CONTROLLER DESIGN AND QUALITY EVALUATION

M. Novák [*] J. Böhm [*]

[*] *Department of Adaptive Systems*
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, CZECH REPUBLIC
fax: +420 2 6605 2068 and e-mail: mira@utia.cas.cz

Abstract: Designing a controller for particular use requires a proper setting of its tuning knobs. The searched controller has to track prescribed setpoint properly while keep some constraints placed on its action variables. The tuning is well explored for three terms controllers. But, almost nothing can be found about tuning of LQG controllers. Tuning presented in this contribution is done for multidimensional adaptive LQG controller. Controller quality is evaluated by Monte Carlo approach using identified system model for performing off-line simulations. It is necessary to keep simulation as short as possible because lots of simulations has to be performed during tuning process. This contribution estabilishes on-line stopping rule for simulation run to achieve enough precise controller with the smallest possible computational demands.

Keywords: LQG controller, adaptive controller, controller design, optimization

1. INTRODUCTION

The adaptive controllers based on modern control theory are still not fully exploited because of incompleteness of solutions and extreme demands on commission skills. Therefore, the simple three-term controllers and firmware tailored solutions are mostly used. A bridge between academicians providing theoretical solution of control problems and designers in real processing is needed—a translator from the theoretical language to the practical one and back. Such a tool is being created—a Matlab toolbox DESIGNER—making an automated off-line design of adaptive controllers. This toolbox combines control aims, restrictions, off-line measurements, and prior knowledge about the system in an automatic way. It provides a completely specified adaptive controller.

1.1 *Controller Tuning Problem Formulation*

Controller tuning is a process aiming at the correct controller set-up to fulfill given constraints and requirements. A controller depends on certain parameters, called tuning knobs, which have to be set properly to obtain desired control loop behavior. Model-based predictive controllers are, in some sense, optimal. However, the optimality is conditioned by the perfect model fit to the controlled plant. And, the optimality from user's point of view need not match the kind of optimality acceptable for the controller.

The reasons for controller tuning are:

(1) The assumed controller uses a system model that does not fit to the reality due to incorrectly identified model parameters, or even the structure or type of model is not perfect. The tuning knobs have to be set to suppress

control error caused by the model mismatch. In another words the controller setting influences its robustness.

The examples of incorrect controller's model: Model is stochastic, has uncertain parameters, is non-linear, and quantities are constrained while the use of LQG controller allows just stochastic linear model with Gaussian noise and with known model parameters.

(2) The optimality criterion of the controller is not able to express the user's desired kind of optimality. The selected controller requires a different formulation of the task. The tuning process converts the desired optimality into the form acceptable by controller.

The example of different optimality conditions: Criterion is quadratic, while the wishes are expressed through desired intervals for quantities.

The constraints in these two examples are also called hard in the first case, because they come from the model itself and soft in the second one, because they are the user's wishes where the variables should approximately be, and it is often allowed to slightly violate them.

The tuning is an optimization method searching the best tuning knob values. The controller behavior is evaluated from predicted closed loop performance.

The tuning, as a process, operates with a wider amount of information than it is acceptable for the tuned controller itself. It can use more precise model of the system, the optimality can include more subtle conditions, or even it can observe the control loop as a whole and optimize the good model excitation for adaptive controllers. The fact allowing to tune the controller in much more complex situation is, that the tuning is done off-line. Thus, it is just weakly limited by numerical feasibility of control action calculation for working on-line in closed loop. Therefore tuning cannot be used directly to generate system input on-line.

The prediction of closed loop performance uses simulation of a model identified from data measured on real plant and user supplied prior information. To use the tuning to reduce drawback of controller model—reality mismatch, reason 1, the identified model must be closer to the reality than the model used by the controller. A sketch of controller tuning process is shown in figure 1. Using model from the exponential family, Section 4, has this properties while its identification and simulation is numerically feasible.

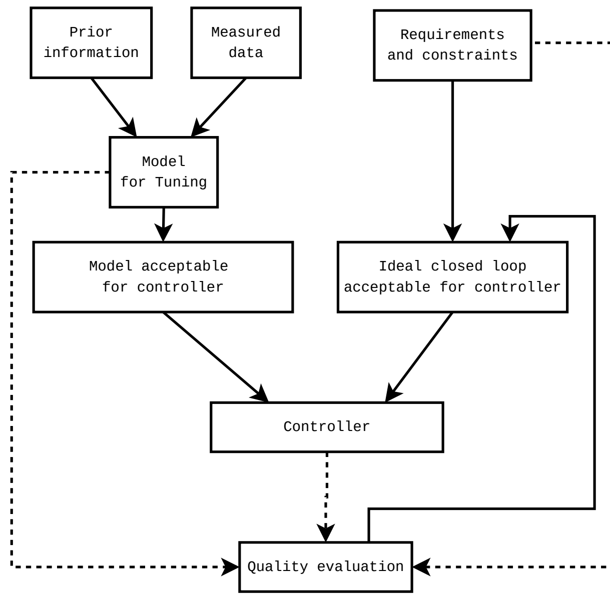

Fig. 1. Conceptual flowchart of controller tuning process.

2. SYMBOLS

Symbol f denotes a probability density function (pdf) for continuously-valued random variable or probability for discrete-valued random variable. The f is distinguished by identifiers in its arguments. Symbol x' denotes a transposition of matrix x. The set of possible values of variable x is denoted by x^*. The number of elements of vector x or number of members of the set x^*, for countable sets, is denoted by $\mathring{x}$. Time t is discrete and integer.

Data d_t in time t consists of input u_t and output y_t vectors, i.e. $d_t = [y_t', u_t']'$. The data are grouped under the following identifiers

$$
\begin{aligned}
\Psi_t &= [y_t', u_t', y_{t-1}', u_{t-1}', \ldots, y_{t-\partial}', u_{t-\partial}', 1]' \\
\psi_t &= [u_t', y_{t-1}', u_{t-1}', \ldots, y_{t-\partial}', u_{t-\partial}', 1]' \\
\varphi_t &= [y_t', u_t', y_{t-1}', u_{t-1}', \ldots, y_{t-\partial+1}', u_{t-\partial+1}', 1]' \\
\Psi_t &= [y_t', \psi_t']' = [y_t', u_t', \varphi_{t-1}']',
\end{aligned}
$$

where symbol ∂ denotes the maximal delay of signals required in the model. The sequence $\{d_\tau\}_{\tau=1-\partial}^{t}$ is denoted by $d(t)$, the same holds for other time dependent variables. Data with negative time represents the prior information such as initial state, initial estimate, etc. Symbol $\mathcal{E}x$ represents unconditional expected value of random variable x. The conditional expected value of x conditioned by y is written as $Ex|y$. Set of all real numbers is denoted by $\mathbf{R}$.

3. GENERAL PRINCIPLE OF CONTROLLER TUNING

Controller tuning task consists of controlled system model, controller itself, parameterized by corresponding tuning knobs, and constraint and objective function definitions.

General description of the dynamic stochastic system is given by pdf

$$f(y_t|u_t, d(t-1)). \tag{1}$$

The actions u_t generated by, generally randomized, controller are described by pdf

$$f(u_t|d(t-1), q), \tag{2}$$

where q denotes the tuning knobs.

Whole evolution of system up to time $t \geq 0$ is described recursively

$$\begin{aligned} &f(d(t)|q) = \\ &f(y_t|u_t, d(t-1))f(u_t|d(t-1), q)f(d(t-1)|q) \end{aligned} \tag{3}$$

starting from known initial data $d(0)$.

The constraints are described by data dependent function Z_c being non-positive, when constraints are met

$$Z_c : d(\mathring{t})^* \mapsto \mathbf{R}^{\mathring{c}}, \quad Z_c \leq 0, \tag{4}$$

where $\mathring{c}$ is dimension of space the function maps to. The controller performance objective function

$$Z_o : d(\mathring{t})^* \mapsto \mathbf{R} \tag{5}$$

is decreasing with increasing controller performance. Symbol $\mathring{c}$ denoused dimension of constrained quantities vector.

The controller tuning is formulated as the stochastic optimization task

$$\begin{aligned} &\text{minimize } \mathcal{E}\{Z_o|q\} \\ &\text{subject to } \mathcal{E}\{Z_c|q\} \leq 0 \\ &\text{over tuning knobs } q \in q^*. \end{aligned} \tag{6}$$

This was obtained applying conditional expectation and using (3) in (4) and (5).

The particular construction elements of the optimization task is described in the following chapters.

4. SYSTEM MODEL

Let us assume that the system model $f(y_t|u_t, d(t-1))$ is known up to a finite-dimensional parameter Θ

$$f(y_t|u_t, d(t-1), \Theta), \tag{7}$$

where Θ is inferred from past data

$$f(\Theta|d(t-1)). \tag{8}$$

The u_t is missing in condition due to natural condition of control Peterka (1981).

The pdf (8) is exploited during the system identification and also during the multiple step ahead predictive pdf or simulation. When controller is adaptive and model based it estimates the Θ as well.

4.1 *Simulation in Exponential Family*

A system model pdf (7) belongs to the exponential family if it can be written in form

$$\begin{aligned} &f(y_t|u_t, d(t-1), \Theta) = \\ &= f(y_t|\psi_t, \Theta) = A(\Theta)\exp(B'(\Psi_t)C(\Theta)), \end{aligned} \tag{9}$$

where A is a non-negative real function on Θ^*, and B and C are scalar functions with same dimensions defined on Θ^* and Ψ_t^*.

The corresponding pdf of Θ is

$$\begin{aligned} &f(\Theta|d(t)) = \\ &= f(\Theta|V_t, \nu_t) = \frac{A^{\nu_t}(\Theta)\exp(V_t'C(\Theta))f(\Theta)}{I(V_t, \nu_t)}, \end{aligned} \tag{10}$$

where V_t, ν_t is sufficient statistics updated recursively

$$\begin{aligned} (V_{t-1}, \nu_{t-1}, \Psi_t) &\mapsto (V_t, \nu_t) \\ V_t &= V_{t-1} + B(\Psi_t) \\ \nu_t &= \nu_{t-1} + 1. \end{aligned} \tag{11}$$

The symbol $\mapsto$, in this context, denotes existence of such a function that maps variable on the left side to the variable on the right side. The recursion (11) starts from V_0, ν_0 determining the prior conjugated pdf $f(\Theta)$, see Peterka (1981). Factor $I(V_t, \nu_t)$ normalizes the pdf (10).

The predictive pdf of the system can be obtained from (9) and (10)

$$\begin{aligned} &f(y_t|u_t, d(t-1)) = \\ &= f(y_t|\psi_t, V_{t-1}, \nu_{t-1}) = \frac{I(V_t, \nu_t)}{I(V_{t-1}, \nu_{t-1})}. \end{aligned} \tag{12}$$

Now, the multiple step ahead predictive pdf with statistic conditioned by system inputs is given recursively for $t \geq 0$, starting from known V_0, ν_0, and ψ_0, as

$$f(y(t)|u(t)) = \prod_{\tau=1}^{T} f(y_\tau|\psi_\tau, V_{\tau-1}, \nu_{\tau-1}). \tag{13}$$

4.2 *Simulation of Gaussian ARX model*

An important type of dynamic model from the exponential family is the Gaussian ARX model. This model with parameter Θ consisting of regressive parameter θ and Gaussian noise covariance R has pdf (9) realized in form

$$f(y_t|\psi_t,\Theta) = f(y_t|\psi_t,\theta,R) \sim \mathcal{N}(\theta\psi_t, R), \quad \Theta = (\theta, R). \tag{14}$$

The random variable Θ from pdf (10) has now Gauss-inverse-Wishart distribution Peterka (1981)

$$f(\theta,R|V_t,\nu_t) = \alpha_t |R|^{-\frac{\nu_t}{2}} \times \times \exp\left\{-\frac{1}{2}\mathrm{tr}\left(R^{-1}\begin{bmatrix}-I\\ \theta\end{bmatrix}' V_t \begin{bmatrix}-I\\ \theta\end{bmatrix}\right)\right\}, \tag{15}$$

where α_t is normalizing constant.

The system output y_t from the outer model single step ahead predictive pdf (12) has now Student distribution

$$f(y_t|\psi_t,V_{t-1},\nu_{t-1}) = = \frac{\kappa_t}{\left(1+\frac{(y_t-\hat{\theta}'_{t-1}\psi_t)'\Lambda_{t-1}^{-1}(y_t-\hat{\theta}'_{t-1}\psi_t)}{1+\zeta_t}\right)^{\frac{\nu_{t-1}+\mathring{y}}{2}}}, \tag{16}$$

where κ_t is normalizing constant. Variables $\hat{\theta}_t$, Λ_t, and ζ_t are obtained from the split

$V_t = \begin{bmatrix} V_{y,t} & V'_{y\psi,t} \\ V_{y\psi,t} & V_{\psi,t}\end{bmatrix}$, with $\mathring{y}$ dimensional square $V_{y,t}$

and

$$\begin{aligned}\hat{\theta}_t &= V_{\psi,t}^{-1} V_{y\psi,t}\\ \Lambda &= V_{y,t} - V'_{y\psi,t}\hat{\theta}_t\\ \zeta_t &= \psi'_t V_{\psi,t}^{-1}\psi_t.\end{aligned}$$

It is hard to calculate the multiple step ahead predictive pdf (13) in closed form. Still it can be sampled using model simulation.

ARX model simulation is done by recursively sampling the recursive definition of the multiple step ahead predictive pdf with statistics included (13).

The Student distribution (16) converges to Gaussian distribution with number of degrees of freedom going to infinity

$$f(y_t|\psi_t,V_{t-1},\nu_{t-1}) \overset{t\to\infty}{\to} \overset{t\to\infty}{\to} \mathcal{N}\left(\hat{\theta}'_t\psi_t, \frac{1+\zeta_{t-1}}{\nu_{t-1}-\mathring{\psi}}\Lambda_{t-1}\right) \tag{17}$$

thus it is reasonable to sample this approximation instead.

5. CONTROLLER

Let us restrict to a subset of controllers (2) that depends only on a finite-dimensional state Υ_t, structure of which is determined by the particular type of controller used. The "predictive" pdf of such controller is

$$f(u_t|y_{t-1},\Upsilon_{t-1},q). \tag{18}$$

The state evolves according to controller type

$$\Upsilon_{t-1} \mapsto \Upsilon_t. \tag{19}$$

5.1 *Control loop*

Using chain rule with the predictive pdf (12) of the system model from exponential family, a predictive pdf of data d_t is obtained

$$f(d_t|\varphi_t,V_{t-1},\nu_{t-1},\Upsilon_{t-1},q). \tag{20}$$

Now, a multiple step ahead predictive pdf can be calculated recursively for $t \geq 2$, starting from (20) with $t=1$, as

$$f(d(t)|q) = = \prod_{\tau=1}^{\mathring{t}} f(d_\tau|\varphi_\tau,V_{\tau-1},\nu_{\tau-1},\Upsilon_{\tau-1},q) \tag{21}$$

The pdf of data over the whole time span t^* is needed by the loss functions for closed loop performance evaluation in the optimization task (6).

It is generally hard to find pdf (21) in closed form. But assuming one can sample y_t from (13) and the u_t from (18), the sample of whole $d(\mathring{t})$ is possible to obtain by recursive sampling of the pdfs according to (21). This is equivalent to closed loop simulation.

5.2 *LQG Controller*

The adaptive LQG controller is used here to demonstrate the state variable and tuning knobs. Adaptive controller tracks recursively the estimates of parameters of the Gaussian ARX model (15). Moreover, it uses exactly the same algorithm as described in section 4.2. Thus the controller state Υ_t is equal to the set of statistics of the model simulator

$$\Upsilon_t = (V_t, \nu_t, \varphi_t).$$

Adaptive LQG controller uses the current point estimates $\hat{\theta}_{t-1}$ while minimizing over the T-th receding-horizon the quadratic loss J_t

$$J_t = \sum_{\tau=t}^{t+T}\begin{bmatrix}y_\tau\\ \psi_\tau\end{bmatrix}' Q_t \begin{bmatrix}y_\tau\\ \psi_\tau\end{bmatrix} + \varphi'_{t+T}S_T\varphi_{t+T}, \tag{22}$$

determined by a positive semi-definite matrices Q_t, S_T.

The optimal inputs are generated by linear control law $u_t = -K'(\hat{\theta}_{t-1}, Q_t)\varphi_{t-1}$ with the state vector φ_{t-1}. Matrix gain $K(\cdot)$ results from dynamic programming that reduces to the solution of Riccati equation. The gain $K(\cdot)$ is re-computed whenever updated parameter estimates are obtained or penalization matrix Q_t is changed.

Often, Q_t is time invariant and consists of a block Q_u penalizing inputs and a block Q_φ penalizing the state φ_t. If we convert the input-output model into its state version, with state transition matrix A and the input-gain matrix B made of current point estimates $\hat{\theta}_{t-1}$, the control law has the form

$$K = (Q_u + B'S_0B)^{-1}B'S_0A. \tag{23}$$

The positive semi-definite Riccati matrix S_0 determining its results from the Riccati equation

$$\begin{aligned} S_i = {} & A'S_{i+1}A - A'S_{i+1}B(Q_u + \\ & + B'S_{i+1}B)^{-1}B'S_{i+1}A + Q_\varphi, \end{aligned} \tag{24}$$

starting from known S_T. Note that a factorized equivalent of (24) is used for safe numerical evaluations, see Kárný (1992).

5.2.1. *Tuning Knobs of the LQG Controller* All parameters of a controller being constant through the simulation process can be tuned by the adopted methodology. In case of the LQG controller, the tuning knobs are represented by the kernel Q of its quadratic criterion (22), the initial value S_0 of the Riccati matrix, cf. (22), (24), and even the horizon length T. Only the matrix Q is tuned in this work for sake of simplicity leaving the other parameters to be set explicitly.

The number of independent elements of matrix Q is often high. Then its full optimization is computationally intensive. Moreover, the necessary positive semi-definiteness of the matrix Q represents a significant constraint placed on its entries.

It is also worthwhile of knowing the physical meaning of particular tuning knobs, which is in situation of whole matrix Q rather difficult. Thus, the quadratic criterion constructed from smaller and simpler pieces is suitable to decrease the problem dimensionality. It is wise to parameterize Q by tuning knobs with good intuitive physical meaning. It allows us to get control over particular properties of the controller. The following quadratic criterion is considered

$$J_t = \sum_{\tau=t}^{t+T}(q_1 l_{1;\tau}^2 + q_{2;\tau} l_{2;\tau}^2 + \ldots + q_{\mathring{q}} l_{\mathring{q};\tau}^2), \tag{25}$$

where the scalar weights $q_\bullet \geq 0$, called penalization coefficients, are taken as the tuning knobs. The linear vector function l_t depends on quantities y_t and ψ_t and measures the signal deviations from the desired values. Generally, the overall criterion (25) depends on quantities y_t and ψ_t in the same way as the full quadratic criterion (22) does. It makes sense, however, to fix the linear functions and let the designer of the LQG controller to find the weights $q_\bullet$ only.

Typical forms of the quadratic criterion, given by specific function l_t, and their correspondence with the constraints placed on particular signals follow.

The regulation problem is the simplest variant. The controller tries to keep the output close to the desired set point y^{ref} and the input close to its reference value u^{ref}. Penalizations are then chosen to penalize the regulation error and to penalize difference of the input from its reference value

$$\begin{aligned} l_t = {} & \left[y_{1;t} - y_1^{\text{ref}}, y_{2;t} - y_2^{\text{ref}}, \ldots, \right. \\ & \left. u_{1;t} - u_1^{\text{ref}}, u_{2;t} - u_2^{\text{ref}}, \ldots \right]. \end{aligned} \tag{26}$$

Penalization weights belonging to the model output $q_1, \ldots, q_{\mathring{y}}$ and input $q_{\mathring{y}+1}, \ldots, q_{\mathring{y}+\mathring{u}}$ have to be set to represent optimal trade-off between regulation error and actuator effort. The discussed penalizations suit well for tuning of the controller that respects constraints on the range of the system input while minimizing regulation error.

The joint servo-regulation problem is obtained by allowing time dependent y^{ref}. Then, limits on input increments are put more often. Then, the appropriate penalization of input increments is

$$\begin{aligned} l_t = {} & \left[y_{1;t} - y_{1;t}^{\text{ref}}, y_{2;t} - y_{2;t}^{\text{ref}}, \ldots, \right. \\ & \left. u_{1;t} - u_{1;t-1}, u_{2;t} - u_{2;t-1}, \ldots \right] \end{aligned} \tag{27}$$

that corresponds with time variant $u_t^{\text{ref}} = u_{t-1}$. This discrete-time analogy of the first derivative can be extended to penalization of the discrete-time analogy of higher order derivatives.

To cope with the constraints of mutually dependent signals in case of MIMO system, the respective l_t vector function entry has to include these signals. For instance, closeness of two input signals is controlled by the following entry

$$u_{i;t} - u_{j;t}, \quad i \neq j. \tag{28}$$

This kind of penalization is called non-diagonal because of its matrix representation as quadratic form.

6. CLOSED LOOP PERFORMANCE EVALUATION

In this section, the requirements and constraints placed on the ideal closed loop behavior are defined. Their fulfillment is measured by loss functions Z_o and Z_c. The construction of these functions is described.

6.1 *Objective Function*

The objective expresses commonly the wish that the quality of the regulation process in certain sense should be as high as possible subject to the present constraints. The typical wish on small tracking error of i-th output is expressed by the objective function Z_o

$$Z_o^T = \frac{1}{T}\sum_{\tau=1}^{T}(d_\tau - d_\tau^{\mathrm{ref}})' W (d_\tau - d_\tau^{\mathrm{ref}}), \quad (29)$$

where W is a positive semi-definite matrix of appropriate dimensions.

The matrix W is usually diagonal with non-zero only those elements corresponding to signals in d with prescribed reference trajectory or setpoint. The values of the non-zero elements of W are usually chosen to be reciprocal to the variances of respective signals in d. This approach puts more importance on proper tracking of less noisy channels, while channels with higher variance take less effort of controller.

This function is used as the objective in the definition of the optimal controller (5).

6.2 *Constraints*

Constraints are placed usually not only on the input and output quantities magnitudes but also on their dynamic behavior such as limited increments. To cope with these constraints uniformly, a vector variable c_t containing all constrained quantities is introduced. Besides values of constrained signals it contains also their functions to be able to express the dynamic behavior needed for increment constraints.

The c_t is extracted from data $d(T)$ by the mapping

$$d(t) \mapsto c_t \in c_t^*. \quad (30)$$

Let for i-th element of c_t, where $i = 1, \ldots, \mathring{c}$ with $\mathring{c}$ as dimension of c, an interval C_i be defined

$$[c_i^{\min}, c_i^{\max}] = C_i \subset c_{i;t}^* \quad (31)$$

The constraints are defined by vector function Z_c

$$c(T) \mapsto Z_c \in R^{\mathring{c}}$$

being non-positive when the constraints are met, see (4).

In most practical tasks, vector c_t contains magnitudes and differences of data

$$c_t = [d_t, d_t - d_{t-1}].$$

Two variant of function Z_c for servo control Z_{c_M} and noise compensation Z_{c_P} tasks are used.

6.3 *Servo Control Task*

The constraint function Z_c collects information about maximal constraints violation during the simulation run

$$Z_{c_M,i}^T = \max_{t=1,\ldots,T} \mathrm{dist}(c_{i,t}, C_i) - \mathrm{dist}(c_{i,t}, \mathrm{comp}(C_i)),$$

where $\mathrm{comp}(C_i)$ is a set complement of C_i, $Z_{c_M,i}^T$ is i-th element of $Z_{c_M}^T$, and $\mathrm{dist}(.,.)$ denotes a set distance. This definition of function is suitable mainly for transient processes, where constrained signals have one or just few important peaks, such as servo control tasks, where T is big enough to cover all instants with significant signal change.

6.4 *Noise Compensation Task*

The second function evaluates a proportional amount of time where constraints are satisfied to the total length of simulation. In the discrete case, it is the relative frequency of constraints satisfaction

$$Z_{c_P,i}^T = \alpha_{\min} - \frac{1}{T}\sum_{t=1}^{T} \chi_{C_i}(c_{i,t}), \quad (32)$$

where χ_{C_i} is characteristic function of the set C_i, and number $\alpha_{\min} \in [0, 1]$ relaxes the requirement of constraint satisfaction to a specified level. This definition is suitable for situations where the constraints can be violated any time during the simulation. This is the case of noise compensation control.

In the case of control loop generates a stationary process, it holds

$$Z_{c_P,i}^T \xrightarrow{T\to\infty} \alpha_{\min} - \mathbf{P}(c_i \in C_i) \quad \text{almost surely},$$

where $\mathbf{P}$ denotes probability.

7. NUMERICAL EVALUATION

7.1 *Expected Value Estimation*

The controller tuning, formulated as the optimization task (6), acts on the conditional expectation of losses Z_c and Z_o. However, their pdf is not known in a closed form, because of general complexity of the dynamic system model (1) and adaptive controller (2) behavior, and can be only sampled. Thus the expected value has to be estimated using the sampling. To unify notation, let $Z_\bullet$ denotes both loss functions depending on the content of placeholder $\bullet \in \{c_M, c_P, o\}$. The expectation $\mathcal{E}\{Z_\bullet|q\}$ is estimated as sample mean

$$Z_\bullet^{N,T}(q) = \frac{1}{N}\sum_{s=1}^{N} Z_{\bullet,s}^{T}(q) \xrightarrow{N\to\infty} \mathcal{E}\{Z_\bullet|q\}, \quad (33)$$

Sequence $\{Z_{\bullet,s}^{T}(q)\}_{s=1}^{N}$ denotes N samples of random variable $Z_\bullet(q)$.

7.2 *Number and Length of Simulations*

Random variable $Z_\bullet^{N,T}$ is evaluated using N independent simulation runs. Length of each run is determined by T. These two values have big influence on the computational complexity of the loss function evaluation. The number of independent simulation runs N is indirectly proportional to variance of loss function $Z_\bullet^{N,T}$.

The similar conflicts occurs for the length of simulation T, which must be long enough in order to:

(1) Contain all important reference trajectory changes
(2) Allow uncertain parameters vary to simulate controller adaptiveness
(3) Lower variance of the controller quality loss functions

The item 1 is straightforward. It is used for transient processes, where kind of constraints measure Z_{c_M} is used. Of course, also all responses of the reference trajectory changes must be included.

The situation of items 2 and 3 is more complicated. Both of the items contribute to the precision of expected value estimate. Even more the item 2 can be substituted by item 3, because if the variance is low, it means that further parameters changes brings no more information to the controller quality loss functions.

Increasing simulation length T for stationary case has the same effect as the increasing number N of the simulations. Thus, one enough long simulation is sufficient.

The proper values of N and T are decided online using probability convergency or according to stabilization of particular pdfs using Kullback-Leibler divergence. The on-line stopping is advantageous in comparison with off-line determination of length and number of simulations, because it considers the contribution of actual data and stopping is optimal for current simulation unlike for all possible simulation runs as in case of off-line stopping.

7.3 *On-line Stopping Rule for Number of Simulations*

The independent simulation runs are connected with non-stationary servo-control tasks. It is hard to find a reasonable distribution for different variants of reference trajectory. Thus, a simple non-parametric stopping rule based on second moment is used in the form of

$$\mathbf{P}(|Z_\bullet^{N,T} - \mathcal{E}Z_\bullet^{N,T}| \geq \gamma) \leq \beta. \quad (34)$$

The independency of averaged loss functions resulting to $Z_\bullet^{N,T}$ used with Chebyshev inequality yields

$$\mathbf{P}(|Z_\bullet^{N,T} - \mathcal{E}Z_\bullet^{N,T}| \geq \gamma) \leq \frac{\mathrm{var}(Z_\bullet^{N,T})}{N\gamma^2}. \quad (35)$$

Because covariance $\mathrm{var}(Z_\bullet^{N,T})$ is unknown, its estimate $Z_{\sigma,\bullet}^{N,T}$ is used

$$Z_{\sigma,\bullet}^{N,T} = \sum_{s=1}^{N} \frac{(Z_{\bullet,s}^{T})^2 - (Z_\bullet^{N,T})^2}{N}, \quad (36)$$

where variable $Z_{\bullet,s}^{T}$ has the same meaning as in (33). Then, stopping is triggered after certain minimal number of simulation is performed and when the following inequality is satisfied

$$\frac{Z_{\sigma,\bullet}^{N,T}}{N\gamma^2} \leq \beta. \quad (37)$$

Typical values for stopping parameters are $\beta = 0.1$ and $\gamma = 0.1$.

7.4 *On-line Stopping Rule for Simulation Length*

The rule for on-line simulation stopping in stationary cases is more complicated. The function $Z_\bullet^{T}$ contain a sum, but the summed terms, or in another words partial loss functions, in (29) and (32) are correlated.

To find reasonable stopping rule, a new quantity v_t related to partial loss function is introduced. A simple dynamic model of v_t is being estimated in Bayesian way. Let the parameters of the model be denoted by Ξ.

$$f(v_{t+1}|v(t), \Xi) \quad (38)$$

When the estimated pdf $f(\Xi|v(t))$ of model parameters Ξ stabilizes, the stopping takes place. The stabilization of pdfs is measured by Kullback-Leibler divergence $\mathcal{D}_{\mathrm{KL}}$ of two successive pdf estimates Kárný et al. (2005). It is defined by

$$\mathcal{D}_{\mathrm{KL}}(f(\Xi_\bullet|d(T))\|f(\Xi_\bullet|d(T-1))) = \tag{39}$$
$$= \int f(\Xi_\bullet|d(T)) \ln \frac{f(\Xi_\bullet|d(T))}{f(\Xi_\bullet|d(T-1))} d\Xi_\bullet.$$

When this divergence, called Q_T, becomes smaller than some threshold value ε

$$Q_T = \mathcal{D}_{\mathrm{KL}}(f(\Xi_\bullet|d(T))\|f(\Xi_\bullet|d(T-1))) \leq \varepsilon,$$

the computation is stopped. At this moment, the pdf is considered to reach steady state. The stationarity means that more data would not bring more information for the estimate.

It is assumed that $f(Z^T)$ is stabilizing as $f(\Xi|T)$ is stabilizing. In another words the divergence $\mathcal{D}_{\mathrm{KL}}(f(Z^{T+1})\|f(Z^T))$ is decreasing as $\mathcal{D}_{\mathrm{KL}}(f(\Xi_\bullet|d(T))\|f(\Xi_\bullet|d(T-1)))$ is decreasing.

The definition of the quantity v_t and the construction of models for the functions Z_o and Z_{c_P} is described in the following sections 7.4.1 and 7.4.2. The stopping rule for whole simulation is triggered when the conditions for both loss and constraints function stopping are activated.

7.4.1. *Approximation by Logarithm ARX Model* The model (38) used for stopping uses partial loss v_t of function Z_o, which is defined as a distance between the controlled variable y_t and its referential value y_t^{ref} in time t

$$v_t = \|y_t - y_t^{\mathrm{ref}}\|. \tag{40}$$

It was shown Kárný et al. (1990), that tracking error of adaptive controller, controlling system with uncertain parameters, closed loop is a random variable with distribution close to log-normal one. Accepting this kind of stationary distribution, a suitable form of dynamic model (38) seems to be a simple autonomous ARX model acting on logarithm of the partial losses v_t

$$\ln(v_t) = a\ln(v_{t-1}) + k + e_t, \quad e_t \sim \mathcal{N}(0, R). \tag{41}$$

The parameters a, k, and R are collected into parameters variable Ξ.

The Bayesian identification of the parameters Ξ leads to self reproducing Gauss-inverse-Wishart prior/posterior pdf

$$f(\Xi|v(t)) = f(\Xi_\theta, \Xi_R|V_t, \nu_t) = \tag{42}$$
$$= \alpha_t |R|^{-\frac{\nu_t}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\Xi_R^{-1}\begin{bmatrix}-I\\ \Xi_\theta\end{bmatrix}' V_t \begin{bmatrix}-I\\ \Xi_\theta\end{bmatrix}\right)\right\},$$

where α_t is normalizing constant. Statistics ν and V and parameter elements Ξ_θ and Ξ_R without Ξ are described in section 4, with the difference that now the data are logarithmed.

The stationarity measure Q_t, in another words KLD of two successive estimated pdfs of Ξ, explored in Kárný et al. (2005), has form

$$Q_t = \mathcal{D}_{\mathrm{KL}}(f(\Xi|d(t))\|f(\Xi|d(t-1))) =$$
$$= \frac{F(\nu_t) + G(\zeta_t) + H(\nu_t, \varrho_t, \zeta_t)}{2}, \tag{43}$$

where

$$F(\nu_t) = 2\ln(\Gamma(\frac{\nu_t - 1}{2})) - 2\ln(\Gamma(\frac{\nu_t}{2})) + \frac{\partial \ln(\Gamma(\frac{\nu_t}{2}))}{\partial \frac{\nu_t}{2}}$$
$$G(\zeta_t) = \ln(1+\zeta_t) - \frac{\zeta_t}{1+\zeta_t}$$
$$\varrho_t = \frac{\hat{e}_t^2}{D_{t-1}^y(1+\zeta_t)}$$
$$H(\nu_t, \varrho_t, \zeta_t) = (\nu_t - 1)\ln(1+\varrho_t) - \frac{\nu_t \varrho_t}{(1+\varrho_t)(1+\zeta_t)}.$$

When the divergence Q_T is less than threshold ε in time T, than it is assumed that enough information has been collected and the loss function $Z_o(T)$ (29) is precise enough.

7.4.2. *Markov Chain Estimation* The calculation of sum in (32) gives a relative frequency of constraints satisfaction. To estimate precision of this estimate, this task is slightly extended.

Given constraints quantity $c_{i;t}$ from (30) and corresponding constraining interval C_i from (31), let $\{v_t\}_t$ is sequence defining relative position of $c_{i;t}$ to C_i

$$v_{i;t} = \begin{cases} 1 & c_{i;t} > C_i \\ 0 & c_{i;t} \in C_i \\ -1 & c_{i;t} < C_i \end{cases}, \tag{44}$$

where inequality symbol is taken as it holds for all the elements of set on its right side.

The dynamic model (38) of sequence $\{v_{i;t}\}_t$ is now modeled by Markov chain

$$f(v_{i;t}|g_{i;t-1}, \Xi) = \Xi_{v_i|g_i}. \tag{45}$$

Because quantity v_t is now discrete, the symbol f represents probability now. Quantity $g_{i;t-1}$ holds the past values of $v_{i;t}$

$$g_{i;t-1} = [v_{i;t-1}, v_{i;t-2}, \ldots, v_{i;t-\eta}].$$

Number η denotes the order of Markov chain. The parameter $\Xi_{v|g}$ has $3^{\eta+1}$ entries. The signal element index i will be omitted to simplify the following text.

Using Bayesian rule and prior on $f(\Xi)$ defined by statistic $V_{0,v|g}$

$$f(\Xi) \propto \prod_g \prod_v \Xi_{v|g}^{V_{0,v|g}-1},$$

we obtain the posterior pdf of parameters Ξ

$$f(\Xi|v(t)) = \frac{\prod_g \prod_v \Xi_{v|g}^{V_{v|g;t}-1}}{B(V_t)},$$

where

$$V_{v|g;t} = V_{0,v|g} + \sum_{\tau=1}^{t} \delta(v, v_t)\delta(g, g_t)$$

and the normalizing factor

$$B(V_t) = \prod_g \frac{\prod_v \Gamma(V_{v|g;t})}{\Gamma(\sum_v V_{v|g;t})}.$$

The stopping rule uses Kullback-Leibler divergence to determine that enough information of the constraint satisfaction statistic Z_c^P was collected. The calculation is stopped whenever the divergence of two succesive pdfs is less or equal to a threshold ε

$$Q_T = \mathcal{D}_{\mathrm{KL}}(f(\Xi|v(T)) \| f(\Xi|v(T-1))) \leq \varepsilon. \quad (46)$$

Derivation of this divergence for Markov chain model is done through converting it to Dirichlet model, for which the divergence is analyzed in Kárný et al. (2005).

Parameters $\Xi_{v|g}$ are independent for different past data g. Thus

$$f(\Xi|v(t)) = \prod_g f(\Xi_{\bullet|g}|v(t)),$$

where the particular factors

$$f(\Xi_{\bullet|g}|v(t)) = \frac{\Gamma(\sum_v V_{v|g;t})}{\prod_v \Gamma(V_{v|g;t})} \prod_v \Xi_{v|g}^{V_{v|g;t}-1}$$

are distributed by Dirichlet distribution. In each time step, only one of these factor is updated—that one with corresponding past data $g = g_{t-1}$. The other factors remain unchanged.

It holds

$$\mathcal{D}_{\mathrm{KL}}(f(a)f(c) \| f(b)f(c)) = \mathcal{D}_{\mathrm{KL}}(f(a) \| f(b)).$$

Thus

$$\mathcal{D}_{\mathrm{KL}}(f(\Xi|v(t)) \| f(\Xi|v(t-1))) = \quad (47)$$
$$= \mathcal{D}_{\mathrm{KL}}(f(\Xi_{\bullet|g_t}|v(t)) \| f(\Xi_{\bullet|g_t}|v(t-1)))$$

is a divergence of two Dirichlet distributions.

Now, the results from Kárný et al. (2005) can be substituted to the stopping rule (46) which yields

$$Q_t = -\ln \frac{V_{v_t|g_t;t-1}}{\sum_v V_{v|g_t;t-1}} +$$
$$+ \frac{\partial}{\partial V_{v_t|g_t;t}} \ln \Gamma(V_{v_t|g_t;t}) - \frac{\partial}{\partial \sum_v V_{v|g_t;t}} \ln \Gamma(\sum_v V_{v|g_t;t}).$$

At the stopping time T, a stabilized MC model is obtained. Its steady state probability of state number zero is the wanted value for calculation of function $Z_{c_P}^T$. Nevertheless, instead of it the function $Z_{c_P}^T$ is calculated in standard way using its definition 32. Both values are asymptotically same.

7.4.3. *Properties of stationarity measures* To show properties of stationarity measures Q using both ARX and MC stopping models, a simple illustrative experiment is presented. The results can be seen in figure 2. The data simulating the

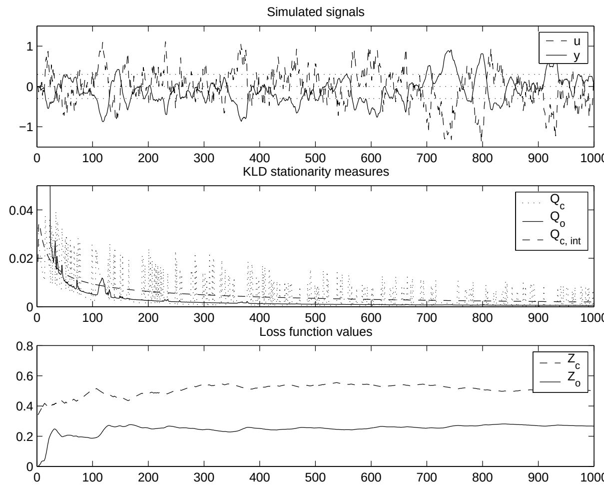

Fig. 2. Properties of stationarity measures.

partial loss functions Z_o were generated using a simple linear system with transfer function

$$\frac{0.00468 + 0.00438s}{1 - 1.81s + 0.8178s^2}$$

model driven by zero mean white noise with variance one. The squares of the generated outputs was used as partial loss function v_t for stopping using ARX model stabilization. The stationarity measure $Q_{o;t}$ for ARX model is seen in the second part of the figure and evolution of mean value estimation is in the third part of figure.

Now, a interval $[-0.3, 0.3]$ is used on the generated data to obtain the discrete three-state signal (44) for purpose of stopping through MC model. The resulting stationarity measure $Q_{c_P;t}$ and corresponding estimation of probability of state zero are shown in second and third part of the picture.

It can be seen that measure $Q_{c_P;t}$ is rather fuzzy. This complicates the decision whether to stop simulation, because the rule to stop whenever the measure is below the threshold is quite unsatisfactory as several next samples immediately increase the value above the threshold. To solve this problem, an interpolation is performed using regression with the following model

$$Q_t = a_0 + a_1 t^{-1/2} + a_2 t^{-1}. \quad (48)$$

The interpolated measure, denoted by $\tilde{Q}_{c_P;t}$, is shown in the figure. The interpolation is, up to a tiny peak close to origin, satisfactory for stopping purposes.

It is possible to think about stopping for the interpolating regression as well and trigger the stopping when the interpolated measure is below the threshold as well as the interpolation itself has been stabilized.

The threshold for the measures $\tilde{Q}_{c_P;t}$ and $Q_{o;t}$ need not be a same value. The stopping models are different and has different number of identified parameters.

8. OPTIMIZATION

In general case the pdf of data $f(d(T))$ is available only through samples. Thus the same holds for functions $Z_\bullet$. Therefore the optimization problem (6) forms a stochastic optimization task. Sample path method is used to approximate it by a deterministic optimization task.

Let for a function $h_\bullet : q^* \times \mathbf{R}^{\mathring{\xi}} \mapsto \mathbf{R}^{\mathring{Z}_\bullet}$ and random vector ξ holds

$$Z_\bullet(q) \equiv h_\bullet(q, \xi).$$

Let the expected value $\mathcal{E}Z_\bullet(q)$ is approximated with $\hat{Z}_\bullet(q)$

$$\hat{Z}_\bullet(q) = \frac{1}{N} \sum h_\bullet(q, \xi_i),$$

where N is a positive integer and $\{\xi_i\}_{i=1}^N$ is a sequence, called sample path, of independent samples of ξ. Fixing this sequence at constant samples, the optimization becomes deterministic.

In general, it is possible to obtain the selection of function $h_\bullet$ and distribution of ξ by infinitesimal perturbation analysis (IPA) or by likelihood ratio method Glasserman (1991). In case of Gaussian ARX model where $y_t \sim \mathcal{N}(\theta\psi_t, R)$ it was selected $\xi_{i,t} \sim \mathcal{N}(0, I)$ and $y_t = \theta\psi_t + R^{\frac{1}{2}}\xi$, where θ and R are samples from (15) and $R^{\frac{1}{2}\prime} R^{\frac{1}{2}} = R$.

As the deterministic optimization method, the quasi-Newtonian BFGS method for constrained optimization *fmincon* from the Matlab Optimization Toolbox Matlab Inc. (2001) was used to solve the deterministic task. The method cited is able to approximate the gradient vector and Hessian matrix internally.

9. CONCLUSION

A method for controller tuning was presented. It starts from the general system model and controller description and its Monte-Carlo simulation. Constraints and objective functions are used to define the ideal controller behavior, which is searched by optimization using sample path method.

The control loop simulation is described firstly rather general and then a particular case of Gaussian ARX model and adaptive LQG controller is given. The definition of constraint includes the input limitations as well as output overshoot.

The speed of the computation can be significantly improved by introducing stopping rules to lower computation demands of simulation and optimization loop.

This method is being developed with the Designer Toolbox project Novák et al. (2003); Böhm et al. (1998), which includes other tasks such as system identification and controller verification to support the design.

Acknowledgment

This work was partially supported by grants GA ČR No. 102/05/0271, GA ČR No. 102/03/0049, and AV ČR S1075351.

References

J. Böhm, P. Nedoma, and J. Rojíček. Algorithms, software and experience with automated controller design DESIGNER 2000. In J. Rojíček, M. Kárný, and K. Warwick, editors, *IFAC Workshop CMP'98*, pages 285–292. Prague, 1998.

P. Glasserman. *Gradient estimation via perturbation analysis.* Boston, Kluwer, 1991.

M. Kárný. Parametrization of multi-output multi-input autoregressive-regressive models for self-tuning control. *Kybernetika*, 28(5):402–412, 1992.

M. Kárný, T. Jeníček, and W. Ottenheimer. Contribution to prior tuning of LQG selftuners. *Kybernetika*, 26(2):107–121, 1990.

M. Kárný, J. Kracík, I. Nagy, and P. Nedoma. When Has Estimation Reached A Steady State? The Bayesian Sequential Test. *International Journal of Adaptive Control and Signal Processing*, 19(1):41–60, 2005.

The Matlab Inc. *Optimization Toolbox for Matlab*, volume MA 01760-2098. Natick, 2001.

M. Novák, J. Böhm, P. Nedoma, and L. Tesař. Adaptive LQG controller tuning. *IEE Proceedings - Control Theory and Applications*, 150(6): 655–665, 2003.

V. Peterka. Bayesian system identification. In P. Eykhoff, editor, *Trends and Progress in System Identification*, pages 239–304. Pergamon Press, Oxford, 1981.