

Miroslav Kárný; Alena Halousková; Josef Böhm; Rudolf Kulhavý; Petr Nedoma  
Design of linear quadratic adaptive control: theory and algorithms for practice

*Kybernetika*, Vol. 21 (1985), No. Suppl, (1a),3--96

Persistent URL: <http://dml.cz/dmlcz/125074>

## Terms of use:

© Institute of Information Theory and Automation AS CR, 1985

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*  
<http://project.dml.cz>

# Kybernetika

---

**DESIGN OF LINEAR QUADRATIC ADAPTIVE CONTROL:  
THEORY AND ALGORITHMS FOR PRACTICE**

MIROSLAV KÁRNÝ, ALENA HALOUSKOVÁ, JOSEF BÖHM,  
RUDOLF KULHAVÝ, PETR NEDOMA

ACADEMIA  
PRAHA

A linear quadratic gaussian adaptive digital controller based on recursively identified (multivariate) regression model is presented. Writing the paper the authors had in mind the basic question of a potential user: "How to use your controller in my specific problem?" That is why more attention than usually is devoted to the thorough problem formulation including motivations for the chosen methodology and description of head-stone algorithms for identification and control synthesis. To provide a feeling for the applicability range, brief information about related software as well as available practical experience is added.

## 1. INTRODUCTION

The adaptive control derives its name from its most useful feature: adaptation with respect to some a priori unknown or slowly varying conditions requiring appropriate changes in adjustment and readjustment of control algorithm. Various ways to achieve this property have been surveyed in [1]. The aim of this paper is to give an exposition of the specific approach to algorithmization of adaptive control developed in the Institute of Information Theory and Automation of Czechoslovak Academy of Sciences. The state of control theory, the impact of microprocessor technology and existing successful applications justify the authors' belief that broader use of the adaptive control has become possible. This is the reason why the research has been focused on practical applicability of its results. Emphasis has been laid on:

- feasible problem formulation reflecting practical requirements in most industrial applications,
- thorough algorithmization respecting both efficiency and numerical stability of calculations,
- universal software support,
- broad laboratory verification as well as some experience from pilot and full scale industrial applications.

The paper should make it possible for the reader to profit from the available control software. The problem formulation and the solution are discussed in details, while the underlying theory and the world state-of-art are limited to inevitable facts.

Experience has confirmed how important it is for a potential user of the adaptive control to ponder its possible advantages and limitations. Properly designed adaptive control is expected:

- to improve the quality of closed loop behaviour because nearly optimal control can be achieved (under realistic requirements on the extent of prior knowledge needed),
- to manage with a simple control law in a wide range of set points owing to the self-tuning properties,
- to shorten the time needed for preliminary adjustment (which is almost exclusively performance-oriented).

On the other hand it is necessary to take into account that:

- no control, including adaptive one, can make up for reliable technology,

- it makes no sense to try out more complicated adaptive control when simple existing control means are fully satisfactory,
- more sophisticated controllers demand higher level of designer's knowledge.

Formally, the discussed control is in many respects well known. The feedback controller is designed (approximately) to minimize the quadratic criterion

$$\lim_{N \rightarrow \infty} E \left[ \frac{1}{N} \sum_{i=1}^N (e'_y(t) Q_y e_y(t) + e'_u(t) Q_u e_u(t)) \right]$$

where expected value  $E(\cdot)$  is taken of an average loss per control step, weighting by positive semidefinite matrices  $Q_y, Q_u (\geq 0)$  the output error  $e_y$  and the input control effort  $e_u$ . The relation of the input  $u$  and output  $y$  is modelled by the regression model

$$y(t) = \sum_{i=1}^{t_y} A_i y(t-i) + \sum_{i=0}^{t_u} B_i u(t-i-t_u) + c + \varepsilon(t)$$

where  $\varepsilon$  is the "white noise" and unknown matrix coefficients  $A_i, B_i$  and the absolute term  $c$  are identified in real time by a well-known recursive least square method.

There is a long way from textbook theory to practice. We shall try to go this way and enter the area of control algorithms. This is the reason why we have felt to be forced to start the paper at another than usual point and to come to the above formulation after tens of pages. We should like to consider with the reader some practical as well as conceptual questions, which we have posed to ourselves in connection with adaptive control and which, we believe, anybody would ask. The red line which should be felt throughout the text rests on the thought dialogue with the potential user. The paper should answer his questions: Why have you chosen just this way? Is it really necessary? Where and why did you make some concessions? Hoping that he (you) becomes the practical user we try sincerely to prepare answers to practical questions: Are there some rules how to "adapt" your adaptive controller to my problem(s)? Which are the guidelines and rules for the choice of this or that characteristic? Where the troubles are to be expected? . . .

Unfortunately, these and similar questions can be answered only in relative terms (greater, smaller); more precise recommendations we could give are dependent on (unknown) properties of the controlled system.

The paper is organized as follows:

After introducing some common conventions, the mutual relations of the user and a real system to be controlled are explained, including the conditions determining possible outcomes of the control algorithm. Within given limitations the controller is expected to optimize the closed loop behaviour. Its quantification under uncertainty is thoroughly discussed. Dynamic programming which admits effective optimization of causal control is then recalled. In connection with optimization the need for a system model appears. Its general form is presented as well as a practical way of model building through the experience accumulation. The regression model is shown to be the model giving just necessary characteristics needed in the particular

case of quadratic criterion. Substantial aspects of identification of the regression model are discussed (including structure determination and tracking of its varying parameters). Principles of some suboptimal strategies which, within reasonable computational complexity, approximately optimize the quadratic criterion (using recursively identified regression model) are then reviewed. Two ways of algorithmization of adaptive control synthesis differing in flexibility and computational efficiency are described at length. A brief account of the software tools and practical experience conclude the paper.

#### ACKNOWLEDGEMENT

Our research in the field of adaptive control has been initiated, guided and stimulated by Dr. V. Peterka, who is the author of the fundamental contributions [2]. By writing this paper we acknowledge his way of theoretical as well as practical reasoning, respecting always the ultimate goal of practical applicability.

#### REFERENCES

- [1] Special issue on Adaptive Control. *Automatica* 20 (1984), 5, 499–711.  
 [2] V. Peterka: *Digital Control of Processes with Stochastic Disturbances and Uncertain Characteristics* (in Czech). Doctor of Science Dissertation, ÚTIA ČSAV, Prague 1975.

## 2. MATHEMATICAL FORMULATION OF THE TECHNICAL PROBLEM

The way of mathematical problem formulation and solution is heavily influenced by computational aspects leaving only a limited space to respect practical constraints. The thorough clarification of the connections and gaps between the technical and mathematical control problems gives the user some hints how to achieve effective control for a broad class of systems.

Fig. 1 will serve as a frame of reference for our explanations.

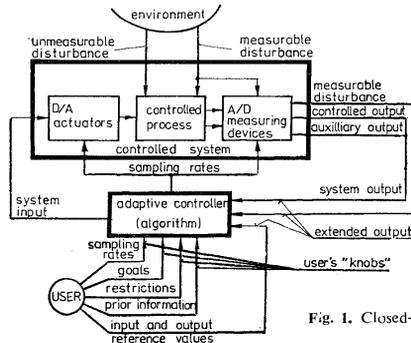


Fig. 1. Closed-loop versus user relations.

## 2.1. Some conventions and notations

### 2.1.1. Common conventions

- [1.2] ... reference [2] in Chap. 1  
(5.3) ... Eq. (3) in Chap. 5  
 $a, b$  ... column vectors  
 $A, B$  ... matrices  
 $a', A'$  ...  $a, A$  transpose  
 $\text{tr}(A)$  ... trace of  $A$  ( $\sum_i A_{ii}$ )  
 $m_x, i_z$  ... dimensions of vectors  $x, z$  ( $i$  is used when the dimension is compound)  
 $a > b$  ... "scalar" operations applied to vectors are understood entrywise  
 $S > 0$  ( $S \geq 0$ ) ... symmetric positive (semi) definite matrix (if  $z \neq 0$  then  $z'Sz > 0$  ( $z'Sz \geq 0$ ))  
 $A \geq B$  ...  $A, B$  are symmetric positive semidefinite and  $A - B \geq 0$   
 $Q_x \geq 0$  ... kernel of the quadratic form  $x'Q_x x$  penalizing the variable  $x$   
 $\int a(x) dx$  ... multivariate and multiple integration is performed over the entire  $x$ -space

### 2.1.2. Closed loop variables

- $u$  ... system input  
 $y_c$  ... controlled system output  
 $y_a$  ... auxiliary system output  
 $y' = (y'_c, y'_a)$  ... system output  
 $v$  ... external measurable disturbance  
 $\varepsilon$  ... innovation representing unmeasurable disturbance  
 $y_0$  ... reference value of  $y$ ; the part corresponding to  $y_c$  is called command signal (if time-varying) or set-point (if time-invariant)  
 $u_0$  ... reference value of input  
 $y'_x = (y', u'_0, y'_0, v')$  ... extended output  
 $d' = (u', y'_x) = (u', y'_c, y'_a, u'_0, y'_0, v')$  ... data item

### 2.1.3. Variables related to time

- $T_s$  ... sampling period; input is generated at time points  $tT_s + T_0$  for some  $T_0$  and  $t = 1, 2, 3, \dots$  ( $t$  is called discrete time)  
 $T_c$  ... the time needed to compute a particular value of  $u$   
 $x(t)$  ... a variable related to, i.e. measured within or computed for, the  $t$ th time interval  $\langle tT_s + T_0 - T_c, (t+1)T_s + T_0 - T_c \rangle$   
 $x(t | \tau)$  ... a variable computed for the  $t$ th interval using information available before the time point  $(\tau + 1)T_s + T_0 - T_c$

$$\begin{aligned}
x(t \dots \tau) &= (x(t), x(t+1), \dots, x(\tau)) \text{ for } \tau > t \\
&= x(t) & \tau = t \\
&= \text{empty list} & \tau < t
\end{aligned}$$

#### 2.1.4. Random variables

$p(a(t) | t-1; b(s))$  ... conditional probability function, abbreviated as c.p.f. (discrete case), or conditional probability density function c.p.d.f. (continuous case) of the random variable  $a(t)$  conditioned on  $d(1 \dots t-1)$  and on another item  $b(s)$ ; conditional probability reduces to the unconditional one when the list in condition part is empty

$E[a(t) | t-1; b(s)] = \int a(t) p(a(t) | t-1; b(s)) da(t)$  ... expected value of  $a(t)$  conditioned on  $d(1 \dots t-1)$  and  $b(s)$

$\text{cov}(a(t) | t-1; b(s)) = E[(a(t) - E(a(t) | t-1; b(s)))(a(t) - E(a(t) | t-1; b(s)))' | t-1; b(s)]$  ... conditional covariance (matrix) of  $a(t)$  conditioned on  $d(1 \dots t-1)$  and  $b(s)$

#### 2.2. Quantities in closed control loop

The part of real world that is to be controlled is called the controlled system. The controller, supposed to be realized by a digital computer, is to control this system by a proper sequence of inputs so that the desired output behaviour might be achieved. The computer algorithm realizing this task operates on two kinds of quantities:

- sampled data measured on the system ( $A/D$  and  $D/A$  converters, actuators as well as measuring devices are necessarily included into the system),
- information supplied by the user (forcing him to quantify appropriately the control objectives, knowledge of the system, technological and economical restrictions etc.).

The particular quantities related to the closed loop are defined here (cf. Fig. 1).

- System input ( $u$ ) is directly manipulated by the controller and fed into the system in order to influence it.
- System output ( $y$ ) consists of quantities influenced by input. The part which is required to be driven is called controlled output ( $y_c$ ). An auxiliary output ( $y_a$ ), when at disposal, improves the attainable quality of control by bringing additional information about system state (samples of an analogue input realization, drawn in Fig. 2, are rather useful in this role).
- External measured variable ( $v$ ), when available, plays the same role as  $y_a$ , however, neither output nor input can influence it, even indirectly. (The part of the controller transforming  $v$  into a component of  $u$  is known as feed-forward).
- The reference values of input ( $u_0$ ) and output ( $y_0$ ) are fed into the controller by the user according to his requirements or some measurements gained (possibly) from another system.

- Extended output ( $y'_s = (y', u'_0, y'_0, v')$ ) is introduced in order to join quantities to be predicted for control design.
- Data item ( $d' = (y'_s, u')$ ) brings together closed-loop signals. Data in the above formulation are always noise-free (with the exception of the round-off errors neglected in formalized design; proposed algorithms are, however, implemented in a numerically safe way).

### 2.3. Sampling

Measured variables have to be usually sampled. In most cases the sampling as well as system input computation is periodic. The input  $u(t)$  is fed into the system at the time instant  $T_0 + tT_s$ , where  $t = 1, 2, 3, \dots$  is a discrete time,  $T_0$  relates discrete time to real time and  $T_s$  is a sampling period. Naturally,  $T_s$  has to be greater than the period  $T_c$  necessary to compute the particular  $u$ . The latest quantities which are available to compute  $u(t)$ , and are not available for  $u'(t-1)$  (sampled in the range  $\langle T_0 + (t-1)T_s - T_c, T_0 + tT_s - T_c \rangle$ ), will be labelled by  $t-1$ , see Fig. 2. Adherence to this (optional) convention is important for avoiding confusion in time indexing.

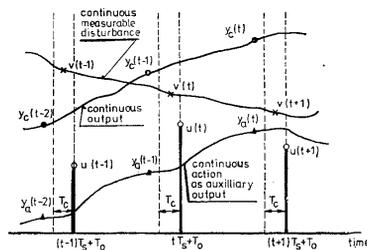


Fig. 2. Sampling and time relations.

**Remark.** The rates of sampling and of the input computation need not be equal. A continuous variable can be sampled many times within an interval of the length  $T_s$ . Intersamples either form a multivariate discrete-time version of a singlevariate continuous variable or can be used to make measurements more reliable by averaging them into a single virtual discrete-time variable.

The choice of the sampling period  $T_s$ , in spite of being fundamental for any digital control, is still open to research. Fortunately, the practical choice presents usually no extra difficulties. Common sense and a few observations listed below give a satisfactory answer.

The sampling process reduces information contained in the continuous measure-

ments, contingently removing unnecessary and misleading details which are a part of any signal. Thus it might be not advantageous to reduce  $T_s$  to zero (even if the hardware admits it).

The choice of sampling rate becomes an integral part of the control design because it influences heavily the discrete output signal to be processed. Very fast sampling makes it possible to observe high-frequency disturbances and the optimal controller tries to eliminate them, sometimes uselessly or even harmfully complicating its function. On the other hand, very slow sampling may bring too little information and the control quality decreases.

Thus a choice of the sampling rate has to take into account disturbance spectra, time constants of the controlled systems and admissible spectra of the closed loop input. The rule of thumb “take about fifteen samples into significant (?) part of the system step response”, even if often successful, is generally oversimplified.

#### 2.4. Data processing

A control algorithm is, as a rule, only a small part of software ensuring the digital control – the dominating computer activity is data management. The following remarks point out some cases (directly related to control) when special data processing is of importance.

- As discussed later, the adaptive controller is sensitive to outliers, rare exceptional data occurring due to a failure (e.g. of a measuring device). Thus a simple test for suitable range of measurements and/or their changes is almost inevitable. Data lying beyond this range must not be fed into the closed loop and appropriate warning is to be sent to the user.
- A user-friendly communication with the computer is conducted in terms of usual units (physical, chemical, etc.). Algorithm-friendly data, however, have to be based on another point of view. Awkward scaling may cause higher “computation noise”, scaling has also its impact on convergence of algorithms. For example, an (almost) linear dependence of data makes convergence of identification algorithms rather slow.
- Filtering is a usual way of making measurements more reliable. Of course, any filter becomes a part of the controlled system, modifying its dynamics. A simple class of filters has been proposed in [1] that handle separately the data measured within one sampling interval, quaranteeing in such a way small influence on the system dynamics. The idea is to measure  $m$ -times during the sampling interval and to approximate the evolution of the obtained data  $x_1^*, \dots, x_m^*$  by a very simple curve, e.g. straight line. The filtered value  $x$  is then given by the relation

$$x = \frac{2}{m} \left[ \frac{3}{m+1} \sum_{i=1}^m i x_i^* - \sum_{i=1}^m x_i^* \right]$$

as illustrated in Fig. 3. A hybrid filter implementing a similar idea can be found in [2].

- Errors often result from inappropriate handling of constrained inputs (cf. Chap. 4). When changed by constraint, the new (constrained) data item has to be fed into the algorithm instead of the computed one.

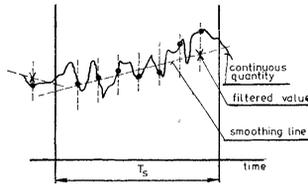


Fig. 3. Filtering of data.

## 2.5. Admissible control strategies

A controller maps sequentially the available information onto system inputs. The sequence of such mappings is called control strategy and its choice forms the essence of the controller design.

The user, acting in open loop (cf. Fig. 1), has to delimit beforehand admissible control strategies specifying the available information for the determination of  $u(t)$  as well as its admissible action range. The strategy can be of a practical use only when corresponding computations are within admissible complexity.

### 2.5.1. Available information

The time-notational conventions introduced in 2.1, 2.2 admit to formalize the simple but important fact that the input  $u(t)$  can be evaluated using the data with time indices up to and including  $t - 1$ , i.e.  $u(t) = u(t | t - 1)$ .

Different control problems can be classified according to available information on reference values. In all cases it proved to be useful, whenever possible, to choose them in such a way that

$$(1) \quad u(t) = u_0(t) \quad \text{implies} \quad y(t) \approx y_0(t).$$

The following cases can be distinguished:

*Program control* – the future values of command signal are known, described by any equivalent form (function, table). Formally,

$$(2) \quad y_0(t) = y_0(t | 0).$$

Such a prior knowledge enables the controller to prepare its actions in advance.

Reference values of inputs are to be adapted to (2) in order to fulfil (1). The simplest but the most usual case is the

*regulation problem*, specified by

$$(3) \quad y_0(t) = y_0 = \text{constant} .$$

The requirement (1) is met (at least in the steady state) either for

$$(4) \quad u_0(t) = u(t - 1)$$

or for

$$(5) \quad u_0(t) = u_0 = \text{suitable constant} .$$

The choice of  $u_0$  will be treated later.

*Model following* – the future values of command signal are unknown but their evolution is described by a causal model which predicts future reference values. The most practical cases can be covered by one-step-ahead predictors  $\hat{y}_0(t | t - 1)$ ,  $\hat{u}_0(t | t - 1)$  when assuming that the prediction uncertainty does not depend even indirectly on inputs. A common subcase is the

*positional servo problem*, for which

$$(6) \quad \hat{y}_0(t | t - 1) = y_0(t - 1) .$$

The process (6), called *generalized random walk*, describes reasonably the situation when the next command signal is expected, but not guaranteed, to be unchanged. The requirement (1) is suitably satisfied when

$$(7) \quad \hat{u}_0(t | t - 1) = u(t - 1) .$$

### 2.5.2. Admissible action range

Limitations given by safety, technology, economy etc. specify admissible action range. Numerical values of  $u(t)$  have to belong to some prespecified set  $\mathcal{U}(t | t - 1)$ . For a wide class of problems the sets  $\mathcal{U}(t | t - 1)$  take the form

$$(8) \quad \mathcal{U}(t | t - 1) = \{u \cdot u_l(t | t - 1) \leq u \leq u_u(t | t - 1)\}$$

where the lower and upper bounds  $u_l, u_u$  are given functions of information available up to and including time  $t - 1$ . The inequalities are understood entrywise. A typical example is the restriction on the speed of the input changes, when for the given vector  $\delta_u$  and sampling period  $T_s$

$$(9) \quad -\delta_u \leq (u(t) - u(t - 1))/T_s \leq \delta_u, \quad \text{i.e.} \\ u_l(t | t - 1) = u(t - 1) - \delta_u T_s, \quad u_u(t | t - 1) = u(t - 1) + \delta_u T_s .$$

### 2.5.3. Admissible complexity

The mapping which describes the control strategy is evaluated on a computer with finite computational speed and finite memory within finite sampling period. The complexity of the controller cannot be usually introduced directly into design procedure. A concession from the realistic problem formulation has to be made. The achieved compromise may then be decisive from the point of view of efficiency and technical reliability of the resulting solution.

### 2.6. Quantification of the desired closed loop behaviour

An algorithmic control design requires to quantify user's idea of the desired closed-loop behaviour. The most direct way, in terms of measurable signals, will be chosen here. Within a planning period determined by some horizon  $N$ , the "ideal" input-output values  $u_0(1 \dots N)$ ,  $y_0(1 \dots N)$  are specified by the user. Then he has to assign to any possible  $u(1 \dots N)$ ,  $y(1 \dots N)$  a scalar loss  $J(1, N)$  measuring the distance of these signals from the ideal values. The input-output behaviour is to be felt the better the lesser is the corresponding loss. As usual, differences between actual and reference values of input and output are introduced and the distance of such a global error vector  $e_\theta$  from zero is measured. The introduced output difference

$$(10) \quad e_y(t) = y(t) - y_0(t) \quad t = 1, 2, 3, \dots, N,$$

is called control error and the input difference

$$(11) \quad e_u(t) = u(t) - u_0(t) \quad t = 1, 2, 3, \dots, N,$$

control effort. The following arrangement is advantageous for algorithmization

$$(12) \quad e'_\theta = (e'_y(N), e'_y(N), \dots, e'_y(1), e'_u(1)).$$

A sensible loss function  $J(\cdot)$ , defined by assigning to any  $e_\theta$  the loss  $J(1, N) = \tilde{J}(e_\theta) = \tilde{J}(d(1 \dots N))$  has to achieve its minimum for  $e_\theta = 0$ . Any sufficiently smooth loss function  $\tilde{J}(\cdot)$  can be approximated (near this minimum) by a quadratic form

$$(13) \quad \tilde{J}(e_\theta) \approx 1/N \ e'_\theta \tilde{Q}_\theta e_\theta$$

with a positive semidefinite kernel (penalization matrix)  $\tilde{Q}_\theta \geq 0$ .

In spite of the fact that the approximation (13) is local in nature, telling nothing about the true loss function when some entry of  $e_\theta$  is very large, we shall restrict ourselves to quadratic loss functions. The decisive reason for this is an acceptable complexity of the resulting adaptive controller. The same reason leads us to assume only block diagonal quadratic forms with the kernel

$$(14) \quad Q_\theta = \text{block-diag} [Q_y(N), Q_u(N), \dots, Q_y(1), Q_u(1)].$$

This is not too restrictive assumption because any  $\tilde{Q}_\theta \geq 0$  can be dominated by

a block-diagonal  $Q_g$  in the sense

$$(15) \quad e_g' \bar{Q}_g e_g \leq e_g' Q_g e_g \quad (\text{abbreviated as } \bar{Q}_g \leq Q_g).$$

For suboptimal control strategies presented later it is useful to scale the data in such a way that the blocks in  $Q_g$  (14) are time-invariant except of a few initial entries, i.e. the most general assumed form of  $Q_g$  will be

$$(16) \quad Q_g = \text{block-diag} [Q_y + Q_{sy}, Q_u + Q_{su}, \dots, Q_y, Q_u]$$

where penalties  $Q_{sy}(Q_{su})$  are added to  $Q_y(Q_u)$  at most  $l_y(l_u + t_u)$  times. Integers  $l_y(l_u + t_u)$  are related to the system description given in 2.8, fixed even for horizon  $N$  growing to infinity. Additional penalties  $Q_{sy}, Q_{su}$  reflect the requirement for closed loop stability (see 2.6.2).

The quadratic loss (13) with the kernel (16) can be rewritten in the more usual form

$$(17) \quad J(e_g) = J(d(1 \dots N)) = J(1, N) = 1/N \left[ \sum_{t=1}^N (g_y(t) + q_u(t)) + J_s \right]$$

with the stabilizing part

$$(18) \quad J_s = \sum_{t=N-l_y+1}^N g_{sy}(t) + \sum_{t=N-l_u-t_u+1}^N q_{su}(t)$$

and partial quadratic losses

$$(19) \quad \begin{aligned} q_y(t) &= e_y'(t) Q_y e_y(t), & q_u(t) &= e_u'(t) Q_u e_u(t), \\ q_{sy}(t) &= e_y'(t) Q_{sy} e_y(t), & q_{su}(t) &= e_u'(t) Q_{su} e_u(t). \end{aligned}$$

The multi-stage quadratic loss (17), (18) which has been related to sufficiently smooth loss is used in our adaptive controller and its characteristics form fundamental user's "adjustment knobs".

### 2.6.1. Choice of penalization matrices and scaling

The numerical choice of penalization matrices  $Q_y, Q_u$  expresses user's preferential ordering among closed loop behaviours. It is by no means a trivial task to evaluate the preference structure. The following notions [3] might be helpful to perform this task.

- Indifference surfaces are defined as sets of global errors for which the loss function is constant. They form multivariate ellipsoids in the quadratic case considered. The penalization matrices have to be chosen in such a way that the global errors lying on the same indifference surface are felt approximately equivalent. For example, in single input, single output case with horizon  $N = 1$ , for  $Q_y = 1, Q_u = 0.5$  the following pairs  $(e_y, e_u)$  are preferentially equivalent ( $\sim$ ) on the indifference curve  $J(1, 1) = 1$ :  $(e_y, e_u) = (1, 0) \sim (0, \sqrt{2}) \sim (1/\sqrt{2}, 1) \sim (\sqrt{3}, \sqrt{3})$ .
- Local substitution rate of  $i$ th and  $j$ th entry of the global error  $e_g$  at the point  $\bar{e}_g$

is defined, if it makes sense, as

$$(20) \quad \kappa_{ij} = \frac{\partial J(1, N)}{\partial e_{gi}} \bigg/ \frac{\partial J(1, N)}{\partial e_{gj}}, \quad \text{derivatives taken at } \bar{e}_g.$$

This rate expresses how many units of  $e_{gi}$  we are willing to “pay” for a unit of  $e_{gj}$ . In the assumed quadratic case  $\kappa_{ij}$  depends on the value  $\bar{e}_g$ . The simplest dependence is achieved for fully diagonal  $Q_g$  when

$$(21) \quad \kappa_{ij} = Q_{gii} \bar{e}_{gj} / (Q_{gjj} \bar{e}_{gj}).$$

Proper data scaling is the general tool to achieve simple situations in which we are able to realize our own willingness-to-pay. For example, such scales should be chosen that a fully diagonal  $Q_g$  is sufficient to express our preference ordering. Moreover, consideration of relative errors with respect to some nominal global error is probably most useful. To scale properly it is worthwhile to observe that

- only the ratio  $Q_{gii} / Q_{gjj}$  is important; some  $Q_{gkk}$  can be arbitrarily normalized,
- the local substitution rate (21) depends just on the  $i$ th and  $j$ th entry of  $\bar{e}_g$ ,
- preliminary filtering of data can be used to cover the case when the importance of errors varies with time, guaranteeing time-invariant values of  $Q_y, Q_u$  (required for strategies used),
- when (in the multivariate case) coincidence of some errors at the same time instant is more dangerous than their separate occurrence in similar magnitude, then the decoupling required for use of diagonal  $Q_y, Q_u$  must be achieved through a suitable rotation of input and/or output spaces.

### 2.6.2. Stabilizing term

The whole control loop is required to be stable: each of the signals in the closed loop has to be bounded. The role of  $Q_{sy}$  and  $Q_{su}$  is to reflect this requirement, positively penalizing any directly or indirectly manipulated variable which is not penalized by  $Q_y$  and  $Q_u$  ( $y_d$  and unpenalized inputs).

For a small (finite) horizon the rules of preceding paragraphs apply to choosing numerical values of  $Q_{sy}, Q_{su}$ . For a large (infinite) horizon particular values of  $Q_{sy}, Q_{su}$  are not theoretically important: when the chosen strategy keeps bounded  $y_d$  and inputs unpenalized by  $Q_u$ , the stabilizing term falls to zero because of finiteness of  $l_y, l_u + t_u$  (the factor  $1/N$  goes to zero). In the opposite case unstable modes grow so quickly that the mere positivity of penalties is sufficient for detecting instability.

This informal discussion demonstrates, that the strategy minimizing the assumed loss (for  $N \rightarrow \infty$ ) stabilizes the closed loop. Rigorous proof of this fact is given in [4] where it is also shown that  $l_y, l_u + t_u$  (related to the structure of regression model, see Section 2.8.3) are generically smallest possible integers to guarantee stability.

Stabilizing property of the optimal strategy is guaranteed for  $N \rightarrow \infty$  only. Avail-

able computing time forces us to use finite  $N$ . The value of  $N$  for which the closed loop stability is achieved depends on the controlled system. The stability is guaranteed the sooner the higher numerical values of penalties  $Q_{sys}$ ,  $Q_{su}$  are.

### 2.6.3. Horizon and problem of nonminimum phase system

Controlled systems are usually dynamic, the current input influences the whole future behaviour of the system. Therefore the activity of the controller has to be planned, its strategy has to optimize a multistage criterion.

The following example demonstrates what happens when dynamic character of the controlled system is neglected. The class of first order systems will be examined, described by

$$y(t) = 0.9y(t-1) + b_0 u(t) + b_1 u(t-1) + \varepsilon(t)$$

where  $\{\varepsilon(t)\}$  is a sequence of unmeasurable disturbances for which

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \varepsilon(t) = 0, \quad \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t) = 1.$$

The values  $b_0$ ,  $b_1$  are chosen so that the static gain of the system is equal to unity and different systems are distinguished by the value  $\lambda = -b_1/b_0$ . Notice that this parametrization (by the system zero) makes sense for  $\lambda \neq 1$ . The stationary values of  $J_x(1, N) = (1/N) \sum_{t=1}^N x^2(t)$ ,  $x = y, u$  can be seen in Fig. 5 for

- the optimal strategy minimizing  $J_y(1, \infty) + 0.5J_u(1, \infty)$ ,
- the one-stage-ahead strategy (still often recommended in current literature) minimizing separately  $J_y(t, t) + 0.5J_u(t, t)$  for each  $t = 1, 2, 3, \dots$ , (i.e. the strategy with receding planning interval, cf. Fig. 4, reduced to one stage only).

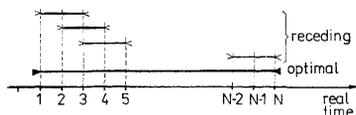


Fig. 4. Planning horizon.

Results presented in Fig. 5 illustrate generally valid observations concerning one-stage-ahead and related strategies:

- For the given loss function stabilizable systems always exist for which closed loop is unstable. Typically, when inputs are not penalized, every nonminimum phase system (having an unstable zero) controlled by one-stage-ahead strategy becomes unstable. Non-zero input penalty makes the set of unstabilized systems

smaller but does not make it empty (in our case this set is given by  $\lambda \in (1, 1.2)$ , decreasing the weight of  $J_u$  to 0.05 it increases to  $\lambda \in (1, 3)$ ). This fact is especially important in the context of adaptation because parameter values are not known beforehand. Moreover the sampled systems are of nonminimum phase rather

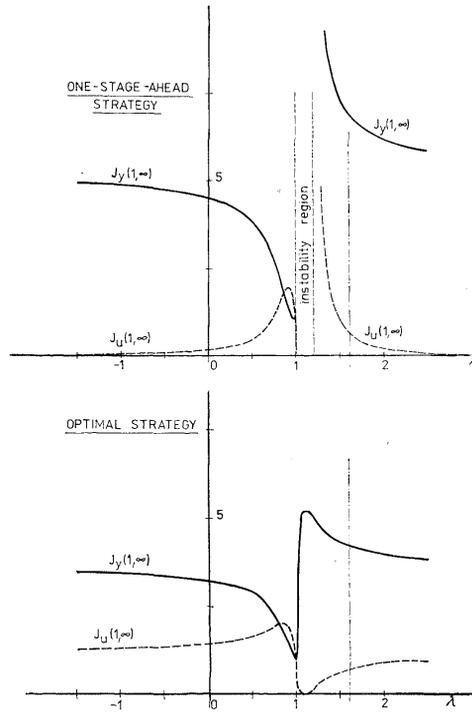


Fig. 5. Comparison of one-stage ahead and optimal strategies.

often. Theoretical treatment of this fact in the case of synchronous sampling can be found in [5].

- The performance of multi-stage strategy can be substantially better than that of one-stage-ahead strategy. Sometimes better output behaviour is achieved with lesser input effort ( $\lambda \in (1.2, 1.56)$  in Fig. 5) or optimally used control effort improves output behavior substantially ( $\lambda \approx 0$ , minimum phase systems!).

#### 2.6.4. Continuous extension of strategies

The above discussion about superiority of long planning horizons together with the fact that no finite horizon guarantees closed loop stability might give the impression that general problem formulation with  $N = \infty$  is the most advantageous. However, the asymptotic value of the loss function  $J(1, \infty) = \lim_{N \rightarrow \infty} J(1, N)$  is insensitive to any finite change of any finite constituent of current losses  $q_y, q_u$  ( $J(k, \infty) = J(1, \infty)$  for any finite  $k$ ). The direct minimization of  $J(1, \infty)$  implies that

- the possibility to guarantee automatically the closed loop stability through the stabilizing term is lost,
- the way in which particular strategy handles the transient behaviour cannot be distinguished. The consequence of this fact is demonstrated in the following example:

Let the controlled system be described by a static model

$$y(t) = \beta u(t), \quad \beta \neq 0$$

and the set point

$$y_0(t | 0) \begin{cases} \neq 0 & \text{for } t \leq T_1 \text{ for some finite } T_1 > 0 \\ = 0 & \text{for } t > T_1 \end{cases}$$

then for  $Q_y = 1, Q_u = 0$  the strategies  $u(t) = y_0(t | 0)/\beta$  and  $u(t) = 0$  are  $J(1, \infty)$  equivalent. Apparently, only the first one is of practical use. It can be verified that this strategy is chosen when optimization is performed for  $N < \infty$  and stationary strategy is found as limit for  $N \rightarrow \infty$ . This approach, called continuous extension of finite-horizon strategies, will be used in the sequel, because it guarantees the appropriate handling of transients.

#### 2.7. Ordering of control strategies and stochastic systems

The controller is expected to optimize the closed loop behaviour. The loss function can be used to evaluate the quality of different closed-loop responses, but generally it is insufficient to compare control strategies a priori in the design stage: ever-present random influences and/or usual incomplete knowledge of the system description admit to evaluate the loss ex post only. The problem is attacked in this part in the way sketched below.

Firstly, the so-called dominated strategies are defined, which have to be excluded by any reasonable design procedure. Then the class of non-dominated strategies is characterized through a suitably chosen optimization procedure, which is parametrized by a positive weighting function  $p(\cdot)$ . An attempt to choose among non-dominated strategies the strategy which is near to "ideal" one leads to the conclusion

that the function  $p(\cdot)$  must have properties of a probability density function (p.d.f.). Hence, the functional used for the purpose of optimization is the well known expected value of the cost function. The interpretation of  $p(\cdot)$ , given in such a way, makes it possible for the user to assign suitable weights both to possible random outcomes and to incompletely known characteristics of the system. The latter feature is of extraordinary importance in adaptive control. Probability can be assigned to constant unknown quantities and it also yields consistent and definite rules how to extract the relevant information from measured data. No operational difference between randomness and incomplete knowledge appears, at least for controller design, therefore no formal distinction will be made between them. The system exhibiting at least one of these features will be called stochastic. It should be emphasized that even a seemingly deterministic system has to be considered as stochastic if some of its parameters or its initial state are not known a priori.

In the following exposition all technicalities are intentionally suppressed in order to prevent the corner stones of the reasoning be hidden by the apparatus of optimization and measure theory. A rigorous treatment of these finesses can be found elsewhere [6, 7]. The presented derivations are practically valid without any additional assumption for cases when treated quantities may take only a finite number of discrete values. It suffices to replace probability density functions by probability functions (p.f.) and integrals by sums.

### 2.7.1. Dominated strategies

We shall assume that the attitude of users to a possible loss is not influenced by stochastic nature of the system, they are neither risk-averse nor risk-prone. This assumption is made because of computational reasons: however, it can be often accepted for technological applications.

A particular control strategy is described by the mapping

$$(22) \quad L: t, d(1 \dots t-1) \rightarrow u(t) \in \mathcal{U}(t | t-1), \quad t = 1, 2, \dots, N.$$

Introducing common symbol  $\omega$  for all unmeasured influences and unknown factors, the value of data is uniquely determined by  $L(\cdot)$  and  $\omega$ , i.e.  $d(1 \dots N) = d(1 \dots N, \omega, L(\cdot))$ . The set of all possible  $\omega$  will be denoted by  $\Omega$ . Then

$$(23) \quad J(1, N) = J(d(1 \dots N, \omega, L(\cdot))) = j(\omega, L(\cdot)).$$

The function  $j(\cdot)$  in (23) exemplifies dependency of the loss on  $\omega$  which collects all stochastic constituents related to the system.

We shall define the following partial ordering among the admissible strategies: The strategy  $L_1(\cdot)$  dominates the strategy  $L_2(\cdot)$  if

$$(24) \quad j(\omega, L_1) \leq j(\omega, L_2) \quad \text{for every } \omega \in \Omega$$

and there exists some  $\omega \in \Omega$  for which this inequality is strict. Even if there is a lot of strategies for which no such relation is valid on  $\Omega$ , the ordering clearly indicates dominated strategies are to be excluded by any reasonable design method.

The non-dominated strategies can be characterized by the following optimization procedure. Let  $p(\omega)$  be any positive function on  $\Omega$  and let

$$(25) \quad L^*(\cdot) = \arg \min_{\text{admissible } L} \int j(\omega, L(\cdot)) p(\omega) d\omega .$$

Such an  $L^*(\cdot)$  can be directly verified (by contradiction) not to be dominated. Returning from  $\omega$  to  $d(1 \dots N)$ , the functional reduces, up to a normalizing factor, to expected value. Thus the strategy minimizing the criterion

$$(26) \quad K_0(1, N) = \int J(d(1 \dots N)) p(d(1 \dots N)) dd(1 \dots N)$$

for any (!) positive  $p(d(1 \dots N))$  is not dominated. This criterion makes all strategies a priori comparable. However, this prior ordering depends heavily on the weight  $p(\cdot)$  chosen. To obtain a strategy near the "ideal" one that minimizes the loss, the weight should reflect the degree at which different possible outcomes are expected to occur. Such a weight will be discussed in the next paragraph.

**Remarks.** (i) The lesser is the uncertainty about possible  $\omega$ , the greater is the chance to choose a proper  $p(\omega)$  and consequently a proper strategy. Thus all important variables influencing the system behaviour are to be at least indirectly measured and any prior information at disposal should be exploited in order to make the system as deterministic as possible.

(ii) The function  $p(\omega)$  in (25) has to be strategy-independent to guarantee that the dominated strategies be removed. As  $p(\omega)$  is to reflect the true situation, good quality requires as much independence of unmeasured disturbances and the chosen strategy as possible. Practical implications coincide with those of the remark (i).

### 2.7.2. Probability for control design

The seemingly arbitrary positive weighting function in (25), (26) proves to be probability density function whenever it is required to reflect the stochastic nature of the system. This essentially coincides with the requirement to find  $p(\omega)$  independent of the particular loss function used.

Properties of  $p(\omega)$  found by treating the criterion (25), (26) with some particular loss function must be valid generally. Loss functions which are zero outside a properly chosen subset  $\mathcal{A} \subset \Omega$  suit this purpose.

The weight  $p(\omega)$  is, by definition, strategy independent. The dependence of the loss function on  $L(\cdot)$  is irrelevant in the sequel, therefore  $L(\cdot)$  is omitted in  $j(\cdot)$ .

The following fundamental steps will be considered in our reasoning:

*Normalization.* Consider the case when  $j(\omega) = j_n = \text{constant}$  independent of  $\omega$ .

Then, the stochastic nature of the system plays no role in the control design: the criterion should coincide with the loss evaluated ex post, i.e.

$$(27) \quad K_0(1, N) = \int_{\Omega} j(\omega) p(\omega) d\omega = \int_{\Omega} p(\omega) d\omega = j_{\Omega}$$

which is equivalent to

$$(28) \quad \int_{\Omega} p(\omega) d\omega = 1.$$

It can be proved [8, 9] that the positive  $p(\omega)$  fulfilling (28) is the probability density function (p.d.f.) corresponding to probability  $P(\cdot)$  defined as the set function

$$(29) \quad P(\mathcal{A}) = \int_{\mathcal{A}} p(\omega) d\omega$$

for any subset for which (29) makes sense.

Particularly, substituting  $d(1 \dots N)$  for  $\omega$  the criterion (26) becomes expected value of the loss function

$$(26') \quad K_0(1, N) = \int j(\omega) p(\omega) d\omega = \int J(d(1 \dots N)) p(d(1 \dots N)) dd(1 \dots N) = E[J(1, N)]$$

and the rule (25) takes the form

$$(30) \quad L^*(\cdot) = \underset{\text{admissible } L(\cdot)}{\arg \min} E[J(1, N)].$$

Referring to any standard textbook we could just list formulae necessary for our treatment because they are implied by proving  $p(\omega)$  to be p.d.f. However, we believe that a discussion of relations between joint, conditional and marginal probabilities from the point of view of control design may provide a useful insight. The reader who feels the topic too far out of his interests can take the rest of the paragraph as a review of necessary formulae.

The properties treated below concern the case of a compound  $\omega$ , formed by pairs  $\omega = (r, c)$ ,  $r \in \mathcal{R}$ ,  $c \in \mathcal{C}$ .

*Conditional and marginal probabilities.* Consider the loss  $j(\omega) = j(r, c)$  which is zero whenever  $c$ -component of  $\omega$  is outside some fixed set  $\mathcal{A}_c \subset \mathcal{C}$ . A necessary description of "stochasticity" thus can be reduced to the set  $\omega = (r, c)$  having  $c \in \mathcal{A}_c$ . Noting that any change of  $\omega$ -range changes the description of stochasticity, a new, now conditional probability denoted by  $P(\mathcal{A}_r | \mathcal{A}_c)$  for any  $\mathcal{A}_r \subset \mathcal{R}$  is to be defined. Only such a redefinition of the probability is acceptable which saves the original strategy ordering for the assumed type of the loss function. This reduces possible  $P(\mathcal{A}_r | \mathcal{A}_c)$  to those proportional to  $P(\mathcal{A}_r, \mathcal{A}_c)$ . Normalization (29) implies that

$$(31) \quad P(\mathcal{A}_r | \mathcal{A}_c) = P(\mathcal{A}_r, \mathcal{A}_c) / P(\mathcal{R}, \mathcal{A}_c).$$

The two special cases of  $\mathcal{A}_c = \mathcal{C}$  and  $\mathcal{A}_c = c$  will be considered. It is worthwhile to introduce marginal probability  $P(\mathcal{A}_R)$  with p.d.f.  $p(r)$  which fits to situations when no extra information about  $c$ -part of  $\omega$  is available, i.e.  $\mathcal{A}_c = \mathcal{C}$ . The identity (31) becomes

$$(32) \quad P(\mathcal{A}_R) = P(\mathcal{A}_R | \mathcal{C}) = P(\mathcal{A}_R, \mathcal{C}) / P(\mathcal{R}, \mathcal{C}) = P(\mathcal{A}_R, \mathcal{C})$$

where the last equality follows from (29) for  $\Omega = (\mathcal{R}, \mathcal{C})$ . Rewriting (32) in terms of p.d.f.'s we have

$$(33) \quad \int_{\mathcal{A}_R} p(r) dr = \int_{\mathcal{A}_R} \int_{\mathcal{C}} p(r, c) dr dc .$$

This equality is valid for any  $\mathcal{A}_R \subset \mathcal{R}$ , consequently,

$$(34) \quad p(r) = \int_{\mathcal{C}} p(r, c) dc .$$

On the other hand, knowing the exact value of  $c$ -part of  $\omega$  then for conditional ( $c$ .) p.d.f.  $p(r | c)$  of the probability  $P(\mathcal{A}_R | \mathcal{C})$  we have

$$(35) \quad p(r | c) = \frac{p(r, c)}{\int p(r, c) dr} = \frac{p(r, c)}{p(c)}$$

using (31), arbitrariness of  $\mathcal{A}_R$  and the relation (34) in which role of  $r$  and  $c$  has been interchanged. The formula (35) is often rewritten as the

*Chain rule* (called also formula of complete probability)

$$(36) \quad p(r, c) = p(r | c) p(c) .$$

Equating the two possible versions of (36)  $p(r | c) p(c) = p(c | r) p(r)$  and using (35) we arrive at

*Bayes rule*

$$(37) \quad p(c | r) = \frac{p(r | c) p(c)}{\int p(r | c) p(c) dc} .$$

This fundamental rule can be viewed as a general "machine" for experience accumulation. It describes the way how observation of  $r$  changes our prior uncertainty expressed by prior p.d.f.  $p(c)$  to posterior one  $p(c | r)$ . A model relating  $r$  to  $c$  ( $p(r | c)$ ) has to be, of course, at disposal.

**Remark.** All above rules can be inductively extended to any finite number of entries of  $\omega$ . Let, for example,  $r = (r_1, r_2)$ , then Bayes rule (37) takes the form

$$\begin{aligned} p(c | r_1, r_2) &= \frac{p(r_1, r_2 | c) p(c)}{\int p(r_1, r_2 | c) p(c) dc} = \frac{p(r_2 | r_1, c) p(r_1 | c) p(c)}{\int p(r_2 | r_1, c) p(r_1 | c) p(c) dc} = \\ &= \frac{p(r_2 | r_1, c) p(c | r_1)}{\int p(r_2 | r_1, c) p(c | r_1) dc} \end{aligned}$$

as it can be verified using the appropriate version of the chain and Bayes rules. It is worthwhile to note that the last equality above represents the recursive form of Bayes rule, the former posterior probability is used as the new prior probability.

### 2.7.3. Some properties of conditional expectation

To minimize the criterion (26) we shall need some formulae related to conditional expectation. They can be mostly verified by using relations for c.p.d.f. and elementary properties of (multiple) integrals.

For the conditional expectation  $E[J(x) | z] = \int J(x) p(x | z) dx$  it holds

$$(38) \quad E[\cdot | z] \text{ is a function of } z$$

$$(39) \quad E[J(x) | x, z] = J(x)$$

$$(40) \quad E[E[J(x) | y, z] | z] = E[J(x) | z]$$

$$(41) \quad E[J(x) | y, z, u(z)] = E[J(x) | y, z] \text{ for any function } u(z).$$

**Notational remark.** Marginal and conditional probabilities coincide whenever the condition brings no additional information, thus p.d.f., expectation etc. conditioned on empty list of variables can and will be understood as their marginal counter-parts.

## 2.8. Optimization and the need for modelling in the controller design

The optimization procedure will be outlined now and the model describing stochastic as well as deterministic (structural) features of the controlled system will be specified.

### 2.8.1. Dynamic programming

The strategy minimizing (26) must generate inputs dependent only on the available information ( $u(t) = u(t | t - 1)$ ). This "causality" feature is guaranteed by optimization procedure known as dynamic programming the main idea of which will be outlined now.

It can be proved [9, 10], [1.2] and we shall take it for granted that the optimal strategy even in the stochastic case can be described by deterministic functions

$$(22) \quad L(\cdot) : t, d(1 \dots t - 1) \rightarrow u(t) \in \mathcal{U}(t | t - 1) \quad t = 1, 2, \dots, N.$$

Let us suppose that the strategy (22) has been chosen for  $t = 1, 2, \dots, N - 1$  and some data  $d(1 \dots N - 1)$  measured. Then the standard minimization of the criterion on  $u^*(N) \in \mathcal{U}(N | N - 1)$  gives the causal optimal input  $u(N) = u(N | N - 1)$ . Performing this step for any possible past of the process history we shall find the optimal strategy for the last control step  $L^*(N, d(1 \dots N - 1))$ . The procedure can

be inductively continued by extending the planning interval. Formally, introducing the cost-to-go ( $1/N$  omitted) as

$$(42) \quad K(t, N | t-1) = \mathbb{E} \left[ \sum_{\tau=t}^N (q_y(\tau) + q_u(\tau)) \mid t-1 \right], \quad t = N, N-1, \dots, 1$$

and using (40), (41), (42) we obtain

$$(43) \quad K(t, N | t-1) = \mathbb{E} [q_y(t) + q_u(t) + K(t+1, N | t) \mid t-1; u(t | t-1)] \\ t = N, N-1, \dots, 1$$

starting from

$$(44) \quad K(N+1, N | N) = 0.$$

Performing the due optimization in every step of the recursion (43), (44), we can find the optimal cost-to-go

$$(45) \quad K^*(t, N | t-1) = \min_{\text{admissible } L(\cdot)} K(t, N | t-1),$$

for which the following analogy of the recursion (43), (44) is valid

$$(46) \quad K^*(t, N | t-1) = \min_{u(t) \in \mathcal{U}(t|t-1)} \mathbb{E} [q_y(t) + q_u(t) + K^*(t+1, N | t) \mid t-1; u(t)] \\ \text{for } t = N, N-1, \dots, 1$$

starting with

$$(47) \quad K^*(N+1, N | N) = 0.$$

The minimizing arguments in (46) form the optimal strategy (observe that not only the value of, but entire functions  $L(\cdot)$  are necessary) and minimal value of the criterion coincides with the last cost-to-go (up to the factor  $1/N$ )

$$(48) \quad K_0^*(1, N) = \frac{1}{N} K^*(1, N | 0).$$

### 2.8.2. Modelling and the need for adaptive control

The computation of the conditional expectation  $\mathbb{E}[\cdot \mid t-1; u(t)]$  and the minimization are repeatedly performed when constructing the optimal strategy. A model necessary to evaluate the required expectation is now specified and its identification is described.

The definition of the optimal cost-to-go (42), (45) and the property (38) imply that  $K^*(t+1, N | t)$  is a function of  $d(1..t)$ . Consequently, the expectation in (46) is taken over the extended output  $y_x(t)$  because  $d(1..t) = (d(1..t-1), d(t)) = (d(1..t-1), (u(t), y_x(t)))$  and the condition consists of  $d(1..t-1), u(t)$ . The ‘‘expectation step’’ of the dynamic programming requires to specify the corresponding c.p.d.f.’s

$$(49) \quad p(y_x(t) \mid t-1; u(t)) \quad \text{for } t = 1, 2, \dots, N.$$

Such a c.p.d.f. is called the predictive c.p.d.f. (of the extended output). It assigns probability to possible extended outputs  $y_x(t)$  when the entire observable past of the process history  $d(1..t-1)$  and current input  $u(t)$  are given. Recalling that the procedure of optimization can be performed only when determining the entire functions  $L^*(\cdot)$ , we can see that  $p(y_x(t) | t-1; u(t))$  has to be known as a function of both  $y_x(t)$  and  $d(1..t-1), u(t)$ . Such a degree of knowledge of the controlled system is, of course, exceptional in practice.

This is the point at which the need for adaptive controllers arises! To overcome user's uncertainty about appropriate choice of  $p(y_x(t) | t-1; u(t))$  the set of models  $\{\mu\}$  is introduced. User's (subjective) uncertainty becomes a part of the stochastic nature of the system and as such it will be treated.

A family of models  $\{\mu\}$ , varying slowly in time, determining

$$(50) \quad p(y_x(t) | t-1; u(t), \mu(t)) \quad \text{for } t = 1, 2, \dots, N$$

is selected. The controller is designed using the current state of knowledge about possible  $\mu$ . When model variations are slow enough, the lacking information about  $\mu$  is gained from data. Notice, that this approach makes it possible to use simplified models, because the system behaviour has to be fitted only around its actual working point!

The needed predictive c.p.d.f. (49) is related to "parametrized" c.p.d.f. (50) by the formula

$$(51) \quad p(y_x(t) | t-1; u(t)) = \int p(y_x(t) | t-1; u(t), \mu(t)) p(\mu(t) | t-1; u(t)) d\mu(t)$$

which follows directly from (34), (36).

The experience accumulation, described below, can be used for randomized input generators described by their predictive c.p.d.f.  $p(u(t) | t-1)$  determining the probability that the particular value  $u(t)$  will be chosen when  $d(1..t-1)$  has been observed. This possibility has proved to be useful when there is an opportunity to gain some information about the system in a preliminary identification experiment. The (optimal) deterministic strategies generating  $u(t) = L(t, d(1..t-1))$  fall in the above class, their predictive c.p.d.f.'s are concentrated just on the values given by  $L(\cdot)$ .

Adaptive controllers gain all information about uncertain model  $\mu$  from the observed data; thus so called "natural conditions of control" [1.2]

$$(52) \quad p(u(t) | t-1) = p(u(t) | t-1; \mu(t))$$

hold for them really "naturally". These conditions, however, are valid for a rather wide class of input generators, including open loop controllers, feedback controllers and manual control, whenever no additional information about  $\mu(t)$  has been supplied. Just such input generators will be assumed in the sequel.

Using the Bayes formula (37), the normalizing property (29) and the natural

conditions of control (52) we find that

$$(53) \quad p(\mu(t) | t-1; u(t)) = p(\mu(t) | t-1),$$

consequently,

$$(54) \quad p(y_s(t) | t-1; u(t)) = \int p(y_s(t) | t-1; u(t), \mu(t)) p(\mu(t) | t-1) d\mu(t).$$

The c.p.d.f.  $p(\mu(t) | t-1)$  describes how our uncertainty about  $\mu(t)$  is influenced by the data  $d(1 \dots t-1)$ . Under (52) the Bayes rule describes updating of this uncertainty by the new measurement

$$(55) \quad p'(\mu(t) | t) = \frac{p(y_s(t) | t-1; \mu(t)) p(\mu(t) | t-1)}{\int p(y_s(t) | t-1; \mu'(t)) p(\mu(t) | t-1) d\mu(t)}.$$

The recursion could be completed by taking into account the time dependence of models. Again, to achieve reasonable computational complexity we restrict ourselves to cases with slowly varying parameters for which  $\mu(t+1) \approx \mu(t)$ . The effective parameter tracking technique which approximates the time-updating  $p(\mu(t) | t) \rightarrow p(\mu(t+1) | t)$  for the used model will be described in Chapter 3.

The user, of course, has to supply the initial condition to recursion (55) by specifying his prior uncertainty about possible models

$$(56) \quad p(\mu(1) | 0) = p'(\mu(1)).$$

**Remarks.** (i) Input values, not entire control strategy, are required to identify a system model whenever (52) holds.

(ii) The recursive form of the Bayes rule (55) supports conceptually real time identification.

(iii) The one-shot identification, the direct determination of  $p(\mu(t+1) | t)$  from  $d(1 \dots t)$  and  $p(\mu(1) | 0)$ , is possible when a description of  $\mu(t)$  evolution in time is known. In particular, for the time-invariant model ( $\mu(t) = \mu(t+1) = \mu$ ) we find

$$(57) \quad p(\mu | t) = \frac{\prod_{\tau=1}^t p(y_s(\tau) | \tau-1; u(\tau), \mu) p(\mu)}{\int \prod_{\tau=1}^t p(y_s(\tau) | \tau-1; u(\tau), \mu) p(\mu) d\mu}.$$

### 2.8.3. Regression models

In the preceding part the need for a model has been shown. Widely applicable family of the so-called regression models will be introduced now. Taking into account the necessary characteristics required for control design with quadratic loss, assumptions are discussed under which the use of regression models is well substantiated. To be more concise we shall use the following notations:

– conditional expectation of  $x$

$$(58) \quad \hat{x}(t | t-1; u(t)) = E[x(t) | t-1; u(t)]$$

(the property (38) implies that  $\hat{x}(\cdot)$ , called regression function, is a function of the data  $d(1 \dots t-1), u(t)$ ),

– conditional cross-covariance matrix of  $x$  and  $z$

$$(59) \quad \begin{aligned} \hat{R}_{xz}(t | t-1; u(t)) &= \\ &= E[(x(t) - \hat{x}(t | t-1; u(t)))(z(t) - \hat{z}(t | t-1; u(t)))' | t-1; u(t)], \end{aligned}$$

– conditional covariance matrix of  $x$

$$(60) \quad \hat{R}_x(t | t-1; u(t)) = \hat{R}_{xx}(t | t-1; u(t)) = \text{cov}(x(t) | t-1; u(t)),$$

– the entire data item related to time  $t$

$$(61) \quad d'(t) = (u'(t), y'(t), u_0'(t), y_0'(t), v'(t)) = (u'(t), y_x'(t)).$$

The partial loss evaluated in the course of dynamic programming now may be written in the form

$$e_u'(t) Q_y e_y(t) + e_u(t) Q_u e_u(t) = d'(t) Q_d d(t),$$

where  $Q_d \geq 0$  has been introduced:

$$(62) \quad Q_d = \begin{bmatrix} Q_u & 0 & -Q_u & 0 & 0 \\ 0 & Q_y & 0 & -Q_y & 0 \\ -Q_u & 0 & Q_u & 0 & 0 \\ 0 & -Q_y & 0 & Q_y & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \left. \begin{array}{l} \} m_u \\ \} m_y \\ \} m_u \\ \} m_y \\ \} m_v \end{array} \right\} i_d$$

$m_x \dots$  dimension of  $x$  for  $x = u, y, v; i_z \dots$  dimension of  $z$  for  $z$  compound.

Using the definitions of  $d, \hat{R}$  and the matrix trace  $\text{tr}(\cdot)$ , it can be verified (by writing entrywise) that

$$(63) \quad \begin{aligned} E[d'(t) Q_d d(t) | t-1; u(t)] &= \hat{d}'(t | t-1; u(t)) Q_d \hat{d}(t | t-1; u(t)) + \\ &+ \text{tr}(Q_d \hat{R}_d(t | t-1; u(t))). \end{aligned}$$

The cost-to-go in (46) will be proved to have quadratic form adding no requirement on our knowledge of moments, only the advantage of sparsity of matrix  $Q_d$  will be lost and all elements of regression function  $\hat{d}$  and covariance matrix  $\hat{R}_d$  are generally needed. The relations (39), (58) and (59) imply

$$(64) \quad \hat{u}(t | t-1; u(t)) = u(t), \quad \hat{R}_{uu}(t | t-1; u(t)) = 0.$$

Thus we can conclude that the knowledge of  $\hat{y}_x$  and  $\hat{R}_{yx}$  is necessary and sufficient to solve the quadratic design problem.

The description of reference values is mostly given by the user. In the opposite case the reference values can be joined either with measured disturbances when

unrelated with other variables or with auxiliary output. Thus without loss of applicability, conditional moments of  $y$  and  $v$  are required. They are rarely known in practice and the lacking information has to be gained from measurements. Fulfilment of this requirement is fundamental for the choice of a convenient model. As the necessary assumptions about  $y$  and  $v$  differ, we shall treat them separately.

For most of real systems  $y(t)$  is weakly influenced by older data. Then there is a finite usually small  $l$  such that

$$(65) \quad \hat{y}(t | t-1; u(t)) \approx r(d(t-l..t-1), u(t)).$$

The function  $r(\cdot)$  can be rather often linearized around the actual operating point  $\gamma(t) = (d(t-l..t-1), u'(t))$ , i.e.

$$(66) \quad \hat{y}(t | t-1; u(t)) \approx r(d(t-l..t-1), u(t)) \approx \\ \approx \sum_{i=1}^{l_y} A_i(\gamma(t)) y(t-i) + \sum_{i=0}^{l_u} B_i(\gamma(t)) u(t-i-t_u) + \sum_{i=1}^{l_v} D_i(\gamma(t)) v(t-i) + c(\gamma(t))$$

where  $A_i, B_i, D_i$  are matrix coefficients of appropriate dimensions, the vector  $c$  is called absolute term, the structural indices are  $l_y, l_v, l_u + t_u \leq l$  and  $t_u \geq 0$  ( $t_u$  reflects possible transport delay).

We shall assume that the point  $\gamma(t)$  changes slowly with time, i.e. the coefficients of the best approximation in (66) vary also slowly. In the opposite case their dependence on  $\gamma$  has to be modelled for inherently nonlinear systems. Such an adaptive controller will not be pursued here, but the presented controller can be extended to it in relatively straightforward way.

To achieve an acceptable computational complexity we have to make one of our most peculiar assumptions: closed-loop-control conditions (disturbances, controller) guarantee that approximation errors made in (66) are negligible.

The level of disturbances can be reduced by proper design of the control system. However, only the practice can solve the "tautology": the approximation error required to be small enough is influenced by the controller designed under the assumption that this error is really small enough.

Under the assumptions made, there is a slowly varying model  $\mu(t)$ , determined partially by  $A_i, B_i, D_i, c, l_y, l_v, l_u, t_u$ , such that

$$(67) \quad \hat{y}(t | t-1; u(t), \mu(t)) = E[y(t) | t-1; u(t), \mu'(t)] = {}^k P' {}^k z(t)$$

where the right-hand side of (67) coincides with that of (66), i.e. the matrix of coefficients  ${}^k P'$  is

$$(68) \quad {}^k P' = [A_1, A_2, \dots, A_{l_y}, B_1, B_2, \dots, B_{l_u}, D_1, D_2, \dots, D_{l_v}, c]$$

and the regressor vector is  ${}^k z(t)$

$$(69) \quad {}^k z(t) = [y'(t-1), \dots, y'(t-l_y), u'(t-t_u), \dots, u'(t-l_u-t_u), \\ v'(t-1), \dots, v'(t-l_v), 1]$$

and where  $k$  is the integer vector of structural indices

$$(70) \quad k' = [l_y, l_u, l_v, l_w].$$

Let us continue our discussion taking into account the "trace" term in (63). The cross-covariances and covariances related to reference values are not usually influenced by input even indirectly; when the opposite is true we should incorporate the corresponding entries of  $y_0, u_0$  into  $y$ . Taking into account the definition of external disturbance we have

$$(71) \quad \text{tr} [Q_d \hat{R}_d(t | t-1; u(t))] = \text{tr} [Q_y \hat{R}_y(t | t-1; u(t))] + f(t)$$

where  $f(t)$  is a strategy-independent term which can be omitted when optimizing.

Our reasoning about  $\hat{R}_y$  will be based on inspection of the so-called (output) innovations defined as

$$(72) \quad e(t) = y(t) - \hat{y}(t | t-1; u(t), \mu(t)) \quad \text{for } t = 1, 2, 3, \dots, N.$$

They are zero mean, mutually uncorrelated and uncorrelated with  $d(1..t-1), u(t)$ . To see this, let  $i \geq 1$ , then

$$\begin{aligned} \text{E}[e(t) y'(t-i)] &= \text{E}[\text{E}(e(t) y'(t-i) | t-i; u(t-i+1), \mu(1..t))] = \\ &= \text{E}\{\text{E}[(y(t) - \hat{y}(t | t-1; u(t), \mu(t))) | t-i; u(t-i+1), \mu(1..t)] y'(t-i)\} = \\ &= 0 \end{aligned}$$

where (39), (40), (67) and (72) have been used. The rest can be proved similarly. Moreover, it can be verified that for any "causal" model-based output predictor say

$$\hat{y}(t | t-1; u(t), \mu(t)) = \hat{y}(d(1..t-1), u(t), \mu(t)), \quad \text{it holds}$$

$$\begin{aligned} \text{E}[(y'(t) - \hat{y}(t | t-1; u(t), \mu(t))) (y'(t) - \hat{y}(t | t-1; u(t), \mu(t)))' | t-1; u(t), \mu(t)] &\geq \\ (73) \quad &\geq \hat{R}_e(t | t-1; u(t), \mu(t)) = \hat{R}_y(t | t-1; u(t), \mu(t)) \end{aligned}$$

(recall that inequality  $\hat{R} \geq \tilde{R}$  means  $x' \hat{R} x \geq x' \tilde{R} x$  for any  $x$ ). This inequality interprets  $\hat{y}(t | t-1; u(t), \mu(t))$  as the best possible causal, model-based predictor of the output  $y(t)$ , innovations being the unpredictable part of the output due to the random nature of the system. Note, that errors caused by user's uncertainty are not present,  $\mu(t)$  is in the condition. As discussed before, the inevitable prerequisite for successful control is the measurement, at least indirect, of all substantial influences of the environment on the controlled system. Then these unmeasurable disturbances, condensed in the innovations, act on the system in (almost) chaotic and, on average, weakly time-dependent manner. Thus, it is reasonable to assume that

$$(74) \quad \hat{R}_e(t | t-1; u(t), \mu(t)) = \hat{R}_y(t | t-1; u(t), \mu(t)) = {}^k R(t) \approx \text{constant}.$$

The innovations sum up larger number of small random influences; consequently, invoking famous central limit theorem [8, 9], their conditional distribution can be

taken as (approximately) normal, i.e.

$$(75) \quad P(\hat{v}(t) | t-1; u(t), \mu(t)) = N(0, {}^kR(t))$$

with the notation

$$(76) \quad p(x) = N(\bar{x}, R_x) = |2\pi R_x|^{-1/2} \exp \left\{ -\frac{1}{2}(x - \bar{x})' R_x^{-1} (x - \bar{x}) \right\}.$$

The definition (72) and the assumptions (67), (74), (75) imply that

$$(77) \quad p(y(t) | t-1; u(t), \mu(t)) = N({}^kP'(t) {}^kz(t), {}^kR(t))$$

where  ${}^kP(t)$ ,  ${}^kR(t)$  are weakly time-dependent as needed for the model identification within reasonable computational complexity (see Chapter 3 and cf. the preceding paragraph).

A model of measurable disturbances is to be specified to complete the discussion. It is possible to follow step by step the way used for the output. However, the assumptions which guarantee innovations to be (almost) normal with constant covariance matrix, are rather often too restrictive. It is preferable to take into account that for a controller design just  $\hat{v}(\cdot)$  is needed, cf. (63), (71). Constancy of  $\hat{R}_v$  and normality is necessary only in the identification step. Choosing some fixed, even if somewhat simplified, model generating  $\hat{v}(\cdot)$ , we make often a smaller error in modelling than by defending assumptions of the type (75) with almost constant covariance. In most cases a generalized random walk modelling evolution of  $v(t)$  can be used. It takes the form

$$(78) \quad E[v(t) | t-1; u(t), \mu(t)] = E[v(t) | v(t-1)] = v(t-1)$$

$$(79) \quad \hat{R}_v(t | t-1; u(t), \mu(t)) = \text{any function of any variable uninfluenced by the controlled system.}$$

The richness of this model can be seen from the particular cases:

- constant  $v(t)$ ,  $\hat{R}_v = 0$
- slow drifts,  $\hat{R}_v$  is a small constant
- random steps of random height,  $\hat{R}_v$  takes on different nonzero values at rare random time instants,  $\hat{R}_v = 0$  at the rest.

Of course, mixed cases are also covered.

To summarize, the model  $\mu(t)$  substantiated in this part, determines the c.p.d.f.  $p(y(t) | t-1; u(t), \mu(t))$  to be normal with conditional expectation given by (67) and covariance matrix  ${}^kR$ . The model is parametrized by the slowly varying matrix of coefficients (68), by the structural indices  $k$  (70) and the covariance matrix  ${}^kR(t)$ . The external measurable disturbance is either described as analogy of (77) [ $p(v(t) | t-1; u(t), \mu(t)) = N({}^{k_v}P'_v(t) {}^{k_v}z_v(t), {}^{k_v}R_v(t))$ ] or by the fixed model generating  $\hat{v}(t | t-1; u(t))$ , which, of course, reduces to  $\hat{v}(t | v(1..t-1))$ . The most restrictive assumption is the possibility to neglect approximation errors in (66).

**Remarks.** (i) The derived linear regression model is often written in the form implied by (67), (72)

$$(80) \quad y(t) = {}^k P' {}^k z(t) + \varepsilon(t)$$

where the random term  $\varepsilon(t)$ , called white noise, has the properties of innovations.

(ii) The normality assumption (75) is essential only for constructing an appropriate identification procedure (presented later), which has proved to be weakly dependent on this assumption unless rare large values of  $\varepsilon(t)$  occur.

(iii) In many cases closed loop signals contain drifts, which could be modelled by an absolute term that changes its value more quickly than the rest of coefficients. In this case it is worthwhile to use the incremental version of regression model [2]:

$$(81) \quad y(t) = y(t-1) + \sum_{i=1}^{I_y} A_i \Delta y(t-1) + \sum_{i=0}^{I_u} B_i \Delta u(t-i-t_u) + \sum_{i=1}^{I_v} D_i \Delta v(t-i) + \varepsilon(t)$$

where  $\Delta y(t) = y(t) - y(t-1)$ , and similarly for  $u$  and  $v$ .

(iv) A more general and wide-spread model, known as ARMAX, could be substantiated in a similar way, by using more complex model for evaluation of  $\hat{y}_x$ . This line has not been followed because there is no satisfactory procedure to identify it. The difficulties met when developing such an algorithm can be shown to be inherent to this model.

(v) External measurable disturbances play in the regressor vector (69) the same role as inputs do (with unit delay) and nonzero weights are added in the dynamic programming to  $\hat{v}(t | t-1)$  only through the quadratic form representing the cost-to-go. Thus modelling of  $v(t)$  evolution becomes important only when using a multi-stage criterion. These are probably the reasons why  $v(t)$  is often (incorrectly) taken as "nonmanipulated" input.

#### REFERENCES

- [1] Z. Hebký: Multiple sampling in digital control (in Czech). *Automatizace* 27 (1984), 6, 142 to 145.
- [2] V. Peterka, J. Böhm, A. Halousková, M. Kárný and J. Maršík: Algorithms for the Adaptive Microprocessor Control of Technological Processes (in Czech). ÚTIA ČSAV, Prague 1982.
- [3] R. L. Keeney and H. Raiffa: Decision analysis with multiple conflicting objectives: preferences and value tradeoffs. IIASA Working Paper WP-75-53, 1975.
- [4] H. Nováková: A Contribution to the Optimal Control of Discrete Stochastic Systems (in Czech). Ph.D. Thesis, ÚTIA ČSAV, Prague 1978.
- [5] K. J. Åström, P. Hagander and J. Sternby: Zeros of sampled systems. IEEE Conf. Decision and Control, Albuquerque, New Mexico 1980.

- [6] T. L. Fine: Theories of Probability: an Examination of Foundations. Academic Press, New York 1973.
- [7] A. Wald: Statistical Decision Functions. Wiley, New York 1950.
- [8] A. Rényi: Theory of Probability (in Czech). Academia, Prague 1972.
- [9] A. Papoulis: Probability, Random Variable and Stochastic Processes. McGraw Hill, New York 1965.
- [10] M. Ullrich: Optimum control of some stochastic systems. Proc. 8th Conf. ETAN, Beograd 1964, 291–298.

### 3. IDENTIFICATION OF NORMAL REGRESSION MODEL

In order to evaluate the conditional expectation of the criterion (2.26) we need to determine the predictive c.p.d.f.  $p(y_s(t) | t-1; u(t))$  of the extended output  $y_s(t) = [y'(t), u_0'(t), y_0'(t), v'(t)]$ . Recall that the reference values  $u_0(t), y_0(t)$  are either specified by the user or could be considered as a component of the measurable disturbance  $v(t)$  or the auxiliary output  $y_a(t)$ . Therefore, it is sufficient to consider only the c.p.d.f.  $p(y(t), v(t) | t-1; u(t))$ . Taking into account the definition of the external measurable disturbance we can split the description of  $y(t)$  and  $v(t)$  as follows:

$$\begin{aligned} p(y(t), v(t) | t-1; u(t)) &= p(y(t) | t-1; u(t)) p(v(t) | t-1; u(t), y(t)) = \\ &= p(y(t) | t-1; u(t)) p(v(t) | v(1..t-1)). \end{aligned}$$

The c.p.d.f.'s  $p(y(t) | t-1; u(t))$  and  $p(v(t) | v(1..t-1))$  may be treated separately (see [11] for details). As both the c.p.d.f.'s are determined analogously, this chapter deals explicitly only with the first of them.

A reasonable way of parametrizing the predictive c.p.d.f.  $p(y(t) | t-1; u(t))$  discussed in Section 2.8 led to the normal regression model (c.f. (2.67), (2.72), (2.75) and (2.80))

$$(1) \quad \begin{aligned} p(y(t) | t-1; u(t), P(t), R(t)) &= (2\pi)^{-m_y/2} |R(t)|^{-1/2} \cdot \\ &\cdot \exp \left\{ -\frac{1}{2} (y(t) - P'(t) z(t))' R^{-1}(t) (y(t) - P'(t) z(t)) \right\}. \end{aligned}$$

In practice the structure of the regression model described by the vector of structural indices  $k' = [l_u, l_y, l_v, l_u]$  is often chosen a priori, especially when some prior experience with system is available. Then the unknown parameters of the model are formed by the  $(i_z, m_y)$  – matrix of the regression coefficients  $P'(t)$  and covariance  $(m_y, m_y)$  – matrix of the innovation sequence (of the “unpredictable” part of the system output)  $R(t)$ . Given the c.p.d.f. (1) the predictive c.p.d.f. not containing the unknown parameters can be obtained as follows:

$$(2) \quad \begin{aligned} p(y(t) | t-1; u(t)) &= \\ &= \int p(y(t) | t-1; u(t), P(t), R(t)) p(P(t), R(t) | t-1) dP(t) dR(t). \end{aligned}$$

At this point the need for parameter estimation arises. Note that in general case the

whole c.p.d.f.  $p(P(t), R(t) | t - 1)$  quantitatively describing uncertainty of the parameters  $P(t), R(t)$  is to be determined.

Sometimes it is sufficient to identify the model in one shot before than control based on the results of this identification is applied, especially when the controlled system can be described by a time-invariant linear (in parameters) model. Such a situation is not characteristic for adaptive control. However, the scheme of one-shot identification enables us to explain in Section 3.1 more simply the questions connected with the use of standard least squares as regards the consequence of bad excitation of the controlled system or asymptotic properties of estimation.

Typical for adaptive control are the cases when the parameters of the model describing the controlled system vary, e.g. due to time variations of system characteristics or changes of a working point about which a nonlinear system is linearized. A proper modelling should guarantee that the model parameters are varying relatively slowly (cf. Section 2.8). In order that the model well may fit the real system behaviour, a recursive parameter tracking is needed. In Section 3.2 we describe a rational way how to attain this tracking and, consequently, adaptivity of controller which is based on suppressing (“forgetting”) obsolete information.

Regardless of adaptivity some inputs to identification must be specified a priori by the user. These “user’s knobs” give possibility of influencing actively the transient behaviour as well as self-tuning properties of the controller. The choice of the prior c.p.d.f.  $p(P(1), R(1) | 0)$  is discussed in Section 3.3.

So far we have discussed the case when the structure of the regression model was chosen in advance. However, sometimes we are uncertain about which structure  $k' = [l_u, l_y, l_v, l_d]$  from several possibilities best fits the real system. The chosen approach to identification enables the user to take this uncertainty into account analogously as the uncertainty of the parameters  $P(t), R(t)$ . Thus, the user can evaluate the probability distribution on possible structures, conditioned by observed data, and then make a suitable decision, i.e. choose the best fitted structure. This line is briefly pursued in Section 3.4.

High numerical reliability of identification must be ensured not only by a thorough formulation and solution of the problem, but also by an adequate algorithmization. In Section 3.5 the reasons for using suitable (Cholesky or  $L-D$ ) factorizations of positive definite matrices are explained.

### 3.1. One-shot identification

Let us suppose that we have observed data up to the time  $t$  on a system describable by the normal regression model with a known structure and unknown but constant parameters  $P, R$ . As pointed out earlier, it is the c.p.d.f. of the unknown parameters  $p(P(t+1), R(t+1) | t)$  which is needed to determine the predictive c.p.d.f. Note that a simpler notation (without the time arguments of the parameters) might

be used, however, the chosen one will make the transition to the recursive version easier later on.

Using repeatedly the Bayes rule (2.55) and taking into account the constancy of the parameters  $P(t+1) = P(t) = \dots = P(1)$ ,  $R(t+1) = R(t) = \dots = R(1)$  we derive

$$(3) \quad p(P(t+1), R(t+1) | t) \propto l(t | 0; P(t+1), R(t+1)) p(P(t+1), R(t+1) | 0)$$

where the form of the likelihood function  $l(t | 0; P(t+1), R(t+1))$  of the unknown parameters (see [11]) follows from (1)

$$(4) \quad l(t | 0; P(t+1), R(t+1)) = \prod_{\tau=1}^t p(y(\tau) | \tau-1; u(\tau), P(t+1), R(t+1)) = \\ = (2\pi)^{-m_y t/2} |R(t+1)|^{-t/2} \cdot \\ \cdot \exp \left\{ -\frac{1}{2} \text{tr} \left( R^{-1}(t+1) \begin{bmatrix} -I \\ P(t+1) \end{bmatrix} \sum_{\tau=1}^t \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix} \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix} \begin{bmatrix} -I \\ P(t+1) \end{bmatrix} \right) \right\}$$

and the symbol  $\propto$  denotes proportionality, i.e. equality up to the normalizing factor of the c.p.d.f. If the prior c.p.d.f.  $p(P(t+1), R(t+1) | 0)$  is sufficiently flat and the observed data carry sufficient information, the form of the posterior c.p.d.f. is determined mainly by (4). Therefore, it is reasonable to retain the functional form of the likelihood function by choosing the prior c.p.d.f. in the same form

$$(5) \quad p(P(t+1), R(t+1) | 0) \propto |R(t+1)|^{-(v(1|0) + I_z + m_y + 1)/2} \cdot \\ \cdot \exp \left\{ -\frac{1}{2} \text{tr} \left( R^{-1}(t+1) \begin{bmatrix} -I \\ P(t+1) \end{bmatrix} V(1|0) \begin{bmatrix} -I \\ P(t+1) \end{bmatrix} \right) \right\}.$$

Note that the quantities specifying (5) must fulfil the conditions  $v(1|0) > m_y - 1$ ,  $V(1|0) > 0$  in order that this function may be a probability density function (the normalizing factor may be finite). Then the posterior c.p.d.f.  $p(P(t+1), R(t+1) | t)$  has the same form as the prior one with the statistics (i.e. quantities accumulating information contained in the observed data)

$$(6) \quad v(t+1 | t) = v(1 | 0) + t$$

$$(7) \quad V(t+1 | t) = V(1 | 0) + \sum_{\tau=1}^t \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix} \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix}.$$

It is suitable to introduce by partitioning the matrix

$$(8) \quad V(t+1 | t) = \underbrace{\begin{bmatrix} V_y(t+1 | t) & V_{zy}(t+1 | t) \\ V_{zy}(t+1 | t) & V_z(t+1 | t) \end{bmatrix}}_{m_y} \underbrace{\begin{bmatrix} V_y(t+1 | t) \\ V_z(t+1 | t) \end{bmatrix}}_{I_z}$$

the following statistics known from the method of least squares

$$(9) \quad A(t+1 | t) = V_y(t+1 | t) - V_{zy}(t+1 | t) V_z^{-1}(t+1 | t) V_{zy}(t+1 | t)$$

$$(10) \quad \hat{P}(t+1|t) = V_z^{-1}(t+1|t) V_{zy}'(t+1|t)$$

$$(11) \quad C(t+1|t) = V_z^{-1}(t+1|t).$$

The only modification lies in respecting prior information through the initial conditions  $v(1|0)$ ,  $V(1|0)$  which are zero in the least squares method. Recall that  $\hat{P}(t+1|t)$  is the least squares estimate of the regression coefficients and  $\Lambda(t+1|t)$  is the sum of residual squares with the minimal trace  $\text{tr}(\Lambda(t+1|t))$ .

Using the above defined statistics the posterior c.p.d.f. can be arranged by completing squares with respect to  $P(t+1)$

$$(12) \quad p(P(t+1), R(t+1)|t) \propto |R(t+1)|^{-v(t+1|t)+i_z+m_y+1/2} \cdot \exp \left\{ -\frac{1}{2} \text{tr} \left( R^{-1}(t+1) [\Lambda(t+1|t) + (P(t+1) - \hat{P}(t+1|t))' \cdot C^{-1}(t+1|t) (P(t+1) - \hat{P}(t+1|t))] \right) \right\}.$$

On the basis of this c.p.d.f. the conditional expectations of the unknown parameters can be evaluated (see [1.2], [11])

$$(13) \quad E[P(t+1)|t] = \hat{P}(t+1|t)$$

$$(14) \quad E[R(t+1)|t] = \hat{R}(t+1|t) = \frac{\Lambda(t+1|t)}{v(t+1|t) - m_y - 1},$$

$$v(t+1|t) > m_y + 1.$$

Uncertainty of the regression coefficients is described by their covariance tensor which is given by the Kronecker product of  $\hat{R}(t+1|t)$  and  $C(t+1|t)$

$$(15a) \quad \text{cov}(P(t+1)|t) = \hat{R}(t+1|t) \otimes C(t+1|t)$$

or in the entry-wise form

$$(15b) \quad E[(P_{ij}(t+1) - \hat{P}_{ij}(t+1|t))(P_{kl}(t+1) - \hat{P}_{kl}(t+1|t))|t] = \hat{R}_{jk}(t+1|t) C_{il}(t+1|t), \quad i, k = 1, 2, \dots, i_z, \quad j, l = 1, 2, \dots, m_y.$$

The predictive c.p.d.f.  $p(y(t+1)|t; u(t+1))$  might be computed now according to (2), but, as will be seen later on, only the conditional expectation and covariance of the output  $y(t)$  are needed for control design. It holds (see [1.2])

$$(16) \quad E[y(t+1)|t; u(t+1)] = \hat{y}(t+1|t; u(t+1)) = \hat{P}'(t+1|t) z(t+1)$$

$$(17) \quad \text{cov}[y(t+1)|t; u(t+1)] = \hat{R}_y'(t+1|t; u(t+1)) = \hat{R}(t+1|t) (1 + \zeta(t+1|t))$$

where the scalar

$$(18) \quad \zeta(t+1|t) = z'(t+1) C(t+1|t) z(t+1).$$

Notice that uncertainty of the regression coefficients (cf. (17)) is projected through  $\zeta(t+1|t)$  onto the increased output covariance. All information needed to determine

the conditional moments (16) and (17) is accumulated (besides the data in the regressor  $z(t+1)$ ) in the statistic

$$(19) \quad \mathcal{S}(t+1|t) = (\hat{P}(t+1|t), \hat{R}(t+1|t), C(t+1|t)).$$

Apparently, identification cannot provide the user with more information than that contained in the observed data. The following comments should help to interpret properly the outputs of identification based on "noninformative" data and, on the other hand, show how the user must prepare data to achieve results needed.

**Example 1.** We shall discuss the estimation of the regression coefficients  $P' = [b, a]$  of the model

$$y'(t+1) = bu'(t+1) + ay'(t) + \varepsilon(t+1), \quad \varepsilon(t+1) \sim N(0, R), \quad m_y, m_a = 1$$

provided that the linear feedback  $u(\tau) = cy'(\tau-1)$ ,  $\tau = 1, 2, \dots, t$ , is applied. Let us suppose that the structure of the real system coincides with the structure of the discussed model, the system has zero initial conditions, the feedback is stabilizing (i.e.  $|a+bc| < 1$ ) and the estimation starts from  $v(1|0) = m_y + 1 + \omega$ ,  $V(1|0) = \omega I$  ( $\omega$  close to zero).

The structure of the matrix  $V_z(t+1|t)$  is influenced by the linear dependence of  $u(\tau)$  and  $y(\tau-1)$ . As  $z'(\tau) = [u(\tau), y(\tau-1)] = [c, 1]y'(\tau-1)$ ,

$$V_z(t+1|t) = V_z(1|0) + \sum_{\tau=1}^t y^2(\tau-1) \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}'.$$

This matrix tends with growing  $t$  to the form

$$V_z(t+1|t) \rightarrow \omega I + tR_y \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}'.$$

where  $R_y = R/(1 - (a+bc)^2)$ .

Inverting  $V_z(t+1|t)$  it can be found that the conditional covariance matrix of  $P(t+1)$  will approach

$$\begin{aligned} & \text{cov}(P(t+1)|t; R) \rightarrow \\ & \rightarrow \begin{bmatrix} 1 & c \\ \sqrt{(1+c^2)} & \sqrt{(1+c^2)} \\ -c & 1 \\ \sqrt{(1+c^2)} & \sqrt{(1+c^2)} \end{bmatrix} \begin{bmatrix} R & 0 \\ \omega & R \\ 0 & \omega + (1+c^2)tR_y \end{bmatrix} \begin{bmatrix} 1 & c \\ \sqrt{(1+c^2)} & \sqrt{(1+c^2)} \\ -c & 1 \\ \sqrt{(1+c^2)} & \sqrt{(1+c^2)} \end{bmatrix}'. \end{aligned}$$

As the conditional probability distribution of the parameters  $P(t+1)$  (supposing  $R$  is known) is normal, using the above factorization of the conditional covariance matrix the reader may easily visualize how this distribution is shaped. Owing to the linear feedback a probability "ridge" along the line  $[1, -c](P(t+1) - \hat{P}(t+1|t)) = 0$  of nearly equiprobable values  $P(t+1)$  arises on the space of all possible values. Note that the finiteness of the corresponding variance  $R/\omega$  is

ensured only by the initial condition  $V_z(1 | 0)$  as the observed data bring no information along the mentioned line. On the other hand, the distribution in the direction orthogonal to the probability "ridge" concentrates — the value of the corresponding variance tends to zero as  $t \rightarrow \infty$ .

**Example 2.** Consider the same problem as in Example 1 with the only exception: the system is excited by the discrete white noise  $u(\tau) \sim N(0, R_u)$ ,  $\tau = 1, 2, \dots, t$ ,  $R_u > 0$ .

If the input  $u(\tau)$  is not correlated with  $y(\tau - 1)$  and  $e(\tau)$ , the matrix

$$V_z(t + 1 | t) = V_z(1 | 0) + \sum_{\tau=1}^t \begin{bmatrix} u(\tau) \\ y(\tau - 1) \end{bmatrix} \begin{bmatrix} u(\tau) \\ y(\tau - 1) \end{bmatrix}$$

tends with growing  $t$  to the diagonal form

$$V_z(t + 1 | t) \rightarrow \omega I + t \begin{bmatrix} R_u & 0 \\ 0 & R_y \end{bmatrix}$$

where  $R_y = (b^2 R_u + R)/(1 - a^2)$ . The conditional covariance matrix of the regression coefficients approaches

$$\text{cov}(P(t + 1) | t; R) \rightarrow \begin{bmatrix} \frac{R}{\omega + tR_u} & 0 \\ 0 & \frac{R}{\omega + tR_y} \end{bmatrix}.$$

Notice that in comparison with the preceding example the whole probability distribution concentrates around  $\hat{P}(t + 1 | t)$ . The corresponding variances tend to zero as  $t \rightarrow \infty$ .

The result of the preceding examples may be generalized (cf. [11], [2.2]). If the trajectories  $z_i(1 \dots t)$ ,  $i = 1, 2, \dots, i_z$  of the particular quantities in the regressor are (almost) linearly dependent, then the observed vectors  $z(\tau)$ ,  $\tau = 1, 2, \dots, t$  do not generate ("excite") the whole linear space of possible values uniformly and the probability distribution concentrates only on some subspace. Therefore, little information (in comparison with prior) is gained about the parameters  $P(t + 1)$  in some directions. Note that the fast variations of the quantities  $z_i(\tau)$  (e.g. the presence of a noise-like quantity in  $z(\tau)$ ) do not imply automatically a good excitation.

Poor excitation of the controlled system may have different causes, e.g. linear feedback (as in Example 1), overparametrization of the model, rare changes of external disturbances, or input saturation. Its consequences depend on whether the missing information will be needed in future. This fact can be illustrated by Example 1. Different values  $\hat{P}(\tau + 1 | \tau)$  along the probability "ridge" provide practically the same conditional expectation of output  $\hat{y}(\tau + 1 | \tau; u(\tau + 1))$  while the linear dependence between the observed data  $u(\tau + 1)$  and  $y(\tau)$ ,  $\tau = t, t + 1, \dots$  is given by the linear feedback  $u(\tau + 1) = c y(\tau)$ . If any change of this dependence occurs,

then the conditional expectation  $\hat{y}(\tau + 1 | \tau; u(\tau + 1))$  considerably differs for the mentioned values  $\hat{P}(\tau + 1 | \tau)$ . Moreover, the lack of information projects on to the large conditional covariance matrix  $\hat{R}_y(\tau + 1 | \tau; u(\tau + 1))$ .

**Remarks.**

- (i) It can be proved (see [12]) that if all accumulated information is saved (no forgetting is applied), the probability distribution of the unknown parameters  $P(t + 1)$ ,  $R(t + 1)$  concentrates. Moreover, in case of a persistent excitation of the controlled system the distribution concentrates around a single point  $\hat{P}$ ,  $\hat{R}$ .
- (ii) The assumption of normality of the innovation sequence is not usually critical. E.g. the uniform probability distribution on an interval of possible values  $e(t)$  can be well approximated by the normal distribution concentrating most probability on the same interval. The exception is made by distributions with heavy-tailed p.d.f.'s (such as the Cauchy distribution) producing "outliers" in the innovation sequence which drastically influence estimation of parameters (see [2] for details). The simple test

$$(20) \quad (y(t + 1) - \hat{y}(t + 1 | t; u(t + 1)))' \hat{R}_y^{-1}(t + 1 | t; u(t + 1)) \cdot (y(t + 1) - \hat{y}(t + 1 | t; u(t + 1))) > \frac{1}{2}(3 + \sqrt{(2m_y - 1)^2}),$$

based on the  $\chi^2$ -distribution of the tested quadratic form, may be used for the detection of such outliers.

- (iii) Whenever the results of the presented identification are used out of the treated area, the danger of misinterpretation arises. The common one lies in an attempt to separate a "deterministic" part of the model by removing the innovation. However, only the regression model as a whole (i.e. with the innovation as an inherent component) provides a complete prediction of output. Recall e.g. that by cancelling the common factor the "deterministic" transfer function remains unchanged but the corresponding prediction may considerably differ.
- (iv) It should be apparent at the end of this section that the outputs of identification are widely dependent on data. Therefore, pre-processing of data in the sense of Section 2.4 can often improve the final results.

**3.2. Parameter tracking**

In practice the assumption of constant parameters is rarely acceptable, at least due to the always present long-term variations of system characteristics. Then it is necessary:

- (i) to identify the model, i.e. to determine the predictive c.p.d.f.  $p(y(t) | t - 1; u(t))$  for  $t = 1, 2, \dots$
- (ii) to find a reasonable (simple but effective) way how to ensure adaptivity of identification even in case that the model of parameter variations is not known.

These requirements imply the need to update the c.p.d.f.'s of unknown parameters recursively

$$p(P(t), R(t) | t-1) \rightarrow p(P(t+1), R(t+1) | t), \quad t = 1, 2, \dots$$

The data updating  $p(P(t), R(t) | t-1) \rightarrow p(P(t), R(t) | t)$  is given by the Bayes rule (2.55). The question is how to approximate the time updating  $p(P(t), R(t) | t) \rightarrow p(P(t+1), R(t+1) | t)$ .

If no information about the evolution of parameters is available, then it is appropriate to seek the solution as a compromise between two extreme forms of the time-updated c.p.d.f.: the posterior c.p.d.f.

$$(21) \quad p_0(P(t+1), R(t+1) | t) \equiv p(P(t), R(t) | t)$$

corresponding to the case of constant parameters  $P(t+1) = P(t)$ ,  $R(t+1) = R(t)$  and the alternative c.p.d.f.

$$(22) \quad p_1(P(t+1), R(t+1) | t) \propto |R(t+1)|^{-(t+m_p+1)/2}$$

corresponding to the case of worst possible changes of parameters when we are maximally uncertain about  $P(t+1)$ ,  $R(t+1)$  (it is the limit form of (12) for  $C^{-1}(t+1 | t) \rightarrow 0$ ,  $A(t+1 | t) \rightarrow 0$ ,  $v(t+1 | t) \rightarrow 0$ ). Intuitively, the faster the parameters are expected to vary, the less we believe in the posterior c.p.d.f. (21) and, consequently, the more information accumulated in  $p_0(P(t+1), R(t+1) | t)$  is to be suppressed ("forgotten"). A thorough analysis of the sketched situation results in the following formula ([8], [9]) formalizing the intuitively expected "flattening" of the posterior c.p.d.f. (21)

$$(23) \quad p(P(t+1), R(t+1) | t) \propto [p_0(P(t+1), R(t+1) | t)]^\varphi [p_1(P(t+1), R(t+1) | t)]^{1-\varphi}$$

where the "forgetting factor"  $\varphi \in (0, 1)$  expresses the measure of our belief in the posterior c.p.d.f. (its subjective probability).

It can be easily shown (see [9]) that if the c.p.d.f.  $p(P(t), R(t) | t-1)$  has the form (12), then the Bayes rule (2.55) and the "exponential forgetting" (23) save it, modifying the appropriate quantities as follows:

$$(24) \quad \hat{P}(t+1 | t) = \hat{P}(t | t-1) + \frac{C(t | t-1) z(t)}{1 + \zeta(t | t-1)} \bar{\varepsilon}'(t | t-1)$$

$$(25) \quad C^{-1}(t+1 | t) = \varphi [C^{-1}(t | t-1) + z(t) z'(t)]$$

$$(26) \quad A(t+1 | t) = \varphi \left[ A(t | t-1) + \frac{\bar{\varepsilon}(t | t-1) \bar{\varepsilon}'(t | t-1)}{1 + \zeta(t | t-1)} \right]$$

$$(27) \quad v(t+1 | t) = \varphi [v(t | t-1) + 1]$$

where the prediction error is given by

$$(28) \quad \bar{\varepsilon}(t | t-1) = y(t) - \hat{P}'(t | t-1) z(t).$$

Recall that the factor  $\zeta(t | t - 1)$  is defined as

$$(29) \quad \zeta(t | t - 1) = z'(t) C(t | t - 1) z(t).$$

To keep the form of the c.p.d.f. self-reproducing along the needed time interval, it is sufficient to choose the prior c.p.d.f.  $p(P(1), R(1) | 0)$  in the form (12). Then the recursive updating of the c.p.d.f. of the unknown parameters is fully described by the above relations and the computation of the needed conditional moments of the unknown parameters and output is given by the same equations as in case of one-shot identification.

For  $\varphi = 1$  the relations (24)÷(27) form the recursive version of the solution found in the preceding section. Recall that in case that the controlled system is poorly excited no or little information about the regression coefficients is gained in some directions in addition to prior information. The situation is radically different when the factor  $\varphi < 1$  is applied. Due to the noninformative character of the alternative c.p.d.f.  $p_1(P(t + 1), R(t + 1) | t)$  prior information is moreover significantly suppressed in all directions.

**Example 1 — continued.** Let us apply the exponential forgetting with  $\varphi < 1$  in example 1 from section 3.1. The matrix

$$V_z(t + 1 | t) = \varphi^t V_z(1 | 0) + \sum_{\tau=1}^t \varphi^{t-\tau+1} y^2(\tau - 1) \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}'$$

tends with growing  $t$  to the form

$$V_z(t + 1 | t) \rightarrow \varphi^t \omega I + \frac{\varphi(1 - \varphi^t)}{1 - \varphi} R_y \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}',$$

therefore, the conditional covariance matrix of  $P(t + 1)$  approaches

$$\begin{aligned} & \text{cov}(P(t + 1) | t; R) \rightarrow \\ & \rightarrow \begin{bmatrix} \frac{1}{\sqrt{(1 + c^2)}} & \frac{c}{\sqrt{(1 + c^2)}} \\ -c & \frac{1}{\sqrt{(1 + c^2)}} \end{bmatrix} \begin{bmatrix} \frac{R}{\varphi^t \omega} & 0 \\ 0 & \frac{1}{\varphi^t \omega + (1 + c^2) \frac{\varphi(1 - \varphi^t)}{1 - \varphi} R_y} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{(1 + c^2)}} & \frac{c}{\sqrt{(1 + c^2)}} \\ -c & \frac{1}{\sqrt{(1 + c^2)}} \end{bmatrix}' \end{aligned}$$

Note that the variance along the line  $[c, 1](P(t + 1) - \hat{P}(t + 1 | t)) = 0$  tends to a nonzero value so that the tracking of varying regression coefficients along this line is ensured. In the orthogonal ("nonexcited") direction the appropriate variance tends quickly to infinity as the regularizing initial condition is successively suppressed.

Thus, the application of the exponential forgetting under data noninformativity has serious practical consequences for the quality and reliability of adaptive control (cf. [1]). The probability distribution of  $P(t)$  does not concentrate on some linear subspace but, on the contrary, disperses so that it approaches the uniform distribution.

In terms of the regression model, some entries of the matrix  $C(t | t - 1)$  are permanently growing due to the constant overweighting by the factor  $1/\varphi$  regardless of measured data. If the trajectories  $z_i'(1 \dots t)$ ,  $i = 1, 2, \dots, i_z$  are linearly dependent in a long time interval, this growing continues without influencing the quality of control till numerical breakdown of estimation ("covariance wind-up") occurs. In case that the linear dependence is broken or changed, the model may not predict the system behaviour satisfactorily for a short time and control synthesis starts to generate inappropriate control actions. A short-term instability of the closed loop ("output bursting") is a typical expression of this lack of information.

This is why we have suggested (see [7], [8], [9]) to restrict the exponential forgetting (23) only to the c.p.d.f. of the "excited" quantities  $R(t + 1)$  and  $z'(t + 1)$ .  $P(t + 1)$ . Then the restricted exponential forgetting increases uncertainty of the regression coefficients  $P(t + 1)$  solely in their projection to the direction of the regressor  $z(t + 1)$ . This characteristic feature of the suggested technique accounts for its special name "directional forgetting".

The directional forgetting modifies the relations (24) and (25) as follows: if  $\zeta(t | t - 1) > 0$ , then

$$(30a) \quad \hat{P}(t + 1 | t) = \hat{P}(t | t - 1) + \frac{C(t | t - 1) z(t)}{1 + \zeta(t | t - 1)} \bar{\varepsilon}'(t | t - 1)$$

$$(31a) \quad C^{-1}(t + 1 | t) = C^{-1}(t | t - 1) + w(t) z(t) z'(t)$$

where

$$(32) \quad w(t) = \varphi - \frac{1 - \varphi}{\zeta(t | t - 1)};$$

if  $\zeta(t | t - 1) = 0$ , then

$$(30b) \quad \hat{P}(t + 1 | t) = \hat{P}(t | t - 1)$$

$$(31b) \quad C(t + 1 | t) = C(t | t - 1).$$

Note that the main difference, when compared with standard least squares, lies in that only the weight of the new data dyad  $z(t) z'(t)$  is influenced by the forgetting factor  $\varphi$ .

**Example 1 — completed.** Let us apply the directional forgetting to the problem given in Example 1. By a simple analysis it can be found that the matrix

$$V_z(t + 1 | t) = V_z(1 | 0) + \sum_{\tau=1}^t w(\tau) y^2(\tau - 1) \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}'$$

tends with growing  $t$  to the form

$$V_z(t + 1 | t) \rightarrow \frac{\varphi' \omega}{1 + c^2} \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}' + \frac{\omega}{1 + c^2} \begin{bmatrix} 1 \\ -c \end{bmatrix} \begin{bmatrix} 1 \\ -c \end{bmatrix}' + \frac{\varphi(1 - \varphi^t)}{1 - \varphi} R_y \begin{bmatrix} c \\ 1 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}'$$

and the conditional covariance matrix of  $P(t+1)$  approaches

$$\text{cov}(P(t+1) | t; R) \rightarrow \begin{bmatrix} \frac{1}{\sqrt{(1+c^2)}} & \frac{c}{\sqrt{(1+c^2)}} \\ -c & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{(1+c^2)}} & \frac{c}{\sqrt{(1+c^2)}} \\ -c & 1 \end{bmatrix} \\ \rightarrow \begin{bmatrix} \frac{1}{\sqrt{(1+c^2)}} & \frac{c}{\sqrt{(1+c^2)}} \\ -c & 1 \end{bmatrix} \begin{bmatrix} \omega & 0 \\ \varphi^t \omega + (1+c^2) \frac{\varphi(1-\varphi^t)}{1-\varphi} & R_y \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{(1+c^2)}} & \frac{c}{\sqrt{(1+c^2)}} \\ -c & 1 \end{bmatrix}$$

It can be observed that the resulting variance along the line  $[c, 1](P(t+1) - \hat{P}(t+1 | t)) = 0$  is the same as in the case of the exponential forgetting. However, in the orthogonal ("nonexcited") direction the prior form is saved.

The underlying principle and simulation as well as practical experience clearly indicate that the described modification substantially improves numerical reliability of identification. It prevents the covariance wind-up and substantially limits the output bursting regardless of a concrete source of data noninformativity. However, it should be emphasized that the directional forgetting is only (though more sophisticated) a way of processing passively gathered information. It cannot solve the problems connected with a real lack of information and does not overcome the uniform forgetting of the excited piece of information (it is the reason why we cannot theoretically exclude the output bursting).

Till now it has been assumed that the forgetting factor is constant and can be fixed a priori on the basis of practical experience and the expected character of parameter variations. A feeling how to choose the value of  $\varphi$  may be based on the asymptotic value of the scalar  $v(t | t-1)$ . The reader easily finds from (27) that

$$(33) \quad v(t | t-1) \rightarrow v^* = \frac{\varphi}{1-\varphi}.$$

The value  $v^*$  can be interpreted as the length of a rectangular "window" fictitiously replacing the real exponential "window" which moves over the observed data. Then, loosely speaking, the importance of new information contained in the latest data with respect to hitherto accumulated information is given by the ratio  $1/v^* = (1-\varphi)/\varphi$  (e.g. the commonly recommended choice  $\varphi = 0.985$  implies the ratio approximately 1 : 66). The rule of thumb follows from this view: estimate the length of the time interval in which observable parameter changes are expected to appear and then compute the value of  $\varphi$  from (33).

There are cases when the choice of the most suitable value of  $\varphi$  is more complicated or the constant value of  $\varphi$  cannot simply cover the varying rate of parameter changes. In these cases it would be desirable to make the choice less dependent on the user and to employ information contained in data. A reasonable procedure has been proposed on the basis of information-theoretic considerations (see [8], [9]). The aim of controlling the forgetting is to achieve the state when, loosely speaking, the

expected amount of "forgotten" information is proportional (through an a priori chosen factor  $\varrho$ ) to the expected amount of new information. The approximate computation of the data and time dependent forgetting factor  $\varphi(t+1 | t)$  (applied after observing data  $u(t), y(t)$ ) gives

$$(34) \quad \varphi^{-1}(t+1 | t) \approx 1 + (1 + \varrho) \left[ \ln(1 + \zeta(t | t-1)) - \frac{\zeta(t | t-1)}{1 + \zeta(t | t-1)} + \psi(t | t-1) \right]$$

where

$$(35) \quad \psi(t | t-1) = \frac{1}{m_y} \frac{\zeta(t | t-1)}{1 + \zeta(t | t-1)} \frac{(v(t | t-1) + 1) \eta(t | t-1)}{1 + \zeta(t | t-1) + \eta(t | t-1)}$$

$$(36) \quad \eta(t | t-1) = \bar{\varepsilon}(t | t-1) A^{-1}(t | t-1) \bar{\varepsilon}(t | t-1)$$

or even, in case of rather slowly varying parameters,

$$(37) \quad \varphi^{-1}(t | t-1) \approx 1 + (1 + \varrho) \psi(t | t-1).$$

Note that instead of one heuristic factor  $\varphi$  a different heuristic factor  $\varrho$  has been involved into the problem. What is the contribution of this modification? Extensive simulation experience indicates that the outputs of identification and the quality of control is much less sensitive to the choice of  $\varrho$  in comparison with the choice of  $\varphi$ . Moreover, the tracking of parameters becomes more adaptive as regards the time variations of the amount of information contained in data. In this way, a roughly exponential growing of the forgetting factor is obtained automatically in the initial phase of identification. This makes the transient tuning shorter and the correction of a poor choice of the prior c.p.d.f.  $p(P(1), R(1) | 0)$  easier.

The choice of the factor  $\varrho$  is facilitated by the approximate relations between  $\varrho$  and the expected value of the forgetting factor  $\varphi^*$  or the asymptotic memory length  $v^*$ . It holds for the approximation (34)

$$(38) \quad \varrho \approx \frac{1}{2} \frac{1 - \varphi^*}{\varphi^*} = \frac{1}{2v^*}$$

and for the approximation (37)

$$(39) \quad \varrho \approx \frac{1 - \varphi^*}{\varphi^*} = \frac{1}{v^*}.$$

It means that e.g. using the approximation (37) and assuming that the parameters significantly vary in 100 steps, we should choose  $\varrho = 0.01$ .

#### Remarks.

- (i) The algorithm of the parameter tracking using the directional forgetting includes inherently one singularity. If the value of  $\zeta(t | t-1)$  is zero, the computation goes to the singular branch (the relations (30b), (31b)) which means that the

updating of the c.p.d.f.  $p(P(t) | t-1; R(t))$  is switched off. In order to avoid numerical problems, we must go through the singular branch whenever  $\zeta(t | t-1) \leq \zeta_0$ , where  $\zeta_0 > 0$  is the a priori chosen value substantially smaller than 1 (e.g.  $10^{-6}$ ). The consequences of such an extension can be usually neglected. Note that the test (20) of the outliers may be joined with the test  $\zeta(t | t-1) \leq \zeta_0$ . Then the updating of  $p(P(t) | t-1; R(t))$  is switched off in case of corrupted data, too.

- (ii) Owing to the forgetting of obsolete information the probability distribution of unknown parameters concentrates around its maximum  $\hat{P}(t+1 | t)$ ,  $\hat{R}(t+1 | t)$  only to a certain extent. However, if the parameters of the model vary rather slowly, then  $\varphi$  relatively close to one (the needed closeness depends on the average amount of information contained in data) or for  $\varrho$  close to zero the statistic  $\mathcal{S}(t+1 | t)$  approaches a stationary value. Moreover, when the system is well excited, at least in the initial phase of control, the uncertainty of  $P(t+1)$  described by the matrix  $C(t+1 | t)$  may become rather small (cf. remark (i) in section 3.1).
- (iii) Recall that the conditional covariance (17) is finite only in the case that  $v(t | t-1) > m_y + 1$ . To be able to find the optimum control in the sense of our criterion, the above condition must be fulfilled at each time instant. It means that the value of the forgetting factor should be checked on

$$(40) \quad \varphi(t+1 | t) > \frac{m_y + 1}{v(t | t-1) + 1}.$$

For slowly varying parameters this inequality brings no constraint.

### 3.3. Choice of prior distribution

The preceding sections provide the user with all relations needed for recursive identification of the normal regression model (with a given structure  $k$ ). In order that the user may apply these results, he must specify some quantities in advance. To summarize, these are

- (i) the values  $\hat{P}(1 | 0)$ ,  $C(1 | 0)$ ,  $A(1 | 0)$ ,  $v(1 | 0)$  fully describing the prior c.p.d.f.  $p(P, 1)$ ,  $R(1 | 0)$  which form the initial conditions of the algebraic recursions (30), (31), (26), (27);
- (ii) the forgetting factor  $\varphi$  or  $\varrho$ ;
- (iii) the discrimination level  $\zeta_0$ .

The choice of the last two quantities has been discussed in section 3.2. Now we shall pay our attention to the question how to quantify prior information about unknown parameters. It should be emphasized that the choice of prior values  $\hat{P}(1 | 0)$ ,  $C(1 | 0)$ ,  $A(1 | 0)$ ,  $v(1 | 0)$  determines the relation between the rate of the estimation convergence (the rate of probability concentration on the parameter values giving good output prediction) and the quality of control in the transient phase. Thus,

a careful specification of  $p(P(1), R(1) | 0)$  enables the user to improve substantially the transient closed-loop behaviour.

Due to variety of controlled systems as well as a different level of user's prior knowledge no standard, universally acceptable choice of these quantities can be given. The often recommended choice

$$(41) \quad C^{-1}(1 | 0) \rightarrow \omega I, \quad A(1 | 0) \rightarrow \omega I, \quad v(1 | 0) \rightarrow m_y + 1 + \omega$$

(with  $\omega$  given by the computational precision) which models total parameter uncertainty (almost uniform prior distribution) is dangerous in that it may cause a short-term instability of the closed loop (see Chapter 4).

Experience from practical applications shows that the most effective way of specifying the prior distribution is to fill the needed quantities by real data, i.e. to start (e.g. from the prior values (41)) the parameter estimation on the real system before closing the adaptive control loop. Usually a couple of steps are fully sufficient.

In the cases when the above procedure is not possible the situation is more complicated. The choice of the prior distribution is then based only on that how the user quantifies his prior information. As a rule, the problem does not consist in the lack of prior information but in expressing it in terms of the quantities  $\hat{P}(1 | 0)$ ,  $C(1 | 0)$ ,  $A(1 | 0)$ ,  $v(1 | 0)$ . The following comments should help the user at this step:

- (i) The choice of  $v(1 | 0)$ . The value of  $v(1 | 0)$  can be taken up as a number of fictitious measurements conditioning user's prior information. Note that the weight of the first real data is given approximately by the ratio  $1/v(1 | 0)$  (cf. the way by which the likelihood function is formed in section 3.1). To ensure the finiteness of  $\hat{R}(1 | 0)$  the condition  $v(1 | 0) > m_y + 1$  should be fulfilled. In case of little information it is sufficient to take the minimum value  $v(1 | 0) = m_y + 2$ .
- (ii) The choice of  $A(1 | 0)$ . Using the relation (14) we can compute the matrix  $A(1 | 0)$  as follows

$$(42) \quad A(1 | 0) = (v(1 | 0) - m_y - 1) \hat{R}(1 | 0)$$

on condition that we know how to estimate the innovation covariance matrix  $R(1)$ . Its estimation may be facilitated by the fact that on average we cannot predict and, consequently, also control the system output more accurately than is given by  $R$ . It means that the value  $\hat{R}(1 | 0)$  can be derived, for instance, from the assumed accuracy with which the output is determined on the basis of last data. Some troubles arise when the output is multivariate. It has proved useful in such a case to understand the innovation  $\varepsilon(1)$  as the output of a linear filter excited by the vector of independent equally distributed innovations

$$(43) \quad \varepsilon(1) = F_\varepsilon(1) \varepsilon^0(1), \quad \varepsilon^0(1) \sim N(0, I).$$

The reader easily finds out that  $R(1) = F_\varepsilon(1) F_\varepsilon^T(1)$ . Then, the  $(m_y, m_y)$ -type matrix  $F_\varepsilon(1)$  is to be estimated instead of  $R(1)$ . However, it is often possible to take a diagonal  $\hat{R}(1 | 0)$ , i.e. to suppose that the entries  $\varepsilon_i(1)$ ,  $i = 1, 2, \dots, m_y$

are mutually uncorrelated and reduce the problem to the independent choices of diagonal entries  $A_{ii}(1 | 0)$ .

- (iii) The choice of  $\hat{P}(1 | 0)$ . It follows from (13) that  $\hat{P}(1 | 0)$  is formed by the most expected values of particular regression coefficients. Their estimation is facilitated by considering all available information such as the expected value of static gain, assumed stability of the model, used scaling etc.

**Example 3.** Suppose that the model

$$y(t) = b u(t) + a y(t - 1) + v(t)$$

is stable. Then the parameter  $a$  has to lie in the interval  $(-1, 1)$ .

In case of little information  $\hat{P}(1 | 0)$  may be chosen as if the real system were static with a time delay, i.e.  $\hat{P}(1 | 0)$  is zero except the coefficient(s) corresponding to the oldest input.

It should be emphasized that a careful choice of  $\hat{P}(1 | 0)$  has practically no effect if a very large initial uncertainty is assigned to it through  $C(1 | 0)$  (see below).

- (iv) The choice of  $C(1 | 0)$ . Recall that the matrix  $C(1 | 0)$  together with the estimate  $\hat{R}(1 | 0)$  determine the covariance tensor (15) of the regression coefficients

$$(44) \quad \text{cov}(P(1) | 0) = \hat{R}(1 | 0) \otimes C(1 | 0).$$

In the univariate case ( $m_y = 1$ ) it is possible to determine  $C(1 | 0)$  from the equality

$$(45) \quad C(1 | 0) = \frac{\text{cov}(P(1) | 0)}{\hat{R}(1 | 0)}$$

on the basis of the covariance matrix  $\text{cov}(P(1 | 0))$ . However, if  $m_y > 1$  a direct use of (44) is usually too complicated.

The matrix  $C(1 | 0)$  may be often chosen in a diagonal form. Owing to normality of the c.p.d.f.  $p(P(1) | 0; R(1))$  the parameters  $P_{ij}(1)$ ,  $i = 1, 2, \dots, i_2$ ,  $j = 1, 2, \dots, m_y$  lie with high probability in the (confidence) intervals

$$(46) \quad \langle -\alpha \sqrt{(\hat{R}_{jj}(1 | 0) C_{ii'}(1 | 0))} + \hat{P}_{ij}(1 | 0), \hat{P}_{ij}(1 | 0) + \alpha \sqrt{(\hat{R}_{jj}(1 | 0) C_{ii'}(1 | 0))} \rangle, \quad \alpha \approx 2 \div 3.$$

Specifying the expected ranges of the regression coefficients (cf. Example 3) we should select such values  $C_{ii'}(1 | 0)$ ,  $i = 1, 2, \dots, i_2$  that the coincidence with the intervals (46) is achieved.

The more user-friendly and flexible way of choosing the matrix  $C(1 | 0)$  is to fill it by the sequence of fictitious regressor vectors  $z(\tau)$ ,  $\tau = -n, -n + 1, \dots, 0$  expressing linear (or linearized) interrelations between the regression coefficients  $P(1)$ . After constructing the fictitious equations in the form

$$(47) \quad [P(1) - \hat{P}(1 | 0)]' z(\tau) = \varepsilon(\tau), \quad \varepsilon(\tau) \sim N(0, \hat{R}(1 | 0))$$

we may determine the matrix  $C(1 | 0)$  according to (7) and (11)

$$(48) \quad C(1 | 0) = [\omega I + \sum_{\tau=-n}^0 z(\tau) z'(\tau)]^{-1}, \quad \omega > 0 \quad \text{close to zero.}$$

The following example illustrates how information about the expected ranges of the regression coefficients as well as the static gain may be easily incorporated.

**Example 4.** Let us consider the model

$$y(t) = b u(t) + a y(t-1) + \varepsilon(t), \quad \varepsilon(t) \sim N(0, R)$$

and assume that it holds

$$\text{cov}(b(1) | 0) = \hat{R}(1 | 0) c_b$$

$$\text{cov}(a(1) | 0) = \hat{R}(1 | 0) c_a$$

$$\text{cov}(\hat{g}(1) a(1) + b(1) - \hat{g}(1) | 0) = \hat{R}(1 | 0) c_g$$

where in the last equation prior information about the expected static gain  $g(1) = b(1)/(1-a(1))$  is respected. The above relations can be rewritten (normalized) to the form (47) with

$$P(1) = \begin{bmatrix} b(1) \\ a(1) \end{bmatrix}, \quad z(-2) = \begin{bmatrix} 1 \\ \sqrt{c_b} \\ 0 \end{bmatrix}, \quad z(-1) = \begin{bmatrix} 0 \\ 1 \\ \sqrt{c_a} \end{bmatrix}, \quad z(0) = \begin{bmatrix} 1 \\ \sqrt{c_g} \\ \hat{g}(1) \\ \sqrt{c_g} \end{bmatrix}.$$

Using (48) the matrix  $C(1 | 0)$  is determined in a nondiagonal form.

Simulation experience indicates that the incorporation of prior information about important model characteristics, such as the static gain or smoothness of the pulse response, may significantly improve the initial phase of self-tuning (see [6] for details).

### 3.4. Structure determination

Let us suppose that there are several possibilities how to choose the structure of the regression model between which we cannot decide a priori. More formally, the vector of structural indices  $k' = [l_u, l_y, l_v, l_n]$  forms the additional parameter of the regression model. It is the undoubted advantage of the chosen (Bayesian) approach to identification that the estimation of the parameters  $P, R$  and the determination of the structure  $k$  can be solved in a unified manner.

In case of one-shot identification the posterior c.p.f.  $p(k | t)$  results from the Bayes updating (2.55) of the prior c.p.f.  $p(k | 0)$  by all observed data. The reader may easily verify using the natural conditions of control (2.53) that

$$(49) \quad p(k | t) = \frac{l(t | 0; k) p(k | 0)}{\sum_{k \in H} l(t | 0; k) p(k | 0)}$$

where  $H$  denotes the set of possible values of  $k$  and

$$(50) \quad l'(t | 0; k) = \int l(t | 0; P(t), R(t), k) p(P(t), R(t) | 0; k) dP(t) dR(t)$$

is the conditional expectation of the likelihood function (4). Substituting for  $l(t | 0; P(t), R(t), k)$  from (4) and integrating over the space of possible values of  $P(t)$  and  $R(t)$  it can be found (see [1.2], [11], [5]) that

$$(51) \quad l(t | 0; k) = \pi^{-m_y t/2} \prod_{j=1}^{m_y} \frac{\Gamma\left(\frac{{}^k v(t | t) + 1 - j}{2}\right) |{}^k C(t | t)|^{m_y/2} |{}^k A(t | t)|^{-k v(t | t)/2}}{\Gamma\left(\frac{{}^k v(1 | 0) + 1 - j}{2}\right) |{}^k C(1 | 0)|^{m_y/2} |{}^k A(1 | 0)|^{-k v(1 | 0)/2}}$$

where  $\Gamma(\cdot)$  denotes the gamma function,

$$(52) \quad {}^k v'(t | t) = {}^k v(1 | 0) + t$$

and the matrices  ${}^k A(t | t)$ ,  ${}^k C(t | t)$  are determined from

$$(53) \quad {}^k V(t | t) = {}^k V(1 | 0) + \sum_{\tau=1}^t \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix} \begin{bmatrix} y(\tau) \\ z(\tau) \end{bmatrix}'$$

in the same way as (9), (11) from (7).

In case of the uniform prior c.p.f.  $p(k | 0)$  the posterior probability ratio for two possible structures is given by the ratio of the likelihoods (51). Analysing this ratio we find out that the choice of an optimum structure is influenced by

- the quality of prediction (through  $|{}^k A(t | t)|^{-k v(t | t)/2}$ ),
- the uncertainty of the regression coefficients (through  $|{}^k C(t | t)|^{m_y/2}$ ),
- the number of data (through  $\prod_{j=1}^{m_y} \Gamma\left(\frac{{}^k v(t | t) + 1 - j}{2}\right)$ ),

– the prior information (through the prior values of the above quantities).

There are different procedures of structure determination which take (or combine) some of the above items (cf. [13]). Note that the Bayesian approach respects all of them in a natural and consistent way.

The computation should be organized as follows:

1. Choose the set of possible structural vectors  $H$ .
2. Specify the prior c.p.f.  $p(k | 0)$  on particular alternatives with respect to available information (uniform in case that no hypothesis is preferred in advance).
3. For each considered structure  $k$  identify the appropriate model, i.e. choose the prior c.p.d.f.  $p(P(1), R(1) | 0; k)$  and process data using the equations (52) and (53).
4. Evaluate the posterior c.p.f.  $p'(k | t)$  according to the relation (49) using the values (51).

An effective way of dealing with the case when there is a large number of possible structures has been suggested in [5].

The resulting c.p.f.  $p(k | t)$  makes possible to eliminate the uncertainty of the model structure in  $p(y(t + 1) | t; u(t + 1), k)$  and to compute the needed predictive c.p.d.f.

$p(y(t+1) | t; u(t+1))$ . However, processing a sufficiently large sample of data the posterior probability of a particular hypothesis  $\hat{k} \in H$  is, as a rule, so high that enables us to approximate

$$(54) \quad p(y(t+1) | t; u(t+1)) \approx p(y(t+1) | t; u(t+1), \hat{k}).$$

Naturally, the structure determination may be solved in a recursive way, too. It follows from the Bayes rule (2.55) that

$$(55) \quad p(k | t) = \frac{l(t | t-1; k) p(k | t-1)}{\sum_{k \in H} l(t | t-1; k) p(k | t-1)}$$

where the function  $l(t | t-1; k)$  is given by the predictive c.p.d.f. ([1.2], [11])

$$(56) \quad \begin{aligned} l(t | t-1; k) &= p(y(t) | t-1; u(t), k) = \\ &= \pi^{-m_y/2} \frac{\Gamma\left(\frac{k v(t | t-1) + 1}{2}\right)}{\Gamma\left(\frac{k v(t | t-1) + 1 - m_y}{2}\right)} |k \Lambda(t | t-1)|^{-1/2} \\ &\cdot (1 + k \zeta(t | t-1))^{-m_y/2} \left(1 + \frac{k \eta(t | t-1)}{1 + k \zeta(t | t-1)}\right)^{-(k v(t | t-1) + 1)/2} \end{aligned}$$

### 3.5. Comments on algorithmization

From the computational viewpoint the critical point of identification is the updating of matrices which must be, by their nature, positive definite. In order that a high numerical reliability of identification and control may be achieved, the employed algorithms must guarantee the positive definiteness of the matrices  $V(t | t-1)$  or  $C(t | t-1)$  and  $\Lambda(t | t-1)$ .

The effective measure is the Cholesky (square root) factorization of the matrix  $V^{-1}$

$$(57) \quad V^{-1} = GG',$$

the result of which is the lower triangular matrix  $G$  with positive diagonal entries, or the  $L-D$  factorization of the same matrix

$$(58) \quad V^{-1} = LDL',$$

the result of which are the lower triangular matrix  $L$  with units on the main diagonal and the diagonal matrix  $D$ . Partitioning the mentioned matrix factors to blocks

$$(59) \quad G = \begin{bmatrix} \underbrace{G_y}_{m_y} & \underbrace{0}_{i_z} \\ \underbrace{G_{zy}}_{m_y} & \underbrace{G_z}_{i_z} \end{bmatrix}, \quad L = \begin{bmatrix} \underbrace{L_y}_{m_y} & \underbrace{0}_{i_z} \\ \underbrace{L_{zy}}_{m_y} & \underbrace{L_z}_{i_z} \end{bmatrix}, \quad D = \begin{bmatrix} \underbrace{D_y}_{m_y} & \underbrace{0}_{i_z} \\ \underbrace{0}_{m_y} & \underbrace{D_z}_{i_z} \end{bmatrix}$$

we can express the statistics  $\hat{P}, C, A$  (inverting the block-arranged matrices) as follows (see [1.2], [11])

$$(60) \quad \hat{P} = -G_{zy}G_y^{-1} = -L_{zy}L_y^{-1}$$

$$(61) \quad C = G_z G_z' = L_z D_z L_z'$$

$$(62) \quad A = G_y G_y' = L_y D_y L_y'$$

Note that in the univariate case ( $m_y = 1$ )  $L_y = 1$  and, consequently,  $\hat{P} = -L_{zy}$ ,  $A = 1/D_y$ .

Updating the factors  $G(L, D)$  or separately  $G_y, G_z(L_y, D_y, L_z, D_z)$  instead of  $V$  or  $A, C$  the positive definiteness of the original matrices is automatically ensured. Moreover, the determinants  $|A|, |C|$  needed in the structure determination may be easily computed in the following way

$$|A|^{1/2} = \prod_{i=1}^{m_y} G_{yii} = \left( \prod_{i=1}^{m_y} D_{yii} \right)^{1/2}$$

$$|C|^{1/2} = \prod_{i=1}^{i_z} G_{zii} = \left( \prod_{i=1}^{i_z} D_{zii} \right)^{1/2}.$$

The specification of the prior values of the matrix factors follows from the defining relations (60)÷(62).

#### Remarks.

- (i) It is slightly more effective to update the matrix factors corresponding to the  $L$ - $D$  factorization: the computation of  $i_z + m_y$  square roots and  $i_z + m_y$  multiplications needed in the Cholesky factorization (the algorithm REFIL [1.1], [10]) are saved.
- (ii) The modified factorizations

$$V^{-1} = G'G = L'DL$$

have a special property important in case of the structure determination: the matrix factors corresponding to the lower order models are all nested in the matrix factor(s) corresponding to the highest order model (see [5] for details). Extraordinarily fast algorithms for estimation of unknown order of the autoregression models have been designed in [3] by means of lattice structures.

#### REFERENCES

- 
- [1] K. J. Åström: Theory and applications of adaptive control — a survey. *Automatica* 19 (1983), 471—486.
  - [2] P. J. Huber: *Robust Statistics*. Wiley, New York 1981.
  - [3] J. Kadlec: Recursive Probabilistic Identification of Autoregression Model with Unknown Order Using Lattice Structures (in Czech). Report No. 1313, Institute of Information Theory and Automation of Czechoslovak Academy of Sciences, Praha 1985.

- [4] M. Kárný: Recursive parameter estimation of regression model when the interval of possible values is given. *Kybernetika 18* (1982), 1, 37–49.
- [5] M. Kárný: Algorithms for determining the model structure of a controlled system. *Kybernetika 19* (1983), 2, 164–178.
- [6] M. Kárný: Quantification of prior knowledge about global characteristics of linear normal model. *Kybernetika 20* (1984), 5, 376–385.
- [7] R. Kulhavý and M. Kárný: Tracking of slowly varying parameters by directional forgetting. Proc. 9th IFAC Congress, Budapest 1984, Vol. X, 81–85.
- [8] R. Kulhavý: Restricted exponential forgetting in real-time identification. Preprints 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, University of York 1985.
- [9] R. Kulhavý: Probabilistic Identification of Time-Variable Systems with Unknown Model of Parameter Evolution (in Czech). Ph. D. Thesis, Institute of Information Theory and Automation of Czechoslovak Academy of Sciences, Praha 1985.
- [10] V. Peterka: Subjective probability approach to real-time identification. Preprints 4th IFAC Symposium on Identification and System Parameter Estimation, Tbilisi 1983, Vol. 3, 83–96.
- [11] V. Peterka: Bayesian approach to system identification. In: Trends and Progress in System Identification (P. Eykhoff, ed.), Pergamon Press, Oxford 1981, 239–304.
- [12] J. Sternby: On consistency for the method of least-squares using martingale theory. *IEEE Trans. Automat. Control, AC-22* (1977), 346.
- [13] A. J. W. Van den Boom and A. W. M. Eenden: The determination of the order of process — and noise dynamics. 3rd IFAC Symposium on Identification and System Parameter Estimation, the Hague 1973.

#### 4. PRINCIPLES OF SUBOPTIMAL CONTROL STRATEGIES

Computational complexity of the resulting adaptive controller is decisive for the most of steps of its design. The computational complexity is, of course, a relative, hardware-dependent notion. According to our experience the demands of the recursive identification described in Chapter 3 are often acceptable in practice. Therefore, the control design will be compared with it in this respect.

The identification has yielded the predictive c.p.d.f.  $p(y_s(t) | t-1; u(t))$  (2.49) necessary for the optimal control design. However, no analytical solution of the dynamic programming equations (2.46), (2.47) exists in the treated case of unknown parameters with control horizon  $N > 1$ . Direct attempts to use general-purpose methods of the numerical mathematics fail due to inherent excessive dimensionality of the problem. Hence, the so-called suboptimal control strategies are used, which try to exploit the special structure of the problem. Simplifying assumptions reduce the design to the solution of a tractable linear-quadratic control problem with known parameters.

The presented description of used principles should help the user to choose a proper strategy.

#### 4.1. Informational structure of the control design

This preparatory paragraph specifies the informational structure of the optimal control design based on the regression model with incompletely known parameters.

The measured data enter the predictive c.p.d.f. (2.49) through

(i) the regressor (2.69) which will always take the form

$$(1) \quad z'(t) = [u'(t), x'(t-1)]$$

where the part  $x'(t-1)$  is the (nonminimal) state of the regression model provided its parameters are known; the exact definition of  $x$  depends on the arrangement of the data in the regressor,

(ii) the sufficient statistics (3.19) (3.6), fully determining the c.p.d.f.

$$(2) \quad \mathcal{S}(t | t-1) = (\hat{P}(t | t-1), \hat{R}(t | t-1), C(t | t-1)), \quad v(t | t-1).$$

Consequently, the conditional expectation  $E[\cdot | t-1; u(t)]$  used in the dynamic programming is a function of the current input  $u(t)$ , of the state  $x(t-1)$  and of the statistics  $\mathcal{S}(t | t-1)$ ,  $v(t | t-1)$ . To stress the explicit dependence on  $x$  and  $\mathcal{S}$  the so-called hyperstate is introduced

$$(3) \quad \mathcal{X}(t-1) = (x(t-1), \mathcal{S}(t | t-1)).$$

*Remark.* The statistics  $v(t | t-1)$  is not incorporated into the  $\mathcal{X}$  because it does not enter explicitly the treated formulae.

The above facts substantiate the following notation

$$(4) \quad \begin{aligned} E[\cdot | t-1; u(t)] &= E[\cdot | \mathcal{X}(t-1); u(t)] = \\ &= E[\cdot | x(t-1), \mathcal{S}(t | t-1); u(t)]. \end{aligned}$$

The definition of the optimal cost-to-go  $K^*$  (2.45) implies that  $K^*$  as well as the optimal control law  $L^*$  (the mapping (2.22)) depend on the known hyperstate.

*Remark.* Both  $K^*$  and  $L^*$  depend also on the admissible range  $\mathcal{U}$  (2.8) which influences the minimization step of the dynamic programming. This dependence is not important for the following discussion, therefore it will be omitted.

Using the notation (4) the dynamic programming equations (2.46), (2.47) read

$$(5) \quad \begin{aligned} K^*(t, N | \mathcal{X}(t-1)) &= \\ &= \min_{u(t) \in \mathcal{U}(t|t-1)} E[q_y(t) + q_u(t) + K^*(t+1, N | \mathcal{X}(t)) | \mathcal{X}(t-1); u(t)] \\ &\quad t = N, N-1, \dots, 1, \end{aligned}$$

$$(6) \quad K^*(N+1, N | \mathcal{X}(N)) = 0.$$

Hence, the original control problem with incompletely known parameters appears

to be a fully specified (in probabilistic terms) metaproblem of the “ordinary” stochastic optimization [1].

The difficulty of the optimization (5), (6) is mainly caused by high dimensionality of the hyperstate (13 for the first order problem). Åström [3.1] reports that the simple three-dimensional case required 180 CPU-hours on the powerful multiprocessor DEC VAX 11/780 computer. The computational demands grow exponentially with the cardinality of  $\mathcal{X}$ . Considering the present day technology it is not realistic to search for the optimal controller even in the case of simple practical problems. This necessitates suboptimal control strategies.

#### 4.2. Dual character of the optimal control; active and passive strategies

The accumulation of the experience is a dynamic process (cf. (3.24)–(3.27)) which can be substantially influenced by a proper excitation of the system (Sec. 3.2). However, the primary objective of the controller is to stabilize the closed loop, to make the overall behaviour “quiet”.

The optimal controller acting with incomplete knowledge of the model has to find a suitable compromise between a proper excitation for the identification and stabilizing effect of the control. This feature has been exemplified by Fel'dbaum [2] who called the optimal controller “dual”, i.e. having two contradictory goals.

Suboptimal control strategies which try to retain this characteristic of the optimal controller are called active. The extent of the information gained in the future is influenced actively by them.

On the contrary, passive strategies do not take explicit care of the influence of the generated actions on the future identification. They are based on the presumption, often fulfilled, that “natural” closed loop disturbances (an excitation due to the adaptation phase of the controller and/or frequent changes of the set point) will bring sufficient amount of information about unknown parameters.

*Remark.* Notice that many published simulation results avoid the problems related to insufficient excitation by using very wild, often unrealistic, command signals.

Some strategies, reported in this chapter, have the propensity to suffer from insufficient excitation. If the results of the control are suspected to be poor due to passivity of the used strategy, some of the following modifications may help:

- (i) To add a suitable term to the optimized criterion taking into account the quality of the identification in this way [3.1], [3], [4]. This approach is often computationally expensive. Moreover, the most published versions are related to the one-stage-ahead strategy with all its drawbacks.
- (ii) To inject permanently some external signal (pseudorandom, white noise) having a fraction of the energy of the signal in the closed loop to which this exciting disturbance is added. The simplicity of this way is paid by the decrease of the attainable quality of control.
- (iii) To add an external signal only at time instants when too small piece of informa-

tion has been obtained. The quantity (3.18) can be, for instance, taken as a simple measure of this information.

Our simulation tests showed the last approach as the most promising. However, we found that the danger caused by the passivity is rather low in practice. This is the reason why only the passive strategies reported below became a part of the software for a standard use.

### 4.3. Receding-horizon formulation of the adaptive control

It has been stressed and explicitly demonstrated (Sec. 2.6.3) that the controller should plan its actions over the entire control horizon. However, possible parameter changes cause that the optimal plan prepared for constant parameters becomes obsolete in the course of real time. Then it makes little sense to apply every action of this plan. It is more reasonable to correct the plan whenever the measured data bring a new piece of information.

This idea is applied in the following way: The (sub)optimal plan is designed for the next  $N$  steps taking  $p(R(t+1), P(t+1) | t)$  as the prior p.d.f. of parameters (cf. discussion of Sec. 2.7.2) because before the choice of the input  $u(t+1)$  this c.p.d.f. forms the best available description of the unknown parameters. Formally, instead of the criterion (2.26), the conditional criterion

$$(7) \quad K_0(t+1, t+N | \mathcal{X}(t)) = \frac{1}{N} E \left[ \sum_{\tau=t+1}^{t+N} (q_y(\tau) + q_u(\tau)) + \sum_{\tau=t+N-l_y+1}^{t+N} q_{sy}(\tau) + \sum_{\tau=t+N-l_u+1}^{t+N} q_{su}(\tau) | \mathcal{X}(t) \right]$$

is minimized. The first action of this plan is applied, the corresponding extended output measured and used for actualization of the c.p.d.f.  $p(R(t+1), P(t+1) | t)$ . The whole procedure is then repeated (cf. Fig. 2.5).

The horizon is permanently receding – this gives the name to such a problem formulation (sometimes the term moving horizon is used).

#### Remarks

- (i) Note that a plain replacement of the original problem by the receding-horizon formulation does not change the practical solvability of the corresponding dynamic programming. It corrects permanently the prior information, but the equations to be solved are the same. Thus further approximations are necessary. They will be dealt with in the next paragraphs.
- (ii) In spite of the fact that only the first action of the prepared plan is really applied it is still inevitable to take into account the dynamic character of the closed loop, to use a multistage criterion (cf. again Sec. 2.6.3).
- (iii) An adaptive controller based on the receding-horizon formulation is permanently at the very beginning of the control process due to applying the first planned

action only. It stresses the importance of the careful handling of transient stage (cf. discussion in Sec. 2.6.4).

- (iv) The receding-horizon formulation decreases sensitivity of the design to different approximations made. Due to repetitive corrections of the strategy, the approximation errors are prevented to accumulate.

#### 4.4. Certainty-equivalence and cautious strategies

Basic feasible strategies approximately solving the receding-horizon version of the control problem are described now. The case without restrictions on inputs is treated, the topics related to the restricted case are postponed to Section 4.6.

The notation is shortened by omitting the time indices whenever no confusion can arise. The equalities of the type  $P' z(t) = P'[u'(t), x'(t-1)]'$  are understood as the definition of the new symbols on the right-hand side ( $z'(t) = [u'(t), x'(t-1)]$ ).

The following simple example of the regulation problem for a single-input single-output first-order system (unknown parameters, no restriction) is used to illustrate the discussion.

The system is described by

$$(8) \quad y(t) = P' z(t) + \varepsilon(t) = [B_0, P_x] \begin{bmatrix} u(t) \\ x(t-1) \end{bmatrix} + \varepsilon(t) = [B_0, A_1, B_1] \begin{bmatrix} u(t) \\ y(t-1) \\ u(t-1) \end{bmatrix} + \varepsilon(t)$$

where the white noise term  $\varepsilon(t)$  is normally distributed with a dispersion  $R$ .

The criterion becomes

$$(9) \quad K_0(t+1, t+N | \mathcal{X}(t)) = \frac{1}{N} \mathbb{E} \left[ \sum_{\tau=t+1}^{t+N} (y^2(\tau) + u^2(\tau) Q_u) | \mathcal{X}(t) \right].$$

Note that the positive input penalty  $Q_u$  allows to omit the stabilizing term.

The hyperstate takes the form

$$(10) \quad \mathcal{X}(t) = (x(t); \mathcal{S}(t+1 | t)) = \left( \begin{bmatrix} y(t) \\ u(t) \end{bmatrix}; \hat{P}(t+1 | t) = \begin{bmatrix} \hat{B}_0 \\ \hat{P}_x \end{bmatrix}, \hat{R}(t+1 | t), C(t+1 | t) = \begin{bmatrix} C_u C_{ux} \\ C_{ux} C_x \end{bmatrix} \right).$$

The first step of the dynamic programming (5), (6) can be written as follows

$$(11) \quad K^*(t+N, t+N | \mathcal{X}(t+N-1)) = \min_{u(t+N)} \mathbb{E} [y^2(t+N) + u^2(t+N) Q_u | \mathcal{X}(t+N-1); u(t+N)] = \min_{u(t+N)} \{ (\hat{P}^*(t+N | t+N-1) z(t+N))^2 + u^2(t+N) Q_u + \hat{R}(t+N | t+N-1) + \hat{R}(t+N | t+N-1) z'(t+N) C(t+N | t+N-1) z(t+N) \}.$$

The expectation has been evaluated according to the formulae (2.39), (3.16), (3.17), (3.19). The minimized function consists of the squared output prediction  $\hat{P}'z$ , of the original input penalty  $u^2Q_u$ , of  $\hat{R}$  reflecting the irreducible loss due to the presence of the white noise in the model and of the term  $\hat{R}z'Cz$  reflecting the increase of the output dispersion because of the parameter uncertainty. The last term can be interpreted as an additional, the lack of information reflecting, penalty on the regressor.

The following completion of squares is advantageous to find the optimizing input as well as the minimum achieved, the optimal control law and the value of the optimal cost-to-go.

$$(12) \quad K^*(t+N, t+N | \mathcal{X}(t+N-1)) = \\ = \min_{u(t+N)} \left\{ (\hat{B}_0^2 + C_u + Q_u) \left[ u(t+N) + \frac{1}{\hat{B}_0^2 + C_u + Q_u} (\hat{P}_x + \hat{R}C_{ux})' x(t+N-1) \right]^2 + \right. \\ \left. + \hat{R} + x'(t+N-1) \left[ \hat{P}_x \hat{P}_x' - \frac{(\hat{P}_x + \hat{R}C_{ux})(\hat{P}_x + \hat{R}C_{ux})'}{\hat{B}_0^2 + C_u + Q_u} + C_x \right] x(t+N-1) \right\}.$$

The optimal input, minimizing the only term which depends on it, is

$$(13) \quad u(t+N) = -\frac{1}{\hat{B}_0^2 + C_u + Q_u} (\hat{P}_x + \hat{R}C_{ux})' x(t+N-1)$$

and the optimal cost-to-go is the quadratic form in  $x(t+N-1)$

$$(14) \quad K^*(t+N, t+N | \mathcal{X}(t+N-1)) = \\ = \hat{R}(t+N | t+N-1) + \\ + x'(t+N-1) \left[ \hat{P}_x \hat{P}_x' - \frac{1}{\hat{B}_0^2 + C_u + Q_u} (\hat{P}_x + \hat{R}C_{ux})(\hat{P}_x + \hat{R}C_{ux})' + C_x \right] x(t+N-1) = \\ = \hat{R}(t+N | t+N-1) + x'(t+N-1) S(1 | \mathcal{S}(t+N | t+N-1)) x(t+N-1).$$

#### Remarks

- (i) The value of  $K^*$  being the smallest value of the nonnegative quantity, for any  $\hat{R}$  and  $x'(t+N-1)$ , has to be nonnegative. Consequently, the kernel  $S$  has to be a non-negative definite matrix.
- (ii) Note that  $S$  depends on the data only through the statistics  $\mathcal{S}$ .

The second step of the dynamic programming reads

$$(15) \quad K^*(t+N-1, t+N | \mathcal{X}(t+N-2)) = \\ = \min_{u(t+N-1)} E[y^2(t+N-1) + u^2(t+N-1)Q_u + \hat{R}(t+N | t+N-1) + \\ + x'(t+N-1) S(1 | \mathcal{S}(t+N | t+N-1)) x(t+N-1) | \mathcal{X}(t+N-2); \\ u(t+N-1)].$$

Evaluation of the expectation of the first two terms does not differ from the preceding step. The third term makes also no difficulties because for slowly varying  $R$  it holds

$$\begin{aligned}
 (16) \quad & \mathbb{E}[\hat{R}(t+N | t+N-1) | \mathcal{X}(t+N-2); u(t+N-1)] = \\
 & = \mathbb{E}[\mathbb{E}[R(t+N) | \mathcal{X}(t+N-1)] | \mathcal{X}(t+N-2)] = \\
 & = \hat{R}(t+N | t+N-2) = \hat{R}(t+N-1 | t+N-2).
 \end{aligned}$$

The decisive difference between the first and the next steps of the dynamic programming is concentrated in the fourth term of Eq. (15).

The nonlinear dependence of  $S(1 | \mathcal{S}(t+N | t+N-1))$  on  $y(t+N-1)$  prevents the analytical evaluation of the required expectation and the nonlinear dependence on  $u(t+N-1)$  makes the minimization step extremely difficult, too.

The performed as well as unfinished computations give the direct hint how to search for feasible strategies. It is reasonable to use an approximative kernel, which behaves as a constant with respect to expectation  $\mathbb{E}[\cdot | \mathcal{X}(t+N-2); u(t+N-1)]$  and minimization over  $u(t+N-1)$  instead of  $S(1 | \mathcal{S}(t+N | t+N-1))$ . With such an approximation it makes no trouble to perform the second (approximative) step of the dynamic programming. Then, however, the same problems appear again. Hence feasibility of the  $N$ -step optimization can be achieved when approximating  $\mathcal{S}(t+i | t+i-1)$  for  $i=1, 2, \dots, N$  by a deterministic function of  $\mathcal{X}(t)$ , say  $\mathcal{S}_a(t+i | t+i-1)$ .

The discussion of the asymptotic behaviour of the identification (Sec. 3.2) implies that at least for large  $t$  and relatively short receding horizon the statistics  $\mathcal{S}(t+1 | t)$  and  $\mathcal{S}(t+1+i | t+i)$ ,  $i=1, 2, \dots, N$  are near each other; the parameter estimates and their uncertainties are near some fixed values. It suggests the following form of the approximative statistic

$$(17) \quad \mathcal{S}_a(t+1+i | t+i) = \mathcal{S}(t+1 | t) \quad \text{for } i=1, 2, \dots, N.$$

The parameter estimates and their uncertainties are frozen when planning under (17). Then it is possible to solve the approximative dynamic programming equations

$$\begin{aligned}
 (18) \quad & K_a^*(\tau, t+N | x(\tau-1); \mathcal{S}_a(\tau | \tau-1)) = \\
 & = \min_{u(\tau)} \mathbb{E}[q_\tau(\tau) + q_a(\tau) + \\
 & + K_a^*(\tau+1, t+N | x(\tau); \mathcal{S}_a(\tau+1 | \tau)) | x(\tau-1); \mathcal{S}_a(\tau | \tau-1); u(\tau)]. \\
 & \tau = t+N, t+N-1, \dots, t+1,
 \end{aligned}$$

$$(19) \quad K_a^*(t+N+1, t+N | x(t+N); \mathcal{S}_a(t+N+1 | t+N)) = 0.$$

The approximative optimal cost-to-go  $K_a^*$  keeps the quadratic form reducing the equations (18), (19) to the algebraic recursion for its kernel  $S$ .

The derived strategy (treated later in detail) differs from the known-parameter case with  $P = \hat{P}$ ,  $R = \hat{R}$  by the additional penalty  $\hat{R}z'Cz$  on the data contained in

the regressor  $z$  (cf. (11)). Roughly speaking the higher is our uncertainty about unknown parameters the more cautious are the actions of the proposed strategy. This feature gave the name to the strategy which is called cautious.

The cautious controller produces restricted actions even under poor information, overcoming the transient stages in a smooth way. This generally positive feature, however, increases the danger of an insufficient excitation. Due to the pessimistic assumption, contained in (17), that uncertainty does not decrease in the future, the strategy is too cautious.

It can be seen from the formula (3.25) that, with a forgetting factor close to unity, new data decrease the uncertainty. This observation suggests further improvement of the above approximation. The decrease of  $C$  should be simulated in some deterministic way.

The following simple rule has been proved to be advantageous

$$(20) \quad C_a(t+1+i | t+i) = \begin{cases} C(t+1 | t) & \text{for } i = 1, 2, \dots, n-1 \\ 0 & \text{for } i = n, n+1, \dots, N \end{cases}$$

$$(21) \quad \begin{aligned} \hat{R}_a(t+1+i | t+i) &= \hat{R}(t+1 | t), \\ \hat{P}_a(t+1+i | t+i) &= \hat{P}(t+1 | t) \quad \text{for } i = 1, 2, \dots, N. \end{aligned}$$

The approximation (20) expresses the confidence that uncertainty will be negligible after the next  $n$  identification steps. According to our experience the excessive actions are usually prevented even with  $n = 1$  for which the danger of passivity is substantially decreased.

Continuing in this vein the uncertainties can be fully neglected relying on the fact that, for sufficiently exciting inputs and rather slowly varying parameters, the uncertainty expressed by  $C$  will be negligible. This most confident strategy plans its actions as in the known-parameter case. It takes with certainty the most recent point estimates of the parameters to be (for a control design) equivalent to the complete system description.

This seemingly simple-minded strategy, known as certainty-equivalence strategy, proved to be rather successful. It has a strong intuitive appeal replacing unknown constants by their best available estimates and it is the most widespread one. Moreover, in comparison with the cautious controller the related computations are simpler. Sometimes the strategy may be too confident causing excessive overshoots in the transient stage. The excessive signals disappear rather quickly because they will excite the closed loop.

A careful choice of the prior distribution  $p(R(1), P(1) | 0)$  (Sec. 3.3) helps to keep the closed-loop signals in an acceptable range.

This measure can be supported by the input restriction (Sec. 4.6) as well as by the use of the cautious strategy.

#### 4.5. Strategy of iterations spread in time

Looking for further improvement of the described strategies we shall start by discussing of their drawbacks related to the choice of the horizon  $N$ .

A reasonable horizon would depend on the identification stage. It makes little sense to search for the "precise" values of suboptimal actions having a poor information about unknown parameters. It wastes the computing time and may be even dangerous especially in conjunction with certainty-equivalence strategy (overshoots). On the contrary, having a good information at disposal the receding horizon should form as high fraction of the entire control period as possible.

The available computing time, however, may prevent finishing a sufficient number of steps of the dynamic programming, decreasing the quality of the control achieved (the extreme cases are illustrated by the example of Sec. 2.6.3). This may be of a practical importance because there are some control problems requiring hundreds of steps of the dynamic programming. Fortunately, a lot of systems is satisfactorily controlled with receding horizon less than ten steps. Then the strategies of the preceding section are directly applicable. If it is not the case then the following improvement overcomes the problems mentioned.

Let  $\mathcal{R}$  denote the mapping reflecting the evolution of kernel  $S$  of the quadratic form forming the approximative cost-to-go, i.e.

$$(22) \quad \begin{aligned} S_i | \mathcal{S}_d(t + N - (i - 1) | t + N - i) &= S(i | \mathcal{S}(t + 1 | t)) = \\ &= \mathcal{R}(S_{i-1} | \mathcal{S}(t + 1 | t)), \mathcal{S}(t + 1 | t) \quad i = 1, 2, \dots, N. \end{aligned}$$

It can be shown [5] that for the models describing stabilizable systems this mapping is a contraction;  $S_i | \mathcal{S}(t + 1 | t)$  converges for growing  $i$  to a stationary value  $S^*$  solving the so-called stationary discrete Riccati equation

$$(23) \quad S^*(\mathcal{S}(t + 1 | t)) = \mathcal{R}(S^*(\mathcal{S}(t + 1 | t)), \mathcal{S}(t + 1 | t)).$$

As it will be demonstrated later, the determination of  $S^*$  forms the difficult part of the control design. Knowing  $S^*$  it is an easy task to evaluate the corresponding control law.

The original dynamic-programming-based equation (22) can be viewed as the successive approximations method applied to the solution of the stationary Riccati equation (23). It is known that the number of steps  $N$  required by this method to achieve the stationary value  $S^*(\mathcal{S})$  strongly depends on the initial value  $S(0 | \mathcal{S})$  used in (22).

Intuitively, the best available guess at (real) time  $t + 1$  is the final value gained at time  $t$ . Under mild assumptions, briefly discussed below, it can be shown [6] that the single step at each (real) time instant is sufficient to produce inputs stabilizing the closed loop. Whenever the inputs are sufficiently exciting the resulting strategy is asymptotically optimal.

Denoting by  $S(t)$  the matrix generated by this strategy (approximating

$S^*(\mathcal{S}(t | t - 1))$  the formal description of the design reads

$$(24) \quad S(t + 1) = \mathcal{R}(S'(t), \mathcal{S}(t + 1 | t))$$

where  $t$  is the real time. Notice that for  $\mathcal{S}(t + 1 | t)$  (almost) equal to a stationary value  $\mathcal{S}$  the recursions (22), (24) coincide. The iterations made at one time instant in (22) are spread in real time when using (24); this gives the name to this strategy: Iterations Spread in Time (IST).

#### Remarks

- (i) One step of the dynamic programming requires approximately the same computational effort as one step of the recursive least squares. The identification and design parts are balanced at a reasonable low level of the computational complexity.
- (ii) Tests made (incomplete theoretical analysis, extensive simulations, pilot plant as well as full scale use) proved high efficiency of this and of the former described strategies.
- (iii) Simulations showed that the rates of convergence of the identification part and of iterations (24) are usually similar. If it is not the case more then one step of the successive approximations can be performed at one real time instant. In this way, also the inevitable higher sensitivity of the IST strategy to parameter changes can be suppressed.
- (iv) The two iterative processes are combined in (24): identification and successive approximations. It is known (from the theory of multiple limits) that independence of the asymptotic value on the way in which these two processes are combined can be achieved under the uniform convergence condition. It is not generally guaranteed here due to nonuniqueness of the solution of the stationary Riccati equation. The danger caused by the nonuniqueness arises whenever some signal is not penalized in the basic part of the criterion. Then only such initial conditions of (22) guarantee  $S^*(\mathcal{S}(t + 1 | t))$  to yield the control law stabilizing the closed loop, which can play the role of the stabilizing term in the criterion (Sec. 2.6.2), [8]. Even if the initial condition of (24) is properly chosen it may happen that some  $S'(t)$  loses the stabilizing property; the rank of  $S'(t)$  falls down too much causing closed loop instability. It is an easy task to prevent this danger:
  - to penalize all closed loop signals
  - to use cautious version of the strategy, where signals are penalized by the matrix  $C$  (cf. (11) but see remark (v))
  - to use data-dependent penalties which are nonzero when necessary; the nontrivial example represents the modified IST strategy (MIST) developed in [7] and briefly discussed in the next paragraph.
- (v) The use of the cautious version suffers from passivity. High uncertainties may switch off the controller for a longer period, because, essentially, the most cautious version corresponding to the long planning horizon is used.

#### 4.6. Restriction on admissible range of inputs

The presence of hard restrictions on admissible inputs, inherent to the majority of the real problems, makes the control design difficult even in the case of known parameters. The following approaches to this problem are of a practical importance:

- (i) Trial-and-error based choice of the input penalty  $Q_u$  forcing  $u(t)$  to stay in the admissible range. In spite of monotonous dependence of action range on  $Q_u$ , an increase of  $Q_u$  implies a decrease of  $u$  (or its changes), this way may be rather time-consuming. Moreover, a conservative choice of the penalty is often necessary. The fixed  $Q_u$  has to cover the worst possible cases.
- (ii) Modification of the unrestricted design. The unrestricted input is cut off whenever it exceeds the boundary of the admissible range. This approach is often successful for single-input systems. Its straightforward application to multi-input cases does not work satisfactorily, cutting off of one input should be followed by an appropriate change of the other inputs (see Fig. 1). A simple problem of the quadratic programming [9] of the type

$$(25) \quad \min_{u \in \mathcal{U}} (u - u^*)' \bar{Q}_u (u - u^*)$$

has to be solved ( $u^*$  denotes the unrestricted optimal input and  $\bar{Q}_u$  the positive semidefinite kernel compressing all penalties on input in the final step of the dynamic programming).

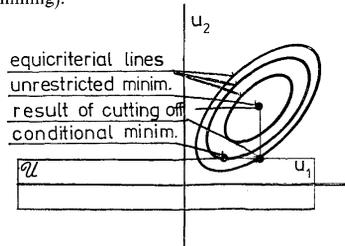


Fig. 1. Restrictions in multivariate case.

The danger of passivity arises whenever cutting off occurs too often. Thus, this way has to be usually combined with a proper choice of  $Q_u$ .

- (iii) The use of data-dependent penalties. Observing of the drawbacks of the above approach led to a simple modification of the IST strategy (MIST) [7]. The IST strategy is essentially used and the optimization of the type (25) performed in each step corresponding to the choice of the really applied input (in each step when the version of IST is used). Kuhn-Tucker multipliers are taken as "natural" data-dependent candidates of penalties necessary to keep input inside the set  $\mathcal{U}$  for states  $x$  similar to the current one.

The most simple version which hopes the two consecutive states to be similar is described in the paper cited. This relatively new strategy passed successfully simulation tests but theoretical analysis as well as practical experience are lacking up to now. This idea of the MIST strategy relaxes the necessity to choose  $Q_u$  beforehand and its use admits to cover fully the available action range. Moreover, it automatically prevents the matrix  $S(t)$  (24) to lose its sufficient rank.

#### 4.7. Summary of strategies based on receding horizon formulation

point of view	name	characterized by	comments
gaining of information	active	attempt to influence the future informational gain	see [2], [3], [4]
	passive	belief in the sufficient excitation due to adaptation and reset stages	usually satisfactory
approximation for numerical feasibility	cautious	freezing of the sufficient statistics throughout planning	related danger of passivity suppressed when fall-down of uncertainties is presumed
	certainty equivalence	freezing of the parameter estimate throughout planning; uncertainties omitted	the most widespread; danger of overshoots
computation	simple controllers	explicit determination of the stationary controller	SISO systems up to second order
	receding horizon	direct use of the cautious and/or certainty equivalence versions of the receding-horizon formulation	available computing time restricts the planning horizon; not too sensitive to parameter changes
	iterations spread in time (IST)	use of optimization results gained at time $t$ as initial values for planning at time $t + 1$	computationally simple; more sensitive to parameter changes; stabilizing property has to be checked
input restrictions	trial-and-error	trial-and-error choice of the proper input penalty	time-consuming; worst-case oriented
	cutting-off	unrestricted design followed by input cutting-off	simple; in multivariate case quadratic programming necessary; danger of passivity
	modified iterations spread in time (MIST)	use of data-dependent input penalties in IST strategy; Kuhn-Tucker multipliers a proper candidate	promising; incompletely tested

## REFERENCES

---

- [1] J. J. Florentin: Optimal probing, adaptive control of a simple Bayesian system. *Internat. J. Control* 13 (1971), 165.
- [2] A. A. Fel'dbaum: Dual control theory I–IV. *Automat. Remote Control* 21 (1960), 874; 21 (1960) 1033; 22 (1969), 1; 22 (1969), 109.
- [3] B. Wittenmark: Stochastic adaptive control methods — a survey. *Internat. J. Control* 21 (1975), 705.
- [4] J. Sternby: Extremum control systems — an area for adaptive control. *Proc. JACC American Automation Council* (1980).
- [5] G. A. Hewer: Analysis of a discrete matrix Riccati equation of linear control and Kalman filtering. *J. Math. Anal. Appl.* 42 (1973), 1.
- [6] J. Böhm and M. Kárný: Asymptotically optimal control strategy of systems with unknown but constant parameters (in Czech). *Conf. about Cybernetics, Prague, I* (1976), 9–25.
- [7] J. Böhm and M. Kárný: Self-tuning regulators with restricted inputs. *Kybernetika* 18 (1982), 6, 529–544.
- [8] H. Nováková: Contribution to the Optimal Control of Discrete-Time Stochastic Systems (in Czech). Ph. D. Dissertation, Institute of Information Theory and Automation of the Czechoslovak Academy of Sciences, Prague 1978.
- [9] G. Zoutendijk: *Mathematical Programming Methods*. North-Holland Publishing Company. Amsterdam—New York—Oxford 1976.
- [10] M. Kárný: Suboptimal strategies for the control of stochastic systems with constant but unknown parameters. Preprints SOCOCO'79, the 2nd IFAC/IFIP Symposium on Software for Computer Control, Prague, II, (1979), AX1—AX4.

## 5. CONTROL SYNTHESIS — A UNIVERSAL ALGORITHM

Broad formulation of the problem involving many types of data (inputs, outputs, disturbances, reference values, command signals) and several approaches how to make use of them (the idea of the receding horizon, the certainty equivalence and the cautious strategies, various ways of approximation) bring about apparent complexity of formulae and algorithms for the control synthesis. The aim of this chapter is to give a general algorithm which covers in a unified way most of the cases conceptually solved in the previous chapters. It is suitable for and flexible in simulation studies for research and education purposes. The universality of the algorithm is, of course, paid for by waste of computations. A more economical way of algorithmization will be outlined in Chapter 6.

After suitable rearrangement of the formulae for the model and the criterion the crucial steps of the synthesis will be shown now. Then the resulting algorithm will be given with the list of variables involved, including references to the definitions and relevant considerations in the previous text.

### 5.1. Compact model

The need for a model necessary to specify the control strategy was discussed in Sec. 2.8. Here we shall take for granted that suitable predictive models of all uncertain variables were chosen and their parameters identified at the time  $t - 1$  using

e.g. the algorithms described in Chapter 3. To perform the dynamic programming procedure, the one-step-ahead prediction of the extended output vector  $y'_x(t) = [y'(t), u'_0(t), y'_0(t), v'(t)]'$  is necessary. Let us express the regression model for  $y'_x(t)$  in the form

$$(1) \quad y_x(t) = B_0 u(t) + P'_x x(t-1) + \varepsilon(t), \quad \text{cov}(\varepsilon) = R.$$

The choice of the partial models has been discussed in Sec. 2.5.1 ( $y_0(t), u_0(t)$ ) and 2.8.3. ( $y(t), v(t)$ ). As an example of the full model (1) construction we shall suppose the following rather general case:

Regression model of the controlled system (identified)

$$(2) \quad y(t) = B_{y0} u(t) + P'_{yx} x_y(t-1) + \varepsilon_y(t), \quad \text{cov}(\varepsilon_y) = R_y.$$

Penalization of input increments (optimal requirement)

$$(3) \quad u_0(t) = u(t-1).$$

Model following (model chosen in advance)

$$(4) \quad y_0(t) = B_m u(t) + P'_{mx} x_m(t-1) + \varepsilon_m(t), \quad \text{cov}(\varepsilon_m) = R_m.$$

Regression model of a disturbance (identified, ev. modelled as generalized random walk)

$$(5) \quad v(t) = P'_v x_v(t-1) + \varepsilon_v(t), \quad \text{cov}(\varepsilon_v) = R_v.$$

Summarizing these into the form (1), we obtain the relation

$$(6) \quad y'_x(t) = \begin{bmatrix} B_{y0} \\ 0 \\ B_{m0} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} P'_x & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & P'_{mx} & 0 \\ 0 & 0 & 0 & P'_v \end{bmatrix} \begin{bmatrix} x_y(t-1) \\ u(t-1) \\ x_m(t-1) \\ x_v(t-1) \end{bmatrix} + \begin{bmatrix} \varepsilon_y(t) \\ 0 \\ \varepsilon_m(t) \\ \varepsilon_v(t) \end{bmatrix},$$

$$R = \begin{bmatrix} R_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & R_m & 0 \\ 0 & 0 & 0 & R_v \end{bmatrix}.$$

Remarks. (i) The arrangement of the state vector elements is, course, optional; it depends on the programmer's will in defining the extended model state in a convenient way.

(ii) In most practical cases (6) simplifies substantially. E. g., the stabilization problem means  $y_0(t) = y_0$ , thus  $B_{m0} = 0$ ,  $P'_{mx} = y_0$ ,  $x_m(t-1) = 1$ ,  $\varepsilon_m(t) = 0$ ,  $R_m = 0$ .

(iii) The transport delay  $t_u$  is taken into account (alternatively to (2.66)) by supposing the regression without delay and equationing  $B_i = 0$ ,  $i = 0, 1, \dots, t_u - 1$ . This approach implies the appropriate increase of  $l_u$ , the new one being  $l_{un} = l_u + t_u$ .

(iv) The case, when the output reference signal  $y_0(t)$  is preprogrammed  $N_p$  steps in advance, must be given special attention, because distinguishing the real time  $t$  from the planning time  $\tau$  within the planning horizon  $N$  is necessary.

## 5.2. Optimality criterion

The receding horizon idea outlined in Chapter 4 implies the minimization of the criterion (4.7) within the time interval  $t + 1, \dots, t + N$  (under the condition that most recent available data are included) and the application of the first control action only. As the same procedure is repeated in every time instant  $t$ , we can take in the sequel  $t = 0$  for simplicity of formulae, if not stressed otherwise. The criterion (4.7) then reads

$$(7) \quad K_0(1, N | \mathcal{X}(0)) = \frac{1}{N} \mathbb{E} \left[ \sum_{\tau=1}^N (q_y(\tau) + q_u(\tau)) + J_s | \mathcal{X}(0) \right],$$

$$(8) \quad J_s = \sum_{\tau=N-I_y+1}^N q_{sy}(\tau) + \sum_{\tau=N-I_u+1}^N q_{su}(\tau),$$

where the hyperstate defined by (4.3) is

$$(9) \quad \mathcal{X}(0) = (x(0); \mathcal{S}(1 | 0)).$$

Careful inspection of the stabilizing term  $J_s$  shows that it involves only the data included in the final model state  $x(N)$  (cf. (1) and (2.19)), thus it can be expressed by a quadratic form with a properly chosen kernel  $Q_{sx}$  constructed from  $Q_{sy}$ ,  $Q_{su}$  (2.19). Then

$$(10) \quad J_s = x'(N) Q_{sx} x(N).$$

Recalling (2.62), (2.62a) and (1) we can write in a similar way

$$(11) \quad q_y(\tau) + q_u(\tau) = x'(\tau) Q_x x(\tau),$$

where the block-diagonal matrix  $Q_x$  has been introduced,

$$(12) \quad Q_x = \text{block-diag} [Q_d, 0].$$

With this notation the criterion (7), (8), (9) turns into the concise formula

$$(13) \quad K_0(1, N | \mathcal{X}(0)) = \frac{1}{N} \mathbb{E} \left[ \sum_{\tau=1}^N x'(\tau) Q_x x(\tau) + x'(N) Q_{sx} x(N) | x(0), \mathcal{S}(1 | 0) \right].$$

When minimizing the criterion (13) by dynamic programming, approximative in the sense outlined in Sec. 4.4, we have to go through the recursion (4.18), (4.19)

$$(14) \quad K_a^*(\tau, N | x(\tau-1), \mathcal{S}_a(\tau | \tau-1)) = \min_{u(\tau) \in \mathcal{U}} \mathbb{E} [x'(\tau) Q x(\tau) +$$

$$\begin{aligned}
& + K_a^*(\tau + 1, N | x(\tau), \mathcal{S}_a(\tau + 1 | \tau)) | x(\tau - 1), \mathcal{S}_a(\tau | \tau - 1); u(\tau) \\
& \quad \tau = N, N - 1, \dots, 1, \\
& K_a^*(N + 1, N | x(N), \mathcal{S}_a(N + 1 | N)) = 0.
\end{aligned}$$

Respecting our convention  $t = 0$ , the approximation (4.14) reads

$$(15) \quad \mathcal{S}_a(\tau + 1 | \tau) = \mathcal{S}(1 | 0), \quad \tau = 1, 2, \dots, N.$$

For the recursion we shall need the first two approximative conditional moments of  $x(\tau)$ .

Separating the new data  $u(\tau)$ ,  $y_x(\tau)$  and shifting the older ones within the previous state  $x(\tau - 1)$  we can write

$$(16) \quad x(\tau) = M_u u(\tau) + M_y y(\tau) + M_x x(\tau - 1),$$

where matrices  $M_u$ ,  $M_y$  and  $M_x$  of appropriate dimensions are simply constructed from identity and zero matrices only. The definition (4.2) of the statistic  $\mathcal{S}(1 | 0)$  gives

$$(17) \quad \mathcal{S}(1 | 0) = (\hat{P}(1 | 0), \hat{R}(1 | 0), C(1 | 0)).$$

Substituting (1) into (16) we can compute the first conditional moment

$$\begin{aligned}
(18) \quad & \mathbb{E}[x(\tau) | x(\tau - 1), \mathcal{S}_a(\tau + 1 | \tau); u(\tau)] = \mathbb{E}[x(\tau) | x(\tau - 1), \mathcal{S}(1 | 0); u(\tau)] = \\
& = [M_u + M_y \hat{B}_0, M_y \hat{P}_x + M_x] \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix} = \hat{M} \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix}
\end{aligned}$$

where the estimates (17) as well as the related matrix  $\hat{M}$  are shortly denoted by the hat symbol. Then using (2.63), (3.17) and (4.1) we have the second conditional moment

$$\begin{aligned}
(19) \quad & \mathbb{E}[x'(\tau) Q x(\tau) | x(\tau - 1), \mathcal{S}_a(\tau + 1 | \tau); u(\tau)] = \\
& = \mathbb{E}[x'(\tau) Q x(\tau) | x(\tau - 1), \mathcal{S}(1 | 0); u(\tau)] = \\
& = \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix}' \hat{M}' Q \hat{M} \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix} + \text{tr} [Q \text{cov}(M_y y_x(\tau) | x(\tau - 1), \mathcal{S}(1 | 0); u(\tau))] \\
& = \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix}' (\hat{M}' Q \hat{M} + C \text{tr} [Q M_y \hat{R} M_y']) \begin{bmatrix} u(\tau) \\ x(\tau - 1) \end{bmatrix} + \text{tr} [Q M_y \hat{R} M_y'].
\end{aligned}$$

### 5.3. Optimization

The dynamic programming equations (14) under the condition (15) (omitting for brevity the fixed condition  $\mathcal{S}(1 | 0)$  and the index  $a$ ) have, for the criterion (13) and no restrictions on  $u(\tau)$ , the form

$$\begin{aligned}
(20) \quad K^*(\tau, N \mid x(\tau - 1)) &= \min_{u(\tau)} E[x'(\tau) Q x(\tau) + \\
&+ K^*(\tau + 1, N \mid x(\tau)) \mid x(\tau - 1); u(\tau)], \\
\tau &= N, N - 1, \dots, 1. \\
Q &= \begin{cases} Q_x + Q_{xx} & \text{for } \tau = N, \\ Q_x & \text{for } \tau < N, \end{cases} \\
K^*(N + 1, N \mid x(N)) &= 0.
\end{aligned}$$

It will be shown that the functions  $K^*$  have a self-reproducing form

$$(21) \quad K^*(\tau + 1, N \mid x(\tau)) = x'(\tau) S(N - \tau) x(\tau) + \gamma(N - \tau),$$

where  $S$  is a nonnegative definite matrix of appropriate dimensions,  $\gamma$  is a nonnegative scalar. With the fixed  $\mathcal{S}(1 \mid 0)$  omitted, the functional recursion (4.18) turns into the algebraic recursion for  $S(N - \tau)$  and  $\gamma(N - \tau)$ :

$$\begin{aligned}
(22) \quad &x'(\tau - 1) S(N - \tau + 1) x(\tau - 1) + \gamma(N - \tau + 1) = \\
&= \min_{u(\tau)} E[x'(\tau) (Q + S(N - \tau)) x(\tau) + \gamma(N - \tau) \mid x(\tau - 1); u(\tau)], \\
\tau &= N, N - 1, \dots, 1, \\
S(0) &= 0, \\
\gamma(0) &= 0.
\end{aligned}$$

For better understanding of the algorithm we shall go through one step of this recursion. Supposing that the solution of (20) for  $\tau + 1$  has the form (21) we shall use it for the subsequent step  $\tau$ :

$$\begin{aligned}
K^*(\tau, N \mid x(\tau - 1)) &= \min_{u(\tau)} E[x'(\tau) Q x(\tau) + \\
&+ x'(\tau) S(N - \tau) x(\tau) + \gamma(N - \tau) \mid x(\tau - 1); u(\tau)].
\end{aligned}$$

The expectation step is performed directly by using the formula (19)

$$\begin{aligned}
(23) \quad K^*(\tau, N \mid x(\tau - 1)) &= \gamma(N - \tau) + \text{tr} [(Q + S(N - \tau)) M_y \hat{R} M_y'] + \\
&+ \min_{u(\tau)} \left[ \begin{matrix} u(\tau) \\ x'(\tau - 1) \end{matrix} \right]' \{ \hat{M}' (Q + S(N - \tau)) \hat{M} + \\
&+ C \text{tr} [(Q + S(N - \tau)) M_y \hat{R} M_y'] \} \left[ \begin{matrix} u(\tau) \\ x'(\tau - 1) \end{matrix} \right].
\end{aligned}$$

Now we shall use the Cholesky decomposition (mentioned already in Sec. 3) of a symmetric nonnegative definite matrix into lower and upper triangular factors. Such a decomposition not only assures the necessary condition of definiteness throughout the iteration, but also substantially simplifies the minimization, as becomes

apparent in the subsequent computations. We shall denote the kernel of the minimized quadratic form in (23) by  $W(N - \tau + 1)$  and assume its decomposition performed (cf. [1.2] for the indispensable generalization) in the form

$$(24) \quad W(N - \tau + 1) = \hat{M}'(Q + S(N - \tau)) \hat{M} + C \operatorname{tr} [(Q + S(N - \tau)) M_y \hat{R} M_y'] = \\ = \begin{bmatrix} \Phi(N - \tau + 1) & 0 \\ \Gamma(N - \tau + 1) & \Psi(N - \tau + 1) \end{bmatrix} \begin{bmatrix} \Phi(N - \tau + 1) & \Gamma(N - \tau + 1) \\ 0 & \Psi(N - \tau + 1) \end{bmatrix}$$

where the dimensions of blocks correspond with the dimension  $m_u$  of  $u(\tau)$  and  $i_x$  of  $x(\tau - 1)$  in (23).

With the decomposition (24) the cost-to-go (23) reads

$$(25) \quad K^*(\tau, N | x(\tau - 1)) = \gamma(N - \tau) + \operatorname{tr} [(Q + S(N - \tau)) M_y \hat{R} M_y'] + \\ + x'(\tau - 1) \Psi(N - \tau + 1) \Psi(N - \tau + 1) x(\tau - 1) + \\ + \min_{u(\tau)} \{ (\Phi(N - \tau + 1) u(\tau) + \Gamma(N - \tau + 1) x(\tau - 1))' \cdot \\ \cdot (\Phi(N - \tau + 1) u(\tau) + \Gamma(N - \tau + 1) x(\tau - 1)) \}.$$

All terms on the right-hand side are nonnegative and the minimal value is obviously obtained for the argument  $u^*(\tau)$  zeroing the last term, i.e. fulfilling the relation

$$(26) \quad \Phi(N - \tau + 1) u^*(\tau) + \Gamma(N - \tau + 1) x(\tau - 1) = 0.$$

The optimal control signal  $u^*(\tau)$  may be obtained (for the known state  $x$ ) by solving this triangular matrix equation (of small dimension  $m_u$ ).

It follows immediately from comparison of (22) and (25) that

$$(27) \quad \gamma(N - \tau + 1) = \gamma(N - \tau) + \operatorname{tr} [(Q + S(N - \tau)) M_y \hat{R} M_y'],$$

$$(28) \quad S(N - \tau + 1) = \Psi'(N - \tau + 1) \Psi(N - \tau + 1).$$

Notice that the nonnegative definiteness of  $S$  is guaranteed by this way of numerical computation.

The value of the optimality criterion is not used for the real-time control, but it is useful for possible comparison of different algorithms. It is given by the formula

$$(29) \quad K_0^*(1, N | x(0)) = \frac{1}{N} K^*(1, N | x(0)) = \frac{1}{N} \gamma(N) + \frac{1}{N} x'(0) S(N) x(0).$$

The criterion  $K_0^*$  is cost-to-go  $K^*$  per step and the last term in (29) becomes negligible for  $N \rightarrow \infty$  and  $x(0)$  finite (cf. Sec. 2.6.4), because  $S(N)$  reaches its steady-state value (for stabilizable system [4.5]) and  $\gamma(N)$  increases (cf. (27)).

**Remarks.** (i) As stressed at the beginning of this section, the case of restricted inputs is not discussed. Hints for various solutions of the problem with restrictions have been given in Sec. 4.6, algorithmical implementation will be outlined in Chapter 6.

(ii) Notice that the certainty-equivalence strategy gives the optimal input  $u^*(\tau)$  independent of  $\hat{R}$  (it is needed only for calculating the minimum of the criterion).

(iii) The transportation lag should be respected when choosing the control horizon,  $N > l_m + l_u + l_v$ .

(iv) When any of the absolute terms contained in the partial states  $x_y$ ,  $x_m$  or  $x_v$  is penalized by the chosen matrices  $Q_x$ ,  $Q_{sx}$  (it has no physical meaning, of course), the corresponding elements of the  $S$  matrix are permanently growing. Therefore the penalization of absolute terms should not be used (not even through uncertainties in the cautious strategy).

(v) When starting the iteration (20), the possibility of the following rearrangement is obvious:  $Q = Q_x$  for every  $\tau$ ,  $S(0) = Q_{sx}$ .

(vi) In the case of program control (cf. Sec. 2.5.1) known future values  $y_0(t)$  of output reference are specified  $N_p$  steps in advance. Then the simplifying convention  $t = 0$  introduced in Sec. 5.2 is not advantageous, as real time  $t$  must be respected. The computation of  $u(t_0 + 1)$  at the real time moment, say  $t_0$ , must include the proper values of  $y_0$ , i.e.  $y_0(t_0 + \tau)$ ,  $\tau = 1, 2, \dots, N_p$ . These values enter sequentially the matrix  $P_x^i$  in  $\hat{M}$  (cf. 18) and make  $\hat{M}$  time-dependent in the course of the recursion.

(vii) The future behaviour has to be specified within the whole horizon  $N$ . In case  $N_p < N$  (for  $N_p$  see remark (vi)), the generalized random walk model analogous to (2.78), (2.79) may be used in the interval  $N_p < \tau \leq N$ .

(viii) The horizon length  $N$  necessary to find the stationary solution for the matrix  $S$  ranges from one to several hundreds of steps (depending on the system properties, not known in advance). It proved to be advantageous to compute the  $u(1)$  vector in every step of the recursion as if the desired horizon length had been just attained, and stop the procedure, when the two successive results do not differ substantially.

#### 5.4. Algorithm for the control synthesis

Inputs  $x(0)$ ,  $M_y$ ,  $\hat{M}$ ,  $\hat{R}$ ,  $C$ ,  $N$ ,  $Q_x$ ,  $S(0)$ , *key*

Outputs  $u^*(1)$ ,  $K_0^*$ .

$\tau = N, N - 1, \dots, 1$

$$\begin{aligned} \alpha(N - \tau) &= \text{tr} [(Q_x + S(N - \tau)) M_y \hat{R} M_y'] \\ \gamma(N - \tau + 1) &= \gamma(N - \tau) + \alpha(N - \tau) \\ W(N - \tau + 1) &= \hat{M}' (Q_x + S(N - \tau)) \hat{M} + \text{key } C \alpha(N - \tau) = \\ &= \begin{bmatrix} \Phi'(N - \tau + 1) & 0 \\ \Gamma'(N - \tau + 1) & \Psi'(N - \tau + 1) \end{bmatrix} \times \\ &\times \begin{bmatrix} \Phi(N - \tau + 1) & \Gamma(N - \tau + 1) \\ 0 & \Psi(N - \tau + 1) \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
S(N - \tau + 1) &= \Psi'(N - \tau + 1) \Psi(N - \tau + 1) \\
\Phi(N - \tau + 1) u^*(\tau) + \Gamma(N - \tau + 1) x(\tau - 1) &= 0 \\
K_0^* &= \frac{1}{N} (\gamma(N) + x'(0) S(N) x(0)) \\
\Phi(N) u^*(1) + \Gamma(N) x(0) &= 0
\end{aligned}$$

Summary of symbols

State

$x(0)$  ( $i_x$ )-vector;  $i_x = i_{xy} + i_{xu_0} + i_{xy_0} + i_{xv}$  def. (5.1); vector involving the past history of the extended output  $y_x$  up to the time  $t$  (for brevity  $t = 0$ ) when the last step of identification was performed

Results of modelling

$M_y$  ( $i_x, i_y$ ) matrix; def. (5.16); auxiliary matrix constructed of identity and zero matrices according to extended output def. (5.1)

$\tilde{M}$  ( $i_x + m_u, i_x$ ) matrix; def. (5.18); auxiliary matrix involving parameters of model (5.1), identified or given in advance (cf. Remark 5.3 (vi))

$\tilde{R}$  ( $i_y, i_y$ ) matrix; def. (5.1); covariance of the extended output

$C$  ( $i_x + m_u, i_x + m_u$ ) matrix; def. (3.11); matrix characterizing uncertainty (needed for cautious strategy only)

Quantities related to criterion and strategy

$N$  scalar; def. (2.17); control horizon

$Q_x$  ( $i_x, i_x$ ) matrix; def. (5.11), (5.12); state penalization

$S(0)$  ( $i_x, i_x$ ) matrix; for receding horizon  $S(0) = Q_{xx}$  def. (5.10), for IST strategy  $[S(0)]_{t=0} = [S(N)]_{t=N-1}$  (cf. Sec. 4.5).

$key$  binary variable; key for the choice of strategy: certainty-equivalence ( $key = 0$ ), cautious ( $key = 1$ ), mixed ( $key = 0$  for  $\tau = N, N - 1, \dots, n + 1$ ,  $key = 1$  for  $\tau = n, n - 1, \dots, 1$ ), (cf. Sec. 4.4).

Outputs

$u^*(1)$  ( $m_u$ ) vector; the computed control signal

$K_0^*$  scalar; def. (5.29); value of the criterion.

## 6. CONTROL SYNTHESIS – SQUARE-ROOT ALGORITHMIZATION

This chapter presents another way of algorithmizing the control synthesis with the aim to achieve higher efficiency and numerical stability of computations. It summarizes ten-years experience with the factorization methods for the LQ optimum control synthesis. Similarly as in Chapter 5, the synthesis is based on dynamic programming but the nonnegative definite matrix  $S$  characterizing the cost-to-go (4.18), (5.20) is propagated throughout the whole recursion in the factorized form. The idea of the approach is to guarantee numerically nonnegative definiteness of the minimized quadratic forms even when computing with reduced word length.

It should be emphasized that the computational demands may vary heavily with the algorithmical organization chosen for the given control problem. To cope with this nonuniqueness, a general solution is outlined below admitting to choose a proper version of the algorithm with respect to the problem solved.

The chapter is organized as follows. First the algorithm is described conceptually and the crucial operations of orthogonal transformations and matrix factorization are briefly sketched. Then the modifications required to cover variety of control problems and/or strategies are discussed.

### 6.1. Main ideas of the square-root-based design

The optimized criterion can be expressed in terms of the global errors  $e_g$  (2.10), (2.11)

$$(1) \quad K_0^* = \frac{1}{N} \min_{u(1), \dots, u(N)} E[e_g' Q_g e_g | 0] = \min_{u(1), \dots, u(N)} \frac{1}{N} E[(e^0)' e^0 | 0],$$

where the vector of weighted global errors  $e^0$  is defined as

$$(2) \quad e^0 = Q_g^{1/2} e_g$$

by means of any square-root factor  $Q_g^{1/2}$  of the weighting matrix  $Q_g \geq 0$ , i.e. any matrix fulfilling the relation

$$(3) \quad Q_g = (Q_g^{1/2})' Q_g^{1/2}$$

Notice that the factor  $Q_g^{1/2}$  is not unique: if  $Q_a$  is a factor of  $Q_g$ , then  $Q_s = TQ_a$  is also a factor of  $Q_g$ , supposing the matrix  $T$  is orthogonal, i.e.  $T'T = I$  (the unit matrix).

Applying the chain rule (2.36) to the conditional expectation and using the possibility to interchange the expectation and minimization [1], the criterion (1) can be rewritten

$$(4) \quad K_0^* = \frac{1}{N} \min_{u(1)} E[ \dots E \min_{u(N)} E[(e^0)' e^0 | N-1; u(N)] | N-2; u(N-1)] \dots | 0]$$

To make the successive effect of particular expectation and minimization operations

on the quadratic form  $(e^0)' e^0$  more transparent, we use the following trick. We express the weighted global errors  $e^0$  formally by means of a vector  $\mathbf{z}$  consisting of all data  $d(1 \dots N)$  related to the control horizon

$$(5) \quad e^0 = H^0 \mathbf{z}$$

By this way, the original quadratic form  $(e^0)' e^0$  is converted into a quadratic form in  $\mathbf{z}$

$$(6) \quad (e^0)' e^0 = \mathbf{z}' (H^0)' H^0 \mathbf{z}$$

Performing the inmost expectation in (4) (within the class of strategies used – cf. Chapter 4), we arrive at a new quadratic form in  $\mathbf{z}$

$$(7) \quad \mathbb{E}[\mathbf{z}' (H^0)' H^0 \mathbf{z} \mid N-1; u(N)] = \mathbf{z}' (H^1)' H^1 \mathbf{z}$$

Notice that the conditional expectation formally changes only the kernel of the quadratic form

$$(8) \quad \mathbb{E}[\cdot \mid N-1; u(N)] : \mathbf{z}' (H^0)' H^0 \mathbf{z} \rightarrow \mathbf{z}' (\tilde{H}^1)' \tilde{H}^1 \mathbf{z}$$

By using a suitable orthogonal transformation  $T$

$$(9) \quad \mathbf{z}' (\tilde{H}^1)' T' T \tilde{H}^1 \mathbf{z} = \mathbf{z}' (H^1)' H^1 \mathbf{z}$$

we can ensure that  $u(N)$  affects only  $m_u$  entries of the vector  $e^1 = H^1 \mathbf{z}$ , indexed by  $k_j, j = 1, 2, \dots, m_u$ . Then the minimization

$$(10) \quad \min_{u(N)} \{ \mathbf{z}' (H^1)' H^1 \mathbf{z} \}$$

is achieved immediately by solving the equations

$$e_{k_j}^1 = 0, \quad j = 1, 2, \dots, m_u,$$

i.e. the optimal control law can be obtained from the corresponding rows of  $H^1$ .

This completes the first step of dynamic programming. Now the operations  $\min_{u(N-1)} \mathbb{E}[\mathbf{z}' (H^1)' H^1 \mathbf{z} \mid N-2; u(N-1)]$  become the inmost in (4) and the above procedure can be repeated.

In this way the entire algorithm is decomposed into the following steps:

- (i) initialization – constructing  $H^0$  from the penalizing matrices according to the chosen structure of  $e_0$  and  $\mathbf{z}$ ;
- (ii) optimization – performing successively
  - (for  $i = 0, 1, \dots, N-1$ )
    - a) expectation – transformation of  $H^i$  into  $\tilde{H}^{i+1}$ ,
    - b) minimization – orthogonal transformation of  $\tilde{H}^{i+1}$  into  $H^{i+1}$ .

The detailed algorithmization depends on the arrangement of the vector  $\mathbf{z}$  in the

starting equation (6). In the sequel, the following well-trying structure will be used:

$$(11) \quad e^0 = H^0 \mathbf{z} = \begin{bmatrix} H_{y_0}^0 & H_c^0 & H_v^0 & H_{y_0}^0 & H_{u_0}^0 \end{bmatrix} \begin{bmatrix} \mathbf{z}_{y_0} \\ \mathbf{z}_c \\ \mathbf{z}_v \\ \mathbf{z}_{y_0} \\ \mathbf{z}_{u_0} \end{bmatrix}$$

where the partitioning of the vector  $\mathbf{z}$  is

$$(12) \quad \mathbf{z}' = [[y'(N), u'(N), \dots, y'(1), u'(1), y'(0), u'(0) \dots y'(-l + 1), \\ u'(-l + 1)], 1, [v'(N), \dots, v'(-l + 1)], [y_0'(N), \dots, y_0'(-l + 1)], \\ [u_0'(N), \dots, u_0'(-l + 1)]] = [\mathbf{z}'_{y_0}, \mathbf{z}'_c, \mathbf{z}'_v, \mathbf{z}'_{y_0}, \mathbf{z}'_{u_0}]$$

and

$$l = \max(l_y, l_u + t_u, l_v, l_{y_0}, l_{u_0})$$

The corresponding parts of the matrix  $H^0$  respect the definition (2.13) of the quadratic criterion in terms of  $e_y$

$$(13) \quad H_{y_0}^0 = \text{block-diag}(Q_y^{1/2}(N), Q_u^{1/2}(N), \dots, Q_y^{1/2}(1), Q_u^{1/2}(1), 0 \dots 0), \\ H_c^0 = 0, \quad H_v^0 = 0, \\ H_{y_0}^0 = \text{block-diag}(-Q_y^{1/2}(N), 0, -Q_y^{1/2}(N-1), 0, \dots, -Q_y^{1/2}(1), 0 \dots 0), \\ H_{u_0}^0 = \text{block-diag}(0, -Q_u^{1/2}(N), 0, -Q_u^{1/2}(N-1), \dots, 0, -Q_u^{1/2}(1), 0 \dots 0),$$

where the matrices  $Q_y^{1/2}$ ,  $Q_u^{1/2}$  are the (uniquely defined) generalized Cholesky square-roots [1.2]. This choice of square-root is advantageous because of the triangular form: moreover it arises naturally in the quantification of the control objectives (cf. Sec. 2.6). Time dependence of the penalties covers the presence of the stabilizing term (2.16).

**Remark.** (i) It should be stressed again that by solving the problem (4) in terms of successive modifications of  $H^i$  in the quadratic form  $\mathbf{z}'(H^i)' H^i \mathbf{z}$  we follow only the aim to make the square-root approach to control synthesis easy to understand and easy to use. As will be seen below, the final algorithm (corresponding to a given control problem and chosen strategy) has substantially smaller requirements on computer memory.

## 6.2. Some operations with square root factors

To simplify the presentation and to give a notion of the computational complexity of the particular design steps, a few operations on square-root factors will be described now. A conditional expectation of quadratic term of the type  $E[y'(\tau) Q y(\tau) | \tau - 1;$

$u(\tau)$ ] is computed repeatedly throughout the design. Assuming that the regression model of the random variable  $y$  (and similarly for  $y_0, u_0, v, y_x$ ) is known, we can evaluate the corresponding first and second conditional moments. For our case (cf. Chapter 3):

$$(14) \quad \begin{aligned} \mathbb{E}[y'(\tau) | \tau - 1; u(\tau)] &= \hat{P}' z(\tau) \\ \text{cov}[y(\tau) | \tau - 1; u(\tau)] &= \hat{R}(1 + z'(\tau) C z(\tau)) \end{aligned}$$

where  $\hat{P}, \hat{R}, C$  form the sufficient statistic,  $z(\tau)$  stands for the appropriate regressor in the model.

Substituting (14) into the formula (2.63) we derive

$$(15) \quad \begin{aligned} \mathbb{E}[y'(\tau) Q y(\tau) | \tau - 1; u(\tau)] &= \mathbb{E}[y'(\tau) Q y(\tau) | z(\tau)] = \\ &= (Q^{1/2} \hat{P}' z(\tau))' (Q^{1/2} \hat{P}' z(\tau)) + (1 + z'(\tau) C z(\tau)) \text{tr}(Q \hat{R}) \end{aligned}$$

Such a sum of quadratic terms can be expressed in the factorized form according to the "augmentation rule"

$$(16) \quad D = (A^{1/2})' A^{1/2} + (B^{1/2})' B^{1/2} = \begin{bmatrix} A^{1/2} \\ B^{1/2} \end{bmatrix}' \begin{bmatrix} A^{1/2} \\ B^{1/2} \end{bmatrix} = (D^{1/2})' D^{1/2}$$

To use this rule we can arrange the relation (15) in seemingly artificial way. Assuming that the vector  $z(\tau)$  contains unity as its (say  $i$ th) entry,  $z'(\tau) = [\dots, 1, \dots]$  and denoting  $\gamma = \sqrt{\text{tr}(Q \hat{R})}$ , we can write the term  $\text{tr}(Q \hat{R})$  in the form  $\frac{\gamma^2}{i}$

$$(17) \quad \text{tr}(Q \hat{R}) = z'(\tau) \begin{bmatrix} 0 \\ \gamma \\ 0 \end{bmatrix} [0, \gamma, 0] z(\tau).$$

Then the augmentation rule applied to (15) (with  $C$  also factorized), gives

$$(18) \quad \mathbb{E}[y'(\tau) Q y(\tau) | z(\tau)] = \left\| \begin{bmatrix} Q^{1/2} \hat{P}' \\ [0, \gamma, 0] \\ \gamma C^{1/2} \end{bmatrix} z(\tau) \right\|^2 = \|\tilde{H} z(\tau)\|^2$$

Of course, this easy construction of  $\tilde{H}$  is paid for by unnecessary high number of rows in  $\tilde{H}$ . However, we shall show that this rectangular matrix can be transformed back into a triangular form making the minimization step easy. As mentioned above,  $H$  and  $\tilde{H}$  must be related by  $H = T \tilde{H}$ , where  $T$  is an orthogonal matrix. There are many procedures performing such a transformation; a simple one based on elementary rotations is outlined here to show the way of thinking and the resulting computational complexity.

We say that the matrix  $T^{ij}$  is the matrix of elementary rotations if  $T^{ij}$  is an ortho-





The optimal control action has to fulfil (for any  $\mathbf{z}$ ) the equation

$$(24) \quad e_2^1 = 0 = \Phi^1 u(N) + \Gamma_{yu}^1 \mathbf{z}_{yu}^x + \Gamma_c^1 + \Gamma_v^1 \mathbf{z}_v^x$$

where  $\mathbf{z}_{yu}^x(\mathbf{z}_v^x)$  represents that part of  $\mathbf{z}_{yu}(\mathbf{z}_v)$  which is multiplied by nonzero coefficients. The vectors  $\mathbf{z}_{yu}^x$  and  $\mathbf{z}_v$  joint together form the "state" part  $x(t-1)$  of the regressor (5.1) Then

$$(25) \quad \Phi u(N) = -\Gamma^1 x(t-1)$$

Formally the control law can be introduced

$$(26) \quad L^1 = (\Phi^1)^{-1} [\Gamma_{yu}^1, \Gamma_c^1, \Gamma_v^1]$$

**Remarks.** (i) The  $\Phi$  and  $\Gamma$  are introduced in accordance with Chapter 5. The bar in  $\bar{\Psi}$  indicates that  $\bar{\Psi}$  slightly differs from  $\Psi$  in Chapter 5. It holds

$$\bar{\Psi}'\bar{\Psi} = \Psi'\Psi + \begin{bmatrix} Q_y & & & \\ & Q_u & & \\ & & \ddots & \\ & & & Q_y \\ & & & & Q_u \end{bmatrix}$$

$\underbrace{\hspace{10em}}_{\dim \Psi}$

In  $\bar{\Psi}$  several submatrices are indicated as they will be used explicitly below.

(ii) It should be noticed that the transformation  $T: \bar{H}^1 \rightarrow H^1$  depends only on the "yu" part of the regression model (it influences of course the "v" and "c" parts.) This is an algorithmic manifestation of the known fact that the design of the feedback controller (the "yu" part) can be made independently of the feedforward part and that the computation of feedback represents the more difficult part of the design.

(iii) The proper stabilizing term (cf. 2.6.2) guarantees that the diagonal elements of the triangular matrix  $\Phi$  are either nonzero or all elements of the corresponding rows are zero simultaneously.

(iv) Assuming optimal action at time  $N$ , neither  $e_1^1$  nor  $e_2^1$  influence subsequent optimization. However, the Remark 6.1 (i) applies.

The next optimization step for  $i = 1$  will be outlined now because the initial step has exceptional position (cf. also Chapter 4) and the form of a general step cannot be felt from it.

Expectation ( $i = 1$ ). Using formula (15) in taking expectation of  $y(N-1)$  and  $v(N-1)$  we have





#### 6.4. Comments on algorithmic extensions and realization of the control strategy

A relatively detailed solution of a special problem in the preceding section should give the reader a general feeling for the square-root based control synthesis. Possible generalizations of the above algorithm to different control problems and/or strategies will be sketched following the structure of Sec. 6.3.

##### 6.4.1. Multivariate systems

There is no essential difference between the univariate and multivariate cases. We shall demonstrate this fact on the evolution of the matrix  $\bar{W}$  defined by (30). Let us follow the main steps:

###### Initialization

The process of optimization starts with the initial value

$$\bar{W}^0 = \text{block-diag} [0, Q_y^{1/2}, Q_u^{1/2}, \dots, Q_y^{1/2}, Q_u^{1/2}, 0, 0]$$

###### Optimization

Consider now the general  $i$ th optimization step. It starts with the matrix  $\bar{W}^i$

$$\bar{W}^i = \begin{array}{c} \begin{array}{ccc} \overbrace{l_y m_y + l_u m_u} & \overbrace{1} & \overbrace{m_v (l_v - 1) m_v} \\ \Gamma_{yu}^i & \Gamma_c^i & \Gamma_v^i \\ \hline \bar{\Psi}_y^i & \bar{\Psi}_{1y}^i & \bar{\Psi}_{1c}^i \quad \bar{\Psi}_{4v}^i \quad \bar{\Psi}_{3v}^i \\ \hline & \bar{\Psi}_{1yu}^i & \bar{\Psi}_{2c}^i \quad \bar{\Psi}_{1v}^i \quad \bar{\Psi}_{2v}^i \end{array} \\ \left. \begin{array}{l} m_u \\ m_y \\ (l_y - 1) m_y + l_u m_u \end{array} \right\} \end{array}$$

The dimension  $l_v = \max(l_v, l_{av})$ . To simplify the notation, we assume the case  $l_v = l_u$ . Furthermore, it is supposed that  $l_u = l_y$ .

Expectation. The expectation at the  $i$ -th step causes

a) a change of the second block row which is advantageous to be stored in an auxiliary memory ( $R^i$  cf. (22), (28));

$$R^i = \begin{bmatrix} \overbrace{l_y m_y + (l_u + 1) m_u} & \overbrace{1} & \overbrace{l_v m_v} \\ \left[ \begin{array}{c} [\bar{\Psi}_{1y}^i, 0] + \\ + \bar{\Psi}_y^i [\hat{B}_0, \hat{A}_1, \hat{B}_1, \dots] \end{array} \right] & \left[ \begin{array}{c} \bar{\Psi}_{1c}^i + \\ + \bar{\Psi}_y^i \hat{c} \end{array} \right] & \left[ \begin{array}{c} [\bar{\Psi}_{3v}^i, 0] + \\ + \bar{\Psi}_{3v}^i [\hat{P}_v, 0] + \\ + \bar{\Psi}_y^i [\hat{D}_1, \hat{D}_2, \dots] \end{array} \right] \end{bmatrix} = \\ = [R_{yu}^i, R_c^i, R_v^i]$$

b) a modification of the submatrices  $\bar{\Psi}_{1v}^i, \bar{\Psi}_{2v}^i$ .

$$[\bar{\Psi}_{1v}^i \mid \bar{\Psi}_{2v}^i] \rightarrow [0 \mid \bar{\Psi}_{2v}^{i+1} \equiv \bar{\Psi}_{2v}^i + \bar{\Psi}_{1v}^i[\hat{P}_v, 0]]$$

The expectation step is finished by the rearrangement of the blocks in the matrix  $\bar{W}^i$  reflecting the fact that the nonzero parts of  $\bar{H}^{i+1}$  are stored in the space of the matrix  $\bar{W}^{i+1}$ . Then the matrix  $\bar{W}^{i+1}$  takes the form

$$(31) \quad \bar{W}^{i+1} = \begin{array}{c} \begin{array}{cc} \overbrace{(l_y - 1) m_y + l_u m_u} & \overbrace{m_y + m_u} & \overbrace{1} & \overbrace{(l_v - 1) m_v} & \overbrace{m_v} \\ \bar{\Psi}_{1yu}^i & 0 & \bar{\Psi}_{2c}^i & \bar{\Psi}_{2v}^{i+1} & 0 \\ \cdot Q_y^{1/2} & 0 & 0 & 0 & 0 \\ \cdot Q_u^{1/2} & & & & \end{array} \end{array}$$

**Minimization.** It consists of reduction of  $R^i$  and control law determination.

a) Reduction of  $R^i$ . By successive application of the elementary rotations performing the orthogonal transformation (9) the rows of  $R_{yu}^i$  are "reduced" (the particular entries are set to zero one by one) to achieve

$$(\bar{W}^{i+1})' \bar{W}^{i+1} + (R^i)' R^i \rightarrow (\bar{W}^{i+1})' \bar{W}^{i+1} + \begin{bmatrix} 0 \\ (R_c^i)' \\ (R_v^i)' \end{bmatrix} [0, R_c^i, R_v^i]$$

The matrix  $\bar{W}^{i+1}$  has then the same form as  $\bar{W}^i$ .

b) The optimal control law can be determined (if necessary) taking the first  $m_u$  rows of  $\bar{W}^{i+1}$  (cf. (24)–(26))

$$(32) \quad (L^{i+1})' = (\Phi^{i+1})^{-1} [\Gamma_{yu}^{i+1}, \Gamma_c^{i+1}, \Gamma_v^{i+1}] = (\Phi^{i+1})^{-1} \Gamma^{i+1}$$

It should be emphasized that the inversion of a triangular ( $m_u \times m_u$ ) matrix  $\Phi^{i+1}$  is relatively simple (cf. Remark 6.3 (i)).

**Remark.** The schemes of matrices are valid only if  $l_v \geq l_{av}$ . In the opposite (unusual) case a slight straightforward modification is necessary.

#### 6.4.2. Various control problems

So far the *regulation problem with*  $y_0 = 0, u_0 = 0$  has been treated. Let us discuss the more general case specified in Sec. 2.5.

A) *Program control.* In this case the reference values  $u_0(t) = u_0(t \mid 0), y_0(t) = y_0(t \mid 0)$  (see (2.5.2)) are known beforehand and the submatrices  $H_{y_0}, H_{u_0}$  (see (11)) should be considered in optimization. However, as the contribution to  $e^i$  correspond-





and finally similar rearrangement of blocks as in (31) is performed with  $\tilde{\Psi}_{2y_0}^{i+1}$ ,  $\tilde{\Psi}_{2u_0}^{i+1}$ , adding  $-Q_y^{1/2}$ ,  $-Q_u^{1/2}$  as seen below

$$(37) \quad \tilde{W}^{i+1} = \begin{array}{c|ccc|ccc} \cdot \tilde{\Psi}_{1y_0}^i & & 0 & \tilde{\Psi}_{2c}^i & \tilde{\Psi}_{2y_0}^{i+1} & \tilde{\Psi}_{2u_0}^{i+1} & \\ \hline & \cdot Q_y^{1/2} & & & 0 & -Q_y^{1/2} & 0 & 0 \\ \hline & & \cdot Q_u^{1/2} & 0 & 0 & 0 & 0 & -Q_u^{1/2} \end{array}$$

Minimization is performed as in previous cases.

It is worth mentioning that more general models for  $\hat{y}_0$  and  $\hat{u}_0$  can be considered within the same structure of the presented method.

#### 6.4.3. Algorithmical implementation of different control strategies

The necessary approximations of the problem statement and the resulting control strategies as well as various computational approaches were discussed in Chapter 4. The results are summarized in Table 4.7.

The algorithmization outlined in this chapter enables the user to build up a special-program for his given problem solution.

Different control strategies are now commented from the point of view of the presented algorithm.

- The certainty-equivalence strategy was used for the explanation; the current parameter estimates are used instead of unknown parameters without considering the uncertainty.

- The cautious strategy is obtained when the uncertainty of the parameter estimates is considered, i.e. the relation (17) includes the Cholesky factor  $C^{1/2}$  of the uncertainty matrix multiplied by the square root of  $\text{tr}(\hat{R}_y \bar{\Psi}_y' \bar{\Psi}_y)$ . Then an additional number of rows in the  $H$  matrix is to be reduced. This number grows with the number of optimization steps until the maximum  $m_u + l(m_u + m_y)$  is reached. Because of reasons discussed in Chapter 4 and also because of substantial increase of the computational complexity the uncertainties are considered only in the last optimization step.

The strategies can be implemented using:

- The receding horizon strategy, when the optimization process starts always with fixed initial  $H^0$  (13) given by penalization matrices, or

- IST (iterations spread in time) strategy, when the initialization is performed only at the beginning of the control process, otherwise is skipped.

When input restrictions are prescribed, the trial-and-error method and cutting-off are straightforward. The most sophisticated strategy is

– MIST (modified iteration spread in time). The optimization is complemented by a procedure solving a simple quadratic programming problem. The resulting Kuhn-Tucker multipliers are used to modify the input penalties for the next real time instant. Details can be found in [3], [4].

#### REFERENCES

---

- [1] K. J. Åström: Introduction to Stochastic Control Theory. Academic Press, New York 1970.
- [2] Ch. Lawson and R. Hanson: Solving Least Squares Problem. Prentice Hall, Englewood Cliffs 1974.
- [3] J. Böhm and M. Kárný: Self-tuning regulators with restricted inputs. *Kybernetika 18* (1982), 6, 529–543.
- [4] J. Böhm: LQ self-tuners with signal level constraints. In: 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, July 1985 (preprints; H. A. Barker, P. C. Young, eds.) Pergamon Press, Oxford 1985. Vol. 1, pp. 131–137.

#### 7. APPLICATIONS

A short review of applications of the described adaptive controllers will be given now in order to encourage potential users. The selection of cases has been influenced by availability of documentation. Details can be supplied by the authors of the references given below.

**Drum boiler:** industrial case study, Nov. 1976 – Feb. 1977.

*Objective:* Regulation of the temperatures and pressure of overheated steam in a branch of a superheater by changing the water injection and/or the position of powdered coal feeders.

*Experiments:* A family of multivariate (regression) models with the structure indices up to  $(l_y, l_u, t_u, m_y, m_u) = (4, 4, 1, 3, 2)$  was tested. The multistep strategy compared favourably to the one-stage-ahead control. The receding horizon strategy dominated the IST strategy due to strong nonstationarity of the process. The cautious strategy appeared preferable probably because no input penalty was used.

*Results:* Despite the lack of experience of the control group with the process as well as with the adaptive controller and in spite of not excellent state of technology hardware the overall stabilization of the process was achieved. The effects of adaptive control were much appreciated by the plant people.

#### *References:*

- J. Fessl: On the experimental verification of a control algorithm with real-time identification. IFAC Workshop on Computer Application in Discrete Manufacturing, Prague 1977, to appear in *Automatica*.
- J. Fessl: Verification of multivariate selftuning controller in heating-plant operation (in Czech). *Automatizace 21* (1978), No 10, 257–263, No 11, 293–297.

**Paper industry:** Full scale repeated use on different paper machines (since 1976, 1980, 1981, 1983, other, as yet unreported, microprocessor based applications exist).

*Objective:* Regulation of the basis weight, the moisture and the ash content by changing the speed of a paper machine, the pressure in dryer zones and the kaolin suspension flow.

*Implementation:* Low order single delayed input and single output models with a measurable disturbance are used. The certainty equivalence strategy and explicitly determined control laws are applied. The formerly used one-stage-ahead strategy has been replaced by the infinite-horizon one making the controllers more reliable.

*Results:* Variances of regulated variables decrease 3–5 times giving a substantial economical benefit.

*References:*

- A. Lízr, A. Lízrová, J. Hejda and F. Dušek: Application of adaptive control algorithms in Czechoslovak paper industry. 5th IFAC Conference PRP-Automation, Antwerp (Belgium), 11–13 October 1983.
- A. Lízr, A. Lízrová and J. Hejda: Microcomputer control system for smaller paper machines. 20 EUCEPA Conference, Budapest 1982.

**Cold-rolling mill:** A chain of industrial case studies since 1978, now in operation since 1984.

*Objective:* Regulation of the longitudinal profile of hard metal strip changing the rolling forces.

*Implementation:* A single delayed input and single delayed output model describing relation among the increments of the input, the output and an external measurable disturbance is used. The model had to be simplified to this level because the needed sampling rate was 0.03 sec. The certainty equivalence strategy with the explicitly evaluated control law is used. Directional forgetting (Chap. 3) has been successfully applied here for the first time in the closed adaptive loop.

*Results:* The control error falls down to the range of the measurement noise ( $\approx 1 \mu\text{m}$ ). Such a quality highly exceeds current requirements and has not been met before. Significant improvements are observed even in transient stages. The major control effect is due to feedforward as the transport delay about of 20 samples prevents feedback from being efficient.

*References:*

- J. Fessl and F. Šíp: Design and verification of a self-tuning regulator for cold-rolled strip thickness (in Czech). *Automatizace* 24 (1981), No 7, 180–184, No 8, 225–230.
- P. Etler, F. Jirkovský and J. Štika: Internal report of ŠKODA ZTS, Vh 29627-S, 1984.

**Sinter rotary furnace:** Industrial case study 1980, in operation since 1985.

*Objective:* Regulation of the temperature in the sintering zone of a rotary furnace (used as the first technological phase in  $Al_2O_3$  production) by changing the amount of fuel.

*Experiments:* The relation between single input single controlled output and single auxiliary output proved to be sufficiently described by the regression model of the second order as preliminary identification and short-time experiments had shown. The receding horizon and IST strategies in certainty equivalent versions exhibited similar behaviour (the one-stage-ahead control was shown to be insufficient again).

*Results:* The required range of  $\pm 20$  °C has been achieved even under heavy disturbances with the overall stabilizing effect on the thermal profile within the furnace.

*References:*

J. Minka: Computer control of sinter rotary furnace (in Slovak). Ph. D. Thesis, Faculty of Electrical Engineering Bratislava, 1980.

At present the following control tasks are known to be at different stages of preparation and testing: regulation of a heat exchanger of 200 MW power plant, stabilization of distillation columns, cooling of a hot-rolled strip, regulation of temperature in plastics production and others.

At the end of this chapter we want to stress that the applications we met

- supported our confidence that the described controllers are ready for industrial use and are able to bring substantial economical effects,
- showed that the overall management, especially instrumentation and data management are decisive with respect to success of any automation,
- exemplified the usefulness of the feedback from practice to theory. Most of the reported theoretical results have been substantially stimulated by such a feedback, which has been possible mainly due to the effort and enthusiasm of control engineers from industrial institutes and plants. We are using the opportunity to express our gratitude to them.

## 8. SOFTWARE SUPPORT

The algorithms described above as well as the others arisen from research and extensive experimentation in the area of adaptive control at the Institute of Information Theory and Automation have been coded and collected in the library SIC [10] (library of subroutines for simulation, identification and adaptive control of stochastic systems). The subroutines contained now in the SIC library enable the user:

- to simulate a controlled system by the regression model or the stochastic

difference/differential equation; or, alternatively, to use data previously measured on a real system;

- to identify a controlled system with the regression model (see Chap. 3), the regression model with a common factor explicitly expressed (see [1]), the ARMAX model (using the recursive extended least squares), the stochastic differential equation (see [2]); to determine the regression model order (see Chap. 3);

- to control a given system by means of a constant (pre-specified) controller, a continuous PID controller, a simple (based on explicit formulas) self-tuning controller (see [2.13]) with the possibility to compensate periodic disturbances (see [1]) and/or non-zero offset (see [3]), a general adaptive controller (see Chap. 5) square-root based adaptive controllers (see Chap. 6), and a univariate adaptive controller with input constraints (see [4.7]);

- to generate variables of different deterministic or statistical properties; to filter variables; to carry out statistical analysis of results.

The design of the above algorithms has been supported by a number of universal procedures solving often encountered subproblems (data handling using shift or cyclic registers, Cholesky or LD-factorizations of given matrices, updating of factorized matrices by new data dyads, matrix and polynomial operations, numerical integration etc.).

At present new algorithms are being prepared or re-coded which enable the user to determine the structure of the regression model including the choice of relevant quantities (see [3.5]), to respect prior constraints on possible values of regression coefficients (see [3.4]) and to apply the hybrid self-tuning controller (see [2]).

### 8.1. Use of algorithms

Subroutines of the SIC-library referred to for simplicity as “algorithms”, can be employed directly in the control loop of a variety of industrial applications. However, a successful application of any algorithm should be preceded by an experimentation. Measurements on the real system should be collected and processed by appropriate identification algorithms solving the questions of structure determination and parameters estimation. The next steps consist of the construction of a simulation system based on the results of identification and of experiments with the algorithm using the simulated reality.

A similar, often more extensive, experimentation is to be carried out when different algorithms are to be compared and a numerical experience with them obtained. Such activity requires a user-friendly software environment to be tailored. The user cannot, for each particular experimental step, write and debug a main “supervising” program and study all details of the intended use of algorithms.

No matter how the supervising program is designed, it should make the problem formulation and solution easy, in a dialogue with the computer. General requirements for its design are illustrated in Figure 1:

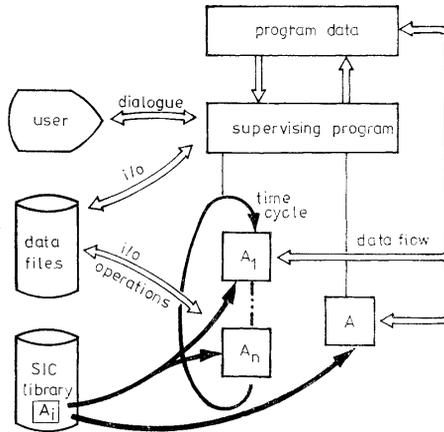


Fig. 1. Supervising program: requirements.

A facility should be present for handling data files that contain e.g. measurements on real systems. The files and program data are to be processed by algorithms applied separately or in a combination – processing of time-series of measurements is required. Results of one algorithm are to be used by another and different methods compared – it implies a large scale support of I/O operations, including a graphical presentation of results. Experience shows that the distribution of separate SIC-algorithms, without such a supporting environment, is of a restricted practical value. The SIC-library is distributed together with a supervising program as an “interactive software package”. Main ideas of its design will be shortly discussed in this chapter, in order to facilitate understanding of software results presented.

### 8.1.1. Interactive software package

Design of interactive software package seems to be a way how to reach a higher level of computer utilization in the field of automatic control and the only effective possibility how to make new developments available for a large group of users and how to encourage use of well-proven, stable and robust numerical software.

Programming effort to tailor an interactive software package is usually considerable and despite of great practical interest very little has been published about a standardization and general tools related to its design. As far as the existing interactive software packages are concerned, a deficiency can be felt that they are usually

closely linked to a specific computer, operating system or a high programming language. Refer to [4] for a general discussion and comparison of packages.

These are reasons standing behind the decision to create, as an additional output of the research in the field of adaptive control, a software base enabling a comfortable design of transferable interactive packages in laboratory conditions.

The development resulted in formation of a "construction library" covering the problems of design of interactive software packages and in creation of a general supervising program that facilitates the handling of any "load module library" of algorithms. Those are the topics discussed in the following sections.

### **8.1.2. Programming tools**

Algorithms are written by researchers using simple programming means. In the field of automatic control, it is generally agreed (refer to [4]) that algorithms should be coded in Standard Fortran IV as given in USA ANS X3.9 1966. Fortran is chosen because its compilers are available in most computer systems and many of already existing subroutines are written in it. Besides, Fortran can be viewed as well-balanced for such a type of programming.

The design of algorithm library is quite arbitrary, the only limitation being that the algorithms should not use common blocks. Our experience showed that the selection of a proper hosting language for the design of the supervising program loses most of its importance if modularity approach is consistently adopted. Fortran was selected to be in harmony with algorithm library and for the sake of system portability.

Fortran does not support mechanisms inevitable for design of interactive software packages. But this can be ignored if the lowest level of the modular software base is written in Assembler. Those machine-dependent parts are concentrated into a relatively small number of well-defined blocks. The rest of software is then machine independent and consists of logical and arithmetical statements and calling of lower level blocks only.

The overall design was aimed at a transferability feature. In this respect mini-computers were considered in the first place, being a standard equipment of research laboratories. Transferability is ensured not only by the use of Fortran, but also by means of a sophisticated overlay structure and by a form of preprocessing of source programs: machine dependent coding is consistently introduced to source texts by an editing program.

## **8.2. Construction library**

The attempt to standardize the design of interactive software package resulted in a form of "construction library". It provides not only powerful software means,

carefully selected and proven, but represents also a methodological support for a specific feature design. Main ideas employed will be shortly described.

**Dynamic storage allocation.** The mechanism is fundamental for a design of interactive packages. Any data are stored in a specific data-area. The access to it is not arbitrary; instead, a free space can be asked for and granted only with the help of "allocation-functions". Allocations are grouped into "blocks" and deleted if not used any longer. The allocation mechanism itself is quite simple, but machine-dependent because of the way the Fortran compiler allocates data.

**Standardization of data structures.** The dynamic storage allocation allows a standardization of data structures used in the design. They are distinguished by their allocation procedure and their data interpretation, e.g. real matrix, a character string etc. Such a data structure is described by its coded "type", by a pointer to the first data item and by a "dope" vector that reflects its internal structure, e.g. dimensions.

**Conversational data structures.** The standardization of data structures gives a possibility to create "external" structures used in the conversational environment. They are shared under their names between subprograms. A table of descriptions is maintained summarizing all coded information related to particular names.

**Input/output support.** The conversational environment needs a facility of line editing and a facility of handling "system files". These files are processed as a stream of binary data items having a sort of data directory at their beginning. Their primary role consists in recording the contents of used data-area and storing results of repetitive calculations.

Such facilities, as well as the usual Fortran-like means related to input and output, are designed on the basis of Fortran unformatted and formatted I/O statements.

**Terminal support.** An alphanumerical display unit is considered to be a basic supporting input/output device: at the input it can optionally substitute the input operation, at the output it can display output lines formed. Subroutines to read a line from terminal keyboard, to display one character line and to clear the screen form the basic machine-dependent level of this support.

**Retrieval of character strings.** The processing of character strings is supported down to a machine dependent level of conversion of a character into integer and vice versa. The main application of the support lies in the processing of control commands and forming an output document ("control commands" refer to statements used to control the run of the package). Specific subroutines take care of command elements detection, syntax analysis and resulting internal program branching.

**Dialogue means.** A group of means supporting the design of dialogues makes possible to centralize all dialogue services and to formalize the dialogue rules.

A dialogue consists of "dialogue units" covering self-contained topics of system-user

interaction. The user's reply is expected, but each unit gives also a possibility to ask for explication (by a question mark), to ask the menu available, to quit the current unit, to select a "default" solution, and to switch to another mode of processing (e.g. from interactive to batch mode).

The most important dialogue unit appears during "external allocation" of data structures. It enables the user to select the name and/or initial data of the structure being allocated. Another frequent dialogue unit is created whenever a syntax error is detected thus making possible to change the command in error.

**Dynamic loading program modules.** The design of SIC-library has been carried out by a group of researchers for several years. It was impossible to appoint an "administrator" responsible for each particular change of the package, instead, a way was developed how to reach "autonomy" in the design for each person of the group. The following idea was expressed in software means:

Subroutines as executable program modules are grouped in a "load module" library. Whenever a subroutine is to be called, the module is loaded to the memory, linked to the calling program and passed control to. The supporting subroutine for such "dynamic library loading" must be written in Assembler using the specific means of a given operating system (OS/VS and DOS versions are available).

It should be noted that load modules are processed by the Linkage Editor program independently to each other and to the supervising program.

**Other means.** Refer to [7] for another problem ranges covered by the construction library. A special note should be devoted to the simple but effective errors and warnings system and to the support for writing interface subroutines mentioned in the next section.

**Documentation.** In the laboratory conditions, it seems almost impossible to maintain other information than comment cards of source programs and short explanatory pieces of information that can be edited with the help of the computer. It was proved useful to place the source programs as well as the explanatory texts into a "source library" and to use an editing program for preparation of manuals [7-10]. In such a way every change of algorithms is automatically reflected in the descriptions.

Distribution of case studies related to specific algorithms as well as to the logic of the supervising program has proved to be a very instructive form of documentation, too.

### 8.3. Load module library

The dynamic loading mechanism gives the possibility of designing the supervising program in quite general terms provided that the contents of the algorithm library is not known beforehand. Then the supervising can supply services to different

libraries without any change of its coding. Similarly, the load module library is quite freely connected with the supervising program and can be given services from different supervising programs. This feature is very important if the overall design is carried out in a group and the software package is to be used in different conditions.

Several types of control commands can be used to obtain access to load modules. The way how the load modules are called makes it possible to distinguish different types of load modules:

**Subroutines.** Each Fortran subroutine located in the on-line library can be called from the supervising program by a control command that is equivalent to the Fortran "CALL" statement. If the subroutine is an algorithm with several arguments, this use does not offer much more than Fortran programming.

**Interfaces.** To reach a higher degree of flexibility in the use of an algorithm, a special "interface" subroutine can be connected with it. Such an interface subroutine to supervising program forms a unit with the algorithm in question and is located under a name in the load module library. Three types of resulting load modules are distinguished:

**Functions.** They are referred to in arithmetic expressions and/or assignment statements. A function has varying number of arguments "returning" a result which may be any data structure, e.g. a matrix. A function makes it possible to check and convert arguments, to modify processing in accordance with the number and type of arguments, and to establish values for arguments not mentioned.

**Self-contained subroutines.** Each control command begins with a keyword. If a name is detected where a keyword is expected, it is interpreted as the name of a load module and the load module is passed control to. No attempt is made to process arguments of such modules on the level of supervising program. The command that follows the name is processed directly by the module loaded. In such a way a space is left for the design of user-defined commands.

**Initialization subroutines.** There is a control command that enables to pass control to "initialization subroutines" used to prepare an "environment" for a repetitive use of algorithms, e.g. for data processing. The initialization subroutine defines the arguments used for calling an algorithm, introduces the name of the algorithm and list of arguments and makes any initial calculation required for the intended use.

The environment is optionally prepared in a dialogue with the user. The information needed for a use of the algorithm can be asked for during the dialogue session. In such a way a "self-explanatory" feature of the package was reached.

**Remarks.** (i) The interface from an algorithm to the supervising program is written by the author of the algorithm. The coding is supported by a set of means from the construction library related to the allocation of arguments, their initialization and the design of a dialogue with users. Despite of forcing the researchers to write

little more than the body of the algorithm, the interface-coding by researchers has its merits. Designer's numerical experience is expressed in coding and need not be looked for in library descriptions. Use of the algorithm is consistently checked and controlled by its author. It should be stressed that coding an interface is in its simplicity equivalent to writing the usual library descriptions.

(ii) A syntactical and semantical analysis of user's commands is left to the designer, but programming is widely supported by the construction library. Related means are carefully selected so that e.g. design of a pre-defined command language is not too difficult. No fixed rules are established – it depends only upon designer's taste how the dialogue related to his problem will look like.

#### **8.4. Supervising program**

The main supervising program forms the interactive organizational framework of the software package. It supplies a broad variety of services related to overall processing, but itself takes no care what algorithms do with program data and files. The mentioned "load module" approach enabled us to design the supervising program of the SIC-library in a form of an "empty interactive package" applicable to any scientific algorithm library in the field of automatic control and statistical data analysis.

Two modes are distinguished within the run of the supervising program: static and dynamic one.

##### **8.4.1. Static mode of processing**

In the static mode, Fortran-like control commands are immediately interpreted. Data-structures are defined, filled with data and displayed. Arithmetic operations and assignment statements are carried out, analytic operators may be applied even to vectors and matrices. Coding of matrix-algebra expressions is facilitated by a selection of built-in functions and by an explicit operator of transposition.

Control commands that handle the load module library are applied to subroutines, initialization subroutines and self-contained modules. Commands for saving and retrieval from the system memory between tasks and/or user's sessions make it possible to establish checkpoints and avoid cumbersome definitions. A set of means handle data and program files. Conditions are specified for a consecutive dynamic mode of the program run: e.g. names of files and control lists of different I/O operations.

##### **8.4.2. Dynamic mode of processing**

In the dynamic mode the pre-specified I/O operations are carried out and algorithms are applied in a cycle (of "time", "case number" etc.) – e.g. various data

files are merged and processed by a series of algorithms and the results displayed in a graphic form. The list of operations that can be carried out in a dynamic mode follows – see Figure 2.

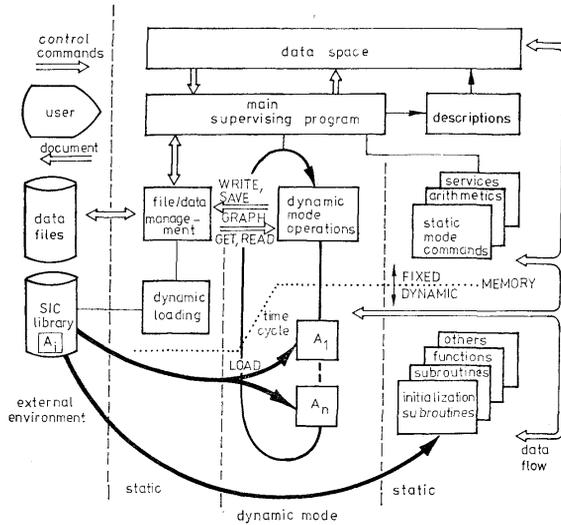


Fig. 2. Supervising program: functions.

- LOAD: algorithms needed are loaded and linked to the supervising program and then repetitively called
- GET/SAVE: system files are read and/or written
- READ/WRITE: any data files are read and/or written in a binary or character form. Character data can be displayed on a terminal
- GRAPH: Data structures are displayed in a graphical form

Any number of such operations can be activated, each being carried out with its frequency in the “time” loop. Operations are repeated without a direct user’s control, but this “automatic run” can be periodically interrupted to allow data changes and results to be displayed.

### 8.4.3. Levels of use

The overall processing is modified using three control indicators:

- message level: controls the amount of information displayed
- dialogue level: expresses user's participation in a program run
- debugging level: controls the display of intermediate results and tracing information.

These indicators possess a nonnegative value with the general meaning: 0-no, 1-basic, otherwise extended in ascending order. The value can be reset at any point of a dialogue or by a special command during a batch processing. The indicators make it possible to define different levels of the use of the package, e.g. novice, advanced, expert. A program run varies between batch and interactive processing of different types.

**Remark.** The features described are illustrated in a simplified graphic form in Figure 2. The data space is used to store any data and the data flow exists between all system components. The current contents of the data space is described in the block of "descriptions". The dynamic loading mechanism is represented by a block cooperating with the disc unit where the load module library is located. Its modules are loaded into the "dynamic" part of the occupied memory. An attempt has been made to illustrate the structure of dynamic as well as static mode of the run. It should be noted that the supervising program runs under control of an operating system (OS). The file management support and access methods, used to process data files, are also under a partial control of the OS.

### 8.5. Summary

The distinguished features of the software base developed can be summarized as follows

- the SIC library covers the actual state of art in the presented design of adaptive controllers, being distributed in the form of the interactive software package,
- the interactive-organizational part of the package is designed in general terms, independently of the specific content of the SIC library, giving the possibility to use supervising part with any algorithm library and prepare intelligent user-friendly flexible packages e.g. for statistical data analysis,
- the software base represents a standardization of design of interactive software packages, reaching it by means of a well-documented and modular subroutine library,
- the overall design is a specific step towards an effective connection of theory and practice, by achieving software portability with the help of FORTRAN, pre-processing of source programs and overlay-fitting structure.

## REFERENCES

---

- [1] T. Jeníček: Adaptive Control and Periodic Disturbance (in Czech). Research Report No. 1321, Institute of Information Theory and Automation of Czechoslovak Academy of Sciences (ÚTIA ČSAV), Prague 1985.
- [2] I. Nagy and V. Peterka: A hybrid LQ self-tuning controller. Preprints 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, University of York 1985.
- [3] I. Rosol: Adaptive Control with Constraints (in Czech). Research Report No. 1314, ÚTIA ČSAV, Prague 1985.
- [4] W. J. M. Lemens and A. J. W. Van den Boom: Interactive computer programs for education and research: a survey. *Automatica* 15 (1979), 113–221.
- [5] M. Kárný, J. Böhm, A. Halousková and P. Nedoma: SIC — Program for System Simulation, Identification and Control (in Czech). Research Report No. 1020, ÚTIA ČSAV, Prague 1977.
- [6] P. Nedoma: Design of experimental statistical packages. Preprints 6th COMPSTAT Symposium, Prague 1984.
- [7] P. Nedoma: KOS — Conversational Service System: System Manual. Research Report No. 1246, release 4: Prague 1985.
- [8] P. Nedoma: KOS — User's Manual. Research Report No. 1237, release 7: Prague 1985.
- [9] P. Nedoma: KOS — Programmer's Guide. Research Report No. 1238, release 2: Prague 1985.
- [10] R. Kulhavý and A. Halousková (eds.): SIC — Library for Simulation, Identification and Adaptive Control of Stochastic Systems: User's Guide. ÚTIA ČSAV, Prague 1985.

