Abstract—Accelerated first-order methods, also called fast gradient methods, are popular optimization methods in the field of convex optimization. However, they are prone to suffer from oscillatory behavior that slows their convergence when medium to high accuracy is desired. In order to address this, restart schemes have been proposed in the literature, which seek to improve the practical convergence by suppressing the oscillatory behavior. This article presents a restart scheme for accelerated first-order methods, for which we show linear convergence under the satisfaction of a quadratic functional growth condition, thus encompassing a broad class of non-necessarily strongly convex optimization problems. Moreover, the worst-case convergence rate is comparable to the one obtained using an (generally nonimplementable) optimal fixed-rate restart strategy. We compare the proposed algorithm with other restart schemes by applying them to a model predictive control case study.

Index Terms—Accelerated first-order methods (AFOMs), convex optimization, linear convergence, restart schemes.

I. INTRODUCTION

In the field of convex optimization, first-order methods (FOMs) are a widespread class of optimization algorithms that only require evaluations of the objective function and its gradient [1], [2]. Some examples of these methods include gradient descent [1], ISTA [3], and ADMM [4]. A subclass of FOMs are the accelerated FOMs (AFOMs), which are characterized by providing a convergence rate $O(1/k^2)$ in terms of the objective function value [5]. Some noteworthy examples are: Nesterov’s fast gradient method [5], FISTA [3], accelerated ADMM [6], [7], fast Douglas–Rachford splitting [8], and fast AMA (FAMA) [9, Sec. 5].

The use of AFOMs in the field of control is a heavily researched topic, especially in the field of model predictive control (MPC), as evidenced by the following examples: [10], which employs FAMA on a condensed MPC optimization problem; [11] and [12], which consider Nesterov’s fast gradient method; [13], where the infinite horizon constrained LQR problem is solved using an accelerated dual proximal method; [14], which uses the accelerated dual gradient-projection algorithm; [15] and [16], which use the fast gradient method along with the augmented Lagrangian method; and [17], where the FISTA algorithm is employed. AFOMs are also used in other areas closely related to the field of control, such as to find the solution of Lasso problems [18], which are employed, for instance, in sparse system identification [19].

A drawback of AFOMs is that they often suffer from oscillating behavior that slows them down [20]. In order to mitigate this, restart schemes have been proposed in the literature, which have been shown to improve the convergence in a practical setting by suppressing the oscillatory behavior. In a restart scheme, the AFOM is stopped when a certain criterion is met, and then restarted using the last value provided by the algorithm as the new initial condition.

Several restart schemes have been proposed in the literature, where we highlight the following three main aspects to consider about them:

i) If they are generally implementable in practice.

ii) If they guarantee linear convergence.

iii) And, if so, under what assumptions.

O’Donoghue and Candes [20] proposed two simple heuristic schemes (the functional and gradient restarts) that worked well in practice but lack, for the most part, linear convergence guarantees. The results of this article were extended in [21], but the conditions for linear convergence are still restrictive. Necoara et al. [22, Sec. 5.2.2] proposed a restart scheme that extends the scheme from [23, Sec. 5.1], for optimization problems satisfying a quadratic functional growth (QFG) condition [22, Def. 4], which can be viewed as a relaxation of strong convexity. This scheme, which guarantees linear convergence, restarts the AFOM after a fixed number of iterations, resulting in a very simple implementation. However, its drawback is that it requires knowledge of parameters of the optimization problem that are generally hard to obtain, such as the parameter that characterizes the QFG condition. A restart scheme for FISTA that also guarantees linear convergence for optimization problems satisfying the QFG condition is presented in [24]. The scheme requires an initial estimation of the QFG parameter, thus not requiring its exact value. However, we find that it is rather sensitive to this initial guess. Finally, Alamo et al. [25] and [26] proposed two restart schemes with linear convergence for optimization problems satisfying the QFG condition that do not require knowledge of hard to obtain parameters of the optimization problem nor estimations of them, including the QFG parameter. However, both schemes are specific to FISTA.

This article presents a novel restart scheme for AFOMs that exhibits linear convergence for optimization problems satisfying a QFG condition, thus encompassing a broad class of non-necessarily strongly convex problems. Furthermore, it does not require hard-to-attain information of the objective function, such as the QFG parameter. We provide a theoretical upper bound on the number of iterations needed to achieve a desired accuracy, and show that the obtained convergence rate is comparable to the one that could be obtained using the restart scheme from [22, Sec. 5.2.2], which is optimal for the class of AFOMs and optimization problems under consideration.

This article extends the preliminary results presented for FISTA in the conference paper [25] by providing a restart scheme that is
applicable to a broad class of AFOM algorithms and with an improved worst-case convergence rate.

The rest of this article is organized as follows. In Section II, we formally present the class of optimization problem and AFOM algorithms under consideration. Section III describes the optimal fixed-rate restart strategy from [22] and provides its iteration complexity for our class of AFOMs. Section IV presents the novel implementable restart scheme with linear convergence. In Section V, we compare the proposed scheme with some of the abovementioned alternatives referred to by applying them to solve an MPC problem using the FAMA algorithm from [10]. Finally, Section VI concludes this article.

Notation: Given a norm \( || \cdot || \) we denote by \( \| \cdot \| \), its dual norm: \( ||x|| = \sup \{ x^T z : \|z\| \leq 1 \} \). The \( l_1 \)-norm is denoted by \( \| \cdot \|_1 \), \( Z_+ \) denotes the set of non-negative integers, and the set of integer numbers from \( i \) to \( j \) is denoted by \( \mathbb{Z}^i_j \), i.e., \( \mathbb{Z}^i_j = \{ i, i + 1, \ldots, j - 1, j \} \). Euler’s number is denoted by \( e \), the natural logarithm by \( \ln(\cdot) \), \( [x] \) denotes the smallest integer greater than or equal to \( x \), \( \lfloor x \rfloor \) the largest integer smaller than or equal to \( x \). The set of proper closed convex functions from \( \mathbb{R}^n \) to \( (-\infty, \infty] \) is denoted by \( \Gamma_\infty \). Given \( f \in \Gamma_\infty \), we denote by \( \text{dom}(f) \) its effective domain, that is, \( \text{dom}(f) = \{ x \in \mathbb{R}^n : f(x) < \infty \} \). Further notation is given in Notations 1 and 2 in the following section.

II. Problem Statement

In this article, we are concerned with finding the solution of optimization problems given by

\[
 f^* = \min_{x \in \mathbb{R}^n} f(x) \tag{1}
\]

which we assume are solvable, and where \( f \in \Gamma_\infty \).

We use the following notation for the optimal set of \( (1) \), the projection operation onto it, and the level sets of \( f \).

Notation 1: Given the solvable problem \( (1) \), the following hold.

i) The optimal set is denoted by \( \Omega_f \). That is,

\[
 \Omega_f = \{ x \in \mathbb{R}^n : f(x) = f^* \}.
\]

ii) For every \( x \in \mathbb{R}^n \), we denote \( \bar{x} \) the closest element to \( x \) in the optimal set \( \Omega_f \) with respect to norm \( \| \cdot \| \), i.e.,

\[
 \bar{x} = \arg \min_{\Omega_f} \| x - z \|.
\]

iii) Given \( \rho \in [0, \infty) \), we denote the level set

\[
 \mathcal{V}_f(\rho) = \{ x \in \mathbb{R}^n : f(x) - f^* \leq \rho \}.
\]

It is well-known that if \( f \in \Gamma_\infty \) is strongly convex, then there exists \( \mu > 0 \), such that

\[
 f(x) - f^* \geq \frac{\mu}{2} \| x - \bar{x} \|^2 \quad \forall x \in \text{dom}(f).
\]

This inequality is called the quadratic functional growth (QFG) condition and it is satisfied, at least locally, for a large class of not necessarily strongly convex functions [22], [27], [28].

Let us consider a fixed-point algorithm \( A \) that can be applied to solve \( (1) \), i.e., given an initial point \( x_0 \in \text{dom}(f) \), algorithm \( A \) generates a sequence \( \{ x_k \} \) with \( k \geq 0 \), such that \( \lim_{k \to \infty} f(x_k) = f^* \). We use the following notation to refer to iterates provided by the algorithm \( A \).

Notation 2: Suppose that the fixed-point algorithm \( A \) is applied to solve problem \( (1) \) using as initial condition \( x_0 \). Given the integer \( k \geq 1 \), we denote with \( A(x_0, k) \) the vector in \( \mathbb{R}^n \) corresponding to iteration \( k \) of the algorithm.

The following assumption characterizes the class of optimization problems and AFOM algorithms we consider.

Assumption 1: We assume the following.

(i) For every \( \rho > 0, f \in \Gamma_\infty \), satisfies the QFG condition

\[
 f(x_0) - f^* \geq \frac{\mu_\rho}{2} \| x_0 - \bar{x}_0 \|^2 \quad \forall x_0 \in \mathcal{V}_f(\rho)
\]

for some \( \mu_\rho > 0 \).

(ii) For every \( x_0 \in \text{dom}(f) \), algorithm \( A \) satisfies

\[
 f(A(x_0, 1)) \leq f(x_0) - \frac{1}{2L_f} \| g(x_0) \|^2 \tag{2}
\]

\[
 f(A(x_0, k)) - f^* \leq \frac{\alpha_f}{(k + 1)^2} \| x_0 - \bar{x}_0 \|^2 \quad \forall k \geq 1 \tag{3}
\]

where \( \alpha_f > 0, L_f > 0 \), and \( g(\cdot) \) is a gradient operator satisfying \( g(x) = 0 \Leftrightarrow x \in \Omega_f \).

(iii) We denote \( n_\rho = \max \{ \frac{1}{\sqrt{\rho}}, \sqrt[2\rho]{\frac{2\alpha_f}{\mu_\rho}} \} \).

The conditions listed in Assumption 1 are expressed in very general terms to be able to account for a broad class of optimization problems and AFOMs. Let us start by providing some insight and examples of the conditions and terms listed in the assumption. Assumption 1.(i) establishes a local QFG condition on the objective function, which encom- pas a broad class of non-necessarily strongly convex functions (see [22, Fig. 1]). We refer the reader to [22], [27], and [28] for examples of functions satisfying this condition, including the case \( f(x) = h(Ex) + c^T x + 1(x) \), where \( h : \mathbb{R}^m \to \mathbb{R} \) is a smooth strictly convex function, \( E \in \mathbb{R}^{m \times n} \) and \( 1(x) \) is the indicator function of a polyhedral set \( \mathcal{X} \), which encompasses a large family of optimization problems. Assumption 1.(ii) is written in terms of the constants \( L_f \) and \( \alpha_f \), which will depend both on the AFOM \( A \) being used and on the structure of \( f \), and in terms of an operator \( g : \mathbb{R}^n \to \mathbb{R}^n \), which plays the role of the gradient operator of \( A \). For instance, in the FISTA algorithm applied to \( f = h(Ex) + c^T x + 1(x) \), where \( h : \mathbb{R}^m \to \mathbb{R} \) is a smooth strictly convex function, \( E \in \mathbb{R}^{m \times n} \) and \( 1(x) \) is the indicator function of a polyhedral set \( \mathcal{X} \), which encompasses a large family of optimization problems. Assumption 1.(iii) is satisfied by most AFOMs [1], [2], including the ones we list in the introduction, although under varying assumptions. Condition (2) is also satisfied by most AFOMs because the first iteration is often the result of the application of a proximal (or composite) gradient mapping operator. If this is not the case, then it can be easily enforced by taking this operator as the first step of the algorithm.

In conclusion, Assumption 1 is satisfied by a broad family of AFOMs and optimization problems of interest in the field of control, including the AFOMs listed in the introduction and optimization problems such as QPs, Lasso, or those that arise from many MPC formulations.

We now present a property on the iterates of \( A \) that serves as the basis for the development and convergence analysis of the optimization schemes of the following sections. An equivalent result can be found in [22, Sec. 5.2.2].

Property 1: Suppose that Assumption 1 holds. Then, for every \( x_0 \in \mathcal{V}_f(\rho) \)

\[
 f(A(x_0, k)) - f^* \leq \left( \frac{n_\rho}{k + 1} \right)^2 (f(x_0) - f^*) \quad \forall k \geq 1. \tag{4}
\]

Proof: Denote \( f_0 = f(x_0) \) and \( f_k = f(A(x_0, k)) \) \( k \geq 1 \). Then,

\[
 f_k - f^* \leq \frac{\alpha_f}{(k + 1)^2} \| x_0 - \bar{x}_0 \|^2 \leq \frac{2a_f}{\mu_\rho(k + 1)^2} (f_0 - f^*) \leq \frac{a_f^2}{(k + 1)^2} (f_0 - f^*). \tag{5}
\]
III. OPTIMAL FIXED-RATE RESTART SCHEME

This section describes the optimal fixed restart scheme presented in [22, Sec. 5.2.2], in which $A$ is restarted each time the iteration counter attains an optimal fixed number of iterations. We analyze, under Assumption 1, its iteration complexity.

A fixed-rate restart scheme takes the recursion

$$v_{j+1} = A(v_j, n), \quad j \geq 0$$

where $n \geq 1$ is a fixed integer, starting at a given $v_0 \in V_f(\rho)$.

Under Assumption 1, the sequence $\{f(v_j)\}_{j \geq 0}$ is nonincreasing and converges monotonically to $f^*$ if $n \geq \n_*,$ (see Property 1). Given an accuracy parameter $\epsilon > 0,$ the following property states the number $M$ of restarts required to satisfy $f(v_{M-1}) - f(v_M) \leq \epsilon$, and shows that the bound on the total number of iterations of $A$ is minimized if $n$ is chosen equal to $\lceil \epsilon \rho \rceil.$ See also [22, Sec. 5.2.2] for a similar result.

Property 2 (Optimal Fixed-Rate Restart Scheme): Let Assumption 1 hold. Given $v_0 \in V_f(\rho)$ and an integer $n$ satisfying $n \geq n_\rho,$ consider the recursion (5). Then, given $\epsilon > 0$:

(i) The inequality $f(v_{M-1}) - f(v_M) \leq \epsilon$ is satisfied for every $M \geq M_\epsilon,$ where

$$M_\epsilon = 1 + \frac{1}{2 \ln (1 + \frac{f(v_0) - f^*}{\epsilon})}.$$ \hspace{1cm} (6)

(ii) If $n = \lceil \epsilon \rho \rceil,$ the total number of iterations of $A$ required to attain $f(v_{j-1}) - f(v_j) \leq \epsilon$ is upper-bounded by

$$N_\epsilon^f = \lceil \epsilon \rho \rceil \left[ 1 + \frac{1}{2} \ln \left( 1 + \frac{f(v_0) - f^*}{\epsilon} \right) \right].$$ \hspace{1cm} (7)

In this case, we call recursion (5) the optimal fixed-rate restart scheme.

Proof: See Appendix A.

One of the key properties of the optimal fixed-rate restart scheme is that it recovers the linear optimal convergence rate provided by Nesterov’s fast gradient method for strongly convex functions [22], [29, Sec. 2.2]. That is, recalling that $n_\rho = \max\{1/2, \sqrt{2a_f/\mu_f}\}$ we easily obtain from Property 2(ii) that an $\epsilon$ accurate solution is obtained in

$$O\left( n_\rho \ln \left( \frac{f(v_0) - f^*}{\epsilon} \right) \right)$$

iterations. However, we note that this scheme is often nonimplementable because the value of $n_\rho$ is generally not available. Nevertheless, (7) and (8) are of interest because they provide the best theoretical convergence rate that can be obtained with a fixed-rate restart strategy.

IV. PROPOSED RESTART SCHEME

In this section, we propose a novel restart scheme that does not require knowledge of $n_\rho$ and that attains a convergence rate similar to the one of the optimal fixed-rate restart strategy described in Section III.

We start by presenting Algorithm 1, which implements a performance-based exit condition on algorithm $A$. Algorithm 1 will then be used to derive the main result of this article: Algorithm 2.

Given an initial condition $x_0$ and a scalar $m,$ which serves as a lower bound on the number of iterations, Algorithm 1 generates a sequence $\{x_k\}_{k \geq 2}$ that satisfies (see Step 5)

$$f(x_k) = \min\{f(x_{k-1}), f(A(x_0, k))\} \quad \forall k \geq 1.$$ Therefore,

$$f(x_k) = \min_{i=0, \ldots, k} f(A(x_0, i)).$$ \hspace{1cm} (9)

Algorithm 1: Performance-Based Exit Condition On $A.$

Prototype: $[z, m] = A_d(r, n)$

Require: $r \in \mathrm{dom}(f), \quad n \in \mathbb{R}$

1. $x_0 \leftarrow r, \quad k \leftarrow 0$

2. Initialize $A$ with $x_0$

3. repeat

4. $k \leftarrow k + 1$

5. $x_k \leftarrow \begin{cases} A(x_0, k) & \text{if } f(A(x_0, k)) \leq f(x_{k-1}) \\ x_{k-1} & \text{otherwise} \end{cases}$

6. $\ell \leftarrow \left\lfloor \frac{\epsilon}{2} \right\rfloor$

7. until $k \geq n$ and $f(x_{\ell}) - f(x_k) \leq \frac{1}{3} (f(x_0) - f(x_\ell))$

Output: $z \leftarrow x_k, \quad m \leftarrow k$

Algorithm 2: Optimal Algorithm Based On $A_d.$

Prototype: $[z_{\text{out}}, j_{\text{out}}] = A_d(z_0)$

Require: $z_0 \in \mathrm{dom}(f), \quad \epsilon > 0$

1. $m_0 \leftarrow 1, \quad m_{\text{-}1} \leftarrow 1, \quad j \leftarrow -1$

2. repeat

3. $j \leftarrow j + 1$

4. $s_j \leftarrow \begin{cases} f(x_{j-1}) - f(x_j) & \text{if } j \geq 2 \\ 0 & \text{otherwise} \end{cases}$

5. $n_j \leftarrow \max\{m_j, 4s_j, m_{\text{-}j+1}\}$

6. $[j+1, m_{j+1}] \leftarrow A_d(z_j, n_j)$

7. until $f(x_j) - f(x_{j+1}) \leq \epsilon$

Output: $z_{\text{out}} \leftarrow z_{j+1}, \quad j_{\text{out}} \leftarrow j$

The algorithm exits after $k \geq n$ iterations if the following inequality is satisfied (see Step 7):

$$f(x_{\ell}) - f(x_k) \leq \frac{1}{3} (f(x_0) - f(x_{\ell}))$$ \hspace{1cm} (10)

where $\ell = \left\lfloor \frac{\epsilon}{2} \right\rfloor.$ The outputs of the algorithm are $z \in \mathbb{R}^n$ and $m \in \mathbb{Z},$ where $z = x_n,$ and $m \geq n$ is the number of iterations required to satisfy the exit condition (10).

Intuitively, exit condition (10) detects a degradation in the performance of the iterations of $A.$ Notice that at iteration $m,$ the reduction corresponding to the last half of the iterations (from $\frac{m}{2}$ to $m$) is no larger than one third of the reduction achieved in the first half of the iterations (from 0 to $\frac{m}{2}$). The constant 1/3 was chosen because it minimized the upper bound on the maximum number of iterations that we show in Theorem 1(iii).

The following property characterizes the number of iterations required to attain the exit condition (10) of Algorithm 1. This result is instrumental to prove the convergence results of Algorithm 2.

Property 3: Suppose that Assumption 1 holds. Then, the output $[z, m]$ from the call $[z, m] = A_d(r, n)$ of Algorithm 1 satisfies, for every $r \in V_f(\rho)$:

(i) $f(z) \leq f(r) - \frac{1}{2\ell} \|g(r)\|^2.$

(ii) $f(z) - f^* \leq \left( \frac{n_{\rho}}{m+1} \right)^2 (f(r) - f^*).$

(iii) $n \in \left( 0, \left[ \frac{m}{4n_\rho} \right] \right) \Rightarrow m \in \left[ n, \left[ 4n_\rho \right] \right].$

Proof: See Appendix B.
We now introduce the main contribution of this article: Algorithm 2. This algorithm makes successive calls to Algorithm 1 (see Step 6) using a minimum number of iterations $n_j$ that is determined by the past evolution of the iterates $z_j$ (see Steps 4 and 5). The main properties of the iterates of Algorithm 2 are given in the following property and theorem.

**Property 4:** Suppose that Assumption 1 holds, and consider Algorithm 2 for a given initial condition $z_0 \in V_f (\rho)$ and accuracy parameter $\epsilon > 0$. Then, the following hold.

(i) Property 3 can be applied to the iterates of Algorithm 2 (i.e., taking $r \equiv z_j, n \equiv n_j, z \equiv z_{j+1}$, and $m \equiv m_{j+1}$).

(ii) The sequence $\{m_j\}$ produced is nondecreasing. In particular

$$m_j \leq n_j \leq m_{j+1} \quad \forall j \in Z^{out}_{m}.$$  \hspace{1cm} (11)

(iii) The sequence $\{s_j\}$ satisfies $s_j \in (0, 1] \quad \forall j \in Z^{out}_{m}$.  

**Proof:** See Appendix C.

**Theorem 1:** Suppose that Assumption 1 holds and consider Algorithm 2 for a given initial condition $z_0 \in V_f (\rho)$ and accuracy parameter $\epsilon > 0$. Then, the following hold.

(i) The number of calls to $A_d$ (Step 5) is bounded. That is, $j_{out}$ is finite.

(ii) The number of iterations of $A$ at each call of $A_d$ (Step 6) is upper-bounded by $[4n_{\rho}]$. That is,

$$m_{j+1} \leq [4n_{\rho}] \quad \forall j \in Z^{out}_{m}.$$

(iii) The total number of iterations of $A$ performed by a call to Algorithm 2, which we denote by $N_A$, is upper-bounded by $N_A \leq N_{A_d}$, where

$$N_A = \frac{\epsilon}{2} \left[ 5 + \frac{1}{\ln 15} \ln \left( 1 + \frac{f(z_0) - f^*}{\epsilon} \right) \right].$$

**Proof:** See Appendix C.

**Remark 1:** From Property 4(i), we have that we can rearrange Property 3(i) to read as

$$\|g(z_j)\|^2 \leq 2L_f (f(z_j) - f(z_{j+1})).$$

Therefore, the exit condition $f(z_j) - f(z_{j+1}) \leq \epsilon$ implies $\|g(z_j)\|^2 \leq 2L_f \epsilon$. Since, as per Assumption 1.(iii), $g(z_j)$ serves to characterize the optimality of $z_j$, we conclude that the exit condition of Algorithm 2 also serves to characterize the optimality of $z_{j+1}$. That is, the exit condition could be replaced by $\|g(z_j)\|, \leq \epsilon$, where $\epsilon > 0$. In this case, the upper bound on the number of iterations given in Theorem 1.(iii) would be the same but replacing $\epsilon$ with $\epsilon/(2L_f)$.

Note that Theorem 1.(iii) shows that the proposed algorithm attains the optimal linear convergence rate of the optimal fixed-rate restart scheme, in the sense that an $\epsilon$ accurate solution is obtained in (8) iterations. This is an important result, since we recover the optimal convergence rate for our class of optimization problems using AFOMs.

Comparing the upper bound provided in Theorem 1.(iii) with the upper bound $\bar{N}_D$ in (7) of the optimal fixed-rate restart scheme presented in Section III, we have

$$\bar{N}_D = \frac{e}{\epsilon} \left[ 5 + \frac{1}{\ln 15} \ln \left( 1 + \frac{f(z_0) - f^*}{\epsilon} \right) \right].$$

from where we obtain that

$$\lim_{\epsilon \rightarrow 0} \frac{\bar{N}_D}{N_F} = \frac{e}{\epsilon} \left[ \frac{4n_{\rho}}{\ln 15} \right] \leq \frac{c(4n_{\rho} + 1)}{en_{\rho} \ln 15} \leq \frac{1}{n_{\rho} \ln 15} \leq \frac{3}{2} \left( 1 + \frac{1}{4n_{\rho}} \right).$$

**V. NUMERICAL RESULTS**

We compare Algorithm 2 with other restart schemes of the literature by applying them to the FAMA algorithm for MPC from [10], where we consider the MPC formulation [30, (2)] but without its terminal constraint (2f). This MPC formulation can be posed as [10, Problem 2.1], which can therefore be solved using the FAMA algorithm [10, Algorithm 1]. As stated in [10, Sec. 3.2], this algorithm is equivalent to applying FISTA to the dual problem of [10, Problem 2.1]. Therefore, the dual-objective function value $D(\lambda)$ (see [10, Sec. 3.2]) and its gradient $\nabla D(\lambda)$, where $\lambda$ are the dual variables, satisfy Assumption 1, with $\lambda_k = \lambda_k$, $D \equiv f, \nabla D \equiv g, L_f$ is equal to the expression $\rho(C)/\sigma_f$ described in [10, Sec. 2] and $n_f = 2L_f$.

We consider the system described in [30, Sec. 3], which consists of three objects connected by springs where external forces can be applied to the two outer-most objects. We use the exact same setup as in [30, Sec. 3] with the exception of the cost function matrix $T$, which we take as the solution of the discrete algebraic Riccati equation, as is often the case in MPC, and the state constraints, which we do not enforce. We perform the preconditioning procedure described in [10, Sec. 4] on the resulting MPC optimization problem.

We solve the MPC’s optimization problem for 1000 randomly generated system states, where the positions of each object is obtained from a uniform distribution on the interval $[-0.5, 0.5]$ m/s. We solve each problem using the FAMA algorithm [10, Algorithm 1] with the following restart schemes.

i) **Functional:** The heuristic restart scheme proposed in [20] that uses restart condition $D(\lambda_{k+1}) \leq D(\lambda_{k})$.

ii) **Gradient:** The heuristic restart scheme proposed in [20] that uses restart condition $\|\nabla D(\lambda_{k})\|, \lambda_{k+1} \geq 0$.

iii) **Optimal:** The restart scheme proposed in [22, Sec. 5.2.2], which is typically nonimplementable. In order to include it for comparison with the proposed scheme, we implement it using the method described at the end of the referenced subsection that requires knowing $D(\lambda^*)$, which we compute using FAMA with a very small exit tolerance.

iv) **GLCR:** The restart FISTA algorithm with linear convergence proposed in [25, Algorithm 2].

v) **GRBF:** The gradient-based restart FISTA algorithm proposed in [26, Algorithm 2].

vi) **Adaptive:** The adaptive scheme for FISTA from [24]. We take the initial guess of the QP parameter as $10^{-5}$.

We also use the proposed restart scheme (Algorithm 2) and the nonrestarted FAMA.

Table I gives the average, median, maximum, and minimum number of iterations of each scheme for the 1000 tests. The results of the table are obtained by terminating FAMA when the iterate $u_k$ of the primal problem (see [10, Algorithm 1]) satisfies $\|u_k - u^*\|_2/\|u^*\|_2 \leq 10^{-5}$, where $\| \cdot \|_2$ stands for the standard Euclidean norm and the optimal solution $u^*$ of the primal problem is obtained using the quadprog solver from MATLAB. We use this exit condition to provide a fair comparison between the different approaches. Fig. 1 shows the evolution of $\|u_k - u^*\|_2/\|u^*\|_2$ for each restart scheme when taking the system state as the origin. The results indicate that the restart schemes tend to reduce the number of iterations required to obtain a solution of
the MPC’s optimization problem when compared with the nonrestarted variant, although this is not always the case.

Additional numerical results can be found in [31, Sec. 3.3], where the abovementioned restart schemes are applied to solve random QP and Lasso problems with varying condition numbers.

VI. CONCLUSION

We propose a restart scheme applicable to a broad class of AFOMs that does not require knowledge of hard-to-obtain parameters of the optimization problem and still retains a linear convergence rate similar to the optimal one for optimization problems satisfying a QFG condition, i.e., its worst case complexity is similar to the best one that can be obtained if the parameters characterizing the convergence of the AFOM were known. The scheme is based on a performance-based exit condition that detects the degradation of the convergence of the AFOM, and which we show is attained in a finite number of iterations. The numerical results indicate that the proposed algorithm is comparable, in practical terms, with other restart schemes of the literature. In our numerical results, we find that the application of restart schemes may be desirable in many situations, since the nonrestarted variant can require an excessive number of iterations.

APPENDIX

A. Proof of Property 2

Suppose that the integer $M$ is such that the inequality $f(v_{M-1}) - f(v_M) > \epsilon$ is satisfied. From Property 1, we have

$$f(v_{j+1}) - f^* \leq \left(\frac{n_p}{\ell}\right)^2 (f(v_j) - f^*) \quad \forall j \geq 0.$$ 

Using this inequality in a recursive manner, we obtain

$$\epsilon < f(v_{M-1}) - f(v_M) \leq f(v_{M-1}) - f^*$$

This leads to

$$M < 1 + \frac{1}{2(\ln n - \ln n_p)} \ln \left(\frac{f(v_0) - f^*}{\epsilon}\right) < M. \quad (13)$$

Thus, we conclude that if $M$ does not satisfy (13), then $f(v_{M-1}) - f(v_M) \leq \epsilon$. This proves the first claim.

Given $\epsilon > 0$, $v_0 \in \{\ell\}_{\ell=1}^{2\ell}$, and $n > n_p$, denote $S \geq 0$ the smallest number of restarts required to satisfy the condition $f(v_{S-1}) - f(v_S) \leq \epsilon$. We infer from the first claim of the property that

$$S \leq 1 + \frac{1}{2(\ln n - \ln n_p)} \ln \left(1 + \frac{f(v_0) - f^*}{\epsilon}\right). \quad (14)$$

Simple calculus yields that the value that minimizes $\frac{n}{\ln n - \ln n_p}$ is $n^* = en_p$. Since $n$ has to be a positive integer, we choose the fixed restart rate given by $n = \lceil en_p \rceil$. Introducing this value in the bound (14), we finally obtain

$$N_F(e n_p) \leq \lceil e n_p \rceil \left[1 + \frac{1}{2} \ln \left(1 + \frac{f(v_0) - f^*}{\epsilon}\right)\right].$$

B. Proof of Property 3

From (9) and Assumption 1, we have

$$f(z) = f(x_m) = \min_{i=0,\ldots,m} f(A(x_0, i))$$

$$\leq f(A(x_0, 1)) = \frac{2}{2L_f} \lVert g(x_0) \rVert^2.$$ 

The first claim now follows directly from $x_m = r$. In view of Property 1, we have, for every $k \in \mathbb{Z}_{\ell}^+$

$$f(x_k) - f^* \leq f(A(x_0, k)) - f^* \leq \left(\frac{n_p}{\ell + 1}\right)^2 (f(x_0) - f^*). \quad (15)$$

The second claim now follows from $x_m = z$ and $x_0 = r$. The inequality $n \leq m$ is trivially satisfied from Step 7. Thus, in order to conclude the proof, we show that inequality (10) is satisfied for

$$k = \lceil \frac{4n_p}{n}\rceil$$

and $\ell = \lceil \frac{2n_p}{n}\rceil \geq \lceil 2n_p \rceil \geq 1$ (where this last inequality follows from Assumption 1.(iii), which states that $n_p \geq 1/2$).

$$f(x_0) - f^* \leq \left(\frac{n_p}{\ell + 1}\right)^2 (f(x_0) - f^*) \leq \left(\frac{n_p}{2n_p}\right)^2 (f(x_0) - f^*) = \frac{1}{4} (f(x_0) - f^*).$$

This implies $f(x_j) \leq \frac{1}{4} f(x_0) + \frac{1}{4} f^* \leq \frac{1}{4} f(x_0) + \frac{1}{4} f(x_j)$. Thus, $f(x_j) - f(x_k) \leq \frac{1}{4} (f(x_0) - f(x_k))$.

$$f(x_0) - f(x_k) = \frac{1}{4} (f(x_0) - f(x_j)) + \frac{1}{4} (f(x_j) - f(x_k)).$$

We conclude that $f(x_j) - f(x_k) \leq \frac{1}{4} (f(x_0) - f(x_j))$. 

\[\hfill\]
C. Proofs for Algorithm 2

Proof of Property 4: Since \( z_0 \in \text{dom}(f) \), we have that \( z_0 \in V_f(\rho) \) for some \( \rho > 0 \). Additionally, each \( z_j \) is obtained from a call to Algorithm 1 (Step 6). As such, in view of Property 3.(i), we have that the iterates \( z_j \) satisfy \( z_j \in V_f(\rho) \) \( \forall j \in \mathbb{Z}^{\text{iter}}_0 \). Therefore, Property 3 can be applied to each call to \( A_\rho \), thus proving Property 4.(i). That is, for every \( j \geq 0 \), the iterates of Algorithm 2 satisfy

\[
\begin{align}
  f(z_{j+1}) &\leq f(z_j) - \frac{1}{2\ell_f} ||g(z_j)||^2, \quad (16a) \\
  f(z_{j+1}) - f^* &\leq \left( \frac{n_p}{m_{j+1}+1} \right)^2 (f(z_j) - f^*) \quad (16b) \\
  n_j &\in \{0, \lfloor 4n_p \rfloor \} \Rightarrow m_{j+1} \in \{n_j, \lceil 4n_p \rceil \}. \quad (16c)
\end{align}
\]

Next, due to Step 5 we have \( m_j \leq n_j, \ j \in \mathbb{Z}^{\text{iter}}_0 \). Moreover, from (16c), we have that \( n_j \leq m_{j+1} \) \( \forall j \in \mathbb{Z}^{\text{iter}}_0 \), which proves Property 4.(ii).

Finally, we prove Property 4.(iii). From the exit condition (Step 7), we have

\[
\begin{align}
  f(z_{j+1}) - f(z_j) &> \epsilon \quad \forall j \in \mathbb{Z}^{\text{iter}}_0. \quad (17)
\end{align}
\]

Additionally, from (16a), we have \( f(z_{j+1}) \geq f(z_j) \quad \forall j \in \mathbb{Z}^{\text{iter}}_0 \).

Thus,

\[
\begin{align}
  f(z_{j+1}) - f(z_j) &\geq f(z_{j+1}) - f(z_j) > \epsilon > 0 \quad \forall j \in \mathbb{Z}^{\text{iter}}_0.
\end{align}
\]

Therefore, from Step 4, taking \( j \geq 2 \), we have

\[
0 < s_j = \sqrt{\frac{f(z_{j+1}) - f(z_j)}{f(z_{j+1}) - f(z_j)}} \leq 1 \quad \forall j \in \mathbb{Z}^{\text{iter}}_0.
\]

The proof of the following lemma relies upon some technical results on the iterates of Algorithm 2, namely Lemmas 2 and 3, which we include in Appendix D.

Lemma 1: Consider Algorithm 2 with the initial condition \( z_0 \in V_f(\rho) \), and \( \epsilon > 0 \). Suppose that Assumption 1 is satisfied and that \( j_{\text{out}} \geq D \), where

\[
D = \left[ 5 + \frac{1}{\ln 15} \ln \left( 1 + \frac{f(z_0) - f^*}{\epsilon} \right) \right].
\]

Then, \( m_{t+1} \leq \sqrt{\frac{m_{t+1}}{D}} \quad \forall t \in \mathbb{Z}_0^{\text{iter}} \).

Proof: The proof is obtained by reducing to absurdity. If there is \( t \in \mathbb{Z}_0^{\text{iter}} \) such that \( m_{t+1} > \sqrt{\frac{m_{t+1}}{D}} \), then we obtain from Lemma 3.(iv) (see Appendix D) that

\[
D < 5 + \frac{1}{\ln 15} \ln \left( 1 + \frac{f(z_0) - f^*}{\epsilon} \right)
\]

which contradicts the definition of \( D \).

Proof of Theorem 1: Let \( T \in \mathbb{Z} \) be such that

\[
f(z_j) - f(z_{j+1}) > \epsilon \quad \forall j \in \mathbb{Z}_0^T
\]

is satisfied. Then, defining \( d_j = f(z_j) - f(z_{j+1}) \), we have

\[
f(z_0) - f(z_{T+1}) = \sum_{j=0}^{T} d_j \geq (T + 1) \left( \min_{j=0, \ldots, T} d_j \right) > (T + 1) \epsilon.
\]

Thus, \( T + 1 < \frac{f(z_0) - f(z_{T+1})}{f(z_0) - f^*} \leq \frac{\epsilon}{\epsilon} \), from where we infer that the largest integer \( T \) satisfying (18) is bounded. Consequently, the exit condition of Algorithm 2 (see Step 7) is satisfied within a finite number of iterations, thus proving claim (i).

To prove Theorem 1.(ii), we start by noting that both \( m_1 \) and \( m_2 \) are no larger than \( [4n_p] \). Indeed, from Step 4, we have that \( s_0 = s_1 = 0 \), which, in virtue of Step 5, implies that \( n_0 = m_0 = 1 \) and \( n_1 = m_1 \). Since \( n_0 = 1 \) is no larger than \( [4n_p] \), we have from (16c) that \( m_1 \) is also upper-bounded by \( [4n_p] \). Moreover, since \( n_1 = m_1 \leq [4n_p] \), we obtain by the same reasoning that \( m_2 \leq [4n_p] \). We now prove that if \( j \geq 2 \) and \( m_j \leq [4n_p] \), then \( m_{j+1} \leq [4n_p] \). From Step 4, we have

\[
s_j^2 = \frac{f(z_{j+1}) - f(z_j)}{f(z_{j+1}) - f(z_j)} = 1 - \frac{f(z_j) - f(z_{j+1})}{f(z_{j+1}) - f(z_j)}
\]

\[
\leq 1 - \frac{f(z_j) - f(z_{j+1})}{f(z_{j+1}) - f^*} \leq \left( \frac{n_p}{m_{j+1}+1} \right)^2.
\]

Thus, we have \( s_j m_{j+1} \leq n_p \). Therefore,

\[
n_j = \max\{m_j, 4s_j m_{j+1}\} \leq \max\{[4n_p], 4n_p\} = [4n_p]
\]

which, along with (16c), leads to \( m_{j+1} \leq [4n_p] \), thus proving the claim.

Finally, to prove Theorem 1.(iii), we start by noting that the computation of each \( z_{j+1} \) is obtained from \( m_{j+1} \) iterations of \( A \). Thus,

\[
N_A = \sum_{j=0}^{\text{iter}} m_{j+1} = \left( 1 + j_{\text{out}} \right) [4n_p].
\]

Let us denote

\[
D = \left[ 5 + \frac{1}{\ln 15} \ln \left( 1 + \frac{f(z_0) - f^*}{\epsilon} \right) \right].
\]

Consider first the case \( j_{\text{out}} < D \). Since both \( j_{\text{out}} \) and \( D \) are integers, we infer from this inequality that \( 1 + j_{\text{out}} \leq D \). This, along with (19), implies that \( N_A \leq [4n_p] \leq N_A \).

Suppose now that \( j_{\text{out}} \geq D \). We first recall that Property 4.(ii) states that the sequence \( \{m_{j+1}\}_{j=0}^{\text{iter}} \) is nondecreasing. We now rewrite \( j_{\text{out}} \) as \( j_{\text{out}} = d + tD \), where \( d \in \mathbb{Z}_0^{d-1} \) and \( t \) is a non-negative integer. Thus,

\[
N_A = \sum_{j=0}^{j_{\text{out}}} m_{j+1} = \sum_{j=0}^{d} m_{j+1} + \sum_{j=1}^{t D} m_{j+1} + d \leq D \sum_{j=0}^{t D} m_{j+1} + D \sum_{j=0}^{t D} m_{j+1} + D \sum_{j=0}^{t D} m_{j+1}.
\]

From Lemma 1, we have

\[
m_{j+1} \leq \frac{m_{d+1} + (d+1)D}{\sqrt{15}} \quad \forall i \in \mathbb{Z}_0^{d-1}.
\]

Thus,

\[
N_A \leq D \sum_{i=0}^{t D} m_{j+1} + D \sum_{j=0}^{t D} m_{j+1} + D \sum_{j=0}^{t D} m_{j+1}.
\]

Using now \( m_{j+1} \leq \tilde{m} = [4n_p] \) [see (12)], we obtain

\[
\frac{N_A}{D m_{j_{\text{out}}}} \leq \sum_{i=0}^{t D} \left( \frac{1}{\sqrt{15}} \right)^{j_{\text{out}}} = \sum_{j=0}^{t D} \left( \frac{1}{\sqrt{15}} \right)^{j} \leq \sum_{j=0}^{t D} \left( \frac{1}{\sqrt{15}} \right)^{j} \leq \frac{\sqrt{15}}{\sqrt{15} - 1} \leq \frac{\epsilon}{2}.
\]

Thus, \( N_A \leq \frac{\epsilon}{2} m_D \leq \frac{\epsilon}{2} [4n_p] D \).


\[ \forall \in \mathbb{R} \rightarrow \mathbb{R} \text{, defined as} \]

\[ \varphi(s) = \left(\frac{1}{s^2} - 1\right) \cdot \max \{1, (4s)^4\} \]

satisfies \( \varphi(s) \geq 15 \) \( \text{for all } s \in (0, \frac{\sqrt{5}}{2}] \).

\text{Proof:} \text{ We have that} \]

\[ \varphi(s) = \begin{cases} 4^4(s^2 - s^4) & \text{if } s > \frac{1}{4} \\ \frac{1}{s^2} - 1 & \text{if } s \leq \frac{1}{4} \end{cases} \]

It is clear that \( \varphi(s) \) is monotonically decreasing in \((0, \frac{1}{4}]\). Thus,

\[ \min_{s \in (0, \frac{\sqrt{5}}{2}]} \varphi(s) = \min_{s \in (\frac{1}{4}, \frac{\sqrt{5}}{2}]} \varphi(s) = \min_{s \in (\frac{1}{4}, \frac{\sqrt{5}}{2}]} 4^4(s^2 - s^4) \]

We notice that the derivative of \( s^2 - s^4 \) is \( 2s(1 - 2s^2) \), which vanishes only once in the interval of interest (at \( s = \frac{1}{\sqrt{2}} \)). From here, we infer that \( s^2 - s^4 \) is increasing in \( (\frac{1}{4}, \frac{1}{\sqrt{2}}) \) and decreasing in \((\frac{1}{4}, \frac{\sqrt{5}}{2}]\). Thus, the minimum is attained at the extremes of the interval \((\frac{1}{4}, \frac{\sqrt{5}}{2}]\). That is, we conclude that

\[ \min_{s \in (0, \frac{\sqrt{5}}{2}]} \varphi(s) = \min \left\{ \varphi \left(\frac{1}{4}\right), \varphi \left(\frac{\sqrt{5}}{4}\right) \right\} = \min\{15, 15\} = 15. \]

\text{Lemma 3 (Technical Results on the Iterates of Algorithm 2): Consider Algorithm 2 with the initial condition} \( s_0 \in \mathbb{R}^j \) \text{and} \( \epsilon > 0 \). \text{Suppose that} \( \Delta \in \mathbb{R}^j \) \text{is satisfied and that} \( j \geq 2 \). \text{Suppose also that} \( T \in \mathbb{Z}^{j_{\text{out}}} \) \text{and} \( \ell \in \mathbb{Z}^j \), \text{such that} \( m_{\ell+1} > m_{j_{\text{out}}} \).

Then, the following hold.

(i) \( s_j \in (0, \frac{\sqrt{5}}{2}] \) \forall \( j \in Z_{\ell+T}^{\ell+T} \).

(ii) \[ \sum_{j=\ell+2}^{\ell+T} \ln(\max \{1, (4s_j)^4\}) < 4 \ln 15. \]

(iii) \[ \sum_{j=\ell+2}^{\ell+T} \ln \left(\frac{s_j - 1}{s_j}\right) < 4 \ln \left(\frac{s_j}{s_j} - 1\right). \]

(iv) \[ T < 5 + \frac{1}{m_{\ell+1}} \ln \left(1 + \frac{\ell^2 s_0 - \ell^2}{\ell}\right). \]

\text{Proof:} \text{ Denote} \( f_j = f(z_j), j \in Z_{0}^{j_{\text{out}}}. \text{From} \( j \geq 2 \) \text{and Step 4 of Algorithm 2, we have} \]

\[ s_j = f_{j-2} - f_j, j \in Z_{2}^{j_{\text{out}}}. \]

The inequality \( s_j > 0 \) \forall \( j \in Z_{\ell+T}^{\ell+T} \) follows from Property 4.(iii). In order to prove the first claim, it remains to prove the inequality \( s_j \leq \frac{\sqrt{5}}{2} \) \forall \( j \in Z_{\ell+T}^{\ell+T} \). We proceed by \textit{reductio ad absurdum}. Suppose that there is \( j \in Z_{\ell+T}^{\ell+T} \) such that \( s_j > \frac{\sqrt{5}}{2} \).

\[ m_{j+1} \geq n_j = \max\{m_j, 4s_jm_{j-1}\} \geq 4s_jm_{j-1} > \sqrt{15}m_{j-1} \]

which along the nondecreasing nature of the sequence \( \{m_j\} \) (Property 4.(iii)) leads to

\[ m_{\ell+1+T} \geq m_{j+1} > \sqrt{15}m_{j-1} \geq \sqrt{15}m_{\ell+1} \]

contradicting the assumption of the property, proving Lemma 3.(i).

From the nondecreasing nature of the sequence \( \{m_j\} \) (Property 4.(iii)), we have, for every \( j \in Z_{\ell+T}^{\ell+T} \)

\[ m_{j+1} \geq n_j = \max\{m_j, 4s_jm_{j-1}\} \geq m_{j-1} \cdot \max\{1, 4s_j\}, \]

Equivalently, \( \ln(\max\{1, 4s_j\}) \leq \ln \frac{m_{j+1}}{m_{j-1}} \forall j \in Z_{\ell+T}^{\ell+1} \).

This implies

\[ \sum_{j=\ell+2}^{\ell+T} \ln(\max\{1, 4s_j\}) \leq \sum_{j=\ell+2}^{\ell+T} \ln \frac{m_{j+1}}{m_{j-1}} \]

\[ = \ln \frac{m_{\ell+1+T}m_{\ell+1}}{m_{\ell+1}m_{\ell+2}} \leq 2 \ln \frac{m_{\ell+1+T}}{m_{\ell+1}} \]

\[ < 2 \ln \sqrt{15} = \ln 15. \]

The second claim is obtained multiplying the last inequality by 4.

To prove the third claim, we notice that

\[ \sum_{j=\ell+2}^{\ell+T} \left(\frac{s_j - 1}{s_j}\right) < \frac{f_{\ell+1} - f_{\ell}}{\epsilon} \leq \frac{f_0 - f_{\ell+1}}{\epsilon} \leq \frac{f_0 - f^*}{\epsilon} \]

from where the third claim directly follows. In order to prove the last claim of the property, we sum the inequalities given by the second and third claims to obtain

\[ \sum_{j=\ell+2}^{\ell+T} \ln \left(\frac{s_j - 1}{s_j}\right) \cdot \max\{1, (4s_j)^4\} \]

\[ < \ln \left(1 + \frac{f_0 - f^*}{\epsilon}\right) + 4 \ln 15. \]

From the first claim, we have \( s_j \in (0, \frac{\sqrt{5}}{2}] \) \forall \( j \in Z_{\ell+2}^{\ell+T} \). Thus, the left-hand side term of (21) can be lower-bounded by means of the following inequality (Lemma 2):

\[ 15 \leq \left(\frac{1}{s^2} - 1\right) \cdot \max\{1, (4s)^4\} \forall s \in (0, \frac{\sqrt{5}}{2}] \].

That is, \( \sum_{j=\ell+2}^{\ell+T} \ln 15 \leq (1 + \frac{f_0 - f^*}{\epsilon}) + 4 \ln 15. \) Equivalently, \( (T-1) \ln 15 \leq (1 + \frac{f_0 - f^*}{\epsilon}) + 4 \ln 15. \)

\text{REFERENCES}

[1] Y. Nesterov, \textit{Lectures on Convex Optimization}, vol. 137. Berlin, Germany: Springer, 2018.

[2] A. Beck, \textit{First-Order Methods in Optimization}, vol. 25. Philadelphia, PA: USA: SIAM, 2017.

[3] A. Beck and M. Teboulle, \“A fast iterative shrinkage-thresholding algorithm for linear inverse problems,\” \textit{SIAM J. Imag. Sci.}, vol. 2, no. 1, pp. 183–202, 2009.

[4] S. Boyd \textit{et al.}, \”Distributed optimization and statistical learning via the alternating direction method of multipliers,\” \textit{Found. Trends Mach. Learn.}, vol. 3, no. 1, pp. 1–122, 2011.

[5] Y. Nesterov, \”A method of solving a convex programming problem with convergence rate O(1/k^2),\” Sov. Math. doklady, vol. 27, no. 2, pp. 372–376, 1983.

[6] I. Pejicic and C. N. Jones, \”Accelerated ADMM based on accelerated Douglas-Rachford splitting,\” in \textit{Proc. IEEE Eur. Conf. Control,}, 2016, pp. 1952–1957.

[7] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, \”Fast ADMM for homogeneous self-dual embedding of sparse SDPs,\” \textit{IFAC-PapersOnline}, vol. 50, no. 1, pp. 8411–8416, 2017.

[8] P. Patrinos, L. Stella, and A. Bemporad, \”Douglas-Rachford splitting: Complexity estimates and accelerated variants,\” in \textit{Proc. 53rd IEEE Conf. Decis. Control}, 2014, pp. 4234–4239.
[9] T. Goldstein, B. O’Donoghue, S. Setzer, and R. Baraniuk, “Fast alternating direction optimization methods,” SIAM J. Imag. Sci., vol. 7, no. 3, pp. 1588–1623, 2014.
[10] Y. Pu, M. N. Zeilinger, and C. N. Jones, “Fast alternating minimization algorithm for model predictive control,” IFAC Proc. Volumes, vol. 47, no. 3, pp. 11980–11986, 2014.
[11] M. Alamir, “Monitoring control updating period in fast gradient based NMPC,” in Proc. IEEE Eur. Control Conf., 2013, pp. 3621–3626.
[12] S. Richter, C. N. Jones, and M. Morari, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” IEEE Trans. Autom. Control, vol. 57, no. 6, pp. 1391–1403, Jun. 2012.
[13] G. Stathopoulos, M. Korda, and C. N. Jones, “Solving the finite-horizon constrained LQR problem using accelerated dual proximal methods,” IEEE Trans. Autom. Control, vol. 62, no. 4, pp. 1752–1767, Apr. 2017.
[14] P. Patrinos and A. Bemporad, “An accelerated dual gradient-projection algorithm for embedded linear model predictive control,” IEEE Trans. Autom. Control, vol. 59, no. 1, pp. 18–33, Jan. 2014.
[15] M. Kögel and R. Findeisen, “Fast predictive control of linear systems combining Nesterov’s gradient method and the method of multipliers,” in Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf., 2011, pp. 501–506.
[16] V. Nedelcu, I. Necoara, and Q. Tran-Dinh, “Computational complexity of inexact gradient augmented lagrangian methods: Application to constrained MPC,” SIAM J. Control Optim., vol. 52, no. 5, pp. 3109–3134, 2014.
[17] P. Krupa, D. Limon, and T. Alamo, “Implementation of model predictive control in programmable logic controllers,” IEEE Trans. Control Syst. Technol., vol. 29, no. 3, pp. 1117–1130, May 2021.
[18] S. Tao, D. Boley, and S. Zhang, “Local linear convergence of ISTA and FISTA on the LASSO problem,” SIAM J. Optim., vol. 26, no. 1, pp. 313–336, 2016.
[19] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Sparse identification of nonlinear dynamics with control (SINDYc),” IFAC-PapersOnLine, vol. 49, no. 18, pp. 710–715, 2016.
[20] B. O’Donoghue and E. Candes, “Adaptive restart for accelerated gradient schemes,” Found. Comput. Math., vol. 15, no. 3, pp. 715–732, 2015.
[21] P. Giselsson and S. Boyd, “Monotonicity and restart in fast gradient methods,” in Proc. 53rd IEEE Conf. Decis. Control, 2014, pp. 5058–5063.
[22] I. Necoara, Y. Nesterov, and F. Glineur, “Linear convergence of first order methods for non-strongly convex optimization,” Math. Program., vol. 175, no. 1, pp. 69–107, 2019.
[23] Y. Nesterov, “Gradient methods for minimizing composite functions,” Math. Program., vol. 140, pp. 125–161, 2013.
[24] O. Fercoq and Z. Qu, “Adaptive restart of accelerated gradient methods under local quadratic growth condition,” IMA J. Numer. Anal., vol. 39, no. 4, pp. 2069–2095, 2019.
[25] T. Alamo, P. Krupa, and D. Limon, “Restart FISTA with global linear convergence,” in Proc. 18th IEEE Eur. Control Conf., 2019, pp. 1969–1974.
[26] T. Alamo, P. Krupa, and D. Limon, “Gradient based restart FISTA,” in Proc. 58th IEEE Conf. Decis. Control, 2019, pp. 3936–3941.
[27] D. Drusvyatskiy and A. S. Lewis, “Error bounds, quadratic growth, and linear convergence of proximal methods,” Math. Operations Res., vol. 43, no. 3, pp. 919–948, 2018.
[28] P.-W. Wang and C.-J. Lin, “Iteration complexity of feasible descent methods for convex optimization,” J. Mach. Learn. Res., vol. 15, pp. 1523–1548, 2014.
[29] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Berlin, Germany: Springer, 2004.
[30] P. Krupa, R. Jaouani, D. Limon, and T. Alamo, “A sparse ADMM-based solver for linear MPC subject to terminal quadratic constraint,” 2021, arXiv:2105.08419v2.
[31] P. Krupa, “Implementation of MPC in embedded systems using first order methods,” Ph.D. dissertation, Dep. Syst. Eng. Automation, Univ. Seville, Seville, Spain, 2021, arXiv:2109.02140.