Automatica 55 (2015) 226–235

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A dual gradient-projection algorithm for model predictive control in fixed-point arithmetic☆

Panagiotis Patrinos, Alberto Guiggiani, Alberto Bemporad

IMT - Institute for Advanced Studies, Piazza San Ponziano 6, 55100 Lucca, Italy

ARTICLE INFO

Article history:
Received 12 March 2013
Received in revised form
25 February 2015
Accepted 27 February 2015
Available online 31 March 2015

Keywords:
Embedded systems
Convex optimization
Predictive control

ABSTRACT

Although linear Model Predictive Control has gained increasing popularity for controlling dynamical systems subject to constraints, the main barrier that prevents its widespread use in embedded applications is the need to solve a Quadratic Program (QP) in real-time. This paper proposes a dual gradient projection (DGP) algorithm specifically tailored for implementation on fixed-point hardware. A detailed convergence rate analysis is presented in the presence of round-off errors due to fixed-point arithmetic. Based on these results, concrete guidelines are provided for selecting the minimum number of fractional and integer bits that guarantee convergence to a suboptimal solution within a pre-specified tolerance, therefore reducing the cost and power consumption of the hardware device.

1. Introduction

Model Predictive Control (MPC) technology is widely popular in many industrial applications due to explicit performance optimization, and its straightforward handling of constraints on inputs, outputs and states (Bemporad, 2006; Mayne & Rawlings, 2009). An MPC controller relies on solving a Quadratic Program to minimize input efforts and the difference between predicted outputs and desired set-points. The fact that a QP needs to be solved within each sampling period has initially limited the diffusion of MPC technologies to low-bandwidth applications where high computational resources are available, as in the chemical and refinery industries. However, in the last years an increasing interest in embedded MPC solutions is spreading in many other industries, such as automotive and aerospace.

Embedding MPC on a hardware platform poses quite a few challenges, both from a system-theoretic and an optimization point of view. Specifically, the main requirements that make a QP solver suitable for embedded MPC are the following: (a) the algorithm should be simple enough to be implemented on simple hardware platforms; (b) one must be able to compute a bound on its worst-case execution time for computing a (reasonably good) solution; (c) stability and invariance guarantees for the resulting closed-loop system must be provided despite suboptimality and/or infeasibility of the solution; (d) the algorithm should be robust to low precision arithmetic, i.e., the effect of round-off errors should be small, no overflow should occur, and one should be able to determine *a priori* the behavior of the algorithm under such hypotheses.

Ling et al. detailed in Ling, Yue, and Maciejowski (2006) an FPGA implementation of an interior-point method for solving the QP problem, showing that the "MPC-on-a-chip" idea is indeed viable. Later, Knagge et al. proposed an active-set QP solver for ASIC and FPGA (Knagge, Wills, Mills, & Ninness, 2009), and tested it for MPC control of nonlinear systems. A "QP-on-a-chip" controller implemented on FPGA with an iterative linear solver was tested in hardware-in-the-loop experiments in Hartley et al. (2012).

All of the solvers proposed in such contributions require *floating-point* numbers. However, when trying to minimize computational effort, power consumption, and chip size, a great positive impact is given by the choice of *fixed-point* number representation (Kerrigan, Jerez, Longo, & Constantinides, 2012). Nevertheless, this significant improvement in performance comes at the price of a reduced range in which numbers can be represented and round-off errors (Wilkinson, 1994). Because of this, algorithms that perform well in floating-point may perform much worse (even completely wrongly) in fixed-point. Therefore, additional challenges arise when dealing with fixed-point arithmetic, mainly studying round-off error accumulation during algorithm iterations, and establishing bounds on the magnitude of the computed variables to avoid overflows. In Jerez, Constantinides, and Kerrigan (2012) an implementation of a modified interior-point solver in

☆ The material in this paper was presented at the 2013 European Control Conference, July 17–19, 2013, Zurich, Switzerland. This paper was recommended for publication in revised form by Editor Frank Allgöwer.

E-mail addresses: panagiotis.patrinos@imtlucca.it (P. Patrinos), alberto.guiggiani@imtlucca.it (A. Guiggiani), alberto.bemporad@imtlucca.it (A. Bemporad).

http://dx.doi.org/10.1016/j.automatica.2015.03.002
0005-1098/

fixed-point is presented. The authors focus on the solution of the linear system required in each algorithm iteration, and propose a preconditioning technique tailored to prevent overflow errors as well as a detailed analysis of the effects of the round-off error.

Recently, the use of first-order methods, and in particular fast gradient methods developed by Nesterov (2004), has been advocated as a viable candidate for embedded optimization-based control (Bemporad & Patrinos, 2012; Patrinos & Bemporad, 2012; Richter, Jones, & Morari, 2009; Richter, Morari, & Jones, 2011). These methods can compute a suboptimal solution in a finite number of iterations, which can be bounded *a priori*, and they are simple enough (usually requiring only matrix–vector products) for hardware implementation. In particular, the accelerated DGP method proposed in Bemporad and Patrinos (2012) and Patrinos and Bemporad (2012), called GPAD, can be applied to linear MPC problems with general polyhedral constraints and with guaranteed global *primal* convergence rates. In Rubagotti, Patrinos, and Bemporad (2014) results of Patrinos and Bemporad (2012) are exploited to show how GPAD can be used in MPC to provide invariance, stability and performance guarantees in a finite number of iterations for the closed-loop system.

1.1. Contribution

In this work, we propose a DGP method, which can be seen as a simplified (non-accelerated) version of GPAD, specifically tailored for fixed-point implementation. The main contribution of this work is that we provide a detailed convergence rate and asymptotic error analysis in terms of *primal cost and primal feasibility* in the presence of round-off errors due to fixed-point arithmetic, thus addressing successfully the last of the requirements described previously for embedded optimization-based control. In addition to that, we give specific guidelines on the number of fractional bits that certify the convergence to a target suboptimal solution, as well as on the number of integer bits to avoid overflow errors. The machinery we use to perform the analysis is based on the notion of the *inexact oracle* proposed by Devolder, Glineur, and Nesterov (2013). However, directly applying the results of Devolder et al. (2013) to our dual gradient projection method would only provide us with convergence rate estimates about the quality of the *dual and not the primal iterates* of the algorithm.

The reason for limiting the analysis to the non-accelerated version of GPAD is that accelerated methods suffer from error accumulation, as shown in Devolder et al. (2013). In Nedelcu and Necoara (2012) and Nedelcu, Necoara, and Dinh (2013) the authors analyze the convergence rate of inexact gradient augmented Lagrangian methods for constrained MPC, where the source of inexactness comes from suboptimal solution of the so called inner problem. In the present work, the source of inexactness comes from round-off errors due to the fixed-point implementation.

1.2. Structure of the paper

After introducing some notation at the end of this section and motivating the work in Section 2, in Section 3 we give general theoretical results when a gradient projection (GP) algorithm runs with an inexact oracle. In Section 4 an inexact DGP method is applied to a modified version of the dual problem and its convergence rate with respect to primal suboptimality and infeasibility is analyzed. In Section 5, the general results of the proposed inexact DGP method are applied to the case of QP based on a fixed-point implementation. Simulation results and experiments on low-cost hardware boards are presented in Section 6. Finally, conclusions are drawn in Section 7.

The main technical contribution of this paper has appeared in Patrinos, Guiggiani, and Bemporad (2013) without providing the proofs of the theoretical results, that are provided here in full detail.

The notation adopted throughout the paper is standard. Let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ denote the sets of real numbers, nonnegative integers, column real vectors of length n, and m by n real matrices, respectively. The transpose of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by A'. For any nonnegative integers $k_1 \le k_2$, the finite set $\{k_1, \dots, k_2\}$ is denoted by $\mathbb{N}_{[k_1,k_2]}$. For $z \in \mathbb{R}^n$, $\Pi_Z(z)$ denotes its Euclidean projection on the set $Z \subseteq \mathbb{R}^n$, while $[z]_+$ denotes its Euclidean projection on the nonnegative orthant, i.e., the vector whose ith coordinate is $\max\{z_i, 0\}$. For a vector $z \in \mathbb{R}^n$, $\|z\|$ and $\|z\|_\infty$ denote the Euclidean and infinity norm of z respectively, while if $A \in \mathbb{R}^{m\times n}$, $\|A\|$ denotes the spectral norm of A (unless otherwise stated).

2. Motivation

When performing computations on low-cost, low-power embedded devices, the adoption of a fixed-point number representation can have a great positive impact in terms of computational speed. However, this comes at the price of a reduced precision and a reduced range when compared to floating-point representation, leading to the occurrence of round-off and overflow errors.

Suppose that an algorithm is running on a fixed-point hardware with a scaling factor 2^{-p}, where $p \in \mathbb{N}_+$ is the number of fractional bits, and assume that real numbers are represented in fixed-point by rounding to the closest value. Therefore, the resolution (i.e., the smallest representable non-zero magnitude) of a fixed-point number is equal to $2^{-(p+1)}$.

It is obvious that addition and subtraction do not result in any loss of accuracy due to rounding. However, multiplication can suffer from rounding. In specific, multiplying two scalars $\zeta = \gamma\xi$ leads to the fixed-point representation $\mathtt{fi}(\zeta)$ of ζ, with $|\zeta - \mathtt{fi}(\zeta)| \le 2^{-(p+1)}$.

For $x, y \in \mathbb{R}^n$ let $\mathtt{fi}(x'y) \triangleq \sum_{i=1}^n \mathtt{fi}(x_i y_i)$. Then the round-off error for the inner product of x and y can be bounded as follows:

$$|x'y - \mathtt{fi}(x'y)| \le 2^{-(p+1)}n. \tag{1}$$

If A is an $m \times n$ matrix and x is an n-vector, then

$$\|Ax - \mathtt{fi}(Ax)\|_\infty \le 2^{-(p+1)}n. \tag{2}$$

Quadratic Programming algorithms based on Gradient Projection method require, at each iteration, the computation of the gradient for the cost function. In a fixed-point architecture, instead of the exact gradient $\nabla\Phi(\cdot)$, we have access to an approximate formulation $\tilde{\nabla}\Phi(\cdot)$.

Convergence proofs have therefore to be reformulated in order to take into account of this approximation. In addition to that, it is of interest to find direct links between the fixed-point precision and bounds on the gradient error, as well as solution quality. Finally, bounds on the magnitude of all the variables are required such that the number of integer bits can be chosen to avoid the occurrence of overflow errors. All these topics will be covered in the next sections.

3. Inexact gradient projection

Consider the problem

$$\begin{aligned} &\text{minimize} \quad \Phi(y) \\ &\text{subject to } y \in Y, \end{aligned} \tag{3}$$

where Y is a nonempty closed convex subset of $\mathbb{R}^m$, and $\Phi : \mathbb{R}^m \to \mathbb{R}$ is convex, L_Φ-smooth, i.e., there exists a $L_\Phi > 0$ such that

$$\|\nabla\Phi(y) - \nabla\Phi(w)\| \le L_\Phi\|y - w\|, \quad y, w \in \mathbb{R}^m.$$

We assume that $\Phi^\star \triangleq \inf_{y\in Y}\Phi(y)$ is finite and $Y^\star \triangleq \operatorname{argmin}_{y\in Y}\Phi(y)$ is nonempty. The goal is to find an approximate solution of (3) by applying the GP method

$$y_{(\nu+1)} = \Pi_Y\left(y_{(\nu)} - \tfrac{1}{L_\Phi}\nabla\Phi(y_{(\nu)})\right). \tag{4}$$

However, it is assumed that the gradient of Φ cannot be computed exactly. Instead, we have at our disposal an inexact oracle (Devolder et al., 2013), according to the following definition.

Definition 1. $\Phi:\mathbb{R}^m\to\mathbb{R}$ is equipped with a *first-order* (δ, L)-*oracle* if for any $w\in\mathbb{R}^m$ one can compute a pair $(\Phi_{\delta,L}(w), s_{\delta,L}(w)) \in \mathbb{R}\times\mathbb{R}^m$ such that

$$0 \le \Delta_{\delta,L}(y;w) \le \frac{L}{2}\|y-w\|^2 + \delta, \quad \forall y\in\mathbb{R}^m, \tag{5}$$

where $\Delta_{\delta,L}(y;w) \triangleq \Phi(y) - \ell_{\delta,L}(y;w)$, and $\ell_{\delta,L}(y;w) \triangleq \Phi_{\delta,L}(w) + s_{\delta,L}(w)'(y-w)$.

Now, there is a difference in the implementation of GP (4) with the inexact oracle

$$y_{(\nu+1)} = \Pi_Y\left(y_{(\nu)} - \tfrac{1}{L}s_{\delta,L}(y_{(\nu)})\right). \tag{6}$$

Notice that in the inexact GP method (6), $s_{\delta,L}(y_{(\nu)})$ is used instead of $\nabla\Phi(y_{(\nu)})$ and the constant L, appearing in (5), is used instead of the Lipschitz constant L_Φ.

We will now introduce two lemmas, essential in proving convergence rates for both primal and dual versions of inexact GP; the first is a direct extension of Chen and Teboulle (1993, Lemma 3.2) in the inexact case and therefore its proof is omitted.[1]

Lemma 2. *Let $\{y_{(\nu)}\}$ be generated by iterating (6) from any $y_{(0)}\in Y$. For any $y\in Y$ and $\nu\in\mathbb{N}$*

$$\begin{aligned}&\ell_{\delta,L}(y_{(\nu+1)};y_{(\nu)}) + \tfrac{L}{2}\|y_{(\nu+1)}-y_{(\nu)}\|^2\\ &\quad\le \ell_{\delta,L}(y;y_{(\nu)}) + \tfrac{L}{2}\|y_{(\nu)}-y\|^2 - \tfrac{L}{2}\|y_{(\nu+1)}-y\|^2.\end{aligned} \tag{7}$$

Lemma 3. *Let $\{y_{(\nu)}\}$ be generated by iterating (6) from any $y_{(0)}\in Y$. For any $y\in Y$ and $\nu\in\mathbb{N}$*

$$\begin{aligned}&\sum_{i=0}^{\nu}(\Phi(y_{(i+1)})-\Phi(y)) + \sum_{i=0}^{\nu}\Delta_{\delta,L}(y;y_{(i)})\\ &\quad+ \tfrac{L}{2}\|y-y_{(\nu+1)}\|^2 \le \tfrac{L}{2}\|y-y_{(0)}\|^2 + (\nu+1)\delta.\end{aligned} \tag{8}$$

Proof. By the second part of (5) and Lemma 2

$$\begin{aligned}\Phi(y_{(\nu+1)}) &\le \ell_{\delta,L}(y_{(\nu+1)};y_{(\nu)}) + \tfrac{L}{2}\|y_{(\nu+1)}-y_{(\nu)}\|^2 + \delta\\ &\le \ell_{\delta,L}(y;y_{(\nu)}) + \tfrac{L}{2}\|y-y_{(\nu)}\|^2\\ &\quad - \tfrac{L}{2}\|y-y_{(\nu+1)}\|^2 + \delta,\end{aligned} \tag{9}$$

or

$$\begin{aligned}&\Phi(y_{(\nu+1)}) - \Phi(y) + \Delta_{\delta,L}(y;y_{(\nu)}) + \tfrac{L}{2}\|y-y_{(\nu+1)}\|^2\\ &\quad\le \tfrac{L}{2}\|y-y_{(\nu)}\|^2 + \delta.\end{aligned} \tag{10}$$

Summing over $0,\ldots,\nu$ we arrive at (8). □

[1] In Chen and Teboulle (1993, Lemma 3.2) the property is proved for general Bregman distances, see also Bertsekas (2009) and Tseng (2008).

Remark 4. Lemma 3 is the main difference of our analysis compared to that of Devolder et al. (2013). It provides key inequality (8) that will allow us to derive convergence rate estimates not only for the primal version of inexact GP (as it is already done in Devolder et al., 2013) but also for its dual counterpart. This way we will be able to deduce convergence rate estimates for primal feasibility and optimality in fixed-point implementations of DGP for MPC problems.

The next theorem provides convergence rate estimates for the inexact primal GP scheme (6). The theorem has already appeared in Devolder et al. (2013, Theorem 4). However, since our proof can be easily inferred by Lemma 3, we include it for completeness.

Theorem 5. *Let $\{y_{(\nu)}\}_{\nu\in\mathbb{N}}$ be generated by iterating (6) from any $y_{(0)}\in Y$ and let $\bar{y}_{(\nu+1)} \triangleq \frac{1}{\nu+1}\sum_{i=0}^{\nu}y_{(i+1)}$. Then*

$$\Phi(\bar{y}_{(\nu+1)}) - \Phi^\star \le \tfrac{L}{2(\nu+1)}\|y^\star - y_{(\nu)}\|^2 + \delta. \tag{11}$$

Proof. Putting $y=y^\star$ in (8), dropping the terms $\sum_{i=0}^{\nu}\Delta_{\delta,L}(y^\star;y_{(i)})$ and $\frac{L}{2}\|y^\star - y_{(\nu+1)}\|^2$ since they are nonnegative, and dividing by $(\nu+1)$, we arrive at

$$\tfrac{1}{(\nu+1)}\sum_{i=0}^{\nu}(\Phi(y_{(i+1)})-\Phi^\star) \le \tfrac{L}{2(\nu+1)}\|y^\star-y_{(0)}\|^2 + \delta. \tag{12}$$

Since Φ is convex one has $\Phi(\bar{y}_{(\nu+1)}) \le \frac{1}{(\nu+1)}\sum_{i=0}^{\nu}\Phi(y_{(i+1)})$, proving (11). □

4. Inexact dual gradient projection

Consider the problem

$$\begin{aligned}&\text{minimize} \quad V(z)\\ &\text{subject to } g(z)\le 0.\end{aligned} \tag{13}$$

We call (13) the *primal problem* and we assume it to be feasible. In (13), $V:\mathbb{R}^n\to\mathbb{R}$ is a differentiable, strongly convex function with convexity parameter κ_V, i.e.,

$$(\nabla V(z_1) - \nabla V(z_2))'(z_1-z_2) \ge \kappa_V\|z_1-z_2\|^2$$

for all $z_1, z_2\in\mathbb{R}^n$, and $g(z) = Az-b$, $b\in\mathbb{R}^m$. The unique solution of (13) is denoted by $z^\star$. Our ultimate goal is to compute an $(\varepsilon_V,\varepsilon_g)$-optimal solution for (13), defined as follows.

Definition 6. Consider two nonnegative constants $\varepsilon_V,\varepsilon_g$. Vector z is an $(\varepsilon_V,\varepsilon_g)$-optimal solution for (13) if

$$V(z) - V^\star \le \varepsilon_V \tag{14a}$$

$$\|[g(z)]_+\|_\infty \le \varepsilon_g, \tag{14b}$$

where (14a) is a bound on the solution suboptimality, i.e. the discrepancy between the optimal and the achieved cost function values, and (14b) is a bound on the solution infeasibility, i.e. the maximal constraint violation.

Next, we consider the *Lagrangian function* of problem (13)

$$\mathcal{L}(z,y) = V(z) + y'g(z).$$

The (negative of the) *dual problem* of (13) can be expressed as (3), with the convex function $\Phi:\mathbb{R}^m\to\mathbb{R}$ given by

$$\Phi(y) = -\inf_{z\in\mathbb{R}^n}\mathcal{L}(z,y) \tag{15}$$

and $Y=\mathbb{R}^m_+$, i.e., the nonnegative orthant (we refer the reader to Bertsekas (1999, 2009) and Bertsekas, Nedic, and Ozdaglar (2003) for standard results on Lagrangian duality).

For the rest of the paper we assume that there is no duality gap, i.e., $V^\star = -\Phi^\star$. This assumption is fulfilled if, for example, Problem (13) is a convex quadratic program that is feasible, or the Slater condition holds (Bertsekas, 1999, 2009; Bertsekas et al., 2003). Since V is strongly convex, $z_y^\star = \operatorname{argmin}_{z\in\mathbb{R}^n}\mathcal{L}(z,y)$ is unique for any $y \geq 0$, and Φ is L_Φ-smooth with $L_\Phi = \|A\|^2/\kappa_V$ (Nesterov, 2005). The gradient of Φ is given by

$$\nabla\Phi(y) = -g(z_y^\star). \tag{16}$$

Furthermore, we can obtain the unique optimal solution of (13) from any dual optimal solution $y^\star \in Y^\star$ via

$$z^\star = \operatorname*{argmin}_{z\in\mathbb{R}^n} \mathcal{L}(z, y^\star). \tag{17}$$

The gradient projection algorithm applied to the dual problem (15) becomes (for a given $y_{(0)}$)

$$z_{(\nu)} = \operatorname*{argmin}_{z\in\mathbb{R}^n} \mathcal{L}(z, y_{(\nu)}) \tag{18a}$$

$$y_{(\nu+1)} = \left[y_\nu + \tfrac{1}{L_\Phi} g(z_{(\nu)})\right]_+. \tag{18b}$$

Next, assume that for every $y \in Y$, instead of $\nabla\Phi(y) = -g(z_y^\star)$, one can only calculate an approximate gradient

$$\tilde{\nabla}\Phi(y) = -g(z_y) + \xi, \tag{19}$$

where

$$\|z_y - z_y^\star\| \leq \epsilon_z, \qquad \|\xi\| \leq \epsilon_\xi, \tag{20}$$

for given positive constants ϵ_z, ϵ_ξ.

4.1. Modified primal–dual pair

The goal is to construct a first-order inexact oracle for Φ (cf. (15)) with $s_{\delta,L}(y) = \tilde{\nabla}\Phi(y)$. Convergence-rate results for GP methods in the presence of an additive disturbance ξ require the constraint set Y of (3) to be bounded (d'Aspremont, 2008; Devolder, 2012). For this reason, the dual problem (3) will be modified in order to have a bounded constraint set. Let $d \in \mathbb{R}^m$ be such that its ith element satisfies

$$d_i \geq \max\{y_i^\star, 1\} \tag{21}$$

for some $y^\star \in Y^\star$, and

$$Y_\alpha \triangleq \{y \in \mathbb{R}^m \mid 0 \leq y \leq \alpha d\}, \quad \alpha \geq 1. \tag{22}$$

Furthermore, let $D \triangleq \|d\|$ and $D_\alpha \triangleq \max_{y_1,y_2\in Y_\alpha}\|y_1 - y_2\| = \alpha D$, the diameter of Y_α. Next, we consider the following modified dual problem

$$\begin{aligned}&\text{minimize} \quad \Phi(y)\\ &\text{subject to } y \in Y_\alpha.\end{aligned} \tag{23}$$

Obviously we have $Y_\alpha^\star \triangleq \operatorname{argmin}_{y\in Y_\alpha}\Phi(y) \subseteq Y^\star$, therefore any optimal solution of the modified dual problem (23) is also a solution of the original dual problem. Hence, one can compute an optimal solution for (23) and recover the optimal solution for (13) via (17).

Remark 7. In principle, determining a vector d such that (21) holds requires one to know bounds on the elements of a dual optimal solution. If (13) is a parametric QP, as in embedded linear MPC, then tight uniform bounds (valid for every admissible parameter vector) can be computed using techniques described in Patrinos and Bemporad (2014). In fact, one has to compute such bounds anyway, since they are imperative for determining the worst-case number of iterations, and consequently the worst-case running time of the algorithm, a central concern in embedded optimization applications (see, e.g., Bemporad & Patrinos, 2012, Patrinos & Bemporad, 2012 and Richter et al., 2011).

4.2. Inexact oracle

We are now ready to derive an inexact oracle for Φ on Y_α under assumptions (19), (20).

Proposition 8. *Consider Φ given by (15). The pair*

$$\Phi_{\delta,L}(y) = -\mathcal{L}(z_y, y) - \alpha D\epsilon_\xi, \tag{24a}$$

$$s_{\delta,L}(y) = \tilde{\nabla}\Phi(y) = -g(z_y) + \xi \tag{24b}$$

furnishes a (δ_α, L)-oracle for Φ on Y_α, where $\delta_\alpha \triangleq L_V\epsilon_z^2 + 2\alpha D\epsilon_\xi$, $L \triangleq \frac{2}{\kappa_V}\|A\|^2$.

Proof. Since $\mathcal{L}(\cdot, y)$ is L_V-smooth and $z_y^\star$ is its unconstrained minimum, we have that

$$\mathcal{L}(z_y, y) - \mathcal{L}(z_y^\star, y) \leq \tfrac{L_V}{2}\|z_y - z^\star\|^2.$$

Therefore, $\|z_y - z^\star\| \leq \epsilon_z$ implies

$$\mathcal{L}(z_y, y) - \mathcal{L}(z_y^\star, y) \leq \tfrac{L_V}{2}\epsilon_z^2. \tag{25}$$

In Devolder et al. (2013, Section 3.2) it is shown that if for every $y \in Y_\alpha$ one is able to compute a z_y such that (25) is satisfied, then $(-\mathcal{L}(z_y, y), -g(z_y))$ is a $(L_V\epsilon_z^2, L)$-oracle for Φ. Next consider any $w, y \in Y_\alpha$ and ξ such that $\|\xi\| \leq \epsilon_\xi$. We have

$$\begin{aligned}\Phi(w) &= -\mathcal{L}(z_w^\star, w) \geq -\mathcal{L}(z_y, w)\\ &\geq -\mathcal{L}(z_y, y) - g(z_y)'(w - y)\\ &= -\mathcal{L}(z_y, y) + (-g(z_y) + \xi)'(w - y) - \xi'(w - y)\\ &\geq \Phi_{\delta,L}(y) + s_{\delta,L}(y)'(w - y),\end{aligned} \tag{26}$$

where the first inequality follows from (15) and $z_w^\star = \operatorname{argmin}_{z\in\mathbb{R}^n}\mathcal{L}(z, w)$, the second inequality by the fact that $(-\mathcal{L}(z_y, y), -g(z_y))$ is a $(L_V\epsilon_z^2, L)$-oracle for Φ and the left part of (5), and the last inequality by Cauchy–Schwarz and (24). On the other hand

$$\begin{aligned}\Phi(w) &\leq -\mathcal{L}(z_y, y) - g(z_y)'(w - y) + \tfrac{L}{2}\|w - y\|^2 + L_V\epsilon_z^2\\ &\leq -\mathcal{L}(z_y, y) + (-g(z_y) + \xi)'(w - y)\\ &\quad - \xi'(w - y) + \tfrac{L}{2}\|w - y\|^2 + L_V\epsilon_z^2\\ &\leq -\mathcal{L}(z_y, y) + (-g(z_y) + \xi)'(w - y)\\ &\quad + \tfrac{L}{2}\|w - y\|^2 + L_V\epsilon_z^2 + \alpha D\epsilon_\xi\\ &= \Phi_{\delta,L}(y) + s_{\delta,L}(y)'(w - y) + \tfrac{L}{2}\|w - y\|^2\\ &\quad + L_V\epsilon_z^2 + 2\alpha D\epsilon_\xi.\end{aligned} \tag{27}$$

where the first inequality follows from the fact that $(-\mathcal{L}(z_y, y), -g(z_y))$ is a $(L_V\epsilon_z^2, L)$-oracle for Φ and the right part of (5), the third inequality by Cauchy–Schwarz, and the equality by (24). Therefore, $(\Phi_{\delta,L}(y), s_{\delta,L}(y))$ given by (24) is a (δ_α, L)-oracle for Φ on Y_α. □

Notice that the oracle error δ_α decreases with α, achieving its minimum value for $\alpha = 1$. Furthermore, the bounding of the dual feasible set is essential (cf. (22)), otherwise it would not be possible to bound quantities such as $\|w - y\|$, for any $w, y \geq 0$.

4.3. Primal convergence rates

Under the assumptions imposed by (19), (20), the ν-th iteration of the inexact DGP scheme applied to Problem (23) with the first-order oracle given by Proposition 8 is

$$\begin{aligned}&y_{(\nu+1)} = \Pi_{Y_\alpha}(y_{(\nu)} + \tfrac{1}{L}(g(z_{(\nu)}) + \xi_{(\nu)})),\\ &\quad\text{with } z_{(\nu)}, \xi_{(\nu)} \text{ s.t. } \|z_{(\nu)} - z_{y_{(\nu)}}^\star\| \leq \epsilon_z,\ \|\xi_{(\nu)}\| \leq \epsilon_\xi.\end{aligned} \tag{28}$$

The Euclidean projection onto Y_α is very easy to compute, since for $w \in \mathbb{R}^m$ we have $\Pi_{Y_\alpha}(w) = \max\{\min\{w, \alpha d\}, 0\}$.

We will next derive global convergence rates to primal optimality and primal feasibility for the ergodic primal iterates

$$\bar{z}_{(\nu)} \triangleq \tfrac{1}{\nu+1}\sum_{i=0}^{\nu} z_{(i)}. \tag{29}$$

First, the following lemma is needed.

Lemma 9. *Let $\{y_{(\nu)}, z_{(\nu)}\}$ be generated by iterating (28) from any $y_{(0)} \in Y_\alpha$. For any $y \in Y_\alpha$ and $\nu \in \mathbb{N}$*

$$\mathcal{L}(\bar{z}_{(\nu)}, y) - V^\star \le \tfrac{L}{2(\nu+1)}\|y - y_{(0)}\|^2 + \delta_\alpha. \tag{30}$$

Proof. For any $y \in Y_\alpha$, one has

$$\begin{aligned}
\Delta_{\delta,L}(y; y_{(\nu)}) &= \Phi(y) + \mathcal{L}(z_{(\nu)}, y_{(\nu)}) \\
&\quad + \alpha D\epsilon_\xi + (g(z_{(\nu)}) - \xi_{(\nu)})'(y - y_{(\nu)}) \\
&\ge \Phi(y) + V(z_{(\nu)}) + \alpha D\epsilon_\xi + g(z_{(\nu)})'y \\
&\quad - \|\xi_{(\nu)}\|\,\|y - y_{(\nu)}\| \\
&\ge \Phi(y) + \mathcal{L}(z_{(\nu)}, y),
\end{aligned} \tag{31}$$

where the equality follows from (24), the first inequality by Cauchy–Schwarz and the second one by (20) and the fact that $y_{(\nu)}$ belongs to Y_α. Summing over $0, \dots, \nu$

$$\begin{aligned}
\sum_{i=0}^{\nu}\Delta_{\delta,L}(y; y_{(i)}) &= (\nu+1)\Phi(y) + \sum_{i=0}^{\nu}\mathcal{L}(z_{(\nu)}, y) \\
&\ge (\nu+1)(\Phi(y) + \mathcal{L}(\bar{z}_{(\nu)}, y)),
\end{aligned} \tag{32}$$

where the inequality follows by convexity of $\mathcal{L}(\cdot, y)$ for any fixed nonnegative $y \in Y_\alpha$. Dropping $\frac{L}{2}\|y - y_{(\nu+1)}\|^2$ from (8), using (32) and the convexity of Φ, we obtain

$$\Phi(\bar{y}_{(\nu+1)}) + \mathcal{L}(\bar{z}_{(\nu)}, y) \le \tfrac{L}{2(\nu+1)}\|y - y_{(0)}\|^2 + \delta_\alpha. \tag{33}$$

Finally, using $\Phi(\bar{y}_{(\nu+1)}) \ge -V^\star$ we arrive at (30). □

The next theorem gives the convergence rate towards primal feasibility for the ergodic primal iterates generated by the inexact dual GP (28).

Theorem 10 (*Bound on Primal Infeasibility*). *Let $\{y_{(\nu)}, z_{(\nu)}\}$ be generated by iterating (28) from any $y_{(0)} \in Y_\alpha$. If $\alpha > 1$, then for any $\nu \in \mathbb{N}$*

$$\|[g_i(\bar{z}_{(\nu)})]_+\|_\infty \le \tfrac{\alpha^2}{\alpha-1}\tfrac{LD^2}{2(\nu+1)} + \delta_\alpha^g, \tag{34}$$

where $\delta_\alpha^g \triangleq \frac{1}{\alpha-1}L_V\epsilon_z^2 + \frac{\alpha}{\alpha-1}2D\epsilon_\xi$.

Proof. Maximizing both sides of (30) with respect to $y \in Y_\alpha$ and using

$$\max_{y\in Y_\alpha}\mathcal{L}(\bar{z}_{(\nu)}, y) = V(\bar{z}_{(\nu)}) + \alpha\sum_{i=1}^{m} d_i[g_i(\bar{z}_{(\nu)})]_+ \tag{35}$$

we obtain

$$V(\bar{z}_{(\nu)}) - V^\star + \alpha\sum_{i=1}^{m} d_i[g_i(\bar{z}_{(\nu)})]_+ \le \tfrac{LD^2}{2(\nu+1)}\alpha^2 + \delta_\alpha. \tag{36}$$

Choose a $y^\star \in Y_\alpha^\star$ with $y^\star \le d$ (it exists by definition of d). By the saddle-point inequality, we have that $V^\star = \mathcal{L}(z^\star, y^\star) \le \mathcal{L}(\bar{z}_{(\nu)}, y^\star)$, or

$$V(\bar{z}_{(\nu)}) - V^\star \ge -g(\bar{z}_{(\nu)})'y^\star \ge -[g(\bar{z}_{(\nu)})]_+'y^\star. \tag{37}$$

Using this in (36), we arrive at

$$\sum_{i=1}^{m}(\alpha d_i - y_i^\star)[g_i(\bar{z}_{(\nu)})]_+ \le \tfrac{LD^2}{2(\nu+1)}\alpha^2 + \delta_\alpha. \tag{38}$$

Since $\alpha > 1$ and $y^\star \le d$,

$$\begin{aligned}
\sum_{i=0}^{m}(\alpha d_i - y_i^\star)[g_i(\bar{z}_{(\nu)})]_+ &\ge (\alpha-1)\min_{i\in\mathbb{N}_{[1,m]}}\{d_i\}\cdot\sum_{i=1}^{m}[g_i(\bar{z}_{(\nu)})]_+ \\
&\ge (\alpha-1)\|[g_i(\bar{z}_{(\nu)})]_+\|_\infty,
\end{aligned} \tag{39}$$

where the last inequality follows from (21). Therefore

$$\|[g(\bar{z}_{(\nu)})]_+\|_\infty \le \tfrac{\alpha^2}{\alpha-1}\tfrac{LD^2}{2(\nu+1)} + \tfrac{\delta_\alpha}{(\alpha-1)}. \quad \square \tag{40}$$

Notice that there is a trade-off in (34) between the constant of the $O(1/\nu)$ term determining the convergence rate to feasibility, and the maximum level of infeasibility that one is able to tolerate, asymptotically. As $\alpha \to \infty$, δ_α^g approaches its infimum, $2D\epsilon_\xi$, while $\frac{\alpha^2}{\alpha-1} \to \infty$. By choosing $\alpha = 2$ (the one that minimizes $\frac{\alpha^2}{\alpha-1}$) we arrive at

$$\|[g(\bar{z}_{(\nu)})]_+\|_\infty \le \tfrac{2LD^2}{\nu+1} + L_V\epsilon_z^2 + 4D\epsilon_\xi. \tag{41}$$

Theorem 11 (*Bound on Primal Suboptimality*). *Let $\{y_{(\nu)}, z_{(\nu)}\}$ be generated by iterating (28) from any $y_{(0)} \in Y_\alpha$. Then*

$$V(\bar{z}_{(\nu)}) - V^\star \le \tfrac{L}{2(\nu+1)}(\|y^\star\|^2 + \|y_{(0)}\|^2) + \delta_\alpha, \tag{42a}$$

$$V(\bar{z}_{(\nu)}) - V^\star \ge -\left(\tfrac{\alpha^2}{\alpha-1}\tfrac{LD^2}{2(\nu+1)} + \delta_\alpha^g\right)D. \tag{42b}$$

Proof. Choose $y^\star \in Y_\alpha$ with $y^\star \le d$. By substituting $y = \bar{y}^\star \ge 0$ in (30), where

$$\bar{y}_i^\star = \begin{cases} y_i^\star, & \text{if } g_i(\bar{z}_{(\nu)}) \ge 0, \\ 0, & \text{if } g_i(\bar{z}_{(\nu)}) < 0, \end{cases}$$

and dropping the term $g(\bar{z}_{(\nu)})'\bar{y}^\star$ since it is nonnegative, we obtain

$$V(\bar{z}_{(\nu)}) - V^\star \le \tfrac{L}{2(\nu+1)}\|\bar{y}^\star - y_{(0)}\|^2 + \delta_\alpha. \tag{43}$$

Now

$$\begin{aligned}
\|\bar{y}^\star - y_{(0)}\|^2 &= \|\bar{y}^\star\|^2 - 2y_{(0)}'\bar{y}^\star + \|y_{(0)}\|^2 \\
&\le \|y^\star\|^2 + \|y_{(0)}\|^2,
\end{aligned} \tag{44}$$

since $\|\bar{y}^\star\| \le \|y^\star\|$ and $2y_{(0)}'\bar{y}^\star \ge 0$. Therefore using the last inequality in (43), we arrive at (42a). To prove (42b), using (37) and Cauchy–Schwarz we obtain

$$V(\bar{z}_{(\nu)}) - V^\star \ge -\left\|[g(\bar{z}_{(\nu)})]_+\right\|\,\|y^\star\|. \tag{45}$$

Using (34), and a $y^\star$ with $\|y^\star\| \le d$ we get (42b). □

Notice that the constant of the $O(1/\nu)$ term in (42a) is independent of α. In fact, if iterations (28) start from $y_{(0)} = 0$, then the cost $V(\bar{z}_{(\nu)})$ is always lower than $V^\star + \delta_\alpha$, the best achievable by the corresponding scheme asymptotically, as it is shown below. In that case, one has to worry only about feasibility.

Corollary 12. *Let $\{y_{(\nu)}, z_{(\nu)}\}$ be generated by iterating (28) starting from $y_{(0)} = 0$. Then*

$$V(\bar{z}_{(\nu)}) - V^\star \le \delta_\alpha, \quad \forall \nu \in \mathbb{N}. \tag{46}$$

Proof. Simply put $y = 0$ in (30). □

4.4. Optimal choice of α for fixed oracle errors ϵ_z, ϵ_ξ

We will next derive the value of the user-defined parameter α that achieves the fastest convergence rate to an $(\varepsilon_V, \varepsilon_g)$-solution, given oracle parameters ϵ_z, ϵ_ξ. For simplicity, we assume that the initial iterate is equal to the zero vector, i.e., $y_{(0)} = 0$. In that case one should only worry about convergence to primal feasibility since, due to Corollary 12, $V(\bar{z}_{(\nu)}) - V^\star \leq \delta_\alpha$, for every $\nu \in \mathbb{N}_+$.

First, one must have $\varepsilon_V \geq \delta_\alpha$, or

$$\alpha \leq \frac{\varepsilon_V - L_V \epsilon_z^2}{2D\epsilon_\xi}. \tag{47}$$

Regarding ε_g, for sure it must be larger than $2D\epsilon_\xi$, the infimum of δ_α^g. Furthermore, by (34) it must satisfy $\varepsilon_g \geq \delta_\alpha^g$, implying that α must satisfy

$$\alpha > \frac{\varepsilon_g + L_V \epsilon_z^2}{\varepsilon_g - 2D\epsilon_\xi}. \tag{48}$$

Notice that the right hand-side of (48) is greater than one, since $\varepsilon_g > 2D\epsilon_\xi$. Eqs. (47), (48), pose the following restriction

$$\varepsilon_V > \frac{\varepsilon_g (L_V \epsilon_z^2 + 2D\epsilon_\xi)}{\varepsilon_g - 2D\epsilon_\xi}. \tag{49}$$

4.5. Bound of the number of iterations

In order to achieve $\|[g_i(\bar{z}_{(\nu)})]_+\|_\infty \leq \varepsilon_g$, according to Theorem 10 the algorithm defined by (28) will need no more than $\nu(\alpha)$ iterations, where

$$\nu(\alpha) = \frac{LD^2\alpha^2}{2(\varepsilon_g - 2D\epsilon_\xi)\alpha - 2(\varepsilon_g + L_V\epsilon_z^2)} - 1. \tag{50}$$

The tightest upper-bound on the number of iterations is given by the next theorem.

Theorem 13. *Suppose that $\varepsilon_g > 2D\epsilon_\xi$, and let ε_V satisfy (49). Then an $(\varepsilon_V, \varepsilon_g)$-solution is obtained by iterating (28) from $y_{(0)} = 0$ with $\alpha = \alpha^\star$,*

$$\alpha^\star \triangleq \min\left\{\frac{2(\varepsilon_g + L_V\epsilon_z^2)}{\varepsilon_g - 2D\epsilon_\xi}, \frac{\varepsilon_V - L_V\epsilon_z^2}{2D\epsilon_\xi}\right\}, \tag{51}$$

no more than $\nu(\alpha^\star)$ times, where $\nu(\alpha)$ is given by (50).

Proof. Let $c_1 = LD^2$, $c_2 = 2(\varepsilon_g - 2D\epsilon_\xi)$, $c_3 = 2(\varepsilon_g + L_V\epsilon_z^2)$. Then $\nu(\alpha) = \frac{c_1\alpha^2}{c_2\alpha - c_3}$. The fastest convergence rate is achieved when $\alpha = \alpha^\star \triangleq \operatorname{argmin}\{\nu(\alpha) | \alpha \in [\underline{\alpha}, \overline{\alpha}]\}$, where $\underline{\alpha} = \frac{c_3}{c_2}$, $\overline{\alpha} = \frac{\varepsilon_V - L_V\epsilon_z^2}{2D\epsilon_\xi}$. Function $\nu(\alpha)$ is convex on $[\underline{\alpha}, \overline{\alpha}]$ since $\nu''(\alpha) = \frac{2c_1c_3^2}{(c_2\alpha - c_3)^3} \geq 0$, for $\alpha \geq \underline{\alpha}$. Setting its derivative equal to zero, we find $\alpha = \frac{2c_3}{c_2}$, which is the first term in the min operator in (51). □

For the nominal case ($\epsilon_z = \epsilon_\xi = 0$), from (51) we obtain $\alpha^\star = 2$.

4.6. Maximum admissible oracle errors ϵ_z, ϵ_ξ

Based on the results of Section 4.3, we will give explicit formulae of the maximum admissible oracle errors ϵ_z, ϵ_ξ as a function of solution accuracy ε_V, ε_g, and consequently of the number of iterations that are executed.

The question just posed is of significant importance in embedded optimization-based control, like MPC, for the following reason. Given hard real-time constraints dictated by hardware specifications and sampling time, as well as sufficiently small values for ε_V, ε_g, that guarantee closed-loop stability (see Rubagotti et al., 2014), one wants to determine the smallest allowable oracle precision (maximum allowable values for ϵ_z, ϵ_ξ) that achieves the aforementioned requirements. As it will become clear in Section 5, ϵ_z, ϵ_ξ correspond to round-off errors due to fixed-point arithmetic. The smaller the number of fractional bits is (i.e., the larger the maximum allowable oracle errors), the smaller the execution time and the power consumption of the hardware device will be.

Again, for simplicity we assume that $y_{(0)} = 0$. Moreover, we suppose that $\epsilon_\xi = \beta\epsilon_z$, for some $\beta > 0$. This last assumption is justified in Section 5. Finally, it is assumed that $\alpha = 2$, since for small ϵ_z, ϵ_ξ this is usually the best choice. First, from (41), we have

$$\tfrac{2LD^2}{\nu+1} + L_V\epsilon_z^2 + 4D\beta\epsilon_z \leq \varepsilon_g. \tag{52}$$

By solving with respect to ϵ_z, we arrive at

$$\epsilon_z \leq \sqrt{\tfrac{\varepsilon_g}{L_V} + \left(\tfrac{2D\beta}{L_V}\right)^2 - \tfrac{2LD^2}{L_V(\nu+1)}} - \tfrac{2D\beta}{L_V}. \tag{53}$$

Due to Corollary 12, one must have $\delta_2 \leq \varepsilon_V$, or

$$\epsilon_z \leq \sqrt{\tfrac{\varepsilon_V}{L_V} + \left(\tfrac{2D\beta}{L_V}\right)^2} - \tfrac{2D\beta}{L_V}. \tag{54}$$

Letting $\nu \to \infty$ in (53), and taking into account that $\sqrt{\varepsilon/L_V + (2D\beta/L_V)^2}$ is increasing as a function of ε, we conclude that, in order to be able to converge asymptotically to an $(\varepsilon_V, \varepsilon_g)$-solution, ϵ_z must satisfy

$$\epsilon_z < \sqrt{\tfrac{\varepsilon}{L_V} + \left(\tfrac{2D\beta}{L_V}\right)^2} - \tfrac{2D\beta}{L_V}, \tag{55}$$

where $\varepsilon = \min\{\varepsilon_g, \varepsilon_V\}$. It is worth mentioning that the maximum oracle error ϵ_z, that allows one to reach accuracy ε decreases as $O(\sqrt{\varepsilon})$, slower than $O(\varepsilon)$ for $\varepsilon < 1$ (which is usually the case of interest).

5. Fixed-point DGP for QPs

The theory presented in Section 4 allows us to analyze the fixed-point implementation of the DGP algorithm defined by (18) for strictly convex quadratic programs. Consider problem (13) with $V(z) = \frac{1}{2}z'Qz + q'z$ and let $\kappa_V = \lambda_{\min}(Q) > 0$. The dual problem (modulo a sign change) is (3) with $\Phi(y) = \frac{1}{2}y'Hy + h'y$, and $Y = \mathbb{R}_+^m$, where $H = AQ^{-1}A'$, $h = AQ^{-1}q + b$. Furthermore,

$$z_y^\star = Ey + e, \tag{56}$$

where $E = -Q^{-1}A'$, $e = -Q^{-1}q$. Therefore, the DGP iterations (18) lead to the following algorithm ($y_{(0)} = 0$)

$$z_{(\nu)} = Ey_{(\nu)} + e \tag{57a}$$

$$g_{(\nu)} = Az_{(\nu)} - b \tag{57b}$$

$$y_{(\nu+1)} = \left[y_{(\nu)} + \tfrac{1}{L}g_{(\nu)}\right]_+ \tag{57c}$$

with stopping criterion

$$\|[A\bar{z}_{(\nu)} - b]_+\|_\infty \leq \epsilon_g \tag{57d}$$

with $\bar{z}_{(\nu)}$ given by (29) and $y_{(0)} = 0$.

5.1. Fixed-point implementation

Let the algorithm defined by (57) be embedded on a fixed-point hardware with a scaling factor 2^{-p}, where $p \in \mathbb{N}_+$ is the number of fractional bits. We assume that real numbers are represented in fixed-point by rounding to the closest value. Therefore, the resolution (i.e., the smallest representable non-zero magnitude) of a fixed-point number is equal to $2^{-(p+1)}$.

In a fixed-point architecture, for a given $y \in \mathbb{R}^m$, instead of the gradient $\nabla \Phi(y) = -g(z_y^\star)$, where $z_y^\star$ is given by (56), we have access to $\tilde{\nabla}\Phi(y)$ of the form (19), with $z_y = \mathtt{fi}(Ey + e)$, and $\xi = g(z_y) - \mathtt{fi}(Az_y - b)$. Due to (2), the vectors ξ, z_y satisfy (20) with

$$\epsilon_z = 2^{-(p+1)} m\sqrt{n}, \tag{58a}$$

$$\epsilon_\xi = 2^{-(p+1)} n\sqrt{m}. \tag{58b}$$

Since $L = \frac{2}{\kappa_V}\|A\|^2$, by properly scaling the problem matrices we can assume that $L = 1$, therefore there is no round-off error in computing the product $\frac{1}{L} g_{(\nu)}$.

According to Proposition 8, the pair $(\Phi_{\delta_\alpha, L}(y), \tilde{\nabla}\Phi(y))$ given by (24) is a (δ_α, L)-oracle for Φ on Y_α. The ν-th iteration of the inexact DGP scheme (28) implemented on the fixed-point hardware platform is

$$z_{(\nu)} = \mathtt{fi}(Ey_{(\nu)} + e), \tag{59a}$$

$$g_{(\nu)} = \mathtt{fi}(Az_{(\nu)} - b), \tag{59b}$$

$$y_{(\nu+1)} = \max\left\{\min\left\{y_{(\nu)} + \tfrac{1}{L} g_{(\nu)}, \alpha d\right\}, 0\right\}. \tag{59c}$$

5.2. Number of fractional bits

We now provide explicit bounds on the number of fractional bits required to grant convergence of (59) to a target primal suboptimal solution satisfying (57d).

Corollary 14. *Let $\{z_{(\nu)}\}$ be generated by iterating (59), with $y_{(0)} = 0$. Assume that real numbers are rounded to the closest fixed-point value. Then, the algorithm converges asymptotically to an $(\varepsilon_g, \varepsilon_V)$-solution, with $\varepsilon_g > 2D\epsilon_\xi$ and ε_V satisfying (49), if the number of fractional bits p is such that*

$$p \geq \log_2 \frac{m\sqrt{n}}{\sqrt{\frac{\varepsilon}{L_V} + \frac{n}{m}\left(\frac{2D}{L_V}\right)^2} - \sqrt{\frac{n}{m}}\frac{2D}{L_V}} - 1, \tag{60}$$

where $\varepsilon = \min\{\varepsilon_g, \varepsilon_V\}$.

Proof. Combine (55) with (58a), (58b). □

5.3. Number of integer bits

Together with round-off errors, another key issue that arises while embedding computations on fixed-point architectures is the occurrence of overflow errors, given by the limited range for number representation. In particular, if the number of bits for the integer part equals r, the computed variables can only assume values in $\left[-2^{r-1}, 2^{r-1} - 1\right]$, where the asymmetry is given by the presence of the zero element. The following corollary will set precise guidelines for choosing a number of integer bits that is sufficiently large to avoid overflows.

Corollary 15. *Let the iterations in (59) be run on a fixed-point architecture with r bits for the integer part and $y_{(0)} = 0$. Then, occurrence of overflow errors is avoided if r is chosen such that*

$$r \geq \log_2\left(\max\left\{\hat{y}, \hat{z}, \hat{g}\right\} + 1\right) + 1, \tag{61}$$

where $\hat{y} = \alpha\|d\|_\infty$, $\hat{z} = \|E\|_\infty \hat{y} + \|e\|_\infty$, $\hat{g} = \|A\|_\infty \hat{z} + \|b\|_\infty$.

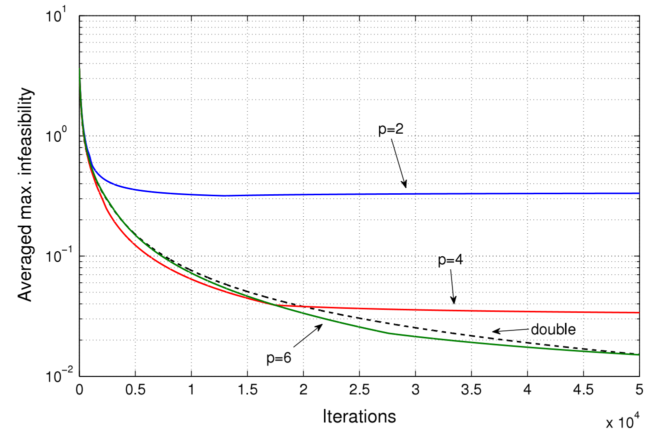


Fig. 1. Primal infeasibility for different precisions p.

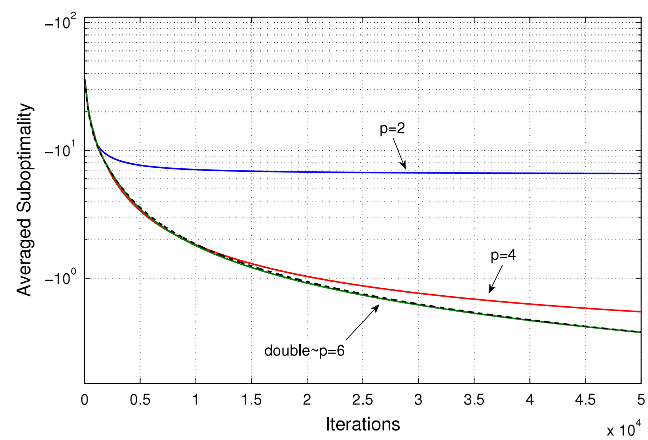


Fig. 2. Primal suboptimality for different precisions p.

Proof. A real number γ lies within the admissible range representable in fixed-point with r integer bits if $r \geq \log_2(\gamma + 1) + 1$. Since $\log_2(\cdot)$ is strictly increasing, r must be such that

$$r \geq \log_2(\max\{\|y_{(\nu)}\|_\infty, \|z_{(\nu)}\|_\infty, \|g_{(\nu)}\|_\infty\} + 1) + 1,$$

for all $\nu \in \mathbb{N}$. Using (59), one obtains $\|y_{(\nu)}\|_\infty \leq \hat{y}$, $\|z_{(\nu)}\|_\infty \leq \hat{z}$, $\|g_{(\nu)}\|_\infty \leq \hat{g}$, for all ν. Again, since $\log_2(\cdot)$ is strictly increasing we arrive at (61). □

6. Simulations

6.1. Sample evolutions

The aim of this section is to show sample infeasibility and suboptimality evolutions generated by (59) for multiple fixed-point precisions p, and compare them with the double-precision case (64 bit floating-point format defined in *IEEE 754* standard IEEE, 2008). Simulations were performed in Matlab R2012b equipped with the Fixed-Point Toolbox v.3.6 on a Mid-2012 Macbook Pro Retina running OSX 10.8.2.

We iterate (59) for a fixed number of steps ν on the dual of randomly generated QP problems, with 10 optimization variables and 20 constraints. Fig. 1 shows the convergence for primal infeasibility $\|[g(\bar{z}_{(\nu)})]_+\|_\infty$, while Fig. 2 for primal suboptimality $|V(\bar{z}_{(\nu)}) - V^\star|$ of the averaged iterates. In both figures, computation in double precision is compared with fixed-point precision for $p = \{2, 4, 6\}$. Simulation for the double-precision evolution was performed according to the standard DGP algorithm shown in (4), without the upper bound given by αd.

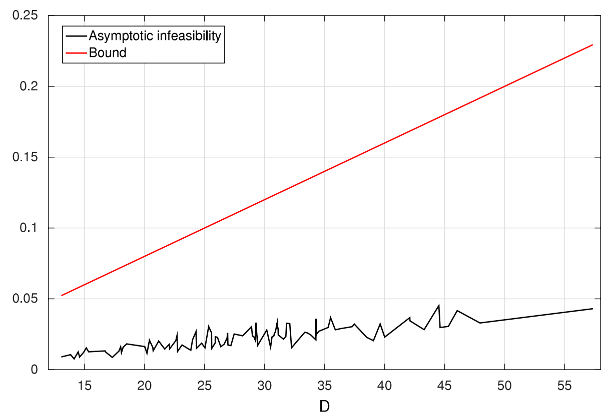


Fig. 3. Asymptotic infeasibility values compared to theoretical bound (34).

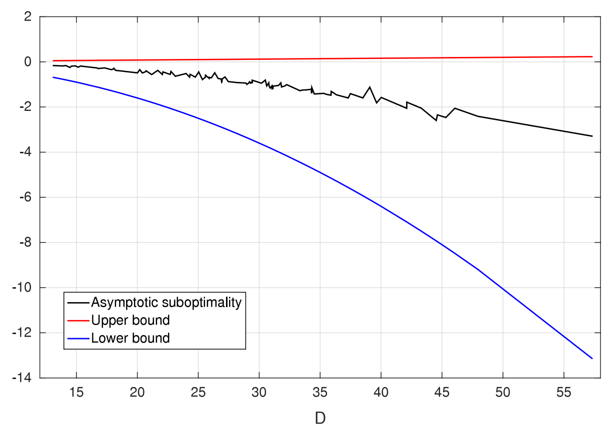


Fig. 4. Asymptotic suboptimality values compared to theoretical bounds (42a) and (42b).

Curve shapes are consistent with the theoretical bounds given by Theorems 10 and 11. In addition, the algorithm presents a remarkable robustness to round-off errors; 4-bits and 6-bits curves already show a convergence comparable with the double-precision case. This fact is of particular interest for embedded implementations, since power consumption is heavily dependent on the number of bits used to represent numbers (Kerrigan et al., 2012).

6.2. Bounds on primal infeasibility and primal suboptimality

The purpose of the second simulation is to test the tightness for the primal infeasibility and suboptimality bounds given by Theorems 10 and 11, respectively.

The analysis was performed on a worst-case scenario, running iterations (28) with $\|\xi_{(\nu)}\| = \epsilon_\xi$ to solve 100 randomly generated QP problems, with 10 optimization variables and 20 constraints. The goal was to compare error bound terms δ_α and δ_α^g with the practical asymptotic values of the primal infeasibility (Fig. 3) and suboptimality (Fig. 4) for $\nu \to \infty$. Different trials are ordered for increasing values of D, term proportional to the constraint set diameter.

Simulation results show an acceptable tightness for bounds, as they exceed the practical values by a factor between 3.33 and 8.32 for infeasibility, and between 3.05 and 5.74 for suboptimality. In addition, the linear (for infeasibility) and quadratic (for suboptimality) theoretical dependencies on D are reflected in the experiment results.

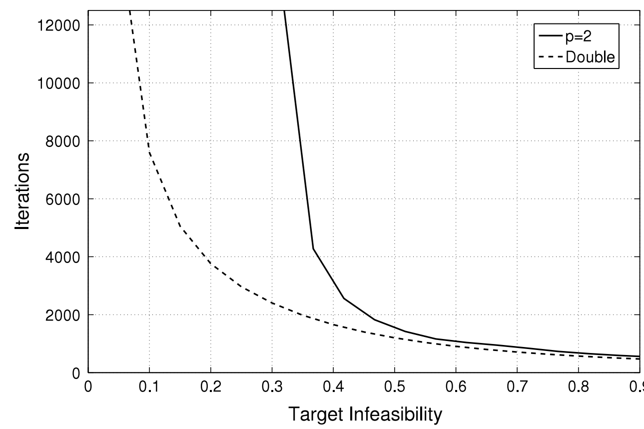


Fig. 5. Iterations for target infeasibility.

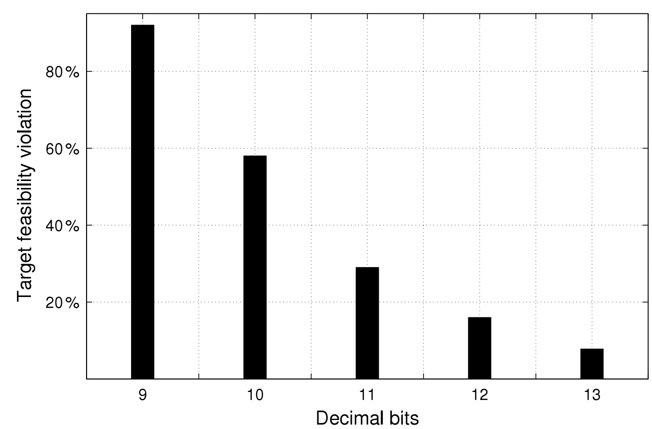


Fig. 6. Fractional bits for target infeasibility.

6.3. Target infeasibility

Fig. 5 shows simulation results on the practical number of iterations needed to reach a target primal infeasibility for a sample, random QP with 5 variables and 10 constraints. A comparison was made between the standard, double-precision gradient projection algorithm (4) and the fixed-point algorithm (59) with 2-bit precision. Results are in accordance with the theoretical results of Theorem 10 since for finite-precision, the number of iterations grows to infinity when target infeasibility reaches a critical value, different from zero.

Fig. 6 shows asymptotic primal infeasibility as a function of the number of fractional bits, based on (60). Note that inequality constraints for the primal QP have been normalized, such that all elements of b are equal to one.

6.4. Bounds on iteration count

The following simulation is performed to test the tightness of the theoretical bound on the number of iterations given by (50).

We let Algorithm (59) run on Matlab R2012b and Fixed-Point Toolbox v.3.6 on a Mid-2012 Macbook Pro Retina running OSX 10.8.2 to solve various random QPs (sizes are equal to 4 and 8 for the primal and the dual, respectively). In Fig. 7, the practical number of iterations needed to reach decreasing target infeasibilities is plotted against the theoretical bound; different colors and markers are chosen for different QP solutions.

Results show that theoretical bounds are about one order of magnitude larger than actual iterations. In addition to this, two interesting properties emerge from the plot: (1) within one single

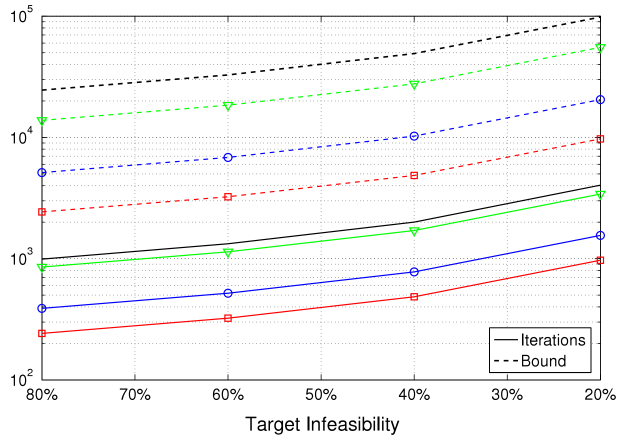


Fig. 7. Comparison between the actual number of iterations and the theoretical bounds predicted by (50) calculated on random QP problems.

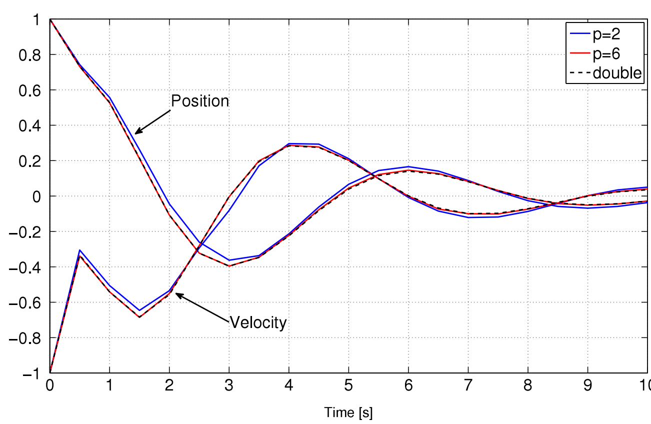


Fig. 8. Mass displacement and velocity evolutions for different precisions.

QP plot, the two curves of practical and theoretical values have similar shapes, and (2) if a first QP needs more iterations than a second QP to be solved, then the first bound is larger; those facts increase the confidence in the formulation of the bound.

6.5. Example: masses serially connected

The purpose of this example is to test the fixed-point implementation (57) of DGP algorithm as a QP solver for an MPC design.

The reference physical system is composed by a series of M elements, each of mass m, connected by springs with constant k and dampeners with constant c. The first and the last element are connected to fixed walls, and actuators are placed between each pair of masses to exert tensions. The state-space model is derived by a set of first-principle ODEs, where the system states are the displacements and velocities of the masses and the inputs are the tensions exerted by the actuators. This is a modification of the example proposed in Wang and Boyd (2010).

Simulations have been performed in Matlab R2012b on a Mid-2012 Macbook Pro Retina running OSX 10.8.2. The QP problem is built forcing the system states to be in $[-4, 4]$ and inputs in $[-1, 1]$, and setting the stage cost equal to $l(x, u) = \frac{1}{2}\left(x'Qx + u'Ru\right)$ with Q and R as identity matrices. The prediction horizon N is equal to 10, and the sampling time 0.5 s.

Fig. 8 shows the evolution of position and velocity for the second mass out of a total of 3 masses. The reference dashed lines (double precision) are obtained closing the loop with an MPC controller supported by IBM ILOG CPLEX v.12.4 as solver of the QP optimization problem. For the remaining plots, the controller is instead supported by fixed-point DGP algorithm implemented with Fixed-Point Toolbox v.3.6; two simulations are performed varying precision to 2 and 6 bits.

Results show a remarkable robustness of the closed-loop evolutions with respect to fixed-point precision. Position and velocity trajectories of the 6-bit simulation are almost undistinguishable with the double precision simulation, while for the 2-bit case a small divergence shows up. This behavior is consistent with what shown in Figs. 1 and 2.

6.6. Hardware implementation

Finally, Algorithm (59) has been implemented on a 32-bit Atmel SAM3X8E ARM Cortex-M3 processing unit; this chipset operates at a maximum speed of 84 MHz and comes with 512 kB of flash memory and 100 kB of RAM.

The microcontroller was assigned to solve random QP problems of increasing size, ranging from 10 to 60 primal variables and 20 to 120 primal constraints. The algorithm was stopped upon reaching a suboptimal solution bounded by 10% primal infeasibility.

Table 1 shows the results when a fixed-point number representation is adopted, with 8 bits for the fractional part and 7 bits for the integer part. For each problem size we report convergence time, average time per iteration (TPI) and size of the binary code; the latter plays an important role in embedded applications, where usually a limited amount of memory is available.

Table 1
Fixed-point hardware implementation.

Size (vars/constr)	Time (ms)	Time/Iter (μs)	Code size (kB)
10/20	22.9	226	15
20/40	52.9	867	17
40/80	544.9	3382	27
60/120	1519.8	7561	43

Table 2
Floating-point hardware implementation.

Size (vars/constr)	Time (ms)	Time/Iter (μs)	Code size (kB)
10/20	88.6	974	16
20/40	220.1	3 608	21
40/80	2240	13 099	40
60/120	5816	30 450	73

In order to evaluate the performance enhancements coming from fixed-point computations, we repeated all the hardware simulations after switching to floating-point number representation. Results are reported in Table 2, which shows how this implementation is about 4 times slower than the fixed-point one, and up to twice as bigger in code size.

Fig. 9 shows the linear relationship between problem size, expressed as variables × constraints, time per iterations, and code size. It is important to notice how the floating-point lines have a steeper slope than the fixed-point counterparts, meaning that the gain in performance increases as the problem becomes larger in size.

This implementation highlights some of the key advantages of the fixed-point format: the computational burden and the memory footprint are lowered, especially on devices lacking hardware support for floating-point operations. However, it has to be noted that the flexibility of the floating-point representation is lost, causing reduced precision and range; the choice of the optimal format is therefore dependent on the specific application and computing capabilities. Especially in the case of chipsets equipped with a floating-point unit (FPU), the benefits from switching to fixed-point arithmetic may be substantially reduced.

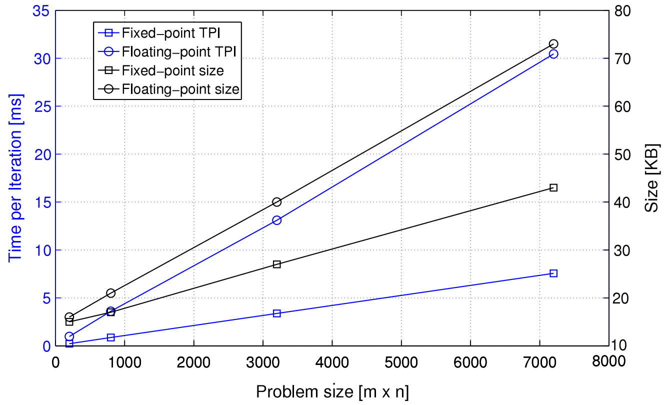


Fig. 9. Comparison between fixed-point and floating-point implementations.

7. Conclusions and future work

This paper has proposed a DGP method for embedding MPC controllers in hardware with fixed-point arithmetic. Concrete and theoretically-proven guidelines for selecting the minimum number of fractional and integer bits that guarantee favorable convergence properties are provided. Future work includes quantifying the effect of fixed-point arithmetic on accelerated versions of the DGP method and perhaps modifications of it to achieve the optimal trade-off between convergence rate and round-off error accumulation, as well as the implementation and testing of the algorithm in a real process control experiment.

References

Bemporad, A. (2006). Model predictive control design: new trends and tools. In *Proc. of 45th conf. on decision and control, San Diego, California* (pp. 6678–6683).

Bemporad, A., & Patrinos, P. (2012). Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. In *Proc. IFAC nonlinear model predictive control conf., Noordwijkerhout, Netherlands* (pp. 14–20).

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific.

Bertsekas, D. P. (2009). *Convex optimization theory*. Athena Scientific.

Bertsekas, D. P., Nedic, A., & Ozdaglar, A. E. (2003). *Convex analysis and optimization*. Athena Scientific.

Chen, G., & Teboulle, M. (1993). Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, *3*(3), 538–543.

d'Aspremont, A. (2008). Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, *19*(3), 1171–1183.

Devolder, O. (2012). Stochastic first order methods in smooth convex optimization. CORE Discussion Papers 2012, 9.

Devolder, O., Glineur, F., & Nesterov, Y. (2013). First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 1–39.

Hartley, E. N., Jerez, J. L., Suardi, A., Maciejowski, J. M., Kerrigan, E. C., & Constantinides, G. A. (2012). Predictive control of a Boeing 747 aircraft using an FPGA. In *Proc. IFAC nonlinear model predictive control conf., Noordwijkerhout, Netherlands* (pp. 80–85).

(2008). IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008, August (pp. 1–70).

Jerez, J. L., Constantinides, G. A., & Kerrigan, E. C. (2012). Towards a fixed point QP solver for predictive control. In *Proc. of 51st conf. on decision and control, Maui, Hawaii* (pp. 675–680).

Kerrigan, E. C., Jerez, J. L., Longo, S., & Constantinides, G. A. (2012). Number representation in predictive control. In *Proc. of IFAC nonlinear model predictive control conf., Noordwijkerhout, Netherlands* (pp. 60–67).

Knagge, G., Wills, A., Mills, A., & Ninness, B. (2009). ASIC and FPGA implementation strategies for model predictive control. In *Proc. European control conf., Budapest, Hungary*.

Ling, K. V., Yue, S. P., & Maciejowski, J. M. (2006). A FPGA implementation of model predictive control. In *Proc. American control conf., Minneapolis, Minnesota* (pp. 1930–1935).

Mayne, D. Q., & Rawlings, J. B. (2009). *Model predictive control: theory and design*. Madison,WI: Nob Hill Publishing, LCC.

Nedelcu, V., & Necoara, I. (2012). Iteration complexity of an inexact augmented Lagrangian method for constrained MPC. In *Proceedings of the IEEE conference on decision and control, Maui, Hawaii* (pp. 650–655).

Nedelcu, V., Necoara, I., & Dinh, Q.T. (2013). Computational complexity of inexact gradient augmented lagrangian methods: application to constrained MPC. ArXiv Preprint arXiv:1302.4355.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, *103*(1), 127–152.

Patrinos, P., & Bemporad, A. (2012). An accelerated dual gradient-projection algorithm for linear model predictive control. In *Proc. of 51st conf. on decision and control, Maui, Hawaii* (pp. 662–667).

Patrinos, P., & Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, *59*(1), 18–33.

Patrinos, P., Guiggiani, A., & Bemporad, A. (2013). Fixed-point dual gradient projection for embedded model predictive control. In *Proc. European control conference, Zurich, Switzerland* (pp. 3602–3607).

Richter, S., Jones, C. N., & Morari, M. (2009). Real-time input-constrained MPC using fast gradient methods. In *Proceedings of 48th conf. on decision and control, Shanghai, China* (pp. 7387–7393).

Richter, S., Morari, M., & Jones, C. N. (2011). Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method. In *Proc. of 50th conf. on decision and control and European control, Orlando, Florida* (pp. 5223–5229).

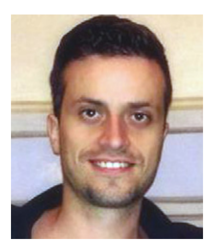
Rubagotti, M., Patrinos, P., & Bemporad, A. (2014). Stabilizing linear model predictive control under inexact numerical optimization. *IEEE Transactions on Automatic Control*, *59*(6), 1660–1666.

Tseng, P. (2008). *On accelerated proximal gradient methods for convex–concave optimization. Technical report*. Department of Mathematics, University of Washington.

Wang, Y., & Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, *18*(2), 267–278.

Wilkinson, J. H. (1994). *Rounding errors in algebraic processes*. Dover Publications.

Panagiotis Patrinos received his Ph.D. in Control and Optimization, M.Sc. in Applied Mathematics and M.Eng. in Chemical Engineering, all from the National Technical University of Athens, in 2010, 2005 and 2003, respectively. After receiving his Ph.D., he was a postdoctoral fellow at the University of Trento (2010–2011) and subsequently at IMT Institute for Advanced Studies Lucca, Italy (2011–2012), where he is currently an Assistant Professor. During fall 2014 he held a visiting Professor position in the department of Electrical Engineering at Stanford University. His current research interests are focused on devising efficient algorithms for large-scale distributed optimization with applications in embedded model predictive control (MPC) and machine learning. He is also interested in stochastic and risk-averse optimization with applications in the energy and power systems domain.



Alberto Guiggiani is currently a Ph.D. Candidate in the Dynamical Systems, Control and Optimization research unit of IMT Institute for Advanced Studies Lucca (Italy). In 2014 he was Visiting Scholar at the Department of Aerospace Engineering of the University of Michigan (US). Previously, he received his Master's degree (with honors) in Automation Engineering from the University of Florence (Italy) in 2011, after spending one semester as visiting student at the University of Lancaster (UK) in 2009 and one semester at the National Research Council of Italy and the Italian Institute of Technology of Genoa (Italy) in 2010. His research interests include embedded implementations of Model Predictive Control for aerospace and automotive applications, and optimization algorithms in finite precision arithmetic.

Alberto Bemporad received his master's degree in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. He spent the 1996/97 academic year at the Center for Robotics and Automation, Department of Systems Science & Mathematics, Washington University, St. Louis, as a visiting researcher. In 1997–1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a senior researcher in 2000–2002. In 1999–2009 he was with the Department of Information Engineering of the University of Siena, Italy, becoming an associate professor in 2005. In 2010–2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. In 2011 he joined the IMT Institute for Advanced Studies Lucca, Italy as a full professor, where he also became the director in 2012. In 2011 he cofounded ODYS S.r.l., a spinoff company of IMT Lucca. He has published more than 250 papers in the areas of model predictive control, hybrid systems, automotive control, multiparametric optimization, computational geometry, robotics, and finance. He is author or coauthor of various MATLAB toolboxes for model predictive control design, including the Model Predictive Control Toolbox (The Mathworks, Inc.). He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001–2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002–2010.

He received the IFAC High-Impact Paper Award for the 2011–14 triennial. He has been an IEEE Fellow since 2010.