MIMO - A SET OF SISO ?
Multivariate system adaptively controlled as a set of single-input single-output models

Miroslav Kárný, Alena Halousková, Josef Böhm

Institute of Information Theory and Automation
Czechoslovak Academy of Sciences
Pod vodárenskou věží 4, 18208, Prague 8, Czechoslovakia
Phone: (422)8152274, E-mail: adapt@cspgas11.bitnet

Abstract. Computational complexity hidden behind the apparent elegance of multi-input multi-output (MIMO) ARX model description used for the adaptive control purposes is addressed. A non-standard modelling parameterization exploited for enhancing structure estimation outcomes leads to innovated identification and control design. They respect richness of MIMO case, for instance multi-rate sampling, and offer a new design algorithm of classical adaptive controllers.

Key words. MIMO systems, model parameterization, structure and parameter estimation, control design, multi-rate sampling, LQG self-tuners.

Introduction

This paper is a research report, openly mirroring the zig-zag way between barriers of computational complexity hidden behind the apparent elegance of multi-input multi-output (MIMO) autoregressive-regression (ARX) used for the adaptive control.

Linear quadratic Gaussian (LQG) adaptive control, based on optimization of a multi-step criterion [2] has nice theoretical and computational properties and its power has been proved in practice. Its computer-aided design (CAD) package called DESIGNER has been developed [4]. DESIGNER prepares automatically the optimal function including a safe start for the adaptive control of the chosen type.

The package in its present state was finished in 1990. During the period of experimental tests several improvements have been suggested (increased convergence rate of iterative computations [5], absolute terms in regression models respected, user-friendly dialogs proposed etc.). It was verified that every piece of prior information used in the design can contribute to the final controller specification and corresponding performance improvement [7].

The package covers MIMO case; however, the overall CAD procedure becomes less transparent in this case and computational aspects come into the foreground. This paper addresses the achievements attained when trying to respect MIMO nature of the controlled system.

It has been suggested that substantial alleviation of the "dimensionality curse" may be reached by using a modified system parameterization [6]. When adopting this parameterization we have found unforeseen consequences both for modelling and control design.

The paper summarizes shortly the present features of the CAD package DESIGNER; disadvantages and unused potential advantages are stressed. It is shown, how the model modification can help the situation. The impact of the model changes on the identification and synthesis is then dealt with. In the conclusions stimulating interplay between theory and computation is stressed, which has led to the present results.

Design of LQG adaptive controller

The controller uses the m_y-dimensional output $y(k)$ of the controlled system, its m_u-dimensional input $u(k)$ and m_v-dimensional external disturbance $v(k)$ (if available). The data are indexed by the discrete time moments $k = 1, 2, ..$ at which inputs can be changed.

Control quality optimized by the controller is quantified by the expected ($\mathcal{E}[\cdot]$) value of the *quadratic loss*

$$\mathcal{K}[Q] = \frac{1}{T}\{Q_\psi[\psi(T) - \psi_0(T-1)] + \sum_{k=1}^{T} Q_y[y(k) - y_0(k-1)] + Q_u[u(k) - u_0(k-1)]\} \quad (1)$$

on a long horizon T. For weighted squared norm of the deviation of a vector x from its reference value x_0, a shorthand (slightly inconsistent) notation $Q_x[x - x_0] = (x - x_0)'Q_x'Q_x(x - x_0)$ is used ($'$ means transposition). The distinction between the norm $Q_x[\cdot]$ and the corresponding matrix weight will be clear from context. Q is a common name for the weights. The regression vector $\psi(k)$ is defined below.

Model describing the controlled system behavior is a

MIMO *linear* (*Gaussian*) model

$$y(k) = \Theta' \psi(k) + e(k) = \sum_{i=0}^{l_u} \Theta_{ui} u(k-i-n_u) \quad (2)$$
$$+ \sum_{i=1}^{l_y} \Theta_{yi} y(k-i) + \sum_{i=1}^{l_v} \Theta_{vi} v(k-i-n_v) + e(k).$$

For $x = y, u, v$, Θ_{xi} denotes the matrix regression coefficients at the x variables delayed by i steps. The number of these coefficients is determined by the "orders" l_x and the delays $n_u,\ n_v$. The sequence $\{e(k)\}_{t \geq 1}$ consists of mutually independent m_y-dimensional random variables normally distributed with zero mean and constant covariance matrix R.

Model of the disturbance v is an autoregressive written similarly as the previous ARX model.

Admissible control strategies are restricted by informational and input limitations. The input $u(k)$ is chosen from $\mathcal{U}(k) \equiv [ul(k), uu(k)]$ under the non-anticipativity condition with the use of the data $d(1), d(2), .., d(k-1)$, where the data item $d(k)$ means $d(k) = (y(k), v(k), u(k))$.

The adaptive controller estimates recursively the unknown coefficients of the system model (2) and of the disturbance model. It exploits the estimates for an approximate minimization of the expected value of the quadratic loss (1) under the restriction $u(k) \in \mathcal{U}(k)$.

In the *practical implementation* of the adaptive controller, the user has to choose in advance the quantities specifying the criterion (1) and process model (2). The main task of the DESIGNER package is to support this choice.

DESIGNER – its disadvantages and unexploited advantages

The preparation phase consists of three tasks:

Structure estimation means the choice of signals and dimensions of fields necessary for controller function. The controlled outputs are usually defined by technical requirements. Much more freedom is often met in choosing controlling inputs, useful external disturbances and in determining the regression (2).

The used probabilistic approach to structure estimation [8] orders the regression elements according to their significance. This powerful outcome is exploited in the subsequent steps only partially. This shortcoming is especially conspicuous in MIMO cases.

Let us illustrate the situation assuming two-input single-output case. Starting from the upper limits on field dimensions, the recommended structure is found and coded as indicators of regression-entries significance. The resulting regressor may look like

$$\psi'(k) = [u_1(k), u_1(k-1), u_2(k-3), y(k-1)].$$

The model (2) used for controller design, however, assumes both *common delay* (n_u) and *common order* (l_u) for all Θ entries. The *information about fine structure is not exploited* (we have to set $n_u = 0$, $l_u = 3$).

The scheme brings very rich information about the inner structure of significant elements inside the Θ matrix (its sparseness). *However, fine as the description is, it is not fine enough.* This statement is illustrated using the schematic description with x marking significant elements (2 by 2 system is assumed)

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} x & 0 & x & 0 \\ 0 & x & 0 & x \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ y_1(k-1) \\ y_2(k-1) \end{bmatrix}.$$

The form of parameter matrix shows, that the system is composed of two independent ones (under the condition of diagonal covariance R). This fact is overlooked if significance of regressor entries is concerned, i.e. if *a global significance of Θ-rows is decided only, not the significance of particular entries.*

Parameter estimation is the main source of the controller adaptivity. Its on-line properties are decisive in long run. However, the desirable bump-less start requires well prepared estimator initialization.

For the model (2), the estimation reduces to updating of statistics which is formally equivalent to recursive least squares (RLS) in *matrix extension of single output case.* The updating of the estimates $\hat{\Theta}$ of the parameter Θ has the structure

$$\hat{\Theta}_{new} = \hat{\Theta}_{old} + \begin{bmatrix} \text{column} \\ \text{gain} \end{bmatrix} \cdot [\text{row prediction error}]$$

which in a schematic example may have the form

$$\begin{bmatrix} x & x \\ x & x \end{bmatrix}_{new} = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix}_{old} + \begin{bmatrix} x \\ x \end{bmatrix} \begin{bmatrix} x & x \end{bmatrix}.$$

It is clear, that the *parameter estimate – updated by an added dyad – cannot keep a finer inner structure.* This introduces unnecessary redundant parameters.

Moreover, *sampling rates differing in various input-output channels are not supported* by the model (2).

Choice of penalization matrices Q specifying the loss function (1) determines the attainable input range and overall quality. Having some preliminary parameter estimate and a penalization ensuring the desirable input range, it is possible to compute the probable range of outputs and so give the user a rough estimate of the attainable control quality. He can judge in advance the economy of the project.

The current solution can be viewed as Monte-Carlo projection of the model uncertainty into closed loop uncertainty. Obviously, any *redundant parameter estimated in previous step increases prediction uncertainty of closed-loop quantities.*

Moreover, both on-line and off-line *design is formulated for common sampling and control rates.*

Adopted MISO models

Surprisingly enough, all problems listed above can be solved much easier if an alternative model parameterization is used. Essentially, separate regressions for the particular outputs (with different regressors including significant items only) are used. Their interrelations are respected by including the other present outputs into the separate models – interrelated set of MISO models is created as an equivalent for MIMO description. Formally, for $i = 1, \dots, m_y$ (number of outputs), we predict ith output entry by

$$y_i(k) = \bar{\Theta}_i' \bar{\psi}_i(k) + \bar{e}_i(k) \quad (3)$$

where $\bar{\Theta}_i$ is coefficient *vector*, ith re-defined regressor $\bar{\psi}_i$(k) includes $y_j(k)$, $j \neq i$, and $cov(\bar{e})$ is diagonal.

Relation of the new model to the form (2) depends on the noise properties. Writing its covariance in the factorized form MDM' (with diagonal D and M possibly triangular), the model (2) can be mapped to

$$M^{-1}y(k) = M^{-1}\Theta'\psi(k) + M^{-1}e(k). \qquad (4)$$

This equation can be rearranged to the modified model (3) with the noise $\bar{e} = M^{-1}e$ and properly extended regression vectors $\bar{\psi}$. The transformation is trivial for diagonal matrix M, but this case is rare (though often misused as the simplest one).

Remark

1. The general way of overcoming the redundancy problem (vectorial organization of Θ) is not used as it is mostly non-acceptable due to dimensionality.

MISO structure estimation

Fine structure description of the optimal regression resulting from the structure estimation algorithm can *reveal specific system properties: multivariate delays and "orders", decomposition into subsystems.*

Especially, it can be estimated whether a signal is an external disturbance: *unrealistic requirement to recognize $v(\cdot)$ a priori is relaxed.* Consequently, it is better to take the variable $v(\cdot)$ as a part of the output.

These new features are paid by a *new task for structure estimation* – the search for the optimal factor M in (4), as the introduced model parameterization (3) depends on its choice. It can be seen on the example

$$\bar{\Theta}_1 = M_1^{-1}\Theta = \begin{bmatrix} x & 0 \\ x & x \end{bmatrix} \begin{bmatrix} x & 0 & 0 & x \\ 0 & x & x & x \end{bmatrix} = \begin{bmatrix} x & 0 & 0 & x \\ x & x & x & x \end{bmatrix}$$

$$\bar{\Theta}_2 = M_2^{-1}\Theta = \begin{bmatrix} x & x \\ 0 & x \end{bmatrix} \begin{bmatrix} x & 0 & 0 & x \\ 0 & x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \end{bmatrix}.$$

Hereafter, we omit bar sign above Θ, ψ etc., as we deal with the model (3) only.

MISO parameter estimation

Full use of structure estimation results motivated us to use the non-standard model parameterization (3). We can *deal with different regression length (and/or transport delay) in the particular output models: this property is preserved in estimation* as the prediction error is scalar.

The separated modelling of particular output regressions opens new freedom for sequence of computations. It is quite natural to update regressor whenever new value appears; the parameter estimate for a given output model can be updated as soon as an appropriate regressor-regressand pair becomes available. This property opens the way for treatment of systems with *multiple sampling rates.* Unified description of such systems can be reached by continuous-time models or by substantial restriction of sampling freedom. Thus, the former case is preferable and we confine ourselves to it. We exploit spline-based models [3] as they represent a direct extension of the ARX models.

Spline-based modelling starts from a convolution model of the system and approximates both convolution kernels and signals by splines. In this way, a model similar to (3) is gained.

$$y_i(t) = \Theta_i' g_i(t)\psi_i(t) + e_i(t). \qquad (5)$$

The model (5) describes input-output relation for any continuous time moment t. Time invariance of the unknown vector Θ_i' is preserved by incorporating a "filtering" matrix $g_i(t)$ which is a simple deterministic function of t. For details see [3].

In order to stay within discrete world of computers we shall deal with *pseudo-continuous time t* which changes by integer multiples of *elementary time quantum, say h, smaller than any inter-sample distance of a single signal.*

Theoretical considerations

Surprising (at least for the authors) richness of the consequences of the simple model change led us to a revision of the standard adaptive control set-up. The relevant considerations are presented here.

Definition 1[*Past, future*] Let s be a specific predicted ($s = y$) or designed ($s = u$) signal. The related *past* ($\mathcal{P}$) consists of all data which have been used for the signal prediction ($s = y$) or determination ($s = u$). The remaining data which are to be predicted/determined are called *future* ($\mathcal{F}$).

For descriptions of time relations on some interval, we deal with a past $\mathcal{P}_b$ *extending* to $\mathcal{P}_a = (\mathcal{P}_b, \mathcal{P}_{ab})$: $\mathcal{P}_a$ is the past $\mathcal{P}_b$ enriched by a past increment $\mathcal{P}_{ab}$ (by new data). Through this evolution, the future ($\mathcal{F}_b$) corresponding to $\mathcal{P}_b$ shrinks to a future ($\mathcal{F}_a$) by a decrement ($\mathcal{F}_{ba} = \mathcal{P}_{ab}$). □

Control design

Adaptive controllers (approximately) solve:

Stochastic control task. A nonnegative *loss function* $\mathcal{K}(u, \mathcal{P}, \mathcal{F})$ is used as closed loop performance index. An *optimal strategy* minimizing expected loss ($\mathcal{E}[\mathcal{K}]$) is searched for among *admissible strategies*, mapping (possibly in a random way) the past on inputs selected from a range $\mathcal{U}$.

Proposition 1[*Solution of control task*] The deterministic strategy is optimal, which assigns the past $\mathcal{P}$ the minimizing arguments in $\min_{u\in\mathcal{U}} \mathcal{E}[\mathcal{K}(u, \mathcal{P}, \cdot)|u, \mathcal{P}]$
Proof. See e.g. [1] □

Multistage stochastic control task. A nonnegative *loss function* $\mathcal{K}(u_a, \mathcal{P}_a, \mathcal{F}_a)$ depending on the extended past $\mathcal{P}_a = (\mathcal{P}_b, \mathcal{P}_{ab})$, the input $u_a = (u_b, u_{ab}) \in (\mathcal{U}_b, \mathcal{U}_{ab})$ and on the corresponding future $\mathcal{F}_b = (\mathcal{F}_a, \mathcal{F}_{ba})$ measures closed loop performance. An *optimal strategy* minimizing expected loss is searched for among *admissible strategies* which map $\mathcal{P}_i \to \mathcal{U}_i$, $i = b, ab$.

Proposition 2[*Solution of multistage task, dynamic programming*] The deterministic strategy assigning to the relevant pasts $\mathcal{P}_i$, $i = a, ab$ the minimizing arguments in the sequence

$$\min_{u_b \in \mathcal{U}_b} \mathcal{E}\left[\min_{u_{ab} \in \mathcal{U}_{ab}} \mathcal{E}[\mathcal{K}(u_b, u_{ab}, \mathcal{P}_a, \cdot)|u_{ab}, \mathcal{P}_a]|u_b, \mathcal{P}_b\right] \tag{6}$$

is the optimal solution of the multistage control task. (It is a version of dynamic programming.)
Proof. Direct consequence of Proposition 1 and of the nesting of the pasts involved. □

Modelling and estimation

The model of the controlled system needed for the multistage design has to provide conditional expectations in (6). Generally, full distributional knowledge is needed for this evaluation. Without restricting scope of this paper, we assume it given by the conditional probability density functions (p.d.f.) $p(\mathcal{F}|u, \mathcal{P})$ (identical symbols for a random variable, its realization and the p.d.f. argument are used, as usual).

The p.d.f. $p(\mathcal{F}|u, \mathcal{P})$ is mostly too complex to be directly available from a system modelling. For this reason, it is gained through *parameter estimation*: Parameterized models

$$p(\mathcal{F}|u, \mathcal{P}, \underline{\Theta}) \tag{7}$$

are constructed and an unknown (multivariate) parameter $\underline{\Theta}$ is estimated (eliminated by observing data on the particular system of interest). *Bayesian identification* we are exploiting eliminates $\underline{\Theta}$ exactly for any past according to the elementary probabilistic rule

$$p(\mathcal{F}|u, \mathcal{P}) = \mathcal{E}\left[p(\mathcal{F}|u, \mathcal{P}, \underline{\Theta})|u, \mathcal{P}\right] \tag{8}$$

for which the Bayesian parameter estimate $p(\underline{\Theta}|u, \mathcal{P})$ is needed. The required estimate is gained by exploiting the parameterized system model and observed data for modification (updating) a prior p.d.f. of $\underline{\Theta}$.

Proposition 3 [*Bayes rule: updating of parameter estimates*] Let the informational structure be as in multistage stochastic control task and control strategies be deterministic and admissible ones (u_b is a deterministic function of $\mathcal{P}_b$). Then $p(\underline{\Theta}|\mathcal{P}_b) = p(\underline{\Theta}|u_b, \mathcal{P}_b)$ and

$$p(\underline{\Theta}|\mathcal{P}_a) \propto p(\mathcal{P}_{ab}|u_a, \mathcal{P}_b, \underline{\Theta}) p(\underline{\Theta}|\mathcal{P}_b) \tag{9}$$

($\propto$ means proportionality by $\underline{\Theta}$-independent factor). The formula (9) is called Bayes rule.
Proof. A statement of probability theory for nested pasts and a simplified version of so called natural conditions of control [9] are exploited. □

For the models (3), the parameter $\underline{\Theta}$ consists of regression coefficients and noise dispersions.

Adaptive control

The adaptive controllers are optimal-control approximations, which use updated parameter estimates for new control design in every period. Both estimation and synthesis consist of action pairs:

$$identification \begin{cases} extending\ of\ past \\ parameter\ estimation \end{cases}$$

$$design \begin{cases} computing\ expectation\ \mathcal{E}[\cdot|\cdot] \\ criterion\ minimization \end{cases}$$

The sequence of these actions is not fixed, the computations can be performed separately; in order to use any piece of information as soon as it is available, they can be controlled by data flow. However, *irregularity of time intervals* must be taken into account. The *spline based modelling* in the form (5) enables us to do so.

Time-scheduling

Definition 2 [*Scheduling, dummy signal, spline model*] We shall call *scheduling* a one-to-one mapping $t \to \omega(t) = (s(t), i(t), \Delta(t))$, $s(t) \in \{y, u\}$, $i(t) \in \{1, \ldots, m_{s(t)}\}$, $\Delta(t) > 0$ which specifies the newest signal (s_i) measured/decided and its time shift (Δ) from the preceding sample of the same signal.

Signal samples will be described using *dummy scalar signal* $\mathcal{S} = Y$ for $s = y$ and $\mathcal{S} = U$ for $s = u$:

$$\begin{aligned} \mathcal{S}(t) &= s_{i(t)}(\zeta_{i(t)}(t)), \text{ with "individual" time} \quad (10) \\ \zeta_i(t) &= \sum_{\tau} \text{Ind}(\tau \le t, s(\tau) = s(t), i(\tau) = i(t))\Delta(\tau) \end{aligned}$$

where $Ind(\cdot)$ denotes set indicator. The elements of the spline model (5) are unified similarly

$$\begin{aligned} \Theta(t) = \Theta_{i(t)},\ G(t) = g_{i(t)}(\zeta_{i(t)}) \\ \Psi(t) = \psi_{i(t)}(\zeta_{i(t)}(t)),\ E(t) = e_{i(t)}(t) \end{aligned} \tag{11}$$

Scheduling determines uniquely the associated past $\mathcal{P}(t)$ = data sampled before $\mathcal{S}(t)$ and future $\mathcal{F}(t)$ = data which may occur after $\mathcal{S}(t)$. □

Proposition 4 [*Scheduling for adaptive control*] For deterministic control strategies, the nontrivial past extensions are generated by the measured outputs: if $Y(t) \neq Y(t-h)$ then $\mathcal{P}_a = \mathcal{P}(t) \supset \mathcal{P}(t-h) = \mathcal{P}_b$.

The adaptive control can be used with a deterministic scheduling if for any non-trivial past extension the model (7) with a constant parameter $\underline{\Theta}$ exists.
Proof. By construction, the input generated by deterministic strategy brings no information about the observed system until an output influenced by it is measured. The constant $\underline{\Theta}$ is prerequisite of the Bayes rule validity (9). This rule generates the model (8) needed for the control design. □

Remarks

1. Uniqueness of the scheduling is achieved when no pair of sampling moments coincide. Otherwise a complementary ordering rule, say lexicographical, has to be added.
2. The need for invariance of $\underline{\Theta}$ explains our choice of continuous-time modelling: in a discrete-time models the freedom in scheduling choice is substantially restricted.
3. The model (5, 11) violates the invariance condition only seemingly: the subscript points to various parts of a constant parameter $\underline{\Theta}$.
4. Deterministic scheduling is treated for simplicity, avoiding Markov-time framework.

MISO control design: reduction to SISO

Freedom in action timing leads immediately to *reduction of MISO to SISO*. If the *strategies which select at* *most single input after each past extension are admissible*, the optimization reduces to a sequence of single input tasks.

Combined use of the models (5) with such strategies leads to the reduction of MIMO design to a set of interconnected SISO tasks.

Design of the controller revised

Control quality optimized by the controller is quantified by the expected value of the quadratic loss

$$\frac{1}{T}\{Q_\psi[\Psi(T) - \Psi_0(T)] + Q_X[X(T)]\} \qquad (12)$$

with the vector $X(T)$ containing jumps of signal-sample deviations from their reference values. (For brevity, the argument T is omitted hereafter).

Model used for the prediction of the controlled system behavior is a SISO *linear* (*Gaussian*) regression

$$Y(t) = \Theta'(t)G(t)\Psi(t) + E(t) \qquad (13)$$

Admissible control strategies are restricted by the domain $\mathcal{P}(t)$ and range $\mathcal{U}(t)$. **The adaptive controller** estimates recursively the unknown coefficients of the model (13) using them for approximative minimization of the expected value of the quadratic loss (12).

Algorithmic aspects

The above formulation is a bit academic until efficient algorithmic solution is designed. Here, the solution is outlined for the main design steps.

Structure estimation

From algorithmic view point, the structure estimation is slightly influenced by the changes made. In [8], an efficient algorithm is sketched which searches for maximum within the space of posterior probabilities of all hypotheses about regressor structures. The algorithm is able to deal effectively with sub-regressors of the regressor with about 100 entries. In the current formulation, the algorithm is directly applicable in a loop over the predicted outputs. As the output dimension m_y is relatively small no problems are foreseen.

The new task of selecting the best factor M in (4) adds some complexity to the structure estimation which is well balanced by the richer outcomes. The key computational tricks of the basic search [8] are directly applicable.

Parameter estimation

Just m_y independent RLS are applied to MISO models working on filtered regressors $G(t)\Psi(t)$.

The gained freedom in time-scheduling can be simply illustrated on this subtask. C-language-type notation is used for describing the action sequence

```
if(s(t)==s(t-h) && i(t)==i(t-h)) do nothing;
else if(s(t)==u) update relevant regressors;
     else { identify i(t)th parameters;
            update relevant regressors; }
```

The possibility to use channel-allocated forgetting should be mentioned under this heading.

Control synthesis

The above theory shows that irregular sampling can be managed. At the same time, it is an example of often met situation that the solution via formulae looks awkward while the algorithm is simple.

As it is seen from Proposition 2 scheduling runs against the course of real time

```
if(s(t+h)==s(t) && i(t+h)==i(t)) do nothing;
else if(s(t)==y) take expectation;
     else minimize over i(t)th input;
```

For algorithmization of both steps, we adopt the factorization-based optimization [2]. It suits both because of excellent numerical properties and because of ease of coping with the faced generalized conditions.

The algorithmization is based on a pair of elementary Propositions. For presentation simplicity, terminal penalty Q_ψ and reference values are set to zero.

Proposition 5[*Conditional expectation of quadratic form*] Let the vector $\bar{X}' = [y, X']$ consist of a (scalar) output y and samples in its past $\mathcal{P}$. Let a regression model determine the expected value $\mathcal{E}[y|u, \mathcal{P}, \Theta] = \Theta'\psi$. Let $\bar{W}$ be factor of the weighting matrix in the quadratic form $\bar{W}[\bar{X}]$ (cf. notation at (1)). Then, $\mathcal{E}[\bar{W}[\bar{X}]|u, \mathcal{P}, \Theta] = W[X] + \text{const}$. The weight W is determined by the equation

$$\bar{W}[\tilde{X}] = W[X] \qquad (14)$$

with $\tilde{X} = [\Theta'\psi, X']$.
Proof. Elementary evaluation of moments. □

The new factor W is by no means uniquely defined. The freedom is used for minimization.

Proposition 6[*Orthogonal-transformation-based minimization*] Let the vector $X' = [u, \underline{X}']$ consist of a (scalar) input u and vector $\underline{X}$ of samples in its past $\mathcal{P}$. Then, a version of the factor W exists such that

$$W[X] = \left(w(u + L'\underline{X})\right)^2 + \underline{W}[\underline{X}] \qquad (15)$$

where w is a scalar weight, and L a column vector (control law). The input $u = -L'\underline{X}$ minimizes the quadratic form $W[X]$. If W is any weight determining $W[X]$ (14), then there is an orthogonal matrix $\mathcal{O}$ such that

$$\mathcal{O}W = \begin{bmatrix} w & L' \\ 0 & \underline{W} \end{bmatrix}. \qquad (16)$$

Proof. See elsewhere, e.g. [2]. □

The re-computation of the weighting matrix $\bar{W}$ to $\underline{W}$ (equivalent to Riccati equation) starts from $\bar{W} = Q_X$. Subsequent application of two steps described in Prop. 3, 4 – controlled by the scheduling mapping – forms the overall synthesis. The procedure is finished when all minimizing inputs within the control horizon are found.

Remarks

1. The adaptive controller approximates the optimal solution by using newest point estimates instead of unknown regression coefficients.

2. The procedure can be interpreted as sequential removal of future samples from the vector $X(T)$. Expectation substitutes an output by its regression and minimization removes corresponding input by zeroing the product $[1, L'] \begin{bmatrix} u \\ \underline{X} \end{bmatrix}$.
3. The time instants of taking the expectation or minimization can be quite arbitrary and models used may have quite different structures.
4. The evaluations admit not only irregular switching between minimizations and expectations but also irregular switching within the set of signals treated: for instance, expectation can be applied several times to a single signal.
5. Note, that scheduling points not only to the signal treated but also to the corresponding weight assigned to a signal in Q_X: it determines whether and to which extent the signal is penalized by the original criterion.
6. It may seem that such a general structure would make the computation very tedious and space demanding as we formally deal with a huge (T/h,T/h)-type matrix (cf. Q_X in (12)). The penalization matrix is, however, band path with elements specified by a few different entries: typically common scalar weights of squares of particular signals $q_{si}, i = 1, \ldots, m_s;\ s \in y, u$ and the operations at some row of W influence it to the depth of the longest regressor. Moreover, during full operation pair two "end" lines are cancelled.

   Thus, the array where real evaluations are performed is much smaller than $(T/h, T/h)$ as both the nonzero band of Q_x and the longest regressor are much shorter than this dimension.

   Essentially we store a cluster of nonzero elements which move from the left upper corner of the array of Q_X-dimensionality to its right bottom corner. This cluster has regular upper triangular form in classical cases.
7. The weight transformation caused by taking expectation can be represented by a matrix multiplication when using appropriate state space model. However, no explicit state space models are necessary in reality. The column corresponding to the variable is deleted and the weights are modified by regression coefficients.
8. For minimization, the orthogonal transformations leading to the weight form (16) is sufficient. Application of further orthogonal transformations can restore the upper quasi triangular form (after adequate column exchange the left part of the cluster would have the upper triangular form). This will keep the cluster more compact.

Conclusions

A recently proposed description of a multi-input multi-output regression [6] has been exploited for broadening the applicability of the CAD package DESIGNER to MIMO systems. The description is based on separated models for the particular outputs. Such a parameterization is in various forms a part of "modelling folklore" but its exploitation has been mostly neglected. This model modification leads to generalization in the scope of the described systems (e.g. different sampling rates allowed) and improves the possibility of detecting and using special system structures.

Behind a usual research report, a story is revealed; a story telling how slowly the progress proceeds, how theory and computation actually stimulate each other:

Theory of multi-input multi-output linear quadratic Gaussian adaptive control led to need of computer aided design, its development resulted in probabilistic structure estimation. Overcoming implementation problems of this theory brought about powerful results, stimulating the introduction of a modified system model. The model itself now opens the field of multi-rate sampling and sequential control synthesis and...

References

[1] Aström K. J. (1970). Introduction to Stochastic Control. Academic Press.

[2] Kárný M., A. Halousková, J. Böhm, R. Kulhavý, P. Nedoma (1985). Design of linear quadratic adaptive control: theory and algorithms for practice. Kybernetika, **21**, Supplement to numbers 3, 4, 5, 6.

[3] Kárný M., J. Böhm, I. Nagy, A. Halousková (1990). Design of Spline-based Self-Tuners. Kybernetika, **26**, 17-30.

[4] Kárný M., Halousková A. (1991). Implementing LQG adaptive control: a CAD approach. 9th IFAC/IFORS Symposium Identification and System Parameter Estimation, Budapest 1991, **2**, 1585-1590.

[5] Kárný M., J. Böhm (1991). Probabilistic modelling of imprecisely known systems for robust LQG design. ECC'91 Grenoble, **1**, 426-431.

[6] Kárný M. (1992). Parametrization of multi - output multi - input autoregressive-regressive models for self-tuning control. To appear in Kybernetika **28**.

[7] Kárný M., Halousková A.(1991). User supplied information in the design of linear quadratic Gaussian self-tuning controllers. Accepted for Symposium ACASP'92 Grenoble.

[8] Kárný M., R. Kulhavý (1988). Structure determination of regression-type models for adaptive prediction and control. In: Spall J. C. (ed.): *Bayesian Analysis of Time Series and Dynamic Models*. Marcel Dekker, New York.

[9] Peterka V. (1981).'Bayesian system identification. In: Eykhoff P. (ed.): *Trends and Progress in System Identification*, Pergamon Press Oxford, 239-304, 1981.