INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2016; **26**:3292–3310
Published online 12 January 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3507

Real-time model predictive control based on dual gradient projection: Theory and fixed-point FPGA implementation

Matteo Rubagotti[1,*,†], Panagiotis Patrinos[2], Alberto Guiggiani[3] and Alberto Bemporad[3]

[1] *Department of Engineering, University of Leicester, University Rd, Leicester, UK*
[2] *Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Katholieke Universiteit Leuven, Leuven 3001, Belgium*
[3] *IMT Institute for Advanced Studies, Lucca 55100, Italy*

SUMMARY

This paper proposes a method to design robust model predictive control (MPC) laws for discrete-time linear systems with hard mixed constraints on states and inputs, in case of only an inexact solution of the associated quadratic program is available, because of real-time requirements. By using a recently proposed dual gradient-projection algorithm, it is proved that the discrepancy of the optimal control law as compared with the obtained one is bounded even if the solver is implemented in fixed-point arithmetic. By defining an alternative MPC problem with tightened constraints, a feasible solution is obtained for the original MPC problem, which guarantees recursive feasibility and asymptotic stability of the closed-loop system with respect to a set including the origin, also considering the presence of external disturbances. The proposed MPC law is implemented on a field-programmable gate array in order to show the practical applicability of the method. Copyright © 2016 John Wiley & Sons, Ltd.

Received 12 June 2014; Revised 23 August 2015; Accepted 30 November 2015

KEY WORDS: model predictive control; uncertain systems; optimization methods

1. INTRODUCTION

Model predictive control (MPC) allows the design of optimal feedback control laws for dynamical systems that take into account constraints on inputs and states [1]. Thanks to the accomplishments of the last years in increasing both the computation capabilities of microcontrollers and the efficiency of fast algorithms for online optimization, the use of MPC is being extended from the traditional process control applications to fields like automotive, aerospace, and mechatronics, where relatively fast sampling times are required. In order to implement embedded MPC controllers for this kind of systems, the worst-case execution time at each sampling time must be known, that is, *real-time* guarantees are needed. To this aim, real-time MPC laws have been recently proposed, studying optimization algorithms that generate an acceptable solution in an a priori bounded number of iterations.

When a linear model of the system is employed, together with a quadratic cost function and linear constraints, the optimization problem to be solved online can be formulated as a quadratic program (QP). In order to provide a solution of the QP in a prescribed time, two different strategies have been considered. The first consists of using the so-called *explicit MPC* approach introduced in [2], in which the optimal control law is explicitly obtained during the design phase as a piecewise affine function of the state vector by means of parametric optimization. For small-size problems, such

*Correspondence to: Matteo Rubagotti, Department of Engineering, University of Leicester, University Rd, Leicester LE1 7RH, UK.

†E-mail: mr298@le.ac.uk

a function can be easily implemented in embedded control systems, giving a precise estimate of the worst-case execution time. Conversely, if the problem is of medium or large size, then online optimization has to be employed. Using standard solvers, based on interior-point or active-set methods, the estimate of the worst-case execution time is usually extremely loose (see, for instance, the discussion in [3, Section IA]). For this reason, algorithms based on variants of the fast gradient methods [4, 5] have been recently applied to MPC in [3, 6–8], providing tighter bounds. In [9], a dual gradient-projection method based on [4] was proposed both in its basic (GPD) and accelerated (GPAD) forms (see also [10]). Despite introducing a *dual* method, [9] provides bounds on the maximum number of iterations in order to achieve given levels of *primal* suboptimality and infeasibility. In [11], the use of GPD was analyzed for embedded MPC in hardware platforms with fixed-point arithmetic.

In case a primal method is used, the obtained suboptimal solution does not violate the inequality constraints of the finite-horizon optimal control problem, and closed-loop stability can be proved using, for instance, the method described in [12]. However, the use of primal methods is limited to problems with simple input constraints, such as box constraints and no state constraints (e.g., [3]). Dual methods (e.g., [9, 13–16]) have been successfully applied to more general problem formulations (with polytopic mixed constraints on input and state variables), but they present the drawback of providing inexact solutions for the associated primal problem because the inequality constraints can be violated. Possible solutions to this drawback were recently proposed in [17–19]. The approaches of [14, 17] do not provide a priori bounds on the maximum number of iterations valid for the entire region of attraction, which are instead provided in [19]. In particular, in [19], starting from the MPC problem to be solved online, and given the maximum constraint violation and suboptimality of the solution provided by the solver (e.g., GPAD), an alternative problem is formulated with tightened constraints. By showing that the obtained inexact solution is feasible for the actual MPC problem, asymptotic stability is guaranteed for the closed-loop system. The main drawback of [19] is that the tightening of the constraints increases of a constant amount at each step along the prediction horizon, which can lead to infeasibility of the alternative MPC problem for long prediction horizons.

The paper is structured as follows. After introducing the main notation in Section 2, it is assumed that there exists a bound on the discrepancy w between the control variable originated from the exact optimal solution of the MPC problem and the one originated from the inexact solution. By extending the approach of [20] to systems with mixed control and state constraints, a robust MPC problem is defined in Section 3, based on tightened constraints, by treating w as a disturbance term. The amount of tightening of the constraints does not increase linearly along the prediction horizon, instead it is shown to converge to a constant value. The scheme is also proved to be robust with respect to bounded external disturbances showing that recursive feasibility is guaranteed in a given domain of attraction, and that the closed-loop system is asymptotically stable with respect to a set including the origin, the size of such set depending on the external disturbance term. In Section 4, we show how to obtain the bounds on w by using the GPD algorithm [9], also analyzing the effect of the number of bits used to represent numbers in embedded fixed-point implementations of the solver. In Section 5, a comparison with [19] is carried out, showing that, as the prediction horizon increases, the proposed method can be preferable to [19]. The last contribution of the paper is described in Section 6, where the implementation of the proposed MPC law on a field-programmable gate array (FPGA) is detailed, and the related results on hardware-in-the-loop experiments are discussed. In Section 7, conclusions are drawn and the contributions of the paper are discussed in detail.

2. BASIC NOTATION

Let $\mathbb{R}_{>0}$, $\mathbb{R}_{\geqslant 0}$, $\mathbb{N}_{>0}$ and $\mathbb{N}_{\geqslant 0}$ denote the sets of positive reals, non-negative reals, positive integers, and non-negative integers, respectively. Given two integers $a \leqslant b$, let $\mathbb{N}_{[a,b]} \triangleq \{a, a+1, ..., b\}$, while $\mathbb{N}_b \triangleq \{0, 1, ..., b\}$. Given a vector $v \in \mathbb{R}^n$, let $\|v\|$ denote its Euclidean norm, while $\mathfrak{B}_\chi \triangleq \{a \in \mathbb{R}^n : \|a\| \leqslant \chi\}$, for any $\chi \in \mathbb{R}_{>0}$. Given two vectors $u, v \in \mathbb{R}^n$, the notation $u \leqslant v$ refers to component-wise inequalities. Given a matrix $M \in \mathbb{R}^{n \times n}$, M' is its transpose, $\rho(M)$ is its spectral

DOI: 10.1002/rnc

radius, and its positive definiteness and semi-definiteness are indicated as $M \succ 0$ and $M \succeq 0$, respectively. Given a set $\mathcal{X} \subseteq \mathbb{R}^n$, its interior is denoted by $\text{int}(\mathcal{X})$. The Hausdorff distance of point $p \in \mathbb{R}^n$ from the set $\mathcal{X}$ is $\delta_h(p, \mathcal{X})$. Given $a \in \mathbb{R}_{\geqslant 0}$, we define $a\mathcal{X} \triangleq \{y \in \mathbb{R}^n \,:\, y = ax,\ x \in \mathcal{X}\}$. Given two sets $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{R}^n$, their Minkowski sum is $\mathcal{A}_1 \oplus \mathcal{A}_2 \triangleq \{x + y : x \in \mathcal{A}_1, y \in \mathcal{A}_2\}$, and their Pontryagin difference is $\mathcal{A}_1 \ominus \mathcal{A}_2 \triangleq \{x \in \mathbb{R}^n : x + y \in \mathcal{A}_1,\ \forall y \in \mathcal{A}_2\}$. We define a *polytope* as a bounded and closed polyhedron obtainable as the convex hull of its vertices.

3. FORMULATION OF THE ROBUST MODEL PREDICTIVE CONTROL CONTROL LAW

3.1. Overview of the model predictive control problem with mixed constraints

Consider the uncertain discrete-time linear time-invariant (LTI) system

$$x(t+1) = Ax(t) + Bu(t) + d(t), \tag{1}$$

where $t \in \mathbb{N}_{\geqslant 0}$, $x \in \mathbb{R}^{n_x}$ is the state vector (which is available for feedback for all $t \in \mathbb{N}_{\geqslant 0}$), $u \in \mathbb{R}^{n_u}$ is a controlled input, and $d \in \mathbb{R}^{n_x}$ is a disturbance input. It is assumed that the pair (A, B) is stabilizable, and that the disturbance term is bounded as

$$d(t) \in \mathcal{D}, \forall t \in \mathbb{N}_{\geqslant 0}, \tag{2}$$

where $\mathcal{D}$ is a non-empty polytope in $\mathbb{R}^{n_x}$ with $0 \in \mathcal{D}$. The state and input vectors are represented using a single vector

$$z \triangleq \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_z},\ n_z \triangleq n_x + n_u. \tag{3}$$

The considered problem consists of regulating $x(t)$ to a set including the origin, while satisfying

$$z(t) \in \mathcal{Z}, \forall t \in \mathbb{N}_{\geqslant 0}, \tag{4}$$

where $\mathcal{Z}$ is a polytope with $0 \in \mathcal{Z}$. First of all, the *auxiliary control law* $\kappa(x) \triangleq Kx$ is defined, where $K \in \mathbb{R}^{n_u \times n_x}$ is a gain matrix defined such that the resulting nominal closed-loop system

$$x(t+1) = A_\kappa x(t), \tag{5}$$

where $A_\kappa \triangleq A + BK$ is asymptotically stable. Because the pair (A, B) is stabilizable, then it is always possible to synthesize K to obtain asymptotic stability, that is, $\rho(A_\kappa) < 1$.

Let the MPC control law be

$$u(x) \triangleq Kx + c(x), \tag{6}$$

which has the same structure of the MPC law proposed in [20], with the difference that [20] did not consider the presence of mixed constraints. These require considering the dynamics of $z(t)$ as a whole, which leads to a more complex formulation with respect to [20]. Also, the application of an inexact control law will be considered in the following as a further disturbance term added to $d(t)$, which was not considered in [20]. The cost function that will be minimized over a *prediction horizon* $N \in \mathbb{N}_{>0}$ is given by

$$V_N(\boldsymbol{c}) \triangleq \sum_{k=0}^{N-1} c_k' \Psi c_k, \tag{7}$$

where $\Psi \in \mathbb{R}^{n_u \times n_u}$, $\Psi = \Psi' \succ 0$. Also, $\mathbf{c}$ in (7) is defined as

$$\mathbf{c} \triangleq \begin{bmatrix} c_0' & \dots & c_{N-1}' \end{bmatrix}' \in \mathbb{R}^n, \quad n \triangleq Nn_u, \tag{8}$$

together with

$$\mathbf{z} \triangleq \begin{bmatrix} z_0' & \cdots & z_{N-1}' & x_N' \end{bmatrix}', \quad z_k \triangleq \begin{bmatrix} x_k \\ Kx_k + c_k \end{bmatrix}.$$

In order to compute the evolution of the system dynamics along the prediction horizon, and therefore define the vectors $\mathbf{c}$ and $\mathbf{z}$, the nominal closed-loop dynamics is used by applying the control law (6) to system (1), assuming $d(t) = 0$ for all $t \in \mathbb{N}_{\geqslant 0}$. Therefore, we will require that $\mathbf{z} \in \mathcal{A}(x)$, with

$$\mathcal{A}(x) \triangleq \{\mathbf{z} : \; x_0 = x, x_{k+1} = A_\kappa x_k + B c_k, \; k \in \mathbb{N}_{N-1}\}. \tag{9}$$

Because the disturbance term is neglected, a particular set of tightened constraints will be defined in the following, so as to satisfy constraint (4) in the presence of disturbances. After the given optimization problem is solved, and a suitable (not necessarily optimal) solution is determined, according to the *receding horizon* principle, only the first control move $u(x) = \kappa(x) + c_0(x)$ is applied to the system at time t, while the optimization process with the same prediction horizon N is repeated at time $t + 1$.

Remark 1
Consider the case when K is defined using infinite-horizon linear quadratic regulation theory, given the weight matrices $Q \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$, with $Q = Q' \succeq 0$, $R = R' \succ 0$, and $(A, Q^{\frac{1}{2}})$ detectable. In this case, we can define the *stage cost* $\ell(x,u) \triangleq \frac{1}{2}(x'Qx + u'Ru)$ and the *terminal cost* $\ell_N(x) \triangleq \frac{1}{2}x'Px$, where $P = P' \succ 0$ is the solution of the associated algebraic Riccati equation. Then, if $\Psi = R + B'PB$, the optimal control sequence obtained by minimizing (7) subjected to a given set of constraints can be also obtained by minimizing $\hat{V}_N(\mathbf{z}) \triangleq \sum_{k=0}^{N-1} \ell(x_k, u_k) + \ell_N(x_N)$, for the same set of constraints. For a detailed discussion on this equivalence result, the reader is referred to [20] and the references therein.

3.2. *Definition of the tightened constraints*

In addition to the uncertainty given by the external disturbance d, it is necessary to take into account the fact that the optimal value of $\mathbf{c}$ is not achieved. The sequence of values of the control variable $c(x)$, generated by the given numerical solver along the prediction horizon, is referred to as

$$\bar{\mathbf{c}}(x) \triangleq \left[\bar{c}_0(x) \; \ldots \; \bar{c}_{N-1}(x)\right], \tag{10}$$

while the optimal control sequence of the same problem is referred to as

$$\mathbf{c}^\star(x) \triangleq \left[c_0^\star(x) \; \ldots \; c_{N-1}^\star(x)\right]. \tag{11}$$

Assumption 1
For all x for which $\mathbf{c}^\star(x)$ is defined, $\exists \, \lambda \in \mathbb{R}_{\geqslant 0}$ s.t. $\bar{c}_i(x) \in c_i^\star(x) \oplus \mathfrak{B}_\lambda \; \forall \, i = 1, \ldots, N-1$.

The information given by Assumption 1 is of paramount importance for proving the stability results in this paper, and it will be shown that such an assumption will be automatically satisfied by the use of the specific solver. The meaning of parameter λ is related to the number of iterations of the solver, and on the numerical precision of the actual implementation (e.g., the effect of fixed-point arithmetic can be taken into account) as will be clarified in Section 4. By means of Assumption 1, each term of the control sequence applied to system (1) can be expressed as $c_i^\star(x) + e_i(x)$, where $e_i(x) \in \mathfrak{B}_\lambda, i = 1, \ldots, N-1$. In order to deal with the new artificial disturbance term, recalling the expression of $c(x)$ in (6), the expression of control law (6) is made explicit in system (1), as follows

$$x(t+1) = A_\kappa x(t) + Bc(t) + Be(t) + d(t), \tag{12}$$

where $c(t) \triangleq c_0^\star(x(t))$, and $e(t) \triangleq e_0(x(t))$. Because no information is available on the control law $\bar{c}_0(x)$ apart from that given by Assumption 1, we study the effect of applying $\bar{c}_0(x)$ as if we applied the optimal control law $c_0^\star(x)$, plus the uncertain term $e(t)$. For the sake of compactness, we use a single variable to refer to all uncertain inputs, as $w(t) \triangleq Be(t) + d(t)$.

Given the assumptions on $e(t)$ and $d(t)$, $w(t)$ belongs to the set $B\mathfrak{B}_\lambda \oplus \mathcal{D} \subset \mathbb{R}^{n_x}$. For computational reasons, it is preferable to consider an over-approximation of this set as a polytope. To this aim, a polytope $\mathcal{E}_0 \subset \mathbb{R}^{n_u}$ is defined s.t. $\mathcal{E}_0 \supset \mathfrak{B}_1$, and $\mathcal{E} \triangleq \lambda\mathcal{E}_0 \supset \mathfrak{B}_\lambda$. At this point, it is possible to state that $w(t) \in \mathcal{W} \triangleq B\mathcal{E} \oplus \mathcal{D} \subset \mathbb{R}^{n_x}$, $t \in \mathbb{N}_{\geqslant 0}$, where $\mathcal{W}$ is a polytope that includes the origin.

DOI: 10.1002/rnc

Applying the auxiliary control law $\kappa(x)$ to system (12) (i.e., setting $c(t) = 0$), one obtains the closed-loop system

$$x(t+1) = A_\kappa x(t) + w(t) \tag{13}$$

for which the set of states x reachable from the origin in j steps for any admissible disturbance sequence $\mathbf{w}_j \in \mathcal{W} \times \ldots \times \mathcal{W}$ is given by $\mathcal{R}_j \triangleq \bigoplus_{i=0}^{j-1} \left(A_\kappa^i \mathcal{W}\right)$, which implies that the minimal robust positively invariant (RPI) ([21, Definition 1]) set for the same system is $\mathcal{R}_\infty \triangleq \lim_{j\to\infty} \mathcal{R}_j$ (e.g., [20]). Even if, for the sake of simplicity, we will assume to be able to obtain $\mathcal{R}_\infty$ exactly, in practice, an over-approximation $\hat{\mathcal{R}}_\infty \supset \mathcal{R}_\infty$ can be employed. For the description of an efficient iterative method to obtain $\hat{\mathcal{R}}_\infty$, the reader is referred to [21]. The dynamics of z in (3) for system (13) are

$$z(t+1) = \tilde{A}_\kappa z(t) + \tilde{B}_\kappa w(t), \tag{14}$$

where

$$\tilde{A}_\kappa \triangleq \begin{bmatrix} A_\kappa & 0 \\ KA_\kappa & 0 \end{bmatrix}, \; \tilde{B}_\kappa \triangleq \begin{bmatrix} I \\ K \end{bmatrix}.$$

The set of extended states z reachable from the origin in j steps for any admissible disturbance sequence $\mathbf{w}_j \in \mathcal{W} \times \ldots \times \mathcal{W}$ is $\tilde{\mathcal{R}}_j \triangleq \bigoplus_{i=0}^{j-1} \left(\tilde{A}_\kappa^i \tilde{B}_\kappa \mathcal{W}\right)$, implying that the minimal RPI set for system (14) is given by $\tilde{\mathcal{R}}_\infty \triangleq \lim_{j\to\infty} \tilde{\mathcal{R}}_j$.

Remark 2
By definition of dynamics (13) and (14), $x \in \mathcal{R}_j$ if and only if $z \in \tilde{\mathcal{R}}_j$, for all $j \in \mathbb{N}_{\geq 0}$.

Assumption 2
The matrices A, B, and K and the sets $\mathcal{E}$ and $\mathcal{D}$ are defined such that $\tilde{\mathcal{R}}_\infty \subseteq \mathcal{Z}$.

Remark 3
In Assumption 2, two different kinds of parameters are considered. On the one hand, A, B, and $\mathcal{D}$ are given as characteristics of the system. On the other hand, K and $\mathcal{E}$ can be modified by the designer. Given the presence of both input and state constraints, the influence of K on $\tilde{\mathcal{R}}_\infty$ is strongly case-dependent. Instead, $\mathcal{E}$ is shrinked as more iterations of the numerical solver are run: in the ideal case of infinite numerical precision, the obtained control law approaches the optimal one as the number of iterations tends to infinity. In this limit case, as $\lambda \to 0$, we would obtain $\mathcal{R}_\infty$ equal to

$$\mathcal{R}_D \triangleq \lim_{j\to\infty} \bigoplus_{i=0}^{\infty} \left(A_\kappa^i \mathcal{D}\right), \tag{15}$$

which is included in $\mathcal{R}_\infty$ for all $\lambda \in \mathbb{R}_{\geq 0}$. Also, as $\lambda \to 0$, the set $\tilde{\mathcal{R}}_\infty$ would become equal to $\tilde{\mathcal{R}}_D \triangleq \lim_{j\to\infty} \bigoplus_{i=0}^{j-1} \left(\tilde{A}_\kappa^i \tilde{B}_\kappa \mathcal{D}\right)$. In case the optimal solution is achieved, a necessary condition needed for the satisfaction of Assumption 2 is $\tilde{\mathcal{R}}_D \subseteq \mathcal{Z}$. Assuming a finite value $\lambda > 0$, then the same necessary condition will require that the polytope $\tilde{\mathcal{R}}_D$ be included in the relative interior of $\mathcal{Z}$. Smaller sets $\tilde{\mathcal{R}}_D$ would lead to the possibility of using larger values of λ and still satisfy Assumption 2. Instead, if $\mathcal{Z} \ominus \tilde{\mathcal{R}}_D$ is very small, only a very small value of λ can be acceptable.

By definition of $\mathcal{R}_j$ and $\tilde{\mathcal{R}}_j$, it is immediate to see that, given $j \in \mathbb{Z}_{>0}$, $\mathcal{R}_j = \mathcal{R}_{j-1} \oplus A_\kappa^{j-1}\mathcal{W}$, $\tilde{\mathcal{R}}_j = \tilde{\mathcal{R}}_{j-1} \oplus \tilde{A}_\kappa^{j-1}\tilde{B}_\kappa\mathcal{W}$. Also, for $j \in \mathbb{R}_{\geq 0}$, we define the tightened sets

$$\mathcal{Z}_j = \mathcal{Z} \ominus \tilde{\mathcal{R}}_j. \tag{16}$$

The *terminal set* $\mathcal{X}_f$ is defined as the maximal RPI set in

$$\mathcal{X}_K \triangleq \left\{ x \in \mathbb{R}^{n_x} : \begin{bmatrix} x \\ Kx \end{bmatrix} \in \mathcal{Z} \right\}, \tag{17}$$

for the closed-loop system (13), that is,

$$x \in \mathcal{X}_f \Rightarrow z = \begin{bmatrix} x \\ Kx \end{bmatrix} \in \mathcal{Z},\ A_\kappa x + Bw \in \mathcal{X}_f, \forall w \in \mathcal{W}. \tag{18}$$

Recalling Remark 2, $x \in \mathcal{R}_\infty$ implies $z \in \tilde{\mathcal{R}}_\infty \subseteq \mathcal{Z}$, and then, by definition of $\mathcal{X}_f$, $\mathcal{R}_\infty \subseteq \mathcal{X}_f$. Therefore, Assumption 2 also implies the existence of $\mathcal{X}_f$.

Assumption 3
A condition slightly stronger than $\mathcal{R}_\infty \subseteq \mathcal{X}_f$ (implied by Assumption 2) is assumed to be satisfied, that is $\mathcal{R}_\infty \subset \text{int}(\mathcal{X}_f)$.

3.3. Definition of the finite-horizon optimal control problem

The finite-horizon optimal control problem leading to the definition of $\bar{c}_0(x)$ is defined as

$$V_N^\star(x) \triangleq \min\{V_N(\mathbf{c}) : \mathbf{z} \in \mathcal{S}_N(x)\}, \tag{19}$$

where

$$\mathcal{S}_N(x) \triangleq \left\{\mathbf{z} \in \mathcal{A}(x) : \begin{bmatrix} x_k \\ Kx_k + c_k \end{bmatrix} \in \mathcal{Z}_k,\ k \in \mathbb{N}_{N-1}, x_N \in \mathcal{X}_f \ominus \mathcal{R}_N \right\},$$

where $V_N(\mathbf{c})$ and $\mathcal{A}(x)$ are defined in (7) and (9), respectively. The set $\mathcal{D}_N$ is defined as the set of states x for which there exists a feasible solution for (19), given the prediction horizon N. For every $x \in \mathcal{D}_N$, the unique optimal solution of (19) is denoted by $\mathbf{z}^\star \triangleq \left[z_0^{\star\prime} \cdots z_{N-1}^{\star\prime}\ x_N^{\star\prime}\right]'$, where $z_k^\star \triangleq \left[x_k^{\star\prime}\ (Kx_k^\star)' + c_k^{\star\prime}\right]'$. Also, we recall that the associated optimal control sequence is denoted by $\mathbf{c}^\star(x)$ [11]. Even though the optimal solution $\bar{\mathbf{z}}^\star(x)$ will not be achieved, for every state vector $x \in \mathcal{D}_N$, we assume that a vector $\bar{\mathbf{z}}(x) = \left[\bar{z}_0' \cdots \bar{z}_{N-1}'\ \bar{x}_N'\right]' \in \mathbb{R}^{Nn_z+n_x}$ can be computed for which, in addition to Assumption 1, the following holds:

Assumption 4
For all $x \in \mathcal{D}_N$, vector $\bar{\mathbf{z}}(x)$ (which is not necessarily a feasible solution of problem [19]) satisfies

$$\bar{\mathbf{z}}(x) \in \mathcal{A}(x), \tag{20}$$

$$x \in \mathcal{X}_f \Rightarrow \bar{\mathbf{z}}(x) = \mathbf{z}^\star(x). \tag{21}$$

Assumption 4 implies that $\bar{\mathbf{c}}(x) = \mathbf{c}^\star(x) = \left[0 \ldots 0\right]$ for all $x \in \mathcal{X}_f$, Because this choice leads to a feasible solution of problem (19) with $V_N(\mathbf{z}) = 0$. For each $x \in \mathcal{D}_N$, let $Z(x)$ denote the set of all vectors $\bar{\mathbf{z}}(x) \in \mathbb{R}^{Nn_z+n_x}$, which can be associated to a feasible solution $\mathbf{z}(x)$ of (19), satisfying Assumption 1. Also, let $\mathcal{C}(x)$ be the set of all $\bar{c}_0(x)$ corresponding to vectors $\bar{\mathbf{z}}(x) \in Z(x)$. Notice that $Z(x)$ (and consequently $\mathcal{C}(x)$) is non-empty for all $x \in \mathcal{D}_N$, because it contains $\mathbf{z}^\star(x)$.

Theorem 1
Let Assumptions 1–4 be satisfied and consider the closed-loop system

$$x(t+1) = \varphi(x(t)) \triangleq Ax(t) + Bu(t) + d(t), \tag{22}$$

where $u(t) = Kx(t) + \mu(x(t))$, $\mu(x(t)) \in \mathcal{C}(x(t))$. Then,

(i) Recursive feasibility for (19) is ensured, that is, $\mathcal{D}_N$ is an RPI set for the closed-loop system;
(ii) $(x(t), \mu(x(t))) \in \mathcal{Z}$, $t \in \mathbb{N}_{\geqslant 0}$;
(iii) The set $\mathcal{R}_D$ in (15) is asymptotically stable for system (22), according to Definition 3 in the Appendix, with domain of attraction $\mathcal{D}_N$.

Proof
To improve readability, the proof is reported in the Appendix. □

4. ANALYSIS OF THE OPTIMIZATION ALGORITHM

The whole theoretical development so far has been based on the assumption that a solver is available for problem (19), such that Assumptions 1 and 4 are satisfied. In this section, we briefly summarize GPD [11], a gradient projection algorithm applied to a modified version of the dual problem of (19), specifically tailored for implementation in fixed point hardware. We prove that the required assumptions are satisfied, and that the value of λ defined in Assumption 1 will depend on the number of iterations of GPD, and on the numerical precision of the fixed-point arithmetic with which GPD is running. By eliminating the equality constraints corresponding to the state equations, problem (19) can be expressed as the following strongly convex QP

$$V_N^{\star}(x) = \min_{\mathbf{c}\in\mathbb{R}^n}\{(1/2)\mathbf{c}'Mc + q(x)'\mathbf{c} \mid F\mathbf{c} \leqslant s(x)\}, \tag{23}$$

where $q : \mathbb{R}^{n_x} \to \mathbb{R}^{Nn_u}$, $s : \mathbb{R}^{n_x} \to \mathbb{R}^m$ are affine mappings (m being the number of inequality constraints and $\mathbf{c}$ being defined in (8)). This leads to the satisfaction of (20) in Assumption 4. The dual function is

$$\Phi(x, y) \triangleq \min_{\mathbf{c}} (1/2)\mathbf{c}'M\mathbf{c} + q(x)'\mathbf{c} + y'(F\mathbf{c} - s(x)), \tag{24}$$

and the dual problem is

$$\Phi^{\star}(x) = \max_{y\geqslant 0} \Phi(x, y). \tag{25}$$

Let $\mathcal{Y}^{\star}(x)$ denote the set of dual optimal solutions for given state $x \in \mathbb{R}^{n_x}$ and

$$\Delta_y \triangleq \max\left\{\max_{x\in\mathcal{D}_N} \min_{y^{\star}(x)\in\mathcal{Y}^{\star}(x)} \|y^{\star}(x)\|_{\infty}, 1\right\}, \tag{26}$$

and consider the problem

$$\Phi^{\star}(x) = \max_{0\leqslant y\leqslant\Delta_y} \Phi(x, y). \tag{27}$$

It is easy to see that problems (25) and (27) are completely equivalent, cf. [11]. However, the boundedness of the feasible set for (27) greatly facilitates the round-off error analysis because of fixed-point arithmetic. For a given $y \in \mathbb{R}^m$, the value of the dual cost can be calculated by computing the argument that achieves the minimum in (24). This is given by

$$\mathbf{c}_y^{\star} = Ey + e(x), \tag{28}$$

where $E = -M^{-1}F'$, $e(x) = -M^{-1}q(x)$. The gradient of the dual function is simply

$$\nabla_y\Phi(x, y) = F\mathbf{c}_y^{\star} - s(x).$$

Therefore, the gradient projection algorithm applied to the dual problem (27) is

$$\mathbf{c}_{(\nu)} = Ey_{(\nu)} + e(x) \tag{29a}$$

$$g_{(\nu)} = F\mathbf{c}_{(\nu)} - s(x) \tag{29b}$$

$$y_{(\nu+1)} = \min\left\{\max\left\{y_{(\nu)} + \tfrac{1}{L}g_{(\nu)}, 0\right\}, 2\Delta_y\mathbf{1}\right\} \tag{29c}$$

where $L = \|F\|^2/\mu$ and $\mu = \lambda_{\min}(M)$. For simplicity, we assume that $y_{(0)} = 0$, which in turn implies (21). Assumption 4 is therefore entirely satisfied.

DOI: 10.1002/rnc

Theorem 2
Let $x \in \mathcal{D}_N$ and

$$\bar{\mathbf{c}}_{(\nu)} \triangleq \frac{1}{\nu+1}\sum_{i=0}^{\nu}\mathbf{c}_{(i)}, \tag{30}$$

where $\{\mathbf{c}_{(\nu)}\}$ is the sequence generated by iteration (29) running on a fixed-point architecture with p fractional bits. Then, we have

$$\bar{c}_i(x) \in c_i^\star(x) \oplus \mathfrak{B}_\lambda \tag{31}$$

after at most

$$\nu_\lambda^\star = \left\lceil \frac{D^2}{\lambda^2 - \delta} \right\rceil - 1 \tag{32}$$

iterations, where

$$D \triangleq \frac{\|F\|^2\Delta_y^2(4\Delta_y+1)}{\mu}, \quad \delta \triangleq \frac{2}{\mu}(1+\Delta_y)\left(\frac{\|F\|^2}{\mu}\epsilon_z^2 + 4\sqrt{m}\Delta_y\epsilon_\xi\right),$$
$$\epsilon_z \triangleq 2^{-(p+1)}m\sqrt{n}, \quad \epsilon_\xi \triangleq 2^{-(p+1)}n\sqrt{m}.$$

Proof
Let $\mathcal{L}(\mathbf{c}, y) = (1/2)\mathbf{c}'M\mathbf{c} + q(x)'\mathbf{c} + y'(F\mathbf{c} - s(x))$ denote the Lagrangian of (23) (we have omitted the dependence on x for sake of clarity). By the assumptions of the statement, $\mathbf{c}^\star$ and $y^\star$ satisfy (see, e.g., [22])

$$\nabla_c\mathcal{L}(\mathbf{c}^\star, y^\star) = 0. \tag{33a}$$

The function $\mathcal{L}(\cdot, y^\star) : \mathbb{R}^n \to \mathbb{R}$ is strongly convex on $\mathcal{Z}$, with convexity parameter μ as the positive weighted sum of the strongly convex function V with the convex functions $\mathbf{c} \mapsto y'(F\mathbf{c} - s)$, $i \in \mathbb{N}_{[1,m]}$ [5, Lemma 2.1.4]. This leads to

$$\mathcal{L}(\bar{\mathbf{c}}_{(\nu)}, y^\star) \geqslant \mathcal{L}(\mathbf{c}^\star, y^\star) + \tfrac{1}{2}\mu\|\bar{\mathbf{c}}_{(\nu)} - \mathbf{c}^\star\|^2. \tag{34}$$

Therefore,

$$\|\bar{\mathbf{c}}_{(\nu)} - \mathbf{c}^\star\|^2 \leqslant \frac{2}{\mu}(\mathcal{L}(\bar{\mathbf{c}}_{(\nu)}, y^\star) - \mathcal{L}(\mathbf{c}^\star, y^\star)) \leqslant \frac{2}{\mu}(V(\bar{\mathbf{c}}_{(\nu)}) - V^\star + y^{\star\prime}(F\bar{\mathbf{c}}_{(\nu)} - s(x))), \tag{35}$$

where the second inequality follows from complementarity, that is, $(F\bar{\mathbf{c}}^\star - c)'y^\star = 0$. From [11, Theorems 2, 3, Section VI], we have

$$\left\|[F\bar{\mathbf{c}}_{(\nu)} - s(x)]_+\right\|_\infty \leqslant \tfrac{2L\Delta_y^2}{\nu+1} + \delta, \tag{36a}$$
$$V(\bar{\mathbf{c}}_{(\nu)}) - V^\star \leqslant \tfrac{L\Delta_y^2}{2(\nu+1)} + \delta, \tag{36b}$$

where $\delta \triangleq L\epsilon_z^2 + 2\Delta_y\epsilon_\xi$, and for $z \in \mathbb{R}^m$, $(z_+)_i = \max\{z_i, 0\}$. Plugging (36a), (36b) in (35), we arrive at

$$\|\bar{\mathbf{c}}_{(\nu)} - \mathbf{c}^\star\| \leqslant \sqrt{\frac{D^2}{\nu+1} + \delta}. \tag{37}$$

Therefore, if $\frac{D^2}{\nu+1} + \delta \leqslant \lambda^2$, then (31) holds. Rearranging the last inequality, we arrive at (32). □

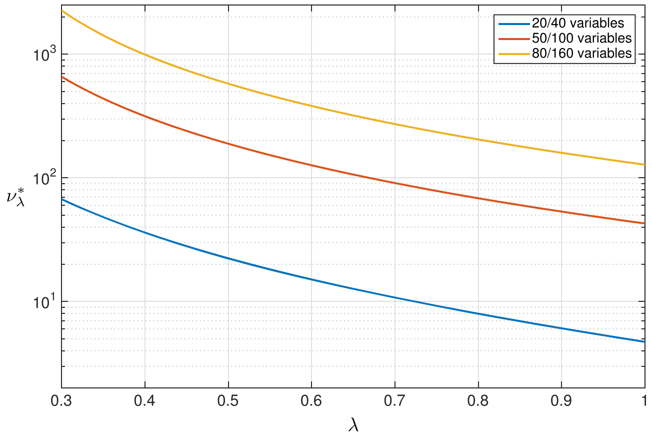


Figure 1. Theoretical bound $\nu^\star_\lambda$ on the maximum number of iterations to achieve a target solution accuracy λ, according to (32), for three sample problems of increasing size.

Theorem 2 has shown that Assumption 1 is satisfied as well by the GPD algorithm, and how the values of λ, $\nu^\star_\lambda$ and p are related.

Remark 4
The estimate on the number of iterations (32) depends on Δ_y given by (26). This entails computing an upper bound on the norm of a dual optimal solution, which is valid for any $x \in \mathcal{D}_N$. Tight uniform bounds (valid for every $x \in \mathcal{D}_N$) can be computed using techniques described in [9]. Specifically, this requires the solution of a linear program with complementarity constraints [9, Theorem 21]. Although linear program with complementarity constraint problems are non-convex, tailored algorithms exist for solving them to global optimality. In embedded MPC, Δ_y can be computed offline, therefore computational time is not a major issue.

Remark 5
So far, we have assumed that (19) is given, which means that the desired value for λ is fixed a priori. Then, by using Theorem 2, the smallest number of iterations $\nu^\star_\lambda$ is found so that λ is smaller or equal than its desired value, for a fixed p which depends on the available hardware. However, in many cases, both p and $\nu^\star_\lambda$ could be fixed a priori, and one might want to find the smallest achievable λ in order to define a control action as close as possible to the optimal one. In this case, one would need to run an iterative procedure as follows. Problem (19) is formulated with $\lambda = \lambda_0 = 0$. Then, by Theorem 2, the corresponding value of $\lambda = \lambda_1$ is found. It is obvious that $\lambda_1 > \lambda_0$, which means that the constraints imposed in (19) would be violated. Therefore, problem (19) is formulated with $\lambda = \lambda_1$, and the whole procedure is repeated iteratively. If $\lambda_{i+1} \leqslant \lambda_i$, then $\lambda = \lambda_i$ is the required value. We would like to remark that extensive simulations have shown that the algorithm terminates at the second iterate (i.e., $\lambda_2 \leqslant \lambda_1$) for most problems.

Figure 1 depicts the computed theoretical bound $\nu^\star_\lambda$, given by (32), on the maximum number of iterations required such that a convergence to a varying desired solution accuracy λ (considered between values of 0.3 and 1) is guaranteed. The plots refer to the solutions of three sample problems of increasing size, with 20, 50, and 80 primal variables and 40, 100, and 160 dual variables, respectively. The fixed-point parameter p is set to 16.

5. COMPARISON WITH A PREVIOUS APPROACH

As already mentioned in the introduction, the approach proposed in this paper can give advantages with respect to that proposed in [19] as the prediction horizon N increases. This is because the shrinked constraint set $\mathcal{Z}_k = \mathcal{Z} \ominus \tilde{\mathcal{R}}_k$, as k increases, tends to $\mathcal{Z} \ominus \tilde{R}_\infty$, which is non-empty if Assumption 2 is satisfied. In [19], instead, the shrinking of the constraint set $\mathcal{Z}$ at the *k-th* prediction

DOI: 10.1002/rnc

step is defined as

$$\mathcal{Z}_k^{\epsilon} \triangleq (1 - k\epsilon)\, \mathcal{Z} = \left\{ z \in \mathbb{R}^{n_z} | F_z z \leqslant (1 - k\epsilon)\, \mathbf{1}_{s_z} \right\} \subseteq \mathcal{Z}, \tag{38}$$

where ϵ is the maximal constraint violation, given the obtained MPC control law. By using the tightened constraints (38), the actual constraints given by $\mathcal{Z}$ are never violated, and theoretical properties analogous to those proved in this paper (i.e., recursive feasibility and stability) hold for [19]. Notice, however, that the size of set $\mathcal{Z}_k^{\epsilon}$ decreases linearly as k increases. Therefore, even if the quality of the solution is rather high (e.g., because the numerical solver can run a large number of iterations), there exists a finite value of the prediction horizon N for which $\mathcal{Z}_{N+1}^{\epsilon} = \emptyset$, and this is the maximum prediction horizon that can be employed. Even if the approach of [19] can be less conservative than the proposed one for relatively small values of N, we can intuitively see that the approach proposed in this paper can be less conservative for relatively high values of N.

Even though the comparison of the performance of the two methods strongly depends on the considered process model, in the remainder of this section, we show numerical results in support of our considerations based on the same system employed in [19], that is,

$$x(t+1) = \begin{bmatrix} 1.09 & 0.22 \\ 0.49 & 0.02 \end{bmatrix} x(t) + \begin{bmatrix} 1.22 & 0.88 \\ -0.78 & -0.34 \end{bmatrix} u(t),$$

and the constraint set $\mathcal{Z}$ is given by the following sets of inequalities

$$\begin{bmatrix} -0.1969 & 0.3132 \\ 0.1531 & -0.3209 \\ -0.1006 & -0.1008 \\ 0.2089 & -0.3352 \\ 0.1032 & 0.0345 \end{bmatrix} x \leqslant \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 0.2068 & -0.1087 \\ -0.2054 & 0.1128 \\ 0.0329 & -0.1573 \\ -0.2014 & 0.0739 \\ 0.2553 & -0.2243 \\ 0.0145 & 0.1016 \\ 0.1569 & -0.0195 \end{bmatrix} u \leqslant \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Given the state weights, input weights, and terminal weights matrices equal to

$$Q = \begin{bmatrix} 5.44 & 5.80 \\ 5.80 & 7.01 \end{bmatrix}, \quad R = \begin{bmatrix} 1.14 & 0.68 \\ 0.68 & 0.62 \end{bmatrix}, \quad P = \begin{bmatrix} 9.46 & 6.38 \\ 6.38 & 7.10 \end{bmatrix},$$

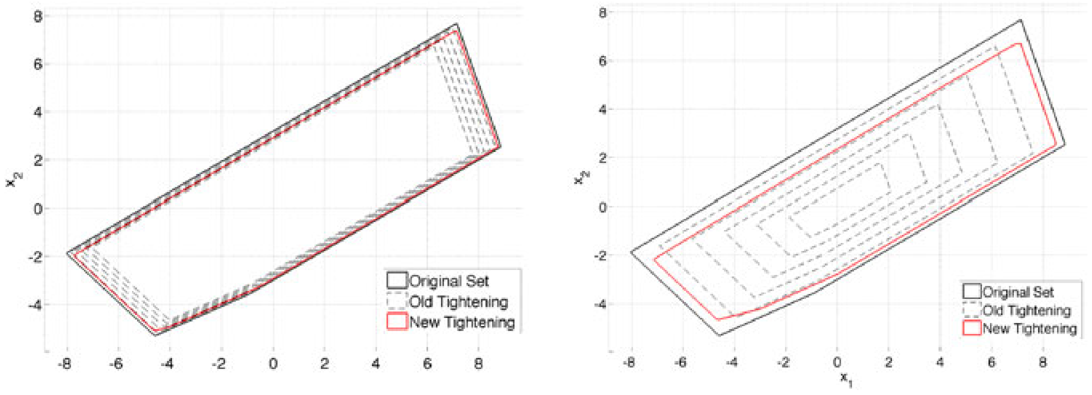
respectively, the resulting linear quadratic (LQR) control action becomes

$$\kappa(x) = \begin{bmatrix} 1.50 & 0.17 \\ -3.39 & -0.48 \end{bmatrix} x.$$

The GPD algorithm runs for $\nu_\lambda^\star = 100$ iterations, assuming infinite numerical precision ($p \to +\infty$). The latter condition is needed for the comparison, because the approach in [19] did not take into account the effect of finite numerical precision. According to these parameters, we build the tightened constraint sets that guarantee recursive feasibility and closed-loop stability, according to (38) (*old* approach, for which we obtain $\epsilon = 2.5 \cdot 10^{-3}$) and to (16) (*new* approach, where $\lambda = 2.78 \cdot 10^{-2}$ is obtained as described in Remark 5).

Figure 2(a) shows the evolution of the state constraint set for $k = 10, 20, 30, 40, 50$ prediction steps. It is clear how, with the old approach, the set shrinks linearly with k. On the other hand, with the new approach, we observe a rapid convergence to a fixed shrinking (in fact, the sets are almost indistinguishable from one another). In this particular case, with a fairly good QP solution quality given by the 100 iterations, the 'old' sets are entirely contained in the 'new' ones only for $N > 40$. For smaller values of N, it can therefore be more convenient to use the approach of [19].

Figure 2(b) shows what happens when the quality of the QP solution is lowered (performing only $\nu_\lambda^\star = 30$ iterations). In this case, we obtain $\lambda = 8.71 \cdot 10^{-2}$ for the new approach, and $\epsilon = 1.56 \cdot 10^{-2}$ for the old approach: the benefits of switching to the new approach become more evident, because the 'old' sets are entirely contained in the 'new' ones already for $N > 18$. Moreover, using the



(a)

(b)

Figure 2. Shrinking of the state constraint set over $k = 10, 20, 30, 40, 50$ prediction steps for (a) good solution quality (100 algorithm iterations) and (b) bad solution quality (30 iterations).

old approach, $N > 62$ makes the feasible set vanish entirely, which makes problem (19) infeasible by definition.

Remark 6

The new approach bases the constraint set reduction on the set $\mathcal{B}_\lambda$, which in turn is proportional to the component-wise maximal QP solution error (cf. [31]). One might argue that this reduction can become more and more conservative as the prediction horizon increases, therefore making the approach unfit for long prediction horizons. It has been observed that this is not the case, because the solution components with larger errors are generally located in the first prediction steps, where input and state constraints are more likely to be active.

6. FIELD-PROGRAMMABLE GATE ARRAY IMPLEMENTATION

As a last contribution of this work, we detail an implementation of the fixed-point GPD algorithm described in [11] on a FPGA device. The theoretical results of the paper at hand are used to guarantee robustness with respect to finite-precision computations. The possible refinement of the employed architectures to achieve higher performance in terms of sampling rates is out of the scope of this paper. However, the reader should be aware that several research groups are currently focused on these implementation aspects. For instance, in [8, 23], high-performance FPGA implementations are proposed for MPC controllers, based on the fast gradient method (for input-constrained problems) and on the alternating direction method of multipliers (for problems with constraints on both inputs and states). Employing a set of design rules leading to efficient implementation of the mentioned algorithms, in [8, 23], sampling rates higher than 1 MHz have been achieved, which allows the use of MPC for processes with very fast dynamics.

6.1. Introduction to field-programmable gate array-based model predictive control

Field-programmable gate arrays are integrated circuits programmable up to the single interconnections between the logic blocks, and are very popular for embedded digital signal processing (DSP) applications because of their speed, scalability, and power efficiency. As previously mentioned, much interest has recently arisen for MPC-on-a-chip architectures based on FPGAs. In addition to the previously mentioned solvers based on gradient projection or alternating direction method of multipliers [8, 23], other algorithms have been proposed based on interior points solvers [24] and active-set solvers [25]. The proposed implementation is supported by the GPD solver running in fixed-point arithmetic. This number representation approach guarantees fast computation times, low delays, and limited chip occupancy as demonstrated in detail in [26]. However, one has to pay the price of reduced numerical precision with the occurrence of *round-off* errors.

DOI: 10.1002/rnc

The main contribution of the proposed implementation is the guarantee of recursive feasibility and closed-loop stability despite numerical errors due to fixed-point number representation. This is achieved as follows: (i) the numbers are represented with a 32-bit fixed-point arithmetic with 16 bits for the fractional part; (ii) for a given number of iterations $\nu_{\lambda}^{\star}$, the results of Theorem 2 are used to define the corresponding MPC problem (19), such that the QP solution is guaranteed to fall within the corresponding target solution accuracy; and (iii) the QP problem is implemented in the FPGA, as described in Section 3, obtaining the aforementioned theoretical results thanks to the results of Theorem 1.

6.2. *GPD algorithm implementation*

The FPGA circuit design was performed according to the graphical approach proposed by *Xilinx System Generator for DSP*, part of the *Xilinx ISE Design Suite* v14.7. With these tools, the single-circuit blocks as multipliers, accumulators, and memories can be placed and connected to each other in the *Simulink* environment. The compiler will then automatically generate the corresponding *VHDL* or *Verilog* code for the FPGA platform of choice. In the proposed implementation, the tests were performed targeting a *Xilinx Kintex 7-xc7k480t*. This is part of the latest 28-nm Kintex generation, and comes with $478K$ logic cells and 1920 DSP slices. We choose to target this device for its fair balance between low cost, low power consumption, and appropriate performance.

Figure 3 shows the top-level view of the QP solver for a sample problem with 2 primal variables and 4 dual variables. The two subsystems are the Matrix-Vector Multiplication (MVM) units performing algorithm steps (29a)–(29b). The output of the first MVM are the primal variables. The accumulator units (blue) multiply the gradient obtained as output of the second MVM by the inverse of the Lipschitz constant and accumulates the result, obtaining the dual variables vector prior to the projection step. Finally, the green units composed by the array of `project.` functions on the left performs the projection, completing step (29c) of the algorithm. Clock signals are pictured in dashed lines, and their behavior will be detailed shortly.

Figure 4 shows the inside of one of the MVM units in Figure 3 (the same structure is used for all MVM blocks). To maximize device compatibility, this block is designed up to the single multipliers/adders/accumulators units, instead of using higher level DSP blocks. This approach requires to individually place blocks for each variable; to automate this process, we developed scripts to build MVM units with arbitrarily large number of variables. For the sake of clarity, Figure 4 shows a small MVM unit that computes $c = Ey + e$, where $E \in \mathbb{R}^{2\times4}$, $y \in \mathbb{R}^4$, and $e, c \in \mathbb{R}^2$. Computations are performed in row-wise parallel fashion.

The path of the computed variables is depicted in green and develops as follows: (i) the left switch selects consecutively the input vector y values; (ii) the current y is split into multiple parallel paths, and each of them is multiplied by the corresponding value of the E matrix rows, stored in the

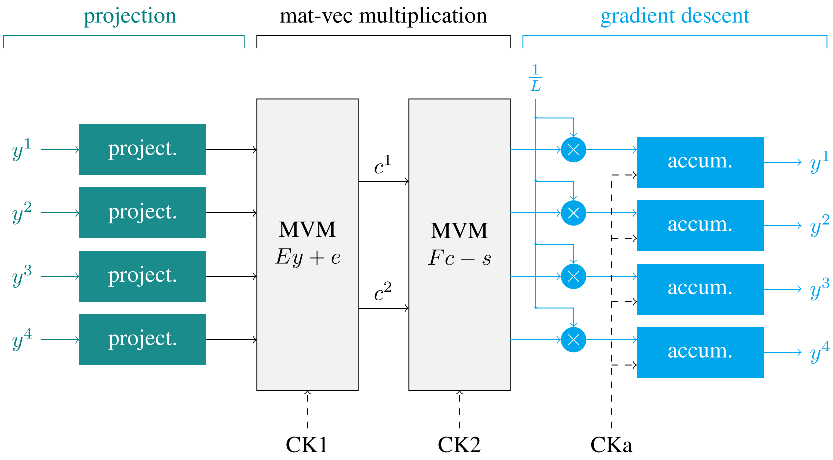


Figure 3. Top-level overview. MVM, Matrix-Vector Multiplication.

DOI: 10.1002/rnc

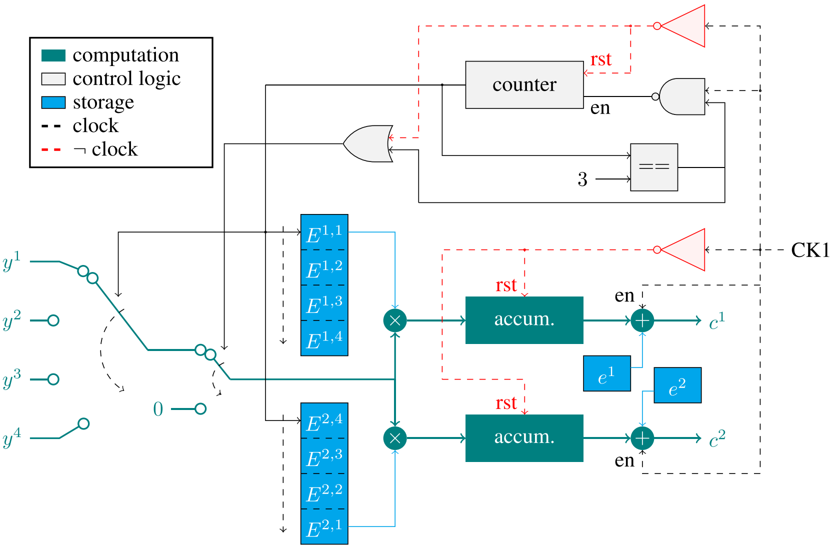


Figure 4. Detail of the Matrix-Vector Multiplication unit.

memory blocks (blue); and (iii) the result is then accumulated obtaining the inner products between the input vector and the matrix rows and added to the corresponding entry of the e vector. The rows of the E matrix are stored in distributed RAM blocks, meaning that they can be placed by the compiler anywhere on the chipset. This is a trade-off that minimizes latency at the cost of increased chip occupancy.

The control logic is depicted in gray. The key element is a counter that directly pilots the input selection of the input switch. Whenever the counter reaches the input size, another switch is triggered, and the multiplier units start to receive the 0 signal, thus stopping the accumulation on the output. Moreover, the output of the `nand` block becomes `FALSE`, disabling the counter itself.

The MVM clock signal is depicted in black dashed lines and its negate in red dashed lines. While `TRUE`, it keeps the adders working. Then, as soon as it turns `FALSE`: (i) the counter resets to 0 and is disabled; (ii) the done switch is set to feed 0 to the downstream units; and (iii) the accumulators reset. As a result, the MVM blackbox behavior works as follows. While the MVM clock is `FALSE`, the unit outputs 0. As soon as a `FALSE`→`TRUE` event is detected, the unit starts reading input variables and computing partial results on the output signals. After $m + 5$ master FPGA clock ticks, where m is the length of the input vector y, the matrix–vector products are ready, and the outputs are kept stable with the final result as long as the MVM clock remains `TRUE`.

Figure 5 shows the evolution of the two MVM clock signals and the accumulator unit clock signal. A single-algorithm iteration is completed in a period T of length equal to $(n + m + 11)$ master FPGA clock cycles (where n and m are the number of primal and dual variables, respectively) and evolves as follows: (i) a `FALSE`→`TRUE` event is triggered on the first MVM unit, which starts it computations starting from the y signals of the previous iteration; (ii) after $(m + 5)$ master clock cycles (green area), the computation is ready, and a `FALSE`→`TRUE` event is triggered on the second MVM unit (in the meanwhile, the first MVM is kept enabled to feed the correct solution to the downstream units); and (iii) after $(m + 5)$ master clock cycles (blue area), all the matrix–vector computations are executed, and a single `TRUE` clock tick is fed to the accumulator unit completing the algorithm iteration.

Remark 7

A key issue when implementing iterative algorithms on a fixed-point architecture is dealing with the occurrence of overflow errors, which happens when a variable becomes larger than the maximum representable value. This value is determined by the number of bits for the integer part.

DOI: 10.1002/rnc

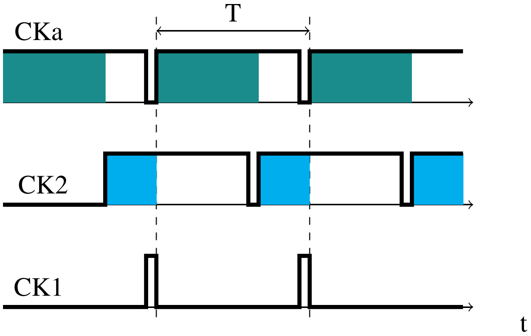


Figure 5. Clock signals for the Matrix-Vector Multiplication units (CK1, CK2) and for the accumulator unit (CKa). Shaded sectors mean that computations are in progress.

Table I. Field-programmable gate array simulations.

Size $[vars/constr]$	**Latency** $[ns]$	**Clock** $[MHz]$	**TPI** $[\mu s]$	**Slice occupation**	**Power** $[W]$
4/8	5.773 ±0.035	173	0.13	1.81%	0.648
8/16	5.773 ±0.035	173	0.2	3.59%	1.102
16/32	5.773 ±0.035	173	0.34	4.77%	8.571

The GPD algorithm chosen for the FPGA simulation is specifically tailored for fixed-point implementation with precise guidelines to avoid overflow errors detailed in [11, Section VI]. The proposed implementation follows those guidelines to ensure an algorithm execution without overflows.

6.3. Field-programmable gate array performance

Table I reports the results of timing and power analysis performed for a *Xilinx Kintex 7-xc7k480t* chipset. The tests are performed starting from randomly generated QP problems of increasing size. Table columns report: (i) the number of primal and dual variables for the QP; (ii) the maximum path latency; (iii) the maximum master clock frequency; (iv) the time needed to complete a single-algorithm iteration; (v) the percentage of occupied slices; and (vi) the power consumption. Because of parallelization and pipelining, the maximum path latency is not affected by the problem size, and the computation time grows only linearly with size in spite of the quadratic complexity of the matrix–vector operations.

Figure 6 shows the state evolution of a simulated physical system when connected in closed-loop to the GPD-based predictive controller. The system is composed of three masses connected by springs and dampers to two actuators placed between them (cf. [27]). The system dynamics (discretized with sampling time $T_s = 0.5$ s) evolve as follows:

$$x(t+1) = \begin{bmatrix} 0.6826 & 0.1163 & 0.0229 & 0.2815 & 0.0823 & 0.0147 \\ 0.1163 & 0.7055 & 0.1163 & 0.0823 & 0.2962 & 0.0823 \\ 0.0229 & 0.1163 & 0.6826 & 0.0147 & 0.0823 & 0.2815 \\ -0.9614 & 0.2634 & 0.1056 & 0.2019 & 0.2480 & 0.0757 \\ 0.2634 & -0.8558 & 0.2634 & 0.2480 & 0.2776 & 0.2480 \\ 0.1056 & 0.2634 & -0.9614 & 0.0757 & 0.2480 & 0.2019 \end{bmatrix} x(t) +$$

$$+ \begin{bmatrix} 0.0716 & 0.0135 \\ -0.0736 & 0.0736 \\ -0.0135 & -0.0716 \\ 0.1992 & 0.0675 \\ -0.2140 & 0.2140 \\ -0.0675 & -0.1992 \end{bmatrix} u(t).$$

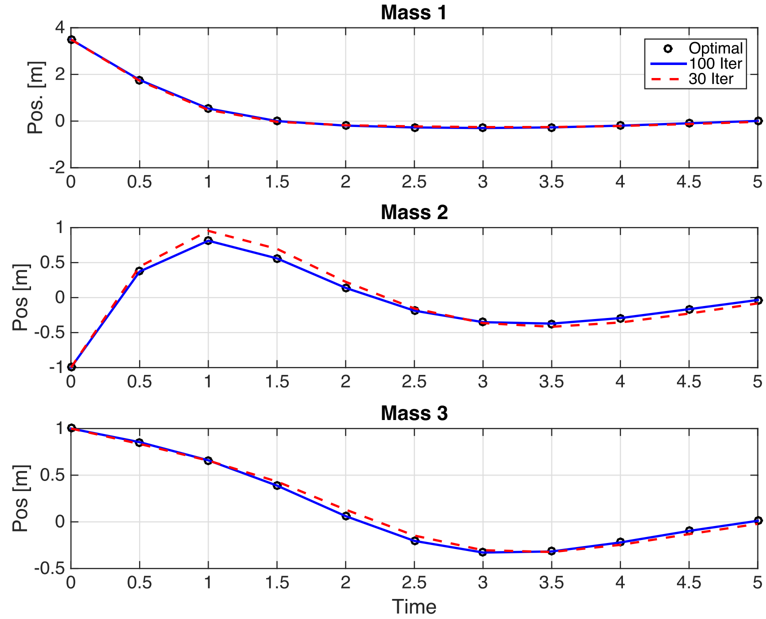


Figure 6. Closed-loop evolutions driven by optimal controller based on *CPLEX* solver and by the proposed GPD-based controller.

The states are constrained as $|x_i(t)| \leqslant 4,\ i = 1, .., 6$, and the inputs as $|u_i(t)| \leqslant 1,\ i = 1, 2$. By setting $N = 10$, the resulting QP problem has 20 and 208 primal and dual variables, respectively. We compare the closed-loop trajectories for mass displacement when driven by a reference controller based on the *CPLEX* solver, with the trajectories driven by two GPD-based controllers, first with high solution accuracy (letting the algorithm run for 100 iterations) then with low solution accuracy (algorithm running for 30 iterations). The initial condition is $x(0) = \begin{bmatrix} 3.5 & -1 & 1 & -3 & 3 & 0 \end{bmatrix}'$, where the first three states are the initial positions and the last three states are the initial velocities. Setting the FPGA clock to 100 MHz, the solution computation times are 239 μs (high accuracy) and 72 μs (low accuracy). Results show how the system evolutions with the sub-optimal controllers are remarkably close to the evolutions with the reference *CPLEX* controller, even for the 30 iterations case.

7. CONCLUSIONS

The paper has introduced an MPC law for linear systems with polytopic mixed constraints. The main contributions of the paper are the following:

- The formulation of an alternative problem using a robust MPC approach to take into account the combined action of external disturbances, of numerical inaccuracies, and of an a priori fixed number of iterations of the numerical solver has never been proposed in the literature, at least to the best of the authors' knowledge. The alternative problem is formulated by extending the contribution of [20] to systems with mixed constraints.
- The formulation of the bound in (32) is also a novel contribution, allowing one to connect the approximation error resulting from a specific solver with the degree of uncertainty that it brings into the system.
- The comparison with [19] shows the improvement that the proposed method can bring with respect to a state-of-the-art approach in case of long prediction horizons.
- As a last contribution, we demonstrate that a stabilizing MPC based on the GPD algorithm can be easily implemented on an FPGA and provide detailed guidelines regarding the actual implementation.

DOI: 10.1002/rnc

APPENDIX

A.1. Definitions regarding the asymptotic stability of a set

The following definitions are introduced in order to precisely describe the properties of the closed-loop system that are proved in Theorem 1. Consider a discrete-time autonomous nonlinear system

$$x(t+1) = \varphi(x(t), w(t)), \tag{39}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $w \in \mathbb{R}^{n_w}$ is the input vector, defined such that $w(t) \in \mathcal{W}$ ($\mathcal{W}$ being a compact set including the origin), and $\varphi(\cdot)$ is a nonlinear function. The following definitions, adapted from [28], are recalled for the reader's convenience. Consider two closed RPI sets $\hat{\mathcal{X}}$ and $\mathcal{X}$, $\mathcal{X} \subseteq \hat{\mathcal{X}} \subset \mathbb{R}^{n_x}$. Then,

Definition 1 (attractivity)
The set $\mathcal{X}$ is *attractive* for system (39), with *domain of attraction* $\hat{\mathcal{X}}$, if, for all $x(0) \in \hat{\mathcal{X}}$, $\lim_{t\to\infty} \delta_h(x(t), \mathcal{X}) = 0$.

Definition 2 (local stability)
The set $\mathcal{X}$ is *locally stable* for system (39) if, for all $\epsilon \in \mathbb{R}_{>0}$, there exists $\delta \in \mathbb{R}_{>0}$ such that, for each $x(0)$ satisfying $x(0) \in \mathcal{X} \oplus \mathfrak{B}_\delta$, one has $x(t) \in \mathcal{X} \oplus \mathfrak{B}_\epsilon$, for all $t \in \mathbb{N}_{\geq 0}$.

Definition 3 (asymptotic stability)
The set $\mathcal{X}$ is *asymptotically stable* for system (39), with *domain of attraction* $\hat{\mathcal{X}}$, if it is locally stable and attractive with domain of attraction $\hat{\mathcal{X}}$.

A.2. Proof of Theorem 1

Part (1): We will first prove positive invariance of $\mathcal{D}_N$ for (22), that is, if $x \in \mathcal{D}_N$, then $\varphi(x) \in \mathcal{D}_N$, or, equivalently, $\mathcal{S}_N(\varphi(x)) \neq \emptyset$ for every $x \in \mathcal{D}_N$ and every $\mu(x) \in \mathcal{C}(x)$. Therefore, it is enough to find a vector $\tilde{\mathbf{z}} \in \mathcal{S}_N(\varphi(x))$ for any $x \in \mathcal{D}_N$, with $\bar{\mathbf{z}}(x) \in Z(x)$. Because $x \in \mathcal{D}_N$, then the optimal solution $\mathbf{z}^\star$ is such that

$$z_k^\star = \begin{bmatrix} x_k^\star \\ K x_k^\star + c_k^\star \end{bmatrix} \in \mathcal{Z}_k, k \in \mathbb{N}_{N-1} \tag{40}$$

and $x_N^\star \in \mathcal{X}_f \ominus \mathcal{R}_N$. Instead, for $x \in \mathcal{D}_N$, we will be able to apply $\bar{u}_0(x) = Kx + \mu(x) \in \mathcal{C}(x)$, which will move the state to $\varphi(x) = A_\kappa x + B\mu(x) = A_\kappa x + Bc_0^\star(x) + w_0$, with $w_0 \in \mathcal{W}$. In order to determine a feasible solution $\tilde{\mathbf{z}} \in \mathcal{S}_N(\varphi(x))$, we define the 'shifted' control vector

$$\tilde{\mathbf{c}}(x) = \left[\tilde{c}_0(x) \;\dots\; \tilde{c}_{N-2}(x) \;\tilde{c}_{N-1}(x)\right] = \left[c_1^\star(x) \;\dots\; c_{N-1}^\star(x) \;0\right]$$

which defines $\tilde{\mathbf{z}} = \left[\tilde{z}_0' \cdots \tilde{z}_{N-1}' \;\tilde{x}_N'\right]'$, with $\tilde{z}_k = \left[\tilde{x}_k' \;K'\tilde{x}_k' + \tilde{c}_k'\right]'$, $k \in \mathbb{N}_{[0,N-1]}$. We prove now that $\tilde{\mathbf{z}}$ is a feasible solution for an initial state equal to $\varphi(x)$. Applying the shifted control vector $\tilde{\mathbf{c}}(x)$, we obtain

$$\tilde{z}_k = \begin{bmatrix} \tilde{x}_k \\ K\tilde{x}_k + \tilde{c}_k \end{bmatrix} = z_{k+1}^\star + \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k w_0 \tag{41}$$

From (40) and (41), one has, for $k = 0, \dots, N-2$,

$$\begin{aligned} \tilde{z}_k &\in \mathcal{Z}_{k+1} \oplus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W} = \left(\mathcal{Z} \ominus \tilde{\mathcal{R}}_{k+1}\right) \oplus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W} \\ &= \left[\left(\mathcal{Z} \ominus \tilde{\mathcal{R}}_k\right) \ominus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W}\right] \oplus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W} = \left(\mathcal{Z}_k \ominus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W}\right) \oplus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^k \mathcal{W} \subseteq \mathcal{Z}_k. \end{aligned}$$

Again, by definition of $\tilde{\boldsymbol{c}}(x)$, we have that $\tilde{z}_{N-1} = \left[\tilde{x}'_{N-1}\ (K\tilde{x}_{N-1})'\right]'$, and by feasibility of $\mathbf{z}^\star(x)$ we know that $x^\star_N \in \mathcal{X}_f \ominus A_\kappa^N \mathcal{W} \subseteq \mathcal{X}_N$. By recalling Remark 2, we obtain $z^\star_N \triangleq \left[x^{\star\prime}_N\ \left(Kx^\star_N\right)'\right]' \in \mathcal{Z}_N$. Then, we can proceed as in (42), obtaining

$$\tilde{z}_{N-1} \in \mathcal{Z}_N \oplus \begin{bmatrix} I \\ K \end{bmatrix} A_\kappa^{N-1}\mathcal{W} \subseteq \mathcal{Z}_{N-1}.$$

Finally, to prove that $\tilde{x}_N \in \mathcal{X}_f \ominus \mathcal{R}_N$, we start investigating the properties of $\tilde{x}_{N-1}$:

$$\tilde{x}_{N-1} \in \left(\mathcal{X}_f \ominus \mathcal{R}_N\right) \oplus A_\kappa^k\mathcal{W} = \left[\left(\mathcal{X}_f \ominus \mathcal{R}_{N-1}\right) \ominus A_\kappa^k\mathcal{W}\right] \oplus A_\kappa^k\mathcal{W} \subseteq \mathcal{X}_f \ominus \mathcal{R}_{N-1}. \tag{42}$$

Being $\mathcal{X}_f$ an RPI set with respect to dynamics (13), we know that, given $x \in \mathcal{X}_f$, $A_\kappa^j x \oplus \mathcal{R}_j \subseteq \mathcal{X}_f$, or, equivalently, $A_\kappa^j x \in \mathcal{X}_f \ominus \mathcal{R}_j$, for all $j \in \mathbb{N}_{\geqslant 0}$ (remember that $\mathcal{X}_f \ominus \mathcal{R}_j \neq \emptyset$, because $\mathcal{R}_\infty \subset \mathcal{X}_f$). Therefore, given $x \in \mathcal{X}_f \ominus \mathcal{R}_{j-1}$, then $A_\kappa x \in \mathcal{X}_f \ominus \mathcal{R}_j$, for all $j \in \mathbb{N}_{\geqslant 1}$. In conclusion, $\tilde{x}_{N-1} \in \mathcal{X}_f \ominus \mathcal{R}_{N-1}$ implies $\tilde{x}_N = A_\kappa \tilde{x}_{N-1} \in \mathcal{X}_f \ominus \mathcal{R}_N$. Recursive feasibility is therefore proved.

Part (ii): From the positive invariance of $\mathcal{D}_N$, being $\mu(x(t)) \in \mathcal{C}(x(t))) \neq \emptyset$, one has $(x(t), \mu(x(t))) \in \mathcal{Z}$, for all $t \in \mathbb{N}_{\geqslant 0}$.

Part (iii): As a first step, we prove that for any initial condition $x(0) \in \mathcal{D}_N$, $x(t)$ converges to $\mathcal{X}_f$ in a finite time. In order to do that, we consider the optimal value of the cost function $V_N(\boldsymbol{c})$, evaluated at $x(t)$ and referred to as $V_N^\star(t)$ for the sake of readability. At time $t+1$, the existence of a new optimal value $V_N^\star(t+1)$ for any realization of $w(t) \in \mathcal{W}$ is guaranteed by recursive feasibility, but its explicit expression is not available. However, using the standard procedure used also in [20], we consider the suboptimal value of the cost function, corresponding to the shifted control sequence $\tilde{\boldsymbol{c}}(x(t))$, that is,

$$\tilde{V}_N(t+1) = \sum_{k=0}^{N-1} \tilde{c}'_k \Psi \tilde{c}_k = \sum_{k=1}^{N-1} c_k^{\star\prime} \Psi c_k^\star = V_N^\star(t) - \sigma'_t \Psi \sigma_t$$

where $\sigma_t = c_0^\star(x(t))$. We know that the optimal cost function at any $x(t+1) \in A_\kappa x(t) + B\sigma_t + w(t)$ will be such that $V_N^\star(t+1) \leqslant \tilde{V}_N(t+1)$, which implies $V_N^\star(t) - V_N^\star(t+1) \geqslant \sigma'_t\Psi\sigma_t \geqslant 0$. Therefore, $\left\{V_N^\star(t)\right\}$ is a non-negative non-increasing sequence, which converges to a finite value $V_N^\star(\infty)$ as $t \to \infty$. If we consider the sum of the terms $V_N^\star(t) - V_N^\star(t+1)$, for $t = 0, \ldots, \infty$, we obtain

$$\infty > V_N^\star(0) - V_N^\star(\infty) \geqslant \sum_{t=0}^{\infty} \sigma'_t \Psi \sigma_t \geqslant 0$$

which implies $\lim_{t\to\infty} \sigma'_t\Psi\sigma_t = 0$, and, by positive definiteness of Ψ, $\lim_{t\to\infty}\sigma_t = 0$. Consider now that, at any $t \in \mathbb{N}_{\geqslant 0}$, the uncertain term $w(t)$ is also applied to the system. Recalling that $\rho(A_\kappa) < 1$,

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty}\left[A_\kappa^t x(0) + \sum_{k=1}^{t} A_\kappa^{k-1}\left(B\sigma_{t-k} + w(t-k)\right)\right] = \lim_{t\to\infty}\left[\sum_{k=1}^{t} A_\kappa^{k-1} w(t-k)\right] \in \mathcal{R}_\infty.$$

Recalling Assumption 3, the asymptotic convergence of $x(t)$ to $\mathcal{R}_\infty$ implies that there exists $\bar{t} \in \mathbb{R}_{\geqslant 0}$ such that $x(\bar{t}) \in \mathcal{X}_f$. Because $\mathcal{X}_f$ is a RPI set, by (21), the applied control law coincides with $\kappa(x)$ (which in turn, coincides with the optimal control law) and then $e(t) = 0$ for all $t \geqslant \bar{t}$. From $\bar{t}$ on, the system dynamics are therefore

$$x(t+1) = A_\kappa x(t) + d(t) \in \mathcal{X}_f. \tag{43}$$

As a consequence, the state converges asymptotically to the minimal RPI set for system (43), that is $\mathcal{R}_D$ in (15). Together with the finite-time convergence to $\mathcal{X}_f$, this implies that the set $\mathcal{R}_D$ is attractive for system (22) with domain of attraction $\mathcal{D}_N$, according to Definition 1.

In order to prove local stability, we consider the closed-loop dynamics (43) in $\mathcal{X}_f$ and take any initial condition $x(0) \in \mathcal{R}_D \oplus \mathfrak{B}_\delta$. Being $\mathcal{R}_D$ an RPI set for (43), we have that $A_\kappa \mathcal{R}_D \oplus \mathcal{D} \subseteq \mathcal{R}_D$, and therefore, iterating the system dynamics

$$x(t) \in \mathcal{R}_D \oplus A_\kappa^t \mathfrak{B}_\delta. \tag{44}$$

Being $\rho(A_\kappa) < 1$, Definition 2 holds for the nominal system (5), the set $\mathcal{X}$ being the origin. As a consequence, for all $\epsilon > 0$, there exists $\delta > 0$, such that $A_\kappa^t \mathfrak{B}_\delta \subseteq \mathfrak{B}_\epsilon$. By substituting this inside (44), we conclude that $x(t) \in \mathcal{R}_D \oplus \mathfrak{B}_\epsilon$, meaning that $\mathcal{R}_D$ is locally stable for system (43), according to Definition 2, for any ϵ such that $\mathcal{R}_D \oplus \mathfrak{B}_\epsilon \subseteq \mathcal{X}_f$

In conclusion, Definition 3 holds for system (22), which proves that $\mathcal{R}_D$ is asymptotically stable with domain of attraction $\mathcal{D}_N$.

REFERENCES

1. Rawlings JB, Mayne DQ. *Model Predictive Control: Theory and Design*. Nob Hill Publishing: Madison, WI, 2009.
2. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. *Automatica* 2002; **38**(1):3–20.
3. Richter S, Jones CN, Morari M. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control* 2012; **57**(6):1391–1403.
4. Nesterov Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, Vol. 27, 1983; 372–376.
5. Nesterov Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer: New York City, NY, 2004.
6. Necoara I, Suykens J. Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control* 2008; **53**(11):2674–2679.
7. Richter S, Morari M, Jones CN. Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method. *Proceedings of the IEEE Conference on Decision and Control*, Orlando, USA, 2011; 5223–5229.
8. Jerez JL, Goulart PJ, Richter S, Constantinides GA, Kerrigan EC, Morari M. Embedded predictive control on an FPGA using the fast gradient method. *Proceedings of the European Control Conference*, Zurich, Switzerland, 2013; 3614–3620.
9. Patrinos P, Bemporad A. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control* 2014; **59**(1):18–33.
10. Bemporad A, Patrinos P. Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. *IFAC Conference on Nonlinear Model Predictive Control*, Noordwijkerhout, Netherlands, 2012; 14–20.
11. Patrinos P, Guiggiani A, Bemporad A. Fixed-point dual gradient projection for embedded model predictive control. *Proceedings of the European Control Conference*, Zurich, Switzerland, 2013; 3602–3607.
12. McGovern LK, Feron E. Closed-loop stability of systems driven by real-time, dynamic optimization algorithms. *Proceedings of the IEEE Conference on Decision and Control*, Phoenix, AZ, 1999; 3690–3696.
13. Giselsson P. Execution time certification for gradient-based optimization in model predictive control. *IEEE Conference on Decision and Control*, Maui, HI, 2012; 3165–3170.
14. Necoara I, Ferranti L, Keviczky T. An adaptive constraint tightening approach to linear model predictive control based on approximation algorithms for optimization. *Optimal Control Applications and Methods* 2015; **36**(5): 648–666.
15. Necoara I. Computational complexity certification for dual gradient method: application to embedded MPC. *Systems & Control Letters* 2015; **81**:49–56.
16. Giselsson P, Boyd S. Preconditioning in fast dual gradient methods. *IEEE Conference on Decision and Control*, Los Angeles, CA, 2014; 5040–5045.
17. Doan MD, Keviczky T, De Schutter B. A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraint. *IEEE Conference on Decision and Control*, Orlando, FL, 2011; 5236–5241.
18. Necoara I, Nedelcu V. Rate analysis of inexact dual first-order methods: application to dual decomposition. *IEEE Transactions on Automatic Control* 2014; **59**(5):1232–1243.
19. Rubagotti M, Patrinos P, Bemporad A. Stabilizing linear model predictive control under inexact numerical optimization. *IEEE Transactions on Automatic Control* 2014; **59**(6):1660–1666.
20. Chisci L, Rossiter JA, Zappa G. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica* 2001; **37**(7):1019–1028.
21. Raković SV, Kerrigan EC, Kouramas KI, Mayne DQ. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control* 2005; **50**(3):406–410.
22. Bertsekas DP, Nedic A, Ozdaglar AE. *Convex analysis and optimization*. Athena Scientific: Nashua, NH, 2003.
23. Jerez JL, Goulart PJ, Richter S, Constantinides GA, Kerrigan EC, Morari M. Embedded online optimization for model predictive control at megahertz rates. *IEEE Transactions on Automatic Control* 2014; **59**(12):3238–3251.

DOI: 10.1002/rnc

24. Jerez JL, Constantinides GA, Kerrigan EC. An FPGA implementation of a sparse quadratic programming solver for constrained predictive control. *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2011; 209–218.
25. Wills AG, Knagge G, Ninness B. Fast linear model predictive control via custom integrated circuit architecture. *IEEE Transactions on Control Systems Technology* 2012; **20**(1):59–71.
26. Kerrigan EC, Jerez JL, Longo S, Constantinides GA. Number representation in predictive control. *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control*, Noordwijkerhout, Netherlands, 2012; 60–67.
27. Wang Y, Boyd S. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology* 2010; **18**(2):267–278.
28. Rawlings JB, Mayne DQ. Postface to Model Predictive Control: Theory and Design, 2011. (Available from: http://jbrwww.che.wisc.edu/home/jbraw/mpc/postface.pdf) [accessed on 19 August 2012].

DOI: 10.1002/rnc