TUNING OF AN ADAPTIVE LQG CONTROLLER

Miroslav Novák

Department of Adaptive Systems
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, CZECH REPUBLIC
E-mail: mira@utia.cas.cz

Abstract: The proper work of model based controller such as LQG or GPC one depends on its right setting expressed by so called tuning knobs. It is often difficult to set these knobs well, mainly for non–experts, especially in MIMO case where the number of knobs rapidly increases. Accurate tuning is hardly possible using trial and error method. A sophisticated algorithm has to be developed for successful use of these controllers. The classical methods for controller settings can not be precise enough for this kind of systems and it is expected the controller function can be improved considerably if they are better tuned.

Keywords: Adaptive Control, LQG Controller, Optimization

1 INTRODUCTION

The LQG controller design depends on several penalization variables called tuning knobs. It is difficult to guess these variables directly, because they do not correspond to the user's requirements placed on the controller. The user usually defines the requirements as constraints placed on the signals on the controlled system. The proper tuning knobs setting fits the controller into the constraints. The constraints can be placed on system input value, system input differences, overshoot or the time to reach a desired state. For the task of tuning it is required to find a dependency of the controller quality on the tuning knobs setting e.g. to calculate the system input range when particular values of the tuning knobs are set.

The tuning, as a minimization of system constraints violation, is an optimization problem. Only the samples of the controller quality—objective function can be evaluated, no gradients are available. For each sample a simulation must be performed.

The tuning would be easily solvable by standard optimization routines but the stochastic nature of the simulation results moves the problem to a stochastic optimization.

The task of LQG controller quality and its tuning was concerned in (Rojíček, 1998). Tuning was designed only for SISO system where simple optimization methods were sufficient.

2 TUNING ALGORITHM

Tuning knobs are represented by parameters Q. They express typically penalization weights. For example in case of LQG controller the tuning knobs are coefficients in its quadratic criterion, but user requirements express usually some constraints of system quantities. For example value of an actuator input must not exceed some bounding

interval, there is a maximum value of output overshoot or maximum input difference is limited.

For each value of a tuning knobs Q we can evaluate an error in sense of user requirements by creating a controller using given tuning knob values. Then a simulation of the closed loop is performed. From results of the simulation, simulated data D, we can say how much the requirements were violated. This violation is measured by a loss function $Z(D)$. Aim of the tuning is to make the loss function zero or at least minimal to satisfy the requirements exactly or as much as possible. The proper choice of the loss function is also important and will be discussed in this paper.

2.1 MODEL DESCRIPTION

Simulated model is a linear one, which can be completely estimated using available Bayesian tools. The model is described using an outer model probability density function (pdf) $f(y|u, \varphi, \Theta)$ where y is process output, u process input, φ is regressor vector and Θ describes model parameters. Quantity Θ is a random variable and its probability density function $f(\Theta)$ is known as result of a previous identification process.

2.2 Loss function distribution

Data D composed of the system input—output quantities recorded during simulation depends on tuning knobs settings Q and on model parameters pdf $f_\Theta = f(\Theta)$. Pdf of the data can be written as

$$f(D|Q, f_\Theta) = \int f(D|Q, \Theta) f(\Theta).$$

Measure of the user requirement violation is a loss function $Z(D)$ depending on data. Distribution $f(Z|Q, f_\Theta)$ of the loss function can be formally found by transforming through the mapping $D \to Z$.

Our goal is to find best tuning knob setting. In the ideal situation there is a distribution of tuning knobs $f(Q|Z, f_\Theta)$ conditioned by loss function Z and distribution f_Θ, which would assign optimal tuning knob values to given loss function definition. In real situation it is hardly possible to compute such pdf directly, because the relationship between the loss function and tuning knobs is usually nonlinear and it is not possible to find it analytically. So that, optimization of its characteristics is quite involved.

2.3 ADAPTIVE CONTROL SIMULATION ALGORITHM

Tuning will be performed by using an optimization technique. For the optimization we need to get a real number instead of pdf. A function which maps the pdf to a real number has to be used. It can be the expected value approximated as a sample mean. Other advisable choice is such value, for which the user requirements satisfied are with given probability.

A1. Optimization algorithm

1. Let is available the initial information such as the identified data model pdf $f(y|\psi)$, user requirements, loss function Z, system parameter estimate $\hat{\Theta}$ for adaptation, etc.
2. Set initial guess of the tuning knobs Q.

: optimization step :

A2. Adaptive simulation loop

(a) Set $t = 1$, initialize simulation, Z quantity and controller model estimate $\hat{\Theta}$.

: simulation step :

(b) Sample data $y_t \sim f(y_t|\psi_t)$

(c) Update the loss Z. Exit the simulation loop A2, when the estimation of the loss Z is representative.

(d) Update the estimate of model $\hat{\Theta}$ using last sampled data

(e) Design controller K based on tuning knobs Q and model $\hat{\Theta}$

(f) Set $t = t + 1$, generate new system input u_t using the controller K and continue with next simulation step

3. Exit the optimization algorithm with Q as a result, when $Z(Q)$ satisfies optimality conditions.

4. Select a new guess of Q and continue with optimization step.

The simulation loop is adaptive which means that the data are sampled from model description including model parameters distribution $f(\Theta)$. Output quantity is sampled from pdf $f(y|\psi) = \int_\Theta f(y|\psi, \Theta) f(\Theta)$. The controller is adapting to sampled data and thus an adaptive control loop is simulated.

Simulation loop can contain a sequential stopping rule for deciding on Z representativeness to reduce computational demands.

2.4 Optimization

An optimization method generating the guesses of Q for tuning algorithm has to be chosen. It is possible to use deterministic quasi–Newtonian methods with a bit artificial modification of simulation loop to reduce randomness. Other suitable methods are nondeterministic optimization methods such as stochastic approximations or evolution algorithms. Nondeterministic methods are expected to provide more objective results, because a real noisy environment can be used in the simulation.

3 CONTROLLER TUNING

The only controller used to illustrate the tuning is LQG controller. However methods used for tuning are general and extension to other types of control should be easy.

LQG controller is designed to minimize a quadratic criterion which penalizes deviations from desired state. Quadratic criterion J can be defined

$$J_t = \sum_{\tau=t}^{t+T} (q_1 l_{1;\tau}^2 + q_2 l_{2;\tau}^2 + \ldots + q_{n_q} l_{n_q;\tau}^2)$$

The T is optimization horizon, weights $q_\bullet$ are nonnegative real numbers and real linear functions $l_{\bullet;\tau}$ depends on input and output quantities y_t and u_t, including their past given by a regression vector. They measure particular deviations from a desired state. The measures l_t^2 called penalizations can be of different kind. Setting of weights $q_\bullet$ is up to designer of the LQG controller. This weights are the tuning knobs which we will try to tune automatically.

Some typical forms of the quadratic criterion are described below.

3.1 Reference tracking

The simplest design of the LQG controller, in sense of its criterion, is to keep the output on its desired set point y^{ref} while the input should be close to its reference value u^{ref}. Penalizations are then chosen to penalize the tracking error

$$l_{1;t}^2 = (y_t - y^{\mathrm{ref}})^2 \tag{1}$$

and to penalize difference of the input from its reference value

$$l_{2;t}^2 = (u_t - u^{\mathrm{ref}})^2 \tag{2}$$

Penalization weights q_1 and q_2 have to be set to represent optimal trade off between tracking error and actuator effort.

3.2 Limited increments

Another useful penalization is to limit increments of input as, for example, when the actuator is not able to open valve arbitrarily fast. The appropriate penalization is

$$l_t^2 = (u_t - u_{t-1})^2 \tag{3}$$

The difference penalization can be extended to approximate higher order derivatives in a similar way.

4 USER REQUIREMENTS AND LOSS FUNCTION

User requirements can be generally expressed using the loss function which should be minimized or should be below some critical value for optimal controller tuning. The first case is useful for minimizing the output error value for example. The second case is used to keep the constraints on system quantities, for instance limited input values etc.

4.1 Constraints placed on system quantities

Constraints placed on system quantities stem from physical system properties, like maximum and minimum value of the quantities, or they they reflect the user idea of good system behavior.

Constraints can be placed on all quantities of the system like input u_t, output y_t or state ϕ_{t-1}. And also on functions of these quantities like input increments $\Delta u_t = u_t - u_{t-1}$, which is a function of the u_t and ϕ_{t-1}.

Constraints are expressed as a set $\mathcal{C}$ of allowed values of selected quantities. The $\mathcal{C}$ is subset of cartesian product of all domains of constrained quantities.

For example, constraints are placed on system input and its increments and the system has two dimensional input. Constrained system quantities, and their functions, are $u_{1;t}$, $u_{2;t}$, $\Delta u_{1;t}$ and $\Delta u_{2;t}$. Let all these quantities be real numbers. Then the constraint set $\mathcal{C}$ is subset of space R^4

$$\mathcal{C} \subset R^4$$

Elements of Euclidian space R^4 will be denoted by c, and they have following meaning

$$c = (u_{1;t}, u_{2;t}, \Delta u_{1;t}, \Delta u_{2;t}) \in R^4 \tag{4}$$

Let the set of the mutually depended constraints $\mathcal{C}_u$ be given on inputs

$$(u_1, u_2) \in \mathcal{C}_u \subset R^2$$

Dependent constraints means the bounding set of one element of input u_i depends on the value of other element of input u_j. An example of dependent constraint is when $\mathcal{C}_u$ is a circle.

4.2 Loss function

Let the loss function be defined as a probability of the constrained quantities vector c being outside the set $\mathcal{C}$.

The pdf of considered quantities is often Gaussian and therefore it is not possible to satisfy the constraint exactly, because such optimum lies in infinity of the penalization domain.

Let $p_{\mathcal{C}}$ be a sufficient probability of a constraint $\mathcal{C}$ satisfaction, given by user. When the constraints are satisfied with equal or higher probability, we say they are satisfied in probabilistic sense.

It is useful to substract the sufficient probability $p_{\mathcal{C}}$ from the loss function Z measuring the probability of constraints violation. Then is the loss zero when the constraints are just sufficiently satisfied.

The criterion is then defined

$$Z = \mathcal{P}(c \notin \mathcal{C}) - p_{\mathcal{C}} = 1 - \int_{c \in \mathcal{C}} f(c)dc - p_{\mathcal{C}} \tag{5}$$

where $f(c)$ is marginal pdf of the constrained quantities. Optimization have following form

$$Q_o = \mathrm{Arg} \min_{Q \in Q^*} (Z(Q))^+ \tag{6}$$

where $(\bullet)^+ \equiv \max(0, \bullet)$. The optimum need not be unique and therefore Q_o is generally a set.

If value of the minimum of Z is positive, the constraints are not satisfied in probability sense and the optimum is just its best approximations. If it is known that it is possible to satisfy the constraints, there exists a penalization with a non–positive loss and the optimum set Q_o can be defined simply by

$$Q_o = \{Q \in Q^* : Z(Q) \leq 0\} \tag{7}$$

The loss function Z depends on the controller tuning variables and on the stochastic noise realization. The loss is approximated using the simulation

$$Z = \frac{\text{number of steps where } c \notin \mathcal{C}}{\text{total simulation length}} - p_{\mathcal{C}}$$

4.3 Second layer optimality criterion

The optimality criterion based on the set of constraints $\mathcal{C}$ gives a set Q_o of the optimal tuning values. It is not easy to find the whole set, especially when using simulation. The solution of this problem is to define other loss function, say Z_2, which will choose the best, and if possible unique, tuning knob value Q_{o2} from the set Q_o. The optimal tuning knob value Q_{o2} is then

$$Q_{o2} = \mathrm{Arg} \min_{Q \in Q_o} Z_2(Q) \tag{8}$$

The simplest useful example of this second layer loss function is the mean tracking error

$$Z_2 = \mathcal{E} \| y - y^{\mathrm{ref}} \|^2$$

as it reflects the original control aim.

The Q_{o2} is subset of the Q_o. Even if it contains more elements then one, the "size" of the Q_{o2} set is reduced and gives the best choice according to criterion Z_2. This second layer loss function Z_{o2} is easy to evaluate once we have the simulated data available.

5 TUNING AS AN OPTIMIZATION

Tuning aiming at the best tuning knobs values is an optimization problem. It is a constrained optimization where the objective function is the second layer criterion (8) and the user requirement constraints (6) placed on the system quantities are mapped to the constraints of the optimization problem.

The loss function is evaluated using Monte Carlo method, therefore a stochastic optimization method is used.

5.1 Stochastic approximations

Stochastic approximations as an optimization method fit well the considered tuning problem without the problems which arises from use of the deterministic optimization.

The stochastic optimization task is

$$\text{minimize} \;\; \mathcal{E} Z(Q) \;\; \text{where} \;\; Q \in Q^* \tag{9}$$

The stochastic approximations uses the steepest descent direction method and the step size is the norm of approximated gradient $\hat{g}_k$ multiplied by a gain a_k which is decreasing function of the iteration count k. The iteration step is

$$Q_{k+1} = Q_k - a_k \hat{g}_k(Q_k) \tag{10}$$

The gradient $\hat{g}_k(Q_k)$ is approximated using finite differences

$$\hat{g}_{ki}(Q_k) = \frac{\mathcal{Z}(Q_k + c_k e_i) - \mathcal{Z}(Q_k - c_k e_i)}{2c_k} \tag{11}$$

where e_i denotes a unit vector in the i-th direction and gain c_k decreases with increasing iteration count.

The convergence rate of the stochastic approximations method is slow compared to the deterministic algorithms. There is a modification to the classical stochastic optimization method called Simultaneous Perturbation Stochastic Approximations (SPSA) which approximates the gradient using just two loss function evaluations, regardless the dimension of the problem.

$$\hat{g}_k(Q_k) = \frac{\mathcal{Z}(Q_k + c_k d_k) - \mathcal{Z}(Q_k - c_k d_k)}{2c_k d_k} \tag{12}$$

where d_k is a random direction perturbation vector, see (Spall, 1999). The SPSA method is fast enough for solving our problem.

Precision of the approximated optimum is not too high, but for purpose of tuning controller is sufficient even quite rude approximation.

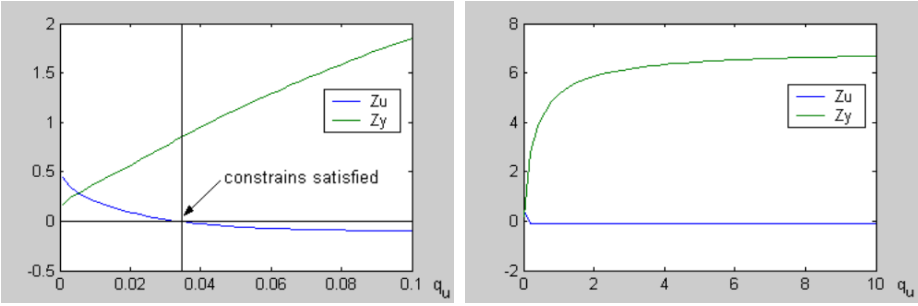

Fig. 1: Typical shape of the loss functions

Gains The gain sequences a_k and c_k in equations (9), (11) and (12) are chosen according to (Spall, 1998) as

$$a_k = a/(A+k+1)^{\alpha} \tag{13}$$
$$c_k = c/(k+1)^{\gamma} \tag{14}$$

The universal recommended values for exponents are $\alpha = 0.602$ and $\gamma = 0.101$, see (Spall, 1998). The choice of the constants a, A and c depends on the mean gradient size, desired approximate iteration step size and variance of the loss function.

There was used a modification for the problem of controller tuning. The optimization algorithm is similar to (10), but the step size does not depend on norm of the gradient, but just on the gain sequence a_k. The optimization step is

$$Q_{k+1} = Q_k - a_k \frac{\hat{g}_k(Q_k)}{\|\hat{g}_k(Q_k)\|} \tag{15}$$

This seems to be necessary, because of the shape of the loss function defined for controller tuning. It changes its gradient size significantly around the solution and therefore it is better to do not depend on its size.

5.2 Loss function transformation

The convergence rate of stochastic approximations method depends on the gradient properties of loss function. A typical shape of loss functions embodies a large gradient size close to the origin and it is nearly flat in regions far from the origin. The example of loss functions for SISO LQG controller with the quadratic criterion

$$J = y^2 + q_u u^2$$

is shown on figure 1.

The large differences in the slope of the both loss functions makes problems for the stochastic approximations. It is known that these methods converges slowly in regions with small gradient and they can diverge in regions with high gradient. Both of these situations can happen in this case. When the iteration of the optimization algorithm reaches the point with high gradient, it makes a big step far from the origin, where the gradient is very small. The step size is then negligible and the optimization stops there.

The solution to this problem was set the step size independent on the gradient size. Only direction of the gradient is used. Step size is decreasing according to the gain sequence a_k of the optimization method.

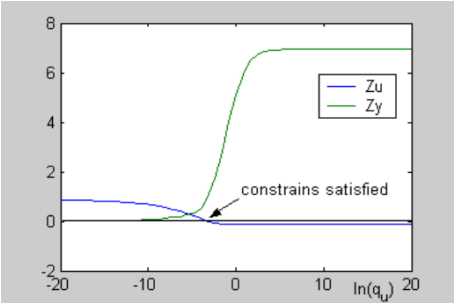

Fig. 2: The loss functions in logarithmic scale

The constant step makes problems near the axis origin, where it can be too big, to be able to find the solution. To solve this problem the optimization is performed in the logarithmic scale, where the loss functions are more regular near the origin, see figure 2.

6 CONCLUSIONS

Proposed methods of the controller tuning follow up with the previous work (Rojíček, 1998). The main extension of this approach is tuning of MIMO controller. Other improvements of the tuning algorithm are in newly proposed tuning loops configuration and use of the non–deterministic methods for the optimization. It admits, moreover, to make tuning of a truly adaptive controller.

The whole process of Bayesian system identification, simulation and control design is implemented in Matlab toolbox Designer 2000 described in (Bůcha *et al.*, 1998). This toolbox will be used for many tasks in tuning algorithms and results of this work will be in future also included in new version of Designer 2000.

ACKNOWLEDGMENT

This work was supported by CZ-SLO grant KONTAKT 2001/020 and grant GA ČR No. 102/02/0204.

REFERENCES

Bůcha, J., Kárný, M., Nedoma, P., Böhm, J. and Rojíček, J. (1998), Overview of DESIGNER 2000 Project, *in* S. Krejčí, B. Jakeš, J. Macháček and I. Taufer, eds, 'Proceedings Process Control '98, ŘÍP 1998', Vol. 1, University of Pardubice, Kouty nad Desnou, pp. 65–70. ISBN 80-7194-138-7.

Rojíček, J. (1998), Controller design under constraints and incomplete knowledge of the system, PhD thesis, Czech Technical University, P.O.Box 18, 182 08 Prague, Czech Republic.

Spall, J. C. (1998), 'Implementation of the simultaneous perturbation algorithm for stochastic optimization', *IEEE Transactions on Aerospace and Electronic Systems* **34**, 817–823.

Spall, J. C. (1999), Stochastic optimization and the simultaneous perturbation methods, *in* 'Proceedings of the Winter Simulation Conference, eds. P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans', pp. 101–109.

Wang, I.-J. and Spall, J. C. (1999), A constrained simultaneous perturbation stochastic approximation algorithm based on penalty functions, *in* 'Proceedings of the American Control Conference', pp. 393–399.