A Tutorial on Adaptive Control: The Self-tuning Approach

Prof. Zoran Vukić
University of Zagreb, Croatia
Faculty of Electrical Engineering and Computing
Department of Control and Computer Engineering in Automation

(Voice) 385-1-6129-840, (Fax) 385-1-6129-809, (Email) zoran.vukic@fer.hr, (URL) http://www.rasip.fer.hr/vukic

March 28, 2000

Contents

List of Figures

I. Introduction

This tutorial is primarily for those who were not previously exposed to the topic of adaptive control. The tutorial will hopefully succeed in the main goal of arousing interest among participants to use adaptive techniques for solving their problems in controlling various processes, helping them to understand the problems and giving them the necessary basic knowledge and bibliography for further study of this interesting topic.

Adaptive control is an interesting topic not only from the theoretical but also from the practical standpoint. It should be used only if it allows more efficient and reliable control.

Definition: Adaptive control

Adaptive control is a specific type of control where the process is controlled in closed-loop, and where knowledge about the system characteristics are obtained on-line while the system is operating. Based upon refreshed information obtained during normal operation, specific interventions in the control loop are made in order to fulfill the control goal. Interventions can be various but mainly they can be categorized as interventions obtained by changing:

a) signals - signal adaptation,

b) parameters - parameter adaptation,

c) structure - structure adaptation.

Generally, various combinations of the above interventions are possible. Here we will focus our attention only to parameter adaptation. Parameters in this context should be treated as observable and uncontrollable elements of the augmented state set, which consist of state variables (controllable and observable set elements) and parameters (observable but uncontrollable set elements). Also, due to its simplicity, only single input single output (SISO) systems will be treated here. Parameter adaptive control systems are characterized by separate specific algorithms for parameter estimation.

Adaptive control is a specific type of control, applicable to processes with changing dynamics in normal operating conditions subjected to stochastic disturbances. Reasons for using adaptive control are:

- variations in process dynamics,
- variations in the character of disturbances,
- engineering efficiency.

Generally, to control a process with changing dynamics is not easy. Today two possible solutions exist for that situation:

- adaptive control and
- robust control.

Adaptive control is used whenever process parameters are changing during operation and we do not know in advance what changes our process will experience. Assumption of process time invariance must be discarded. The solution has to be sought in a specific form of control where parameter estimation and regulator design will be realized on-line during normal control system operation.

When we talk today of industrial automation (Fig.1) then we should be aware that a distributed control systems in use today are complex hierarchical systems and that they are build upon reliable operation of lower levels such as device level (process interface) and control level (single loop control) systems. Equipment level (group control) represent even higher level where many processes are coordinated. Adaptive and robust control presented in this tutorial should be situated in the single loop control level. Without good and reliable system operation in lower levels, there is no sense talking about higher hierarchical levels at all.

A. Why the need for adaptive control

The need for some sort of control which will be capable to adjust to the changes brought about by change of dynamics of the process, disturbance or some other cause intensified during seventies.

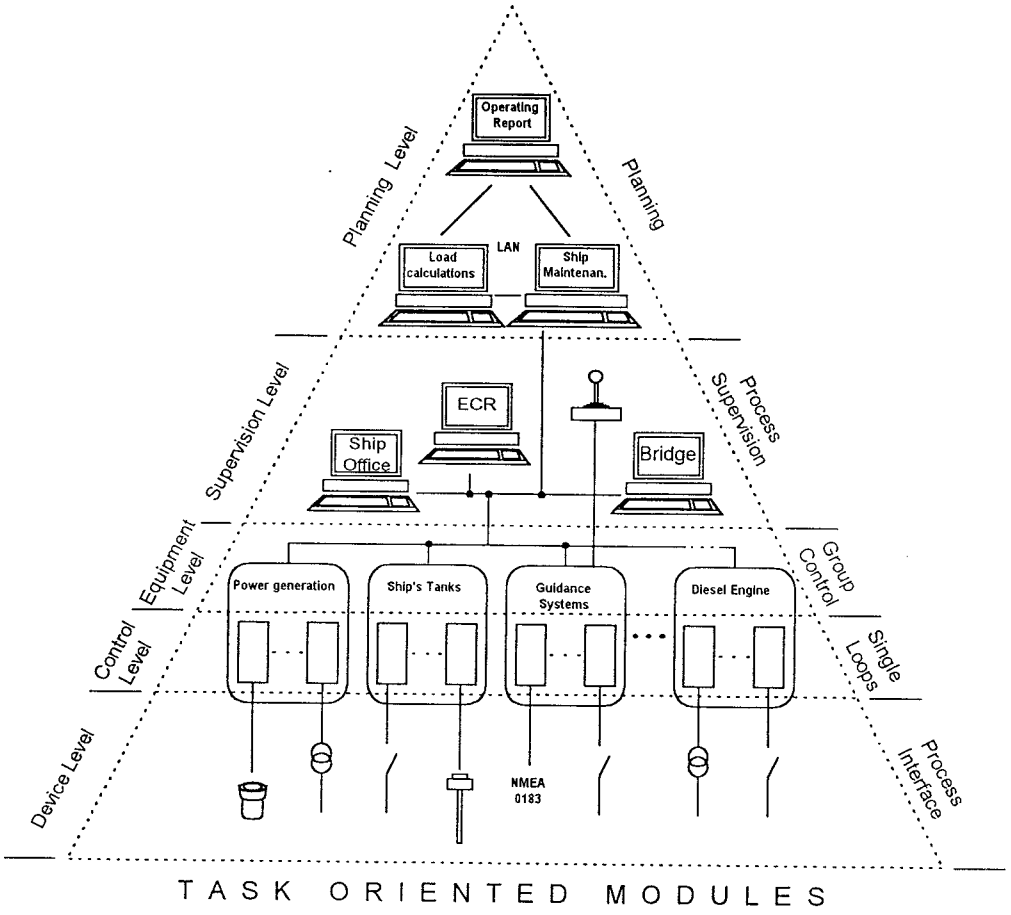


Fig. 1. Distributed control systems on board a ship.

Control community was aware of the fact that there should be some solution to that problem, already in the fifties, [1], but the technology of that time and control theoretical achievements prevented the use of the technique. However, many ideas developed in this early stage, proved later very useful. It is known that process parameters may vary due to:

- nonlinear actuators,
- changes of the operating conditions of the process,
- nonstationary disturbances acting upon the process.

Conventional controller with fixed parameters is not capable to adjust to changes experienced by the process even though the feedback by itself decrease the sensitivity to parameter variations. As is well known, increasing the loop gain of the system have as a consequence that the sensitivity is decreased, because it is related with the gain as:

$$Sensitivity = \frac{1}{1+GH}$$

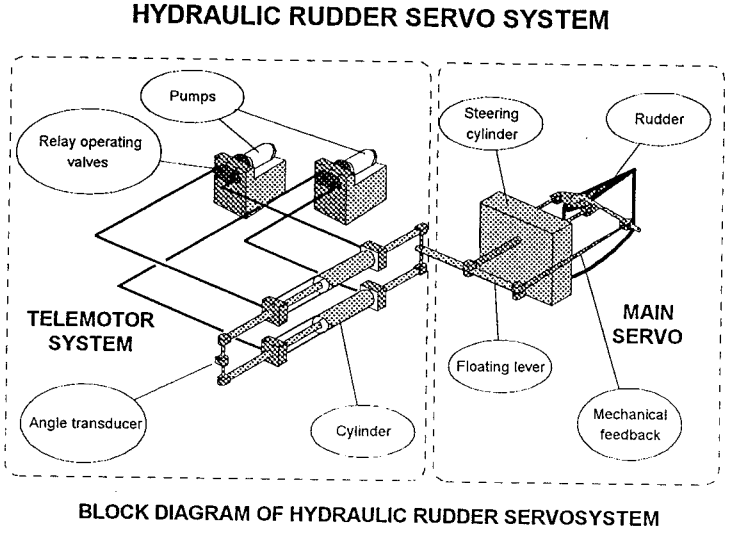
If there are bounds on the uncertainty of the process parameters, it is possible to design *robust controllers* by increasing the complexity of the controller. To use this approach it is necessary:

1. to know the process structure fairly accurately,
2. to have bounds on the variations of the parameters.

If this is not known, than the only possibility we have to solve the problem is to use the *adaptive controller*.

That the time invariance is not an adequate assumption can be shown by the following examples:

Example 1: **Nonlinear actuator - ship's rudder servosystem, [2]**



Change of the dynamics of this system can be caused by:

- *leakage of the fluid,*
- *intrusion of the air somewhere in the system,*
- *uneven wear of the pumps,*
- *etc.*

Example 2: **Path guidance of ships in shallow water, [3].**

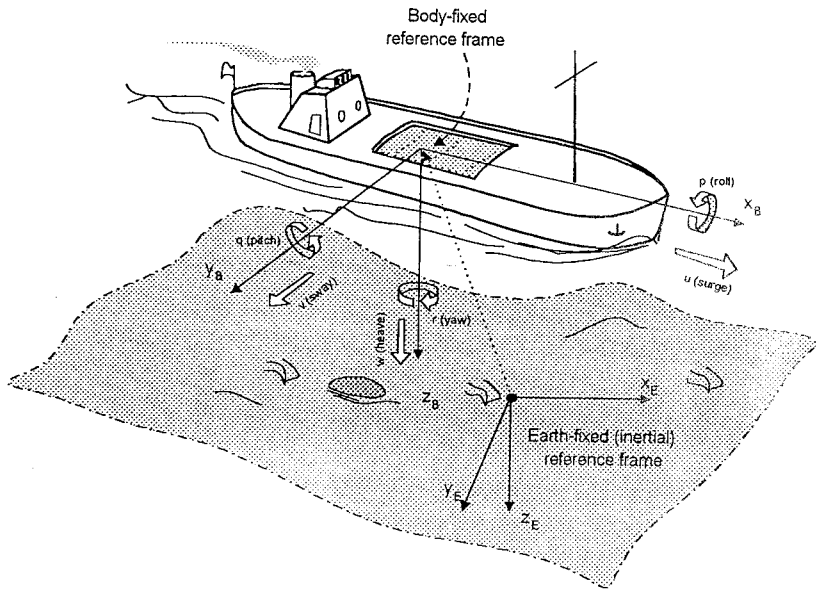


Fig. 2. Coordinate frame for a ship.

The state variable linearized mathematical model for a ship with a rudder servosystem in the horizontal plane (small perturbation) is:

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{\beta} \\ \dot{\eta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & a_{25} \\ 0 & a_{32} & a_{33} & 0 & a_{35} \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} \psi \\ r \\ \beta \\ \eta \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} \delta_c + \begin{bmatrix} 0 & 0 \\ \gamma_{21} & \gamma_{22} \\ \gamma_{31} & \gamma_{33} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} N \\ Y \end{bmatrix}$$

where :

ψ - *ship's heading (course),*

$r = \dfrac{d\psi}{dt}$ - *ship's heading rate,*

β - *drift angle,*

η - *the lateral offset or deviation from the desired path,*

δ_c *and* δ - *commanded rudder deflection and rudder deflection respectively.*

Planar Motion Mechanism (PMM) and oblique tow tests were conducted for Tokyo Maru ($\approx 180,000$ *DWT) tanker at various water depth-to draft ratio* H/T. *It was shown that dynamics of the tanker can change dramatically from being course stable at high* $H/T > 3$ *to being course unstable at low* $H/T < 3$ *ratio. The ship dynamics can change also due to:*

- *cargo (full load or ballast),*
- *trim,*
- *disturbances,*
- *etc.*

Example 3: **Airplane dynamics, [4].**
The dynamics of an airplane depend on:

- *speed,*
- *altitude,*
- *angle of attack,*
- *etc.*

The linearized mathematical model around stationary flight conditions (speed=const., altitude=const. and small angle of attack) with state variables N_z - *normal acceleration,* $q = \frac{d\Theta}{dt}$ - *pitch rate,* and δ_e - *elevons angle is:*

$$\begin{bmatrix} \dot{N}_z \\ \dot{q} \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} N_z \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} b \\ 0 \\ a \end{bmatrix} u$$

Flight Condition	1	2	3	4
Mach $(1Mach \approx 334[m/s])$	0.5	0.85	0.9	1.5
Altitude $[feet]$	5000	5000	35000	35000
a_{11}	-0.9896	-1.702	-0.667	-0.5162
a_{12}	17.41	50.72	18.11	26.96
a_{13}	96.15	263.5	84.34	178.9
a_{21}	0.2648	0.2201	0.08201	-0.6896
a_{22}	-0.8512	-1.418	-0.6587	-1.225
a_{23}	-11.39	-31.99	-10.81	-30.38
b	-97.78	-272.2	-85.09	-175.6
λ_1(*short period dynamics*)	-2.07	-4.90	-1.87	-0.87+j4.3
λ_2 (*short period dynamics*)	1.23	1.78	0.56	-0.87-j4.3

First three flight conditions represents subsonic speed while the fourth is supersonic speed flight condition. The airplane is unstable at subsonic speeds and stable at supersonic speed. Constant parameter autopilot is not a good solution for this particular airplane (F4-E with canards[1]) so the gain scheduling autopilot with dynamic pressure and Mach number as scheduling variables is used here.

There are many examples which can show that the assumption of time invariance is not fulfilled, and that the conventional constant parameter regulator is not appropriate. We then have to try to find the type of the controller to solve the problem. Two possibilities are at the moment at our disposal:

1. adaptive controller,
2. robust controller.

Contrary to adaptive controller which is also time variant element, robust controller belongs to constant parameters type of controllers. However, we must assume that bounds on the uncertainty of the process parameters are known in advance to be able to design the robust controller. We must also assume that our process will cover during operation just those conditions which we predicted before. If the process somehow change in unpredictable way that we didn't predict before designing it, than again we will have a problem, where probably the adaptive controller will be used instead of robust one .

B. Mathematical models used in adaptive control

For any control, and adaptive control in particular, it is necessary to choose the class of mathematical models which describe the process. The chosen mathematical model has to describe the dynamics of the process as well as to be appropriate for application in control design. Because of that, mathematical models will be defined first, followed by procedures for parameter estimation, and design of the adaptive regulator.

When self-tuning control is applied, it is necessary just like in any other case to choose the mathematical model which will be used. The chosen mathematical structure has to comply to certain conditions defined by a designer. With self-tuning control the mathematical model must allow description of input/output behavior for various processes. For that purpose, most often the linearized model around the nominal working point is used:

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{B_1(s)}{A_1(s)} e^{-\tau s}$$

[1]Canards make it easier to manoeuver the airplane, at the cost of decreased stability.

where:

$s = \sigma + j\omega$ - complex variable (frequency),
$G_p(s)$ - process transfer function,
$Y(s)$ - process response (output) signal,
$U(s)$ - process excitation (input) signal,
$A_1(s)$, $B_1(s)$ - polynomials of complex variable s of degree na and nb respectively,
τ - process time delay.

This type of mathematical model is general enough to describe various processes. It is essential to notice that the concentrated parameter model has to be linearized around the working regime. To satisfy the causality condition of any physical process, the following is necessary: $\deg(A_1) \geq \deg(B_1)$ or $na \geq nb$.

Due to the fact that processes are influenced by more or less stochastic disturbances the model must be augmented with the disturbance signal $w(t)$ as:

$$Y(s) = \frac{B_1(s)}{A_1(s)} e^{-\tau s} U(s) + W(s) \tag{1}$$

By introducing the differential operator $p = d/dt$ the model (1) becomes:

$$y(t) = \frac{B_1(p)}{A_1(p)} u(t - \tau) + w(t) \tag{2}$$

The algebraic equation (2) is easier to deal with, because the system can now be treated as an operator acting upon the input signal to form the output signal. See:[5], [6], [7], [8], and [9]

Disturbance signal $w(t)$ is generally composed of the following components:

1. constant or drift component $w_1(t) = const. = d_0$, or $w_1(t) = d_0 + d_1 t + \cdots + d_{nd} t^{nd}$ which results in real situation when a zero mean stochastic disturbance is biased by a constant or drift component. Sea current acting on a ship belongs to this type of disturbance,
2. slowly varying disturbance $w_2(t)$ such as the disturbance caused by a developing sea or the influence of stochastic disturbances in instances when a ship is travelling close to another ship in a channel,
3. disturbance which can be measured $w_3(t)$, and because of that used for the feedforward disturbance compensation. Wind disturbance acting on a ship can be categorized to this class of disturbances,
4. stationary stochastic component $w_4(t)$, which can be generated by white noise colored by a linear stable filter (shaping filter) for disturbance $w_4(t)$. This type of disturbance can represent for instance the disturbance of a developed sea on a ship. The shaping filter is given by:

$$w_4(t) = \frac{C_1(p)}{A_2(p)} e(t)$$

where:

$e(t)$ - white noise zero mean signal ($E\{e(t)\} = 0$) of independent equally distributed random variables with covariance function:

$$r(\tau) = \left\{ \begin{array}{c} \sigma_e^2 \text{ for } \tau = 0 \\ 0 \text{ for } \tau \neq 0 \end{array} \right\},$$

$A_2(p)$, $C_1(p)$ - shaping filter polynomials.

The majority of references in adaptive control use almost exclusively $w_4(t)$ component of the disturbance signal, despite the fact that all four components probably describe the real situation more precisely.

The mathematical model used quite often in self-tuning control is then given by:

$$y(t) = \frac{B_1(p)}{A_1(p)}u(t-\tau) + \frac{C_1(p)}{A_2(p)}e(t)$$

Because a digital computer is mostly used for adaptive control, discretization of this model is necessary. The step invariant (ZOH) discretization with the sampling period h will give:

$$G_p(z) = (1-z^{-1})Z\left\{\frac{G_p(s)}{s}\right\} = \frac{B_1(z)}{A_1(z)}$$

where:

z - complex variable (frequency),
$G_p(z)$ - process transfer function in the z domain,
$A_1(z) = z^{na} + a_1 z^{na-1} + ... + a_{na}$,
$B_1(z) = b_o z^{nb} + b_1 z^{nb-1} + ... + b_{nb}$,
na, nb - degree of polynomials A and B respectively,
Z - z-transform operator.

Here we should be aware of problems which can arise due to discretization of the continuous process. It is well known that the transfer function of the continuous system (which is a rational function of the complex variable s) has na finite poles, nb finite zeros and $d = na - nb$ zeros at infinity[2] (iff $na > nb$). Namely, na poles of the continuous system are mapped to the na poles of the discretized system by the law $z = e^{sh}$, (where h is the sampling period). However, zeros are not mapped by the same law. The discretized system in general has $na - 1$ finite zeros. For particular values of the sampling period. some zeros may go to ∞ or they may be cancelled by poles, resulting in hidden modes, see [10]. Hidden modes should not be considered as zeros of the discretized system. The discretization process also generates $d - 1$ finite zeros. So, we can say that in general discretized system will have na mapped poles, nb mapped zeros and $d - 1$ finite zeros. It is a rule more than an exception that by discretization of the continuous minimum-phase system, a non-minimum-phase discretized system is obtained. The positions of zeros of the discretized system depend on the sampling period and also on the type of the reconstructor used. There are in general no simple closed-form expressions for mapping the zeros. The limiting cases for small or large sampling periods can be characterized by the following examples (Theorems are given in [11]):

1. If the continuous system has $na - 1$ or na zeros, then those zeros will be mapped according to $z \approx e^{sh}$ only for small sampling periods. Consequently minimum-phase continuous system will be mapped to a minimum-phase discrete system.
2. If $G_p(s)$ has d more poles than zeros (pole excess is d) then for small sampling periods, the discretized model will be proportional to the process with d integrators. If $d > 2$ then at least one zero will be outside of the unit circle, and as a consequence the minimum-phase continuous system will be mapped to a non-minimum-phase discrete system.
3. If $G_p(s)$ has $na > nb$, then for large sampling periods, the discretized mathematical model will tend toward $G_p(0)/z$ having no finite zeros.
4. If $G_p(s)$ is causal and if τ is the delay time, which is not equal to an integer factor of the sampling period (fractional delay time is then $d = kh - \tau$). Then, if $G_p(s)$ is causal and if $d \to h$ at least one zero of the discretized system will be outside of the unit circle. As an example $e^{-\tau s}/s$ will have a discretized model with a zero outside of the unit circle if $h/2 < d < h$.

It is often the case that a continuous system is sampled with a high frequency. This is justified from the standpoint of better approximation of the continuous system by the discretized one, but sometimes it is not recommended in design due to effects mentioned. Because of these facts, the designer does

[2] d is called pole excess for continuous system.

not want the regulator he is designing to depend upon the positions of zeros of the original system (process).

Assuming that $h = 1$ the following difference equation for the process without disturbances is obtained:

$$y(k+na) + a_1 y(k+na-1) + ... + a_{na} y(k) = b_o u(k+nb) + ... + b_{nb} u(k) \tag{3}$$

By introducing the shift operator q defined by: $qf(k) = f(k+1)$ the equation (3) becomes[3]:

$$A_1(q)y(k) = B_1(q)u(k) \tag{4}$$

where:

$A_1(q) = q^{na} + a_1 q^{na-1} + .. + a_{na}$,

$B_1(q) = b_o q^{nb} + b_1 q^{nb-1} + .. + b_{nb}$,

The equation (4) can be transformed by use of the delay operator q^{-1} defined by: $q^{-1}f(k) = f(k-1)$, to:

$$A_1^*(q^{-1})y(k) = q^{-d} B_1^*(q^{-1})u(k) \tag{5}$$

where:

$A_1^*(q^{-1}) = 1 + a_1 q^{-1} + .. + a_{na} q^{-na}$,

$B_1^*(q^{-1}) = b_o + b_1 q^{-1} + .. + b_{nb} q^{-nb}$, $b_o \neq 0$,

$d = na - nb$

Polynomials labeled by superscript * are reciprocal polynomials[4], obtained by reversing the order of the coefficients in the original polynomial as: $A^*(q^{-1}) = q^{-na} A(q)$. Some care must be exercised when operating with reciprocal polynomials, because A^{**} is not necessarily the same as A.

Example 4:

$A(z) = z$ has the reciprocal polynomial $A^*(z) = 1$ while the reciprocal polynomial of $A^*(z)$ is $A^{**}(z) = 1$ which is apparently different from $A(z)$.

By a similar procedure, the discretized model of the disturbance can be obtained as:

$$A_2(q)w_4(k) = C_1(q)e(k)$$

where:

$e(k)$ - zero mean discrete-time white noise sequence of independent equally distributed random variables with covariance function:

$$r(\tau) = \left\{ \begin{array}{c} \sigma_e^2 \text{ for } \tau = 0 \\ 0 \text{ for } \tau = \pm 1, \pm 2, ... \end{array} \right\}$$

$w_4(k)$ - disturbance sequence,

$A_2(q), C_1(q)$ - disturbance shaping filter polynomials.

Introducing the following relations:

$$A = A_1 A_2$$
$$B = B_1 A_2$$
$$C = C_1 A_1$$

[3] Auto Regressive eXogenous (ARX) model.

[4] Later on, we will abandon use of the superscript * to have simpler designations.

we obtain:

$$A(q)y(k) = B(q)u(k) + C(q)e(k) \tag{6}$$

or:

$$A^*(q^{-1})y(k) = q^{-d}B^*(q^{-1})u(k) + C^*(q^{-1})e(k) \tag{7}$$

which is the well known ARMAX (Auto Regressive Moving Average eXogenous) model. This model is used not only in control but also in econometrics and elsewhere, [12], [13]. Often the name CARMA (Controlled AutoRegressive Moving Average) model is also used, [14].

Mathematical model (6) is more convenient when analyzing the system using the characteristic equation, such as stability analysis or defining the degree of the system. The reason for that is that shift operator q used here will not disregard poles in the origin. Mathematical model (7) is appropriate for causality analysis and numerical calculations.

ARMAX models and state variable stochastic models are equivalent models from the input/output point of view. However, ARMAX models have parsimonious parametrization, [12] and because of that are more often used in parameter identification. The principle of parsimony defined by Akaike [13], says that with other characteristics equal mathematical models with a smaller number of parameters should be preferred. This is the main reason why ARMAX models are preferred in identification and self-tuning adaptive systems. The ARMAX mathematical model is the basic model from which other models can be obtained. Clarke and Gawthrop [15], added a constant (bias) disturbance component to the right side of the equation (7) to get the possibility of modelling the bias component of the disturbance $w_1(t)$:

$$A^*(q^{-1})y(k) = q^{-d}B^*(q^{-1})u(k) + C^*(q^{-1})e(k) + w_1(k) \tag{8}$$

If we include in the picture the measurable disturbance $w_3(t)$ and the drift type disturbance $w_1(t)$ the discretized model becomes:

$$A^*(q^{-1})y(k) = q^{-1}B^*(q^{-1})u(k) + C^*(q^{-1})e(k) + F^*(q^{-1})v(k) + w_1(k) \tag{9}$$

where:

$v(k)$ - measurable disturbance,

$w_1(k)$ - drift disturbance which in general can be modelled as a polynomial function of time, given by:

$$w_1(k) = d_0 + d_1k + d_2k^2 + \cdots + d_{nd}k^{nd}$$

The block diagram of the discrete-time stochastic process model (9) is given in Fig.3.

$nd = 1$ gives that $w_1(k) = d_0$ and this covers many practical situation for constant drift acting upon a process. When $w_1(k) = 0$ and $v(k) = 0$ we end up with ARMAX (CARMA) model discussed already.

If the disturbance is Brownian motion type disturbance, $j(k) = j(k-1) + e(k)$, (j is non-zero mean stochastic signal $E\{j(k)\} \neq 0$), it is more convenient to use so called CARIMA (Controlled AutoRegressive Integrated Moving Average) model, [16]:

$$A^*(q^{-1})y(k) = q^{-d}B^*(q^{-1})u(k) + C^*(q^{-1})\frac{e(k)}{\Delta} \tag{10}$$

where:

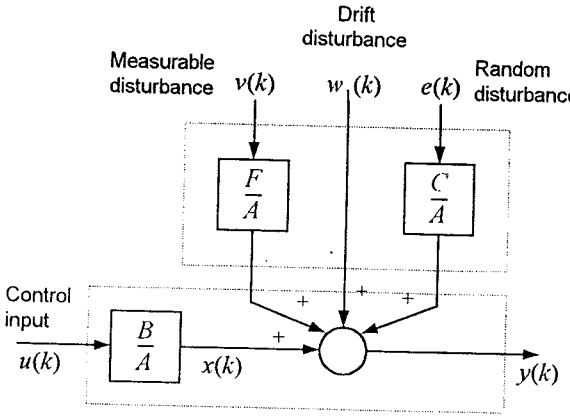


Fig. 3. Block diagram of a discrete-time stochastic ARMAX (CARMA) process.

$\Delta = 1 - q^{-1}$ is the difference operator.

The CARIMA model (10) is more convenient than CARMA model (7) in practical situations, when disturbances are not stationary. In many industrial processes, but also in some ship sailing regimes, two types of nonstationary disturbances are encountered:

- step-like disturbance, occurring in stochastic instances. The change of quality of products has this character, as well as the sudden change of depth under the keel of a ship sailing in a channel.
- Brownian type disturbances occurring in stochastic systems based on energy balance. A ship sailing close to the coast experiences this type of disturbance.

To obtain tracking control systems, the regulator must have integral mode. If the CARIMA model is used as the mathematical model of a process, then the model-based design will result with the regulator having the necessary integral mode.

In the literature various models can be found which are based on the ARMAX form. The choice of the mathematical model is dictated not only by an accurate description of the process dynamics, but also by the control configuration, or the role of the regulator set by the designer. In the following text the superscript * will be abandoned and instead of $A^*(q^{-1})$ we will use $A(q^{-1})$. The reason is that this will be more convenient, because from now on mainly the reciprocal polynomials will be used.

C. Problem formulation

Self-tuning adaptive control is derived from the larger class of optimal adaptive control systems, where the regulator must obtain two different functions:

- real time (on-line) process identification (recursive identification),
- control according to some chosen algorithm.

During the developing stages of adaptive control theory, many definitions for the term "adaptive control" were recommended. Today, a specific approach to the design of suboptimal controllers for nonlinear stochastic systems is implied for "adaptive control". Even though the structure of the self-tuning adaptive control is based on heuristic arguments, it can be carried out from basic principles of stochastic optimal control theory, when the system as well as its environment are described by stochastic models.

Stochastic optimal control theory, [17], [18], is concerned with the problem of finding the appropriate control signal which will minimize a performance index subject to constraints on the nonlinear process described by the mathematical model (discrete-time case):

$$\begin{aligned} x_{k+1} &= f_1(x_k, u_k, w_k, k) \\ y_k &= f_2(x_k, v_k, k) \end{aligned} \tag{11}$$

where:

k - time index,

x_k - state vector,

y_k - response (output) vector,

u_k - control vector,

w_k - vector of process disturbances,

v_k - measuring noise vector,

For nonlinear systems the main result of stochastic optimal control theory, [19], is in the recognition that the optimal controller for stochastic nonlinear processes must posses two distinct functions:

1. Estimation of the conditional probability distribution of state x_k, based upon measured previous input and output signals,

$$p(x_k \mid y_{k-1}) \quad ; \qquad y_{k-1} = (u_1, u_2, ...u_{k-1}, y_1, y_2, ...y_{k-1}) \tag{12}$$

   This distribution is called the problem hyperstate.

2. Control algorithm u_k which can be obtained by use of:
   - known estimation of the conditional probability distribution of state x_k,
   - known previously measured input/output signals,

$$u_k = f[p(x_k \mid y_{k-1})] \tag{13}$$

The expression (13) can be interpreted as a nonlinear function which maps the hyperstate to the space of control variables.

The control system structure based on these principles has one big disadvantage in the necessity of calculating the conditional probability distribution during system operation. The general structure of any stochastic optimal control system is given in the block diagram in Fig.4, from [19].

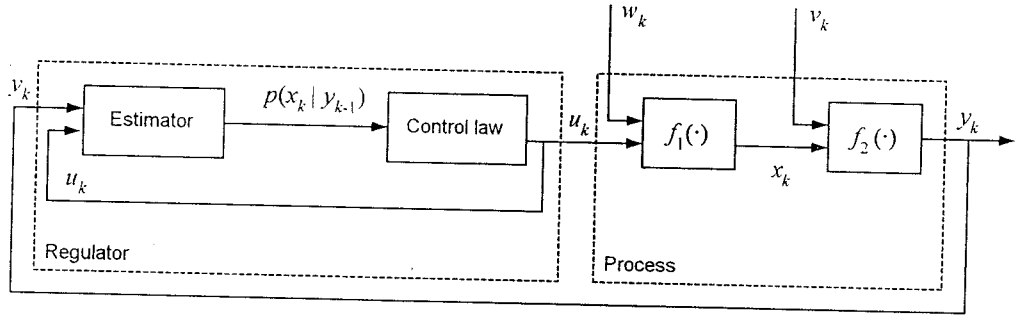


Fig. 4. Optimal stochastic control structure.

Optimal control in the sense described above, can be determined for a specific class of systems, i.e. systems having linear processes ($f_1(.)$ and $f_2(.)$ are linear), quadratic performance indexes, and stochastic variables (w_k, v_k and x_k) with Gaussian type probability distribution. These problems are known as LQG (Linear Quadratic Gaussian) problems. For all other systems control will be suboptimal. Namely, real processes almost always contain nonlinearities, the performance index is rarely justified as quadratic, and stochastic disturbances do not have to be Gaussian.

The first assumption, needed in order to be able to solve the problem of adaptive control, is that we are dealing with the LQG problem, and that the real process can be described by a linear mathematical model. Synthesis of the linear controller can then be done. Every linear controller can be designed by use of conventional or optimal control theory methods. In both cases controller design uses the linear

mathematical process model with known coefficients a_i, b_i, c_i, and possible drift disturbance w_1:

$$\sum_{i=0}^{na} a_i y(k-i) = \sum_{i=0}^{nb} b_i u(k-d-i) + \sum_{i=0}^{nc} c_i e(k-i) + w_1$$

or:

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + C(q^{-1})e(k) + w_1 \tag{14}$$

where:

$e(k)$ - stochastic white noise signal with $E\{e\} = 0$ and $E\{e^2\} = \sigma_e^2$

w_1 - DC component of the disturbance,

d - system delay,

$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_{na} q^{-na}$,

$B(q^{-1}) = b_o + b_1 q^{-1} + b_2 q^{-2} + ... + b_{nb} q^{-nb}$; $(b_o \neq 0)$,

$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_{nc} q^{-nc}$,

na, nb, nc degree of A, B and C polynomials respectively,

Optimal adaptive control systems are derived from optimal control, and they are intended for control of nonlinear time-varying stochastic processes, i.e. for control of processes which need adjustment of controller parameters. Their analysis and synthesis is complex. Namely, if processes were linear they could be controlled with a linear optimal controller, and if they were time-invariant and deterministic we will not need learning during operation in the form of self-tuning and/or estimation. Contrary to optimal control, which is based upon the assumption that coefficients of the mathematical model (14) are known and time invariant, here the basic assumption is that those coefficients (or some of them) are not known, and that they are changing during system operation, so their estimation is needed.

Optimal adaptive systems can be represented by block diagram in Fig.5.

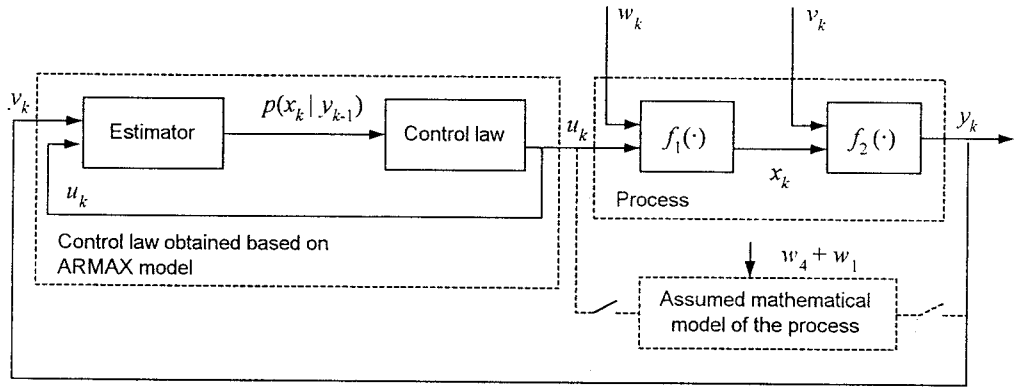


Fig. 5. The structure of optimal adaptive control system.

The optimal control law has one interesting characteristic, namely that the control signal despite its function of controlling the process according to some performance index, has also the function of perturbing the process in order to improve parameter (or state) estimation. So, by doing that this type of control tends to somehow balance the need for small control signals with the need for accurate estimation. This type of control is called dual control because of that.

The control algorithm is formed based on an assumed mathematical model (14), which is linear only if all coefficients are known. That the model (14) is nonlinear, if coefficients are not known, can be shown if this model is expressed by state variables, where unknown coefficients are treated as augmented state variables.

Example 5:
For the first order system ($na = nb = nc = 1$) from (14) follows, from [19]:

$$y_k + ay_{k-1} = bu_{k-1} + e_k + ce_{k-1} + w_1 \tag{15}$$

System dynamics can be represented by one state variable:

$$x_5 = y - e \tag{16}$$

while four unknown coefficients a, b, c and disturbance w_1 can represent augmented state variables:

$$x_1 = -c; \quad x_2 = b; \quad x_3 = c - a; \quad x_4 = w_1 \tag{17}$$

From (15), (16) and (17) the nonlinear mathematical model is obtained:

$$x_5(k+1) = x_1(k)x_5(k) + x_2(k)u(k) + x_3(k)y(k) + x_4(k) \tag{18}$$

Equation (18) shows that not knowing coefficients in (15) generates one dominant nonlinearity x_1x_5 and two bilinear nonlinear components of smaller impact (x_2u and x_3y). Not knowing coefficients of $C^*(q^{-1})$ polynomial (x_1 in (18)), which models disturbance effects on a process, results in a dominant nonlinearity, and makes estimator design and control algorithm design difficult. Not knowing coefficients of $B^*(q^{-1})$ polynomial (x_2 in (18)), which defines how the process is connected with its environment, does not have such an impact. Not knowing coefficients of $A^*(q^{-1})$ polynomial (x_3 in (18)), which define the process dynamics, also results in a bilinear component of smaller impact. Not knowing a bias component of the disturbance (x_4 in (18)) does not result in any nonlinearity, but can cause unobservability of a system.

C.1 Some structural properties of stochastic control

Stochastic optimal control theory defines specific concepts such as: neutrality, separability and certainty equivalence. These concepts characterize optimal stochastic systems and are important in design of adaptive control systems, [20], [21], [22].

- Neutrality, [23], defines the characteristic of those stochastic control systems for which the estimation improvement is not dependent on the control signal. This is a characteristic of the estimation function of the stochastic regulator which shows that the form of the hyperstate $p(x_k \mid y_{k-1})$ is independent of previous control signals. We can say that neutral control excludes probing signals in the control signal.
- Separability is a characteristic of stochastic control systems for which the control signal is calculated upon the point state estimation $\widehat{x}_k$ and not its conditional probability function $p(x_k \mid y_{k-1})$. Contrary to the general optimal stochastic regulator which should use $p(x_k \mid y_{k-1})$ for calculating the control signal, optimal regulators of the separable stochastic problem use $\widehat{x}_k$ instead. We can conclude that separability is a characteristic of the controller algorithm which shows that the optimal control is independent of the accuracy of information about the state at the particular moment. This means that the probing signals, needed for improving the accuracy of state estimation, and cautious control, needed because of the inaccuracy of state estimation, become superfluous in separable problems.
- Certainty equivalence is a more strict concept than neutrality or separability. It relates only to the control function of the optimal regulator of the separable problem. The separable problem is said to be certainty equivalent when the control function of the optimal stochastic regulator $u^c = f_1^c(x_k)$ is identical to the control function of equivalent deterministic optimal regulator $u^c = f_2^c(x_k)$, which does not have unreliable information about process state. Certainty equivalence imply that the control law of the stochastic system can be designed without taking into

consideration stochastic effects. For specific classes of problems (such as LQG problems) it is proved to possess all three characteristics because the sufficient conditions are:

1. For neutrality - linearity of $f_1(\cdot)$ and $f_2(\cdot)$ of a process mathematical model (11), [22],
2. For separability - linearity of $f_1(\cdot)$ and $f_2(\cdot)$ of a mathematical model (11), with stochastic variables with Gaussian probability distribution, [17],
3. For certainty equivalence - linearity of $f_1(\cdot)$ and $f_2(\cdot)$ of a process mathematical model (11) together with the quadratic performance index, [22].

The importance of theses concepts lies in the possibility to define the basis of design of suboptimal control systems, like an adaptive one. Because the linearity of the mathematical model is the basic requirement for all three characteristics, it follows that adaptive control will not posses neutrality, separability and certainty equivalence if we do not know process parameters. So, we will need parameter estimation to regain the certainty equivalence type of the regulator. This means that an adaptive control algorithm should be designed in such a way to allow mutually conflicting activities of two basic functions of suboptimal control: estimation and control. When this is obtained we have a certainty equivalent type regulator.

As was already said, an adaptive regulator always consists of state and/or parameter estimation as well as a control algorithm. The control signal generated in the regulator must posses the ability to somehow harmonize these functions which asks for mutually conflicting actions. Namely, estimation asks for adequate process excitation in order to be able to identify process parameters, while the control algorithm should reach the desired goal (performance index) with minimal control effort. These mutually conflicting requirements for the control signal can be represented by splitting the control signal to:

1. cautious control, when unfamiliarity with the proper state (or parameters) of a process has as a consequence uncertainty about the accuracy of the obtained estimation of state. This necessarily results in more cautious control than if this uncertainty is not present,
2. probing control, injected during normal operation in order to get faster and more accurate state and/or parameter estimation.

D. Methods of adaptation

For better understanding the problem of adaptive control, some concepts and principles valid for the particular class of stochastic systems (such as self-tuning adaptive system) has to be known.

Parameter adaptive control is usually implemented as:

- gain scheduling adaptive control
- model reference adaptive control
- self-tuning adaptive control

Due to the time limitation of this tutorial, only the self-tuning control will be covered in more detail.

D.1 Gain scheduling adaptive control

The block diagram of this type of control is given in Fig.6.

Gain scheduling control can be applied in those situations when it is possible to determine a change in process dynamics by measuring some variable(s) that correlate well with this change. In flight control systems, the height of flight and the speed of the airplane can be used for this purpose. In marine control systems the autopilot parameters are scaled according to ship speed. The advantage of this structure is that it is fast in adaptation, easy to implement and reliable in operation. However, the main disadvantage is that the process may experience some unexpected change, which were not previously modeled and memorized in the table of gains. Some authors does not categorize this control structure as adaptive control, because strictly speaking the adaptive loop does not exist in the gain scheduling structure.

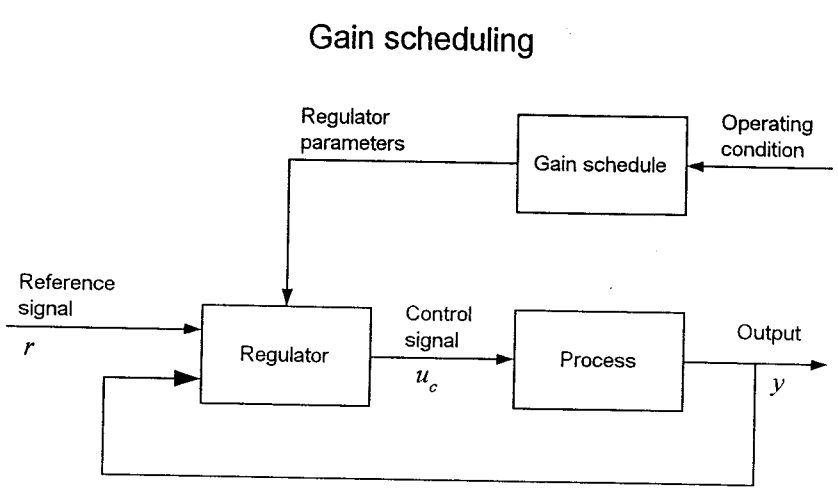


Fig. 6. Gain scheduling adaptive control.

D.2 Model reference adaptive control

The block diagram of the model reference adaptive control system is given in Fig.7.

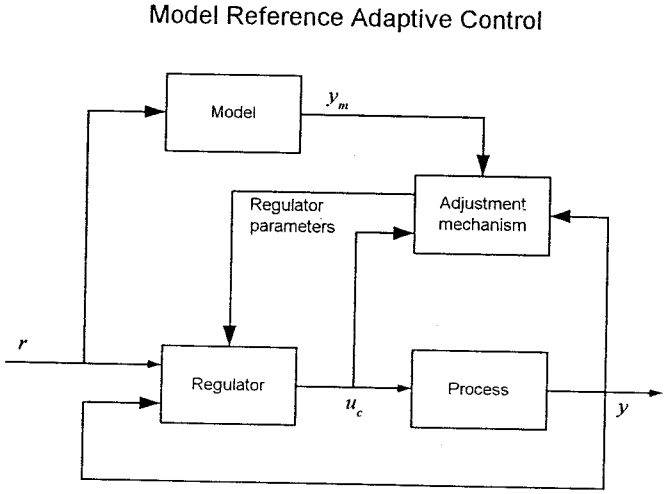


Fig. 7. Model reference adaptive control.

The main characteristic of this structure is that the model of the closed-loop dynamics must be carefully chosen. During operation, the model and a real process output are compared and the control signal accordingly changed in order to diminish the difference between the model and the closed-loop system. If the model is not properly chosen, this type of control can give large control signals and consequently cause excessive wear in the actuator. Because of this disadvantage the MRAS system is used mainly for deterministic systems with very good results.

D.3 Self-tuning adaptive control

The block diagram of the self-tuning (ST) adaptive control system is given in the Fig.8.

The structure of ST adaptive control have the necessary components for applying it in the stochastic setting. In the Fig.8 the indirect structure is given where the parameters of the process are first estimated, and then the parameters of the regulator are designed, based upon the estimated process

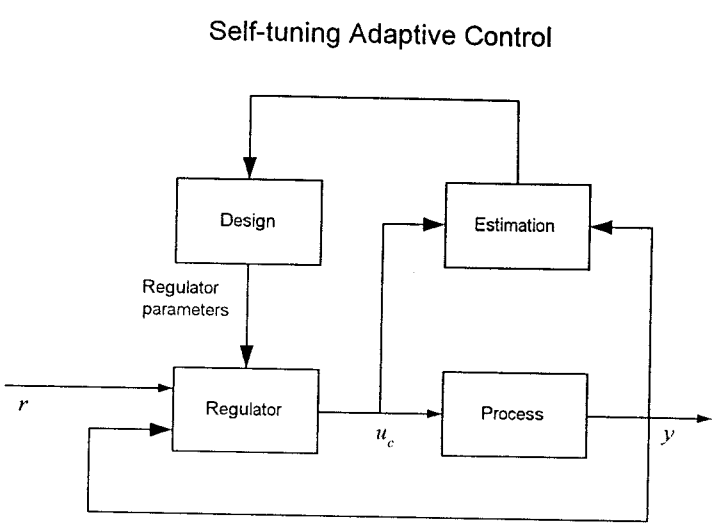


Fig. 8. Self-tuning control system structure.

parameters. The direct form unites these two steps in one, so the estimator gives automatically the parameters of the regulator, which must be given in the functional relation with process parameters. The advantage of the ST structure is that it is capable of resolving many practical situations where we do not now in advance what type of change our process will go through. Contrary to MRAS systems, which can be realized in analog and digital form, the ST system is almost exclusively realized in a digital technique. Because of the necessity of estimating parameters, the ST control is sometimes slower than the MRAS control.

II. System identification

System identification covers two main tasks:

1. Mathematical model building,
2. Parameter identification.

While the mathematical model building is concerned with obtaining the mathematical model which will describe the dynamics of the process in the best possible way, usually resulting in a complex nonlinear model, the parameter identification, especially for ST adaptive control purposes, has to find parameters of the model, which doesn't have to be as complex and exact, but instead have to cover the basic dynamics of the process.

Parameter identification includes:

- choice of experimental conditions - planning of experiment,
- selection of model structure,
- parameter estimation,
- model validation,

A. Choice of experimental conditions

When the experimental condition has to be chosen we have to meet the following requirements:

1. experiment has to be cost effective,
2. experiment has to be simple,
3. identifiability must be met (i.e. all parameters could be possible to identify),
4. choice of excitation signal must be such to enhance interesting parameters and parameter combinations, i.e. all system modes must be excited.

It should be stressed that under closed-loop operation, identifiability can be lost due to too simplistic feedback (like proportional one) or because of using not-persistent excitation. However, ST adaptive systems have a natural time variation in the feedback, because the feedback gains are based on parameter estimates, and consequently the identifiability problem does not exist in adaptive self-tuning systems. Test signals should be chosen to be as far as possible similar to input signals used during normal operation. Of test signals used for identification purposes we will mention here only two:

1. Pseudo Random Binary Signals (PRBS), which are convenient test signals for identification purposes, because they have only two amplitude levels and a rich spectrum. Limited amplitude is useful, since it is generally necessary to limit the amplitude of a test signal to avoid taking a system outside its linear operating range.
2. Square wave signals which are also convenient and have advantages that they are easy to generate and also have strictly limited amplitude. The disadvantage is that their spectrum is not so rich as PRBS spectrum. Square wave amplitude should be selected in such a way that the process is not pushed into a nonlinear operating regime, and the frequency should be $\approx 0.16BW$ or the period of the square wave signal should be approximately six time larger than the dominant time constant of the process ($T_{SW} \approx 6T_d$). These are rough guides aimed at ensuring that most of the square wave power (associated with the first three harmonic component) is inside the system bandwidth.

B. Selection of the model

Four factors that should be taken into account when selecting the model are:

- flexibility - the model should be capable of describing different system dynamics which can be expected during normal operation, keeping in mind that the number of parameters and the way they enter the model are very important,
- parsimony - principle of parsimony says: "If other things are equal, the model with the smallest number of parameters should be preferred".

- algorithm complexity - depends on the number of parameters to be identified, as well as on the structure of the model,
- properties of the criterion function - does this function has global minimum, maximum, local extremal points etc.

Example 6:
Criterion function:

$$J(\Theta) = \sum_{k=1}^{N} g\left[\varepsilon(k)\right] \tag{19}$$

where:
Θ - *vector of parameters,*
ε - *some type of error (input error, output error or a generalized error),*
$g\left[\cdot\right]$ - *criterion function at user's disposal*
For disturbances with Gaussian distributions it is optimal to let $g\left[\cdot\right]$ *be a quadratic function of* ε,*i.e.* $g\left[\varepsilon\right] = \varepsilon^2$.

C. Parameter estimation

The quality of the parameter estimation depends on the performance (criterion) chosen, method used and type of identification. The identification can be calculated:

- off-line,
- on-line.

For off-line identification, optimal accuracy is achieved, and Cramer-Rao lower bound obtained if the criterion function $g\left[\varepsilon(k)\right]$ is chosen as the maximum likelihood one:

$$g\left[\varepsilon(k)\right] = -\log \overline{f}\left[\varepsilon(k)\right] \tag{20}$$

where:
$\overline{f}$ - probability density function of the prediction errors
When the measured data set contains some values that are abnormal (outliers) due to sensor failures (or other cause), use of quadratic criterion function will give substantial jumps of the parameter estimates, and moreover, a long time elapses before the estimates converge back to their previous level - recovery is slow.
The algorithm can be made robust by the following interventions:

1. filtering all data before processing them,
2. making reliability checks on data before processing them. For instance comparing prediction errors with a specified limit. Large prediction errors means that an outlier or measurement error is probable. The predicted value can than be substituted for the measurement. This is applicable when there are only few outliers in the data.
3. using a criterion function that grows more slowly with ε than the quadratic one, so that large prediction errors will have less influence on the parameter estimate. Instead of using ε use the following:

$$f(\varepsilon) = \frac{\varepsilon}{1 + a\left|\varepsilon\right|}$$

D. Model validation

The main problems attracting attention of researchers are still:

- choice of the model,
- choice of the model order.

The choice of the model for SISO systems is based on flexibility and parsimony. The model for the identification purposes should be chosen from the general family of models given by:

$$A(q^{-1})y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}e(k) \tag{21}$$

With total number of parameters given, the best flexibility is usually obtained if they are spread out to some different polynomials.

If $A \equiv C \equiv D \equiv F \equiv 1$ the model (21) becomes $y(k) = B(q^{-1})u(k) + e(k)$. This model (with a limited number of parameters) will certainly not be good for slow systems having a slow impulse response.

The trade-off between flexibility and parsimony should thus best be met by using three or more polynomials. Most often used model is ARMAX model:

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

If we want good prediction than the following model should be used:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{1}{D(q^{-1})}e(k)$$

The recommended model for estimating the dynamics of a system is:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + e(k)$$

or

$$A(q^{-1})y(k) = B(q^{-1})u(k) + e(k)$$

Some comments are necessary here:

1. If it is known or expected that the system is unstable, then $A(q^{-1})$ must be included in the model.
2. If the system is unstable (assuming it is stabilized during experiment with an appropriate feedback) it is important to note that $F(q^{-1})$ is constrained to be asymptotically stable.
3. It is common practice to use $A(q^{-1})$ or $F(q^{-1})$, but not both in the model.

The choice of the model order is not a trivial problem. Careful trade-off between good description and model complexity is needed. Most methods of model order selection are developed for the off-line situation: Compare the performance of models of different orders and test if the higher order models is worthwhile. For on-line identification this idea will require parallel identification of several models which is clumsy, especially in the adaptive control setting, so here we usually fix the model order, and this data becomes the initial data for the recursive identification algorithm.

III. Least squares parameter identification

As was already said, self-tuning adaptive control consist of two basic modules: on-line parameter estimation module and regulator design module. Both modules are essential for proper functioning of the adaptive control system, because only then are we dealing with certainty equivalence type regulators. A description of the most often used algorithm for recursive identification will be given,

and some interventions needed to overcome possible pathological situations in the estimation algorithm will be briefly mentioned.

The references from the area of estimation theory and identification application are abundant today, and we will give a selective list. References [24], [25], [26] and [18] are today classic ones where the "black box" mathematical models are mainly used for identification purposes and none or very few apriori information about process dynamics are utilized during identification. In many situations this approach gives satisfactory results. However, it must be stressed that "grey modelling" could give more accurate results because we are using some reliable information about our process from the beginning. In on-line design of the regulator it is essential that the mathematical model describe input/output process dynamics, and it is not so important that its structure correspond to the process. ARMAX models are very popular because they can describe external dynamics of various processes. The parameter identification procedure chooses "the best" mathematical model from the chosen class of models, where the most important question is "does the chosen class of the mathematical models describe the external (input/output) process dynamics"? Validation of the model must give answers to the following questions, [27], [26]:

- does a chosen model correspond well with measured data from the real process ?
- is a chosen model good enough for the purpose set by the designer ?
- does a chosen model describe the "real system" ?

The answer to the third question is not possible, so the model is chosen if we can answer first two questions positively. Usually for model validation the following is used:

- AIC (Akaike Information Criterion) criterion, [13];
- test of likelihood between the residual and the white noise;
- comparison of good fit between the process response and the process model (with estimated parameters) response;
- comparison of the loss functions for various mathematical models etc.

All those performances give quantitative indices upon which the proper mathematical model can be chosen.

A. Least squares algorithm development

In order to identify process parameters, the model used must be linear in the parameters. The mathematical model (9) then has to be reparametrized to a regressive form:

$$y(k) = \Theta^T(k-1)\varphi(k) + e(k) = \varphi^T(k)\Theta(k-1) + e(k) \tag{22}$$

where:

$y(k)$ - measured process output signal,
$e(k)$- disturbance signal (white noise with $E\{e\} = 0$),
$\varphi(k)$- regression vector,
$\varphi^T(k) = [-y(k-1) \cdots \text{-}y(k-na)\ u(k-d) \cdots u(k-d-nb)\ v(k)\ v(k-1) \cdots v(k-nf)$
$1\ k\ k^2 \cdots k^{nd}\ e(k-1)\ e(k-2) \cdots e(k-nc)]$
$\Theta(k)$ - vector of process parameters,
$\Theta^T(k) = [a_1\ a_2 \cdots a_{na}\ b_o \cdots b_{nb}\ f_0\ f_1 \cdots f_{nf}\ d_0\ d_1 \cdots d_{nd}\ c_1\ c_2 \cdots c_{nc}]$

This mathematical model structure restricts possible control algorithms to be applied for process control. In this model, unknown parameters are linearly dependent, i.e. the output signal linearly depends on the parameters. If we do not know $e(k)$, which is the sequence of uncorrelated white noise realizations, we can think of:

$$\widehat{y}(k \mid \Theta) = \Theta^T(k-1)\varphi(k) \tag{23}$$

as a natural guess or *prediction* of what is $y(k)$ going to be, having observed previous values of $y(k-i)$ and $u(k-i)$: $i = 1, 2, \ldots$.So, we can say that *prediction depends on the* Θ *vector linearly.* If and only

if $e(k)$ is a sequence of independent random variables with $E\{e(k)\} = 0$ i.e. white noise sequence, then (23) is the prediction in the exact statistical sense.

Because $\widehat{y}(k \mid \Theta)$ depends on parameters linearly, the estimation problem is simple.

Subtracting (22) and (23) we get the *prediction error* as:

$$\varepsilon(k) = y(k) - \widehat{y}(k \mid \Theta) \tag{24}$$

If we have an exact description of the system given by (22), i.e. this model represent the true data-generating mechanism, and if we wish to determine from available data collected in the regression vector $\varphi(k)$, the true system parameters in the Θ vector, then under the assumption of the known correct structure, we can utilize the model:

$$y(k) = \varphi^T(k)\widehat{\Theta} + \widehat{\varepsilon}(k) \tag{25}$$

where:

$\widehat{\Theta}$ - vector of adjustable model parameters,

$\widehat{\varepsilon}(k)$ - fitting error at time k,

Our aim is to select $\widehat{\Theta}$ so that the overall modelling (fitting) error $\widehat{\varepsilon}(k)$ is minimized in some sense. Equation (22) and (25) imply:

$$\widehat{\varepsilon}(k) = e(k) + \varphi^T(k)\left[\Theta - \widehat{\Theta}\right] \tag{26}$$

i.e. the modelling (fitting) error depends on $\widehat{\Theta}$ and when $\Theta \approx \widehat{\Theta}$, the "minimized" fitting error will be equal to the white noise sequence corrupting the system output data.

$$\widehat{\varepsilon}(k) \approx e(k) \text{ when } \Theta \approx \widehat{\Theta}$$

Assume that the system described by (22) has been running for a sufficient time to form N consecutive data vectors. The data obtained in this way allows the model (25) to be expressed in the vector (matrix) form:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix} \widehat{\Theta} + \begin{bmatrix} \widehat{\varepsilon}(1) \\ \widehat{\varepsilon}(2) \\ \vdots \\ \widehat{\varepsilon}(N) \end{bmatrix} \tag{27}$$

To be able to uniquely find $\widehat{\Theta}$ we must have $N \geq npar$, (approximation problem), where $npar$ is the number of estimated parameters in the vector $\widehat{\Theta}$. When $N = npar$ we have the interpolation problem, and in the noise free case ($e(k) = 0$), the equation (27) can be solved as a set of linear equations in $npar$ unknowns. The resulting fitting errors are zero. When noise is present, and indeed in any practical situation, even in nominally noise-free systems, we must have $N >> npar$. Linear least squares methods are most widely used for such problems. Since there are more measurements (N) than unknowns ($npar$), we have to choose an estimator of Θ that minimizes in some arbitrary chosen sense the effects of errors. Rewriting equation (27) in the stacked notation, we have:

$$Y = \Phi\widehat{\Theta} + \widehat{E} \tag{28}$$

where:

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}; \; \Phi = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix}; \; \widehat{E} = \begin{bmatrix} \widehat{\varepsilon}(1) \\ \widehat{\varepsilon}(2) \\ \vdots \\ \widehat{\varepsilon}(N) \end{bmatrix}$$

Rearranging (28) in terms of the error vector $\widehat{E}$, we get:

$$\widehat{E} = Y - \Phi\widehat{\Theta} \tag{29}$$

Definition: Least squares principle

The unknown parameters of a model should be chosen in such a way that "The sum of squares of the differences between the actually observed data Y and computed values of data $\widehat{Y} = \Phi\widehat{\Theta}$ multiplied by numbers (weighting factors) that measure the degree of precision (or trust) is minimal.

The sum of squares of errors with unit weights (LS criterion) can be defined as:

$$J(\widehat{\Theta}) = \frac{1}{2}\sum_{i=1}^{N}\widehat{\varepsilon}_i^2(k) = \frac{1}{2}\widehat{E}^T\widehat{E} = \frac{1}{2}\left\|\widehat{E}\right\|^2 \tag{30}$$

$\widehat{\Theta}_{LS}$ is defined as the least squares estimator of Θ, given data, if it minimizes $J(\widehat{\Theta})$. The weighted sum of squares of errors (WLS criterion) is:

$$J(\widehat{\Theta}) = \frac{1}{2}\sum_{i=1}^{N}q_i\widehat{\varepsilon}_i^2(k) = \frac{1}{2}\widehat{E}^TQ\widehat{E} \tag{31}$$

The LS criterion (30) is then[5]:

$$J(\widehat{\Theta}) = \frac{1}{2}\widehat{E}^T\widehat{E} = (Y - \Phi\widehat{\Theta})^T(Y - \Phi\widehat{\Theta})$$

$$J(\widehat{\Theta}) = Y^TY - Y^T\Phi\widehat{\Theta} - \widehat{\Theta}^T\Phi^TY + \widehat{\Theta}^T\Phi^T\Phi\widehat{\Theta} \tag{32}$$

Setting to zero the derivative of $J(\widehat{\Theta})$ with respect to $\widehat{\Theta}$ for a stationary point (Jacobian matrix) we end up with:

$$\frac{\partial J(\widehat{\Theta})}{\partial\widehat{\Theta}} = \Phi^T\Phi\widehat{\Theta} - \Phi^TY = 0 \tag{33}$$

These normal equations can be solved for a unique minimum if the Hessian of $J(\widehat{\Theta})$ is positive semidefinite i.e.:

$$\left|\frac{\partial^2 J(\widehat{\Theta})}{\partial\widehat{\Theta}^2}\right| \geq 0$$

This Hessian $\partial^2 J/\partial\widehat{\Theta}^2 = \Phi^T\Phi$ will be positive semidefinite if and only if Φ has full rank.
The least squares parameter estimation follows from (33) as:

$$\widehat{\Theta}_{LS} = \left(\Phi^T\Phi\right)^{-1}\Phi^TY = \Phi^{\dagger}Y \tag{34}$$

where:

[5]We will use unit weights for the sake of simplicity.

$\Phi^\dagger$ - pseudoinverse of Φ.

The pseudoinverse of Φ exist if $\Phi^T\Phi$ is nonsingular and this will be satisfied when Φ is full rank i.e. only when test signals are persistently exciting[6]. The resulting fitting error $\widehat{E}$ can be denoted by:

$$\widehat{E} = R^T = [\eta(1)\ \eta(2)\ \cdots\ \eta(N)]$$

where the components of R are called **residuals**.

Premultiplying equation (28) by Φ^T, yields:

$$\Phi^T Y = \Phi^T\Phi\widehat{\Theta}_{LS} + \Phi^T R \tag{35}$$

also (34) is:

$$\widehat{\Theta}_{LS} = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y \tag{36}$$

These two equations imply that:

$$\Phi^T R = 0 \tag{37}$$

Because we have by definition (29) that $R = (Y - \Phi\widehat{\Theta}_{LS})$ so the residual indicates that $\widehat{\Theta}_{LS}$ must be chosen such that R is orthogonal to the columns of the Φ matrix. Writing (37) out in full, gives:

$$\left[\ \varphi(1)\ \varphi(2)\ \ \cdots\ \ \varphi(N)\ \right] R = 0 \tag{38}$$

Recalling the definition of $\varphi(k)$, this can be rewritten as:

$$\sum_{k=1}^{N} y(k-i)\eta(k) = 0 \quad \text{for } i = 1, 2, \ldots na \tag{39}$$

and

$$\sum_{k=1}^{N} u(k-i)\eta(k) = 0 \quad \text{for } i = 1, 2, \ldots nb+1 \tag{40}$$

and so on, for any other entries in $\varphi(k)$, if present (like: $v(k), k, e(k)$ etc.). As N becomes large, and under the ergodic assumption[7] (i.e. weakly stationary[8] stochastic processes) these equations imply that:

$$E\left\{y(k-i)\eta(k)\right\} = 0 \quad \text{for } i = 1, 2, \ldots na \tag{41}$$

and

$$E\left\{u(k-i)\eta(k)\right\} = 0 \quad \text{for } i = 1, 2, \ldots nb+1 \tag{42}$$

Equations (39), (40), (41) and (42) are sometimes said to express *the orthogonality property* of LS, and are the basis of a number of other results, like the self-tuning property discussed later.

[6]So called "excitation condition".

[7]Stochastic processes for which expectations can be replaced by time averages over a single realization are said to be *ergodic*.

[8]Mathematical expectation is constant, and the autocovariance expressed as $r_{xx}(t_1 - t_2)$.

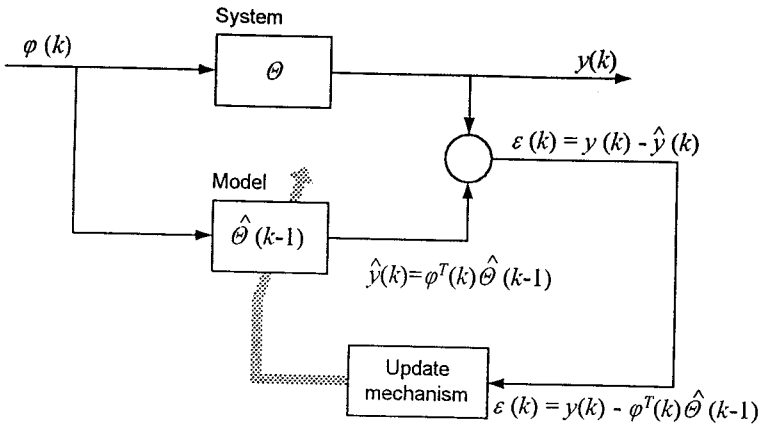


Fig. 9. Block diagram of the recursive least squares method.

For the $\widehat{\Theta}_{LS}(k)$ using data from time 1 to k, we have:

$$\widehat{\Theta}_{LS}(k) = \left[\Phi^T(k)\Phi(k)\right]^{-1}\Phi^T(k)Y(k) \tag{45}$$

where:

$$Y(k) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \end{bmatrix}; \Phi(k) = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(k) \end{bmatrix}$$

At time $k+1$ we obtain new measurements from the process which enable us to form:

$$Y(k+1) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} Y(k) \\ y(k+1) \end{bmatrix} \tag{46}$$

$$\Phi(k+1) = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(k) \\ \varphi(k+1) \end{bmatrix} = \begin{bmatrix} \Phi(k) \\ \varphi(k+1) \end{bmatrix} \tag{47}$$

The estimates at $k+1$ are then given by:

$$\widehat{\Theta}_{LS}(k+1) = \left[\Phi^T(k+1)\Phi(k+1)\right]^{-1}\Phi^T(k+1)Y(k+1) \tag{48}$$

Now:

$$\begin{aligned}\Phi^T(k+1)\Phi(k+1) &= \begin{bmatrix} \Phi^T(k) & \varphi(k+1) \end{bmatrix}\begin{bmatrix} \Phi(k) \\ \varphi^T(k+1) \end{bmatrix} \\ &= \Phi^T(k)\Phi(k) + \varphi(k+1)\varphi^T(k+1)\end{aligned} \tag{49}$$

A.1 Properties of the least squares estimator

$\widehat{\Theta}_{LS}$ is a random variable whose properties can be analyzed using the equation:

$$y(k) = \varphi^T(k)\widehat{\Theta} + \widehat{\varepsilon}(k) \tag{43}$$

which defines the actual system with disturbances. Two properties are important in this respect:

1. bias - refers to the systematic error which can occur in the $\widehat{\Theta}$,
2. covariance - is related to the spread of estimates arising from random errors.

Ad1) - bias: stacking equation (43) for $k = 1, 2, \ldots N$ we get: $Y = \Phi\widehat{\Theta} + \widehat{E}$, and substituting in equation $\widehat{\Theta}_{LS} = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y$, yields:

$$\widehat{\Theta}_{LS} = \left(\Phi^T\Phi\right)^{-1}\left[\Phi^T\Phi\Theta + \Phi^T\widehat{E}\right] = \Theta + \left(\Phi^T\Phi\right)^{-1}\Phi^T\widehat{E} \tag{44}$$

The average deviation of the estimate of the parameter from its true value (bias in the estimator) is given by a rearrangement of (44):

$$\widehat{\Theta}_{LS} - \Theta = \left(\Phi^T\Phi\right)^{-1}\Phi^T\widehat{E}$$

When the data which makes up Φ is deterministic, the expected values of $\widehat{\Theta}_{LS} - \Theta$ can be written as:

$$E\left\{\widehat{\Theta}_{LS} - \Theta\right\} = \left(\Phi^T\Phi\right)^{-1}\Phi^T E_E\left\{\widehat{E}\right\} = 0 \quad \text{iff } \widehat{E} \text{ is zero mean}$$

When the elements of Φ are random, but independent of $\widehat{E}$, then:

$$E\left\{\widehat{\Theta}_{LS} - \Theta\right\} = E_\Phi\left\{\left(\Phi^T\Phi\right)^{-1}\Phi^T\right\}E_E\left\{\widehat{E}\right\} = 0 \quad \text{iff } \widehat{E} \text{ is zero mean}$$

here subscripts $_\Phi$ and $_E$ denote expectations over the Φ and $\widehat{E}$ objects respectively.

When the data and the noise are correlated the $p\lim$ concept[9] can be used to show that the estimates are unbiased if $p\lim\limits_{N\to\infty}(N^{-1}\Phi^T\widehat{E}) = 0$. The entries in the vector $\Phi^T\widehat{E}$ are essentially the cross correlations between the data and noise sequences. We can conclude that asymptotic uncorrelatedness of the data and the noise is necessary for unbiased estimates in the probability limit sense.

B. Recursive least squares algorithms

Instead of recalculating $\widehat{\Theta}_{LS}$ in its entirety, requiring the storage of all previous data, it is efficient to merely store the "old" estimate calculated at time k, denoted by $\widehat{\Theta}_{LS}(k)$, and to obtain the "new" estimate $\widehat{\Theta}_{LS}(k+1)$ by an updating step involving the new observation only. The recursive estimation process can then be visualized as in Fig.9

In the block diagram of the RLS method, new input/output data $\varphi(k)$ become available at each sample interval. The model based on past information (summarized in $\widehat{\Theta}_{LS}(k-1)$) is used to obtain an estimate $\widehat{y}(k)$ of the current output. This is then compared with the observed output $y(k)$ to generate an error $\varepsilon(k)$. This error is used in an update mechanism to correct $\widehat{\Theta}_{LS}(k-1)$ to the new value $\widehat{\Theta}_{LS}(k)$. This recursive "predictor-corrector" form allows significant saving in computation. To see how it is done, compare a LS estimate based on data from time samples 1 to k, with the estimate based on data from time samples 1 to $k+1$.

[9] $p\lim x(k) = X$ i.e. $P\{|x(k) - X| > \epsilon\} \to 0$

Thus, given $\varphi(k+1)$ we can easily update the old matrix of correlations $\Phi^T(k)\Phi(k)$ to obtain the new matrix $\Phi^T(k+1)\Phi(k+1)$. However, we actually need to find a way to update the inverse of $\Phi^T(k)\Phi(k)$ directly without requiring a matrix inversion at each time step. In addition, we also need to update the term $\Phi^T(k+1)Y(k+1)$. From (46) and (47) we get:

$$\begin{aligned}\Phi^T(k+1)Y(k+1) &= \begin{bmatrix} \Phi^T(k) & \varphi(k+1) \end{bmatrix} \begin{bmatrix} Y(k) \\ y(k+1) \end{bmatrix} \\ &= \Phi^T(k)Y(k) + \varphi(k+1)y(k+1) \end{aligned} \tag{50}$$

Introduce: $P(k) = \left[\Phi^T(k)\Phi(k)\right]^{-1}$ and $B(k) = \Phi^T(k)Y(k)$. The equations (45) and (48) becomes now:

$$\widehat{\Theta}_{LS}(k+1) = P(k+1)B(k+1) \tag{51}$$

$$\widehat{\Theta}_{LS}(k) = P(k)B(k) \tag{52}$$

also for $\Phi^T(k+1)\Phi(k+1) = \Phi^T(k)\Phi(k) + \varphi(k+1)\varphi^T(k+1)$ follows:

$$P^{-1}(k+1) = P^{-1}(k) + \varphi(k+1)\varphi^T(k+1) \tag{53}$$

and for $\Phi^T(k+1)Y(k+1) = \Phi^T(k)Y(k) + \varphi(k+1)y(k+1)$ follows:

$$B(k+1) = B(k) + \varphi(k+1)y(k+1) \tag{54}$$

Equation (54) gives a direct update from $B(k)$ to $B(k+1)$. We seek the same direct update from $P(k)$ to $P(k+1)$. To do this we should use the matrix inversion lemma.

Matrix inversion lemma

Let $A = P^{-1}(k)$, $B = \varphi(k+1)$, $C = 1$ and $D = \varphi^T(k+1)$ be matrices of compatible dimensions, so that the product $BCD = \varphi(k+1)\varphi^T(k+1)$ and the sum $A + BCD = P^{-1}(k) + \varphi(k+1)\varphi^T(k+1)$ exist. Then:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B\left[C^{-1} + DA^{-1}B\right]^{-1}DA^{-1}$$

or:

$$\left[P^{-1}(k) + \varphi(k+1)\varphi^T(k+1)\right]^{-1} = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1 + \varphi^T(k+1)P(k)\varphi(k+1)} \tag{55}$$

The matrix inversion lemma solved the problem of matrix inversion in each step, by transforming it to the division by a scalar in each step. So, the equation (55) now becomes:

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1 + \varphi^T(k+1)P(k)\varphi(k+1)} \tag{56}$$

where:

$P(k)$ - positive definite matrix.

For linear regression models and normal (Gaussian) distribution of the disturbance, $P(k)$ is the covariance matrix of the aposteriori $\Theta(k)$ with the expectation $\widehat{\Theta}(k)$.

If we substitute $y(k+1)$ from the relation $\varepsilon(k) = y(k+1) - \varphi^T(k+1)\widehat{\Theta}_{LS}(k) = y(k+1) - \widehat{y}(k+1)$ into equation (54) we get:

$$B(k+1) = B(k) + \varphi(k+1)\varphi^T(k+1)\widehat{\Theta}_{LS}(k) + \varphi(k+1)\varepsilon(k+1) \tag{57}$$

from (51):

$$P^{-1}(k+1)\widehat{\Theta}_{LS}(k+1) = \Big[\underbrace{P^{-1}(k)+\varphi(k+1)\varphi^T(k+1)}_{P^{-1}(k+1)}\Big]\widehat{\Theta}_{LS}(k)+\varphi(k+1)\varepsilon(k+1) \tag{58}$$

after multiplying the above expression from the left with $P(k+1)$ we get:

$$\widehat{\Theta}_{LS}(k+1) = \widehat{\Theta}_{LS}(k) + \underbrace{P(k+1)\varphi(k+1)}_{L(k+1)}\varepsilon(k+1) \tag{59}$$

$L(k+1)$ is a column vector of adjustment gains.
From (56) and (59) we can get the Recursive Least Squares (RLS) algorithm - (RLS-1):

Algorithm: **RLS-1**

Initialize algorithm with: $P(0)$, $\varphi(0)$ *and* $\widehat{\Theta}_{LS}(0)$,
At time step $k+1$ *go through the following steps:*
Step 1: Form $\varphi(k+1)$ *using new measured data,*
Step 2: Form $\varepsilon(k+1) = y(k+1) - \varphi^T(k+1)\widehat{\Theta}_{LS}(k)$,
Step 3: Form $P(k+1)$ *as:*

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k+1)}$$

Step 4: Update $\widehat{\Theta}_{LS}(k)$ *as:*

$$\widehat{\Theta}_{LS}(k+1) = \widehat{\Theta}_{LS}(k) + P(k+1)\varphi(k+1)\varepsilon(k+1)$$

Step 5: Wait for the next time step to elapse and loop back to step (1).

In short the RLS-1 can be given by:

RLS-1 Algorithm:

$$\widehat{\Theta}_{LS}(k+1) = \widehat{\Theta}_{LS}(k) + P(k+1)\varphi(k+1)\varepsilon(k+1) \tag{60}$$

$$\varepsilon(k+1) = y(k+1) - \varphi^T(k+1)\widehat{\Theta}_{LS}(k) \tag{61}$$

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k+1)} \tag{62}$$

with the regression vector[10]:
$\varphi^T(k) = [-y_{k-1} \cdots -y_{k-na}\ u_{k-1} \cdots u_{k-nb-1}]$
and parameter vector:
$\widehat{\Theta}_{LS} = \left[\widehat{a}_1\ \widehat{a}_2\ \cdots\ \widehat{a}_{na}\ \widehat{b}_0\ \widehat{b}_1\ \cdots \widehat{b}_{nb}\right]$

Remark 1: RLS algorithm estimates only A and B polynomial coefficients.

Remark 2: Do not use RLS if the disturbance to the process is colored.

[10]For the ARMAX model with $C(q^{-1}) = 1$.

Remark 3: If we consider the term $C(q^{-1})\widehat{e}(k)$ as simply "the error", and estimate A and B polynomial coefficients using RLS estimation, the estimates will be **biased.**

The form of the RLS algorithm given above is general enough and many recursive algorithms use this basic form. The differences between them are in the way that they form the covariance matrix $P(k)$, regression vector $\varphi(k)$ and the prediction error $\varepsilon(k)$.

Example 7:
Consider the process described by, [4]:

$$y(k) = ay(k-1) + \xi(k); \qquad |a| < 1 \tag{63}$$

where:

$$\xi(k) = e(k) + ce(k-1); \qquad |c| < 1 \tag{64}$$

with: $E\{e(k)\} = 0$ *and* $var\{e^2\} = \sigma_e^2$ *white noise sequence. The off-line estimates of parameter* a *is:*

$$\widehat{a}(N) = \frac{\sum_{k=1}^{N} y(k)y(k-1)}{\sum_{k=1}^{N} y^2(k-1)} = a + \frac{\sum_{k=1}^{N} \xi(k)y(k-1)}{\sum_{k=1}^{N} y^2(k-1)}$$

For large N*, assuming ergodicity we will have:*

$$\widehat{a}(N) \approx a + \frac{E\{\xi(k)y(k-1)\}}{E\{y^2(k-1)\}}$$

For the white noise: $E\{\xi(k-1)\xi(k-i)\} = 0;\ \ i = 3, 4, 5, \ldots$. *Using (63) and (64) we will have:*

$$y(k-1) = \xi(k-1) + a\xi(k-2) + a^2\xi(k-3) + \cdots$$

leading to:

$$E\{\xi(k)y(k-1)\} = E\{\xi(k)\xi(k-1)\} = c\sigma_e^2$$

and:

$$\begin{aligned} E\{y^2(k-1)\} &= (1 + a^2 + a^4 + \cdots)E\{\xi^2(k-1)\} + 2a(1 + a^2 + a^4 + \cdots)E\{\xi(k-1)\xi(k-2)\} \\ &= \frac{1+c^2}{1-a^2}\sigma_e^2 + \frac{2ac}{1-a^2}\sigma_e^2 \end{aligned}$$

Finally we have:

$$\widehat{a}(N) - a \approx \frac{c(1-a^2)}{1 + c^2 + 2ac}$$

This result shows that a bias is present unless c *is zero i.e.* $C(q^{-1}) = 1$.

B.1 Residuals and prediction errors

A key variable in the RLS algorithm is the modelling (fitting) error $\varepsilon(k)$. We can now relate this error to the residual associated with the least squares procedure. The modelling (fitting) error is defined as:

$$\varepsilon(k) = y(k) - \varphi^T(k)\widehat{\Theta}_{LS}(k-1) = y(k) - \widehat{y}(k \mid k-1)$$

In words: modelling error at time k is the difference between the system output $y(k)$ at time k and the *predicted output* $\widehat{y}(k \mid k-1)$ at time k, based on known $\widehat{\Theta}_{LS}(k-1)$ at previous time $k-1$. This error is called *a priori output prediction error* because of that.

The true modelling error (or *residual* or *a posteriori prediction error)* at time k is:

$$\eta(k) = y(k) - \varphi^T(k)\widehat{\Theta}_{LS}(k) = y(k) - \widehat{y}(k \mid k)$$

$\varepsilon(k)$ and $\eta(k)$ differ only in that $\eta(k)$ is based upon parameter estimates $\widehat{\Theta}_{LS}(k)$ at the current time step k, while $\varepsilon(k)$ is based upon $\widehat{\Theta}_{LS}(k-1)$, i.e. previous time step parameter estimates. As $k \to \infty$ the difference between a priori and a posteriori prediction errors approaches zero, but during the first few recursions the difference is significant, and various estimation algorithms exploit this. The relation between $\varepsilon(k)$ and $\eta(k)$ is given by:

$$\eta(k) = \varepsilon(k)\frac{1}{1 + \varphi^T(k)P(k-1)\varphi(k)}$$

As seen from the above expression the denominator is already calculated in the RLS algorithm, so a posteriori prediction (residual) can be obtained from $\varepsilon(k)$ at any time step by a scalar division.

In order to estimate C polynomial coefficients, knowledge of $\widehat{e}(k-1), \widehat{e}(k-2), \cdots \widehat{e}(k-nc)$ is required. However, $\{\widehat{e}(k)\}$ is an unobservable error process. A number of estimation procedures exist, which replace $\widehat{e}(k)$ by an estimate usually taken to be:

- a posteriori prediction error (residual) $\eta(k)$ - then we have approximate maximum likelihood (AML) estimation,
- a priori output prediction error $\varepsilon(k)$ - then we have recursive extended least squares (RELS) estimation.

The recursive extended least squares algorithm - RELS:

Algorithm: **RELS**

$$\widehat{\Theta}_{LS}(k+1) = \widehat{\Theta}_{LS}(k) + P(k+1)\varphi(k+1)\varepsilon(k+1) \tag{65}$$

$$\varepsilon(k+1) = y(k+1) - \varphi^T(k+1)\widehat{\Theta}_{LS}(k) \tag{66}$$

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1 + \varphi^T(k+1)P(k)\varphi(k+1)} \tag{67}$$

with the regression vector:

$$\varphi^T(k) = [-y_{k-1} \cdots - y_{k-na} u_{k-1} \cdots u_{k-nb-1} \varepsilon_{k-1} \varepsilon_{k-2} \cdots \varepsilon_{k-nc}]$$

and parameter vector:

$$\widehat{\Theta}_{LS} = \left[\widehat{a}_1\ \widehat{a}_2\ \cdots\ \widehat{a}_{na}\ \widehat{b}_0\ \widehat{b}_1\ \cdots \widehat{b}_{nb}\ \widehat{c}_1\ \widehat{c}_2\ \cdots\ \widehat{c}_{nc}\right]$$

C. Exploring recursive estimators

In practical situations there are issues which should be addressed, such as:

1. Initialization of the estimator,
2. Model selection - specification of the estimator,
3. Choice of the operating conditions,
4. Manipulation of the covariance matrix P.

C.1 Initializing the estimator

The issues which arise here are:

- the choices of initial data (regression) vector, $\varphi(0)$
- the choice of initial estimate, $\widehat{\Theta}_{LS}(0)$
- the choice of initial covariance matrix, $P(0)$

C.2 Initializing data (regression) vector - $\varphi(0)$

The data (regression) vector is initialized by collecting data before the recursive estimator is started. The number of necessary time steps ($\varkappa$) needed for collecting data depends on the model:

$$\varkappa = \max\{na, nb+1, nd+1, nf+1, nc\}$$

Example 8:
Suppose the process model is given by: $y(k) = \varphi^T(k)\widehat{\theta}$
where:
$\widehat{\Theta}^T = \begin{bmatrix} \widehat{a}_1 & \widehat{b}_0 & \widehat{b}_1 & \widehat{b}_2 \end{bmatrix}$
$\varphi^T(k) = \begin{bmatrix} -y(k-1) & u(k-1) & u(k-2) & u(k-3) \end{bmatrix}$
The regression vector requires three time steps in order to collect sufficient past values of $u(k)$. *At the fourth sample interval, the recursive estimator may be started.*

C.3 Initializing parameter estimates vector - $\widehat{\Theta}_{LS}(0)$

The initial estimate is not crucial for convergence behavior. However, it is much better if the user already knows some parameters of his process. If not then a useful technique is to assume that the system is a single integrator with unit gain. The step invariant (ZOH) sampling of the integrator gives: $Z\left\{\frac{1}{s}\right\} = \frac{h}{z-1}$, so the initial parameters are:

$$a_1 = -1 \text{ and } a_i = 0 \text{ for } i \neq 1 \text{ and also } b_0 = h \text{ and } b_i = 0 \text{ for } i \neq 0$$

Once, one recursive estimation run of a particular system has been completed, it is a standard practice to use the final estimates to initialize subsequent runs.

C.4 Initializing covariance matrix - $P(0)$

Initial values of the covariance matrix depends on our uncertainty concerning the unknown parameters. If we have no prior knowledge of the system parameters, then a large initial covariance would reflect this. Often, the parameters associated with A, B, C etc. polynomials have different initial uncertainties associated with them. In such circumstances $P(0)$ is chosen diagonal, but with each diagonal entry reflecting the uncertainty associated with the corresponding parameter. In this context it is not uncommon to give different weights to the initial coefficient estimates of the A polynomial, B polynomial and so on. However, a standard choice is:

$$P(0) = rI_{npar}$$

where r is a scalar.

"Large" $P(0)$ means $100 < r < 1000$ while "small" $P(0)$ means $1 < r < 10$. Terms "large" and "small" depends upon the experiment, experience, and the particular application. In the self-tuning context, we should be aware of the following facts:

1. "Large" $P(0)$ will result in a more responsive identification algorithm, which will as a result have higher fluctuations of the control signal and as a consequence more strain or wear of the actuator.
2. "Small" $P(0)$ will result in sluggish identification algorithm, smaller control signal fluctuations, and less actuator activity.

Example 9:

Consider the process described by: $y(k) = \varphi^T(k)\widehat{\theta}$

where:

$\widehat{\Theta}^T = \begin{bmatrix} \widehat{a}_1 & \widehat{a}_2 & \widehat{b}_0 & \widehat{b}_1 \end{bmatrix}$

If $\widehat{b}_0$ *and* $\widehat{b}_1$ *of the B polynomial are poorly known, reflecting poor knowledge of the process gain and zeros, while prior transient experiments established the approximate values of the process poles (and hence* $\widehat{a}_1$ *and* $\widehat{a}_2$*), then a typical initial* $P(0)$ *in this case would be:*

$$P(0) = \begin{bmatrix} r_{a1} & 0 & 0 & 0 \\ 0 & r_{a2} & 0 & 0 \\ 0 & 0 & r_{b0} & 0 \\ 0 & 0 & 0 & r_{b1} \end{bmatrix}$$

with $r_{a1} = r_{a2} \approx 100$ *and* $r_{b0} = r_{b1} \approx 1$

To visualize how the initial covariance matrix $P(0)$ influences subsequent covariance values, we should recall that the covariance matrix at time k can be written as:

$$P(k) = \left[P^{-1}(0) + \sum_{i=1}^{k} \varphi(i)\varphi^T(i) \right]^{-1}$$

When $P(0)$ is "large" its influence upon $P(k)$ is much less than the information collected in $\{\varphi(i); i = 1, 2, \ldots k\}$. Conversely, if $P(0)$ is "small", its influence upon $P(k)$ is correspondingly greater.

C.5 Performance of a recursive estimator

The initial performance of the recursive estimator depends on:

- choice of $\varphi(0)$, $\widehat{\Theta}_{LS}(0)$, and $P(0)$
- type of the system
- form of the excitation signals applied

The next example will show how recursive algorithms behave. Some pathological cases will be shown and the way how to treat them given.

Example 10:

The mathematical model of the process is:

$$y(k) + ay(k-1) = bu(k-1) + e(k) + ce(k-1)$$

where:

$a = -0.8$

$b(k) = 0.5 \quad \forall k > 0$

$c = 0$

$e(k)$ is the white noise with $E\{e(k)\} = 0$ and $E\{e^2(k)\} = 0.25$
The estimator is initialized with:

$$P(0) = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \; ; \; \widehat{\Theta}(0) = \begin{bmatrix} \widehat{a}(0) \\ \widehat{b}(0) \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

$$\varphi(k-1) = \begin{bmatrix} -y(k-1) & u(k-1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \text{ for } k = 0$$

The excitation signal is unit amplitude square wave signal with the period $T_{SW} = 100$ [s].
The estimated parameters (full line) and the true parameters (dotted line) are given in the Fig.10.

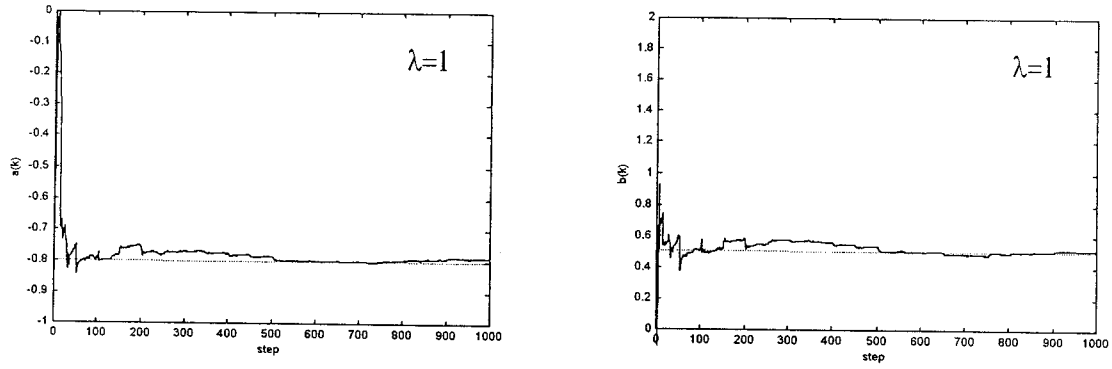


Fig. 10. Parameter identification by RLS for the time invariant process.

Now if the process becomes time variant i.e. we have:

$$b(k) = \left\{ \begin{array}{l} 0.5 \text{ for } 0 < k < 400 \\ 1 \text{ for } 400 \leq k < \infty \end{array} \right\}$$

RLS will give the estimations as given in Fig.11.

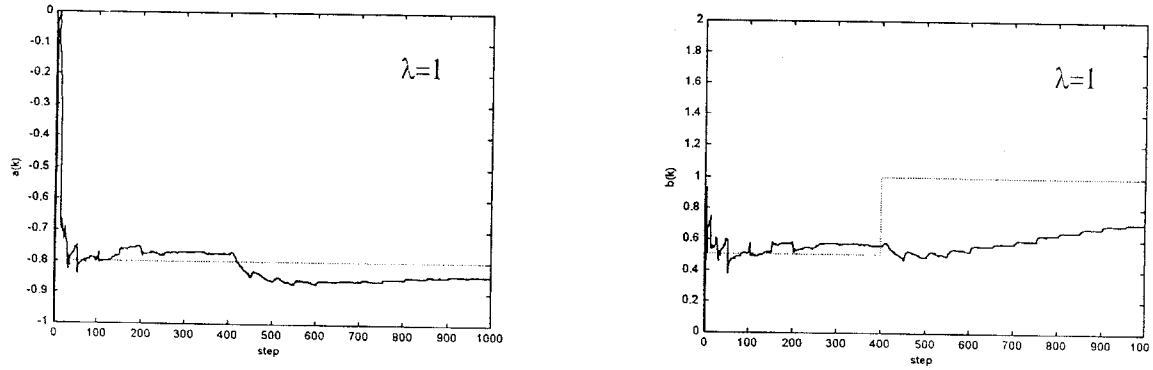


Fig. 11. Parameter estimation by RLS for the time variant process.

As seen on Fig.11 the RLS algorithm was not capable of estimating the time variant process parameters. The reason for this is that during the operation of the RLS estimator the trace of the covariance matrix approached zero ($\lim_{k\to\infty} tr\,[P(k)] \to 0$), and we say that $P(k)$ have turned off , see Fig.12 (for $\lambda = 1$).

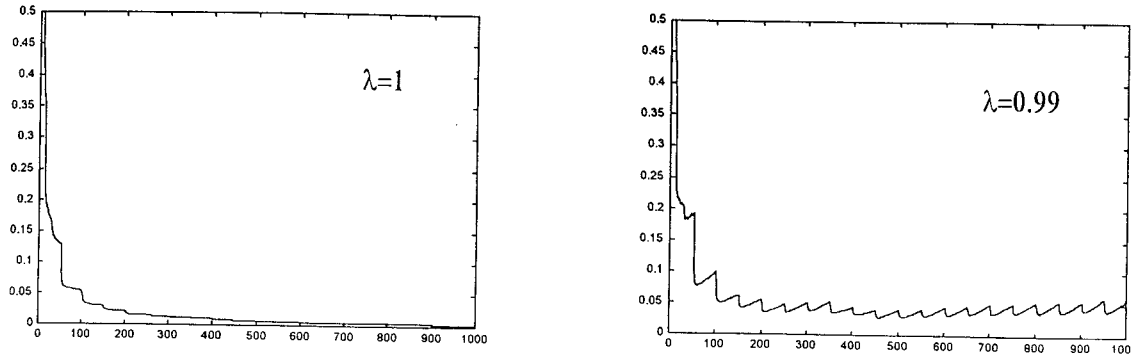


Fig. 12. The trace of covariance matrix for various exponential forgetting factors.

When this happens the RLS algorithm calculates the new estimation based on the previous one, (see (59)), the new measurement is not taken into account, i.e. the algorithm turned off, and the capability of estimating the changing parameters is lost.

In adaptive self-tuning control where the process is not time invariant, this can lead to disastrous results because the regulator parameters will use wrong parameter estimations in the design procedure.

So, something has to be done to improve the alertness of the RLS algorithm. In the literature there are many recommended procedures for this. We will mention the following:

- Exponential data weighting, [28].
- Automatic change of the exponential forgetting factor, [29], [30].
- Constraining the trace of the covariance matrix P, by appropriate change of the forgetting factor, [31], [28].
- Covariance resetting, [32].
- Covariance matrix modification, [33].

These interventions can improve the estimation algorithm.

C.6 Exponential data weighting

Minimization of the weighted least squares criterion (31) for the time-invariant process will give the estimation of the mean parameters inside the period $0 \leq i \leq N$. To obtain the estimation corresponding to estimation in a particular moment k, the criterion should be changed so that new measured data are taken into account with higher impact, while the older ones are taken into account with less impact. This criterion is given by, [25]:

$$J(\widehat{\Theta}) = \frac{1}{2}\sum_{i=1}^{N} \lambda^{k-1} q_i \widehat{\varepsilon}_i^2(k) \tag{68}$$

The choice of the $\lambda(k)$ is dictated by fast adaptation and good estimation. When $\lambda \approx 1$ we will have that the prediction error older than T_m time units will have the impact factor $e^{-1} \approx 36\%$ compared with newest measured data[11]. So, T_m can be called the time constant of the criterion memory (estimator).

The RLS covariance update becomes then:

$$P(k+1) = \left[I - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{\lambda(k+1) + \varphi^T(k+1)P(k)\varphi(k+1)}\right] \frac{P(k)}{\lambda(k+1)} \tag{69}$$

where:

[11] $T_m = \frac{1}{1-\lambda}$

$\lambda(k)$ - exponential forgetting (data weighting) factor, $0 < \lambda < 1$

Experience has shown that the best is to try $0.95 < \lambda < 0.99$ with $\lambda = 0.98$ as a recommended choice, see Fig.13.

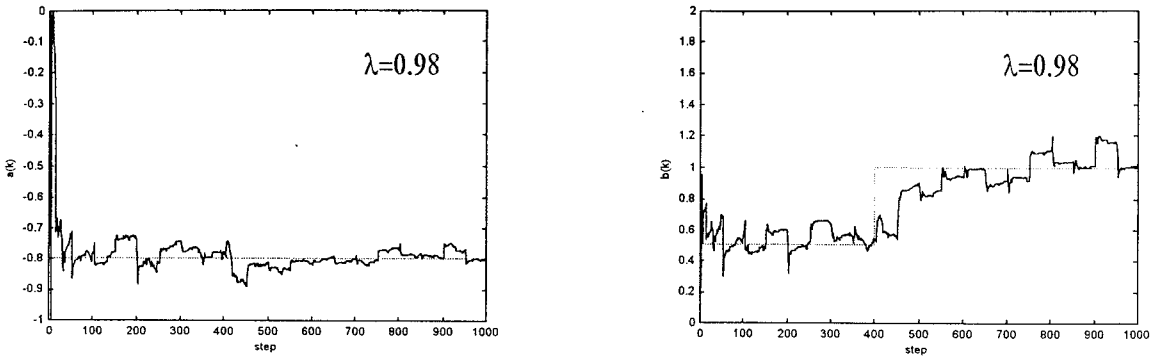


Fig. 13. Exponentially weighted RLS in identification of time variant process parameters.

Standard RLS when $\lambda(k) = 1\ \forall k$ is not capable of estimating parameters that change during the estimation process, while exponentially weighted one is capable to estimate them. The trace of the covariance matrix is shown in Fig.14 .

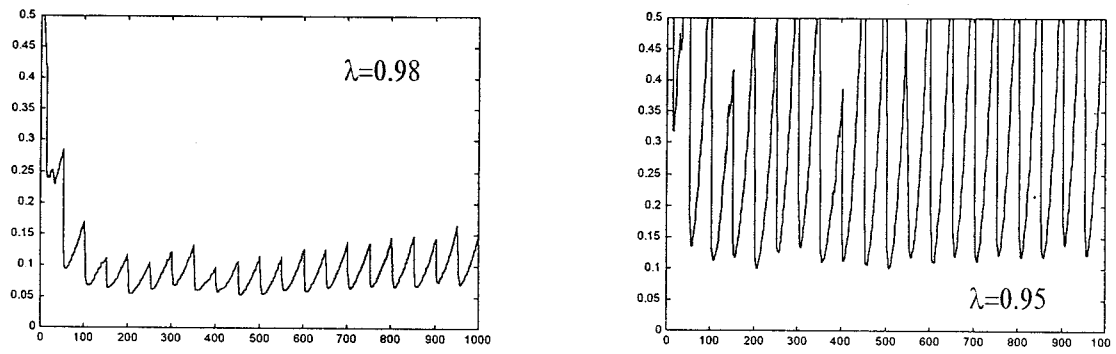


Fig. 14. Trace of the covariance matrix for two exponential forgetting factors.

As is seen on Fig.14 smaller λ have as a consequence alert RLS algorithm, but in the adaptive control this also means that the estimation will oscillate, causing changes of the regulator parameters which will result in wear and tear of the actuator. So, some balance has to be found, and as is already said $\lambda = 0.98$ is recommended for that purpose. Exponentially weighted RLS in adaptive control will have $0.95 < \lambda < 0.99$ and the exponential forgetting factor will be useful only if persistency of excitation exists. In case that there is no persistency of excitation, estimator windup can result (for $\lambda < 1$), see (69) when only the term: $P(k+1) = \frac{P(k)}{\lambda(k+1)}$ remains if no new information feeds the regression vector. Time-variant exponential forgetting factor can then be used, defined by, [29], [30]:

$$\lambda(k) = \lambda_0 \lambda(k-1) + (1 - \lambda_0) \tag{70}$$

where:

$\lambda_0 = 0.99$

$\lambda(0) = 0.95$

With this forgetting factor we have that λ changes from 0.95 to 0.99 and is not constant during estimation routine. So, with this forgetting factor we have at the beginning high alertness of the

algorithm, which decays with progress of the estimation. When the covariance matrix becomes large and the regression vector is dominated by random noise, then $\widehat{\Theta}(k)$ is unlikely to move in a direction that improves the input/output model used, and we have so called parameter drift. In the self-tuning control setting, if parameters drift into an unstable region, the result will be large perturbations or bursting in the input/output variables. This will as a consequence have rich excitation to the system and improved estimation, so this is a self-correcting capability of the self-tuning control. The optimal choice of forgetting factor must balance two contradictory performances:

1. Fast tracking of the changing parameters - higher susceptibility to noise,
2. Less oscillations of the estimated parameters - bigger estimator inertia.

Sometimes, especially when change of process parameters are fast, it is advantageous to quickly forget measured signals, and slower when the change is minor or when there are no persistency of excitation.

D. Covariance resetting

Covariance resetting is the technique for ensuring that the trace of the covariance matrix never becomes too small, i.e. the RLS algorithm is always kept alert. The covariance is every k_i samples reset to some value which is capable to keep alertness of the algorithm. For instance at k_i resetting occurs and the covariance restarts with a new value:

$$P(k_i) = \beta_i I_{npar}$$

where β_i is a scalar and usually $\beta_i < r$. The problem with covariance resetting is that in the adaptive control the actuator is always kept active, so wear and tear of the actuator can be expected. Also, the exponential convergence of the estimation toward true values of parameters is lost.

E. Numerical properties of the covariance matrix

Analysis of the equation (69) shows that P matrix is calculated by subtracting two positive definite matrices, and that the result must be a positive definite matrix. Due to the characteristic of the RLS algorithm to diminish the P matrix norm with each step, it can happen that the P matrix loses this characteristic of positive definiteness. The RLS algorithm breaks down and instability of the algorithm results. Due to that possibility, reliable numerical algorithms must be used for calculation of the P matrix. Two numerical algorithms are recommended:

- normalized Cholesky decomposition (square root Cholesky decomposition), [34]
- UD factorization by Bierman and Thorton, [35].

F. Conclusion

The estimator role is to estimate process parameters based on previously measured input and output process signals $u(k)$ and $y(k)$. That can be realized by various methods, [36], [25], [26]. However, in practice we will most often deal with:

- RLS - Recursive Least Squares,
- RELS - Recursive Extended Least Squares,
- RPE - Recursive Prediction Error,
- RML - Recursive Maximum Likelihood,
- RSA - Recursive Stochastic Approximation,
- RIV - Recursive Instrumental Variable,

In the majority of applications RLS, RELS or RPE methods are found. RELS or RPE identification methods are used whenever a disturbance acting on a process is not of white noise type, but is colored. The reason for such a popularity of these methods is probably in the fact that they are methods with the fastest convergence. Despite all, the practical experience gained with them confirmed that they are reliable, but also they have some disadvantages worth knowing. When the number of parameters is

high, then probably the RSA method is recommended. The reason is that stochastic gradient methods are relatively simpler when we need to refresh and save data for the covariance matrix. Problems related with the RLS parameter estimation, especially in the self-tuning context, were given and some interventions most often used discussed. The identification algorithm in the self-tuning adaptive control is very important because the design which is calculated on-line during normal operation of the system is based on the estimated parameters. So, if the estimator breaks down, the whole system is jeopardized, and some emergency plan has to be implemented in order to save the system from crash. Causes of defects are various, but most often the following should be mentioned:

- use of constant forgetting factor $\lambda < 1$,
- use of unstable numerical algorithm for calculating the covariance matrix P,
- use of non-persistently exciting signals,
- unmeasurable disturbances,
- unmodelled process dynamics.

Unmeasurable disturbances can cause divergence of the estimation $\widehat{\Theta}(k)$. This can happen only in situation when the reference signal is not persistently exciting, [37]. Namely, if in the RLS algorithm instead of the prediction error $\varepsilon(k)$ the following signal is used:

$$\nu(k) = z(k) - \widehat{y}(k \mid \widehat{\Theta}) \tag{71}$$

where:

$z(k) = y(k) + w(k)$

and if we assume that $w(k)$ is a harmonic disturbance, then $\varphi(k)$ as well as $\nu(k)$ will have the harmonic component in phase in both vectors, which will cause divergence of $\widehat{\Theta}(k)$. It is observed that a harmonic disturbance is most devastating for the estimation, while high passband disturbances are not.

Unmodelled dynamics are always present in any design using a mathematical model of a process. This means that the process is described by:

$$y(k) = q^{-d}\frac{B(q^{-1})}{A(q^{-1})}\left[1 + P(q^{-1})\right] \tag{72}$$

where $P(q^{-1})$ represents the unmodelled dynamics of the process, which is unknown but stable rational transfer function of known order. Presented algorithms assume that $P(q^{-1}) = 0$. With this assumption it is possible to get unstable control, because, for instance, the estimator trying to find the best possible models among the set of simplified models (models without $P(q^{-1})$) and unable to calculate acceptable model parameters gives large estimations which consequently can result during the design stage in unstable control algorithms. Classical interventions in the form of conditioning of signals, signal filtration, signal normalization, use of dead zone in the RLS algorithm etc. can be used here to circumvent this problem.

IV. Design of adaptive self-tuning controllers

Design of the adaptive self-tuning controllers is specific because only design methods capable to calculate controller parameters in real time can be used. Classical methods (such as frequency methods[12], root locus and others) are not useful here. The algebraic control theory developed mainly in late seventies [5], [9], [7], and [38] allow real time application of the design procedure. We will cover the pole/zero method which can be directly related with the MRAS approach of adaptive control.

A. Algebraic method of design

A.1 Pole/zero placement

The basic idea of the pole/zero placement method is very simple. It consists in defining the transfer function of the desired dynamics of closed-loop control system and equates it with the closed-loop transfer function of the real system, where controller parameters are unknown and can be related with the process parameters. For a known causal process transfer function given by:

$$G_p(q) = \frac{B(q)}{A(q)}$$

where $na > nb$ and coprime polynomials A and B, the controller parameters must be calculated so that the desired dynamics of the closed-loop system given by the model:

$$G_m(q) = \frac{B_m(q)}{A_m(q)} \tag{73}$$

is obtained. The control structure capable to solve this problem is given in Fig.15.

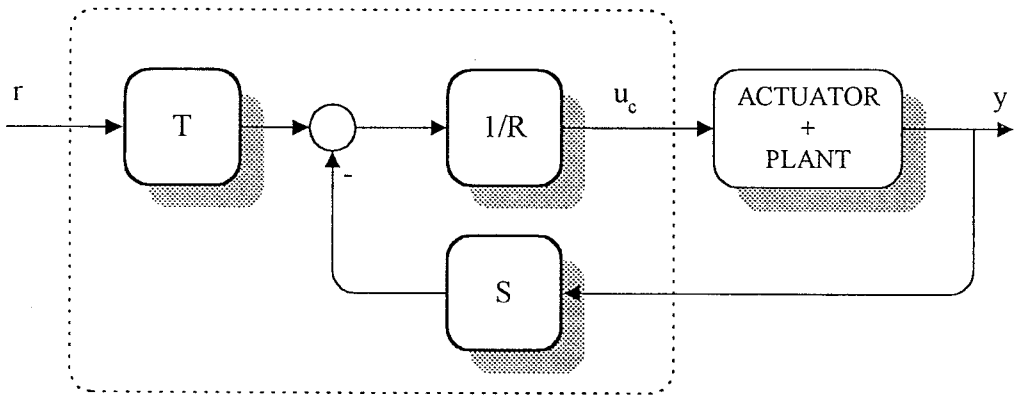


Fig. 15. Two parameter controller topology.

The desired model transfer function (73) is implementable if the following conditions are satisfied, [6]:

1. the controller must be causal,
2. the closed-loop control system is well posed (closed-loop causal),
3. the closed-loop control system is totally stable,
4. there is no plant leakage.

The system is well posed or closed-loop causal if every transfer function for all possible input/output combinations is causal. System is totally stable if every transfer function for all possible input/output combinations is stable. There is no plant leakage if there is no direct branch from the reference signal to the output signal, i.e. the transfer function of the closed-loop control system is strictly proper. The

[12]Nyquist or Bode.

first condition is necessary if we want to design a causal regulator. The second condition is necessary if we want to develop a system with small noise susceptibility, while the third condition is necessary if we want to exclude unstable pole-zero cancellations, i.e. deal with detectable and/or stabilizable system. The fourth condition is necessary if we want that the whole energy goes through the process and that there are no parallel branches where this energy can bypass the process.

From above conditions on implementable transfer function $G_m(q)$ the following theorem follows, [6]:

Theorem: **Implementable** $G_m(q)$

For the causal process given by $G_p(q) = B(q)/A(q)$ *the transfer function* $G_m(q) = B_m(q)/A_m(q)$ *will be implementable if and only if* $G_m(q)$ *is stable, and also:*

$$M(q) = \frac{G_m(q)}{G_p(q)} = \frac{N(q)}{D(q)} \tag{74}$$

is stable and causal.

Condition no.4 boils down to the fact that $M(q)$ represents the transfer function from the reference signal to the control signal. Condition no.3 has as a consequence that $G_m(q)$ and $M(q)$ must be stable, while condition no.2 results in $G_m(q)$ and $M(q)$ must be causal. The consequences of this theorem are:

$$G_m(q) = M(q)G_p(q)$$

$$\deg(A_m) - \deg(B_m) = \deg(A) - \deg(B) + \deg(D) - \deg(N)$$

If $M(q)$ is causal then: $\deg(D) \geq \deg(N)$ and consequently:

$$\deg(A_m) - \deg(B_m) \geq \deg(A) - \deg(B) \tag{75}$$

So, if (75) is valid then $M(q)$ is causal. Stability of $G_m(q)$ and $M(q)$ is satisfied when $A_m(q)$ and $D(q)$ polynomials are Hurwitz (roots inside unit circle). From (74) follows:

$$M(q) = \frac{N(q)}{D(q)} = \frac{G_m(q)}{G_p(q)} = \frac{B_m(q)A(q)}{A_m(q)B(q)} \tag{76}$$

From (76) it is obvious, that if the $B(q)$ polynomial has roots outside the unit circle, and if they are not cancelled with $B_m(q)$ then $D(q)$ can not be a Hurwitz (stable) polynomial and consequently $M(q)$ stable. So, in order to have stable $M(q)$ it is necessary that "unstable" roots of the polynomial $B(q)$ must be contained in the polynomial $B_m(q)$. In other words, "unstable" process zeros must be kept in the desired polynomial $B_m(q)$, i.e. the regulator should not cancel them.

Corollary: **Implementable** $G_m(q)$

If the process $G_p(q) = B(q)/A(q)$ is causal, then $G_m(q) = B_m(q)/A_m(q)$ will be implementable if and only if the following is valid:

1. $\deg(A_m) - \deg(B_m) \geq \deg(A) - \deg(B)$, meaning that the process delay must be smaller than the closed-loop control system delay,
2. All "unstable" process zeros must be contained in the set of zeros of the desired closed-loop dynamics,
3. polynomial $A_m(q)$ must be Hurwitz (roots inside unit circle).

As is seen from the above, "unstable" process zeros constrain the zeros of the desired implementable closed-loop dynamics, while unstable process poles do not have any such effect on the closed-loop poles. The reason for that is in the fact that process poles can be arbitrarily positioned anywhere by

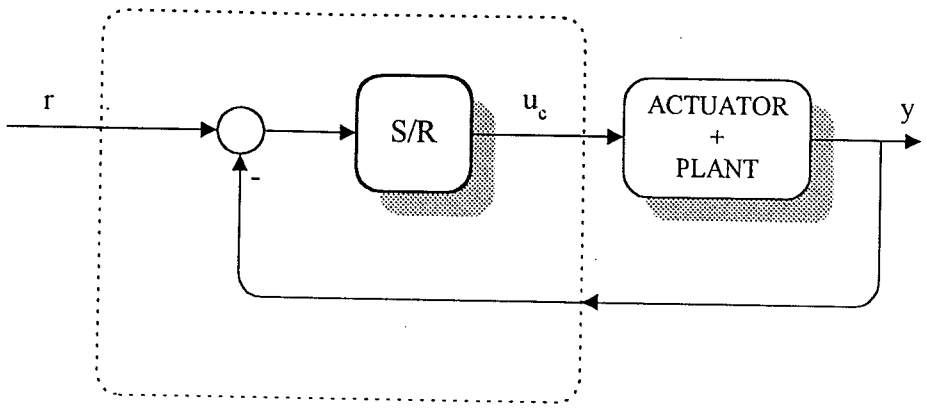


Fig. 16. Conventional (unit feedback) topology.

the use of feedback. However, the only way to move process zeros is (assuming no process leakage) by cancelation with poles or by feedforward compensation. Feedback does not play any role here.

The unit feedback (conventional) regulator topology (see Fig.16) where the regulator is behind the comparator allow positioning of the closed loop poles only, while zeros can not be moved. This is shown from the relation which equates the transfer function of the closed loop system with the desired (implementable) transfer function of the closed loop dynamics:

$$G_{cl}(q) = \frac{B(q)S(q)}{A(q)R(q) + B(q)S(q)} = \frac{B_m(q)}{A_m(q)} = G_m(q) \tag{77}$$

As is seen from equation (77) the closed poles can be positioned by solving the Diophantine equation:

$$A(q)R(q) + B(q)S(q) = A_m(q) \tag{78}$$

Equation (78) can be solved for polynomials $R(q)$ and $S(q)$ if we know polynomials $A(q), B(q)$ and $A_m(q)$. However, closed loop zeros can not be positioned at will because from (77) follows that closed loop zeros are defined by:

$$B(q)S(q) = B_m(q)$$

Due to the fact that the polynomial $S(q)$ is already obtained from (78) the closed loop zeros will consists of the open loop process zeros given by roots of polynomial $B(q)$ and controller zeros given by roots of polynomial $S(q)$. So, we do not have any possibility with conventional topology to position the closed loop zeros at the desired locations given by $B_m(q)$. However, the two-parameter controller topology (given in Fig.15) can solve the problem of pole and zero placement. For this topology (77) becomes:

$$G_{cl}(q) = \frac{B(q)T(q)}{A(q)R(q) + B(q)S(q)} = \frac{B_m(q)}{A_m(q)} = G_m(q) \tag{79}$$

As seen from (79) the Diophantine equation responsible for pole placement remains the same as before (78), while the equation for placement of zeros becomes:

$$B(q)T(q) = B_m(q) \tag{80}$$

So, we can conclude that two-parameter topology give us two degrees of freedom[13]:

[13]Contrary to conventional topology.

1. to position poles (by polynomials $R(q)$ and $S(q)$ - part of the controller acting in the feedback loop) and
2. to position zeros (by polynomial $T(q)$ - part of the controller acting in the feedforward loop).

Due to its capability and simplicity of solution this topology is most often found in the adaptive literature dealing with the self-tuning control.

As the equation $B(q)T(q) = B_m(q)$ shows the process zeros (roots of $B(q)$ polynomial) are contained in the set of closed loop zeros as well as the remaining zeros defined by the polynomial $T(q)$. The only way to get rid[14] of those process zero(s) is by cancelling them with some pole(s), while other closed loop zeros can be positioned at will with $T(q)$ polynomial.

If the polynomial B is factorized as: $B = B^- B^+$, where B^- represents polynomial with unstable or poorly damped roots, while B^+ represent monic[15] polynomial with stable or well damped roots, then it follows:

1. To get stable $G_m(q)$[16], then $B_m(q)$ must be factorized as:

$$B_m(q) = B^-(q)B'_m(q) \tag{81}$$

which implies that "unstable" process zeros can not be changed, but must be included in B_m.

2. The assumption was that polynomials A and B are coprime and that only stable or well damped zeros can be cancelled. Since B^+ is a factor of B, it follows that it must be also a factor of R to factorize closed loop polynomial $AR + BS$. So, we have:

$$R(q) = B^+(q)R'(q) \tag{82}$$

Finally from (79) we have:

$$\frac{B^- B^+ T}{B^+(AR' + B^- S)} = \frac{B^- B'_m}{A_m}$$

or:

$$\frac{T}{AR' + B^- S} = \frac{B'_m}{A_m} \tag{83}$$

With these factorizations and cancellations we reduced the degree of the closed-loop system. To uniquely solve the Diophantine equation the degree of polynomials from both sides of the equation must be equal. The question is: Is this satisfied?

$$\deg(AR + BS) > \deg(AR' + B^- S) \stackrel{?}{=} \deg(A_m)$$

If we get that $\deg(AR' + B^- S) > \deg(A_m)$, then it follows that some polynomial is missing here. In order to solve Diophantine equation uniquely there should exist a polynomial A_0 (called observer polynomial[17]) which should be cancelled on the right side of the (83):

$$\frac{T}{AR' + B^- S} = \frac{B'_m A_0}{A_m A_0} \tag{84}$$

[14] If we want that at all.

[15] Monic polynomial is the polynomial with a unit at the highest power of the polynomial. Monic polynomial is needed to get the unique factorization.

[16] Since B^- can not be a factor of $AR + BS$ it follows that it must divide B_m.

[17] In relation to the observer from the state space design method, where the dynamics of the observer was not present in the closed loop transfer function of the control system.

Finally, we have the following equations which must be solved for the pole/zero placement problem:
For closed-loop poles the (Diophantine) equation:

$$AR' + B^- S = A_m A_0 \tag{85}$$

For closed-loop zeros:

$$T = B'_m A_0 \tag{86}$$

If we want to have system which is robust for low frequency disturbances as well as for low frequency modelling errors, high gain is needed at low frequency. The integrator must be included in the regulator i.e. the $R(q)$ polynomial should become:

$$R(q) = (q-1)^\nu R_1 \tag{87}$$

With this R polynomial the equation (85) becomes:

$$A(q-1)^\nu R'_1 + B^- S = A_m A_0 \tag{88}$$

Simple calculations show that the causality conditions:

$$\deg R \geq \deg T$$

$$\deg R \geq \deg S$$

are satisfied if:

$$\deg A_m - \deg B_m \geq \deg A - \deg B \tag{89}$$

holds and if:

$$\deg A_0 \geq 2\deg A - \deg A_m - \deg B^+ + \nu - 1 \tag{90}$$

The algorithm for the pole/zero placement is then:

Algorithm: **Pole/zero placement**
Initial data:

- *process model given by the transfer function:* $\frac{B(q)}{A(q)}$
- *known observer polynomial* $A_0(q)$
- *desired closed-loop dynamics given by the implementable transfer function:* $\frac{B_m(q)}{A_m(q)}$
- *the "stability" region* Ω *defined by the user*

The following conditions must be met:

- $B_m(q) = B^-(q)B'_m(q)$
- $\deg A_m - \deg B_m \geq \deg A - \deg B$
- $\deg A_0 \geq 2\deg A - \deg A_m - \deg B^+ + \nu - 1$

Step 1: Factorize B *polynomial as:* $B = B^- B^+$
Step 2: Factorize B_m *polynomial as:* $B_m = B^- B'_m$
where B^+ *is monic with all its roots inside "stability" region* Ω*, and* B^- *has all its roots outside "stability" region* Ω*.*
Step 3: Solve the Diophantine equation:

$$A(q-1)^\nu R'_1 + B^- S = A_m A_0 \tag{91}$$

with respect to R_1' and S. Choose solution such that:

$$\deg S < \deg A + \nu \tag{92}$$

and also

$$\deg R_1' = \deg A_0 + \deg A_m - \deg A - \nu \tag{93}$$

Step 4: The control law is then:

$$R(q)u_c(k) = T(q)r(k) - S(q)y(k) \tag{94}$$

where:

$$R = B^+R' \text{ and } R' = (q-1)^\nu R_1' \tag{95}$$

$$\deg R \geq \deg S \tag{96}$$

and:

$$T = B_m' A_0 \tag{97}$$

$$\deg T \leq \deg R \tag{98}$$

NOTE: From the polynomial identity (95) follows that R' is monic. This also implies that R polynomial is monic.

For stochastic systems described by ARMAX mathematical models, optimal observer polynomial A_0 is equal to polynomial C. Equation (88) then becomes, [39]:

$$A(q-1)^\nu R_1' + B^- S = A_m C \tag{99}$$

From equation (99) follows that C polynomial must also be Hurwitz if we want to have a stable closed-loop system. The pole/zero placement algorithm given above is applicable not only for deterministic but also for stochastic systems as well.

NOTE: *Adaptive version of the pole/zero placement algorithm instead of the polynomials A, B, and C use their estimations $\widehat{A}$, $\widehat{B}$, and $\widehat{C}$.*

So, for stochastic systems we will have the following Diophantine equation:

$$\widehat{A}(q-1)^\nu R_1' + \widehat{B}^- S = A_m \widehat{C} \tag{100}$$

and the pole/zero polynomial equations for deterministic systems will become:

$$\widehat{B} = \widehat{B}^- \widehat{B}^+ \tag{101}$$

$$B_m = \widehat{B}^- B_m' \tag{102}$$

$$\widehat{A}(q-1)^\nu R_1' + \widehat{B}^- S = A_m A_0 \tag{103}$$

$$R = \widehat{B}^+ R' \text{ and } R' = (q-1)^\nu R_1' \tag{104}$$

A.2 Numerical problems related to pole/zero placement method

As is seen from the pole/zero placement algorithm given above, there are couple of equations which should be solved numerically on-line in order to realize the adaptive self-tuning system. From the numerical point of view the main problem still remains in the factorization of B polynomial. Diophantine equation can be solved by the following methods:

1. by use of Jezek's algorithm, [40]
2. by use of Euclid's algorithm, [7]
3. by use of set of linear equations, [34]
4. by use of iterative method of residual correction, [41]
5. by use of recursive LS method, [42]

First three methods solve the Diophantine equation exactly in each step, while other two are iterative converging to exact solutions under assumption of time invariance of the process. Because of that they have the possibility to smooth possible isolated singularities[18], but are not applicable for time varying processes. Due to this disadvantage first three methods are recommendable for adaptive control applications. From the experience gained with first three methods the author recommends the first one as the best for real-time application due to compact code and reliability in operation. The third method can broke if the singularity of the Sylvester matrix occur during calculations. Using Gauss method with complete pivoting[19] will not resolve this problem. The second method (Euklid's algorithm) enable elimination of possible common factors in A and B polynomials in case of a pathologic situation, have more compact code, but is not better than Jezek's algorithm which does not posses these problems at all.

Example 11:
The linear time variant process is given by the transfer function:

$$G(s) = \frac{1}{s^3 + a_2 s^2 + 3s + 1}$$

It is controlled by the two-parameter topology regulator. Parameter a_2 change according to:

$$a_2 = \left\{ \begin{array}{c} 3 \text{ for } 0 < t < 170 \\ 5 \text{ for } 170 \leq t < \infty \end{array} \right\}$$

With the sampling period (ZOH discretization) $h = 0.5$ [s] the following discretized models are obatined:

$$G(z) = \frac{0.0144z^2 + 0.0397z + 0.0068}{z^3 - 1.8196z^2 + 1.1036z - 0.2231} \text{ for } a_2 = 3$$

$$G(z) = \frac{0.012z^2 + 0.0279z + 0.0035}{z^3 - 1.792z^2 + 0.9174z - 0.0821} \text{ for } a_2 = 5$$

The desired dynamics is described by the Bessel ($n+1$ th order) filter with the cuttof frequency 2 [rad/s]:

$$G_m(s) = \frac{16}{s^4 + 6.2479s^3 + 17.5662s^2 + 25.6087s + 16}$$

or ZOH discretized:

$$G_m(z) = \frac{0.022z^3 + 0.1259z^2 + 0.0673z + 0.0034}{z^4 - 1.4792z^3 + 0.9778z^2 - 0.324z + 0.044}$$

[18] Of low probability.
[19] Which is numerically stable algorithm.

Observer polynomial is chosen as the denominator of the Bessel (n th order) filter with the cutoff frequency 4 [rad/s]:

$$A_0(s) = s^3 + 1.7315s^2 + 39.4594s + 64$$

ZOH discretized:

$$A_0(z) = z^3 - 0.2185z^2 + 0.0608z - 0.0077$$

*Measured input/output data are filtered with the second order Butterworth filter with cutoff frequency 0.2 [rad/s]. The response to the reference signal when the regulator parameters are fixed is given in Fig.17. The response to the same reference signal with the self-tuning regulator is given in Fig.*18. *The*

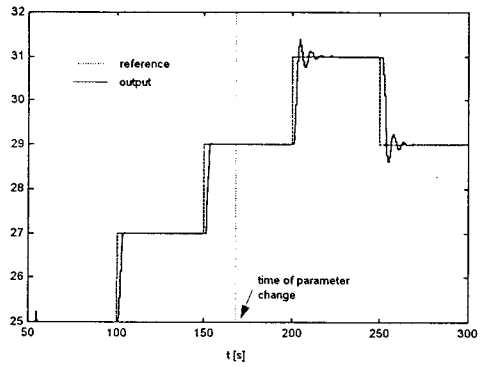


Fig. 17. Response of the non-adaptive control system with time variant process.

self-tuning regulator use RLS parameter estimation algorithm and pole placement design. Regulator parameters change during operation.

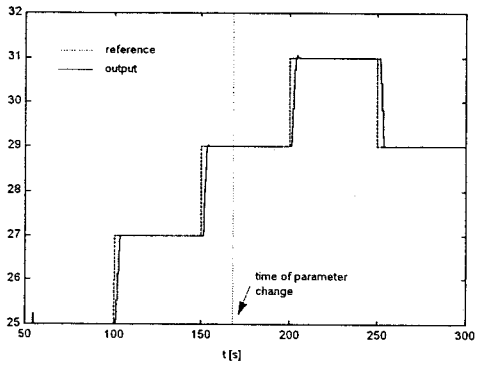


Fig. 18. Response of the ST control system to reference signal.

This example shows that the ST adaptive regulator is capable to control the process and obtain the desired dynamics of the closed-loop system despite process parameter changes.

B. Minimum variance control

Minimum variance control (MV) tries to minimize variances of the process output signal. By doing that this type of control keeps the system in the operating point characterized with minimal waste. Because of that this type of control is interesting for the industry where minimal waste is of prime concern. The performance index of minimum variance is:

$$J_{MV} = E\left\{y^2(k)\right\} \tag{105}$$

where:
E - mathematical expectation,
$y(k)$ - process output signal (response).

The basic idea of the minimum variance control, is that in the k-th instant the regulator must cancel d-step prediction of the system response, where d is the process delay,[43]. As a consequence we state that the minimum variance control will be good when the prediction of the system output is reliable, when process disturbances are slow and when the prediction horizon is short (small process delay). To find the necessary control algorithm for the MV control an optimal d-step predictor (estimator) has to be found which will give at the moment k the prediction of what the output of the process will be at the moment $k+d$. This prediction must be such that the prediction error is minimal. Due to the fact that we are dealing here with stochastic systems, this means that the variance of the prediction error has to be minimized. The prediction error is here defined as:

$$\widetilde{y}(k+d \mid k) = y(k+d) - \widehat{y}(k+d \mid k) \tag{106}$$

and we are seeking:

$$E\left\{\widetilde{y}^2(k+d \mid k)\right\} = \min. \tag{107}$$

The mathematical model of the process is given by the equation (7):

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + C(q^{-1})e(k) \tag{108}$$

where: $e(k)$ is white noise with $E\{e(k)\} = 0$ and $E\{e^2(k)\} = \sigma_e^2$

Optimal predictor of this process can be obtained from the d-step predicted form of the (108) given as:

$$y(k+d) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})}e(k+d) \tag{109}$$

The MA part of the process d-step predicted form is given by:

$$y_{MA}(k+d) = \frac{C(q^{-1})}{A(q^{-1})}e(k+d) = \Gamma(q^{-1})e(k+d) \tag{110}$$

where $\Gamma(q^{-1}) = 1 + \gamma_1 q^{-1} + \gamma_2 q^{-2} + \cdots$ is a polynomial of infinite degree, obtained by polynomial division of C and A polynomials. This polynomial will converge assuming that C is Hurwitz polynomial! The mathematical model is only then invertible because $e(k)$ can be obtained from previous realizations of output signal as:

$$e(k) = \frac{A(q^{-1})}{C(q^{-1})}y(k) = \left[\Gamma(q^{-1})\right]^{-1}y(k)$$

Bear in mind that the white noise sequence $e(k)$ have independent realizations of each other, which means that we are not capable to predict the next realization, knowing previous one. However, we can separate the right side of the equation (110) to one part which we are unable to predict and the another part which at the moment k is already known through realizations of the white noise up to k-th instant. We then have:

$$y_{MA}(k+d) = \left[\underbrace{e(k+d) + \gamma_1 e(k+d-1) + \cdots + \gamma_{d-1}e(k+1)}_{\text{Unpredictable at moment } k}\right] + \left\{\underbrace{\gamma_d e(k) + \gamma_{d+1}e(k-1) + \gamma_{d+2}e(k-2) + \cdots}_{\text{Realized up to moment } k}\right\} \tag{111}$$

So, the best prediction we can make, boils down to the part which is inside braces:

$$\widehat{y}_{MA}(k+d \mid k) = \underbrace{\gamma_d e(k) + \gamma_{d+1} e(k-1) + \gamma_{d+2} e(k-2) + \cdots}_{\text{Realized up to moment } k} \tag{112}$$

As a consequence it follows that for the MA process the prediction error is given by the unpredictable part at the moment k given by:

$$\widetilde{y}_{MA}(k+d \mid k) = \underbrace{e(k+d) + \gamma_1 e(k+d-1) + \cdots + \gamma_{d-1} e(k+1)}_{\text{Unpredictable at moment } k} \tag{113}$$

From (110) and (113) it follows that the polynomial $\Gamma(q^{-1})$ can be factorized to the following:

$$y_{MA}(k+d) = \frac{C(q^{-1})}{A(q^{-1})} e(k+d) = \Gamma(q^{-1}) e(k+d) = R'(q^{-1}) e(k+d) + \frac{S(q^{-1})}{A(q^{-1})} e(k) \tag{114}$$

where:
$R'(q^{-1}) = 1 + r'_1 q^{-1} + \cdots + r'_{d-1} q^{-d+1}$
$S(q^{-1}) = s_0 + s_1 q^{-1} + \cdots + s_{n-1} q^{-n+1}$
From(114) the Diophantine equations follows:

$$C(q^{-1}) = A(q^{-1}) R'(q^{-1}) + q^{-d} S(q^{-1}) \tag{115}$$

or:

$$q^{d-1} C(q) = A(q) R'(q) + S(q) \tag{116}$$

Optimal predictor of the MA process is obtained from minimizing the variance of the prediction error:

$$E\left\{\widetilde{y}^2_{MA}(k+d \mid k)\right\} = E\left\{\left[y_{MA}(k+d) - \widehat{y}_{MA}(k+d \mid k)\right]^2\right\} = \min . \tag{117}$$

$$\begin{aligned} \min . = E\left\{\left[R'(q^{-1}) e(k+d)\right]^2\right\} + E\left\{\left[\frac{S(q^{-1})}{C(q^{-1})} y_{MA}(k) - \widehat{y}_{MA}(k+d \mid k)\right]^2\right\} + \\ 2E\left\{\left[R' e(k+d)\right]\left[\frac{S}{C} y_{MA} - \widehat{y}_{MA}(k+d \mid k)\right]\right\} \end{aligned} \tag{118}$$

Due to the fact that the white noise sequence $e(k+d)$, $e(k+d-1), \ldots, e(k+1)$ is independent from $y_{MA}(k)$, $y_{MA}(k-1), \ldots$ with zero mean, then the third component in the expression above will be equal to zero. The only possible condition for the minimum of the variance of the prediction error is then:

$$\frac{S(q^{-1})}{C(q^{-1})} y_{MA}(k) - \widehat{y}_{MA}(k+d \mid k) = 0 \tag{119}$$

And the optimal predictor for the MA process is:

$$\widehat{y}_{MA}(k+d \mid k) = \frac{S(q^{-1})}{C(q^{-1})} y_{MA}(k) \tag{120}$$

With this predictor the variance of the prediction error is:

$$E\left\{\left[R'(q^{-1})e(k+d)\right]^2\right\} = (1 + r_1'^2 + \cdots + r_{d-1}'^2)\sigma_e^2 \tag{121}$$

Introducing (114) in (109) we get the prediction form for the ARMAX process as:

$$y(k+d) = R'(q^{-1})e(k+d) + \frac{S(q^{-1})}{A(q^{-1})}e(k) + \frac{B(q^{-1})}{A(q^{-1})}u(k) \tag{122}$$

From (108) is:

$$e(k) = \frac{A(q^{-1})}{C(q^{-1})}y(k) - q^{-d}\frac{B(q^{-1})}{C(q^{-1})}u(k) \tag{123}$$

and (122) then becomes:

$$y(k+d) = R'e(k+d) + \frac{S}{A}\left[\frac{A}{C}y(k) - q^{-d}\frac{B}{C}u(k)\right] + \frac{B}{A}u(k)$$

$$= R'e(k+d) + \frac{S}{C}y(k) + \frac{B(C - q^{-d}S)}{AC}u(k)$$

$$= R'e(k+d) + \frac{S}{C}y(k) + \frac{BR'}{C}u(k) \tag{124}$$

Optimal predictor can be obtained from the minimum variance of the prediction error:

$$\min. = E\left\{\left[R'e(k+d)\right]^2\right\} + E\left\{\left[\frac{S}{C}y(k) + \frac{BR'}{C}u(k) - \widehat{y}(k+d \mid k)\right]^2\right\} +$$
$$2E\left\{\left[R'e(k+d)\right]\left[\frac{S}{C}y(k) + \frac{BR'}{C}u(k) - \widehat{y}(k+d \mid k)\right]\right\}$$

Following the same reasoning as before with MA process, we get the optimal predictor for the ARMAX process:

$$\widehat{y}(k+d \mid k) = \frac{S(q^{-1})}{C(q^{-1})}y(k) + \frac{B(q^{-1})R'(q^{-1})}{C(q^{-1})}u(k) \tag{125}$$

The minimum variance control algorithm is obtained by cancelling the d-step prediction of the process output given by (125). Thus, the control algorithm of the MV control is:

$$u(k) = -\frac{S(q^{-1})}{B(q^{-1})R'(q^{-1})}y(k) = \frac{-S(q^{-1})}{R(q^{-1})}y(k) \tag{126}$$

where:
$R(q^{-1}) = B(q^{-1})R'(q^{-1})$
As is seen from (126) the MV control structure is quite simple and can be represented by the block diagram given in Fig.19.

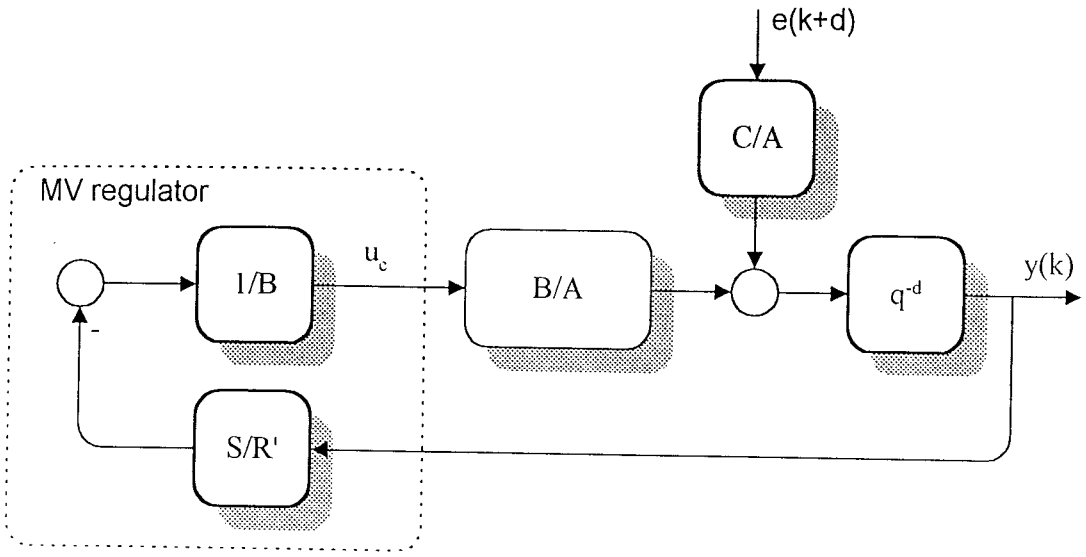


Fig. 19. Block diagram of the minimum variance regulator.

MV regulator is a regulator with one degree of freedom, because we are able to position the poles only. Diophantine equations for the MV problem are:

$$A(q^{-1})R(q^{-1}) + q^{-d}B(q^{-1})S(q^{-1}) = B(q^{-1})C(q^{-1}) \tag{127}$$

or:

$$A(q)R(q) + B(q)S(q) = q^{d-1}B(q)C(q) \tag{128}$$

In the pole/zero setting it can be concluded that the MV algorithm is obtained if the desired implementable $A_m(q)A_0(q)$ polynomial is set to be:

$$A_m(q)A_0(q) = q^{d-1}B(q)C(q) \tag{129}$$

So, the MV control sets the closed loop poles at the roots of B and C polynomials, while the remaining $d-1$ poles are set at the origin. Because of this characteristic the MV control is applicable only for the minimum phase systems. If our process have unstable zeros, than the MV regulator will be unstable. So, factorization of B polynomial is needed, in order to cancel only stable process zeros (B^+) and leaving the unstable process zeros (B^-). If this is made, than we have suboptimal MV control called Moving Average (MA) control:

$$A(q)R(q) + B(q)S(q) = q^{d-1}B^+(q)C(q) \tag{130}$$

The minimum of variance can not be achieved with the MA control! MV and MA control are similar and differ in number of process zeros cancelled. MV control cancel all process zeros while MA control cancel only B^+ zeros.

The same idea of cancelling the prediction of the output signal have been used for control of stochastic systems such as:

1. Linear Quadratic Gaussian (LQG) control, [14]
2. Generalized Minimum Variance (GMV) control, [15]
3. Generalized Predictive Control (GPC), [44], [45]
4. etc.

Minimum variance control is a special case of linear quadratic optimal control, which can be represented with the following performance index:

$$J = E\left\{[y(k) - r(k)]^2 + \rho u^2(k)\right\} = \min. \tag{131}$$

The control law minimizing this criterion is the Linear Quadratic Gaussian (LQG) control. If $\rho = 0$ and $r(k) = 0$ we have the MV regulator. Because the weighting factor for control signal is zero, and the weighting factor for the output is 1, the MV control will not care about the control energy needed for the task. Due to the fact that weighting factors are necessary and desirable because we have more freedom when defining the close loop dynamics of the system, other controllers were suggested in the literature. Reference [46] is reccomendable for that purpose because it gives a unified approach to many predictive control regulators suggested over time in the literature.

C. LQG Control

LQG control can be related with the pole placement problem quite easy. Minimization of the performance index (131) leads to a fixed gain controller which can be interpreted as the pole placement controller which will place the closed-loop poles at the roots of:

$$A_m(q)A_0(q) = P(q)C(q) \tag{132}$$

where the polynomial $P(q)$ is the polynomial that satisfies the spectral factorization problem given as:

$$\mu P(q)P(q^{-1}) = \rho A(q)A(q^{-1}) + B(q)B(q^{-1}) \tag{133}$$

The Diophantine equation is then:

$$A(q)R(q) + B(q)S(q) = P(q)C(q) \tag{134}$$

To get the unique solution with $\deg(R) = \deg(S) = n$ some restrictions has to be made to the solution of the Diophantine equation (134), see [4] p.164 for more detail.

D. Industrial adaptive regulators - a short review

Industrial adaptive regulators are today common in process control. They have been accepted cautiously in the industry due to conservatism and good experience which the process industry have had with conventional PID type regulators. Today selftuning controllers are well accepted in industry. Rough estimation from 1989, [4] gives the number of 100,000 loops with some sort of adaptive control applied. We can mention the following regulators offered first at the market:

- Adaptive PID selftuning regulator ElectroMax by Leeds & Northrup,
- ASEA Novatune minimum variance selftuning regulator (from 1982) by ASEA (today ABB), then incorporated in ASEA MASTERPIECE distributed control system for process industry. More than 2500 control loops were realized with this type in late eighties,
- First pole placement adaptive regulator was offered around mid eighties by Swedish firm First Control Systems, AB,
- Soon followed others like: DPR (by Fisher Control, UK) ECA40 (by SATT Control Instruments, Sweden), Eurotherm (by Eurotherm, UK), West (by West Instruments, UK), Supertuner (by Techmation, US), P-200 (by Powell Process Instruments, US) etc.

Number of control loops in industry, which use adaptive controllers, is estimated to rise around 4000 loops per year.

V. Conclusion

Theory of adaptive control systems developed during late sixties and early seventies. First implementation in industry occurs in mid seventies. Today adaptive regulators are offered as a standard choice for the user in the open systems such as DCV700 system for AC Drives by ABB (Sweden) or many other companies like Siemens, Allen Bradley etc. Due to technological developments and computer engineering achievements, the user has today on his disposal a user friendly graphical compiler(s) which consist of many functional elements realized as software modules, enabling easy programming of any type of controller. However, the adaptive control systems should be carefully designed and justified for a particular application. Feasibility study is of a paramount importance here, because use of adaptive control should be justified and compared with the use of other type of control.

Research today is directed toward higher level control systems where the adaptive control is only one of many possible control algorithms (Fig.20, from [4]).

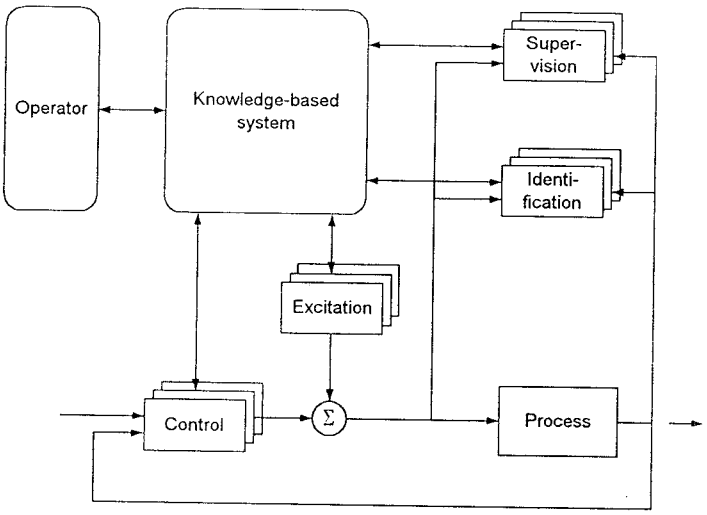


Fig. 20. Knowledge based control system.

We can mention here autonomous vehicles as a challenge for a control engineering point of view. Autonomous vehicles of any kind (underwater, terrain or flying) should operate by its own (without operator), be capable to reach some conclusions about the mission of a vehicle under various unpredictable situations and reach the goal(s) set by the operator. A lot has to be done before this scenario can be realized. We can mention only couple of topics which are today in focus of research in control community: FDIA (Failure Detection Isolation and Accommodation), FTC (Fault Tolerant Control), reconfigurable control, intelligent (autonomous) control, self-learning control, fuzzy/neuro control etc. This tutorial covers only a small part of the whole picture, but we are certain that without the knowledge of achievements of the adaptive control theory, this task can not be resolved.

References

[1] Åström K. J., "Adaptive control arround 1960," *IEEE Control Systems*, pp. 44–49, June 1996.
[2] Vukic Z., "Design of adaptive ship guidance system," Ph.D. dissertation, University of Zagreb, Croatia (in Croatian), 1989.
[3] H. Cuong, "Investigation of methods for adaptive path control of surface ships," Ph.D. dissertation, Naval Architecture and Marine Engineering, The University of Michigan, 1980.
[4] Åström K. J. and B. Wittenmark, *Adaptive Control*, Reading, Massachusets: Addison-Wesley, second edition, 1995.
[5] Blomberg H. and R. Ylinen, *Algebraic Theory for Multivariable Linear Systems*, New York: Academic Press, 1983.
[6] Chen C. T., *Linear System Theory and Design*, New York: Holt, Rinehart and Winston, 1984.
[7] Kucera V., *Discrete Linear Control: The Polynomial Equation Approach*, Chichester: John Wiley and Sons, 1979.
[8] Wolovich W. A., *Linear Multivariable Systems*, New York: Springer - Verlag, 1974.
[9] Wonham W. M., *Linear Multivariable Control - A Geometric Approach*, New York: Springer - Verlag, 1974.
[10] E. I. Jury, "Hidden oscillations in sampled-data control systems," *Trans. AIEE*, vol. 75, pp. 391–393, 1956.
[11] Åström K. J., P. Hagander, and J. Sternby, "Zeros of sampled systems," *Automatica*, vol. 20, no. 1, pp. 31–38, 1984.

[12] Box G. P. and G. H. Jenkins, *Time Series Analysis Forecasting and Control*, Holden Day, 1976.

[13] Akaike H., "Fitting autoregressive models for prediction," *Ann. Inst. Statist. Math.*, vol. 21, pp. 243–247, 1969.

[14] Clarke D. W. P.P. Kanjila and C. Mohtad, "A generalized LQG approach to self-tuning control: Part i aspects of design," *International Journal of Control*, vol. 41, pp. 1509–1523, 1985.

[15] Clarke D. W. and P. J. Gawthrop, "Self-tuning controller," *Proc. IEE*, vol. 122, pp. 929–934, 1975.

[16] Clarke D. W., A. J. Hodgson, and P. S. Tuffs, "Offset problem and k-incremental predictors in self-tuning control," *Proc IEE*, vol. 130, pp. 217–225, 1983.

[17] Bryson A. E. and Y-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*, Washington DC: Hemisphere Publ. Co. (John Wiley and Sons), 1975.

[18] Grimble M. J. and M. A. Johnson, *Optimal Control and Stochastic Estimation: Theory and Applications*, vol. 1 2, Chichester: John Wiley and Sons, 1988.

[19] Jacobs O. L, "Introduction to adaptive control," in *Self-Tuning and Adaptive Control Theory and Applications*, Harris C. J. and S. A. Billings, Eds. London: Peter Peregrinus, 1981.

[20] Patchel P. C. and O. L. Jacobs, "Separability neutrality and certainty equivalence," *International Journal of Control*, vol. 13, pp. 337–342, 1971.

[21] Jacobs O. L. and J. W. Patchel, "Caution and probing in stochastic control," *International Journal of Control*, vol. 16, pp. 189–199, 1972.

[22] Bar-Shalom Y. and E. Tse, "Dual effect, certainty equivalence and separation in stochastic control," *IEEE Trans. AC*, vol. 19, pp. 494–500, 1974.

[23] Feldbaum A. A, *Optimal Control Theory (in Russian)*, Moscow: Nauka, 1970.

[24] Eykhoff P., *System Identification: Parameter and State Estimation*, London: John Wiley and Sons, 1974.

[25] Ljung L. and K. J. Söderström, *Theory and Practice of Recursive Identification*, Cambridge: The MIT Press, 1983.

[26] Ljung L., *System Identification: Theory for the User*, Englewood Cliffs: Prentice Hall Inc., 1987.

[27] Bohlin T., "Model validation," Tech. Rep. TRITA-REG-8203, KTH, Stockholm, 1982.

[28] Sripada N. R. and D. G. Fisher, "Improved least squares identification," *International Journal of Control*, vol. 46, pp. 1889–1913, 1987.

[29] Söderström T., L. Ljung, and I. Gustavsson, "A theoretical analysis of recursive identification methods," *Automatica*, vol. 14, pp. 231–244, 1978.

[30] Wellstead P. E. and S. P. Sanoff, "Extended self-tuning control," *International Journal of Control*, vol. 34, pp. 433–455, 1981.

[31] Fortescue T. R., L. S. Kershenbaum, and B. E. Ydstie, "Implementation of self-tuning regulators with variable forgetting factors," *Automatica*, vol. 17, pp. 831–835, 1981.

[32] Goodwin C. G. and S. K. Sin, *Adaptive Filtering Prediction and Control*, Englewood Cliffs: Prentice Hall Inc., 1984.

[33] Vogel E. F. and T. F. Edgar, "Application of an adaptive pole-zero placement controller to chemical processes with variable dead time.," in *Proc. 1982 American Control Conference*, Arlington, 1982. pp. 536–544.

[34] Strang G., *Linear Algebra and its Applications*, Orlando: Academic Press, 1980.

[35] Bierman G. J., *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.

[36] Bohlin T., "On the maximum likelihood method of identification," *IBM J. Res. Development*, vol. 14, pp. 41–51, 1970.

[37] Rohrs C. L., M. Valavani, M. Athans, and G. Stein, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Trans. AC*, vol. 30, pp. 881–889, 1985.

[38] Desoer G. A. and M. Vidyasagar, *Feedback Systems: Input-Output Properties*, New York: Academic Press, 1975.

[39] Wellstead P. E., J. M. Edmunds, D. Prager, and P. Zanker, "Selftuning Pole/Zero assignment regulators," *International Journal of Control*, vol. 30, pp. 1–26, 1979.

[40] Jezek J., "New algorithm for minimal solution of linear polynomial equations," *Kybernetika*, vol. 18, pp. 505–516, 1982.

[41] Sharoki M. and M. Morari, "Using an adaptive observer to design pole placement controllers for discrete-time systems.," *International Journal of Control*, vol. 36, pp. 695–710, 1982.

[42] Leal L. R. and I. Landau, "Quazi-direct adaptive control for nonminimum phase systems," *Journal of Dynamics Systems, Measurement and Control*, vol. 104, pp. 311–316, 1982.

[43] Åström K. J., *Introduction to Stochastic Control Theory*, New York: Academic Press, 1970.

[44] Clarke D. W., C. Mohtad, and P. S. Tuffs, "Generalized predictive control - part i: The basic algorithm," *Automatica*, vol. 23, pp. 137–148, 1987.

[45] Clarke D. W., C. Mohtad, and P. S. Tuffs, "Generalized predictive control - part II: Extensions and interpretations," *Automatica*, vol. 23, pp. 149–160, 1987.

[46] Soeterboek R., *Predictive Control - a Unified Approach*, New York: Prentice Hall Int., 1992.