Experiments in Multi-parametric Quadratic Programming

Zsuzsa Preitl, Radu-Emil Precup

Dept. of Automation and Applied Inf., “Politehnica” University of Timisoara
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: zsuzsa.preitl@aut.upt.ro, radu.precup@aut.upt.ro

József K. Tar, Márta Takács

Inst. of Intelligent Engineering Systems, Budapest Tech Polytechnical Institution
Bécsi út 96/B, H-1034 Budapest, Hungary
E-mail: tar.jozsef@nik.bmf.hu, takacs.marta@nik.bmf.hu

Abstract: The paper deals with a short presentation of the basic ideas concerning the multi-parametric quadratic programming (mp-QP) problems. The model predictive control is considered also as particular mp-QP problem. Since the solutions to mp-QP problems can be expressed as piecewise affine linear functions of the state, a new implementation in terms of adaptive network-based fuzzy inference systems is proposed. The presentation is focused on the double integrator plant in several settings.

Keywords: Multi-parametric quadratic programming, model predictive control, adaptive network-based fuzzy inference systems

1 Introduction

By multi-parametric programming, a linear or quadratic optimization problem is solved off-line. Even though the multi-parametric approaches rely on off-line computation of a feedback law, the computation can quickly become prohibitive for larger problems. This is not only due to the high complexity of the multi-parametric programs involved, but mainly because of the exponential number of transitions between regions which can occur when a controller is developed in a dynamic programming fashion.

The main method to solve multi-parametric linear programming problems was proposed in [1] and described in [2] where a nice overview on multi-parametric programming is presented. This method is based on constructing the critical regions iteratively, by examining the graph of bases associated to the linear programming tableau of the original problem. Other methods are presented in [3, 4, 5].

The most cited method to solve multi-parametric quadratic (mp-QP) problems was formulated in [6]. The method constructs a critical region in a vicinity of a given parameter using Karush-Kuhn-Tucker conditions for optimality, and then it explores recursively the parameters space outside such regions. The most efficient implementation until now is given in [2], and other methods to solve mp-QP problems are presented in [7, 8, 9].

Model predictive control (MPC) represents the accepted standard for complex constrained multi input-multi output (MIMO) control problems in industrial applications [10]. During each sampling interval, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. During the next sampling interval, the calculation is repeated starting from the new state and over a shifted horizon, leading to a moving horizon strategy. The solution is based on a linear model of the controlled plant, fulfils all input and output constraints, and optimizes a quadratic performance index playing the role of objective function. Thus, the MPC ensures very good control system (CS) performance indices as much as a quadratic performance index in combination with various constraints can be employed to express the desired CS performance objectives. The drawback of MPC is the relatively high on-line computational effort, which limits its applicability to control relatively slow plants.

This paper addresses the process of moving the necessary calculations for the implementation of MPC off-line in the conditions of considering it a special case of mp-QP problem [6].

Convenient solutions to mp-QP problems can be expressed as piecewise affine linear functions of the state, which can be subject to efficient implementations [2, 11, 12]. In case of MPC algorithms solved in terms of mp-QP problems with piecewise affine linear solutions as functions of the state, the implementation problem is relatively complex [10]. One of the aims of the paper is to propose a new implementation of mp-QP solutions in terms of adaptive network-based fuzzy inference systems [13, 14].

This paper is organized as follows. The next Section presents the main aspects concerning the problem setting in mp-QP and an algorithm to solve mp-QP problems accompanied by comments. In Section 3 the MPC as particular case of mp-QP is analyzed. Then Section 4 is dedicated to the implementation of the piecewise affine linear solutions as functions of the state in case of mp-QP in terms of a neuro-fuzzy approach using adaptive network-based fuzzy inference systems. Section 5 deals with the applications of the mp-QP problems in case studies concentrated on the well accepted double integrator plant in several settings, and the last Section highlights the conclusions.

2 Problem Setting in Multi-parametric Quadratic Programming

The definition of an mp-QP problem is [2, 6]:

$$\hat{J}(x) = \min_{z} J(z,x) = \frac{1}{2} z' Hz \text{ subject to } Gz \le W + Sx, \tag{1}$$

where $z \in R^s$ are the optimization (manipulated) variables, $x \in R^n$ is the parameter vector, $H \in R^{s \times s}$, $H > 0$, $W \in R^m$, $S \in R^{m \times n}$. Other non-homogenous problems with the general objective function (2):

$$J(z,x) = z' Hz + x' Fz, \tag{2}$$

can always be transformed in the problem (1) using the variable substitution (3):

$$\tilde{z} = z + H^{-1} F' x. \tag{3}$$

To solve the mp-QP problem (1) it is necessary to calculate the polyhedral partition of the parameter space. Polytopic (or, more general, polyhedral) sets are an integral part of multi-parametric programming. With this respect, the following three definitions are useful [15].

Definition 1: A convex set $\mathrm{Q} \subseteq R^n$ given as an intersection of a finite number of closed half-spaces:

$$\mathrm{Q} = \left\{ x \in R^n \mid Q^x x \le Q^c \right\}, \tag{4}$$

is called polyhedron.

Definition 2: A bounded polyhedron $\mathrm{P} \subseteq R^n$:

$$\mathrm{P} = \left\{ x \in R^n \mid P^x x \le P^c \right\}, \tag{5}$$

is called polytope.

It is obvious from these two definitions that every polytope represents a convex, compact (i.e., bounded and closed) set.

Definition 3: The linear inequality $a' x \le b$ is called valid for a polyhedron P if $a' x \le b$ holds for all $x \in \mathrm{P}$. A subset of a polyhedron is called a face of P if it is represented as:

$$\mathrm{F} = \mathrm{P} \cap \left\{ x \in R^n \mid a' x = b \right\}, \tag{6}$$

for some valid inequality $a' x \le b$. The faces of polyhedron P of dimension 0, 1, $(n-2)$ and $(n-1)$ are called vertices, edges, ridges and facets, respectively.

Connecting to the mp-QP problem (1), given a close polyhedral set $\mathrm{K} \subset R^n$ of parameters:

$$\mathrm{K} = \{x \in R^n \mid Tx \le z\}, \tag{7}$$

it is denoted by $\hat{\mathrm{K}} \subset \mathrm{K}$ the region of parameters $x \in \mathrm{K}$ such that the mp-QP problem (1) is feasible and the optimum $\hat{J}(x)$ is finite.

The fundamental aspect of multi-parametric approaches to optimization is that for any given $\bar{x} \in \mathrm{K}$, $\hat{J}(x)$ denotes the minimum value of the objective function in (1) for $x = \bar{x}$, the function $\hat{J} : \hat{\mathrm{K}} \to R$ called value function, expresses the dependence on x of the minimum value of the objective function over $\hat{\mathrm{K}}$, and the single-valued function $\hat{z} : \hat{\mathrm{K}} \to R$ describes for any fixed $x \in \hat{\mathrm{K}}$ the optimizer $\hat{z}(x)$ corresponding to $\hat{J}(x)$.

To solve the mp-QP problem (1), the algorithm consisting of the following steps can be used [2]:

- Define the matrices H, G, W and S of the problem and set K in (7) according to the desired CS performance objectives.
- Calculate the partition of K according to the steps 1 … 9:

1: Let $x_0 \in \mathrm{K}$ the centre of the largest ball contained in K for which a feasible z exists, and ε the solution to the linear programming problem (8) related to this centre:

$$\varepsilon = \max_{z,x} f \text{ subject to } T_i x + f \| T_i \| \le Z_i,\ i = \overline{1, n_T},\ Gz - Sx \le W, \tag{8}$$

where n_T stands for the number of rows T_i of the matrix T.

2: If $\varepsilon \le 0$, then the partition is calculated (no full dimensional critical region is in K).

Else, continue with step 3.

3: Solve the mp-QP (1) for $x = x_0$ to obtain $(\hat{z}_0, \hat{\lambda}_0)$.

4: Determine the set of active constraints A_0 when $z = \hat{z}_0$, $x = x_0$, and build G_{A0}, W_{A0} and S_{A0}.

5: If $r = \operatorname{rank} G_{A0} < l$ (the number of rows of G_{A0}), then take a subset of r linearly independent rows, and redefine G_{A0}, W_{A0} and S_{A0} accordingly.

6: Determine $\hat{\lambda}_{A0}(x)$ and $\hat{z}_{A0}(x)$ from (9):

$$\hat{\lambda}_{A0}(x) = -(G_{A0}H^{-1}G_{A0}')^{-1}(W_{A0} + S_{A0}x),\ \hat{z}_{A0}(x) = -H^{-1}G_{A0}'\hat{\lambda}_{A0}(x)\,. \tag{9}$$

7: Characterize the critical region from (10) where the first inequality corresponds to the constraints in (1) and the second one to the Lagrange multipliers in (9) that must remain nonnegative as *x* is variable:

$$\begin{aligned} &GH^{-1}G_{A0}'(G_{A0}H^{-1}G_{A0}')^{-1}(W_{A0} + S_{A0}x) \le W + Sx, \\ &-(G_{A0}H^{-1}G_{A0}')^{-1}(W_{A0} + S_{A0}x) \ge 0. \end{aligned} \tag{10}$$

8: Define and partition the rest of the region according to [2].

9: Partition each new sub-region.

This algorithm explores the set K of parameters recursively. The partition of the rest of the region into polyhedral sets can be represented as a search tree, with a maximum depth equal to the number of combinations of active constraints.

Concerning the controller implementation, at each sampling interval a polyhedral partition is calculated, with as many different partitions as the length of the prediction horizon. If the Receding Horizon Technique (RHT) is used, then only one polyhedral partition is used out of those which were calculated. That is the reason why the generated controller has only one form.

Concerning the algorithm presented before, the following three comments must be highlighted. Firstly, the very first partition is based on choosing x_0 adequately. Since a choice is involved, there is no single solution to the algorithm.

Secondly, the determination of the set of active constraints is critical, since the remaining regions (the regions which are left when subtracting from the feasible solutions K) the starting, central region are partitioned based on some combination of active constraints. Namely, each region represents a region in the parameter space where a number of combinations is active. There is an upper bound for the maximum number, 2^m, where *m* is the cardinal of the set of all constraints.

Thirdly, the selection of active constraints is not a simple task. All algorithms are based on an iterative procedure that builds up the parametric solution by generating new polyhedral regions of the parameter space at each step. The methods differ in the way they explore the parameter space, that is, in the way they identify active constraints corresponding to the critical regions neighbouring to a given critical region [2]. In [6] the unconstrained critical region is constructed and then the neighbouring critical regions are generated by enumerating all possible combinations of active constraints.

Note that the expression of (9) and the fact that the algorithm is applied for several regions, the control signal will be a piecewise affine linear function of the state.

3 Model Predictive Control as Multi-parametric Quadratic Programming Problem

MPC is employed to solve the constrained regulation problem [6] consisting of regulating towards the origin the plant represented by the discrete-time linear time invariant model (11):

$$\begin{aligned} x(t+1) &= A\,x(t) + B\,u(t), \\ y(t) &= C\,x(t), \end{aligned} \tag{11}$$

fulfilling the constraints (12):

$$y_{\min} \le y(t) \le y_{\max},\ u_{\min} \le u(t) \le u_{\max}, \tag{12}$$

at all time instants $t \ge 0$, where $x(t) \in R^n$, $u(t) \in R^m$ and $y(t) \in R^p$ are the state, input and output vectors, respectively, the variables in (12) are vectors with appropriate dimensions and the pair (A, B) is stabilizable.

Assuming that the state $x(t)$ is fully available at the current time instant t, the MPC is defined in terms of (13):

$$\begin{aligned} \hat{U} = [\hat{u}'_1,...,\hat{u}'_{t+Nu-1}]' = \underset{U=[u'_1,...,u'_{t+Nu-1}]'}{\arg\min}\ & J(U, x(t)) = x'_{t+Ny|t} P x_{t+Ny|t} + \\ & + \sum_{k=0}^{Ny-1} [x'_{t+k|t} Q x_{t+k|t} + u'_{t+k} R\, u_{t+k}] \text{ subject to} \\ y_{\min} \le y_{t+k|t} \le y_{\max},\ & k = \overline{1, N_c}, \\ u_{\min} \le u_{t+k} \le u_{\max},\ & k = \overline{0, N_c}, \\ x_{t|k} = x(t),\ & \\ x_{t+k+1|t} = A\,x_{t+k|t} + B\,u_{t+k},\ & k \ge 0, \\ y_{t+k|t} = C\,x_{t+k|t},\ & k \ge 0, \\ u_{t+k} = K\,x_{t+k|t},\ & N_u \le k < N_y. \end{aligned} \tag{13}$$

The MPC problem (13) is solved at each time instant t, where $x_{t+k|t}$ is referred to as the predicted state vector at time t+k, obtained by applying the input sequence u_t, …, u_{t+k-1} to the plant (11) starting from the state $x(t)$ in the conditions of K being the state-feedback gain. It is assumed in (13) that $Q = Q' \ge 0$, $R = R' > 0$, $P \ge 0$ and $(Q = C'C, A)$ is detectable. The other parameters in (13) specific to MPC are N_y, N_u and N_c representing the output, input and constraint horizon, respectively, with $N_u \le N_y$ and $N_c \le N_y - 1$.

Substituting the state vector $x_{t+k|t}$:

$$x_{t+k|t} = A^k \mathrm{x(t)} + \sum_{j=0}^{k-1} A^j B u_{t+k-1-j} \ , \tag{14}$$

(13) can be re-expressed as the mp-QP problem (15):

$$V(x(t)) = \frac{1}{2} x'(t) Y x(t) + \min_U \frac{1}{2} U' H\, U + x'(t) F\, U \text{ subject to}$$
$$GU \leq W + Ex(t), \tag{15}$$

where the column vector $U \in R^s$, $s = mN_u$, is the optimization vector, $H = H' > 0$, and the matrices H, F, Y, G, W and E are obtained from Q, R and (13) [6].

Connecting the approaches in Section 2 and Section 3, the control signal elaborated by the model predictive controller is also a piecewise affine linear function of the state. The following Section will be dedicated to a neuro-fuzzy implementation of the controllers as solutions to mp-QP problems.

4 Neuro-fuzzy Implementation of Piecewise Affine Linear Functions of the State

Examining the algorithm presented in Section 2, the control signal u as piecewise affine linear function is considered in the particular case of a second-order system, $n = 2$, where the general function can be expressed as:

$$u = f(x_1, x_2) \ . \tag{16}$$

Considering three linguistic terms for each of the two input variables, defined in their initial forms in Fig. 1, the complete rule base of a Takagi-Sugeno fuzzy system [16] consisting of 9 rules, $R_1 \dots R_9$, can be expressed in terms of (17):

$$\begin{aligned}
&R_1 : \text{IF } [x_1 \text{ IS } X_{1N}] \, AND \, [x_2 \text{ IS } X_{2N}] \text{ THEN } [u = A_1 x_1 + B_1 x_2 + C_1] \\
&R_2 : \text{IF } [x_1 \text{ IS } X_{1Z}] \, AND \, [x_2 \text{ IS } X_{2N}] \text{ THEN } [u = A_2 x_1 + B_2 x_2 + C_2] \\
&R_3 : \text{IF } [x_1 \text{ IS } X_{1P}] \, AND \, [x_2 \text{ IS } X_{2N}] \text{ THEN } [u = A_3 x_1 + B_3 x_2 + C_3] \\
&R_4 : \text{IF } [x_1 \text{ IS } X_{1N}] \, AND \, [x_2 \text{ IS } X_{2Z}] \text{ THEN } [u = A_4 x_1 + B_4 x_2 + C_4] \\
&R_5 : \text{IF } [x_1 \text{ IS } X_{1Z}] \, AND \, [x_2 \text{ IS } X_{2Z}] \text{ THEN } [u = A_5 x_1 + B_5 x_2 + C_5] \ . \\
&R_6 : \text{IF } [x_1 \text{ IS } X_{1P}] \, AND \, [x_2 \text{ IS } X_{2Z}] \text{ THEN } [u = A_6 x_1 + B_6 x_2 + C_6] \\
&R_7 : \text{IF } [x_1 \text{ IS } X_{1N}] \, AND \, [x_2 \text{ IS } X_{2P}] \text{ THEN } [u = A_7 x_1 + B_7 x_2 + C_7] \\
&R_8 : \text{IF } [x_1 \text{ IS } X_{1Z}] \, AND \, [x_2 \text{ IS } X_{2P}] \text{ THEN } [u = A_8 x_1 + B_8 x_2 + C_8] \\
&R_9 : \text{IF } [x_1 \text{ IS } X_{1P}] \, AND \, [x_2 \text{ IS } X_{2P}] \text{ THEN } [u = A_9 x_1 + B_9 x_2 + C_9]
\end{aligned} \tag{17}$$

The membership function shapes in Fig. 1 have been obtained supposing that the fuzzy system includes the scaling factors (which can be non-linear in case of implementing MPC), and they correspond to so-called dsigmoidal-type functions according to [14] with the expressions in (18) as difference of two sigmoidal ones:

$$\mu_i(x) = 1/\{1+\exp\{-a_i(x-c_i)]\} - 1/\{1+\exp\{-a_{i+1}(x-c_{i+1})]\},$$
$$x \in \{x_1, x_2\},\ i \in \{1,3,5,7,9,11\}. \tag{18}$$

In the conditions of using use the max and min operators in the inference engine and the weighted average method for defuzzification due to the special form of the consequence in each rule it is easy to observe that the Takagi-Sugeno fuzzy system described here is well suited to model piecewise affine linear functions of the state appearing in case of mp-QP problems.

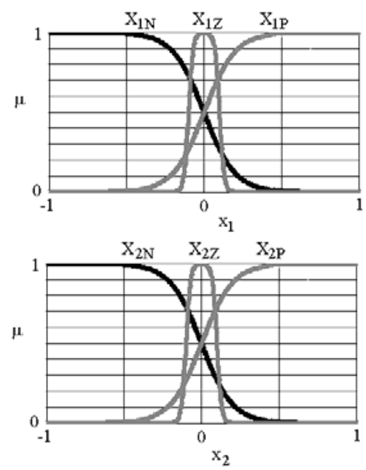

Figure 1
Initial membership functions of input linguistic terms

The adaptive network-based fuzzy inference system, with the schematic diagram illustrated in Fig. 2, has 5 layers:

- Layer I: The inputs x_1 and x_2 are passed through the input linguistic terms with the membership functions having the parameters presented in (18),
- Layer II: The fuzzy rules in (17) are constructed according to the inference engine using the product (Π) of the two antecedents, the output of each node representing the firing strength of a rule,
- Layer III: The firing strengths are normalized dividing them to the sum of all rule firing strengths,

- Layer IV: The normalized firing strength is multiplied by the output functions of the fuzzy inference system,
- Layer V: The overall system output is calculated as the sum of all incoming signals.

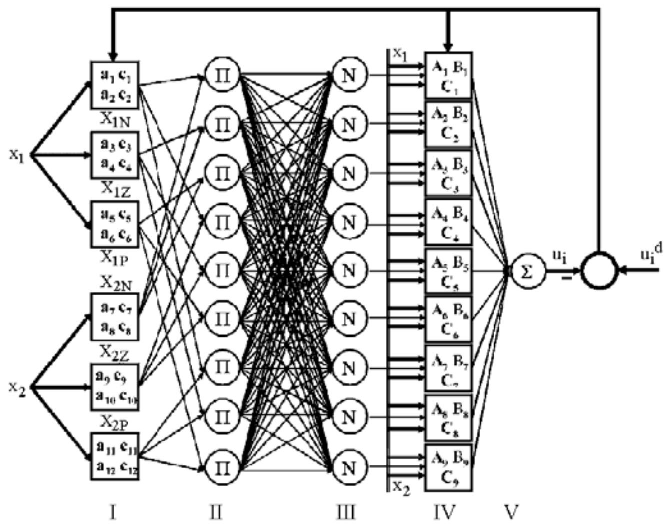

Figure 2
Schematic diagram of adaptive network-based fuzzy inference system

Due to this structure similar to that of certain neural networks, the learning techniques specific to neural networks can be applied to minimize the quadratic objective function *E*:

$$E = \sum_{i=1}^{N} (u_i^d - u_i)^2 \ , \tag{19}$$

where: u_i^d – desired control signal, piecewise affine linear function of the state, to be implemented by the Takagi-Sugeno fuzzy system, u_i – control signal elaborated by the Takagi-Sugeno fuzzy system, N – number of input-output data pairs.

Minimizing the objective function in (19) using the adaptive structure presented in Fig. 2 and appropriate learning techniques leads to the correction of the parameters of the Takagi-Sugeno fuzzy systems in both the antecedents and the consequents.

5 Case Studies

Consider the double integrator plant in its continuous-time representation (for zero initial conditions) (20):

$$y(s) = \frac{1}{s^2} u(s)\,, \tag{20}$$

and its equivalent discrete-time state-space representation obtained by discretization in terms of the forward rectangular method for a sampling period of 1 s:

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t), \end{aligned} \tag{21}$$

and its first setting where the system constraints are:

$$-1 \le u \le 1\,,\quad -15 \le y \le 15\,,\quad -15 \le x_1, x_2 \le 15\,. \tag{22}$$

The CS behaviour is illustrated in Fig. 3.

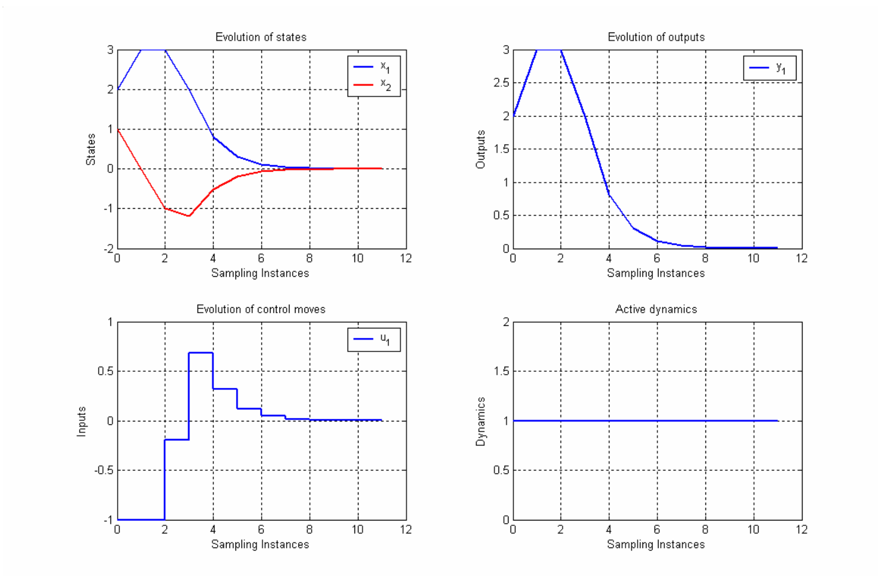

Figure 3
Control system behaviour

Applying the mp-QP algorithm presented in Section 2 leads in the first phase to 25 controller regions. Merging the controller regions, in this case it was achieved from 25 regions to 21 regions, for the last sampling interval. These regions can be plotted. The control signal will have the piecewise affine linear form (23) that can be well modelled using the neuro-fuzzy approach presented in the previous Section:

$$u = \begin{bmatrix} -0.57917087112176 & -1.54562698132617 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.43504549869264 & -1.42514751747608 \\ -0.43504549869264 & -1.42514751747608 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.34224024245851 & -1.33799618358994 \\ -0.34224024245851 & -1.33799618358994 \\ -0.21419288842852 & -1.21419288842852 \\ -0.21419288842852 & -1.21419288842852 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1.00000000000000 \\ -1.00000000000000 \\ -1.00000000000000 \\ -1.00000000000000 \\ -1.00000000000000 \\ -1.00000000000000 \\ -1.00000000000000 \\ 0.45607670792517 \\ -0.45607670792517 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 0.87990997751597 \\ -0.87990997751597 \\ 1.72724809760682 \\ -1.72724809760682 \end{bmatrix}. \quad (23)$$

In the second setting, the objective is to regulate the double integrator (21) towards the origin while minimizing the following quadratic performance index and input constraint [6]:

$$J = \sum_{t=0}^{\infty} [y'(t)y(t) + 0.01u^2(t)], \ -1 \le u \le 1 \ . \quad (24)$$

This problem can be solved by using the MPC algorithm in (13) with $N_u = N_y = 2$, $R = 0.01$, $Q_{11} = 1$, $Q_{12} = Q_{21} = Q_{22} = 0$, and the matrices K and P obtained solving a Riccati equation [6].

Transforming the MPC problem (13) into a mp-QP one (15), the solution to the MPC algorithm can be calculated as piecewise affine linear functions of the state.

Conclusions

The paper presents aspects concerning the mp-QP problems with solutions implemented in state-feedback form as piecewise affine linear functions of the state, very convenient to he implemented as Takagi-Sugeno fuzzy systems using neuro-fuzzy techniques.

Future research will deal with the computer-aided solving of mp-QP and MPC problems and with applications to discrete-event systems in manufacturing and robot control [17, 18, 19].

Acknowledgement

The support stemming from the cooperation between Budapest Tech Polytechnical Institution and "Politehnica" University of Timisoara in the framework of the Hungarian-Romanian Intergovernmental Science & Technology Cooperation Program no. 35 ID 17 and from two CNCSIS grants is kindly acknowledged.

References

[1] T. Gal, J. Nedoma: Multiparametric linear programming, in Management Science, Vol. 18, 1972, pp. 406-442

[2] F. Borelli: Constrained Optimal Control of Linear and Hybrid Systems, Springer-Verlag, Berlin, Heidelberg, 2003

[3] M. Schechter: Polyhedral functions and multiparametric linear programming, in Journal of Optimal Theory and Applications, Vol. 53, No. 2, 1987, pp. 269-280

[4] T. Gal: Postoptimal Analyses, Parametric Programming, and Related Topics, de Gruyer, Berlin, 2nd edition, 1995

[5] C. Filippi: On the geometry of optimal partition sets in multiparametric linear programming, Technical Report 12, Department of Pure and Applied Mathematics, University of Padova, Italy, 1997

[6] A. Bemporad, M. Morari, V. Dua, S. Pistikopoulos: The explicit linear quadratic regulator for constrained systems, in Automatica, Vol. 38, No. 1, 2002, pp. 3-20

[7] M. M. Seron, J. A. De Doná, G. C. Goodwin: Global analytical model predictive control with input constraints, in Proceedings of 39th IEEE Conference on Decision and Control, Sydney, Australia, 2000, pp. 154-159

[8] P. Tøndel, T. A. Johansen, A. Bemporad: An algorithm for multi-parametric quadratic programming and explicit MPC solutions, in Proceedings of 40th IEEE Conference on Decision and Control, Orlando, FL, Vol. 3, 2001, pp. 2849-2854

[9] M. Baotic: An efficient algorithm for multi-parametric quadratic programming, Technical Report AUT02-04, Automatic Control Laboratory, ETH Zürich, Switzerland, 2002

[10] E. F. Camacho, C. Bordons: Model Predictive Control, Springer-Verlag, Berlin, Heidelberg, 2nd edition, 2004

[11] H. W. Chiou, E. Zafiriou: Frequency domain design of robustly stable constrained model predictive controllers, in Proceedings of the American Control Conference, Baltimore, USA, Vol. 3, 1994, pp. 2852-2856

[12] T. A. Johansen, I. Petersen, O. Slupphaug: On explicit suboptimal LQR with state and input constraints, in Proceedings of 39th IEEE Conference on Decision and Control, Sydney, Australia, 2000, pp. 662-667

[13] J. Jang: ANFIS: Adaptive network-based fuzzy inference system, in IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, 1993, pp. 665-685

[14] Mathworks: Fuzzy Logic Toolbox User's Guide V. 3, Natick, MA, 2000

[15] M. Kvasnica, P. Grieder, M. Baotić, F. J. Christophersen: Multi-Parametric Toolbox (MPT), Tutorial

[16] T. Takagi, M. Sugeno: Fuzzy identification of systems and its applications to modelling and control, in IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, No. 1, 1985, pp. 116-132

[17] L. Horváth, I. J. Rudas, J. F. Bitó, A. Szakál: Feature driven associative part and machining process models, in Knowledge and Technology Integration in Production and Services, Kluver Academic Publishers, Boston, 2002, pp. 11-18

[18] L. Horváth, I. J. Rudas: Modeling and Problem Solving Methods for Engineers, Elsevier, Academic Press, 2004

[19] L. Horváth, I. J. Rudas: Possibilities for application of associative objects with built-in intelligence in engineering modeling, in Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 8, No. 5, 2004, pp. 544-551