Computational Aspects of Approximate Explicit Nonlinear Model Predictive Control

Alexandra Grancharova*,**, Tor A. Johansen* and Petter TØndel*
* *Department of Engineering Cybernetics*
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
Tor.Arne.Johansen@itk.ntnu.no, Petter.Tondel@itk.ntnu.no
** *Institute of Control and System Research*
Bulgarian Academy of Sciences, P.O.Box 79, Sofia 1113, Bulgaria
alexandra@icsr.bas.bg

August 4, 2005

Keywords: Nonlinear model predictive control, multi-parametric Nonlinear Programming, piecewise linear controllers.

Abstract: It has recently been shown that the feedback solution to linear and quadratic constrained Model Predictive Control (MPC) problems has an explicit representation as a piecewise linear (PWL) state feedback. For nonlinear MPC the prospects of explicit solutions are even higher than for linear MPC, since the benefits of computational efficiency and verifiability are even more important. Preliminary studies on approximate explicit PWL solutions of convex nonlinear MPC problems, based on multi-parametric Nonlinear Programming (mp-NLP) ideas show that sub-optimal PWL controllers of practical complexity can indeed be computed off-line. However, for non-convex problems there is a need to investigate practical computational methods that not necessarily lead to guaranteed properties, but when combined with verification and analysis methods will give a practical tool for development and implementation of explicit NMPC. The present paper focuses on the development of such methods. As a case study, the application of the developed approaches to compressor surge control is considered.

1 Introduction

Nonlinear Model Predictive Control (MPC) involves the solution at each sampling instant of a finite horizon optimal control problem subject to nonlinear system dynamics and state and input constraints [1] – [5]. A recent survey of the main on-line optimization strategies of Nonlinear MPC (NMPC) is given in [6]. A novel approach for NMPC design for input-affine nonlinear systems is suggested in [7], which deploys state space partitioning and graph theory to retain the on-line computational efficiency.

It has recently been shown that the feedback solution to linear and quadratic constrained MPC problems has an explicit representation as a piecewise linear (PWL) state feedback defined on a polyhedral partition of the state space [8]. The benefits of an explicit solution, in addition to the efficient on-line computations, include also verifiability of the implementation, which is an essential issue in safety-critical applications. For nonlinear MPC the prospects of explicit solutions are even higher than for linear MPC, since the benefits of computational efficiency and verifiability are even more important. Preliminary studies on approximate explicit PWL NMPC solutions [9], [10], [11], based on multi-parametric Nonlinear Programming (mp-NLP) ideas [12], show that sub-optimal PWL controllers of practical complexity can indeed be computed off-line. In the case of convex problems, it is straightforward to impose tolerances on the level of approximation such that theoretical properties like asymptotic stability of the sub-optimal feedback controller can be ensured [10], [13]. However, for non-convex problem there is a need to investigate practical computational methods that not necessarily lead to guaranteed properties, but when combined with verification and analysis methods will give a practical tool for development and implementation of explicit NMPC.

The present paper focuses on computational and implementation aspects of explicit NMPC and is structured as follows. In section 2, the formulation of the NMPC problem is given. In section 3, computational methods for approximate explicit NMPC are suggested. The application of the developed approaches to compressor surge control is considered in section 4.

2 Formulation of nonlinear model predictive control problem

Consider the discrete-time nonlinear system:

$$x(t+1) = f(x(t), u(t)) \tag{1}$$
$$y(t) = Cx(t) \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input and output variable. It is also assumed that the function f is sufficiently smooth. It is supposed that a full measurement of the state $x(t)$ is available at the current time t. For the current $x(t)$, MPC solves the following optimization problem:

$$V^*(x(t)) = \min_U J(U, x(t)) \tag{3}$$

subject to $x_{t|t} = x(t)$ and:

$$y_{\min} \leq y_{t+k|t} \leq y_{\max},\ \ k = 1, ..., N \tag{4}$$
$$u_{\min} \leq u_{t+k} \leq u_{\max},\ \ k = 0, 1, ..., N-1 \tag{5}$$
$$x^T_{t+N|t} x_{t+N|t} \leq \delta \tag{6}$$
$$x_{t+k+1|t} = f(x_{t+k|t}, u_{t+k}),\ \ k \geq 0 \tag{7}$$
$$y_{t+k|t} = Cx_{t+k|t},\ \ k \geq 0 \tag{8}$$

with $U = \{u_t, u_{t+1}, ..., u_{t+N-1}\}$ and the cost function given by:

$$J(U, x(t)) = \sum_{k=0}^{N-1} \left[x^T_{t+k|t} Q x_{t+k|t} + u^T_{t+k} R u_{t+k} \right] + x^T_{t+N|t} P x_{t+N|t} \tag{9}$$

Here, N is a finite horizon. From a stability point of view it is desirable to choose δ in (6) as small as possible [14]. If the system is asymptotically stable (or pre-stabilized) and N is large, then it is more likely that the choice of a small δ will be possible.

The following assumptions are made:

A1. $P, Q, R \succ 0$.

A2. $y_{\min} < 0 < y_{\max}$.

A3. There exists $u_{st} \in \mathbb{R}^m$ satisfying $u_{\min} \leq u_{st} \leq u_{\max}$, and such that $f(0, u_{st}) = 0$.

Assumption A3 means that the point $x = 0$, $u = u_{st}$, is a steady state point for system (1). The optimization problem can be formulated in a compact form as follows:

$$V^*(x(t)) = \min_U J(U, x(t)) \tag{10}$$

subject to:

$$G(U, x(t)) \leq 0 \tag{11}$$

This MPC problem defines an mp-NLP, since it is NLP in U parameterized by $x(t)$. An optimal solution to this problem is denoted $U^* = \left\{u_t^*, u_{t+1}^*, ..., u_{t+N-1}^*\right\}$ and the control input is chosen according to the receding horizon policy $u(t) = u_t^*$. Define the set of N-step feasible initial states as follows:

$$X_f = \{x \in \mathbb{R}^n \mid G(U, x) \leq 0 \text{ for some } U \in \mathbb{R}^{Nm}\} \tag{12}$$

If assumption A3 is satisfied and δ in (6) is chosen such that the problem (3)–(9) is feasible, then X_f is a non-empty set. Then, due to assumption A2, the origin is an interior point in X_f.

In parametric programming problems one seeks the solution $U^*(x)$ as an explicit function of the parameters x in some set $X \subseteq X_f \subseteq \mathbb{R}^n$ [12]. The explicit solution allows us to replace the computationally expensive real-time optimization with a simple function evaluation. However, for general nonlinear functions J and G an exact explicit solution can not be found. In this paper we suggest practical computational methods for constructing an explicit approximate PWL solution of general non-convex nonlinear MPC problems. They can be considered as a further extension of the method proposed in [10] where the NMPC problem was assumed to be convex.

3 Computational aspects of approximate explicit nonlinear model predictive control

3.1 Close-to-global solution of mp-NLPs

In general, the cost function J can be non-convex with multiple local minima. Therefore, it would be necessary to apply an efficient initialization of the mp-NLP problem (10)-(11) so to find a close-to-global solution. One possible way to obtain this is to find a close-to-global solution at a point $w_0 \in X$ by comparing the local minima corresponding to several initial guesses and then to use this solution as an initial guess at the neighbouring points $w_i \in X$, $i = 1, 2, ..., l$, i.e. to propagate the solution. This is described in the following procedure:

Procedure 1 (close-to-global solution of mp-NLP):

Consider any hyper-rectangle $X_0 \subseteq X_f$ with vertices $\Theta^0 = \left\{\theta_1^0, \theta_2^0, ..., \theta_M^0\right\}$ and center point w_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ with vertices respectively $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$, $j = 1, 2, ..., N_j$. Suppose $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$. For each of the *hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$, determine a set of points that belongs to its facets and denote this set $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$, $j = 0, 1, 2, ..., N_j$. Define the set of all points $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$, where $w_i \in \left\{\bigcup_{j=0}^{N_j} \Theta^j\right\} \cup \left\{\bigcup_{j=0}^{N_j} \Psi^j\right\}$, $i = 1, 2, ..., N_1$. Then:*

a). *Determine a close-to-global solution of the NLP (10)-(11) at the center point w_0 through the following minimization:*

$$U^*(w_0) = \arg \min_{U_i^{local} \in \left\{U_1^{local}, U_2^{local}, ..., U_{N_U}^{local}\right\}} J(U_i^{local}, w_0) \tag{13}$$

where U_i^{local}, $i = 1, 2, ..., N_{N_U}$ correspond to local minima of the cost function $J(U, w_0)$ obtained for a number of initial guesses U_i^0, $i = 1, 2, ..., N_{N_U}$

b). *Determine a close-to-global solution of the NLP (10)-(11) at the points $w_i \in W$, $i = 1, 2, ..., N_1$ in the following way:*

1. Determine a close-to-global solution of the NLP (10)-(11) at the center point w_0 by solving problem (13). Let $i = 1$.

2. Let $W^s = \{w_0, w_1, w_2, ..., w_{N_2}\} \subset W$ be the subset of points at which a feasible solution of the NLP (10)-(11) has been already determined.

3. Find the point $\widetilde{w} \in W^s$ that is most close to the point w_i, i.e. $\widetilde{w} = \arg \min_{w \in W^s} \|w - w_i\|$. Let the solution at $\widetilde{w}$ be $U^*(\widetilde{w})$.

4. Solve the NLP (10)-(11) at the point w_i with initial guess for the optimization variables set to $U^*(\widetilde{w})$.

5. If a solution of the NLP (10)-(11) at the point w_i has been found, mark w_i as feasible and add it to the set W^s. Otherwise, mark w_i as infeasible.

6. Let $i = i + 1$. If $i \leq N_1$, go to step 2. Otherwise, terminate.

□

Procedure 1 is illustrated on Fig.1. First, a close-to-global solution to the NLP (10)-(11) is determined at the center point w_0 of the hyper-rectangle X_0 (the case when no feasible solution at the center point w_0 exists is discussed in section 3.4). Then, this solution is used as an initial guess when solving the NLP at the points $w_1, w_2, ..., w_8$ which represent the vertices and the facets centers of the smallest interior hyper-rectangle X_0^1. Then, the solutions at these points are used as initial guesses when solving the NLP at the points $w_9, w_{10}, ..., w_{16}$ which are the vertices and the facets centers of the interior hyper-rectangle X_0^2. Next, the solutions at these points are used as initial guesses when solving the NLP at the points $w_{17}, w_{18}, ..., w_{24}$ which represent the vertices and the facets centers of the interior hyper-rectangle X_0^3. At the end, the solutions at these points are used as initial guesses when solving the NLP at the points $w_{25}, w_{26}, ..., w_{32}$ which are the vertices and the facets centers of the hyper-rectangle X_0.

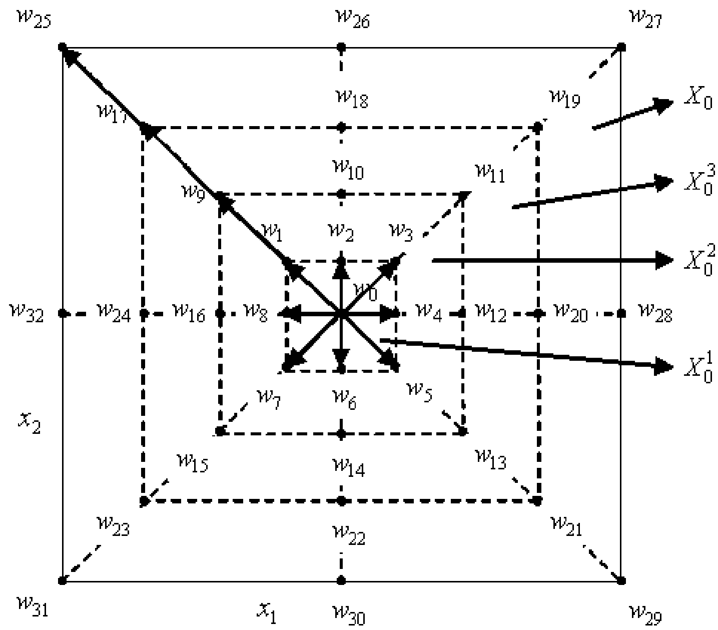

Fig. 1. Illustration of Procedure 1.

3.2 Computation of feasible approximate solution

Definition 1:

Let $X = \{w_1, w_2, ..., w_L\} \subset \mathbb{R}^n$ be a discrete set. A function $U(x)$ is feasible on X if $G(U(w_i), w_i) \leq 0$, $i \in \{1, 2, ..., L\}$.

We restrict our attention to a hyper-rectangle $X \subset \mathbb{R}^n$ where we seek to approximate the optimal solution $U^*(x)$ to the mp-NLP (10)-(11). We require that the state space partition is orthogonal and can be represented as a $k-d$ tree [15], [16]. The main idea of the approximate mp-NLP approach is to construct a feasible piecewise linear (PWL) approximation $\widehat{U}(x)$ to $U^*(x)$ on X, where the constituent affine functions are defined on hyper-rectangles covering X. In case of convexity, it suffices to compute the solution of problem (10)-(11) at the 2^n vertices of a considered hyper-rectangle X_0 by solving up to 2^n NLPs. In case of non-convexity, it would not be sufficient to impose the constraints only at the vertices of the hyper-rectangle X_0. One approach to resolve this problem is to include some interior points in addition to the set of vertices of X_0 [10]. These additional points can represent the vertices and the facets centers of one or more hyper-rectangles contained in the interior of X_0. Based on the solutions at all points, a feasible local linear approximation $\widehat{U}_0(x) = K_0 x + g_0$ to the optimal solution $U^*(x)$, valid in the whole hyper-rectangle X_0, is determined by applying the following procedure:

Procedure 2 (computation of approximate solution):

Suppose A1–A3 hold, and consider any hyper-rectangle $X_0 \subseteq X_f$ with vertices $\Theta^0 = \left\{\theta_1^0, \theta_2^0, ..., \theta_M^0\right\}$ and center point w_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ with vertices respectively $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$, $j = 1, 2, ..., N_j$. Suppose $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$, determine a set of points that belongs to its facets and denote this set $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$, $j = 0, 1, 2, ..., N_j$. Define the set of all *points* $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$*, where* $w_i \in \left\{ \bigcup_{j=0}^{N_j} \Theta^j \right\} \cup \left\{ \bigcup_{j=0}^{N_j} \Psi^j \right\}$*,* $i = 1, 2, ..., N_1$*. Compute* K_0 *and* g_0 *by solving the following NLP:*

$$\min_{K_0, g_0} \sum_{i=0}^{N_1} (J(K_0 w_i + g_0, w_i) - V^*(w_i) + \mu \left\| K_0 w_i + g_0 - U^*(w_i) \right\|_2^2) \tag{14}$$

subject to:

$$G(K_0 w_i + g_0, w_i) \leq 0 \ , \ \ i \in \{0, 1, 2, ..., N_1\} \tag{15}$$

where N_1 *is the total number of points.*

□

In order to give an appropriate initialization of the NLP problem (14)-(15) for the region X_0, the already computed solutions of this problem in some of the neighbouring regions can be used as initial guesses.

3.3 Estimation of error bounds

Suppose that a state feedback $\widehat{U}_0(x)$ that is feasible in X_0 has been determined by applying Procedure 2. Then it follows that the sub-optimal cost $\widehat{V}(x) = J(\widehat{U}_0(x), x)$ is an upper bound on $V^*(x)$ in X_0, such that for all $x \in X_0$ we have:

$$0 \leq \widehat{V}(x) - V^*(x) \leq \varepsilon_0 \tag{16}$$

As already mentioned, the cost function J can be non-convex with multiple local minima. Therefore, in (16) $V^*(x)$ denotes a close-to-global solution. The following procedure can be used to obtain an estimate $\widehat{\varepsilon}_0$ of the maximal approximation error ε_0 in X_0.

Procedure 3 (computation of the error bound):

Consider any hyper-rectangle $X_0 \subseteq X_f$ *with vertices* $\Theta^0 = \{\theta_1^0, \theta_2^0, ..., \theta_M^0\}$ *and center point* w_0*. Consider also the hyper-rectangles* $X_0^j \subset X_0$*,* $j = 1, 2, ..., N_j$ *with vertices respectively* $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$*,* $j = 1, 2, ..., N_j$*. Suppose* $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$*. For each of the hyper-rectangles* X_0 *and* $X_0^j \subset X_0$*,* $j = 1, 2, ..., N_j$*, determine a set of points that belongs to its facets and denote this set* $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$*,* $j = 0, 1, 2, ..., N_j$*. Define the set of all points* $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$*, where* $w_i \in \left\{ \bigcup_{j=0}^{N_j} \Theta^j \right\} \cup \left\{ \bigcup_{j=0}^{N_j} \Psi^j \right\}$*,* $i = 1, 2, ..., N_1$*. Compute an estimate* $\widehat{\varepsilon}_0$ *of the error bound* ε_0 *through the following maximization:*

$$\widehat{\varepsilon}_0 = \max_{i \in \{0, 1, 2, ..., N_1\}} (\widehat{V}(w_i) - V^*(w_i)) \tag{17}$$

where N_1 *is the total number of points.*

□

3.4 Procedure and heuristic rules for splitting a region

The following procedure is applied to determine the best split of a region X_0 for which a feasible local state feedback $\widehat{U}_0(x)$ is found, but the required accuracy is not achieved.

Procedure 4 (determination of the best split of a region):

Consider a hyper-rectangle X_0 *and suppose that a feasible local state feedback* $\widehat{U}_0(x)$ *was found by applying Procedure 2. Suppose also that the required accuracy is not achieved. Then, determine the best split of* X_0 *in the following way:*

1. Let $j = 1$.

2. Split X_0 *by a hyperplane through its center and orthogonal to the axis* x_j*. Denote the new hyper-rectangles with* X_1^j *and* X_2^j.

3. Compute feasible local state feedbacks $\widehat{U}_1^j(x)$ *and* $\widehat{U}_2^j(x)$*, valid respectively in* X_1^j *and* X_2^j*, by applying Procedure 2.*

4. Compute estimates $\widehat{\varepsilon}_1^j$ *and* $\widehat{\varepsilon}_2^j$*, respectively of the error bounds* ε_1^j *in* X_1^j *and* ε_2^j *in* X_2^j*, by applying Procedure 4. Let* $\widehat{\varepsilon}^j = \widehat{\varepsilon}_1^j + \widehat{\varepsilon}_2^j$.

5. Let $j = j + 1$*. If* $j \leq n$*, go to step 2.*

6. Split X_0 *by a hyperplane through its center and orthogonal to the axis* x_j *where* $\widehat{\varepsilon}^j$ *is minimal.*

□

The following rule is applied when no feasible solution to the NLP problem (10)-(11) was found at some of the points $w_i \in W$, $w_i \neq w_0$, where the set $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$ is defined in Procedure 1.

Heuristic splitting rule 1 (handling infeasibility):

Consider the following two cases:

1). The set of the feasible points in X_0 *includes the center point* w_0 *and some of the points* $w_i \in W$, $w_i \neq w_0$ *(the set* $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$ *is defined in Procedure 1). Then, split* X_0 *into two types of hyper-rectangles by hyperplanes containing some of the feasible points* $w_i \in W$*:*

i. Hyper-rectangles $X_1^f, X_2^f, ..., X_{N_f}^f$ *containing only feasible points.*

ii. Hyper-rectangles $X_1^{nf}, X_2^{nf}, ..., X_{N_{nf}}^{nf}$ *containing some infeasible points.*

Denote the number of the new hyper-rectangles $N_s = N_f + N_{nf}$*. The optimal choice of dividing hyperplanes is the one which minimizes the number* N_s *of the new hyper-rectangles.*

2). The center point w_0 *of* X_0 *is the only feasible point. Then, split* X_0 *on all state space axes by hyperplanes through* w_0.

□

This rule is illustrated on Fig.2.

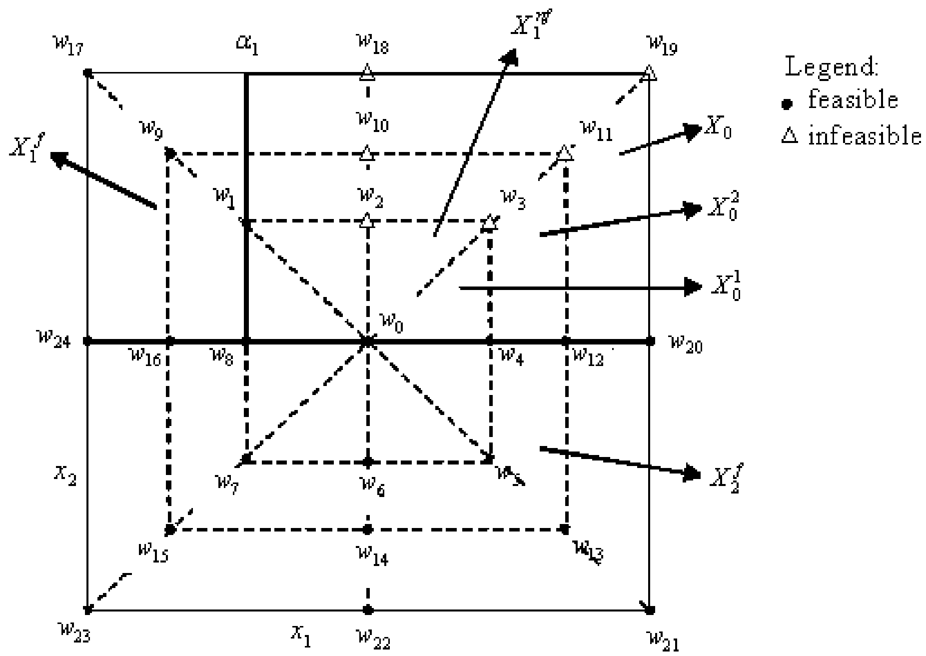

Fig. 2. Illustration of heuristic rule 1.

In Fig.2, the hyper-rectangle X_0 will be split into the hyper-rectangles X_1^f with vertices $\{w_{24}, w_8, w_{17}, \alpha_1\}$, X_2^f with vertices $\{w_{23}, w_{21}, w_{24}, w_{20}\}$ and X_1^{nf} with vertices $\{w_8, w_{20}, \alpha_1, w_{19}\}$.

The following rule is applied when there is no feasible solution to the NLP problem (10)-(11) at the center point w_0 of the hyper-rectangle X_0.

Heuristic splitting rule 2 (handling infeasibility):

If there is no feasible solution of the NLP (10)-(11) at the center point w_0 of X_0, split the hyper-rectangle X_0 by a hyperplane through w_0 and orthogonal to an arbitrary axis.

□

The following rule is used when the NLP problem (14)-(15) in Procedure 2 has no feasible solution.

Heuristic splitting rule 3 (handling infeasibility):

If the NLP problem (14)-(15) in Procedure 2 is infeasible, split the hyper-rectangle X_0 by a hyperplane through its center and orthogonal to an arbitrary axis.

□

3.5 Approximate algorithm for explicit solution of mp-NLPs

Assume the tolerance $\bar{\varepsilon} > 0$ of the cost function approximation error is given. The following algorithm is proposed to design explicit NMPC controller for constrained nonlinear systems:

Algorithm 1 (approximate explicit mp-NLP)

Step 1. Initialize the partition to the whole hyper-rectangle, i.e. $P = \{X\}$. Mark the hyper-rectangle X as unexplored.

Step 2. Select any unexplored hyper-rectangle $X_0 \in P$. If no such hyper-rectangle exists, the algorithm terminates successfully.

Step 3. Compute a solution to the NLP (10)-(11) at the center point w_0 of X_0 by applying Procedure 1. If the NLP has a feasible solution, go to step 4. Otherwise, split the hyper-rectangle

X_0 into two hyper-rectangles X_1 and X_2 by applying *the heuristic splitting rule 2.* Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2.

Step 4. Define a set of hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ contained in the interior of X_0. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$, in addition to its vertices, determine a set of points that belongs to its facets. Denote the set of all points (including the center point w_0) with $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$.

Step 5. Compute a solution to the NLP (10)-(11) for x fixed to each of the points w_i, $i = 1, 2, ..., N_1$ of the set W by applying Procedure 1. If all NLPs have a feasible solution, go to step 7. Otherwise, go to step 6.

Step 6. Compute the size of X_0 using some metric. If it is smaller than some given tolerance, mark X_0 infeasible and explored and go to step 2. Otherwise, split the hyper-rectangle X_0 into hyper-rectangles X_1, X_2,..., X_{N_s} by applying *the heuristic splitting rule 1.* Mark X_1, X_2,..., X_{N_s} unexplored, remove X_0 from P, add X_1, X_2,..., X_{N_s} to P, and go to step 2.

Step 7. Compute an affine state feedback $\widehat{U}_0(x)$ using Procedure 2, as an approximation to be used in X_0. If no feasible solution was found, split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2 by applying *the heuristic splitting rule 3.* Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2.

Step 8. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 in X_0 by applying Procedure 3. If $\widehat{\varepsilon}_0 \leq \bar{\varepsilon}$, mark X_0 as explored and feasible and go to step 2. Otherwise, split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2 by applying Procedure 4. Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2.

□

In contrast to the conventional MPC based on real-time optimization, the explicit MPC makes the rigorous verification and validation of the controller performance much easier [10]. Hence, problems due to lack of convexity and numerical difficulties can be addressed during the design and implementation. Notice that the off-line computational complexity and real-time computer memory requirements may grow very quickly with the number of states.

4 Application of the approximate explicit NMPC approach to compressor surge control

Consider the following 2-nd order compressor model [9],[17] with x_1 being normalized mass flow, x_2 normalized pressure and u normalized mass flow through a close-coupled valve in series with the compressor:

$$\dot{x}_1 = B(\Psi_e(x_1) - x_2 - u) \tag{18}$$

$$\dot{x}_2 = \frac{1}{B}(x_1 - \Phi(x_2)) \tag{19}$$

The following compressor and valve characteristics are used:

$$\Psi_e(x_1) = \psi_{c0} + H\left(1 + 1.5\left(\frac{x_1}{W} - 1\right) - 0.5\left(\frac{x_1}{W} - 1\right)^3\right) \tag{20}$$

$$\Phi(x_2) = \gamma sign(x_2)\sqrt{|x_2|} \tag{21}$$

with $\gamma = 0.5$, $B = 1$, $H = 0.18$, $\psi_{c0} = 0.3$ and $W = 0.25$. Like in [9], the control objective is to avoid surge, i.e. stabilize the system. This is formulated as [9]:

$$J(U, x(t)) = \sum_{k=0}^{N-1} \left[\alpha (x_{t+k|t} - x^*)^T (x_{t+k|t} - x^*) + k u_{t+k}^2 \right] + R v^2 + \beta (x_{t+N|t} - x^*)^T (x_{t+N|t} - x^*) \tag{22}$$

with $\alpha, \beta, k, R \geq 0$ and the set-point $x_1^* = 0.4$, $x_2^* = 0.6$ corresponds to an unstable equilibrium point. We have chosen $\alpha = 1$, $\beta = 0$ and $k = 0.08$. The horizon is chosen as $T = 12$, which is split into $N = 15$ equal-sized intervals, leading to a piecewise constant control input parameterization. Valve capacity requires the following constraint to hold:

$$0 \leq u(t) \leq 0.3 \tag{23}$$

The pressure constraint:

$$x_2(t) \geq 0.4 - v \tag{24}$$

avoids operation too far left of the operating point. The variable $v \geq 0$ is a slack variable introduced in order to avoid infeasibility and $R = 8$ is a large weight. Numerical analysis of the cost function shows that it is non-convex [9]. It can be seen that this NMPC problem formulation differs from that in section 2 in the absence of a terminal constraint and in the use of a slack variable.

The mp-NLP (10)-(11) has 16 free variables, 46 constraints and 2 parameters. The mp-NLP (14)-(15) has 46 free variables (the elements of the matrix K_0, vector g_0 and the slack variable v). The constraints in the problem (14)-(15) are 811, from which 765 are related to the control input and state constraints imposed at the vertices, the facets centers and the center point of a given region X_0, and the vertices and the facets centers of one internal region $X_0^1 \subset X_0$. In (14), it is chosen $\mu = 10$ and the control input only at the first sample is considered. The approximation tolerance is determined in the following way:

$$\bar{\varepsilon}(X_0) = \max(0.005, cV_{\min}^*) \tag{25}$$

where $c = 0.03$ and $V_{\min}^* = \min_{x \in X_0} V^*(x)$. Here, $V^*(x)$ denotes a close-to-global solution.

The state space to be partitioned is defined by $X = [0, 0.9] \times [0, 0.75]$. The partition of the approximate explicit NMPC controller is shown in Fig.3. It has 364 regions and 12 levels of search. With one scalar comparison required at each level of the $k - d$ tree, 12 arithmetic operations are required in the worst case to determine which region the state belongs to. Totally, 16 arithmetic operations are needed in real-time to compute the control input (12 comparisons, 2 multiplications and 2 additions).

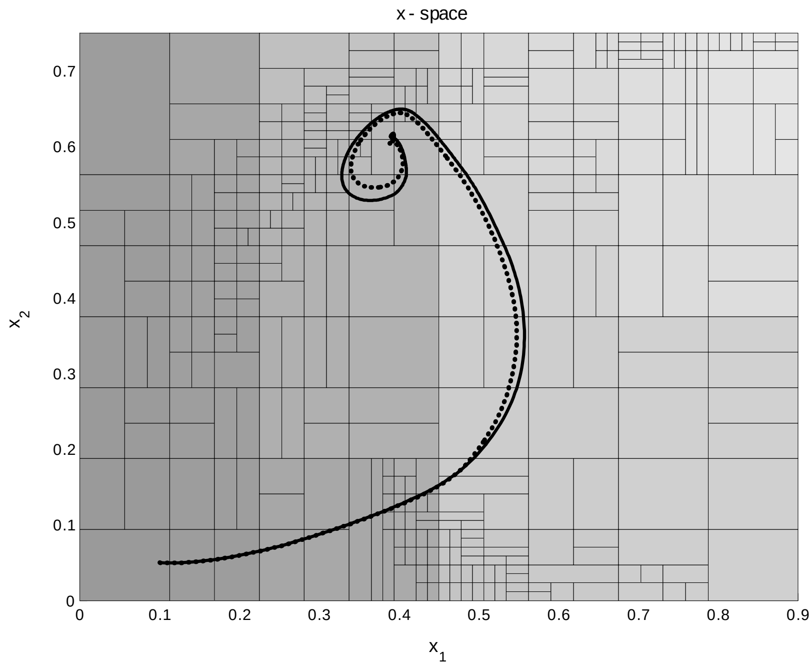

Fig. 3. The state space partition of the approximate explicit NMPC, the approximate (the solid curve) and the exact (the dotted curve) state trajectories.

The performance of the closed-loop system is simulated for initial condition $x(0) = [0.1 \;\; 0.05]^T$ and with sampling time $T_s = 0.02$. Euler integration with step size T_s is applied to solve the ordinary differential equations (18)-(19). The resulting closed-loop response is depicted in the state space (Fig.3), as well as trajectories in time (Fig.4 to Fig.6).

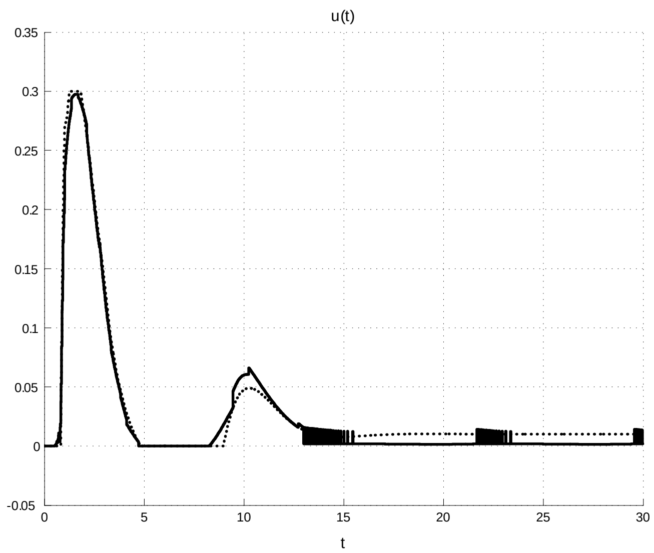

Fig. 4. Control input (the solid curve is with the approximate explicit NMPC and the dotted curve is with the exact NMPC).

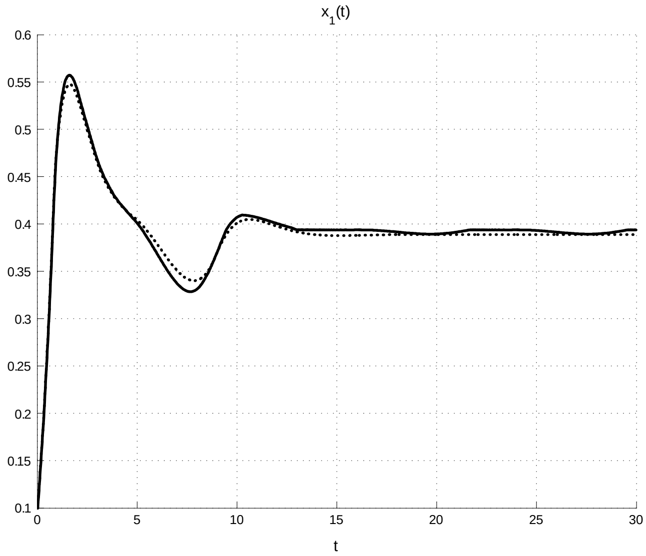

Fig. 5. State variable x_1 (the solid curve is with the approximate explicit NMPC and the dotted curve is with the exact NMPC).

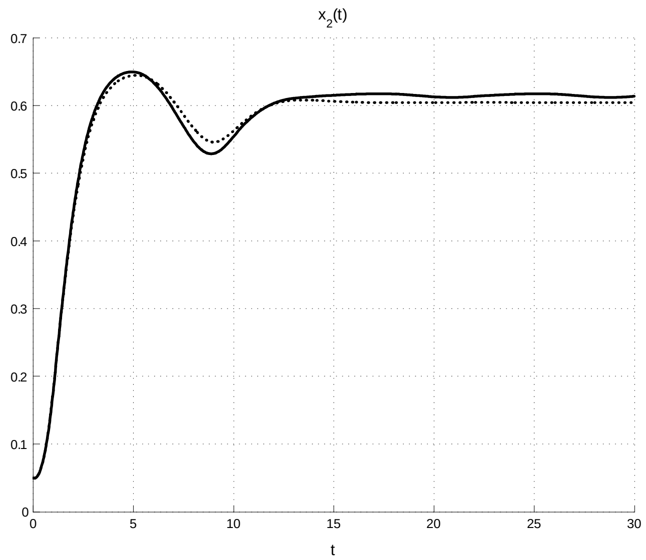

Fig. 6. State variable x_2 (the solid curve is with the approximate explicit NMPC and the dotted curve is with the exact NMPC).

The performance of the explicit NMPC controller can be improved by using a smaller approximation tolerance when partitioning the regions near the set point.

5 Conclusions

In this paper, practical computational methods for constructing approximate explicit PWL solutions of NMPC problems are developed. They represent an extension of the approximate approach in [10] since they provide some additional mechanisms to practically handle also the case of non-convexity of the resulting mp-NLP problem. The proposed methods when combined with verification and analysis methods give a practical tool for development and implementation of explicit NMPC. As a case study, the design of an approximate explicit NMPC for compressor surge control is considered.

References

[1] Keerthi, S. S. and Gilbert, E. G., "Optimal infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations", *Journal of Optimization Theory and Applications*, **vol.57**, pp.265–293, (1988).

[2] Mayne, D. Q. and Michalska, H., "Receding horizon control of nonlinear systems", *IEEE Transactions on Automatic Control*, **vol.35**, pp.814–824, (1990).

[3] Michalska, H. and Mayne, D. Q., "Robust receding horizon control of constrained nonlinear systems", *IEEE Transactions on Automatic Control*, **vol.38**, pp.1623–1633, (1993).

[4] Chen, H. and Allgöwer, F., "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability", *Automatica*, **vol.34**, pp.1205–1217, (1998).

[5] Magni, L., De Nicolao, G., Magnani, L. and Scattolini, R., "A stabilizing model-based predictive control for nonlinear systems", *Automatica*, **vol.37**, pp.1351–1362, (2001).

[6] Magni, L. and Scattolini, R., "Stabilizing model predictive control of nonlinear continuous time systems", *Annual Reviews in Control*, **vol.28**, pp.1-11, (2004).

[7] Bacic, M., Cannon, M. and Kouvaritakis, B., "Constrained NMPC via state-space partitioning for input-affine non-linear systems", *Proceedings of American Control Conference*, Denver, Colorado, pp.4881-4886, (2003).

[8] Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E. N., "The explicit linear quadratic regulator for constrained systems", *Automatica*, **vol.38**, pp.3-20, (2002).

[9] Johansen, T. A., "On multi-parametric nonlinear programming and explicit nonlinear model predictive control", *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, NV, **vol. 3**, pp. 2768-2773, (2002).

[10] Johansen, T. A., "Approximate explicit receding horizon control of constrained nonlinear systems", *Automatica*, **vol.40**, pp.293-300, (2004).

[11] Grancharova, A. and Johansen, T. A., "Explicit min-max model predictive control of constrained nonlinear systems with model uncertainty", *Proceedings of 16-th IFAC World Congress*, Prague, Czech Republic, Th-M03-TO/3, (2005).

[12] Fiacco, A. V., *Introduction to sensitivity and stability analysis in nonlinear programming*, Orlando, Fl: Academic Press, (1983).

[13] Bemporad, A. and Filippi, C., "Approximate multiparametric convex programming", *Proceedings of 42th IEEE Conf. on Decision and Control*, Maui, Hawaii, pp.3185–3190, (2003).

[14] Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality", *Automatica*, **vol.36**, pp.789-814, (2000).

[15] Bentley, J. L., "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, **vol.18**, pp.509–517, (1975).

[16] Grancharova, A. and Johansen, T. A., "Approximate explicit model predictive control incorporating heuristics", *Proceedings of IEEE International Symposium on Computer Aided Control System Design*, Glasgow, Scotland, U.K., pp.92-97, (2002).

[17] Gravdahl, J. T. and Egeland, O., "Compressor surge control using a close-coupled valve and backstepping", *Proceedings of American Control Conference*, Albuquerque, NM., **vol. 2**, pp. 982 –986, (1997).