2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Model Predictive Control for Discrete Fuzzy Systems via Iterative Quadratic Programming

Carlos Ariño, Emilio Pérez and Andrés Querol
D. de Ingeniería de Sistemas Industriales y Diseño
Universitat Jaume I, Avenida Vicent Sos Baynat, s/n.
12071 Castelló de la Plana, Spain.
Email: arino@uji.es

Antonio Sala
Inst. Universitario de Automática e Informática Industrial
Universidad Politécnica de Valencia
Camino de Vera S/N 46022 - Valencia, Spain.

***Abstract*—Takagi-Sugeno fuzzy models are exact representations of nonlinear systems in a compact region. Guaranteed-cost linear matrix inequalities produce controllers which minimize a shape-independent bound on a quadratic cost; however, the controller has a fixed structure (possibly suboptimal), say a Parallel Distributed Compensator (PDC), and does not allow input saturation. By posing the problem as a Model Predictive Control one, the ideas of terminal set, terminal controller and feasible set can be used in order to improve the performance of usual guaranteed-cost controllers for Takagi-Sugeno systems via Quadratic Programming. A Polya-based approach has been introduced in order to (conservatively) transform the invariant set problem into a polytopic one, as well as computing the controller feasibility region. The optimal controller is computed iteratively.**

***Index Terms*—Discrete Takagi-Sugeno Fuzzy Models, Invariant sets, Contractive sets and Robust Stability.**

I. Introduction

Takagi-Sugeno (TS) fuzzy models are exact representations of nonlinear systems in a compact region (modelling region, Ω) if well-known systematic sector-nonlinearity methodologies [1] are used.

Techniques based on Linear Matrix Inequalities (LMI) have allowed to obtain a wide range of fuzzy controllers following different specifications (stability, decay, $\mathcal{H}_\infty$, ...). Many of them result in closed-loop expressed as multi-dimensional fuzzy summations. In particular, guaranteed-cost ones [2] are those which generalize to TS models, with some conservativeness, the usual optimization of infinite-cost quadratic indices in linear quadratic regulator (LQR) control.

However, as fuzzy models are usually valid only locally in the compact region Ω, performance guarantees are usually stated only on level sets of the obtained Lyapunov functions included in the modelling region [3]. So, implicitly, the actual fuzzy control problem *should* incorporate state constraints arising from the local modelling setup. Such constraints are usually enforced via Lyapunov level sets but the actual valid initial condition region might be quite larger than that arising from the level sets [4]. Also, in realistic applications, there is always control saturation which is not easy to handle in LMI framework: most conditions actually require the control action to *avoid* saturation in the outermost Lyapunov level set or, if that is not the case [5], either cannot prove improvement with respect to non-saturating laws or require iteration/Bilinear Matrix Inequalities (BMI) [6].

In a linear case, state and input constraints are handled with on-line finite-horizon optimization in model predictive control [7] (MPC). Stability and infinite-horizon optimality of receding-horizon predictive laws is ensured for all initial states in a so-called *feasible set* if a so-called *terminal controller* can be found which does not hit any constraint in future time for all initial states in a *terminal set.* These are well known concepts in the linear MPC framework [7] which, however, are much harder to deal with in nonlinear systems.

The objective of this paper is adapting the above considerations to TS fuzzy systems. As there are some causes of conservatism (in particular shape-independence and fuzzy summation issues [8]), subsets of the actual invariant and feasible sets are computed for a PDC terminal controller. Also, as future optimal trajectories are unknown, an iterative procedure is reported in order to converge to the optimal one for the original nonlinear system (actually suboptimal because convergence is sought only in the finite-horizon segment). The result, albeit suboptimal (because the terminal controller is conservative because of shape-independence and Polya-like fuzzy summation issues), improves over the terminal controller both in achieved cost and in the enlarged feasible zone.

The structure of the paper is as follows: next section discusses preliminary notation and the concrete problem statement above outlined. Section III discusses the proposed setup for adapting MPC to fuzzy systems, first considering the terminal cost, and later the terminal set, feasible set, plus an iterative algorithm to compute the optimal transient system trajectories (Section III-D). Section IV proposes an example in which the main concepts are illustrated. A conclusion section closes the paper.

II. Preliminaries and Problem statement

Consider a nonlinear discrete-time system to be controlled, given by a model:

$$x_{k+1} = f_x(x_k) + f_u(x_k)u_k \tag{1}$$

such that $f_x(0) = 0$. This system can be expressed *locally* in a compact region of interest Ω containing the origin as a TS [9] fuzzy system with r rules or local models in the form:

$$x_{k+1} = \sum_{i=1}^{r} \mu_i(x_k)(A_i x_k + B_i u_k) \tag{2}$$

where $x_k \in \mathbb{R}^n$ represent the states, $u_k \in \mathbb{R}^m$ the control actions and μ_i represent membership functions such that:

$$\sum_{i=1}^{r} \mu_i(x_k) = 1, \quad \mu_i(x_k) \geq 0 \ \forall x \ i : 1 \ldots r \tag{3}$$

If a fuzzy PDC state-feedback controller were used,

$$u_k = -\sum_{i=1}^{r} \mu_i K_i x_k \tag{4}$$

the closed loop has the form:

$$x_{k+1} = \sum_{i=1}^{r} \sum_{j=1}^{r} \mu_i \mu_j ((A_i - B_i K_j) x_k) \tag{5}$$

Note that the dependence of the membership functions on x_k has been omitted for brevity.

Let us also consider in our problem formulation some input and state constraints. When these constraints are linear they can be defined by the appropriated matrices R and S, and vectors r, s such that:

$$\mathbb{X} = \{x \in \mathbb{R}^n \mid Lx + l \leq 0\} \tag{6}$$

$$\mathbb{U} = \{u \in \mathbb{R}^m \mid Su + s \leq 0\} \tag{7}$$

A. Problem statement

In literature, guaranteed cost control is used to synthesize PDC controllers in the form (4) without taking into account the state and input constraints.

The objective of this paper is using such controllers as *terminal* controllers in predictive-control-like strategies for fuzzy TS systems in order to (partly) overcome the conservativeness arising from:

- the worst-case (membership independent) approach,
- the limited choice of Lyapunov functions and
- ensuring the satisfaction of the above defined constraints in the largest possible initial condition region.

In summary, even if terminal controllers are conservative, results (guaranteed cost bounds) will improve due to the addition of a finite-horizon segment with less conservative assumptions.

III. Fuzzy Model Predictive Control

MPC can be defined as a Constrained online optimization based on a model prediction. The essential parts of a MPC are:

- A model that will be able to describe the behavior of the future states.
- An objective function that represents the performance of the controlled system.
- An optimizer, that minimizes the objective function subject to the proper constraints.
- The receding horizon strategy, that implies that the optimizer has to solve the problem at each step.

The model that will be used on the MPC formulation is the following TS one:

$$x_{k+1} = \sum_{i=1}^{r} \mu_i(\tilde{x}_k)(A_i x_k + B_i u_k) \tag{8}$$

Note that the main difference between (2) and (8) is that the membership functions depend on a new variable $\tilde{x}_k$. This variable is an estimation of the optimal states at time k, at the prior iteration in an algorithm to be later introduced. The introduction of that variable simplifies the problem significantly, because the non-linear dependence of the TS model can be evaluated at the beginning and then it will not be introduced into the optimization problem. The obvious drawback of that simplification is that many times $\tilde{x}_k$ may be quite different from the predicted optimal x_k, so this motivates the mentioned iterative approach.

A. The objective function

Ideally, the proposed objective function would be a quadratic performance function of the states and the inputs such as:

$$J_\infty = \sum_{k=0}^{\infty} \left(x_k^T H x_k + u_k^T F u_k \right) \tag{9}$$

However, the main drawback of the function (9) is that it is not numerically tractable (except in the well-known linear time-invariant case) because of the infinite-horizon objective. In order to avoid this problem, a finite horizon function is usually introduced in MPC:

$$J_N = x_N^T P x_N + \sum_{k=0}^{N-1} \left(x_k^T H x_k + u_k^T F u_k \right) \tag{10}$$

For performance and stability reasons, it will be interesting that our proposed finite horizon performance function J_N bounds the optimal infinite-horizon one J_∞ ($J_\infty \leq J_N$) while making the gap between both functions as small as possible.

To do so, analogous considerations as those in [10] for continuous systems have been done for the discrete TS case in this paper. This way, a matrix P must be found which bounds the term of the infinite horizon $J_{N\to\infty}$

$$J_{N\to\infty} = \sum_{k=N}^{\infty} \left(x_k^T H x_k + u_k^T F u_k\right) \tag{11}$$

such that

$$J_{N\to\infty} \leq x_N^T P x_N \tag{12}$$

This bounding can be achieved by constraining the per-stage weighting with the condition (13)

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k < -(x_k^T H x_k + u_k^T F u_k) \tag{13}$$

Indeed, if (13) holds, summing from $k = N$ to $k = \infty$ and assuming the resulting controller will be stabilising so $x_\infty = 0$, the cost index (11) can be bounded by $x_N^T P x_N$

$$\sum_{k=N}^{\infty} (x_k^T H x_k + u_k^T F u_k) < x_N^T P x_N \tag{14}$$

Using the Schur complement, following well-known argumentations [11], a controller for which (13) holds can be found if there exist matrices M_i, $X > 0$ such that

$$\Gamma_{ij} = \begin{pmatrix} X & XA_i^T - M_j^T B_i^T & X & M_j^T \\ A_i X - B_i M_j & X & 0 & 0 \\ X & 0 & H^{-1} & 0 \\ M_j & 0 & 0 & F^{-1} \end{pmatrix} \tag{15}$$

$$\sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j \Gamma_{ij} > 0 \tag{16}$$

where $P = X^{-1}$ and the controller is defined as a PDC (4) with

$$K_i = M_i X^{-1} \tag{17}$$

The worst-case bound of the cost function (11) is minimized if the eigenvalues of X are maximized, that is

$$\min -\lambda$$

subject to (16) and $X > \lambda$.

Note that (16) is a fuzzy summation which can be, conservatively, expressed as an LMI following any of the relaxations in [12], [13], [14].

B. Terminal Set

Many times, the PDC controller (17) can not be applied in the whole state space definition $\mathbb{X}$ defined in (6), because some inputs defined with this PDC controller will not verify the input constraints defined in (7). Also, maybe even if they do in a particular instant, the future optimal trajectory may exit Ω or even without exiting, it might violate the input bounds.

Then it is mandatory to obtain a set from with this controller can be applied and the system will be stable and optimal, and future states must also belong to that set. Following predictive-control argumentations, the invariant set of this controller has to be computed. It can be done following the algorithm presented in [15]. This algorithm is based on [16], adapted to TS Fuzzy models applying the Polya theorem and is also presented here as Algorithm 1.

Algorithm 1 Calculation of the closed-loop N-step invariant set $\mathbb{K}_N(\Omega, \mathbb{T})$

1) Make $i = 0$ and $\mathbb{K}_0(\Omega, \mathbb{T}) = \mathbb{T}$
2) While $i < N$:
   a) $\mathbb{K}_{i+1}(\Omega, \mathbb{T}) = \mathcal{Q}(\mathbb{K}_i(\Omega, \mathbb{T})) \cap \Omega$
   b) If $\mathbb{K}_{i+1}(\Omega, \mathbb{T}) = \mathbb{K}_i(\Omega, \mathbb{T})$, end algorithm and $\mathbb{K}_N(\Omega, \mathbb{T}) = \mathbb{K}_\infty(\Omega, \mathbb{T}) = \mathbb{K}_i(\Omega, \mathbb{T})$.
   c) i=i+1

where $\mathbb{T}$ is a target set; Ω is a generic set in the states space; and $\mathbb{K}_i(\Omega, \mathbb{T})$ denotes the subset of Ω that steers the system to $\mathbb{T}$ in at most i steps.

The algorithm needs to compute iteratively the one-step set $\mathcal{Q}(\Omega) = \{x \in \Omega | x_{k+1} \in \Omega\}$. This set, in a general case, is a complicated one arising from the non-linear dynamics embedded in the TS models.

In order to avoid this problem, an approximation of $\mathcal{Q}$ can be done using the Polya expanded TS model

$$x_{k+1} = \left(\sum_{i=1}^{r} \mu_i(x)\right)^{d-2} \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i(x)\mu_j(x) G_{ij} x_k \tag{18}$$

where $G_{ij} = A_i - B_i K_j$. Note that this model is equivalent to (5) as $\sum_{i=1}^{r} \mu_i(x) = 1$.

The Polya-expanded model in (18) is a d degree vector polynomial of μ_i and a suitable matrix $\tilde{G}_{\mathbf{i}}$ can be found such that

$$x_{k+1} = \sum_{\mathbf{i} \in \mathbb{I}_d^+} n_{\mathbf{i}} \mu_{\mathbf{i}}(x) \tilde{G}_{\mathbf{i}} x_k \tag{19}$$

where each $\mu_{\mathbf{i}}$ represents one of the possible monomials $\prod \mu_j$ of degree d; $\mathbb{I}_d^+$ is the set of all the different monomials of degree d; and $n_{\mathbf{i}}$ is the number of times that this monomial appears in (18). For further details, the reader is referred to [15].

With this notation, the one-step set can be expressed as

$$\mathcal{Q}(\Omega) = \{x \in \mathbb{R}^n | \sum_{\mathbf{i} \in \mathbb{I}_d^+} n_{\mathbf{i}} \mu_{\mathbf{i}}(x) \tilde{G}_{\mathbf{i}} x \in \Omega\} \tag{20}$$

Due to $\mu_{\mathbf{i}}$ and $n_{\mathbf{i}}$ being positive, a sufficient condition for a point x to belong to (20), can be given by ensuring that

$$\mathcal{Q}(\Omega) \supset \tilde{\mathcal{Q}}_d(\Omega) = \{x \in \mathbb{R}^n | \tilde{G}_{\mathbf{i}} x \in \Omega\} \tag{21}$$

Note that $\tilde{\mathcal{Q}}_d(\Omega)$ is a polytopic subset of the one-step set. Furthermore, it can be proved that as d increases, $\tilde{\mathcal{Q}}_d(\Omega)$ asymptotically approaches to the maximal shape-independent subset of $\mathcal{Q}(\Omega)$ [4]. Now, this set can be used in Algorithm 1 in order to obtain an inner approximation of the Invariant Set which can be used as the terminal set in the MPC problem. We define this set as $\mathbb{Z} = \mathbb{K}_\infty(\Omega, \Omega)$ which is a polytope. Hence, it can be represented as

$$\mathbb{Z} = \{x \in \mathbb{R}^n | Zx \leq z\} \tag{22}$$

for a suitable choice of matrix Z and vector z.

C. Optimization Problem

Once the terminal cost P and the terminal set $\mathbb{Z}$ are obtained, given a known initial state x_0 and a first guess of the future optimal trajectory $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_{N-1})$, the following Quadratic Programming (QP) optimization problem $\mathcal{P}_N(x_0, \tilde{\mathbf{x}})$ can be stated:

$$\mathcal{P}_N(x_0, \tilde{\mathbf{x}}) : \text{find } u_0, \ldots, u_{N-1} \text{ which minimize}$$

$$J_N^{OPT}(x) = x_N^T P x_N + \sum_{k=0}^{N-1} \left(x_k^T H x_k + u_k^T F u_k \right) \tag{23}$$

subject to:

$$u_k \in \mathbb{U} \text{ for } k = 0, \ldots, N-1$$

$$x_{k+1} = \sum_{i=1}^{r} \mu_i(\tilde{x}_k)(A_i x_k + B_i u_k) \in \mathbb{X} \tag{24}$$

$$\text{for } k = 0, \ldots, N$$

$$x_N \in \mathbb{Z} \subset \mathbb{X}$$

At this point it is useful to remark that $\tilde{x}_k$ for $k = 1 \ldots N-1$ have to be already known in order to avoid the nonlinearities of the model's memberships and express this problem as a QP. In the next section, an iterative procedure will be presented to obtain this state estimates.

Let us show that, indeed, the problem is a standard QP one. First, note that the matrices below are known at the time of the computation

$$A(\tilde{x}_k) = \sum_{i=1}^{r} \mu_i(\tilde{x}_k) A_i, \quad B(\tilde{x}_k) = \sum_{i=1}^{r} \mu_i(\tilde{x}_k) B_i \tag{25}$$

With these matrices the prediction model can be expressed as:

$$\mathbf{x} = \Theta x_k + \Gamma \mathbf{u} \tag{26}$$

where $\mathbf{x} = (x_1^T \ldots x_N^T)^T$, $\mathbf{u} = (u_0^T \ldots u_{N-1}^T)^T$, Θ is defined in (27) and Γ in (28) on top of next page.

$$\Theta = \begin{pmatrix} A(\tilde{x}_0) \\ A(\tilde{x}_1)A(\tilde{x}_0) \\ \vdots \\ A(\tilde{x}_{N-1}) \ldots A(\tilde{x}_0) \end{pmatrix} \tag{27}$$

As Θ and Γ are easily computable matrices, following standard MPC procedures (see [7] for details), the optimization problem can be expressed as the following quadratic program on the vector of future controls $\mathbf{u}$:

$$\mathcal{P}_N(x_0, \tilde{\mathbf{x}}) : \text{minimize } \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + x_0^T \mathbf{F} \mathbf{u} \tag{29}$$

subject to:

$$\Phi \mathbf{u} \leq \Delta - \Lambda x_0 \tag{30}$$

where

$$\Phi = \begin{pmatrix} \mathbf{L}\Gamma \\ \mathbf{S} \end{pmatrix}, \quad \Delta = \begin{pmatrix} \mathbf{l} \\ \mathbf{s} \end{pmatrix}, \quad \Lambda = \begin{pmatrix} \mathbf{L}\Theta \\ 0 \end{pmatrix} \tag{31}$$

$$\mathbf{L} = \text{diag}(L, \ldots, L, Z), \quad \mathbf{l} = \text{diag}(l, \ldots, l, z) \tag{32}$$

$$\mathbf{S} = \text{diag}(S, \ldots, S), \quad \mathbf{s} = \text{diag}(s, \ldots, s) \tag{33}$$

$$\mathbf{H} = \Gamma^T [\text{diag}(H, \ldots, H, P)] \Gamma + \text{diag}(F, \ldots, F) \tag{34}$$

$$\mathbf{F} = \Theta^T [\text{diag}(H, \ldots, H, P)] \Gamma \tag{35}$$

D. Iterative computation of the state trajectory estimate

As previously stated, in the proposed optimization problem an state estimate $\tilde{x}_k$ is needed in intermediate steps. For a good prediction of the trajectories, it is needed that this estimate is as close as possible to the real future optimal state, $\tilde{x}_k \approx x_k^{OPT}$. However, as these future trajectories are unknown until the actual control action is computed, an iterative setup is needed in order to compute the optimal control action as well as the optimal trajectory.

To this end, Algorithm 2 below is presented. It has been implemented with end conditions considering some (application dependent) limitations on the available time ϵ_t for the computation and desired precision in the solution ϵ_x.

Algorithm 2 Iterative computation of the state estimate

1) Obtain initial estimate $\tilde{\mathbf{x}}$ from previous sampling step.
2) Solve the program $\mathcal{P}_N(x_0, \tilde{\mathbf{x}})$ obtaining u_k
3) $\tilde{x}_0^* = x_0$, $\tilde{x}_{k+1}^* = \sum_{i=1} \mu_i(\tilde{x}_k^*)(A_i \tilde{x}_k^* + B_i u_k)$ for $k = 0 \ldots N-2$
4) If $|\tilde{\mathbf{x}}^* - \tilde{\mathbf{x}}| > \epsilon_x |\tilde{\mathbf{x}}|$ and $t - t_0 < \epsilon_t$ go to step 2 with $\tilde{\mathbf{x}} := \tilde{\mathbf{x}}^*$

E. Feasible region

At this point, it is important to know the set of states where the proposed problem $\mathcal{P}_N(x_0, \tilde{\mathbf{x}})$ has a solution, given an horizon N. Otherwise, Algorithm 2 may be infeasible. This feasible set, can be computed as the set of states that can reach the terminal set in N steps while holding the imposed constraints in inputs and states. Of course, the larger the horizon, the larger the resulting set would be.

A possible way to compute this feasible set is applying Algorithm 1 with horizon N and $\mathbb{T} = \mathbb{Z}$, where $\mathbb{Z}$ is the terminal set previously computed in Section III-B. Now, as the input is not determined by a given "optimal" controller (only the existance of a valid input is needed), the one-step set is redefined as

$$\mathcal{Q}(\Omega) = \left\{ x \in \mathbb{R}^n | \exists u \in \mathbb{U}, \sum_{i=1}^{r} \mu_i(\tilde{x}_k)(A_i x + B_i u) \in \Omega \right\} \tag{36}$$

$$\Gamma = \begin{pmatrix} B(\tilde{x}_0) & 0 & \dots & 0 \\ A(\tilde{x}_1)B(\tilde{x}_0) & B(\tilde{x}_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\prod_{i=1}^{N-1} A(\tilde{x}_i))B(\tilde{x}_0) & (\prod_{i=2}^{N-1} A(\tilde{x}_i))B(\tilde{x}_1) & \dots & B(\tilde{x}_{N-1}) \end{pmatrix} \tag{28}$$

As the values of $\tilde{x}$ are uncertain, an inner approximation of this set is here proposed, which is shape-independent, i.e., valid for any possible value of μ_i:

$$\tilde{\mathcal{Q}}(\Omega) = \{x \in \mathbb{R}^n | \exists u \in \mathbb{U},\ A_i x + B_i u \in \Omega,\ \forall i = 1 \dots r\} \tag{37}$$

and the standard algorithm is applied with the above set for a number of steps equal to the finite-horizon N.

F. Receding Horizon Optimization and Stability

The optimal controller obtained by solving problem (29) is implemented, as usual in MPC, in a receding-horizon strategy in which only the first action u_0 is applied and, then, a new state is measured and everything is recomputed.

Given the fact that the terminal cost verifies (13), using the results in [7], assuming Algorithm 2 has converged to the optimal trajectory, then stability of the receding horizon implementation can be ensured; also, some contractive-set constraints [17] can be additionally enforced to ensure stability even if Algorithm 2 has not converged (details omitted for brevity).

IV. Example

This example will illustrate the proposed MPC methodology for a TS system

$$x_{k+1} = \sum_{i=1}^{r} \mu_i(x_k)(A_i x_k + B_i u_k) \tag{38}$$

with the local models and membership functions defined as (39)-(41)

$$A_1 = \begin{pmatrix} -0.9 & 0.3 \\ 0 & 0.4 \end{pmatrix} A_2 = \begin{pmatrix} 0.8 & 0.6 \\ -0.5 & 0.2 \end{pmatrix} \tag{39}$$

$$B_1 = \begin{pmatrix} 0.4 \\ 1.1 \end{pmatrix} B_2 = \begin{pmatrix} 0.9 \\ 0.3 \end{pmatrix} \tag{40}$$

$$\mu_1 = \frac{10 - x_1(k)}{20} \quad \mu_2 = 1 - \mu_1 \tag{41}$$

The system will be constrained in the input and states as given by

$$-1 \le u_k \le 1 \quad -10 \le x_k \le 10 \tag{42}$$

where state restrictions are understood as component-wise.

First, a terminal state weighting P and terminal PDC controller $u_k = \sum_{i=1}^{r} \mu_i K_i x_k$ are computed as discussed in section III-A with weighting matrices H and F being chosen as:

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad F = 1 \tag{43}$$

The obtained PDC controller gains K_i are

$$K_1 = (-0.3519 \quad 0.3136),\ K_2 = (0.3898 \quad 0.5664) \tag{44}$$

and the resulting terminal weighting P matrix is:

$$P = \begin{pmatrix} 8.5967 & -0.1159 \\ -0.1159 & 5.5136 \end{pmatrix} \tag{45}$$

Next, the terminal set is obtained following Algorithm 1 with constraints (46) and a Polya complexity index $d = 50$ in the computation of the inner approximation of the one-step set (21), with the state constraints arising from the use of the terminal controller, i.e.:

$$\begin{matrix} -10 \le x_k \le 10 \\ -1 \le K_1 x_k \le 1 \quad -1 \le K_2 x_k \le 1 \end{matrix} \tag{46}$$

The obtained terminal set is illustrated in Figure 1.

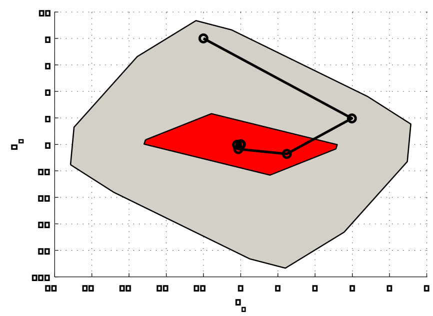

Fig. 1. Terminal Set (red), Feasible Set (grey), and state trajectory

At this point all the required elements for stating the QP fuzzy predictive control problem are already available. However, it is also interesting to obtain the set of states for which the optimization problem will we feasible, i.e. the feasible set, as described in Section III-E. Choosing an horizon of $N = 6$ the shape-independent feasible set presented in grey in Figure 1 is found.

Finally, in order to evaluate the MPC controller performance, the closed-loop trajectory from an arbitrarily chosen point $x_0 = (-1\ 8)^T$ is also shown in Figure 1. Algorithm 2 needs 2 iterations to find an state estimate of relative precision of $\epsilon_x = 0.1\%$.

Additionaly, time responses of the states and the control action are shown in figures 2 and 3 respectively, showing a fast convergence to the equilibrium point.

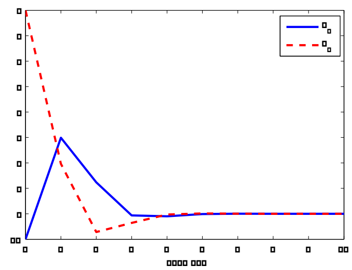

Fig. 2. States time response.

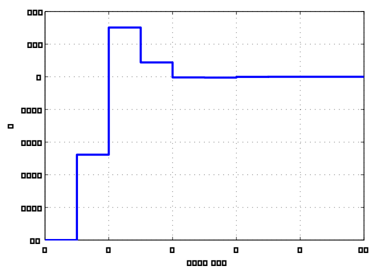

Fig. 3. Control action time response.

Comparative analysis and discussion:

The use of the MPC approach allows improving over the performance index from the shape-independent LMI PDC controller in two ways:

First, by allowing a larger feasible zone, in which input constraints may be hit for several time steps. The standard literature controller would only be valid in its invariant set (and, actually, published fuzzy guaranteed-cost literature would only consider a Lyapunov level set inside it).

Second, even inside the terminal set, a few steps of actual optimization will beat in most cases the worst-case cost with a fixed PDC controller structure. For instance, in this example, the state $\psi = (-2, 0)$ inside the terminal set yields a computed cost of 0.1411 with the terminal controller (note that the cost bound proved with the LMIs is $\psi^T P \psi = 34.387$ as it is a shape-independent worst-case estimation –almost 250 times higher than the actual cost–), whereas the actual cost index computed with the predictive iterative controller reduces it to 0.0916 (i.e, a 35% reduction). Random trials with states in the terminal set result in a reduction between 0% and 93% over the LMI-based PDC controller. Note also that the larger the prediction horizon the less relevant the role of the terminal cost and terminal controller is, as usual in dynamic-programming based optimal control setups.

V. Conclusions

This paper presents an application of predictive-control ideas to fuzzy control. The MPC algorithm follows an standard structure in which a fuzzy PDC terminal controller and terminal state weighting are calculated by LMIs. An algorithm for obtaining an inner approximation of the terminal set for this controller with a Polya-based approach is also introduced. As future memberships are unknown, an iterative quadratic programming procedure is proposed. Stability guarantees are also discussed.

Acknowledgment

This research is partially supported by Spanish government (research projects DPI2011-27845-C02-01 and DPI2011-27845-C02-02).

References

[1] K. Tanaka, T. Hori, and H. O. Wang, “A fuzzy lyapunov approach to fuzzy control system design,” in *American Control Conference, 2001. Proceedings of the 2001*, vol. 6. IEEE, 2001, pp. 4790–4795.

[2] H.-N. Wu, “Reliable lq fuzzy control for nonlinear discrete-time systems via lmis,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 2, pp. 1270–1275, 2004.

[3] J. L. Pitarch, C. Ariño, F. Bedate, and A. Sala, “Local fuzzy modeling: Maximising the basin of attraction,” in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–7.

[4] C. Ariño, E. Pérez, A. Sala, and F. Bedate, “Polytopic invariant and contractive sets for closed-loop discrete fuzzy systems,” *Submitted to Journal of The Franklin Institute*.

[5] Y.-Y. Cao and Z. Lin, “Robust stability analysis and fuzzy-scheduling control for nonlinear systems subject to actuator saturation,” *Fuzzy Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 57–67, 2003.

[6] C. Ariño, E. Pérez, and A. Sala, “Guaranteed cost control analysis and iterative design for constrained takagi–sugeno systems,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1420–1427, 2010.

[7] G. C. Goodwin and J. A. De Doná, *Constrained Control and Estimation: And Optimization Approach*. Springer, 2005.

[8] A. Sala, “On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems,” *Annual Reviews in Control*, vol. 33, no. 1, pp. 48–58, 2009.

[9] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *Systems, Man and Cybernetics, IEEE Transactions on*, no. 1, pp. 116–132, 1985.

[10] K. Tanaka, T. Taniguchi, and H. O. Wang, “Robust and optimal fuzzy control: A linear matrix inequality approach,” in *1999 International federation of automatic control World Congress*, 1999, pp. 213–218.

[11] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.

[12] K. Tanaka and H. O. Wang, *Fuzzy control systems design and analysis: a linear matrix inequality approach*. Wiley. com, 2004.

[13] H. D. Tuan, P. Apkarian, T. Narikiyo, and Y. Yamamoto, “Parameterized linear matrix inequality techniques in fuzzy control system design,” *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 324–332, 2001.

[14] A. Sala and C. Ariño, “Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of polya’s theorem,” *Fuzzy Sets and Systems*, vol. 158, no. 24, pp. 2671–2686, 2007.

[15] C. Arino, E. Perez, F. Bedate, and A. Sala, “Robust polytopic invariant sets for discrete fuzzy control systems,” in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–7.

[16] E. G. Gilbert and K. T. Tan, “Linear systems with state and control constraints: The theory and application of maximal output admissible sets,” *Automatic Control, IEEE Transactions on*, vol. 36, no. 9, pp. 1008–1020, 1991.

[17] S. L. de Oliveira Kothare and M. Morari, “Contractive model predictive control for constrained nonlinear systems,” *Automatic Control, IEEE Transactions on*, vol. 45, no. 6, pp. 1053–1071, 2000.