat{y}(t)$ denote the approximation of $\hat{y}(t|\hat{\theta}(t-1))$

$$\hat{y}(t)+\hat{c}(t-1)\hat{y}(t-1)=[\hat{c}(t-1)-\hat{a}(t-1)]y(t-1)+\hat{b}(t-1)u(t-1) \qquad (4.21)$$

With the same argument we can use $\psi(t)$ as an approximation of $\psi(t,\hat{\theta}(t-1))$, where $\psi(t)$ is given by

$$\psi(t) + \hat{c}(t-1)\psi(t-1) = \begin{pmatrix} -y(t-1) \\ u(t-1) \\ \bar{\varepsilon}(t-1) \end{pmatrix} \qquad (4.22a)$$

Here we let the <u>residual</u> $\bar{\varepsilon}(t)$ be defined by

$$\bar{\varepsilon}(t)+\hat{c}(t)\bar{\varepsilon}(t-1)=y(t)+\hat{a}(t)y(t-1)-\hat{b}(t)u(t) \qquad (4.22b)$$

It differs from $y(t)-\hat{y}(t)$ since we have used the updated estimates $\hat{\theta}(t)$ (rather than $\hat{\theta}(t-1)$) for the computation.

Using these approximation in (4.12) now gives the algorithm:

$$\begin{pmatrix} \hat{a}(t) \\ \hat{b}(t) \\ \hat{c}(t) \end{pmatrix} = \begin{pmatrix} \hat{a}(t-1) \\ \hat{b}(t-1) \\ \hat{c}(t-1) \end{pmatrix} + \gamma(t)R^{-1}(t)\psi(t)\varepsilon(t) \qquad (4.23a)$$

together with (4.21), (4.22) and

$$\varepsilon(t)=y(t)-\hat{y}(t) \qquad (4.23b)$$

$$R(t)=R(t-1)+\gamma(t)[\psi(t)\psi^T(t)-R(t-1)] \qquad (4.23c)$$

□

The example showed how to modify the algorithm (4.12) so as to make it a recursive one. The general principle of how to construct the approximation $\varepsilon(t)$ and $\psi(t)$ can be described in words as follows:

"In the time recursions defining $\psi(t,\theta)$ and $\varepsilon(t,\theta)$ from Z^t for any given θ, replace, at time k, the parameter θ by the currently available estimate $\hat{\theta}(k)$. Denote the resulting approximation of $\psi(t,\hat{\theta}(t-1))$ and $\varepsilon(t,\hat{\theta}(t-1))$ by $\psi(t)$ and $\varepsilon(t)$." (4.24)

This gives as the family of <u>recursive prediction error methods (RPEM)</u>:

$$\hat{\theta}(t)=\hat{\theta}(t-1)+\gamma(t)R^{-1}(t)\psi(t)\varepsilon(t) \qquad (4.25a)$$

$$\varepsilon(t)=y(t)-\hat{y}(t) \qquad (4.25b)$$

ψ is the gradient of prediction w.r.t. θ. $R(t)$ is a positive semidefinite matrix, e.g.

$$R(t)=R(t-1)+\gamma(t)[\psi(t)\psi^T(t)-R(t-1)]$$
(Gauss-Newton) (4.25c)

or

$$R(t)=R(t-1)+\gamma(t)[\,|\psi(t)|^2I-R(t-1)]$$
(Normalized gradient) (4.25d)

It should be noted that the predictor filters, corresponding to (4.21) and (4.22) should be stable. It is therefore necessary to include a test whether $\hat{\theta}(t)$ corresponds to a stable predictor. If not, $\hat{\theta}(t)$ will have to be projected into the stability region.

<u>Asymptotic properties</u>

The properties of the general algorithm (4.25) are analysed in Chapter 4 of Ljung and Sôderstrôm (1983). The results can be summarized as follows (subject to some regularity conditions).

- Assume that $\lim_{t\to\infty} t\cdot\gamma(t)=c$
- $R(t)\geq\delta I$ all t

Then

$$\hat{\theta}(t)\to\theta^* \text{ w.p. 1 as } t\to\infty \qquad (4.26)$$

where θ^* is a local minimum of $\bar{V}(\theta)$ defined by (3.18).

Furthermore, if

- $\lim_{t\to\infty} t\cdot\gamma(t)=1$
- $R(t)$ is the Gauss-Newton choice
- $\theta^*=\theta_0$ is a unique true value $(\varepsilon(t,\theta_0)$ is white noise$)$ then

$$\sqrt{t}(\hat{\theta}(t)-\theta_0)\in \text{AsN}(0,P) \qquad (4.27)$$

where P is given by (3.20)

The result thus is that the asymptotic properties of the recursively computed estimate coincide with those of the corresponding off-line estimate.

5 IMPLEMENTATION ASPECTS

Consider the basic algorithms (4.25). In actual implementations one would of course not construct the matrix R(t) at each sample and then invert it. There are several ways of handling this problem. One is to define

$$\gamma(t)R^{-1}(t)=P(t) \qquad (5.1)$$

and apply the matrix inversion lemma to (4.25c). This gives

$$P(t)=\left[P(t-1)-\frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{\lambda(t)+\psi^T(t)P(t-1)\psi(t)}\right]/\lambda(t) \qquad (5.2)$$

where $\lambda(t)$ is given by (4.3)-(4.4).

We could also define

$$L(t)=P(t)\psi(t)=\frac{P(t-1)\psi(t)}{\lambda(t)+\psi^T(t)P(t-1)\psi(t)} \qquad (5.3)$$

To achieve better numerical properties, it is often advisable to factorize P(t) and update its factors separately. Bierman's (1977) algorithm employs a UDU-factorization. See this reference or Chapter 6 of Ljung and Sôderstrôm (1983) for details. Here we shall give some details of a related algorithm, which is directly based on Householder transformations. It was given by Morf and Kailath (1975).

Let P(t) in (5.1) be factorized as

$$P(t) = Q(t)Q^T(t) \qquad (5.4)$$

which, for triangular Q is the Cholesky decomposition.

Step 1. At time t-1, let Q(t-1) be a lower triangular square root of P(t-1) as in (5.3). Let $\mu(t)$ be $\sqrt{\lambda}(t)$. Form the $(1+d)\times(1+d)(d=\dim\Theta)$ matrix

$$\mathcal{S}(t-1) = \begin{bmatrix} \mu(t) & 0 \\ Q^T(t-1)\psi(t) & Q^T(t-1) \end{bmatrix} \qquad (5.5)$$

Step 2. Apply an orthogonal $(1+d)\times(1+d)$ transformation

$$T \quad (T^T T = I)$$

to $\mathcal{S}(t-1)$ so that $T\,\mathcal{S}(t-1)$ becomes an upper triangular matrix. T can, e.g. be found by Householder transformations. Let $\Pi(t)$, $\tilde{L}(t)$ and $\bar{Q}(t)$ be the 1×1 $d\times 1$ and $d\times d$ matrices defined by

$$T\,\mathcal{S}(t-1) = \begin{bmatrix} \Pi(t) & \tilde{L}^T(t) \\ 0 & \bar{Q}^T(t) \end{bmatrix} \qquad (5.6)$$

(Clearly $\bar{Q}$ is lower triangular).

Step 3. Now with L(t) and P(t) as in (5.2)-(5.3), we have

$$L(t) = \tilde{L}(t)\,\Pi(t)$$

$$P(t) = \bar{Q}(t)\bar{Q}^T(t)/\lambda(t) \qquad (5.7)$$

$$\Pi(t)\Pi^T(t) = \lambda(t) + \psi^T(t)P(t-1)\psi(t)$$

Hence

$$Q(t) = \bar{Q}(t)/\sqrt{\lambda(t)} \qquad (5.8)$$

Verification:

Multiplying (5.6) with its transpose gives

$$\begin{bmatrix} \Pi(t) & 0 \\ \tilde{L}(t) & \bar{Q}(t) \end{bmatrix} \begin{bmatrix} \Pi(t) & \tilde{L}^T(t) \\ 0 & \bar{Q}^T(t) \end{bmatrix} =$$

$$= \begin{bmatrix} (\Pi(t))^2 & \Pi(t)\tilde{L}^T(t) \\ \tilde{L}(t)\Pi(t) & \bar{Q}(t)\bar{Q}^T(t)+\tilde{L}(t)\tilde{L}^T(t) \end{bmatrix} =$$

$$= \mathcal{S}^T(t-1)T^T T\,\mathcal{S}(t-1) = \mathcal{S}^T(t-1)\mathcal{S}(t-1) =$$

$$= \begin{bmatrix} (\mu(t))^2+\psi^T(t)Q(t-1)Q^T(t-1)\psi(t) & \psi^T(t)Q(t-1)Q^T(t-1) \\ Q(t-1)Q^T(t-1)\psi(t) & Q(t-1)Q^T(t-1) \end{bmatrix}$$

Using the facts that $Q(t-1)Q^T(t-1)=P(t-1)$ and $\mu^2(t)=\lambda(t)$ it is now immediate to verify the equalities in (5.7) by a comparison with (5.2)-(5.3)

There are several advantages with this particular way of performing (5.2)-(5.3). First, the only essential computations to perform is the triangularization step (or "Q-R factorization") (5.6), for which several good numerical procedures exist. This step both gives the new Q and the gain L after simple additional calculations. Second, in the update (5.6) we only deal with square roots of P. Hence the conditioning number of the matrix $\mathcal{S}(t-1)$ is much better than that of P. Third, with the triangular square root Q(t) it is easy to introduce regularization, i.e. measures to ensure that the eigenvalues of P stay bounded, at the same time as P remains positive definite.

Ljung and Ljung (1985) contains some further investigations of the numerical properties of least-squares type algorithms.

应付

6 COPING WITH TIME-VARYING SYSTEMS

The gain or "step-size" $\gamma(t)$ in the algorithms of Section 4 determines how much influence the last measurement will have on the estimate. Clearly, the choice of this $\gamma(t)$ reflects the "relative information contents" of that measurement. For a time-invariant system this will simply be inversely proportional to the number of previous measurements, which suggests that $\gamma(t)$ decays like $\gamma(t)\sim 1/t$. Comparing with (4.3)-(4.4) we see that this corresponds to $\lambda(j)\equiv 1$ and $\beta(t,k)=1/t$, which makes sense. The asymptotic results quoted in Section 4 also referred to this case.

For a time-varying system, $\gamma(t)$ will determine the trade-off between noise-sensitivity and tracking ability of the algorithm. This trade-off may not be so easy to reach, and should probably in many cases be adaptive, reflecting nonstationary properties in the system's behaviour. We shall discuss two approaches to deal with this problem.

Choice of forgetting factors λ(t)

The choice of forgetting profile $\beta(t,k)$ is conceptually simple: Select it so that the criterion essentially contains those measurements that are relevant for the current properties of the system. For a system that changes gradually and in a "stationary manner", the most common choice is to take a constant forgetting factor:

$$\beta(t,k) = \lambda^{t-k} \qquad \text{i.e.} \qquad \lambda(t) \equiv \lambda \qquad (6.1)$$

The constant λ is always chosen slightly less than 1, so that

$$\beta(t,k) = e^{(t-k)\log\lambda} \approx e^{-(t-k)(1-\lambda)}. \qquad (6.2)$$

This means that measurements that are older than $T_0=1/(1-\lambda)$ samples are included in the criterion with a weight that is $e^{-1}\approx 36\%$ less than that of the most recent measurement. We could call

$$T_0 = 1/(1-\lambda) \qquad (6.3)$$

the memory time constant of the criterion. If the system remains approximately constant over T_0 samples, a suitable choice of λ can then be made from (6.3). Since the sampling interval typically reflects the natural time constants of the system dynamics, we could thus select λ so that $1/(1-\lambda)$ reflects the ratio between the time constants of variations in the dynamics and those of the dynamics itself. Typical choices of λ are in the range between 0.98 and 0.995.

For a system that undergoes abrupt and sudden changes, rather than steady and slow ones, an adaptive choice of λ could be conceived. When an abrupt system change has been detected, it is suitable to decrease $\lambda(t)$ to a small value for one sample, thereby "cutting off" past measurements from the criterion, and then increase it to a value (close to) 1 again. Such adaptive choices of λ are discussed, e.g. in Fortesque et al (1981) and Hägglund (1984).

Including a model of parameter changes

For a linear regression model with time varying parameters, we could postulate a formal model:

$$\Theta(t)=\Theta(t-1)+w(t) \qquad (6.4a)$$

$$y(t)=\varphi^T(t)\Theta(t)+e(t) \qquad (6.4b)$$

Here $\{w(t)\}$ and $\{e(t)\}$ are assumed to be white noises with variances $R_1(t)$ and $R_2(t)$, respectively. Applying the Kalman filter to (6.4) gives the algorithm

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + L(t)\varepsilon(t)$$

$$\varepsilon(t) = y(t) - \hat{y}(t) \qquad (6.5)$$

$$L(t) = P(t-1)\varphi(t)[R_2(t)+\varphi^T(t)P(t-1)\varphi(t)]^{-1}$$

$$P(t)=P(t-1)- \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1))}{[R_2+\varphi^T(t)P(t-1)\varphi(t)]^{-1}} +R_1(t)$$

when $\{e(t)\}$ and $\{w(t)\}$ are Gaussian noises.

In this case, this algorithm does give the optimal trade-off between tracking ability and noise sensitivity. in terms of minimal a posteriori parameter error covariance matrix (This follows from the original derivation of the Kalman filter, Kalman and Bucy (1961), as pointed out in Bohlin (1970) and Åström and Wittenmark (1971)).

The same idea can of course be applied to the other model structures in Section 2, even if (6.5) is entirely ad hoc for structures other than the linear regression. The advantage with (6.5) is that the matrix $R_1(t)$ describes (the variance of) expected parameter changes. It could thus be a very useful alternative, in particular if we have some insight into how the parameters might vary, e.g. if certain parameters vary more rapidly than others.

The case where the parameters are subject variations, that themselves are of a nonstationary nature (i.e. $R_1(t)$ in (6.4) varies substantially with t) can be dealt with a parallel algorithm structure, as described in Andersson (1985). A related technique for systems whose parameters change as a Markov Chain is given by Millnert (1982).

7 CONCLUSIONS

We have discussed basic techniques for recursive parameter estimation in a general parametrizations of dynamical systems. We have seen that the off-line identification problem offers a good insight into basic ideas and techniques for adaptation algorithms.

A fairly complete asymptotic theory can be given for the case where the gain tends to zero. This corresponds to a constant, time invariant system. It is reasonable to assume that similar properties will hold also when the true system is slowly time varying and a small, constant gain $\gamma(t)\equiv\gamma>0$ is used. It would be interesting, though, to develop a more complete theory for how to deal with time-varying systems, with a nonstationary behaviour.

REFERENCES

Andersson, P (1985). Adaptive forgetting in recursive identification through multiple models. Int. J. of Control, to appear.

Åström , K J (1980). Maximum likelihood and prediction error methods, Automatica, vol 16, pp 551-574.

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

Åström, K J and T Bohlin (1965). Numerical identification of linear systems from normal operating records, IFAC Symposium on Self-adaptive systems. Teddington, England. Also in P H Hammond, ed.: Theory of self-adaptive control systems, Plenum Press, New York,

Åström, K J and P Eykhoff (1971). System identification - a survey. Automatica vol 7, pp 123-167.

Åström, K J and B Wittenmark (1971). Problems of identification and control. Journal of Mathematical Analysis and Applications. vol 34, pp 90-113.

Bierman, G J (1977). Factorization methods for Discrete Sequential Estimation, Academic Press, New York.

Bohlin, T (1970). Information pattern for linear discrete time models with stochastic coefficients. IEEE Transactions on Automatic Control, vol AC-15, pp 104-106.

Dennis, J. E. and R. B. Schnabel (1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs.

Fortesque, T. R., L. S. Kershenbaum and B. E. Ydstie (1981). Implementation of Self-tuning Regulators with Variable Forgetting Factors, Automatica, Vol 17, pp 831-835.

Hägglund (1984). Adaptive control of systems subject to large parameter changes. Preprints, 9th World Congress, Budapest, Hungary, July 2-6.

Hsia, T. C. (1977). Identification: Least Squares Methods. Lexington Books, Lexington, Mass.

Kalman, R. E. and R. S. Bucy (1961). New results in linear filtering and prediction theory. Transactions ASME, Journal of Basic Engineering (ser D), vol 83, pp 95-108.

Lawson, C. L. and R. J. Hanson (1974). Solving Least Squares Problems. Prentice Hall, Englewood Cliffs, N J.

Ljung, L. (1977). On positive real transfer functions and the convergence of some recursions. IEEE Transactions on Automatic Control, vol AC-22, pp 539-551.

Ljung, L. (1978). Convergence analysis of parametric identification methods. IEEE Transactions on Automatic Control, Vol AC-23, pp 770-783.

Ljung, L. (1985). Asymptotic variance expressions for identified transfer functions. *IEEE Transactions on Automatic Control*, Vol AC-30, September 1985. To appear.

Ljung, L. and P. Caines (1979). Asymptotic normality of prediction error estimation for approximate system models. *Stochastics*, Vol 3, pp 29-46.

Ljung, S. and L. Ljung (1985). Error propagation properties of recursive least squares adaptation algorithms. *Automatica*, vol 21, pp 157-167.

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

Millnert, M. (1982). Identification and control of systems subject to abrupt changes. Linköping Studies in Science and Technology, Dissertation No 82. Linköping University, Linköping, Sweden.

Morf, M. and T. Kailath (1975). Square root algorithms for least squares estimation. *IEEE Transactions on Automatic Control*, Vol AC-20, pp 487-497.

Robbins, H. and S. Monro (1951). A stochastic approximation methos. *Annals of Mathematical Statistics*, Vol 22, pp 702-710.

Söderström, T. and P. Stoica (1983). *The Instrumental Variable Approach to System Identification*, Springer, Berlin.

MODEL-BASED TUNING OF CONTROLLERS

Pierre R. Bélanger

McGill University, Montreal, Canada

SUMMARY

This paper is intended to survey identification-based methods of tuning controllers. It begins with a discussion of some of the issues and tradeoffs. Some description of performance measures, plant and controller structures follows. Various methods of tuning parameter calculations, given a process model, are then reviewed. This is followed by a discussion of a few identification methods, including those where oscillations are elicited to identify a small number of parameters. A particular technique using Laguerre functions is described and examples of its use in a pulp mill are given. Finally, adaptive control is discussed, with some applications.

1. INTRODUCTION

For a process control engineer, tuning controllers is a fact of life. The tuning process consists of selecting a (small) number of controller parameter values in order to improve (or optimize) performance. In process control, dynamic models are usually crude; rough estimates of gain, major time constants and delay are often the only available data. This may be enough to choose a controller structure, but good performance can only be obtained by adjusting the controller parameters. This is in contrast to the situation which prevails in aerospace systems. There, considerable effort is spent in testing and modeling, because after-the-fact tuning is often out of the question.

Whether, and how often, a controller needs tuning depends on the sensitivity of the performance with respect to the plant variations and on the extent of those plant variations. Figure 1 shows two plots of "performance" (by some measure) vs "plant" (assumed to take on values between a and b). The nominal plant is represented by n, and the controller is assumed to optimize performance for that plant. The two curves correspond to two different measures of performance: it is clear that (1) will likely call for tuning, and that (2) may not require it. Curve (1) corresponds to the case where high performance is the objective; for example, the performance measure could be the rise time of the step response, with a heavy penalty term for overshoot. On the other hand, curve (2) corresponds to the case where performance goals are relatively modest, but relatively insensitive; for example, the performance measure could be the steady-state error to a step input (which would be zero with internal control for all plants leading to a stable closed loop). In other words, "tight" control needs more frequent attention than "loose" control. This may explain why in the one industry at least with which the author is the most familiar (pulp and paper), most loops are loosely tuned.

Much, if not most, tuning in the process industries is done by trial and error. Basically, this is on-line parameter optimization. This is quite practical if the performance measure is a relatively flat function of the controller parameters. Such measures are usually also relatively insensitive to plant variations, as in (2) of Figure 1. On the other hand, relatively stringent requirements will normally lead to a sharper optimum of the performance measure with respect to the controller parameters. To come near the minimum will (i) require many trials and (ii) require that each trial yield an accurate determination of the performance measure. The first is onerous if the process time constant is long; the second requires that test signals be large enough to raise the response above the noise level. It is not surprising to find that loops with long time constants are often loosely tuned, given the reluctance to interfere with normal production for long time periods. It is theoretically possible to have a parameter adjustment scheme which depends directly on measured performance. Some such schemes, e.g. based on stochastic approximation, have been proposed in the literature, but there seems to have been few, if any, applications. Other schemes combining stochastic approximation with partial identification have had more success.

In model-based tuning, one starts with a model structure, with variable parameters. Given the model parameter values, the controller parameters are determined by some algorithm (possibly optimization). The model parameters may depend in some known manner on process conditions; for example, the delay in a paper machine between thick stock flow and basis weight is inversely proportional to machine speed. Alternately, the model parameters may be estimated by an identification scheme.

The paper is organized as follows: Preliminary considerations, such as: performance measures, controller structures and plant structures are discussed. Next, several methods for the calculation of controller settings, given the plant model, are described. Both the continuous-time and discrete-time cases are given. Some identification techniques relevant in this context are presented, including both open- and closed-loop experiments. The following section offers practical examples of off-line identification followed by tuning. Adaptive control, indirect and direct, is the subject of the next section. The final section presents conclusions.

2. PRELIMINARIES

2.1 Performance Measures

As was pointed out in the Introduction, the need for tuning is determined in large part by the definition of a performance measure. Performance specifications are normally expressed as a set of constraints in the time domain, the frequency domain or both. For analytical and computational convenience, a single number is often used as performance measure, in the hope that minimization of that number, or performance index, will lead to satisfaction of the performance specifications.

The integral-squared error (ISE) is easily the most convenient index from an analytical point of view. It is expressed as

$$\mathrm{ISE} = \int_o^\infty e^2(t)\,dt = 2\int_o^\infty |E(f)|^2 df \qquad (1)$$

where e(t) is the error response to a given test input, usually a step.

Time or frequency weights can be inserted to control the behavior of the optimal response. For example, the integral-time-squared error (ITSE), or

$$\mathrm{ITSE} = \int_o^\infty t\,e^2(t)\,dt \qquad (2)$$

tolerates large errors in the initial portion of the response, but less as time grows. Integral penalty terms may be added to the error term in order to avoid excessive control action; the integral of the square of the control is most often used.

In the stochastic case, the error variance is almost universally used, often with the control variance as a penalty term. In contradistinction to the deterministic case, where the value of the integral squared has no direct interpretation, the variance does have meaning if the distribution is known. For example, in the Gaussian case, the error variance is related to the probability of the error being within certain bounds.

The variance of the error is related to the ISE criterion for the deterministic case. With reference to Figure 2, let $e(s)/y_d(s) = T(s)$. Then, $e(s)/w(s) = -T(s)$. By Parseval's theorem, the integral-squared error for a deterministic input $y_d(s)$ is:

$$\mathrm{ISE} = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} e(s)\,e(-s)\,ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} T(s)\,T(-s)\,y_d(s)\;y_d(-s)ds \qquad (3)$$

For a stationary random disturbance process w(•), the error variance is:

$$\mathrm{var} = \frac{1}{2\pi j}\int_{-j^\infty}^{j^\infty} T(s)T(-s)\phi_{ww}(s)ds \qquad (4)$$

where ϕ_{ww} is the transform of the autocorrelation function of w(•), i.e. its power density spectrum.

If $y_d(t)$ is chosen such that $y_d(s)y_d(-s) = \phi_{ww}(s)$, then the ISE for the input y_d is the same as the variance for the process w(•); minimizing the one is also minimizing the other. For example, $y_d(s) = 1$ (unit impulse) corresponds to $\phi_{ww} = 1$ (white noise); $y_d(s) = 1/s$ (unit step) corresponds to $\phi_{ww} = -\frac{1}{s^2}$ (Wiener process). In general, then, an appropriate deterministic input can be used to optimize the stochastic performance of the system, given some knowledge of the spectrum of the disturbance.

2.2 Plant Models

The plant models used in this paper are all linear and time-invariant. The reason for this is that the problem under consideration is regulation at a given steady state operating point. Linearity is a result of assuming small variations about the operating point; time invariance follows from the fact that the operating point is fixed. Of course, different operating points result in different models, which explains partially the need for tuning.

Only single-input-single-output plants will be considered, even though much of the work summarized here can be extended to the multivariable case.

2.3 Controllers

Only linear time-invariant controllers are considered here, in the configuration of Figure 2. The Proportional-Integral-Derivative (PID) controller, the workhorse of process control, is described by the formula:

$$u = \frac{1}{T_i}\int e\,dt + K p\,e + T_d\,\frac{d\,e}{d\,t}\,u \qquad (5)$$

where u = control variable
e = error
T_i = integral time constant
Kp = proportional gain
Td = derivative time constant

The PID controller is, of course, just one particular member of the class of controllers represented by:

$$u(s) = H(s)\,e(s) \qquad (6)$$

For the PID controller,

$$H(s) = \frac{1}{T_i s} + Kp + T_d s \qquad (7)$$

The discrete equivalent of the PID controller is generated by replacing s by $\frac{1-q^{-1}}{h}$, where q^{-1} is the unit delay operator and h is the sampling time. Thus,

$$H(q^{-1}) = \frac{h}{T_i}\,\frac{1}{1-q^{-1}} + Kp + \frac{T_d}{h}(1-q^{-1}) \qquad (8)$$

The difference equation corresponding to (8) is:

$$(1-q^{-1})u(t) = \frac{h}{T_i}e(t) + Kp(1-q^{-1})e(t) + \frac{T_d}{h}(1-q^{-1})^2 e(t)$$

$$u(t) = u(t-1) + \left(\frac{h}{T_i} + Kp + \frac{T_d}{h}\right)e(t) - \left(Kp + 2\frac{T_d}{h}\right)e(t-1) + \frac{T_d}{h}e(t-2) \qquad (9)$$

Here again, the discrete PID controller is merely a special case of

$$u(t) = H(q^{-1})\,e(t) \qquad (10)$$

where $H(q^{-1})$ is a ratio of polynomials in q^{-1}.

The Smith predictor [1] has proven to be useful to control systems with delay. The continuous-time version is illustrated in Figure 3, where the Gp block in the controller is a model of process, without delay. The transfer function is:

$$\frac{y}{y_d} = \frac{H\,Gp}{1\ \ H\,Gp}\,e^{-sT} \qquad (11)$$

Except for the delay, this is exactly the transfer function obtained by controlling the plant Gp by the controller H. Therefore, if a controller can be designed for Gp, the plant Gp e^{-sT} can be controlled