



Piecewise quadratic Lyapunov functions for stability verification of approximate explicit MPC

Morten Hovd¹ Sorin Olaru²

¹*Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail: Morten.Hovd@itk.ntnu.no*

²*SUPELEC Systems Sciences (E3S) - Automatic Control Department, Gif sur Yvette, France E-mail: Sorin.Olaru@supelec.fr*

Abstract

Explicit MPC of constrained linear systems is known to result in a piecewise affine controller and therefore also piecewise affine closed loop dynamics. The complexity of such analytic formulations of the control law can grow exponentially with the prediction horizon. The suboptimal solutions offer a trade-off in terms of complexity and several approaches can be found in the literature for the construction of approximate MPC laws. In the present paper a piecewise quadratic (PWQ) Lyapunov function is used for the stability verification of an of approximate explicit Model Predictive Control (MPC). A novel relaxation method is proposed for the LMI criteria on the Lyapunov function design. This relaxation is applicable to the design of PWQ Lyapunov functions for discrete-time piecewise affine systems in general.

Keywords: Piecewise Quadratic Lyapunov function, Linear Matrix Inequalities, Model Predictive Control, Piecewise Affine Systems

1 Introduction

Model Predictive Control (MPC) has been a great success in the process industries (Qin and Badgwell, 2003). However, due to heavy computational requirements, it has mainly been applied to relatively slow processes. Explicit MPC removes the requirement for the online solution of the MPC optimization problem (Bemporad et al., 2002), and can therefore make MPC useful for applications where fast sampling is required (Johansen and Grancharova, 2003). Explicit MPC uses multiparametric programming to partition the state space, and finds controllers that are affine in the state for each partition. These structural properties transform the closed loop dynamics into a piecewise affine system and makes the link with the class of hybrid systems. Readers wanting a more thorough introduction to multi-parametric programming and control may consult the books (Bank et al., 1983; Kvasnica, 2009).

However, as pointed out in Bemporad et al. (2002), the optimal partition of the state space may become very complex for large problems (problems with many states and/or a long prediction horizon). Approaches to simplifying the required partitioning of the state space have therefore been proposed (Bemporad and Filippi, 2003; Johansen and Grancharova, 2003; Jones et al., 2007). These approaches allow for some degree of suboptimality, and guarantee stability by ensuring that the approximate cost function is within a pre-derived bound of the exact cost function, which is designed to be a Lyapunov function.

An alternative method for verifying the stability of the approximate MPC is presented in Hovd et al. (2009), who avoid the need for the approximate cost function to be 'close' to the exact cost function, by verifying that the approximate cost function is itself a Lyapunov function. Unfortunately, the approach in Hovd et al. (2009) requires the solution of a number of optimiza-

tion problems, many of which may be non-convex. Although it is not in general necessary to find the optimal solution to these optimization problems, the approach may become excessively demanding for large systems.

In this paper we do not use the exact nor the approximate cost function as a Lyapunov function for verifying stability of the approximate MPC. However, we do take advantage of the fact that the formulation of the (exact) MPC problem does guarantee stability, and thus a Lyapunov function should exist as the approximate solution is refined to approximate the exact solution more closely.

In this work, stability is verified using Piecewise Quadratic (PWQ) Lyapunov functions. The computation of PWQ Lyapunov functions for Piecewise Affine (PWA) continuous time systems using LMI techniques was introduced in a comprehensive study by [Johansson and Rantzer \(1998\)](#); [Rantzer and Johansson \(2000\)](#). These results have since been extended to discrete-time systems ([Feng, 2002](#); [Ferrari-Trecate et al., 2002](#)).

A similar approach is used in the present work for verifying the stability of approximate explicit MPC, with the contribution of a modified LMI relaxation that has to be introduced in order to successfully apply this approach to larger problems. This novel LMI relaxation provides a useful tool for stability analysis of PWA systems in general.

It may need pointing out that explicit MPC is not a likely alternative to conventional MPC for the type of problems where conventional MPC has been successful, i.e., large problems with relatively long sampling intervals, where computer hardware costs are a minor concern, and where backup systems (possibly operators) can take over control should the MPC fail. This explains why conventional MPC is rarely critical for maintaining closed loop stability - stability instead being ensured by lower-level control loops. The focus for conventional MPC applications is instead on improving performance (in a broad sense), in particular by keeping the plant close to the optimal operating point while adhering to constraints. The complex software required for implementing conventional MPC makes it practically impossible to guarantee that the implementation will work as intended in all circumstances.

As mentioned above, the complexity of the solution for explicit MPC increases rapidly with problem size. The size of problems that can be handled will depend on what is considered an acceptable computational load at the design stage, and the available computer memory for implementation. However, even taking the current research on approximate explicit MPC into account, it is hard to foresee applications in the near future where the number of states (or, more generally, the number of 'parameters', in parametric programming parlance) go

into the double-digit range. Thus, explicit MPC should be seen as a candidate for relatively small problems, often requiring high sampling rates (sampling intervals in the milli- to microsecond range), and where computer hardware costs should be kept to a minimum or system integrity is a major concern. For problems fitting this description, there is no need for lower-level stabilizing control loops when explicit MPC is applied.

The paper is organized as follows: Section 2 introduces the MPC formulation. In Section 3 we investigate the approximate MPC characterizations, leading to the description of the resulting closed loop dynamics in Section 4. Section 5 applies the PWQ Lyapunov function to stability analysis of closed loop PWA systems. In Section 6 several numerical examples are presented, while Section 7 introduces the novel LMI relaxation technique and shows how this can be applied to refining the approximate MPC solution and prove stability of the resulting closed loop system.

2 MPC formulation

We consider a linear dynamical discrete-time system

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$

where the state vector is of length n_x , and a 'standard' quadratic QP formulation of MPC

$$\begin{aligned} \min_{\mathbf{u}=[u_0, \dots, u_{N-1}]} J(\mathbf{u}, x_k) &= \|x_{k+N}\|_P^2 \\ &+ \sum_{i=0}^{N-1} (\|x_{k+i}\|_Q^2 + \|u_{k+i}\|_R^2) \end{aligned} \quad (2)$$

subject to constraints

$$x_k = \text{measured}$$

$$u_{k+i} \in \mathbb{U} \quad (3)$$

$$x_{k+i} \in \mathbb{X} \quad (4)$$

The constraints on the inputs and states are assumed to define polytopic regions in the input and state spaces respectively, each with the origin in their interior. The constraints on the inputs are enforced from the present timestep ($i \geq 0$), whereas the constraints on the states are enforced from the next timestep ($i \geq 1$). The prediction model equation (1) is naturally assumed to hold within the prediction horizon, it is assumed that (A, B) is controllable and (\sqrt{Q}, A) observable, and the matrices $Q \succeq 0$ and $R \succ 0$ are tuning matrices. The matrix $P \succ 0$ is then the corresponding solution to the discrete algebraic Riccati equation. This solution to the

discrete algebraic Riccati equation also provides a stabilizing linear state feedback controller K . If no constraints are active, the MPC is equivalent to this linear state feedback controller.

Defining

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+N} \end{bmatrix} \\ \hat{Q} &= \text{diag}\{Q, \dots, Q, P\} \\ \hat{R} &= \text{diag}\{R, \dots, R\} \\ \hat{A} &= \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \\ \hat{B} &= \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ B & 0 & \dots & 0 & 0 \\ AB & B & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & AB & B \end{bmatrix} \end{aligned} \quad (5)$$

the optimization criterion may be reformulated as

$$J(\mathbf{u}, x_k) = \mathbf{x}^T \hat{Q} \mathbf{x} + \mathbf{u}^T \hat{R} \mathbf{u} \quad (6)$$

with x_k given. Using the identity

$$\mathbf{x} = \hat{A}x_k + \hat{B}\mathbf{u} \quad (7)$$

we get

$$J(\mathbf{u}, x_k) = \|\mathbf{u}\|_H^2 + 2x_k^T F \mathbf{u} + \|x_k\|_Y^2 \quad (8)$$

subject to constraints

$$\begin{aligned} x_k &= \text{given} \\ G\mathbf{u} &\leq W - Ex_k \end{aligned} \quad (9)$$

where $H = \hat{B}^T \hat{Q} \hat{B} + \hat{R}$, $F = \hat{A}^T \hat{Q} \hat{B}$ and $Y = \hat{A}^T \hat{Q} \hat{A}$. Note that $H \succ 0$ follows from $R \succ 0$.

The region \mathbb{X}_F of the state space for which there exists an input sequence \mathbf{u} fulfilling (9), i.e., the feasible region, can be found by projecting the constraints (9) onto the states x . Similarly, using the theory in Gilbert and Tan (1991), it is possible to describe a region in the neighborhood of the origin (the so-called 'maximal output admissible set (MOAS)', \mathbb{M}_0), for which the linear state feedback controller K fulfills all constraints and which is positive invariant for the system

(1) when the controller K is applied. We therefore include in (9) not only the constraints resulting directly from (3) and (4), but add the constraints $x_{k+N} \in \mathbb{M}_0$ (see (Scibilia et al., 2009) for further details of the MPC formulation). It is noted above that the MPC controller and the state feedback controller K are equivalent when no constraints are active. Adding the constraints $x_{k+N} \in \mathbb{M}_0$ therefore ensures that the system with the MPC controller is closed loop stable provided the initial state is feasible.

3 Approximate MPC

Bemporad and Filippi (2006) proposed an algorithm for approximate multiparametric convex programming which partitions the feasible region \mathbb{X}_F (or whatever subset thereof they are exploring) into simplices. Let v_i be the $n_x + 1$ vertices of a simplex, and $\mathbf{u}(v_i)$ be the optimal solution to the MPC problem ((8)-(9)) at vertex i .

Given a state x_0 inside a simplex, define the matrix

$$M = \begin{bmatrix} v_1 & \dots & v_{n_x+1} \\ 1 & \dots & 1 \end{bmatrix}$$

The state x_0 can be expressed as a linear interpolation between the vertices of the simplex as

$$x_0 = v_1 \lambda_1 + \dots + v_{n_x+1} \lambda_{n_x+1}$$

where the interpolation parameters λ_i are found from

$$\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{n_x+1} \end{bmatrix} = M^{-1} \begin{bmatrix} x_0 \\ 1 \end{bmatrix} \quad (10)$$

Note that the matrix M is by construction non-singular and hence invertible, provided the points v_1, \dots, v_{n_x+1} define a full-dimensional simplex. Equation (10) and the definition of M ensure that the λ_i 's sum to one, and they will also be non-negative provided x_0 is inside the simplex. From (10) observe that the interpolation parameters can be split into a state-dependent part and a state-independent part. That is

$$\lambda = \lambda_x x_0 + \lambda_0 \quad (11)$$

where λ_x consists of the n_x leading columns of M^{-1} and λ_0 is the last column. Define the matrix

$$U = [\mathbf{u}(v_1) \quad \dots \quad \mathbf{u}(v_{n_x+1})]$$

and let \mathbf{u}_a be the approximately-optimal input sequence obtained by linear interpolation between the solutions at the vertices of the simplex. This gives

$$\mathbf{u}_a = U \lambda = K_a x_0 + k_a \quad (12)$$

where K_a and k_a are constant within a given simplex. Interpolating between the solutions at the vertices of the simplex we therefore obtain an affine state feedback controller that is valid for the whole simplex. In [Bemporad and Filippi \(2006\)](#) it is shown that this input is a feasible (but obviously not in general optimal) solution to the MPC problem at the state x_0 .

It clearly makes sense to use the controller K inside \mathbb{M}_0 , where it is known to be optimal. Instead of calculating the Delaunay tessellation (or triangulation in the 2-dimensional case) of \mathbb{X}_F , we instead calculate the tessellation for $\mathbb{X}_F \setminus \mathbb{M}_0$. Here, the ‘\’ denotes the set difference. That is, the set $\mathbb{X}_F \setminus \mathbb{M}_0$ is the part of the set \mathbb{X}_F that is *not* in the set \mathbb{M}_0 . However, since \mathbb{M}_0 is in the interior of \mathbb{X}_F , the set difference is clearly non-convex. This non-convex set is therefore partitioned into convex parts using the procedure in [\(Scibilia et al., 2009\)](#), and each convex part is subsequently tessellated.

We note also that by using arbitrary many points in the tessellation (not only on the extreme points of the MOAS and the feasible region, but also at points in $\mathbb{X}_F \setminus \mathbb{M}_0$ in between), we can make both the approximate MPC and the approximate cost function arbitrarily close to the optimal control and optimal cost function.

In the following, we will assume that the feasible region X_F (or whatever subset thereof for which we are interested in designing an MPC controller) is known, together with an initial partition thereof into non-overlapping, convex regions X_i . The index value $i = 0$ is used for the region \mathbb{M}_0 , whereas $i > 0$ identifies a region different from \mathbb{M}_0 .

4 Piecewise Linear Closed Loop Dynamics

Although the MPC solution provides the predicted input sequence for the entire horizon of the MPC controller, only the first block of this sequence is actually applied at each timestep. For online implementation of explicit MPC, only the first block of K_a and k_a for each region of the tessellation therefore need to be saved. Thus, for region i we have

$$\begin{aligned} K_i &= K_{a,i}(1 : n_u, 1 : n_x) \\ k_i &= k_{a,i}(1 : n_u) \end{aligned} \quad (13)$$

In the unconstrained region \mathbb{M}_0 , the MPC controller is equivalent to the LQR controller K , and we thus get

$$\begin{aligned} K_{\mathbb{M}_0} &= K \\ k_{\mathbb{M}_0} &= 0 \end{aligned}$$

Combining the linear model in [\(1\)](#) and the PWA controller in [\(13\)](#) we obtain the PWA closed loop dynamics

$$x_{k+1} = (A + BK_i)x_k + k_i \quad (14)$$

Taking into account that the general formulation of the closed loop dynamics is affine, following [\(Rantzer and Johansson, 2000\)](#) we introduce a lifting transformation to a homogenous space of dimension $n_x + 1$. Thus, we introduce

$$\bar{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}; \quad x \notin \mathbb{M}_0 \quad (15)$$

The region \mathbb{M}_0 is positively invariant, and we therefore define

$$\bar{x} = \begin{bmatrix} x \\ 0 \end{bmatrix}; \quad x \in \mathbb{M}_0 \quad (16)$$

This leads to the following closed loop dynamics

$$\bar{x}_{k+1} = \bar{A}_i \bar{x}_k \quad (17)$$

$$\bar{A}_i = \begin{bmatrix} (A + BK_i) & k_i \\ 0 & 1 \end{bmatrix}; \quad i > 0 \quad (18)$$

$$\bar{A}_i = \begin{bmatrix} (A + BK) & 0 \\ 0 & 0 \end{bmatrix}; \quad i = 0 \quad (19)$$

We also have the case when $x_k \notin \mathbb{M}_0$ while $x_{k+1} \in \mathbb{M}_0$. Such regions are easily identifiable. Strictly speaking, we should for such regions use

$$\bar{A}_{i0} = \begin{bmatrix} (A + BK_i) & k_i \\ 0 & 0 \end{bmatrix}; \quad x_k \notin \mathbb{M}_0, x_{k+1} \in \mathbb{M}_0 \quad (20)$$

However, ignoring this and using \bar{A}_i as defined in [\(18 - 19\)](#) merely means that the last element of \bar{x} is set to zero one timestep delayed. In most cases, the conservatism thus introduced is negligible.

5 Stability Analysis using PWQ Lyapunov Functions

Following the development in [Feng \(2002\)](#), we consider PWQ Lyapunov functions. Thus, for each region i of the state space, the Lyapunov function $V(x)$ takes the value of $V_i(x)$, defined as follows:

$$V_i(x) = \bar{x}^T \bar{P}_i \bar{x}; \quad x \in X_i \quad (21)$$

The closed loop system is then stable provided

$$\bar{P}_i = \bar{P}_i^T \quad (22)$$

$$V_i(x) > 0 \quad \forall x \in X_i \quad (23)$$

$$V(x_k) > V(x_{k+1}) \quad (24)$$

where [\(24\)](#) should hold for all x_k and x_{k+1} that may occur according to the closed loop dynamics in [\(17 -](#)

19). Let the index i identify the region in which the state is at time k , and the index j identify the region in which the state is at time $k+1$. All possible transitions between regions may then be identified with a set \mathcal{S} , where each element of \mathcal{S} consists of a pair (i, j) that is consistent with the partition of the state space and the closed loop dynamics (17 - 19).

5.1 Expressing the stability conditions as an LMI

In this work, the Matlab toolbox YALMIP (Löfberg, 2004) is used. Using YALMIP, (22) is fulfilled simply by defining \bar{P}_i to be a symmetric (matrix valued) variable.

The conditions (23) and (24) will be fulfilled if the following LMIs are fulfilled:

$$\bar{P}_i \succ 0; \quad \forall i \quad (25)$$

$$\bar{A}_i^T \bar{P}_j \bar{A}_i - \bar{P}_i \prec 0; \quad \forall (i, j) \in \mathcal{S} \quad (26)$$

However, the condition (23) only has to hold for $x \in X_i$, whereas fulfilling (25) means that it is fulfilled for the entire extended state space (i.e., for all \bar{x}). Similarly, condition (24) only has to hold for $x_k \in X_{ij}$, where $X_{ij} \in X_i$ is the subregion of region X_i for which the state moves to X_j in the next timestep. Clearly, X_{ij} is a polytope, since both X_i and X_k are polytopes and the closed loop dynamics is piecewise affine. In contrast, (26) has to hold for the entire extended state space.

5.2 Relaxing the LMI conditions

Clearly, some way of relaxing the LMI conditions is desirable, to reduce the conservatism resulting from taking conditions on the Lyapunov function that must be fulfilled only in specific regions of the state space, and converting these conditions into LMIs that by default imply that the conditions are fulfilled for the entire state space.

To this end, let us introduce the quadratic functions

$$f_i(x) = \bar{x}^T F_i \bar{x}; \quad f_i(x) > 0, \forall x \in X_i \quad (27)$$

Note that $f_i(x) < 0$ is allowed for $x \notin X_i$. It is then easy to see that (23) is fulfilled, provided

$$\bar{P}_i - F_i \succ 0 \quad (28)$$

Similarly, we introduce functions

$$g_{ij}(x) = \bar{x}^T G_{ij} \bar{x}; \quad g_{ij}(x) > 0, \forall x \in X_{ij} \quad (29)$$

Then, (24) is fulfilled provided

$$\bar{A}_i^T \bar{P}_j \bar{A}_i - \bar{P}_i + G_{ij} \prec 0 \quad (30)$$

Let the region X_i be defined by

$$E_i x \geq e_i \Leftrightarrow \bar{E}_i \bar{x} \geq 0 \quad (31)$$

where $\bar{E}_i = \begin{bmatrix} E_i & -e_i \end{bmatrix}$. Similarly, the region X_{ij} is defined by $\bar{E}_{ij} \bar{x} \geq 0$. The relaxations proposed in (Rantzer and Johansson, 2000) for continuous-time dynamics are then given by

$$F_i = \bar{E}_i^T U_i \bar{E}_i \quad (32)$$

$$G_{ij} = \bar{E}_{ij}^T W_{ij} \bar{E}_{ij} \quad (33)$$

where U_i and W_{ij} are symmetric, non-negative matrices. It appears that the same type of relaxations have been used for discrete-time dynamics by other authors, e.g. (Feng, 2002; Ferrari-Trecate et al., 2002).

6 Examples of PWQ Lyapunov functions using relaxed LMI conditions

The approach described above will next be illustrated on a few examples.

6.1 Example 1

Consider first the simple one-state example in Hovd et al. (2009). The system is open loop unstable

$$x_{k+1} = 1.1x_k + u_k \quad (34)$$

and the input is constrained $-2 \leq u_k \leq 2$. We consider a prediction horizon of 5, and the weights $Q = 10$, $R = 0.1$. The LQR controller is $u = -1.089x$. The corresponding unconstrained region \mathbb{M}_0 is given by $-1.836 < x < 1.836$, while the feasible region X_F is given by $-8.722 < x < 8.722$ (for the present choice of prediction horizon). The initial partition for the approximate solution has three regions, and the local closed loop dynamics for each of these regions are:

$$\bar{A}_0 = \begin{bmatrix} 0.011 & 0 \\ 0 & 0 \end{bmatrix} \text{ for } R_0 = \{-1.836 \leq x_k \leq 1.836\}$$

$$\bar{A}_1 = \begin{bmatrix} 1.1 & -2 \\ 0 & 1 \end{bmatrix} \text{ for } R_1 = \{1.836 \leq x_k\}$$

$$\bar{A}_2 = \begin{bmatrix} 1.1 & 2 \\ 0 & 1 \end{bmatrix} \text{ for } R_2 = \{x_k \leq -1.836\}$$

Using the LMI approach to find a PWQ Lyapunov function, we find that the system is stable in closed loop, although both region 1 and region 2 have unstable local dynamics. In fact, it is easy with the LMI approach to show stability for a larger region than the

feasible region for the original MPC problem. Simple inspection will show that the closed loop is stable for $-20 < x < 20$. The LMI approach can prove stability for nearly the same region of the state space - numerical problems occur when including states very close to ± 20 .

The importance of relaxing the LMI stability criteria can be illustrated by Fig. 1. The figure shows the relaxed and unrelaxed solutions to the Lyapunov function stability criterion for states originating in region 1 and staying in region 1 at the next timestep. This corresponds to states $x > 3.487$. It can be observed that although the unrelaxed solution fulfills the criterion (24) for the relevant part of the state space, the LMI condition (26) fails to hold globally. In contrast, the relaxed solution fulfills (30) over the entire state space. The relaxation makes the criterion (30) harder to fulfill (reduces the decrease of the Lyapunov function) *inside* the region in question (here, the region $3.487 < x < 20$) while it makes the criterion easier to fulfill outside the region. We are using quadratic Lyapunov functions. Hence, since the unrelaxed solution fulfills (24) locally, but fails to hold globally, it is necessarily convex. In contrast, the relaxed solution has to be concave in order to hold globally.

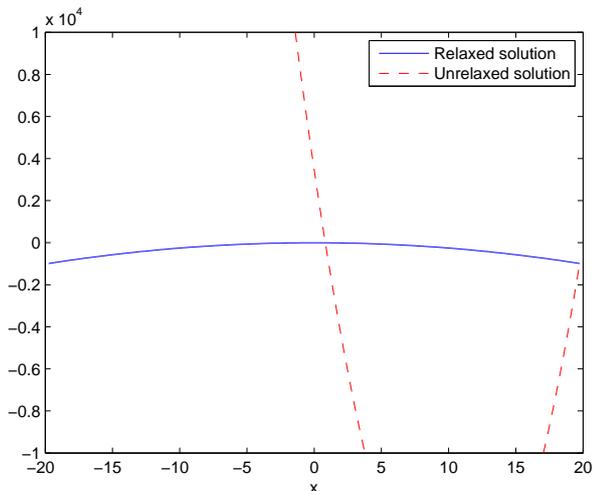


Figure 1: Relaxed and unrelaxed Lyapunov function stability criteria for Example 1, for states originating in region 1 and staying in region 1. The unrelaxed Lyapunov function is given by $\bar{x}^T (\bar{A}_1^T \bar{P}_1 \bar{A}_1 - \bar{P}_1) \bar{x}$ (the LHS of (26)), while the relaxed Lyapunov function is $\bar{x}^T (\bar{A}_1^T \bar{P}_1 \bar{A}_1 - \bar{P}_1 + G_{11}) \bar{x}$ (the LHS of (30)). The criteria are here evaluated for $\bar{x} = [x \ 1]^T$. The unrelaxed solution holds for the required region $x > 3.487$, while the relaxed solution holds globally.

It should be mentioned that the exact explicit MPC

solution for this problem (with a prediction horizon of 5) has 11 regions for the feasible region $-8.722 \leq x \leq 8.722$. Extending the exact solution to give a larger feasible region will require significantly increasing the prediction horizon and correspondingly increasing the number of regions for the size of the feasible region to approach the maximal stabilizable region. Subsequent postprocessing of the exact solution allows reducing the number of regions to 3.

6.2 Example 2

We consider the same system as the one studied in (Scibilia et al., 2009; Hovd et al., 2009).

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

with constraints

$$\begin{aligned} -2 &\leq u_k \leq 2 \\ \begin{bmatrix} -5 \\ -5 \end{bmatrix} &\leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix} \end{aligned}$$

The weight matrices used are $Q = I$ and $R = 1$, whereas the prediction horizon $n = 15$ is used. In (Scibilia et al., 2009) it is found that the exact solution requires 101 regions. Proceeding with the approximate MPC, we obtain an initial partition of the state space with 33 regions (including \mathbb{M}_0). Merging regions where the controllers are identical, we can reduce the number of regions to 17. Using the LMI approach, closed loop stability is proven.

In order to compare this direct result with previous approaches, we mention that starting from the initial tessellation with 33 regions, and using the approximate cost function as a Lyapunov function, it was found in Hovd et al. (2009) that the tessellation had to be further refined, ending up with 53 regions in order to prove stability. Postprocessing the resulting solution allowed reducing the number of regions to 17, resulting in the same solution that was proven stable in a more direct way using the LMI approach. Finally, it should be mentioned that for the exact explicit MPC, we were only able to reduce the number of regions to 53 by post-processing the solution.

6.3 Example 3

This example is a slight modification of an example in (Hovd and Braatz, 2001). The system is described by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned}$$

with

$$\begin{aligned}
 A &= \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
 B &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
 C &= \begin{bmatrix} -1 & 0 & 2 \end{bmatrix}
 \end{aligned}$$

The input constraints are given by $-2 \leq u_k \leq 2$, whereas the output constraints are $-1 \leq y_k \leq 1$. The MPC prediction horizon is 10, and the weights are $Q = C^T * C$ and $R = 2$. Note that the plant has transmission zeros at $z = \pm 1.4142$, and these also enter the MPC problem (despite assuming state feedback) through the particular choice of state weight and through the output constraint.

The initial tessellation of the state space results in 199 regions. Through merging, the number of regions can be reduced to 147. After merging, there are 1478 transitions between regions (including 'transitions' where $i = j$). Using the LMI approach described above fails for this case. The LMI constraint (28) fails for 86 regions and the constraint (30) fails for 1353 transitions between regions. Further refining the partitioning of the state space seems futile, given that the LMI conditions fail for such a high number of regions and transitions between regions. This result shows the limitations of the stability analysis applied. However, if an improved relaxation of the LMI conditions can be found, the LMI approach may nevertheless be of use for the present example. The next section describes the construction of such an improved relaxation in a general PWA framework.

7 A novel LMI relaxation

7.1 Construction of an improved LMI relaxation

From (28) and (30) it is clear that we only need the relaxations ($f_i(x)$ and $g_{ij}(x)$) to be positive within specific polytopes. Outside those polytopes the functions may be negative, and may thereby make it easier to find a valid solution to the LMIs. The relaxations (32) and (33) do fulfill these requirements, and have proven effective for problems of modest size. However, the resulting relaxation functions are somewhat arbitrary, and there is a possibility that more careful specification of the functional form of the relaxations can be beneficial.

A reasonable choice for a relaxation would seem to be a concave quadratic function centered in a point

contained in the interior of the polytope considered. Clearly, the quadratic function should be positive over the polytope. Thus, a reasonable relaxation function would appear to be

$$h(x) = (x - x_0)^T H (x - x_0) + c \quad (35)$$

where H is a symmetric negative definite matrix and c is a scalar that is sufficiently large to make h positive for the entire polytope. Clearly, the relaxation function reaches its maximum at (is 'centered on') $x = x_0$. The function $h(x)$ in (35) may equivalently be expressed as

$$h(x) = \bar{x}^T \begin{bmatrix} H & -Hx_0 \\ -x_0^T H & x_0^T H x_0 + c \end{bmatrix} \bar{x} = \bar{x}^T [\bar{H}_i + \bar{C}] \bar{x} \quad (36)$$

with

$$\begin{aligned}
 \bar{H} &= \begin{bmatrix} H & -Hx_0 \\ -x_0^T H & x_0^T H x_0 \end{bmatrix} \\
 \bar{C} &= \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix}
 \end{aligned}$$

Thus, \bar{H} can be chosen as any symmetric negative definite matrix, provided we add the additional constraints that $h(x) \geq 0$ at all the vertices of the polytope in question.

Remark: From the explanation above, it follows that it is actually only H that should be negative definite. The Scur complement of \bar{H} is

$$x_0^T H x_0 - x_0^T H (H)^{-1} H x_0 = 0$$

and thus \bar{H} should be negative *semi*-definite. However, a simple reformulation of (36) gives

$$h(x) = \bar{x}^T [\bar{H} + \bar{C}] \bar{x} = \bar{x}^T [(\bar{H} + \bar{C}_a) + \bar{C}_b] \bar{x} \quad (37)$$

Thus, we can 'move part of the constant c into \bar{H} ', to make $\bar{H} + \bar{C}_{i,a}$ semi-definite. The non-negativity at the vertices and the concavity of the relaxation function then ensures that the relaxation is of the correct form.

7.2 Example 3 revisited

Next, we consider again example 3, with the alternative relaxation formulation. Of the 147 regions and 1478 transitions between regions, the relaxed solution now fails to fulfill the criteria only for 4 transitions between regions. Closer inspection shows that these 'transitions' actually represent the state staying in the same region, i.e., 'transitions' where $i = j$. Inspecting the four regions in question, it turns out that each of them has a fixed point inside the region. This happens even though the input in within each region corresponds to linear interpolation between the inputs at

the vertices of the (simplicial) region, and the input at each vertex being the optimal input for an MPC formulation with guaranteed closed loop stability. With the new relaxations attention is thus effectively focused on the regions where the control was inadequate.

It is next observed that the fixed points occur pairwise in adjacent regions. For each pair of adjacent regions with fixed points, an additional point for tessellation was added on the interface between the regions, on the line connecting the fixed points. This resulted in a total of 155 regions and 1768 transitions between regions. With this refined tessellation, the closed loop system is found to be stable.

For comparison, the (exact) optimal solution contains 472 regions. Merging regions with identical controllers, the number of regions could be reduced to 274. In Fig. 2 the closed loop behavior of the optimal and approximate controllers are shown, both starting from the point $x_0 = [0.98 \ 0.98 \ 0.98]^T$. It can be seen that both controllers respect the constraint $y \leq 1$, and the difference in performance is minor.

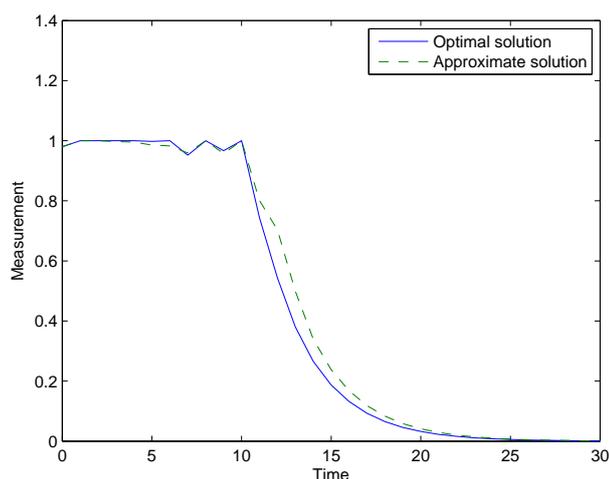


Figure 2: Comparison of the response with the optimal and approximate controller for Example 3.

8 Conclusions

In this paper, a PWQ Lyapunov function is used to prove the stability of approximate explicit MPC. The PWQ Lyapunov functions are calculated using LMIs. It is illustrated how the approximate solution can be refined for regions where the LMI conditions are not fulfilled. A novel LMI relaxation is proposed, and is shown to be effective for a large and complex case that could not be handled with the conventional relaxation. This new relaxation is applicable to LMI-based stability analysis of PWA systems in general.

References

- Bank, M., Guddat, J., Klatte, D., Kummer, B., and Tammer, K. *Non-linear Parametric Optimization*. Birkhauser, Stuttgart, Germany, 1983.
- Bemporad, A. and Filippi, C. Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *Journal of optimization theory and applications*, 2003. 117(1):9–38. doi:[10.1023/A:1023696221899](https://doi.org/10.1023/A:1023696221899).
- Bemporad, A. and Filippi, C. An algorithm for approximate multiparametric convex programming. *Computational optimization and applications*, 2006. 35:87–108. doi:[10.1007/s10589-006-6447-z](https://doi.org/10.1007/s10589-006-6447-z).
- Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N. The explicit linear quadratic regulator for constrained systems. *Automatica*, 2002. 38:3–20. doi:[10.1016/S0005-1098\(01\)00174-1](https://doi.org/10.1016/S0005-1098(01)00174-1).
- Feng, G. Stability analysis of piecewise discrete-time linear systems. *IEEE Trans. Autom. Contr.*, 2002. 47:1108–1112. doi:[10.1109/TAC.2002.800666](https://doi.org/10.1109/TAC.2002.800666).
- Ferrari-Trecate, G., Cuzzola, F. A., Mignone, D., and Morari, M. Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 2002. 38:2139–2146. doi:[10.1016/S0005-1098\(02\)00142-5](https://doi.org/10.1016/S0005-1098(02)00142-5)
- Gilbert, E. and Tan, K. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. Autom. Contr.*, 1991. 36:1008–1020. doi:[10.1109/9.83532](https://doi.org/10.1109/9.83532)
- Hovd, M. and Braatz, R. D. On the use of soft constraints in mpc controllers for plants with inverse response. In *Preprints Dycops*. Jejudo Island, Korea, pages 295–300, 2001.
- Hovd, M., Scibilia, F., Maciejowski, J. M., and Oлару, S. Verifying stability of approximate explicit MPC. In *Conference on Decision and Control*. Shanghai, China, 2009. doi:[10.1109/CDC.2009.5400788](https://doi.org/10.1109/CDC.2009.5400788)
- Johansen, T. A. and Grancharova, A. Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 2003. 48:810–815. doi:[10.1109/TAC.2003.811259](https://doi.org/10.1109/TAC.2003.811259)
- Johansson, M. and Rantzer, A. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 1998. 43:555–559. doi:[10.1109/9.664157](https://doi.org/10.1109/9.664157)

- Jones, C. N., Baric, M., and Morari, M. Multiparametric linear programming with application to control. *European Control Journal*, 2007. 13:152–170. doi:[10.3166/ejc.13.152-170](https://doi.org/10.3166/ejc.13.152-170)
- Kvasnica, M. *Real-time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*. VDM Verlag, Saarbruecken, Germany, 2009.
- Löfberg, J. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004. URL <http://control.ee.ethz.ch/~joloef/yalmip.php>.
- Qin, S. J. and Badgwell, T. A. A survey of industrial model predictive control technology. *Control Engineering Practice*, 2003. pages 733–764. doi:[10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7)
- Rantzer, A. and Johansson, M. Piecewise linear quadratic optimal control. *IEEE Transactions on Automatic Control*, 2000. 45:629–637. doi:[10.1109/9.847100](https://doi.org/10.1109/9.847100)
- Scibilia, F., Olaru, S., and Hovd, M. Approximate explicit linear MPC via Delaunay tessellation. In *European Control Conference*. Budapest, Hungary, 2009.