Certification of the proximal gradient method under fixed-point arithmetic for box-constrained QP problems*

Pablo Krupa[a,*], Omar Inverso[b], Mirco Tribastone[c], Alberto Bemporad[c]

[a]*Systems Engineering and Automation Department, Universidad de Sevilla, Seville, Spain.*
[b]*Gran Sasso Science Institute (GSSI), L'Aquila, Italy*
[c]*IMT - School for Advanced Studies, Lucca, Italy*

Abstract

In safety-critical applications that rely on the solution of an optimization problem, the certification of the optimization algorithm is of vital importance. Certification and suboptimality results are available for a wide range of optimization algorithms. However, a typical underlying assumption is that the operations performed by the algorithm are exact, i.e., that there is no numerical error during the mathematical operations, which is hardly a valid assumption in a real hardware implementation. This is particularly true in the case of fixed-point hardware, where computational inaccuracies are not uncommon. This article presents a certification procedure for the proximal gradient method for box-constrained QP problems implemented in fixed-point arithmetic. The procedure provides a method to select the minimal fractional precision required to obtain a certain suboptimality bound, indicating the maximum number of iterations of the optimization method required to obtain it. The procedure makes use of formal verification methods to provide arbitrarily tight bounds on the suboptimality guarantee. We apply the proposed certification procedure on the implementation of a non-trivial model predictive controller on 32-bit fixed-point hardware.

Keywords: Convex optimization, Embedded Systems, Predictive control, Fixed-point arithmetic, Gradient method, Certification

1. Introduction

Quadratic programming (QP) problems arise in various areas of systems engineering and control, such as model predictive control (MPC), see Rawlings et al. (2017), or reference governors, see Garone et al. (2017), to name a few. Various practical control-related applications, such as the ones listed above, require solving parameter-dependent QP problems at regular intervals on embedded hardware, which poses a challenge due to computational and memory limitations. In recent years there has been a significant advance in this area due to the proposal of efficient QP solvers, some of them for generic QP problems, such as the OSQP solver presented in Stellato et al. (2020), and some tailored to specific problems, such as the solvers proposed in Krupa et al. (2021b); Frison and Diehl (2020), which address MPC optimization problems.

In many practical applications of MPC, such as safety-critical systems and space applications, the certification of the maximum number of iterations required by the optimization algorithm and a guarantee of the suboptimality of its provided solution are mandatory for real deployment. Most solvers are based on optimization algorithms with well-known convergence and suboptimality guarantees. The issue is that these guarantees are typically derived considering ideal conditions, e.g., under the assumption that the mathematical operations performed by the algorithm are error-free; an assumption, however, that is no longer valid when the optimization algorithm is implemented on hardware. This is particularly noticeable on fixed-point hardware, where quantization and round-off errors may lead to significant differences with respect to the "exact" counterpart. The magnitude of this difference depends on the number of fractional bits, which must be selected large enough to provide the required guarantees.

In linear-time-invariant (LTI) MPC, the use of *explicit* MPC (Bemporad, 2019) instead of an iterative solver provides a direct certification of the computation time. However, explicit MPC may require a considerable amount of memory to implement and is only applicable to LTI systems (Bemporad, 2019).

In Patrinos et al. (2015), the authors present a dual gradient-projection algorithm for MPC tailored to fixed-point arithmetics. The authors present convergence guarantees and concrete guidelines for selecting the fractional precision to obtain the required suboptimality tolerance. The analysis is done using the notion of the *inexact oracle* from Devolder et al. (2014), which presents a generic framework for analyzing first-order optimization algorithms in which the oracle provides inexact information. This framework can be used to derive convergence rates when inexact gradient information is available by considering the maximum error when computing the gradient. It has, however, two downsides when applied to fixed-point arithmetic. The first is that it only considers errors in the gradient information, i.e., it considers the other operations performed by the algorithm to be exact, which may not always be the case in fixed-point precision. Second, the convergence results are pre-

*This work was supported in part by Grant PDC2021-121120-C21 funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR", and in part by Grant Margarita Salas (grant number 20122) funded by the Ministerio de Universidades and the European Union (NextGenerationEU).
*Corresponding author
Email addresses: pkrupa@us.es (Pablo Krupa), omar.inverso@gssi.it (Omar Inverso), mirco.tribastone@imtlucca.it (Mirco Tribastone), alberto.bemporad@imtlucca.it (Alberto Bemporad)

sented in terms of the average of the iterates of the algorithm (instead of with respect to the current iterate), whose value is not generally available in fixed-point arithmetic, since its computation requires dividing by the current number of iterations.

Another approach for analyzing the error propagation is to use *affine arithmetic* (Fang et al., 2003; Vakili et al., 2013). This framework provides less conservative error bounds than simply considering the worst-case error due to how it handles error propagation in affine operations (addition, substraction and multiplication by a constant), although the error bounds are still conservative in the presence of multiplications between variables. The bounds can be improved by taking a probabilistic approach, as proposed in Fang et al. (2003), at the expense of no longer having a guaranteed certification. This framework was used in Nadales et al. (2022) to certify the minimum number of fractional bits required to satisfy the desired error bound when performing Lipschitz interpolation for data-driven learning-based control. In this case a tight certification could be proposed due to the simplicity of the algorithm, which only required affine operations. However, its application to iterative convex optimization algorithms would provide conservative results, due, precisely, to their iterative nature along with the presence of multiplication operations.

A non-conservative approach for analyzing error propagation in fixed-point arithmetic was proposed in Simić et al. (2022). The technique allows to check arbitrarily tight error bounds via a bit-vector encoding into integer arithmetics and then into propositional satisfiability, which allows to use mature SAT-based technology. The authors use their procedure to calculate accurate error bounds on a first-order optimization algorithm up to a given number of iterations. However, due to the bit-precise encoding the computational cost of the analysis becomes prohibitive after only a few iterations, even for small optimization problems; this is an issue since first-order methods may require a significant number of iterations to converge.

In this article we analyze the implementation of the proximal gradient method (PGM) (Parikh and Boyd, 2013) under fixed-point arithmetic applied to strongly convex QP problems with box constraints; a choice motivated by its simplicity but practical relevance and by its linear convergence guarantees under exact arithmetic. We provide convergence guarantees in terms of the maximum number of iterations of the algorithm as well as suboptimality guarantees of its output. Our certificate is based on the formal verification procedure presented in Simić et al. (2022), which we use to derive arbitrarily tight bounds on the quantities that determine the convergence and suboptimality guarantees. The proposed approach provides a procedure for selecting the minimum fractional precision required to guarantee the given suboptimality and computation-time specifications. Notably, the important difference with respect to Simić et al. (2022) is that the verification only needs to analyze a single iteration of the PGM, thus remaining tractable for a wider range of problems. The main features of our approach are:
(i) The subptimality guarantees are provided in terms of the output of the algorithm, instead of for the averaged iterates used in Patrinos et al. (2015).
(ii) The bounds that determine the suboptimality guarantees are obtained using the formal verification procedure from Simić et al. (2022), which allows us to provide a non-conservative error-bound of the gradient computation, in that the exact maximum error committed in the computation of the gradient can be approximated to an arbitrarily large precision.
(iii) The certification results are formal guarantees, instead of the probabilistic ones that would be obtained using probabilistic *affine arithmetic* (Fang et al., 2003) or Monte Carlo analysis (Saracco et al., 2012).
We present the application of our approach on a non-trivial MPC problem implemented using 32-bit fixed-point arithmetic.

Notation: Given two vectors $x, y \in \mathbb{R}^n$, $x \leq (\geq)\ y$ denotes componentwise inequalities and $\langle x, y\rangle$ is their standard inner product. The standard Euclidean norm of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\| \doteq \sqrt{\langle x, x\rangle}$. The closed ball of radius $r \geq 0$ in $\mathbb{R}^n$ is defined as the set $\mathcal{B}_r^n \doteq \{x \in \mathbb{R}^n : \|x\| \leq r\}$. The indicator function of a set $\mathcal{C}$ is denoted by $\mathcal{I}_\mathcal{C}$, i.e., $\mathcal{I}_\mathcal{C}(x) = 0$ if $x \in \mathcal{C}$ and $\mathcal{I}_\mathcal{C}(x) = \infty$ if $x \notin \mathcal{C}$. The subdifferential of a function f is denoted by ∂f. We denote by $\mathbb{R}^+$ the set of strictly positive real numbers. We denote by $\mathbb{R}^n_{(p.q)} \subset \mathbb{R}^n$ the set of vectors whose integer and fractional parts are representable using p and q binary digits, respectively. This notion readily extends to the space of matrices $\mathbb{R}^{n\times m}$, where $\mathbb{R}^{n\times m}_{(p.q)} \subset \mathbb{R}^{n\times m}$ represents the space of matrices whose every element is representable in $\mathbb{R}_{(p.q)}$. For any set $\mathcal{C} \subseteq \mathbb{R}^n$, we denote $\mathcal{C}_{(p.q)} = \{x \in \mathbb{R}^n_{(p.q)} : x \in \mathcal{C}\}$. It is obvious that $x \in \mathcal{C}_{(p.q)} \implies x \in \mathcal{C}$, but not vice-versa. It is also easy to see that $\mathcal{C}_{(p'.q')} \subseteq \mathcal{C}_{(p.q)}$ for any $p \geq p'$ and $q \geq q'$, and therefore that $x \in \mathcal{C}_{(p'.q')} \implies x \in \mathcal{C}_{(p.q)}$.

2. Exact proximal gradient method

We consider the class of strongly-convex QP problems

$$\begin{aligned} \min_{x\in\mathbb{R}^n}\ & \frac{1}{2}x^\top Q x + c^\top x \\ \text{s.t. } & \ell \leq x \leq u, \end{aligned} \tag{$\mathcal{P}$}$$

where $c, \ell, u, \in \mathbb{R}^n_{(p'.q')}$, with $\ell < u$, and $Q \in \mathbb{R}^{n\times n}_{(p'.q')}$, for some finite positive p' and q'. In this article we are interested in finding a suboptimal solution of problem $(\mathcal{P})$, using the proximal gradient method (PGM) (Parikh and Boyd, 2013), for any given realization of the ingredients of problem $(\mathcal{P})$ satisfying

$$Q \in \mathbb{Q} \subset \mathbb{R}^{n\times n}_{(p'.q')},\ c \in \{c \in \mathbb{R}^n_{(p'.q')} : c_{\min} \leq c \leq c_{\max}\}, \tag{1a}$$

$$\ell \in \{\ell \in \mathbb{R}^n_{(p'.q')} : \ell_{\min} \leq \ell \leq l_{\max}\}, \tag{1b}$$

$$u \in \{u \in \mathbb{R}^n_{(p'.q')} : u_{\min} \leq u \leq u_{\max}\}, \tag{1c}$$

where $c_{\min}, c_{\max}, \ell_{\min}, \ell_{\max}, u_{\min}, u_{\max} \in \mathbb{R}^n_{(p'.q')}$, $c_{\min} \leq c_{\max}$, $\ell_{\max} < u_{\min}$, and $\mathbb{Q}$ is a compact set. In particular, we are interested in providing convergence guarantees when implementing the PGM in fixed-point arithmetic. To that end, let us start by recalling the classical "exact" PGM, i.e., its implementation when operating using exact arithmetic.

We denote by $\mathbb{P}$ the set of problems $(\mathcal{P})$ whose ingredients satisfy (1). Let $\mathcal{X} \doteq \{x \in \mathbb{R}^n : \ell \leq x \leq u\}$, $f : \mathbb{R}^n \to \mathbb{R}$ be given by $f(x) = \frac{1}{2}x^\top Q x + c^\top x$, $x^* \in \mathbb{R}^n$ be the optimal solution of problem $(\mathcal{P})$ and f^* its optimal value, i.e., $f^* = f(x^*)$. Let $L, \sigma \in \mathbb{R}^+$ be the largest and smallest smoothness and strong convexity parameters of f for any realization of $(\mathcal{P}) \in \mathbb{P}$, i.e., the scalars for which the well-known inequalities

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2,$$
$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{\sigma}{2}\|x - y\|^2,$$

are satisfied $\forall x, y \in \mathbb{R}^n$ for any given $(\mathcal{P}) \in \mathbb{P}$.

Algorithm 1: PGM applied to $(\mathcal{P}) \in \mathbb{P}$

Input: $x^0 \in \mathcal{X}^0$, $0 < \rho \leq L^{-1}$

1 **For each** $k = 0, 1, 2, \ldots$ **repeat**

2 $\quad x^{k+1} \leftarrow \mathcal{T}_\rho(x^k) = \min\{u, \max\{\ell, x^k - \rho \nabla f(x^k)\}\}$

Algorithm 1 shows the PGM applied to a realization of problem $(\mathcal{P}) \in \mathbb{P}$, where the min and max operators are taken componentwise and $0 < \rho \leq L^{-1}$. It generates a sequence $\{x^k\} \in \mathcal{X}$ starting at an initial point $x^0 \in \mathcal{X}^0$, where $\mathcal{X}^0 \subseteq \mathcal{X}$ is the set of possible initial guesses. Step 2 of the algorithm performs the operator $\mathcal{T}_\rho : \mathcal{X} \to \mathcal{X}$ given by

$$\mathcal{T}_\rho(x) \doteq \arg\min_{y \in \mathcal{X}} \frac{1}{2} \|y - (x - \rho \nabla f(x))\|^2 ,$$

which, when particularized to problem $(\mathcal{P})$, is the evaluation of the *proximal* operator (Parikh and Boyd, 2013) at $x - \rho \nabla f(x)$. Operator $\mathcal{T}_\rho$ in this setting can also be viewed as the evaluation of the composite gradient mapping (Nesterov, 2013, §2). Furthermore, Algorithm 1 is equivalent in this case to the projected gradient method (Parikh and Boyd, 2013, §4,2).

The following theorem recalls the linear convergence of Algorithm 1 when working under "exact" arithmetic. In the following section we will derive a similar result when operating under fixed-point arithmetic.

Theorem 1 (Theorem 10.29 from Beck (2017))**.** *Let $\{x^k\}$ be the sequence generated by Algorithm 1 starting at $x^0 \in \mathcal{X}^0$ applied to $(\mathcal{P}) \in \mathbb{P}$. Then,*

(i) $\|x^k - x^*\|^2 \leq (1 - \rho\sigma)^k \|x^0 - x^*\|^2$, $\forall k \geq 0$,

(ii) $f(x^k) - f^* \leq \dfrac{1}{2\rho} (1 - \rho\sigma)^k \|x^0 - x^*\|^2$, $\forall k \geq 1$.

3. Proximal gradient method in fixed-point arithmetic

The maximum number of iterations of Algorithm 1 required to guarantee a given suboptimality can be certified using Theorem 1. Indeed, an immediate result of Theorem 1 is that $\|x^k - x^*\|^2 \leq \epsilon$, for a given $\epsilon \in \mathbb{R}^+$, is satisfied for every iteration k satisfying

$$k \geq \frac{\log\left(\dfrac{\epsilon}{\|x^0 - x^*\|^2}\right)}{\log(1 - \rho\sigma)}. \tag{2}$$

We are now interested in providing an iteration and suboptimality certification when Algorithm 1 is implemented using fixed-point arithmetic. Therefore, let us consider Algorithm 1 when working under fixed-point arithmetic for some predetermined choice of integer and fractional precision $(p.q)$ satisfying $p \geq p'$ and $q \geq q'$, where we recall that $(p'.q')$ is the precision under which the ingredients of $(\mathcal{P})$ are representable. Under this paradigm, we can view the implementation of the PGM algorithm as performing an inexact proximal operator, where the source of inexactness is due to the fixed-point arithmetic and representation of variables.

Algorithm 2 shows the implementation of Algorithm 1 under fixed-point arithmetic. It generates a sequence of iterates $\{\hat{x}^k\} \in \mathcal{X}_{(p.q)}$ starting from an initial point $\hat{x}^0 \in \hat{\mathcal{X}}^0_{(p.q)} \subseteq \mathcal{X}_{(p.q)}$.

Algorithm 2: PGM applied to $(\mathcal{P}) \in \mathbb{P}$ under fixed-point arithmetic with precision $(p.q)$

Input: $\hat{x}^0 \in \hat{\mathcal{X}}^0_{(p.q)}$, $0 < \rho \leq L^{-1}$, $\hat{\epsilon} \in \mathbb{R}^+_{(p.q)}$, $k_{\max} \in \mathbb{R}^+$

1 **For each** $k = 0, 1, 2, \ldots$ **repeat**

2 $\quad \hat{x}^{k+1} \leftarrow \min\{u, \max\{\ell, \hat{x}^k - \hat{g}_\rho(\hat{x}^k)\}\}$

3 **until** $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$ *or* $k \geq k_{\max}$

Step 2 evaluates the operator $\hat{g}_\rho : \mathcal{X}_{(p.q)} \to \mathcal{X}_{(p.q)}$, which is defined as the operator that performs the computation of $\rho \nabla f(\cdot)$ in the fixed-point paradigm. That is, $\hat{g}_\rho(\hat{x}^k)$ returns the result of the evaluation of $\rho \nabla f(\hat{x}^k)$ when performed under fixed-point arithmetic. Thus, $\rho \in \mathbb{R}^+_{(p.q)}$ is an obvious requirement of Algorithm 2. Additionally, the algorithm includes an exit condition given by the satisfaction of the condition $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$, where $\hat{d}^2 : \mathcal{X}_{(p.q)} \to \mathbb{R}^+_{(p.q)} \cup \{0\}$ is the operator that performs the computation of $\|\hat{x}^{k+1} - \hat{x}^k\|^2$ in fixed-point arithmetic. This exit condition plays a key role in the suboptimality guarantees provided in this section.

The value of $\hat{g}_\rho(\hat{x}^k)$ will generally differ from the exact value of the expression $\rho \nabla f(\hat{x}^k)$ due to the arithmetic errors that occur when using fixed-point arithmetic, thus leading to the source of discrepancy between the sequences generated by Algorithms 1 and 2. The magnitude of this discrepancy, which we formalize in the following definition, will depend on the value of the fractional precision q, with higher values of q obviously leading to smaller errors.

Definition 1. *Given a choice of q, we denote by $\Omega \in \mathbb{R}^+$ a scalar satisfying $\|\rho \nabla f(\hat{x}) - \hat{g}_\rho(\hat{x})\| \leq \Omega$, $\forall \hat{x} \in \mathcal{X}_{(p.q)}$, $\forall (\mathcal{P}) \in \mathbb{P}$.*

The rest of the operations in Step 2 of Algorithm 2 do not incur in any additional error, as formally stated in the following lemma, so they do not contribute towards the discrepancy between the "exact" and "fixed-point" implementations.

Lemma 1. *Let $\underline{v}, \overline{v}, \hat{x}, \hat{y} \in \mathbb{R}^n_{(p.q)}$, with $\underline{v} \leq \overline{v}$, and consider the set $\mathcal{C} \doteq \{v \in \mathbb{R}^n : \underline{v} \leq v \leq \overline{v}\}$. Then, the result of the fixed-point computations $\min\{\overline{v}, \max\{\underline{v}, \hat{x} + \hat{y}\}\}$ performed with any precision $(\hat{p}.\hat{q})$ satisfying $\hat{p} \geq p$ and $\hat{q} \geq q$ is the exact Euclidean projection of $\hat{x} + \hat{y}$ onto $\mathcal{C}$ if $\hat{x} + \hat{y}$ does not result in an overflow.*

The proof of the lemma is omitted because the claim is a direct result of the fact that the min, max and addition operations do not incur in any error under fixed-point arithmetic as long as there is no overflow in the addition.

Corollary 1. *Variable $\hat{x}^{k+1}$ obtained from Step 2 of Algorithm 2 is the exact Euclidean projection of $\hat{x}^k - \hat{g}_\rho(\hat{x}^k)$ onto $\mathcal{X}$, assuming that no overflow occurs during the computations. Therefore, $\hat{x}^k \in \mathcal{X}_{(p.q)} \subset \mathcal{X}$, $\forall k \geq 0$.*

The reader will note that Lemma 1 is only applicable to Algorithm 2 as long as there is no overflow during its execution, i.e., if the integer precision p is large enough. The certification tool presented in Section 4 can be used to compute the minimum value of p required to avoid overflow. Thus, we henceforth simply consider that p is chosen so that no overflow occurs during the execution of Algorithm 2.

Another useful consequence of Lemma 1 is presented in the following lemma, which states that a scalar Ω satisfying Definition 1 also bounds the error in the computation of $\hat{x}^{k+1}$.

Lemma 2. *Consider Algorithm 2 and let Ω satisfy Definition 1. Then, $\|\hat{x}^{k+1} - \mathcal{T}_\rho(\hat{x}^k)\| \leq \Omega$, $\forall \hat{x}^k \in \mathcal{X}_{(p.q)}$, $\forall(\mathcal{P}) \in \mathbb{P}$.*

Proof. The claim is a direct consequence of Corollary 1 and the fact that the projection operator to non-empty closed convex sets is non-expansive (Ryu and Boyd, 2016, §3.1). ■

We now present the main result of this section, where we characterize the local linear convergence of Algorithm 2, in terms of the error-bound Ω, when sufficiently far away from the optimal solution.

Theorem 2. *Let $\{\hat{x}^k\}$ be the sequence generated by Algorithm 2 applied to a realization of problem $(\mathcal{P}) \in \mathbb{P}$ with starting point $\hat{x}^0 \in \hat{\mathcal{X}}^0_{(p.q)}$ and taking $0 < \rho \leq L^{-1}$, $\rho \in \mathbb{R}_{(p.q)}$. Choose $\epsilon \in \mathbb{R}^+$ satisfying $\epsilon\rho\sigma > 4\Omega$, where $\Omega \in \mathbb{R}^+$ is given by Definition 1. Then, as long as $\|\hat{x}^{k+1} - x^*\| \geq \frac{\epsilon}{2}$, the sequence $\{\hat{x}^k\}$ satisfies:*

(i) $\|\hat{x}^k - x^*\|^2 \leq \left(\frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}\right)^k \|\hat{x}^0 - x^*\|^2$, $\forall k \geq 0$.

(ii) $f(\hat{x}^k) - f^* \leq \frac{1-4\Omega\epsilon^{-1}}{2\rho}\left(\frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}\right)^k \|\hat{x}^0 - x^*\|^2$, $\forall k \geq 1$.

Proof. See Appendix A.

Theorem 2 provides a linear convergence result similar to the one shown in Theorem 1, but where the convergence constant degrades by the factor $(1 - 4\Omega\epsilon^{-1})$. That is, the convergence guarantee worsens as Ω increases and as the desired suboptimality tolerance ϵ decreases. Since the theorem only holds as long as $\|\hat{x}^{k+1} - x^*\| \geq \frac{\epsilon}{2}$, we need to be able to check the satisfaction of this condition during the execution of Algorithm 2.

Lemma 3. $\|\hat{x} - \mathcal{T}_\rho(\hat{x})\| \leq 2\|\hat{x} - x^*\|$, $\forall \hat{x} \in \mathcal{X}$.

Proof. From (Alamo et al., 2019, Property 1.(i)), particularized to our problem formulation and notation, we have that

$$f(\mathcal{T}_\rho(\hat{x})) - f^* \leq \rho^{-1}\langle \hat{x} - \mathcal{T}_\rho(\hat{x}), \hat{x} - x^*\rangle - \frac{1}{2\rho}\|\hat{x} - \mathcal{T}_\rho(\hat{x})\|^2,$$

which along with $f(\mathcal{T}_\rho(\hat{x})) - f^* \geq 0$, leads to

$$\frac{1}{2}\|\hat{x} - \mathcal{T}_\rho(\hat{x})\|^2 \leq \langle \hat{x} - \mathcal{T}_\rho(\hat{x}), \hat{x} - x^*\rangle \leq \|\hat{x} - \mathcal{T}_\rho(\hat{x})\| \cdot \|\hat{x} - x^*\|$$

by making use of the Cauchy-Schwarz inequality. ■

The previous lemma allows us to guarantee that the condition $\|\hat{x}^k - x^*\| \geq \frac{\epsilon}{2}$ in Theorem 2 holds for the iterates of Algorithm 2 as long as we can guarantee that $\|\hat{x}^k - \mathcal{T}_\rho(\hat{x}^k)\| \geq \epsilon$. The following assumption allows us to use the exit condition of Algorithm 2 as a means to guarantee that the condition $\|\hat{x}^k - x^*\| \geq \frac{\epsilon}{2}$ is satisfied at iteration k.

Assumption 1. *The exit tolerance $\hat{\epsilon}$ of Algorithm 2 satisfies $\hat{d}^2(\hat{x}^k) \geq \hat{\epsilon} \implies \|\hat{x}^k - \mathcal{T}_\rho(\hat{x}^k)\|^2 \geq \epsilon^2$.*

In the following section we present a tractable procedure for certifying the satisfaction of this assumption. In practice, we find that one can choose ϵ so that the smallest value of $\hat{\epsilon}$ satisfying Assumption 1 is the smallest positive representable number in precision q, i.e., 2^{-q}. In this case the exit condition of Algorithm 2 becomes $\hat{d}^2(\hat{x}^k) = 0$.

Under Assumption 1, the convergence guarantee provided in Theorem 2 holds as long as the exit condition $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$ is not satisfied. If it is satisfied at some iteration k, then the convergence guarantee provided by Theorem 2 is only guaranteed to hold until iteration $k-1$. The following theorem provides the suboptimality guarantees of Algorithm 2 when the exit condition is satisfied at some iteration k. The result makes use of the bounds provided in the following definition. The following section will provide computationally tractable procedures for computing arbitrarily tight values of said bounds.

Definition 2. *Consider Algorithm 2. For a given choice of fractional precision q and tolerance $\hat{\epsilon}$, we denote by $\delta, \omega, \Theta \in \mathbb{R}^+$ the scalars satisfying*

$$\|\hat{x}^k - \mathcal{T}_\rho(\hat{x}^k)\| \leq \delta,\ \|\hat{x}^{k+1} - \mathcal{T}_\rho(\hat{x}^k)\| \leq \omega,\ \|\hat{x}^k - \hat{x}^{k+1}\| \leq \Theta$$

for all $\hat{x}^k \in \mathcal{X}_{(p.q)}$ satisfying $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$, $\forall(\mathcal{P}) \in \mathbb{P}$.

Theorem 3. *Let $\{\hat{x}^k\}$ be the sequence generated by Algorithm 2 applied to a realization of problem $(\mathcal{P}) \in \mathbb{P}$ with starting point $\hat{x}^0 \in \hat{\mathcal{X}}^0_{(p.q)}$ and taking $0 < \rho \leq L^{-1}$, $\rho \in \mathbb{R}_{(p.q)}$. Denote $T \doteq \sigma^{-1}(\rho^{-1} + L)$. Then, if $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$,*

(i) $\|\hat{x}^{k+1} - x^*\| \leq \omega + \delta T$,

(ii) $f(\hat{x}^{k+1}) - f^* \leq \rho^{-1}\left((\Theta + \Omega)(\omega + \delta T) + \frac{1}{2}\Theta^2\right)$.

Proof. From (Nesterov, 2013, Lemma 3), particularized to our notation, we have that $\|\hat{x}^k - \mathcal{T}_\rho(\hat{x}^k)\| \geq T^{-1}\|\mathcal{T}_\rho(\hat{x}^k) - x^*\|$, which leads to $\|\hat{x}^k - \mathcal{T}_\rho(\hat{x}^k)\| \leq \delta \implies \|\mathcal{T}_\rho(\hat{x}^k) - x^*\| \leq T\delta$. Claim *(i)* follows from adding the previous inequality with $\|\hat{x}^{k+1} - \mathcal{T}_\rho(\hat{x}^k)\| \leq \omega$ and applying the triangle inequality. By the same procedure, we also derive $\|\hat{x}^k - x^*\| \leq \delta(T+1)$. Claim *(ii)* then follows from particularizing Lemma 5.(i) to $y = x^*$, using the Cauchy-Schwarz inequality and then taking the previous inequalities along with the inequalities presented in Definitions 1 and 2. ■

The following corollary gathers the guarantees that are obtained from Algorithm 2 in terms of the error-bounds and tolerances presented throughout this section.

Corollary 2 (Suboptimality guarantee of Algorithm 2)**.** *Let $k_{\max} \geq \log\left(\frac{\epsilon^2}{4D}\right) / \log(C)$, where $C \doteq \frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}$ and D satisfies $D \geq \max_{\hat{x} \in \hat{\mathcal{X}}^0_{(p.q)}} \|\hat{x} - x^*\|^2$. The following hold:*

(i) If $\hat{d}^2(\hat{x}^k) \geq \hat{\epsilon}$ for all $k = \{0, \dots k_{\max}\}$, then

$$\|\hat{x}^{k_{\max}} - x^*\| \leq \frac{\epsilon}{2} \text{ and } f(\hat{x}^{k_{\max}}) - f^* \leq \frac{(\epsilon^2 - 4\Omega\epsilon)}{8\rho}.$$

(ii) If $\hat{d}^2(\hat{x}^k) < \hat{\epsilon}$ then $\|\hat{x}^{k+1} - x^*\| \leq \omega + \delta T$ and

$$f(\hat{x}^{k+1}) - f^* \leq \rho^{-1}\left((\Theta + \Omega)(\omega + \delta T) + \frac{1}{2}\Theta^2\right).$$

Remark 1. *From Lemma 2 we have that the bound ω from Definition 2 can be substituted by Ω from Definition 1. We find that for high fractional precision there can be an insignificant difference between the two quantities, thus not meriting the additional computation time required to compute ω.*

Algorithm 3: Example of input algorithm for formal verification

Parameters: $(p.q)$, $b \in \mathbb{R}_{(p.q)}$, $\hat{a} \in \mathbb{R}^+$, $\xi \in \mathbb{R}$, $\chi \in \mathbb{R}^+$
Non-deterministic: $a \in \mathcal{A} = \{a \in \mathbb{R}^m : ||a||_\infty \leq \hat{a}\}$
1 $r = \langle a, a \rangle$
2 $\mu = br$
3 **if** $|\texttt{err}(r)| > \chi$ **then** `FAIL`
4 **if** $\texttt{exact}(\mu) < \xi$ **then** `FAIL`

4. Obtaining error-bounds for Algorithm 1

This section presents procedures for obtaining arbitrarily tight values of the bounds Ω, δ, ω and Θ introduced in Definitions 1 and 2 as well as a procedure for checking the satisfaction of Assumption 1. The procedures consist on the application of the formal verification procedure presented in Simić et al. (2022) to check the conditions provided in the definitions and assumption. We first introduce the formal analysis technique for fixed-point arithmetic, and then show how it fits within our certification process.

4.1. Formal verification for fixed-point arithmetic

Simić et al. (2022) consider the problem of estimating the numerical accuracy of algorithms in fixed-point arithmetic with variables of arbitrary precision and possibly non-deterministic values. The idea is to re-compute in a greater precision the result of each fixed-point operation, so that the numerical error can be estimated based on the difference between the two values; at the same time, the different errors are in turn accounted for and propagated through the re-computations. When sufficient precision is used to store the re-computed values, this yields an accurate error tracking for each variable at any point of the computation. The techniques relies on a bit-precise encoding to transform the sequences of operations under analysis into operations in integer arithmetic over vectors of bits; these are in turn encoded as a SAT formula that is satisfiable if and only if the algorithm under analysis exceeds a given bound on the numerical error. The technique is quite accurate in that it allows to formally verify arbitrarily tight bounds on the numerical error up to a given number of iterations. On the other hand, as observed by the authors, the required program unfolding pass along with the bit-vector encoding ends up introducing considerable overhead; the analysis can become quickly intractable, even for small problems and a few iterations.

We now provide an illustrative example to introduce the concepts and notation relevant to this article. Consider Algorithm 3 working under a given fixed-point precision $(p.q)$, where r is a fixed-point variable, $\texttt{exact}\{r\}$ is its "exact" counterpart and $\texttt{err}\{r\}$ is its error, i.e., $\texttt{exact}\{r\} = r + \texttt{err}\{r\}$. The same applies for the other variables, such as μ, whose exact value will generally differ from its value computed in fixed-point arithmetic due to the multiplication operations. The error of r is propagated when computing the error of μ, i.e., $\texttt{exact}\{\mu\}$ will contain the value of $\langle a, a \rangle$. We can perform assessments on the fixed-point variables, their errors and their exact values (as shown in Steps 3 and 4) for all possible values of the non-deterministic inputs (variable a in this case). If all the assessments are satisfied for all possible values of the non-deterministic variables then the procedure will return a `PASS`. Otherwise, it will return a `FAIL`.

Algorithm 4: Algorithm for asserting Assumption 1

Parameters: $(p.q)$, $Q, u, l, c_{\min}, c_{\max}$, $\hat{\epsilon}$, ϵ^2
Non-deterministic: $\hat{x}^0 \in \mathcal{X}_{(p.q)}$, $(\mathcal{P}) \in \mathbb{P}$
1 $\hat{x}^{k+1} \leftarrow \min\{u, \max\{\ell, \hat{x}^k - \hat{g}_\rho(\hat{x}^k)\}\}$
2 $d \leftarrow \langle \hat{x}^k - \hat{x}^{k+1}, \hat{x}^k - \hat{x}^{k+1} \rangle$
3 **if** $d \geq \hat{\epsilon}$ **and** $\texttt{exact}\{d\} < \epsilon^2$ **then** `FAIL`

Algorithm 5: Algorithm for deriving Ω

Parameters: $(p.q)$, $Q, u, l, c_{\min}, c_{\max}$, Ω^2
Non-deterministic: $\hat{x}^0 \in \mathcal{X}_{(p.q)}$, $(\mathcal{P}) \in \mathbb{P}$
1 $\hat{g} \leftarrow \rho(Q\hat{x}^0 + c)$
2 $e \leftarrow \texttt{err}\{\hat{g}\}$
3 $v \leftarrow \langle e, e \rangle$
4 **if** $\texttt{exact}\{v\} > \Omega^2$ **then** `FAIL`

Algorithm 6: Algorithm for deriving Def. 2 bounds

Select: b $\in \{\delta^2, \omega^2, \Theta^2\}$
Parameters: $(p.q)$, $Q, u, l, c_{\min}, c_{\max}$, $\hat{\epsilon}$
Non-deterministic: $\hat{x}^0 \in \mathcal{X}_{(p.q)}$, $(\mathcal{P}) \in \mathbb{P}$
1 $\hat{x}^{k+1} \leftarrow \min\{u, \max\{\ell, \hat{x}^k - \hat{g}_\rho(\hat{x}^k)\}\}$
2 **if** $\langle \hat{x}^k - \hat{x}^{k+1}, \hat{x}^k - \hat{x}^{k+1} \rangle < \hat{\epsilon}$ **then**
3 **switch** b **do**
4 **case** δ^2 **do** $s \leftarrow \hat{x}^k - \hat{x}^{k+1}$
5 **case** ω^2 **do** $s \leftarrow \texttt{err}\{\hat{x}^{k+1}\}$
6 **case** Θ^2 **do**
7 $\texttt{err}\{\hat{x}^{k+1}\} \leftarrow 0$
8 $s \leftarrow \hat{x}^k - \hat{x}^{k+1}$
9 **end case**
10 **end switch**
11 **end if**
12 $v \leftarrow \langle s, s \rangle$
13 **if** $\texttt{exact}\{v\} > b$ **then** `FAIL`

Example 1. *We run the verification procedure presented in Simić et al. (2022) on Algorithm 3 with $p = 8$, $q = 8$, $m = 20$, $\hat{a} = 0.125$, $b = 1.5$, $\xi = 0$, and $\chi = 0.069580078125$. We obtain a* `PASS`*, that is, there is no value of a inside the box defined by $\hat{a}$ for which the assertions stated in Steps 3 and 4 of Algorithm 3 are violated. The result of this example highlights one of the main benefit of this procedure, which is the bound $\chi = 0.069580078125$. Variable r is the inner product of a by itself. As stated in Patrinos et al. (2015), the standard theoretical bound for the maximum error committed by an inner product is given by $\texttt{err}(\langle a, a \rangle) \leq 2^{-q} m$, which is equal to 0.078125 for the values of q and m in this example. However, the formal verification procedure has found that this theoretical bound can be improved to $\texttt{err}(\langle a, a \rangle) = 0.069580078125$, which is a 12.28% improvement. That is, the procedure may lead to tighter bounds on the errors committed by the fixed-point algorithm than the ones obtained by simply evaluating its execution under the theoretical worst-case scenario. Interestingly, considering the dimension of a ($m = 20$), the value of q, and the size of the box, thanks to our specific formulation of the problem at hand we were able to obtain this verification verdict in a very small amount of time and with limited hardware resources (5ms using a standard machine with an Intel i5 processor).*

4.2. Verification procedure for Algorithm 2

The application of the verification tool presented in the previous subsection to Algorithm 4 certifies if the given $\hat{\epsilon}$ and ϵ^2 satisfy Assumption 1. Its application to Algorithms 5 certifies if the provided value of Ω satisfies the condition presented in Definition 1. A `PASS` indicates that the given Ω^2 satisfies the condition $\|\mathcal{T}_\rho(\hat{x}^k) - \hat{x}^{k+1}\|^2 \leq \Omega^2$, $\forall \hat{x}^k \in \mathcal{X}_{(p.q)}$, $\forall(\mathcal{P}) \in \mathbb{P}$, whereas a `FAIL` indicates that there is at least one combination of $\hat{x}^k \in \mathcal{X}_{(p.q)}$ and $(\mathcal{P}) \in \mathbb{P}$ for which the condition is not satisfied. Note that the condition is asserted with respect to Ω^2, since the verification tool does not allow the use of the square-root operation. Similarly, for a given value of $\hat{\epsilon}$, Algorithm 6 is used to certify if the bounds δ, ω or Θ satisfy the conditions presented in Definition 2. Arbitrarily tight values of Ω, δ, ω or Θ can be obtained by applying the bisection method to Algorithms 5 and 6. Note that Algorithms 4, 5 and 6 only execute a single iteration of Algorithm 2. Thus, the proposed verification procedure remains tractable for moderately-sized problems, in contrast with the approach taken in Simić et al. (2022).

Remark 2. *The verification tool from Simić et al. (2022) can be configured to return a `FAIL` in the event of a numerical overflow. Thus, we can verify that no overflow occurst in Algorithm 2 for a given choice of the integer precision p if the tool does not fail due to an overflow when applied to Algorithm 4.*

5. Numerical case study

We apply the verification procedures presented in the previous section to certify the fixed-point implementation of the PGM to solve the optimization problem of a linear MPC controller for a discrete-time, time-invariant system given by a state-space model $\tilde{x}(t) = A\tilde{x}(t) + B\tilde{u}(t)$, where $\tilde{x}(t) \in \mathbb{R}^{n_x}$ and $\tilde{u}(t) \in \mathbb{R}^{n_u}$ are the state and control input at sample time t.

In particular, we consider the system of three masses connected by springs presented in (Krupa et al., 2021a, §3), where we take the mass all three objects equal to 1kg and the spring constants as 1N/m. The 6-dimensional system state is given by the position and velocity of each of the three objects, while the control input is given by the two external forces applied to the outer objects. We take the following MPC formulation:

$$\min \sum_{i=0}^{N_p-1} \left(\|\tilde{x}_i - \tilde{x}_r\|^2_{W_x} + \|\tilde{u}_i - \tilde{u}_r\|^2_{W_u}\right) + \|\tilde{x}_{N_p} - \tilde{x}_r\|^2_P \quad (3a)$$

$$\text{s.t. } \tilde{x}_0 = \tilde{x}(t) \quad (3b)$$

$$\tilde{x}_{i+1} = A\tilde{x}_i + B\tilde{u}_i,\ i \in \mathbb{Z}_0^{N_p-1} \quad (3c)$$

$$\tilde{u}_i = \tilde{u}_{N_c-1},\ i \in \mathbb{Z}_{N_c}^{N_p-1} \quad (3d)$$

$$\tilde{u}_- \leq \tilde{u}_i \leq \tilde{u}_+,\ i \in \mathbb{Z}_0^{N_c-1}, \quad (3e)$$

where $N_c \in \mathbb{R}^+$ is the control horizon; $N_p \geq N_c$ is the prediction horizon; $W_x, P \in \mathbb{R}^{n_x \times n_x}$ and $W_u \in \mathbb{R}^{n_u \times n_u}$ are positive definite; (x_r, u_r) are the state and input references; and $\tilde{u}_-, \tilde{u}_+ \in \mathbb{R}^{n_u}$ satisfying $\tilde{u}_- \leq \tilde{u}_+$ define the bounds on the control input. We take $N_c = 2$, $N_p = 5$, $\tilde{u}_+ = (0.5, 0.5)$, $\tilde{u}_- = -\tilde{u}_+$, $W_x = 0.5I_{n_x}$, $W_u = 0.25I_{n_u}$ and P as the solution of the associated discrete algebraic Riccati equation. Problem (3) can be transformed into $(\mathcal{P})$ by eliminating the states and rewriting it in condensed form, see e.g., Richter (2012); Jerez et al. (2011), leading to a $(n_u N_c)$-dimensional QP problem. In this case, ingredients Q, ℓ and u of problem $(\mathcal{P})$ are fixed, whereas the value of c will depend on the value of the reference (x_r, u_r) as well as the current state $\tilde{x}(t)$. We compute $c_{\min}$ and $c_{\max}$ by assuming that the position of the objects belong to the interval $[-0.5, 0.5]$m and the velocities to $[-1, 1]$m/s. A non-deterministic Q would be taken if we allowed the possibility of changing the weight W_u online or if we considered a time-varying model of the system.

We now certify the PGM applied to the resulting condensed MPC problem when implemented in fixed-point arithmetic on a 32-bit device, where we take $p = 10$ for the integer precision and $q = 21$ for the fractional precision (the remaining bit is used for storing the sign). We store the matrices of the QP problem in the selected precision. The resulting problem has $L = 4.9645$ and $\sigma = 0.3532$. We take ρ as the largest number representable in $\mathbb{R}_{(p.q)}$ that satisfies $\rho \leq 1/L$.

In our formal verification procedure we used the prototype tool of Simić et al. (2022) for generating the bit-vector encoding, CBMC 5.4 (Clarke et al., 2004) for generating the SAT formula from the bit-vector encoding and MiniSat (Eén and Sörensson, 2004) to check for the satisfiability of the SAT formula. All computations are performed on an Intel i5 processor running at 1.6GHz. We start by computing the value of Ω by selecting an initial value of Ω^2 and then applying the bisection method on the verification of Algorithm 5 with an exit tolerance of 10^{-14}, i.e., until the difference between the largest and smallest values of Ω^2 resulting in a `FAIL` and a `PASS`, respectively, is smaller than 10^{-14}. Table 1 shows the value of Ω obtained from this procedure, along with the selected exit tolerance, initial guess of Ω^2, number of tests resulting in a `PASS`, number of tests resulting in a `FAIL`, average computation times of calls resulting in a `PASS` or `FAIL`, and total computation time of the bisection method. We take $\hat{\epsilon} = 2^{-q}$, which is the smallest value it can take, and then find the largest value of ϵ satisfying Assumption 4 by applying the bisection method on Algorithm 4. The results are presented in Table 1, where we note that the value of ϵ satisfies $\epsilon > 4\Omega/(\rho\sigma) = 9.6195 \cdot 10^{-5}$. Therefore, we can use the exit condition $\hat{d}^2(\hat{x}^k) = 0$. Finally, we obtain the bounds δ and Θ following the same bisection procedure used to compute Ω. The results are also presented in Table 1. We take $\omega = \Omega$, as stated in Remark 1.

Plugging the results into Corollary 2.*(ii)*, we obtain the following: $k_{\max} = 250$, if $\hat{d}^2(\hat{x}^k) = 0$ then $\|\hat{x}^{k+1} - x^*\| \leq 0.0389$ and $f(\hat{x}^{k+1}) - f^* \leq 2.7165 \cdot 10^{-4}$, are the best suboptimality bounds that can be guaranteed, since the ones from Corollary 2.*(i)* are smaller. The value of $k_{\max}$ required to obtain the same $\epsilon/2$-suboptimality under exact arithmetic is 217, c.f., (2).

6. Conclusions

This article has presented a procedure for certifying the implementation of the PGM under fixed-point arithmetic when applied to strongly-convex box-constrained QP problems. We have proven that the PGM maintains a linear convergence guarantee when sufficiently far away from the optimal solution, indicated by the choice of ϵ, whose value can be reduced up to a maximum bound given by the fixed-point error-bound. We have then presented a procedure based on recent formal verification tools to obtain a arbitrarily tight values of this error-bound and the other bounds that characterize the suboptimality of the output of the PGM. Finally, we have shown that the computation times of the proposed verification procedures are tractable for a non-trivial MPC example.

Bound b	Value	b^2	Tol.	# P/F	Av. `PASS` time [s]	Av. `FAIL` time [s]	Total time [s]
Ω	$1.711 \cdot 10^{-6}$	2^{-2q}	10^{-14}	4/8	218.1	982.8	8738.8
ϵ	$6.8949 \cdot 10^{-4}$	$(4\Omega/(\rho\sigma))^2$	10^{-9}	11/4	73.8	75.2	1117.4
δ	$1.383 \cdot 10^{-3}$	2^{-q}	10^{-9}	10/4	84.7	90.8	1215.0
Θ	$1.381 \cdot 10^{-3}$	δ^2	10^{-9}	3/8	12.1	28.3	266.6

Table 1: Bounds obtained from the verification procedures for the three-mass-spring case study.

Appendix A. Proofs and auxiliary lemmas

We start by providing two lemmas whose results are used in the proofs of Theorems 2 and 3.

Lemma 4. *Consider Algorithm 2. For any $\alpha \in \mathbb{R}^+$,*

$$\alpha \left(\hat{x}^k - \hat{x}^{k+1} - \hat{g}_\rho(\hat{x}^k) \right) \in \partial \mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1}),\ \forall k \geq 0.$$

Proof. The claim follows from $\hat{x}^{k+1}$ being the Euclidean projection of $\hat{x}^k - \hat{g}_\rho(\hat{x}^k)$ onto $\mathcal{X}$ (see Corollary 1) along with the optimality condition of the projection operator (Bertsekas, 2009, Prop. 5.4.7) and the equivalence between the subdifferential of the indicator function of a non-empty convex set and its normal cone (Bertsekas, 2009, Example 5.4.1). ■

The following lemma particularizes (Alamo et al., 2019, Property 1) to the fixed-point PGM paradigm.

Lemma 5. *Consider Algorithm 2 and let $v^k \in \mathcal{B}^n_\Omega$ be the vectors that satisfy $\rho \nabla f(\hat{x}^k) = \hat{g}_\rho(\hat{x}^k) + v^k$ for every $k \geq 0$. Denote $s^k \doteq \hat{x}^k - \hat{x}^{k+1}$. Then,*

$$f(\hat{x}^{k+1}) - f(y) \leq \rho^{-1}\langle s^k + v^k, \hat{x}^{k+1} - y\rangle + \frac{1}{2\rho}\|s^k\|^2 \qquad (i)$$

$$= \rho^{-1}\langle s^k, \hat{x}^k - y\rangle - \frac{1}{2\rho}\|s^k\|^2 + \rho^{-1}\langle v^k, \hat{x}^{k+1} - y\rangle \qquad (ii)$$

$$= \frac{1}{2\rho}\|\hat{x}^k - y\|^2 - \frac{1}{2\rho}\|\hat{x}^{k+1} - y\|^2 + \rho^{-1}\langle v^k, \hat{x}^{k+1} - y\rangle \qquad (iii)$$

Proof. From Lemma 4 we have that $\rho^{-1}(\hat{x}^k - \hat{x}^{k+1} - \hat{g}_\rho(\hat{x}^k)) \in \partial \mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1})$. Therefore, from the definition of the subdifferential (Parikh and Boyd, 2013, §2.3), we have that

$$\mathcal{I}_{\mathcal{X}}(y) \geq \mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1}) + \rho^{-1}\langle \hat{x}^k - \hat{x}^{k+1} - \hat{g}_\rho(\hat{x}^k), y - \hat{x}^{k+1}\rangle,$$

where taking $y \in \mathcal{X}$ and recalling that $\hat{x}^{k+1} \in \mathcal{X}$ (see Corollary 1), leads to

$$0 \geq \rho^{-1}\langle \hat{x}^k - \hat{x}^{k+1} - \hat{g}_\rho(\hat{x}^k), y - \hat{x}^{k+1}\rangle. \qquad (A.4)$$

From the convexity of f we have that

$$f(y) \geq f(\hat{x}^k) + \langle \nabla f(\hat{x}^k), y - \hat{x}^k\rangle. \qquad (A.5)$$

Additionally, from the L-smoothness of f and since $\rho \leq L^{-1}$, we have that

$$f(\hat{x}^k) \geq f(\hat{x}^{k+1}) - \langle \nabla f(\hat{x}^k), \hat{x}^{k+1} - \hat{x}^k\rangle - \frac{1}{2\rho}\|s^k\|^2. \qquad (A.6)$$

Claim (i) follows from adding (A.4), (A.5) and (A.6) along with the definition of v^k. Claims (ii) and (iii) then follow from simple algebraic manipulations; c.f. Property 1.(i) in Alamo et al. (2019). ■

We now present the proof of Theorems 2, which closely follows the proofs of (Beck, 2017, Theorem 10.16 and Theorem 10.29), although various modifications have to be made to extend the results to the fixed-point arithmetic paradigm.

Proof of Theorem 2. Let $s^k \doteq \hat{x}^{k+1} - \hat{x}^k$ and $\gamma^k \doteq \hat{x}^k - x^*$. Consider the function $\psi : \mathcal{X} \to \mathbb{R}$ given by

$$\psi(y) = f(\hat{x}^k) + \langle \nabla f(\hat{x}^k), y - \hat{x}^k\rangle + \mathcal{I}_{\mathcal{X}}(y) + \frac{1}{2\rho}\|y - \hat{x}^k\|^2.$$

Since ψ is an ρ^{-1}-strongly convex function, it follows from (Beck, 2017, Theorem 5.24) that

$$\psi(y) - \psi(\hat{x}^{k+1}) \geq \langle \mu, y - \hat{x}^{k+1}\rangle + \frac{1}{2\rho}\|y - \hat{x}^{k+1}\|^2, \qquad (A.7)$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}$, $\forall \mu \in \partial\psi(\hat{x}^{k+1})$. Since $\rho \leq L^{-1}$, we have that $f(\cdot)$ satisfies the well-known descent lemma (Beck, 2017, Lemma 5.7)

$$f(\hat{x}^{k+1}) \leq f(\hat{x}^k) + \langle \nabla f(\hat{x}^k), \hat{x}^{k+1} - \hat{x}^k\rangle + \frac{1}{2\rho}\|\hat{x}^{k+1} - \hat{x}^k\|^2,$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}$, $\forall y \in \mathcal{X}$, which along with the definition of ψ and noting that $\mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1}) = 0$ by virtue of Corollary 1, leads to

$$\psi(\hat{x}^{k+1}) = f(\hat{x}^k) + \langle \nabla f(\hat{x}^k), s^k\rangle + \frac{L}{2}\|s^k\|^2 \geq f(\hat{x}^{k+1}),$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}$. Thus, we can rewrite (A.7) as

$$\psi(y) - f(\hat{x}^{k+1}) \geq \langle \mu, y - \hat{x}^{k+1}\rangle + \frac{1}{2\rho}\|y - \hat{x}^{k+1}\|^2, \qquad (A.8)$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}, \forall \mu \in \partial\psi(\hat{x}^{k+1})$. The subdifferential of ψ evaluated at $\hat{x}^{k+1}$ is given by

$$\partial\psi(\hat{x}^{k+1}) = \nabla f(\hat{x}^k) + \rho^{-1}(\hat{x}^{k+1} - \hat{x}^k) + \partial\mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1}). \qquad (A.9)$$

From Definition 1 we have that for each $k \geq 0$ there exists a vector $v^k \in \mathcal{B}^n_\Omega$ satisfying $\rho\nabla f(\hat{x}^k) = \hat{g}_\rho(\hat{x}^k) + v^k$. Therefore, we can rewrite (A.9) as

$$\partial\psi(\hat{x}^{k+1}) = \rho^{-1}(v^k + \hat{g}_\rho(\hat{x}^k) + \hat{x}^{k+1} - \hat{x}^k) + \partial\mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1}).$$

From Lemma 4 we have $0 \in \rho^{-1}(\hat{g}_\rho(\hat{x}^k) + \hat{x}^{k+1} - \hat{x}^k) + \partial\mathcal{I}_{\mathcal{X}}(\hat{x}^{k+1})$, thus $\rho^{-1}v^k \in \partial\psi(\hat{x}^{k+1})$. This allows us to rewrite (A.8) as

$$\psi(y) - f(\hat{x}^{k+1}) \geq \langle Lv^k, y - \hat{x}^{k+1}\rangle + \frac{1}{2\rho}\|y - \hat{x}^{k+1}\|^2,$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}$, for some $v^k \in \mathcal{B}^n_\Omega$. Undoing the expression of $\psi(y)$ and particularizing to $y = x^* \in \mathcal{X}$ leads to

$$f(\hat{x}^*) - f(\hat{x}^{k+1}) \geq \frac{1}{2\rho}\|\gamma^{k+1}\|^2 - \frac{1}{2\rho}\|\gamma^k\|^2 + f(\hat{x}^*) - f(\hat{x}^k) - \langle \nabla f(\hat{x}^k), \hat{x}^* - \hat{x}^k\rangle + \rho^{-1}\langle v^k, \hat{x}^* - \hat{x}^{k+1}\rangle.$$

Since f is a σ-strongly convex function, we have that (Beck, 2017, Theorem 5.24.*(ii)*)

$$f(\hat{x}^*) - f(\hat{x}^k) - \langle \nabla f(\hat{x}^k), \hat{x}^* - \hat{x}^k \rangle \geq \frac{\sigma}{2}\|\hat{x}^* - \hat{x}^k\|^2.$$

Thus,

$$f(\hat{x}^*) - f(\hat{x}^{k+1}) \geq \frac{1}{2\rho}\|\gamma^{k+1}\|^2 - \frac{\rho^{-1}-\sigma}{2}\|\gamma^k\|^2 - \rho^{-1}\langle v^k, \gamma^{k+1}\rangle. \tag{A.10}$$

By definition of $\hat{x}^*$, we have that $f(\hat{x}^*) - f(\hat{x}^{k+1}) \leq 0$. Therefore, the right hand side of (A.10) must also be less or equal to 0, which leads to

$$\begin{aligned}\frac{1}{2}\|\hat{x}^{k+1} - \hat{x}^*\|^2 &\leq \frac{1-\rho\sigma}{2}\|\hat{x}^k - \hat{x}^*\|^2 + \langle v^k, \hat{x}^{k+1} - \hat{x}^*\rangle \\ &\overset{(*)}{\leq} \frac{1-\rho\sigma}{2}\|\hat{x}^k - \hat{x}^*\|^2 + \|v^k\| \cdot \|\hat{x}^{k+1} - \hat{x}^*\| \\ &\overset{(**)}{\leq} \frac{1-\rho\sigma}{2}\|\hat{x}^k - \hat{x}^*\|^2 + 2\Omega\epsilon^{-1}\|\hat{x}^{k+1} - \hat{x}^*\|^2,\end{aligned}$$

where in $(*)$ we are making use of the Cauchy-Schwarz inequality and in $(**)$ of the fact that $\|v^k\| \leq \Omega$ and $\|\hat{x}^{k+1} - \hat{x}^*\| > \frac{\epsilon}{2}$. Since by construction $\rho\sigma \leq 1$, the assumption $4\Omega\epsilon^{-1} < \rho\sigma$ implies $2\Omega\epsilon^{-1} < \frac{1}{2}$. Thus, we derive

$$\|\hat{x}^{k+1} - \hat{x}^*\|^2 \leq \left(\frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}\right)\|\hat{x}^k - \hat{x}^*\|^2,$$

which leads to claim *(i)* if applied recursively, where we note that the assumption $4\Omega\epsilon^{-1} < \rho\sigma$ guarantees that the sequence is convergent.

We now prove claim *(ii)* by rearranging (A.10) and proceeding as follows:

$$\begin{aligned}f(\hat{x}^{k+1}) - f(\hat{x}^*) &\leq \frac{\rho^{-1}-\sigma}{2}\|\gamma^k\|^2 - \frac{1}{2\rho}\|\gamma^{k+1}\|^2 + \rho^{-1}\langle v^k, \gamma^{k+1}\rangle \\ &\leq \frac{\rho^{-1}-\sigma}{2}\|\gamma^k\|^2 - \frac{1}{2\rho}\|\gamma^{k+1}\|^2 + 2\rho^{-1}\Omega\epsilon^{-1}\|\gamma^{k+1}\|^2 \\ &\leq \frac{\rho^{-1}-\sigma}{2}\|\gamma^k\|^2 + \rho^{-1}\left(2\Omega\epsilon^{-1} - \frac{1}{2}\right)\|\gamma^{k+1}\|^2 \\ &\overset{(*)}{\leq} \frac{1-\rho\sigma}{2\rho}\|\gamma^k\|^2 = \frac{1-4\Omega\epsilon^{-1}}{2\rho}\left(\frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}\right)\|\gamma^k\|^2 \\ &\overset{(**)}{\leq} \frac{1-4\Omega\epsilon^{-1}}{2\rho}\left(\frac{1-\rho\sigma}{1-4\Omega\epsilon^{-1}}\right)^{k+1}\|\hat{x}^0 - \hat{x}^*\|^2,\end{aligned}$$

$\forall \hat{x}^{k+1} \in \mathcal{X}_{(p.q)}$, $\forall k \geq 0$, where $(*)$ holds since $2\Omega\epsilon^{-1} < \frac{1}{2}$ and $(**)$ follows from claim *(i)*. ■

References

Alamo, T., Krupa, P., Limon, D., 2019. Restart FISTA with global linear convergence, in: 2019 18th European Control Conference (ECC), IEEE. pp. 1969–1974.

Beck, A., 2017. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics.

Bemporad, A., 2019. Explicit Model Predictive Control. Springer London.

Bertsekas, D.P., 2009. Convex Optimization Theory. Athena Scientific.

Clarke, E., Kroening, D., Lerda, F., 2004. A tool for checking ANSI-c programs, in: Jensen, K., Podelski, A. (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. Springer Berlin Heidelberg. volume 2988, pp. 168–176. Series Title: Lecture Notes in Computer Science.

Devolder, O., Glineur, F., Nesterov, Y., 2014. First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming 146, 37–75.

Eén, N., Sörensson, N., 2004. An extensible SAT-solver. Lecture notes in computer science 2919, 502–518.

Fang, C., Rutenbar, R., Tsuhan Chen, 2003. Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs, in: International Conference on Computer Aided Design, IEEE. pp. 275–282.

Frison, G., Diehl, M., 2020. HPIPM: a high-performance quadratic programming framework for model predictive control. IFAC-PapersOnLine 53, 6563–6569.

Garone, E., Di Cairano, S., Kolmanovsky, I., 2017. Reference and command governors for systems with constraints: A survey on theory and applications. Automatica 75, 306–328.

Jerez, J.L., Kerrigan, E.C., Constantinides, G.A., 2011. A condensed and sparse QP formulation for predictive control, in: IEEE Conference on Decision and Control and European Control Conference, IEEE. pp. 5217–5222.

Krupa, P., Jaouani, R., Limon, D., Alamo, T., 2021a. A sparse ADMM-based solver for linear MPC subject to terminal quadratic constraint. arXiv:2105.08419 .

Krupa, P., Limon, D., Alamo, T., 2021b. Implementation of model predictive control in programmable logic controllers. IEEE Transactions on Control Systems Technology 29, 1117–1130.

Nadales, J.M., Manzano, J.M., Barriga, A., Limon, D., 2022. Efficient FPGA parallelization of lipschitz interpolation for real-time decision-making. IEEE Transactions on Control Systems Technology 30, 2163–2175.

Nesterov, Y., 2013. Gradient methods for minimizing composite functions. Mathematical Programming 140, 125–161.

Parikh, N., Boyd, S., 2013. Proximal algorithms. Foundations and Trend in Optimization 1, 123–231.

Patrinos, P., Guiggiani, A., Bemporad, A., 2015. A dual gradient-projection algorithm for model predictive control in fixed-point arithmetic. Automatica 55, 226–235.

Rawlings, J.B., Mayne, D.Q., Diehl, M., 2017. Model predictive control: theory, computation, and design. 2nd edition ed., Nob Hill Publishing.

Richter, S., 2012. Computational complexity certification of gradient methods for real-time model predictive control. Ph.D. thesis. ETH Zurich.

Ryu, E.K., Boyd, S., 2016. A primer on monotone operator methods. Appl. comput. math. 15, 3–43.

Saracco, P., Batic, M., Hoff, G., Pia, M.G., 2012. Uncertainty quantification (UQ) in generic MonteCarlo simulations, in: 2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC), IEEE. pp. 651–656.

Simić, S., Bemporad, A., Inverso, O., Tribastone, M., 2022. Tight error analysis in fixed-point arithmetic. Formal Aspects of Computing 34, 1–32.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S., 2020. OSQP: An operator splitting solver for quadratic programs. Mathematical Programming Computation 12, 637–672.

Vakili, S., Langlois, J.M.P., Bois, G., 2013. Finite-precision error modeling using affine arithmetic, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE. pp. 2591–2595.