SVR-AMA

An asynchronous alternating minimization algorithm with Variance Reduction for Model Predictive Control applications

Ferranti, Laura; Pu, Ye; Jones, Colin N.; Keviczky, Tamas

DOI
10.1109/TAC.2018.2849566

Publication date
2018

Document Version
Final published version

Published in
IEEE Transactions on Automatic Control

SVR-AMA: An Asynchronous Alternating Minimization Algorithm With Variance Reduction for Model Predictive Control Applications

Laura Ferranti, Ye Pu, Colin N. Jones, *Member, IEEE*, and Tamás Keviczky

***Abstract*—This paper focuses on the design of an asynchronous dual solver suitable for model predictive control (MPC) applications. The proposed solver relies on a state-of-the-art variance reduction (VR) scheme, previously used in the context of proximal stochastic gradient methods (Prox-SVRG) and on the alternating minimization algorithm (AMA). The resultant algorithm, a stochastic AMA with VR (SVR-AMA), shows geometric convergence (in the expectation) to a suboptimal solution of the MPC problem and, compared to other state-of-the-art dual asynchronous algorithms, allows one to tune the probability of the asynchronous updates to improve the quality of the estimates. Two novel accelerated versions of the Prox-SVRG (and, by duality, of SVR-AMA) are also provided. We apply the proposed algorithm to a specific class of splitting methods, that is, the decomposition along the length of the prediction horizon. Numerical results on the longitudinal control problem of an Airbus passenger aircraft show the benefits that we can gain in terms of computation time when using our proposed solver with an adaptive probability distribution.**

***Index Terms*—Aerospace, control systems, linear systems, optimization methods, predictive control, quadratic programming.**

I. Introduction

MODEL predictive control (MPC) applications to systems with fast dynamics are still relatively limited [1]–[9]. Applications in fields, such as automotive and aerospace, have to deal often with embedded legacy systems. These systems usually run on certified (for safety purposes) hardware architectures with limited availability of, for example, parallel computation units and only support a small set of (certified) mathematical functions. In particular, the availability of optimization toolboxes suitable for MPC purposes on these platforms are limited (or nonexistent).

First-order methods are promising to address the issues mentioned earlier to use fast MPC applications. Recently, growing attention has been dedicated to the design of *simple* first-order solvers for MPC [10]–[14]. These solvers are relatively easy to certify (in terms of the level of *suboptimality* of the solution), use only simple algebraic operations, and require little memory. In [12] and [14], operator-splitting methods, such as the alternating direction method of multipliers [15] and the fast alternating minimization algorithm (AMA) [16], have been used to exploit the MPC problem structure and speed-up the computation of the solution. These algorithms can also benefit parallel hardware architectures. These algorithms, however, often require frequent exchange of information at given synchronization points leading to computational bottlenecks. To reduce the bottlenecks at the synchronization points, a solver that can offer more *flexibility* in how the solutions are computed (for example, by allowing asynchronous updates) would be attractive. Motivated by the aforementioned issues, in this paper, we are interested in extending the use of splitting methods to this asynchronous framework.

A. Contribution

The contribution of the paper is threefold. First, we propose a novel algorithm, a stochastic AMA with variance reduction (SVR-AMA), suitable for MPC applications with state and input constraints. The proposed algorithm operates in the dual space and combines the advantages of the variance reduction (VR) scheme proposed in [17] and [18] for the proximal stochastic gradient method with the AMA [19]. The result is that the solution of the MPC problem can be computed in an asynchronous fashion (i.e., at each iteration, the algorithm updates a randomly selected subset of the dual variables instead of the whole set of dual variables) and the resultant algorithm has geometric convergence (in the expectation) to the optimal solution. Furthermore, the proposed algorithm allows the use of a generic probability distribution for the asynchronous updates. In addition, the probability distribution can be updated online to improve the

Manuscript received January 10, 2018; revised January 11, 2018 and April 21, 2018; accepted June 4, 2018. Date of publication June 21, 2018; date of current version April 24, 2019. The work of L. Ferranti and T. Keviczky was supported by the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant AAT-2012-RTD-2314544 (RECONFIGURE). The work of Y. Pu was supported by the Swiss NSF Grant Projects P2ELP2_165137. The work of C. N. Jones was supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) REA Grant 607957 (TEMPO). Recommended by Associate Editor I. Kolmanovsky. *(Corresponding author: Laura Ferranti.)*

L. Ferranti and T. Keviczky are with the Delft Center for Systems and Control, Delft University of Technology, Delft 2628, CD, The Netherlands (e-mail: l.ferranti@tudelft.nl; t.keviczky@tudelft.nl).

Y. Pu is with the EECS Department, University of California at Berkeley, Berkeley, CA 94720-5800 USA (e-mail: yepu@eecs.berkeley.edu).

C. N. Jones is with the Automatic Control Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: colin.jones@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2018.2849566

quality of the estimates, as our numerical results show. Finally, the proposed algorithm is suitable for a more general class of optimization problems, namely strongly convex quadratic programming problems with linear coupling constraints.

Second, we show how the proximal stochastic gradient method with VR (Prox-SVRG) can be accelerated, by relying on its similarities with the inexact proximal gradient method (I-PGM) in [20] and the existing convergence results in [18]. In particular, first we show how the VR scheme in [18] can be viewed as an error in the calculation of the gradient that converges (according to the analysis of [18]) geometrically in the expectation. By exploiting this observation, we can rely on the analysis of the work in [20] to accelerate the VR loop (or inner loop) of Prox-SVRG. Second, we derive similar conclusions for the outer loop of Prox-SVRG by combining the convergence analysis derived in [18] for the inner loop with classical stability arguments based on dynamical systems theory. Finally, we apply the proposed acceleration strategies for Prox-SVRG in the dual framework to derive an inner-accelerated SVR-AMA (IA-SVR-AMA) and an outer-accelerated SVR-AMA (OA-SVR-AMA) that can be used to solve problems that come from MPC applications.

Third, we show how we can use SVR-AMA (the same analysis can be extended to its accelerated versions) for a specific splitting technique, that is, the decomposition along the length of the prediction horizon (or *time splitting* [12]), and present simulation results on a practical aerospace application, that is, the longitudinal control of an Airbus passenger aircraft [21]. The results show that the proposed algorithms (i.e., SVR-AMA and its accelerated versions) are more robust when solving ill-conditioned problems, outperforming synchronous methods in terms of computation time (measured in terms of number of iterations) and suboptimality level of the solution.

B. Related Work

SVR-AMA derives from the application to the dual problem of the proximal stochastic gradient method with VR (Prox-SVRG) proposed in [18]. SVR-AMA operates in the dual framework motivated by the presence of coupling constraints in the MPC problem formulation. SVR-AMA has been proposed by the same authors of this paper in [22]. Compared to [22], we extend the analysis of the proposed algorithm with two different acceleration techniques, which is not trivial, provide additional proofs for the convergence of SVR-AMA, and improve the numerical results. An accelerated version of the inner loop of Prox-SVRG has been proposed in [23]. Compared to [23], we approach the analysis of the acceleration of the inner loop from the perspective of an inexact proximal gradient algorithm, which significantly simplifies the analysis, and we provide guidelines to select the number of inner-loop iterations to exploit both the benefits of the acceleration and the VR. Furthermore, an accelerated version of the outer loop of an algorithm similar to Prox-SVRG (i.e., the stochastic dual coordinate ascent method [24]) has been proposed in [25]. The algorithm in [25] focused on regularized loss minimization. Compared to our proposed outer-loop acceleration of Prox-SVRG, the work in [25] requires the minimization of a regularized version of the original cost function in the outer loop. Furthermore, our analysis derives from the results in [18] leading to a simplified proof and an algorithm that can be used to minimize the sum of two functions in which one of the two terms does not have to be strongly convex, which is of importance in order to handle control problems.

The investigation of asynchronous dual algorithms for MPC is gaining more attention recently. In [26], for example, an asynchronous dual algorithm is proposed. Compared to [26], SVR-AMA allows the use of a generic (i.e., not necessarily uniform) probability distribution and, consequently, more flexibility in the tuning phase of the algorithm.

Finally, the idea of the time splitting has been previously proposed in [12]. Their work relies on a synchronous alternating direction method of multipliers. In this context, we reformulate the approach for AMA to exploit SVR-AMA.

C. Outline

The paper is structured as follows. Section II introduces the MPC problem formulation. Section III summarizes AMA and Prox-SVRG. Section IV details the two acceleration techniques (Section IV-A describes the acceleration of the inner loop and Section IV-C describes the acceleration of the outer loop). Section V introduces SVR-AMA and its accelerated versions. Then, Section VI shows how to reformulate the proposed MPC problem for SVR-AMA using the time splitting. Section VII presents numerical results using an aerospace example. Section VIII concludes the paper. Finally, Appendices A–C provide all the proofs contained in this paper.

D. Notation

For $u \in \mathbb{R}^n$, $\|u\| = \sqrt{\langle u, u \rangle}$ is the Euclidean norm. Let $\mathbb{C}$ be a convex set. Then, $\mathbf{Pr}_{\mathbb{C}}(u)$ is the projection of u onto $\mathbb{C}$. Let $f : \mathcal{D} \to \mathcal{C}$ be a function. Then, $f^\star(y) = \sup_x(y^{\mathrm{T}}x - f(x))$ and $\nabla f(x)$ are the conjugate function and the gradient of $f(x)$, respectively. Furthermore, $\delta_{\mathbb{C}}(\sigma)$ is the indicator function on the convex set $\mathbb{C}$, which is zero if $\sigma \in \mathbb{C}$ and infinity otherwise. Let $A \in \mathbb{R}^{n\times m}$. Then, $\mathrm{eig}_{\max}(A)$ and $\mathrm{eig}_{\min}(A)$ are the largest and the smallest (modulus) eigenvalues of $A^{\mathrm{T}}A$. $P \in \mathbb{S}_+^{n\times n}$ denotes that $P \in \mathbb{R}^{n\times n}$ is positive definite. In addition, let $x \in \mathbb{R}^n$ be a random variable, $\mathbb{E}[x]$ is its expected value. Finally, details on the notions of strong convexity and Lipschitz continuity used in the paper can be found in [27].

II. Problem Formulation

Consider the discrete linear time-invariant (LTI) system as follows:

$$x(t+1) = Ax(t) + Bu(t), \quad t = 0, 1, 2, \ldots. \tag{1}$$

The state $x(t) \in \mathbb{R}^n$ and the control input $u(t) \in \mathbb{R}^m$ are subject to the following polyhedral constraints:

$$Cx(t) + Du(t) \leq d \tag{2}$$

where $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. Note that the definition of constraints (2) can include constraints on $x(t)$ only or on $u(t)$ only. We aim to regulate the state $x(t)$ to the origin using the control input $u(t)$ while respecting constraints (2). This goal

Algorithm 1: AMA [19].

Given μ_0, T, and $\tau < \sigma_f / \operatorname{eig}_{\max}(H_y)$.
while $k = 1, \dots, T$ **do**
1a. $y_k = \operatorname{argmin}_y \; f(y) + \langle \mu_{k-1}, -H_y y \rangle$.
1b. $z_k = \operatorname{argmin}_z \; g(z) + \langle \mu_{k-1}, -H_z z \rangle + \frac{\tau}{2} \| d - H_y y_k - H_z z \|^2$.
2. $\mu_k = \mu_{k-1} + \tau(d - H_y y_k - H_z z_k)$.
end while

can be translated into the following MPC problem:

$$\min_{x,u} \; \frac{1}{2} \sum_{t=0}^{N} x_t^{\mathrm{T}} Q x_t + \frac{1}{2} \sum_{t=0}^{N-1} u_t^{\mathrm{T}} R u_t \tag{3a}$$

$$\text{s.t.: } x_{t+1} = A x_t + B u_t, \quad t = 0, \dots, N-1 \tag{3b}$$

$$C x_t + D u_t \leq d, \qquad t = 0, \dots, N-1 \tag{3c}$$

$$C x_N \leq d, \quad x_0 = x_{\text{init}} \tag{3d}$$

where x_t and u_t represent the t-step-ahead state and control predictions, respectively. Furthermore, N indicates the length of the prediction horizon, $Q \in \mathbb{S}_+^{n \times n}$, $R \in \mathbb{S}_+^{m \times m}$. Finally, x_{init} is the initial (measured) state vector. We assume that x_{init} is such that a solution of the aforementioned MPC problem exists.

The MPC law implemented in closed loop is given by the first element of the optimal control sequence obtained by solving Problem (3), that is, $u_{\text{MPC}} = u_0^*$.

Our goal is to solve Problem (3) in an embedded environment. In particular, we assume that explicit MPC [28] cannot be used due to the problem size and that the computational resources are limited, that is, parallel architectures are not available, memory resources are limited, and only simple algebraic operations are supported. With this framework in mind, in the remainder of the paper, we focus on the design of a simple solver for Problem (3). The proposed solver relies on: 1) operator-splitting methods (which, for example, usually rely on parallel hardware architectures) and *2)* asynchronicity (which allows one to perform updates of a randomly selected subset of variables to reduce the computational effort). The following section introduces the techniques we rely on to solve Problem (3): AMA [19] and the proximal stochastic gradient descent method with VR (Prox-SVRG) [18].

III. Preliminaries

A. AMA: Alternating Minimization Algorithm

Consider the following problem:

$$\text{minimize } f(y) + g(z) \tag{4a}$$

$$\text{subject to } H_y y + H_z z = d \tag{4b}$$

where $f(y) := \sum_{t=0}^{N} f^{(t)}(y)$ under the following assumptions.

Assumption 1: $f^{(t)}$ is a strongly convex function and $\sigma_{f^{(t)}}$ denotes its convexity parameter ($t = 0, \dots, N$).

Assumption 2: $f^{(t)}$ has a Lipschitz continuous gradient with modulus $L_{f^{(t)}}$ ($t = 0, \dots, N$).

Assumption 3: g is a convex function not necessarily smooth.

The MPC problem (3) that we aim to solve is a particular case of Problem (4). Section VI provides more details on the relationship between the two problems.

Recall the following properties of the conjugate function $f^\star$.

Lemma III.1 (see Th. 4.2.1 in [29]): If f is strongly convex with convexity parameter σ_f, then $f^\star$ has a Lipschitz continuous gradient with constant $L(\nabla f^\star) = \sigma_f^{-1}$.

Lemma III.2 (see Th. 4.2.2 in [29]): If f is convex and has a Lipschitz continuous gradient with modulus L_f, then $f^\star$ is strongly convex with convexity parameter L_f^{-1}.

An algorithm suitable to solve Problem (4) is the AMA proposed in [19]. AMA operates as a proximal gradient algorithm, such as the iterative shrinkage-thresholding algorithm (ISTA) [30], on the dual of Problem (4). Specifically, given the dual of Problem (4) (under the assumptions mentioned earlier), described as follows:

$$\underset{\mu \in \mathbb{R}^{n_\mu}}{\text{maximize}} \; D(\mu) \tag{5}$$

where $D(\mu) := -F_d(\mu) - G_d(\mu)$, $F_d(\mu) := \sum_{t=0}^{N} F_t(\mu)$, $F_t(\mu) := f^{(t)^\star}(H_y^{(t)^{\mathrm{T}}} \mu)$ for $t = 0, \dots, N$, and $G_d(\mu) := g^\star(H_z^{\mathrm{T}} \mu) - d^{\mathrm{T}} \mu$, the following holds.

Lemma III.3: If Assumptions 1–3 are satisfied, $F_d(\mu)$ is strongly convex with Lipschitz continuous gradient characterized by Lipschitz constant $L^\star = L(\nabla F_d) := \operatorname{eig}_{\max}(H_y) \sigma_f^{-1}$. Furthermore, $G_d(\mu)$ is convex with convexity parameter σ_{G_d}.

Proof: We can use Lemmas III.1 and III.2 to derive the properties of $F(\mu)$. Convexity of $G(\mu)$ follows from the properties of the conjugate of a convex function and from the fact that $d^{\mathrm{T}} \mu$ is a linear function. ■

Remark 1: An example of $g(z)$ that satisfies Assumption 3 is the indicator function on the closed convex set $\mathbb{C}$, that is, $g(z) = \delta_{\mathbb{C}}(z)$. In particular, if $g(z) = \delta_{\mathbb{C}}(z)$, then $G_d(\mu)$ is a support function, that is, $G_d(\mu)$ is a convex function. Furthermore, note that σ_{G_d} is only required for theoretical purposes and its value is not needed to tune the parameters of Algorithm 5 and its accelerated versions.

AMA updates the dual variables $\mu \in \mathbb{R}^{p_\mu}$ as described in Algorithm 1. In general, AMA uses only simple algebraic operations (if y and z are unconstrained, Steps 1a and 1b can be performed efficiently) and does not require advanced hardware architectures. Nevertheless, the algorithm requires frequent exchange of information at given synchronization points (e.g., Step 1b requires y computed at Step 1a, which can lead to bottlenecks in the computation of the problem solution). Hence, it would be better to have some flexibility in the update strategy. Motivated by this observation, the following section introduces Prox-SVRG used to derive our proposed asynchronous AMA, as described in Section V.

B. Prox-SVRG: Proximal Stochastic Gradient Method With VR

Consider the following primal problem:

$$\underset{y \in \mathbb{R}^{n_y}}{\text{minimize}} \; P(y) \tag{6}$$

Algorithm 2: Prox-SVRG [18].

Given $\tilde{y}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}$, η, and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{y} = \tilde{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{y})$.
0c. Set $y_0 = \tilde{y}$.
0d. Set $\Pi := \{\pi_0, \dots, \pi_N\}$.
for $k = 1, \dots, T$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(y_{k-1}) - \nabla F_i(\tilde{y})}{\pi_i}$.
3. $y_k = \text{prox}_{\eta G}(y_{k-1} - \eta\beta_k)$.
end for
4. $\tilde{y}_s = (1/T)\sum_{k=1}^{T} y_k$.
6. $s = s + 1$.
end while

where $P(y) := F(y) + G(y)$, $F(y) := \sum_{t=0}^{N} F_t(y)$, and $G(y)$ satisfy the following assumptions.

Assumption 4: $F(y)$ is a strongly convex function with convexity parameter σ_F and Lipschitz continuous gradient characterized by a Lipschitz constant $L \leq \sum_{t=0}^{N} L_t$, where L_t are the Lipschitz constants of each $F_t(y)$.

Assumption 5: $G(y)$ is a convex function.

Furthermore, define the *proximal operator* as follows:

$$\text{prox}_{\eta G}(x) := \underset{y \in \mathbb{R}^{n_y}}{\text{argmin}} \left\{ \frac{1}{2} \|y - x\|^2 + \eta G(y) \right\}. \tag{7}$$

The main idea behind Prox-SVRG [18] is to eliminate the dependence of the number of iterations (typical of stochastic gradient methods) in the definition of the step size and to reduce the burden in the computation of $\nabla F(y)$ (typical of classical gradient methods). As pointed out in [18], proximal stochastic gradient methods (such as [31] and [32]) suffer from sublinear convergence [to a suboptimal solution of Problem (6)] given that the step size decreases at each iteration of the algorithm, but behave well when N is large. On the other hand, classical proximal gradient methods require at each iteration of the algorithm to compute the full gradient of $F(y)$, which can be an involved operation if N is large, but the step size is fixed and independent of the number of iterations (leading to better theoretical convergence properties). Hence, Prox-SVRG aims to exploit the benefits of the two techniques as explained ahead and described in Algorithm 2.

Prox-SVRG uses a multistage strategy to gradually reduce the variance in the estimation of the full gradient $\nabla F(y)$ (without computing the actual full gradient at each iteration). In particular, the full gradient of $F(y)$ is updated only every T iterations to reduce the computational effort compared to the classical gradient methods, and the proximal step (Step 3) uses a modified direction β_k (Step 2) that leads to a smaller variance $\mathbb{E}\|\beta_k - \nabla F(y_{k-1})\|^2$ compared to the one obtained using classical stochastic gradient methods $\mathbb{E}\|\nabla F_i(y_{k-1}) - \nabla F(y_{k-1})\|^2$ ($i \in \mathcal{D}_N$), where $\nabla F_i(y_{k-1})$ is used as update direction (refer to [18] for more details). Furthermore, the random sampling (Step 1) is performed on a probability distribution $\Pi := \{\pi_0, \dots, \pi_N\}$ that does not necessarily have to be uniform, that is, the algorithm allows more flexibility in the tuning phase by supporting other distributions as well, such as Poisson distributions, normal distributions, etc. Algorithm 2 achieves geometric convergence in the expectation, as stated in the following theorem.

Theorem III.1 (Th. 3.1 in [18]): Suppose Assumptions 4 and 5 hold. Let $y_* = \text{argmin}_y P(y)$ and $L_\Pi := \max_t L_t/\pi_t$. Assume that $0 < \eta < 1/(4L_\Pi)$ and T is sufficiently large so that

$$\rho := \frac{1}{\eta\sigma_F T(1 - 4\eta L_\Pi)} + \frac{4\eta L_\Pi (T+1)}{T(1 - 4\eta L_\Pi)} < 1. \tag{8}$$

Then, for $\bar{s} > 1$, Algorithm 2 has geometric convergence in the expectation

$$\mathbb{E}\left(P(\tilde{y}_{\bar{s}})\right) - P(y_*) \leq \rho^{\bar{s}} \left[P(\tilde{y}_0) - P(y_*)\right]. \tag{9}$$

Remark 2: The dependence on the probability π_t in the choice of the step size η can be problematic when $\pi_t \to 0$ or when $N \to \infty$ (e.g., the constrained infinite horizon linear quadratic regulator (LQR)). Nevertheless, this dependence can be removed in the special case in which $F(y) = \sum_{t=1}^{N} F_t(y_t)$, that is, when the cost is separable in y_t. In this scenario, we can select $0 < \eta < 1/(4 \max_t L_t)$. From the MPC perspective, this covers many scenarios, such as regulation and tracking problems in which we optimize with respect to states and inputs. Distributed MPC applications can have separable costs in the local decision variables. A negative example, in which the cost is not separable in the decision variables, is when a condensed centralized MPC formulation (in which we optimize with respect to the control variables eliminating the dynamic constraints) is used. Hence, this observation can be taken into account in the design phase (when designing the controller itself) to improve the choice of the step size and the quality of the MPC solution.

IV. Accelerated Prox-SVRG

In the following, we propose two different acceleration strategies for Algorithm 2. In particular, Section IV-A describes a strategy to accelerate the inner loop of Prox-SVRG (see Algorithm 3), whereas Section IV-C describes a strategy to accelerate the outer loop of Prox-SVRG (see Algorithm 4). We rely on the following observation in order to show that we can accelerate Prox-SVRG.

A. Analysis of Algorithm 3

The analysis of Algorithm 3 relies on some properties of Algorithm 2. In particular, Algorithm 2 can be interpreted as a proximal gradient method (such as ISTA [30]) in which the gradient of F is computed inexactly, that is, by using a modified direction β_k obtained by using the VR strategy. In particular, Prox-SVRG reduces to the inexact proximal gradient proposed in [20], in which the full gradient of F is computed inexactly (Step 3 of Algorithm 2)

$$y_k = \text{prox}_{\eta G}\left(y_{k-1} - \eta\beta_k\right) \tag{10a}$$

$$= \text{prox}_{\eta G}\left(y_{k-1} - \eta\left(\nabla F(y_{k-1}) + e_k\right)\right). \tag{10b}$$

Algorithm 3: Inner-Loop Acceleration of Prox-SVRG.

Given $\tilde{y}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \ldots, N\}$ η, $\gamma := \frac{\sigma_F}{L}$, and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{y} = \tilde{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{y})$.
0c. Set $y_0 = \tilde{y}$.
0d. Set $\Pi := \{\pi_0, \ldots, \pi_N\}$.
for $k = 1, \ldots, T+1$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(\hat{y}_{k-1}) - \nabla F_i(\tilde{y})}{\pi_i}$.
3. $y_k = \text{prox}_{\eta G}(\hat{y}_{k-1} - \eta\beta_k)$.
4. $\hat{y}_k = y_k + \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(y_k - y_{k-1})$.
end for
5. $\tilde{y}_s = (1/T)\sum_{k=1}^{T} y_k$.
6. $s = s + 1$.
end while

The error in the gradient calculation is defined as follows:

$$e_k = \beta_k - \nabla F(y_{k-1}). \tag{11}$$

Hence, Algorithm 2 is a particular case of the I-PGM proposed in [20]. As a consequence, we can derive a framework similar to the one proposed in [20] to show the acceleration of Prox-SVRG. In this respect, we proceed as follows.

1) The first step is to use the convergence results in [20, Proposition 4] to derive conclusions on the convergence of the inner loop of Algorithm 3.
2) Second, we check that the expectation on the gradient calculation error $\mathbb{E}\|e_k\|^2$ is bounded and converges linearly to zero, in order to satisfy the assumptions in [20].
3) Third, we show how the acceleration of the inner loop impacts the outer loop of Algorithm 3.

Analysis of the inner loop of Algorithm 3: In the following, we reformulate [20, Proposition 4] for the problem we take into account, that is, we do not consider the error in the calculation of the proximal operator and we consider that we deal with stochastic variables in the inner loop of Algorithm 3.

Proposition IV.1: Suppose Assumptions 4 and 5 hold. For $k \geq 0$ and y_k computed according to Algorithm 3, the following holds:

$$\mathbb{E}P(y_k) - P(y_*) \leq (1-\sqrt{\gamma})^k \left[\sqrt{2(P(\tilde{y}) - P(y_*))} + \Gamma_k(\tilde{y})\sqrt{\frac{2}{\sigma_F}}\right]^2 \tag{12}$$

with $\Gamma_k(\tilde{y}) := \sum_{i=1}^{k} \mathbb{E}\|e_{i-1}(\tilde{y})\|(1-\sqrt{\gamma})^{-i/2}$.

Proof: We exploit the equivalence between Prox-SVRG and I-PGM. In particular, recall (10) and (11). Then, the proof follows directly from [20, Proposition 4] by taking into account that the proximal step is computed exactly and the only error we take into account is the one in the gradient calculations, which is a stochastic error (hence the expectation in the definition of Γ_k). Furthermore, we take into account that each y_k is a stochastic variable and, consequently, we can consider the expectation in the value of $P(y_k)$. ■

Bound on the variance in Algorithm 3: The error in the gradient calculations that we take into account by using Prox-SVRG is a stochastic error and it is bounded (in the expectation) according to [18, Corollary 3.5]. We can compute the bound on the error when β_k is computed according to Step 2 in Algorithm 3.

Corollary IV.1: Consider β_k, which is defined as follows ($k = 0, \ldots, T$):

$$\beta_k = \nabla F(\tilde{y}) + \frac{\nabla F_{i_k}(\hat{y}_{k-1}) - \nabla F_{i_k}(\tilde{y})}{\pi_{i_k}}. \tag{13}$$

In addition, let $L_\Pi = \max_{i=1,\ldots,N} L_i/(\pi_i)$.[1] Conditioned on y_{k-1}, we have $\mathbb{E}\beta_k = \nabla F(\hat{y}_{k-1})$ and

$$\mathbb{E}\|e_{k-1}(\tilde{y}_{s-1})\|^2 \leq 4L_\Pi[P(\hat{y}_{k-1}) + P(\tilde{y}_{s-1}) - 2P(y_*)]. \tag{14}$$

Proof: The proof follows from the one of [18, Corollary 3.5] by using the update rule for β_k in Step 2 of Algorithm 3. ■

Remark 3: Accelerating the inner loop does not affect the upper bound on the error, as can be easily shown by looking at the proof of [18, Corollary 3.5]. The upper bound above suggests that when $\tilde{y}_s \to y_*$ and $\hat{y}_{k-1} := (1-\alpha)y_{k-1} + \alpha y_{k-2} \to y_*$ ($\alpha = (1-\sqrt{\gamma})/(1+\sqrt{\gamma}) < 1$), the expected error in the gradient calculation is zero, that is, the error goes to zero at the same rate of the estimates (i.e., $\hat{y}_k$ and $\tilde{y}_s$) of the optimal solution of Problem (6). As shown in Theorem III.1, in which the convergence of Prox-SVRG is discussed, $\tilde{y}_s \to y_*$ and consequently $\mathbb{E}\|e_k\|$ is guaranteed by design to converge to zero. For $\mathbb{E}\|e_k\|^2$ to decrease to zero linearly (in order to exploit the upper bound provided in Proposition IV.1), the following condition must be verified (we omit the dependence on $\tilde{y}_{s-1}$ when it is clear from the context):

$$\mathbb{E}\|e_k\|^2 \leq \frac{1}{k}\mathbb{E}\|e_{k-1}\|^2. \tag{15}$$

The aforementioned condition can be checked on the error upper bound (14). If we formulate this condition, we notice immediately that we can derive guidelines to select the batch size T, that is, the number of inner-loop iterations. In particular, the following holds:

$$P(y_{k-1}) \leq \frac{1-\alpha-k\alpha}{k(1-\alpha)}P(y_{k-2}) + \frac{\alpha}{k(1-\alpha)}P(y_{k-3}) + \frac{2}{(1-\alpha)}P(y_*) + \frac{1}{k(1-\alpha)}P(\tilde{y}).$$

The right-hand side of the previous equation is the sum of positive terms if and only if the coefficient of $P(y_{k-2})$ is greater than or equal to zero. Hence, we have to limit the number of inner-loop iterations. In particular, we can derive the following upper bound:

$$T \leq \left\lceil \frac{2\sqrt{\gamma}}{1-\sqrt{\gamma}} \right\rceil \tag{16}$$

which means that if the problem is ill-conditioned, the acceleration is not recommended.

[1] i in this case is not the iteration counter, but indicates that L_i is associated with the function F_i, $i = 1, \ldots, N$.

Algorithm 4: Outer-Loop Acceleration of Prox-SVRG.

Given $\tilde{y}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}$ η, γ, and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{y} = \hat{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{y})$.
0c. Set $y_0 = \tilde{y}$.
0d. Set $\Pi := \{\pi_0, \dots, \pi_N\}$.
for $k = 1, \dots, T$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(y_{k-1}) - \nabla F_i(\tilde{y})}{\pi_i}$.
3. $y_k = \text{prox}_{\eta G}(y_{k-1} - \eta\beta_k)$.
end for
4. $\tilde{y}_s = (1/T)\sum_{k=1}^{T} y_k$.
5. $\hat{y}_s = \tilde{y}_s + \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(\tilde{y}_s - \tilde{y}_{s-1})$.
6. $s = s + 1$.
end while

Analysis of the outer loop of Algorithm 3: By exploiting Proposition IV.1, we are able to immediately derive an upper bound for the cost computed using Algorithm 3 that depends on Γ_k, as described in the following theorem.

Theorem IV.1: Suppose Assumptions 4 and 5 hold. Let y_* be the optimal solution of Problem (6), $\gamma := (\sigma_F/L_\Pi)$, and $0 < \eta \leq (1/L_\Pi)$. For $\lfloor \frac{4}{\sqrt{\gamma}}(1-\sqrt{\gamma}) \rfloor \leq T \leq \lceil \frac{2\sqrt{\gamma}}{1-\sqrt{\gamma}} \rceil$ and $\bar{s} > 0$, the following holds:

$$\mathbb{E}P(\tilde{y}_{\bar{s}}) - P(y_*) \leq \rho_{\text{IA,I-PGM}}^{\bar{s}} \left[P(\tilde{y}_0) - P(y_*)\right] + \frac{4}{T\sigma_F} \sum_{s=0}^{\bar{s}-1} \rho_{\text{IA,I-PGM}}^{s} \Gamma(\tilde{y}_s) \tag{17}$$

where $\rho_{\text{IA,I-PGM}}$ and $\Gamma(\tilde{y}_s)$ are defined as follows:

$$\rho_{\text{IA,I-PGM}} := \frac{4}{T\sqrt{\gamma}} \left[(1-\sqrt{\gamma}) - (1-\sqrt{\gamma})^{T+1}\right] < 1 \tag{18a}$$

$$\Gamma(\tilde{y}_s) := \sum_{k=1}^{T} \frac{\mathbb{E}\|e_{k-1}(\tilde{y}_{s-1})\|^2}{(1-\sqrt{\gamma})^k}. \tag{18b}$$

The proof can be found in Appendix A-A.

Remark 4: Note that $T > \frac{4}{\sqrt{\gamma}}(1-\sqrt{\gamma})$ is required in order to have $\rho_{\text{IA,I-PGM}} < 1$. If we analyze the upper and lower bound on T in the statement of Theorem IV.1, Algorithm 3 can be used only when the conditioning of Problem (6) is such that the following holds:

$$1 \leq 2\sqrt{\gamma} \tag{19}$$

which only holds if $1 > \gamma > 0.25$.

B. Analysis of Algorithm 4

In the following, we propose an alternative acceleration strategy of Prox-SVRG. Compared to the results provided in Section IV-A, this section focuses on the acceleration of the outer loop of Prox-SVRG. In this respect, we proceed by analyzing Algorithm 4 as follows.

1) First, we check that the acceleration of the outer loop does not affect the bound on the error in the gradient calculations of the inner loop.
2) Second, we provide a proof of convergence of Algorithm 4. This proof mainly relies on the convergence analysis in [18].

Bound on the variance in Algorithm 4: In the following, we show that the acceleration of the outer loop does not affect the VR strategy in the inner loop and that the results of [18, Corollary 3.5] hold to prove the convergence of Algorithm 4. In particular, when β_k is computed according to Step 2 in Algorithm 4, the following holds.

Corollary IV.2: Consider β_k, which is defined as follows:

$$\beta_k = \nabla F(\tilde{y}) + \frac{\nabla F_{i_k}(y_{k-1}) - \nabla F_{i_k}(\tilde{y})}{\pi_{i_k}}. \tag{20}$$

In addition, let $L_\Pi := \max_i L_i/(N\pi_i)$, $\beta_s := \sum_{k=1}^{T} \beta_k$, and $y_{s-1,\text{avg}} := \frac{1}{T}\sum_{k=1}^{T} y_{k-1}$. Conditioned on y_{k-1}, the following holds:

$$\mathbb{E}\beta_s := \nabla F(y_{\text{avg}})$$

and

$$\mathbb{E}\|e_s\|^2 \leq \frac{2L_\Pi}{T}\left[P(y_{s-1,\text{avg}}) + P(\tilde{y}) - 2P(y_*)\right]. \tag{21}$$

The proof can be found in Appendix B-A.

Accelerating the outer loop does not affect the upper bound on the error expectation provided in [18]. Furthermore, notice that (21) provides guidelines to select the number of inner-loop iterations. In particular, if the following holds, the error in the gradient calculations decreases linearly with respect to $\tilde{y} \to y_*$ and $y_{s-1,\text{avg}} \to y_*$:

$$T > \lceil 2L_\Pi \rceil. \tag{22}$$

Convergence of Algorithm 4: We can now show the convergence of Algorithm 4 using the following theorem.

Theorem IV.2: Suppose Assumptions 4 and 5 hold. For $s \geq 0$, the following holds for $\tilde{y}_s$ computed according to Algorithm 4:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq (1-\sqrt{\gamma})^s (P(\tilde{y}_0) - P(y_*)). \tag{23}$$

The proof can be found in Appendix B-B.

Remark 5: The outer-loop acceleration converges geometrically at a rate that depends on $\gamma := \mu/L$. If $(1-\sqrt{\gamma}) < \rho$, we can expect to converge to the optimal solution at a faster rate, compared to Algorithm 2. If that does not hold, due to the problem conditioning, the acceleration of the outer loop is not beneficial. In general, note that ρ depends on the number of inner-loop iterations T and $\rho \ll 1$ only if T is large. Hence, in general, we expect that the condition for the acceleration $(1-\sqrt{\gamma}) < \rho$ holds in many applications to keep the overall computation time of the algorithm bounded (i.e., we expect to keep T small to reduce the computation time of the algorithm).

V. SVR-AMA and its Accelerated Versions

Our goal is to solve Problem (4) in an asynchronous fashion, that is, by allowing updates of a randomly selected subset of the

Algorithm 5: SVR-AMA.

Given $\tilde{\mu}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}$, η, γ, and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{\mu} = \tilde{\mu}_{s-1}$, $\tilde{y} = \tilde{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{\mu})$.
0c. Set $\mu_0 = \tilde{\mu}$.
0d. Set $\Pi := \{\pi^{(0)}, \dots, \pi^{(N)}\}$.
for $k = 1, \dots, T$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2a. $y_k = \operatorname{argmin}_y f^{(i)}(y) + \langle \mu_{k-1}, -H_y^{(i)} y \rangle$.
2b. $z_k = \operatorname{argmin}_z g(z) + \langle \mu_{k-1}, -H_z z \rangle + \frac{\eta}{2}\|d - H_y y_k - H_z z\|^2$.
3. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(\mu_{k-1}) - \nabla F_i(\tilde{\mu})}{\pi_i}$.
4. $\mu_k = \mu_{k-1} - \eta(\beta_k + H_z z_k - d)$.
end for
5. $\tilde{\mu}_s = \frac{1}{T}\sum_{k=1}^{T} \mu_k$, $\tilde{y}_s = \frac{1}{T}\sum_{k=1}^{T} y_k$.
6. $s = s + 1$.
end while

Algorithm 6: Inner-Loop Accelerated SVR-AMA.

Given $\tilde{\mu}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}\eta$, γ, and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{\mu} = \tilde{\mu}_{s-1}$, $\tilde{y} = \tilde{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{\mu})$.
0c. Set $\hat{\mu}_0 = \mu_0 = \tilde{\mu}$.
0d. Set $\Pi := \{\pi_0, \dots, \pi_N\}$.
for $k = 1, \dots, T+1$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2a. $y_k = \operatorname{argmin}_y f^{(i)}(y) + \langle \hat{\mu}_{k-1}, -H_y^{(i)} y \rangle$.
2b. $z_k = \operatorname{argmin}_z g(z) + \langle \hat{\mu}_{k-1}, -H_z z \rangle + \frac{\eta}{2}\|d - H_y y_k - H_z z\|^2$.
3. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(\hat{\mu}_{k-1}) - \nabla F_i(\tilde{\mu})}{\pi_i}$.
4. $\mu_k = \hat{\mu}_{k-1} - \eta(\beta_k + H_z z_k - d)$.
5. $\hat{\mu}_k = \mu_k + \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(\mu_k - \mu_{k-1})$.
end for
6. $\tilde{\mu}_s = (1/T)\sum_{k=1}^{T} \mu_k$, $\tilde{y}_s = (1/T)\sum_{k=1}^{T} y_k$.
7. $s = s + 1$.
end while

Algorithm 7: Outer-Loop Accelerated SVR-AMA.

Given $\hat{\mu}_0$, $\tilde{y}_0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}$ η, γ and T.
while $s \leq \bar{s}$ **do**
0a. Set $\tilde{\mu} = \hat{\mu}_{s-1}$, $\tilde{y} = \tilde{y}_{s-1}$.
0b. Set $\tilde{\beta} = \nabla F(\tilde{\mu})$.
0c. Set $\mu_0 = \tilde{\mu}$.
0d. Set $\Pi := \{\pi_0, \dots, \pi_N\}$.
for $k = 1, \dots, T$ **do**
1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
2a. $y_k = \operatorname{argmin}_y f^{(i)}(y) + \langle \mu_{k-1}, -H_y^{(i)} y \rangle$.
2b. $z_k = \operatorname{argmin}_z g(z) + \langle \mu_{k-1}, -H_z z \rangle + \frac{\eta}{2}\|d - H_y y_k - H_z z\|^2$.
3. $\beta_k = \tilde{\beta} + \frac{\nabla F_i(\mu_{k-1}) - \nabla F_i(\tilde{\mu})}{\pi_i}$.
4. $\mu_k = \mu_{k-1} - \eta(\beta_k + H_z z_k - d)$.
end for
5. $\tilde{\mu}_s = (1/T)\sum_{k=1}^{T} \mu_k$, $\tilde{y}_s = (1/T)\sum_{k=1}^{T} y_k$.
6. $\hat{\mu}_s = \tilde{\mu}_s + \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(\tilde{\mu}_s - \tilde{\mu}_{s-1})$.
7. $s = s + 1$.
end while

dual variables at each iteration of the solver. Hence, given that Algorithm 2 cannot be directly applied to Problem (4), we proceed as explained in Section III-A, that is, we apply Algorithm 2 to the dual of Problem (4). The resulting algorithm (SVR-AMA) is described by Algorithm 5. In order to derive convergence results for Algorithm 5, we consider the dual formulation of [18, Lemma 3.6]. Note that in the remainder of the paper, we use $F(\mu)$ and $G(\mu)$ instead of $F_d(\mu)$ and $G_d(\mu)$ to simplify the notation.

Lemma V.1: Let $D(\mu) = -F(\mu) - G(\mu)$ defined in (5), where $\nabla F(\mu)$ is Lipschitz continuous with parameter $L^\star := \sigma_f^{-1}$ (according to Lemma III.1), and $F(\mu)$ and $G(\mu)$ have convexity parameters $\sigma_F := L_f^{-1}$ and σ_G, respectively. For any $\mu \in \operatorname{dom}(G)$ and $\beta \in \mathbb{R}^{n_\mu}$, define

$$\mu^+ = \operatorname{prox}_{\eta G(\mu)}(\mu - \eta\beta) \tag{24a}$$

$$h = \frac{1}{\eta}(\mu - \mu^+) \tag{24b}$$

$$\Delta = \beta - \nabla F(\mu) \tag{24c}$$

where $\eta \leq \sigma_f$. Then, for any $\tilde{\mu} \in \mathbb{R}^{n_\mu}$, we have

$$D(\mu^+) - D(\tilde{\mu}) \geq h^{\mathrm{T}}(\tilde{\mu} - \mu) + \frac{\eta}{2}\|h\|^2 + \frac{\sigma_F}{2}\|\tilde{\mu} - \mu\|^2 + \frac{\sigma_G}{2}\|\tilde{\mu} - \mu^+\|^2 + \Delta^{\mathrm{T}}(\mu^+ - \tilde{\mu}). \tag{25}$$

Proof: The definition of μ^+ in (24a) follows from [16, Th. 4] (in particular using [16, Th. 4], it is shown that Steps 2 and 4 in Algorithm 5 are equivalent to $\operatorname{prox}_{\eta G}(\mu - \eta\beta)$, which is equivalent to the proximal Step 3 of Algorithm 2). Then, (25) follows from the proof of [18, Lemma 3.6] by taking into account the definition of $D(\mu)$. ■

We can now establish the convergence of Algorithm 5.

Theorem V.1: Suppose Assumptions 1–3 hold. Let $\mu^* = \operatorname{argmax}_\mu D(\mu)$, where $D(\mu)$ is the dual cost defined in (5). Let $L_\Pi^\star := \max_{t=0,\dots,N} \operatorname{eig}_{\max}(H_y)(\pi_t \sigma_f)^{-1} = \max_{t=0,\dots,N} \pi_t^{-1} L^\star$, $\pi_t \in \Pi$. Assume that $0 < \eta < 1/(4L_\Pi^\star)$ and $T \geq 1$ such that

$$\rho^\star := \frac{L_f}{\eta T(1 - 4\eta L_\Pi^\star)} + \frac{4\eta L_\Pi^\star (T+1)}{T(1 - 4\eta L_\Pi^\star)} < 1. \tag{26}$$

Then, for $\bar{s} > 0$, Algorithm 5 has geometric convergence in the expectation

$$D(\mu_*) - \mathbb{E}D(\tilde{\mu}_{\bar{s}}) \leq \rho^{\star \bar{s}} \left[D(\mu_*) - D(\tilde{\mu}_0)\right]. \tag{27}$$

The proof can be found in Appendix C-A.

Corollary V.1 (Corollary 3.5 in [18] on the dual): Consider β_k, which is defined in Step 3 of Algorithm 5. Conditioned on μ_{k-1}, the following holds:

$$\mathbb{E}\beta_k = \nabla F(\mu_{k-1}) \tag{28}$$

and

$$\mathbb{E}\|\beta_k - \nabla F(\mu_{k-1})\|^2 \leq 4L_\Pi^\star[P(y_k) + P(\tilde{y}) - 2P(y_*)]. \tag{29}$$

The proof can be found in Appendix C-B.

Remark 6: According to Corollary V.1, by exploiting the definition of $\nabla F(\mu)$, we can relate the upper bound on the variance in the gradient calculations in the dual framework with the primal cost. This observation is useful to preserve (in the dual framework) the results obtained in the primal framework in Sections IV-A and IV-C to select the inner- and outer-loop iterations for $\mathbb{E}\|\beta_k - \nabla F(\mu_{k-1})\|^2 \to 0$.

Sections IV-A and IV-C showed how to accelerate Prox-SVRG, while Section V showed how Algorithm 5 (i.e., SVR-AMA) is equivalent to Algorithm 2 (i.e., Prox-SVRG [18]) applied to the dual of Problem (6). This observation allows us to formulate the accelerated versions of SVR-AMA, knowing that their convergence can be derived from Theorems IV.1 and IV.4 applied to the dual. In this respect, Algorithm 6 describes IA-SVR-AMA, whereas Algorithm 7 describes OA-SVR-AMA.

VI. MPC Formulation For SVR-AMA

Our aim is to solve the MPC Problem (3) presented in Section II using SVR-AMA and its accelerated versions. In this respect, we rely on the decomposition provided in [22] summarized ahead.

First, we decompose Problem (3) along the length of the prediction horizon N into $N+1$ smaller subproblems, according to the *time-splitting* strategy proposed in [12]. This result is achieved thanks to the introduction of N consensus variables $z_t \in \mathbb{R}^n$ $(t = 1, \ldots, N)$ used to break up the dynamic coupling (3b). This decomposition allows us to reformulate Problem (3) as follows:

$$\min_{x,u} \ \frac{1}{2}\sum_{t=0}^{N} x_t^{(t)^\mathrm{T}} Q x_t^{(t)} + \frac{1}{2}\sum_{t=0}^{N-1} u_t^{(t)^\mathrm{T}} R u_t^{(t)} \tag{30a}$$

$$\text{s.t.: } z_{t+1} = Ax_t^{(t)} + Bu_t^{(t)}, \quad t = 0, \ldots, N-1 \tag{30b}$$

$$z_{t+1} = x_{t+1}^{(t+1)}, \qquad t = 0, \ldots, N-1 \tag{30c}$$

$$Cx_t^{(t)} + Du_t^{(t)} \leq d, \qquad t = 0, \ldots, N-1 \tag{30d}$$

$$Cx_N^{(N)} \leq d,\ x_0^{(0)} = x_{\text{init}} \tag{30e}$$

where the original dynamic coupling (3b) has been replaced by the consensus constraints (30b) and (30c). Note that we introduced the superscript t to emphasize that x_t and u_t are local variables of the subproblems obtained after the time splitting.

Remark 7: The time-splitting strategy supports more general MPC formulations, such as MPC problems with terminal cost, terminal set, known disturbances, and time-varying constraints. More details can be found in [12].

Finally, if we introduce $N+1$ additional slack variables $\sigma_t \in \mathbb{R}^p$ to remove the inequality constraints (30d) and define $\mathbb{C} := \{\sigma_t \in \mathbb{R}^p \,|\, \sigma_t \geq 0\}$, Problem (30) can be written as the sum of the following subproblems:

$$\min_{y_t} f_t(y_t) + \sum_{i=1}^{p} \delta_{\mathbb{C}}(\sigma_{t_i}) \tag{31a}$$

$$\text{s.t.: } w_t : \ z_t = H_1 y_t \tag{31b}$$

$$v_{t+1} : \ z_{t+1} = H_2 y_t \tag{31c}$$

$$\lambda_t : \ \sigma_t = d - G y_t \tag{31d}$$

where we define $y_t^\mathrm{T} := [x_t^{(t)^\mathrm{T}}\ u_t^{(t)^\mathrm{T}}]$, $f_t(y_t) := y_t^\mathrm{T}\mathcal{Q}y_t$, $\mathcal{Q} := \operatorname{diag}\{Q, R\}$, $G := [C\ D]$, $H_1 := [I_n\ 0_{n\times m}]$, $H_2 := [A\ B]$. Furthermore, for each equality constraint in Problem (31), the corresponding Lagrange multipliers have been highlighted. If we define $\mathbf{y}^\mathrm{T} = [y_0^\mathrm{T} \ldots y_N^\mathrm{T}]$, $\mathbf{z}^\mathrm{T} = [z_1^\mathrm{T} \ldots z_N^\mathrm{T}\ \sigma_0^\mathrm{T} \ldots \sigma_N^\mathrm{T}]$, $f(\mathbf{y}) = \mathbf{y}^\mathrm{T}\mathbf{Q}\mathbf{y} = \sum_{t=0}^{N} f_t(y_t)$, $\mathbf{Q} = \operatorname{diag}\{\mathcal{Q} \ldots \mathcal{Q}\}$, $g(\mathbf{z}) = \sum_{t=0}^{N} \delta_{\mathbb{C}}(\sigma_t)$

$$h_{y_0}^\mathrm{T} = [H_2^\mathrm{T} | -G^\mathrm{T}], h_y^\mathrm{T} = [H_1^\mathrm{T}\ H_2^\mathrm{T} | -G^\mathrm{T}]$$

$$h_{y_N}^\mathrm{T} = [H_1^\mathrm{T} | -[C\,0]^\mathrm{T}],\ h_{d_0}^\mathrm{T} = h_{d_N}^\mathrm{T} = [0_n | -d^\mathrm{T}]$$

$$h_d^\mathrm{T} = [0_n\ 0_n\ |\ -d^\mathrm{T}], H_\mathbf{y} = \operatorname{diag}\{h_{y_0}, h_y, \ldots, h_y, h_{y_N}\}$$

$$\mathbf{d} := \left[\, h_{d_0}^\mathrm{T}\ h_d^\mathrm{T} \ldots h_d^\mathrm{T}\ h_{d_N}^\mathrm{T} \,\right]^\mathrm{T}$$

$$H_\mathbf{z} := \left[\begin{array}{cccc|cccc} I_n & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & I_p & 0 & \ldots & 0 \\ \hline I_n & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & I_n & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & I_p & \ldots & 0 \\ \hline \vdots & & \ddots & & \vdots & & \ddots & 0 \\ \hline 0 & 0 & \ldots & I_n & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & I_p \end{array}\right].$$

Problem (3) can be rewritten as follows:

$$\text{minimize } f(\mathbf{y}) + g(\mathbf{z}) \tag{32a}$$

$$H_\mathbf{y}\mathbf{y} + H_\mathbf{z}\mathbf{z} = \mathbf{d}. \tag{32b}$$

According to the definition of $\mathcal{Q}$, $f(\mathbf{y})$ is strongly convex, has a convexity parameter $\sigma_f := \operatorname{eig}_{\min}(\mathcal{Q}) = \operatorname{eig}_{\min}(\operatorname{blockdiag}\{Q, R\}) = \sigma_{f_t}$, and a Lipschitz constant $L_f := \operatorname{eig}_{\max}(\mathcal{Q})$. In addition, $g(\mathbf{z})$ is a convex function.

For the proposed splitting, Assumptions 2 and 3 are satisfied. Concerning Assumption 1, note that $L(\nabla F) := \operatorname{eig}_{\max}(H_\mathbf{y})\ \sigma_f^{-1} = \max_t(\operatorname{eig}_{\max}(H_{y_t})\sigma_{f_t}^{-1}) = \max_t(L_t(\nabla F_t)) = L_t(\nabla F_t) \leq \sum_{t=0}^{N} L_t(\nabla F_t)$, where the last equality follows from the fact that we deal with LTI systems ($L_0 = L_1 = \ldots = L_N$). Hence, on the dual, Assumption 4 still holds and, consequently, we can use SVR-AMA to solve Problem (3).

The associated SVR-AMA algorithm to solve Problem (30) is detailed in Algorithm 8. The accelerated versions are omitted here due to space limitations, but can be easily derived from Algorithms 6 and 7, respectively. In particular, defining $\boldsymbol{\mu}^\mathrm{T} := [v_1^\mathrm{T}\ \lambda_0^\mathrm{T}\,|\,w_1^\mathrm{T} \ldots w_{N-1}^\mathrm{T}\ v_N^\mathrm{T}\ \lambda_{N-1}^\mathrm{T}\,|\,w_N^\mathrm{T}\ \lambda_N^\mathrm{T}]$, according

Algorithm 8: SVR-AMA for Problem (30).

Given $\tilde{\mu}^0$, N, $\bar{s}$, $\mathcal{I}_N := \{0, \dots, N\}$, $L^\star := (\sigma_f)^{-1}\text{eig}_{\max}(H_y)$, η, and T.
while $s \leq \bar{s}$ **do**
  0a. Set $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}^{s-1}$, $\tilde{\mathbf{v}} = \tilde{\mathbf{v}}^{s-1}$, $\tilde{\boldsymbol{\lambda}} = \tilde{\boldsymbol{\lambda}}^{s-1}$, and $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{y}}^{s-1}$.
  0b. Set $\tilde{\beta}_{\mathbf{w}} = \nabla F(\tilde{\mathbf{w}})$, $\tilde{\beta}_{\mathbf{v}} = \nabla F(\tilde{\mathbf{v}})$, and $\tilde{\beta}_{\boldsymbol{\lambda}} = \nabla F(\tilde{\boldsymbol{\lambda}})$.
  0c. Set $\mathbf{w}^0 = \tilde{\mathbf{w}}$, $\mathbf{v}^0 = \tilde{\mathbf{v}}$, and $\boldsymbol{\lambda}^0 = \tilde{\boldsymbol{\lambda}}$.
  0d. Set $\Pi := \{\pi_0, \dots, \pi_N\}$ on $\mathcal{I}_N$
  for $k = 1, \dots, T$ **do**
    1. Pick $i \in \mathcal{I}_N$ randomly according to Π.
    2a. $y_i^k = \text{argmin}_y\ f_i(y_i) + \langle w_i, H_1 y_i\rangle + \langle v_{i+1}, H_2 y_i\rangle + \langle \lambda_i, -G y_i\rangle$.
    2.b $\sigma_i^k = \mathbf{Pr}_{\mathbb{C}}(G y_i^k - d - \eta\lambda_i)$.
    2c. $z_i^k = \frac{1}{2}\left[H_1 y_i^k + H_2 y_{i-1} - \eta(w_i + v_i)\right]$.
    3a. $\beta_{w_i}^k = \tilde{\beta}_{w_i} + \frac{(y_i^k - \tilde{y}_i)^{\mathrm{T}} H_1^{\mathrm{T}}}{\pi_i}$.
    3b. $\beta_{v_i}^k = \tilde{\beta}_{v_i} + \frac{(y_{i-1} - \tilde{y}_{i-1})^{\mathrm{T}} H_2^{\mathrm{T}}}{\pi_i}$.
    3c. $\beta_{\lambda_i}^k = \tilde{\beta}_{\lambda_i} - \frac{(y_i^k - \tilde{y}_i)^{\mathrm{T}} G^{\mathrm{T}}}{\pi_i}$.
    4a. $w_i^k = w_i + \eta\left(z_i^k - \beta_{w_i}^k\right)$.
    4b. $v_i^k = v_i + \eta\left(z_i^k - \beta_{v_i}^k\right)$.
    4c. $\lambda_i^k = \lambda_i + \eta\left(\beta_{\lambda_i}^k + d - \sigma_i^k\right)$.
  end for
  5. $\tilde{\mathbf{w}}^s = \frac{1}{T}\sum_{k=1}^T \mathbf{w}^k$, $\tilde{\mathbf{v}}^s = \frac{1}{T}\sum_{k=1}^T \mathbf{v}^k$ $\tilde{\boldsymbol{\lambda}}^s = \frac{1}{T}\sum_{k=1}^T \boldsymbol{\lambda}^k$, and $\tilde{\boldsymbol{y}}^s = \frac{1}{T}\sum_{k=1}^T \boldsymbol{y}^k$.
  6. $s = s + 1$.
end while

to the partitioning of $H_{\mathbf{y}}$ and $H_{\mathbf{z}}$, $F(\boldsymbol{\mu}) = f^\star(H_{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\mu})$. Furthermore, $\nabla F(\mathbf{w})$ is the gradient of F at $\mathbf{w}$, $\nabla F(\mathbf{v})$ is the gradient of F at $\mathbf{v}$, and $\nabla F(\boldsymbol{\lambda})$ is the gradient of F at $\boldsymbol{\lambda}$. Note that the calculation of the gradient step for this particular splitting is very simple and requires the evaluation of the product $H_{\mathbf{y}}\mathbf{y}$, which can be performed efficiently by exploiting the structure of matrix $H_{\mathbf{y}}$. Finally, note that, given the structure of $F(\boldsymbol{\mu})$, probability π_t does not affect the choice of the step size η, according to Remark 2.

The following complexity upper bound on the primal sequence can be defined.

Theorem VI.1: Consider Problem (30). Let $\{\mathbf{y}^k\}$ and $\{\boldsymbol{\mu}^k\}$ be the sequence of primal and dual variables, respectively, generated by Algorithm 9. If Assumptions 1–3 are satisfied, given $\tilde{\boldsymbol{\mu}}^0 \in \text{dom}(G)$, where $G := g^\star(H_{\mathbf{z}}^{\mathrm{T}}\boldsymbol{\mu}) - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\mu}$, then, the following holds:

$$\mathbb{E}\|\tilde{\boldsymbol{y}}^s - \boldsymbol{y}^*\|^2 \leq \frac{2}{\sigma_f}(D(\boldsymbol{\mu}^*) - \mathbb{E}D(\tilde{\boldsymbol{\mu}}^0)). \tag{33}$$

Proof: The inequality can be derived by the results of [14, Th. 5.3] by noticing that the primal updates in the inner loop are the same as AMA. Then, we have to take into account for Algorithm 8 that the primal variables are stochastic variables and that we must consider their expected values. These observations combined with the results of Theorem V.1 lead to (33). ■

Remark 8: The initial value of the dual variables $\tilde{\boldsymbol{\mu}}^0$ should be a feasible starting point in order to use the results of Theorem 5.3. This can be accomplished by noticing the following. Concerning the $\tilde{\lambda}_t^0$ components of $\tilde{\boldsymbol{\mu}}^0$, they must be in $\mathbb{C}_t$. Concerning the $\tilde{w}_t^0$ and $\tilde{v}_t^0$ components of $\tilde{\boldsymbol{\mu}}^0$, by providing an initial primal solution satisfying the consensus constraints (e.g., by using the evolution of the state starting from x_{init} under the associated unconstrained LQR control law $u_t = K_{\text{LQR}}x_t$), they can be set equal to zero.

In theory, the decomposition along the length of the prediction horizon allows one to fully parallelize the solution of Problem (3), thanks to the introduction of the consensus variables. If $N + 1$ independent workers are available, the dual update of each subproblem can be assigned to its dedicated worker that exchanges information with its neighbors only at dedicated synchronization points to update the consensus variables, as detailed in [12]. If the prediction horizon, however, is larger than the number of available workers, the computation of the solution has to be partially (or fully, if only one worker is available) serialized. This scenario can be quite common for embedded legacy control systems, where the *serial* hardware architecture is formally verified and the costs to upgrade to the *parallel* one are too high. In this scenario, Algorithm 5 plays a fundamental role to compute a suboptimal solution of Problem (3).

Algorithm 5 applied to Problem (30) translates into the possibility of *asynchronous updates* of the independent subproblems. Compared to solving the subproblems in a serialized fashion (i.e., one after the other) in a synchronous framework, the asynchronous updates lead to less costly (in terms of computation time) iterations of the algorithm. In particular, assuming that only one worker is available, at each inner-loop iteration (Steps 1–4 of the algorithm), only one subproblem is randomly selected for the update. In a synchronous framework, the update of all the subproblems would have been required, which can be costly if the length of the horizon is large.

Compared to other asynchronous dual algorithms (see, e.g., [26]), Algorithm 5 allows one to *tune and adapt (online) the probability distribution* Π. This is particularly useful, for example, to give priority in the update to those subproblems whose associated dual variables vary the most between two iterations of the algorithm, as shown in the following section.

VII. Numerical Example

This section considers the linearized model (at a given trim condition) of an Airbus passenger aircraft [21] to test the proposed design. Aerospace applications offer several challenges for MPC from the computational perspective. First, these applications usually have strict real-time requirements. Second, the aircraft has different modes (some slower than others), reflected in the eigenvalues of the dynamic matrix, that lead to an ill-conditioned control problem. Finally, the problem size is relatively large and a long prediction horizon is required.

We focus on the longitudinal control of the aircraft. In this respect, the model we consider has $n = 6$ states (to describe the longitudinal dynamics) and $m = 4$ control actuators. In particular, the states associated with the longitudinal dynamics are

pitch rate (deg/s), roll rate (deg/s), ground speed (km), angle of attack (deg), pitch angle (deg), and altitude (ft). Finally, the control surfaces for the longitudinal dynamics are the four elevators on the tail of the aircraft.

The sampling time of the system is $T_s = 0.04$ s and we consider an horizon length $N = 60$. The total number of decision variables is 600. Furthermore, we have 3000 inequality constraints and 600 equality constraints. The goal of the MPC controller is to regulate the state of the system to the origin starting from a nonzero initial condition close to the saturation limits of the system.

We compared the behavior of Algorithms 5 (SVR-AMA), 6 (SVR-AMA with inner-loop acceleration), and 7 (SVR-AMA with outer-loop acceleration) with Algorithm 1 (AMA). The baseline for the comparison is the trajectory obtained using the MPC functions of MPT3 [33].

First, we are interested in showing that the possibility of tuning the probability distribution that the algorithm offers can lead to improvements in terms of performance of the MPC controller, especially when the solver runs for a limited number of iterations to reach a medium accuracy. In this respect, we consider three different probability distributions (depicted in Figs. 1 and 2): 1) uniform, 2) generalized Pareto, and 3) adaptive. The adaptive distribution is computed online by the algorithm according to the following guidelines. We initialize Π to be the Pareto distribution. Then, every T inner-loop iterations, we check, for each $t = 0, \ldots, N$ whether the following condition is verified $\|\tilde{\mu}^T - \tilde{\mu}^{T-1}\|^2 < 0.01$. If the condition is verified, $\pi_t \leftarrow 0.5\pi_t$ and the probabilities of its neighbors become $\pi_{t+1} \leftarrow \pi_{t+1} + 0.25\pi_t$ and $\pi_{t-1} \leftarrow \pi_{t-1} + 0.25\pi_t$. Fig. 2 shows how the distribution varies when running Algorithm 5.

Second, we are interested in showing the benefits that the acceleration of the inner and outer loops can bring in terms of number of iterations needed to reach a suboptimal solution of the MPC problem (4). In this respect, we run the proposed algorithms (i.e., Algorithms 5–7) and AMA for one problem instance that causes active inequality constraints at the optimum.

In the remainder of the section, we first analyze the proposed algorithms in open loop (see Section VII-A). Then, based on the achieved results, we propose a real-time implementation of the most promising algorithm in closed loop (see Section VII-B). Note that the units on the vertical axes of the plots presented ahead have been removed upon request of our industrial partners.

A. Open-Loop Analysis

Figs. 3–10 show the results obtained when testing the proposed algorithms for one problem instance. For the open-loop comparison, we fixed the total number of iterations for all the algorithms ($T\bar{s} = 15 \cdot 10^5$) and we use the same step size η. Concerning the tuning of the number of inner iterations, on one hand, we select $T = 500$ when using Algorithm 6 (according to the guidelines provided to select the number of inner-loop iterations for IA-SVR-AMA), leading to a larger number of outer-loop iterations. On the other hand, we select $T = 3000$ when using Algorithm 7 (according to the guidelines provided to select the number of inner-loop iterations for OA-SVR-AMA),

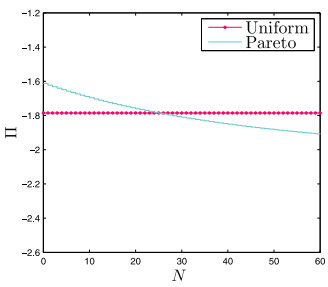

Fig. 1. Probability distributions used to test the performance of Algorithms 5, 6, and 7 plotted in $\log_{10}$ scale.

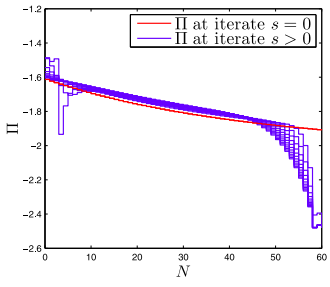

Fig. 2. Adaptive probability generated when running Algorithm 5 (SVR-AMA) plotted in $\log_{10}$ scale.

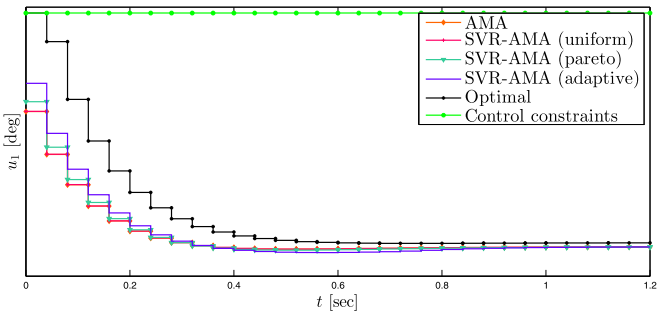

Fig. 3. Control trajectories obtained using different probability distributions Π in Algorithm 5 in open loop.

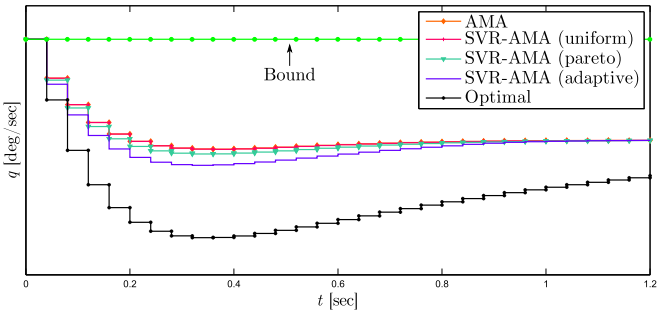

Fig. 4. Pitch-rate trajectories obtained using different probability distributions Π in Algorithm 5 in open loop.

leading to a smaller number of outer-loop iterations (that requires the update of the full gradient). Concerning the tuning of η, we selected the value according to the guidelines that derive from the theory.

Figs. 3 and 4 show the behavior of one of the actuators and one of the states (the pitch rate) when using Algorithm 5. The behavior of SVR-AMA is compared with AMA and the optimal trajectory. We notice that, within the limited number of iterations, the possibility to tune Π leads to improvements in terms of quality of the solution. In particular, notice that the uniform distribution leads to a behavior comparable to AMA, while using the Pareto and the adaptive distribution lead to improved trajectories.

Figs. 5 and 6 show the behavior of one of the actuators and one of the pitch rate when using Algorithm 6 to accelerate the inner-loop iterates. The behavior of the inner-accelerated algorithm is compared with SVR-AMA (with adaptive distribution) and AMA. We notice that the algorithm leads to some

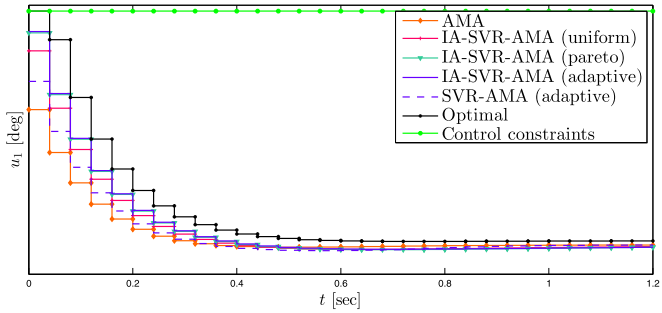

Fig. 5. Control trajectories obtained using different probability distributions Π in Algorithm 6 in open loop.

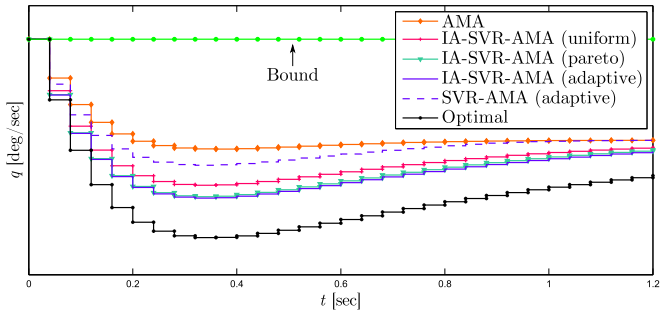

Fig. 6. Pitch-rate trajectories obtained using different probability distributions Π in Algorithm 6 in open loop.

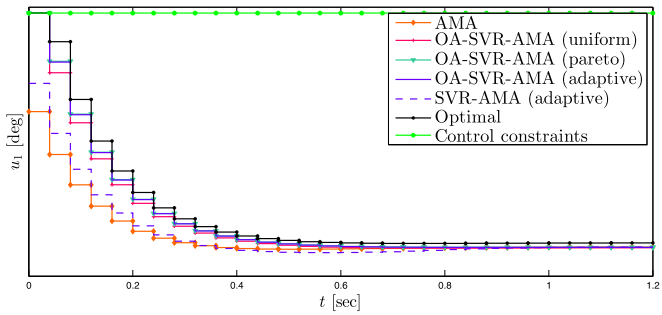

Fig. 7. Control trajectories obtained using different probability distributions Π in Algorithm 7 in open loop.

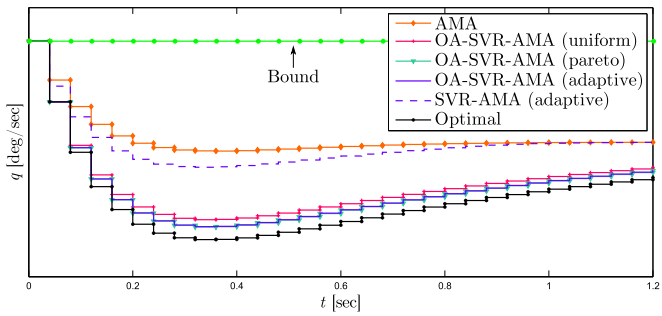

Fig. 8. Pitch-rate trajectories obtained using different probability distributions Π in Algorithm 7 in open loop.

improvements in the calculation of the optimal solution of the MPC problem. The main issue with the inner acceleration is that the algorithm requires more updates of the full gradient of F, which can be problematic when N is large. It is, however, interesting to notice that the combined use of VR (which is also an acceleration strategy from the stochastic point of view) and inner-loop acceleration (which is an acceleration strategy in a more classical sense) still lead to some benefits in the computation of the optimal solution.

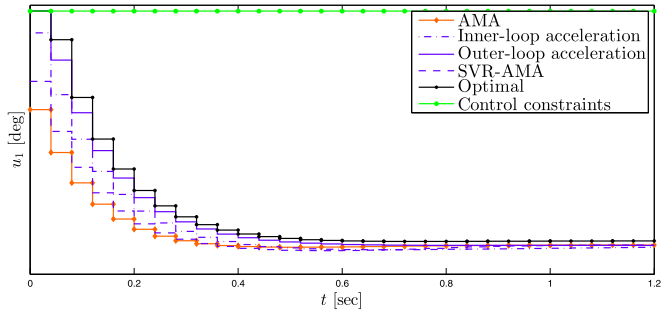

Fig. 9. Comparison of the *best* control trajectory computed by SVR-AMA, IA-SVR-AMA, and OA-SVR-AMA in open loop.

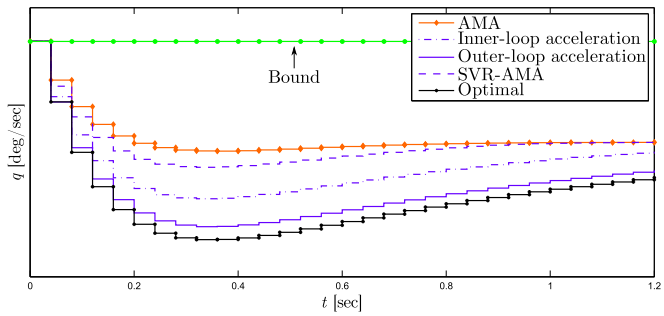

Fig. 10. Comparison of the *best* pitch-rate trajectory computed by SVR-AMA, IA-SVR-AMA, and OA-SVR-AMA in open loop.

Figs. 7 and 8 show the behavior of one of the actuators and the pitch rate when using Algorithm 7 to accelerate the outer-loop iterates. As in the previous case, the behavior of the outer-accelerated algorithm is compared with SVR-AMA (with adaptive distribution) and AMA. We notice that the solution returned by OA-SVR-AMA is closer to the optimal one. Furthermore, we can also observe the benefits of tuning Π online. In particular, note that the Pareto and the adaptive distributions clearly outperform the uniform distribution.

Figs. 9 and 10 directly compare the *best* results obtained using SVR-AMA, IA-SVR-AMA, and OA-SVR-AMA. In particular, the plots highlight the significant improvements obtained using the outer-loop acceleration. Furthermore, note that Algorithm 7 requires less full gradient updates and fully exploits the benefits of the VR scheme in the inner loop. Hence, it can be more efficient when used for applications with large N, compared to Algorithm 6.

B. Closed-Loop Analysis

Encouraged by the results obtained in open loop, we tested Algorithm 7 with adaptive distribution in closed loop, taking into account the limited computation time available for online optimization ($T_s = 0.04$ s). Based on the median of 1001 experiments, the maximum number of inner-loop and outer-loop iterations allowed within the sampling time of the system are 360 and 50, respectively. Given the limited number of iterations

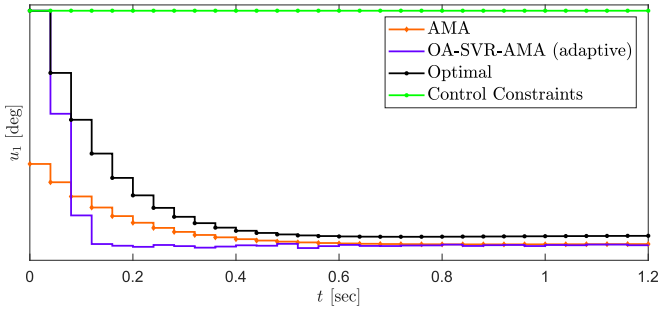

Fig. 11. Comparison of the control command obtained using AMA and OA-SVR-AMA, respectively, in closed loop.

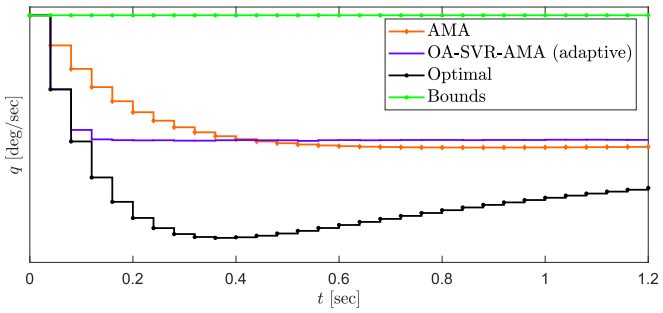

Fig. 12. Comparison of the pitch rate behavior obtained using AMA and OA-SVR-AMA, respectively, in closed loop.

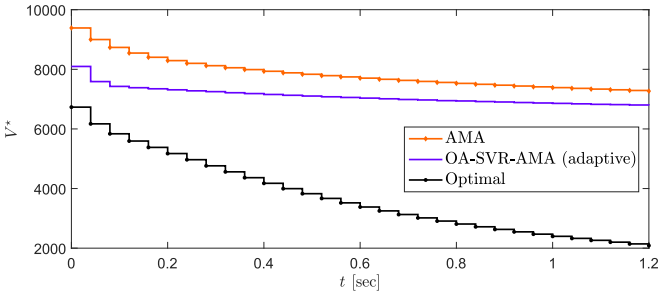

Fig. 13. Comparison of the costs obtained using AMA and OA-SVR-AMA, respectively, in closed loop.

available within the sampling time, we empirically used a larger (compared to the guidelines) step size η. We tested different initial conditions within the flight envelope. Figs. 11–13 show the behavior of the algorithm at an initial condition that causes the control command and the pitch rate to saturate. The behavior of Algorithm 7 is compared with the optimal one and with AMA (tuned to run in real time, as well). By simply looking at Figs. 11 and 12, this could lead to wrong conclusions, given that it seems that the solution obtained using the proposed solver is better than the optimal one. Fig. 13 clarifies this issue. In particular, the figure compares the optimal cost with the one obtained with OA-SVR-AMA and AMA. Fig. 13 is important to judge and interpret the performance of the different solvers in closed loop using the MPC cost as impartial metric. Given the limited number of iterations, the performance of the controller using Algorithm 7 (solid blue line) is clearly suboptimal, as Fig. 13 shows. Nevertheless, the closed-loop simulations show that the proposed algorithm, given the limited number of iterations, still improves with respect to AMA. Furthermore, as Fig. 11 shows, compared to AMA, the ability of MPC to exploit the full range of the actuators is preserved.

VIII. Conclusion

We presented an asynchronous AMA with VR (SVR-AMA) scheme and its accelerated versions suitable for MPC applications. As our numerical example showed, the proposed algorithms, compared to a state-of-the art solver (i.e., AMA), provide higher accuracy solutions within the same number of overall iterations. Furthermore, compared to other state-of-the-art asynchronous dual solvers that only perform random updates according to a uniform distribution, the proposed algorithms allow one to prioritize the update of the variables at the beginning of the prediction horizon, leading to improved behavior in closed loop, as our numerical example showed.

We analyzed the possibility of tuning the probability distribution to improve the performance of the algorithm in closed loop when the computation time for online optimization is limited. As part of our future work, we plan to further investigate the benefits that the proposed algorithm (SVR-AMA and its accelerated versions) can have in a distributed framework. In particular, we plan to investigate how to use the probability distribution as a tuning parameter to plan the communications among the agents in the network.

Appendix A
Proofs of Section IV-A

A. Proof of Theorem IV.1

According to Proposition IV.1, the following holds:

$$\mathbb{E}P(y_k) - P(y_*) \leq 4(1-\sqrt{\gamma})^k (P(\tilde{y}_{s-1}) - P(y_*)) + \frac{4}{\sigma_F}(1-\sqrt{\gamma})^k \Gamma_k^2(\tilde{y}_{s-1}) \quad (34)$$

where we used the following relationship: for any $a, b \in \mathbb{R}$ $2ab = a^2 + b^2 - (a-b)^2$. We are interested in the convergence of the outer loop of the algorithm. Hence, we sum, for $k = 1, \ldots, T$, the aforementioned inequality

$$\sum_{k=1}^{T} [\mathbb{E}P(y_k) - P(y_*)] \leq \sum_{k=1}^{T} \left[4(1-\sqrt{\gamma})^k (P(\tilde{y}_{s-1}) - P(y_*))\right] + \frac{4}{\sigma_F} \sum_{k=1}^{T} (1-\sqrt{\gamma})^k \Gamma_k^2(\tilde{y}_{s-1}).$$

Then, the following holds:

$$\sum_{k=1}^{T} \mathbb{E}P(y_k) - TP(y_*) \leq \sum_{k=1}^{T} \left[4(1-\sqrt{\gamma})^k\right] [P(\tilde{y}_{s-1}) - P(y_*)] + \frac{4}{\sigma_F} \sum_{k=1}^{T} (1-\sqrt{\gamma})^k \Gamma_k^2(\tilde{y}_{s-1}).$$

Using $\sum \mathbb{E}[\cdot] = \mathbb{E}\sum[\cdot]$ and dividing by $T > 0$, the following holds:

$$\mathbb{E}\left(\frac{1}{T}\sum_{k=1}^{T} P(y_k)\right) - P(y_*) \leq \frac{4}{T}\sum_{k=1}^{T}\left[(1-\sqrt{\gamma})^k\right]\left[P(\tilde{y}_{s-1}) - P(y_*)\right] + \frac{4}{T\sigma_F}\sum_{k=1}^{T}(1-\sqrt{\gamma})^k\Gamma_k^2(\tilde{y}_{s-1}).$$

Exploiting the convexity of P and the fact that $1-\sqrt{\gamma} < 1$, we have

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq \frac{4}{T\sqrt{\gamma}}\left((1-\sqrt{\gamma}) - (1-\sqrt{\gamma})^{T+1}\right)\left[P(\tilde{y}_{s-1}) - P(y_*)\right] + \frac{4}{T\sigma_F}\sum_{k=1}^{T}(1-\sqrt{\gamma})^k\Gamma_k^2(\tilde{y}_{s-1}).$$

The following holds:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq \rho_{\text{IA,I-PGM}}\left[P(\tilde{y}_{s-1}) - P(y_*)\right] + \frac{4}{T\sigma_F}\Gamma(\tilde{y}_{s-1}).$$

Applying recursively for $s = 1, 2, \ldots, \bar{s}$

$$\mathbb{E}P(\tilde{y}_{\bar{s}}) - P(y_*) \leq \rho^{\bar{s}}_{\text{IA,I-PGM}}\left[P(\tilde{y}_0) - P(y_*)\right] + \frac{4}{T\sigma_F}\sum_{s=0}^{\bar{s}-1}\rho^{s}_{\text{IA,I-PGM}}\Gamma(\tilde{y}_s)$$

which proves the theorem.

Appendix B
Proofs of Section IV-C

A. Proof of Corollary IV.2

The proof follows the same steps of the one of [18, Corollary 3.5] but takes into account that we are considering β_s instead of β_k. In particular, the following holds:

$$\mathbb{E}\|e_s\|^2 = \mathbb{E}\left\|\frac{1}{T}\sum_{k=1}^{T}(\beta_k - \nabla F(y_{k-1}))\right\|^2 \leq \frac{1}{T^2}\mathbb{E}\sum_{k=1}^{T}\|\beta_k - \nabla F(y_{k-1})\|^2 \leq \frac{1}{T^2}\mathbb{E}\sum_{k=1}^{T}\|v_k\|^2$$

where $v_k := \frac{\nabla F_{i_k}(y_{k-1}) - \nabla F_{i_k}(\tilde{y})}{\pi_{i_k}} + \nabla F(\tilde{y}) - \nabla F(y_{k-1})$. Using the fact that $\sum_i \mathbb{E}[x_i] = \mathbb{E}\sum_i x_i$ and $\mathbb{E}\|x - \mathbb{E}x\|^2 = \mathbb{E}\|x\|^2 - \|\mathbb{E}x\|^2$ according to [18], the following holds:

$$\begin{aligned}
&\mathbb{E}\|e_s\|^2 \\
&\leq \frac{1}{T^2}\mathbb{E}\sum_{k=1}^{T}\|v_k\|^2 = \frac{1}{T^2}\sum_{k=1}^{T}\mathbb{E}\left\|\frac{\nabla F_{i_k}(y_{k-1}) - \nabla F_{i_k}(\tilde{y})}{\pi_{i_k}}\right\|^2 \\
&\quad - \frac{1}{T^2}\sum_{k=1}^{T}\|\nabla F(y_{k-1}) - \nabla F(\tilde{y})\|^2 \\
&\leq \frac{1}{T^2}\sum_{k=1}^{T}\sum_{i=1}^{N}\frac{1}{N\pi_i}\|\nabla F_i(y_{k-1}) - \nabla F_i(\tilde{y})\|^2 \\
&\leq \frac{1}{T^2}\sum_{k=1}^{T}\sum_{i=1}^{N}\frac{2L_i}{N\pi_i}\left(P_i(y_{k-1}) - P_i(y_*) + P_i(\tilde{y}) - P_i(y_*)\right) \\
&\leq \frac{2L_\Pi}{T^2}\sum_{k=1}^{T}\left(P(y_{k-1}) - P(y_*) + P(\tilde{y}) - P(y_*)\right) \\
&\leq \frac{2L_\Pi}{T}\left(P(y_{s-1,\text{avg}}) - P(y_*) + P(\tilde{y}) - P(y_*)\right).
\end{aligned}$$

B. Proof of Theorem IV.4

In the following, we provide a proof for the convergence of Algorithm 4.

Given that the inner loop of Algorithm 4 is not affected by the acceleration, the following result from [18, Theorem III.1] holds:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq \rho\left(P(\hat{y}_{s-1}) - P(y_*)\right). \tag{35}$$

Consider for simplicity $\alpha := \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}$. Then, define the following quantity (according to [20, Sec. 6.4]):

$$v_s := \left(1 - \frac{1}{\sqrt{\gamma}}\right)\tilde{y}_{s-1} + \frac{1}{\sqrt{\gamma}}\tilde{y}_s \tag{36}$$

and

$$\theta = \frac{\sqrt{\gamma} - \gamma}{1 - \gamma} < 1. \tag{37}$$

Hence, the following holds:

$$\hat{y}_s = (1-\theta)\tilde{y}_s + \theta v_s. \tag{38}$$

Given that $\sqrt{\gamma} < 1$, by using the convexity of P and the definition of v_s in (36), the following holds:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq (1-\sqrt{\gamma})(P(\tilde{y}_{s-1}) - P(y_*)) + \sqrt{\gamma}(\mathbb{E}P(v_s) - P(y_*)). \tag{39}$$

At the same time, by using (35) and the definition of $\hat{y}_s$ in (38), the following holds:

$$\mathbb{E}P(\tilde{y}_{s+1}) - P(y_*) \leq \rho(1-\theta)(P(\tilde{y}_s) - P(y_*)) + \rho\theta(P(v_s) - P(y_*)). \tag{40}$$

From (40), the following holds:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \geq \frac{\mathbb{E}P(\tilde{y}_{s+1}) - P(y_*) - \rho\theta(\mathbb{E}P(v_s) - P(y_*))}{\rho(1-\theta)}. \quad (41)$$

At the same time, using (39), the following holds:

$$\begin{aligned}&(\rho(1-\theta))(1-\sqrt{\gamma})(P(\tilde{y}_{s-1}) - P(y_*)) \\ &+ (\rho(1-\theta))\sqrt{\gamma}(\mathbb{E}P(v_s) - P(y_*)) \\ &\geq \rho(1-\theta)(\mathbb{E}P(\tilde{y}_s) - P(y_*)) \\ &\geq (\mathbb{E}P(\tilde{y}_{s+1}) - P(y_*)) - \rho\theta(\mathbb{E}P(v_s) - P(y_*)).\end{aligned}$$

Then, the following holds:

$$\begin{aligned}&(\rho(1-\theta))(1-\sqrt{\gamma})(P(\tilde{y}_{s-1}) - P(y_*)) \\ &+ ((\rho(1-\theta))\sqrt{\gamma} + \rho\theta)(\mathbb{E}P(v_s) - P(y_*)) \\ &\geq \rho(1-\theta)(\mathbb{E}P(\tilde{y}_s) - P(y_*)) + \rho\theta(\mathbb{E}P(v_s) - P(y_*)) \\ &\geq (\mathbb{E}P(\tilde{y}_{s+1}) - P(y_*)). \quad (42)\end{aligned}$$

Given the fact that $\rho\theta(\mathbb{E}P(v_s) - P(y_*)) \leq ((\rho(1-\theta))\sqrt{\gamma} + \rho\theta)(\mathbb{E}P(v_s) - P(y_*))$ and, at the same time, (42) holds, the following must hold for all $s > 0$:

$$(\mathbb{E}P(\tilde{y}_s) - P(y_*)) \leq (1-\sqrt{\gamma})(P(\tilde{y}_{s-1}) - P(y_*)). \quad (43)$$

Hence, if we apply the aforementioned inequality recursively, the following holds:

$$\mathbb{E}P(\tilde{y}_s) - P(y_*) \leq (1-\sqrt{\gamma})^s(P(\tilde{y}_0) - P(y_*)). \quad (44)$$

APPENDIX C
PROOFS OF SECTION V

A. Proof of Theorem V.1

If the assumptions of the theorem are satisfied, exploiting a similar argument to the one in [16, Th. 4] for AMA, we can show that Algorithm 5 is equivalent to apply Algorithm 2 to Problem (5), that is, the dual of Problem (4). In particular, according to the proof of Theorem 4, using the optimality conditions that derive from Steps 2a and 2b, we can show that Steps 2 and 4 are equivalent to computing $\text{prox}_{\eta G}(\mu_k - \eta\beta_k)$, which is equivalent to the proximal Step 3 of Algorithm 2. The definition of β_k in Step 3 of Algorithm 5 follows the same logic of Prox-SVRG on the dual. In particular, by defining $\tilde{\beta} = \nabla F(\tilde{\mu})$, we replace the calculation of the full gradient of $F(\mu)$ with the following:

$$\beta_k = \tilde{\beta} + \frac{\nabla F_i(\mu_{k-1}) - \nabla F_i(\tilde{\mu})}{\pi_i} \quad (45a)$$

$$= \tilde{\beta} + \frac{(y_k - \tilde{y})^{\mathrm{T}} H_y^{(i)^{\mathrm{T}}}}{\pi_i} \quad (45b)$$

where $H_y^{(i)}$ indicates the components of H_y associated with $\nabla F_i(\mu)$. Hence, we can conclude that Algorithm 5 is equivalent to Algorithm 2 applied to the dual of Problem (4). Then, we can use Lemmas III.1 and III.2 to derive the upper bound on η and to satisfy the assumptions of Theorem III.1. Finally, by using Lemma V.1, we can derive the results of the theorem by following the proof of Theorem III.1 (applied to the dual).

B. Proof of Corollary V.1

We first prove (28) by using the same logic in the proof of [18, Corollary 3.5]. In particular, the following holds:

$$\begin{aligned}\mathbb{E}\beta_k &= \mathbb{E}\nabla F(\tilde{\mu}) + \mathbb{E}\frac{\nabla F_i(\mu_{k-1})}{\pi_i} - \mathbb{E}\frac{\nabla F_i(\tilde{\mu})}{\pi_i} \\ &= \nabla F(\tilde{\mu}) + \sum_{i=0}^{N}\frac{\pi_i \nabla F_i(\mu_{k-1})}{\pi_i} - \sum_{i=0}^{N}\frac{\pi_i \nabla F_i(\tilde{\mu})}{\pi_i} \\ &= \nabla F(\tilde{\mu}) + \sum_{i=0}^{N}\nabla F_i(\mu_{k-1}) - \sum_{i=0}^{N}\nabla F_i(\tilde{\mu}) \\ &= \nabla F(\tilde{\mu}) + \nabla F(\mu_{k-1}) - \nabla F(\tilde{\mu}) \\ &= \nabla F(\mu_{k-1}).\end{aligned}$$

Then, the proof of (29) follows exactly the proof in [18] with the only difference that in the last step, we use the definition of $\nabla F(\mu)$. Hence, the following holds:

$$\begin{aligned}&\mathbb{E}\|\beta_k - \nabla F(\mu_{k-1})\|^2 \\ &\leq 2\sum_{i=0}^{N}\frac{1}{\pi_i}\|\nabla F_i(\mu_{k-1}) - \nabla F_i(\mu_*)\|^2 \\ &\quad + 2\sum_{i=0}^{N}\frac{1}{\pi_i}\|\nabla F_i(\tilde{\mu}) - \nabla F_i(\mu_*)\|^2 \\ &\Downarrow \text{ using the definition of } \nabla F(\mu) \\ &= \sum_{i=0}^{N}\frac{2}{\pi_i}\|H_y^{(i)}(y_k - y_*)\|^2 + \sum_{i=0}^{N}\frac{2}{\pi_i}\|H_y^{(i)}(\tilde{y} - y_*)\|^2 \\ &\Downarrow \text{ using the Cauchy–Schwarz inequality} \\ &\leq 2\sum_{i=0}^{N}\frac{\|H_y^{(i)}\|^2}{\pi_i}(\|y_k - y_*\|^2 + \|\tilde{y} - y_*\|^2) \\ &\Downarrow \text{ using } \|y - y_*\|^2 \leq \frac{2}{\sigma_F}(P(y) - P(y_*)) \\ &\leq 4\sum_{i=0}^{N}\frac{\|H_y^{(i)}\|^2}{\sigma_F \pi_i}[P(y_k) - P(y_*) + P(\tilde{y}) - P(y_*)] \\ &\leq 4L_\pi^\star[P(y_k) - P(y_*) + P(\tilde{y}) - P(y_*)]\end{aligned}$$

where $L_\Pi^\star := \max_i(\text{eig}_{\max}(H_y^{(i)})/(\sigma_F \pi_i))$.

REFERENCES

[1] T. Keviczky and G. J. Balas, "Receding horizon control of an F-16 aircraft: A comparative study," *Control Eng. Pract.*, vol. 14, no. 9, pp. 1023–1033, 2006, doi: 10.1016/j.conengprac.2005.06.003.

[2] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Proc. IEEE Int. Conf. Control Appl.*, Oct. 2012, pp. 295–302, doi: 10.1109/CCA.2012.6402735.
[3] L. Ferranti and T. Keviczky, "Operator-splitting and gradient methods for real-time predictive flight control design," *J. Guid., Control, Dyn.*, vol. 40, no. 2, pp. 265–277, 2016, doi: 10.2514/1.G000288.
[4] B. Stellato, T. Geyer, and P. J. Goulart, "High-speed finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 32, no. 5, pp. 4007–4020, May 2017, doi: 10.1109/TPEL.2016.2584678.
[5] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System*. Berlin, Germany: Springer, 2017, pp. 3–39, doi: 10.1007/978-3-319-54927-9_1.
[6] S. Vazquez *et al.*, "Model predictive control: A review of its applications in power electronics," *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, Mar. 2014.
[7] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using robust model predictive control," in *Proc. 2016 IEEE Intell. Veh. Symp.*, Jun. 2016, pp. 233–238, doi: 10.1109/IVS.2016.7535391.
[8] J. Maciejowski, E. Hartley, and K. Siaulys, "A longitudinal flight control law to accommodate sensor loss in the RECONFIGURE benchmark," *Annu. Rev. Control*, vol. 42, pp. 212–223, 2016, doi: 10.1016/j.arcontrol.2016.07.001.
[9] L. Ferranti, Y. Wan, and T. Keviczky, "Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults," *Int. J. Robust Nonlinear Control*, 2018, pp. 1–17, doi: 10.1002/rnc.4063.
[10] S. Richter, C. Jones, and M. Morari, "Real-time input-constrained MPC using fast gradient methods," in *Proc. 48th IEEE CDC Held Jointly With 28th Chin. Control Conf.*, 2009, pp. 7387–7393, doi: 10.1109/CDC.2009.5400619.
[11] 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, Jul. 2014, doi: 10.1109/TAC.2013.2275667.
[12] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems," in *Proc. IEEE Eur. Control Conf.*, 2013, pp. 3089–3094.
[13] L. Ferranti and T. Keviczky, "A parallel dual fast gradient method for MPC applications," in *Proc. 54th IEEE Control Conf. Decis. Control*, Dec. 2015, pp. 2406–2413, doi: 10.1109/CDC.2015.7402568.
[14] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Fast alternating minimization algorithm for model predictive control," in *Proc. 19th IFAC World Congr.*, 2014, pp. 11980–11986, doi: 10.3182/20140824-6-ZA-1003.01432.
[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011, doi: 10.1561/2200000016.
[16] 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, doi: 10.1137/120896219.
[17] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *in Proc. Adv. Neural Inf. Process. Syst.*, 2013, vol. 26 pp. 315–323.
[18] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM J. Optim.*, vol. 24, no. 4, pp. 2057–2075, 2014, doi: 10.1137/140961791.
[19] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim.*, vol. 29, no. 1, pp. 119–138, 1991, doi: 10.1137/0329006.
[20] M. Schmidt, N. L. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1458–1466.
[21] P. Goupil, J. Boada-Bauxell, A. Marcos, P. Rosa, M. Kerr, and L. Dalbies, "An overview of the FP7 RECONFIGURE project: Industrial, scientific and technological objectives," in *Proc. 9th IFAC Symp. SAFEPROCESS*, 2015, pp. 976–981.
[22] L. Ferranti, Y. Pu, C. N. Jones, and T. Keviczky, "Asynchronous splitting design for model predictive control," in *Proc. 55th IEEE Conf. Decis. Control*, 2016, pp. 2345–2350.
[23] A. Nitanda, "Stochastic proximal gradient descent with acceleration techniques," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1574–1582.
[24] S. Shalev-Shwartz, and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res.*, vol. 14, pp. 567–599, 2013.
[25] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Math. Program*, vol. 155, pp. 105–145, 2016.
[26] I. Notarnicola and G. Notarstefano, "Randomized dual proximal gradient for large-scale distributed optimization," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 712–717.
[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[28] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002, doi: 10.1016/S0005-1098(01)00174-1.
[29] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*. Berlin, Germany: Springer-Verlag, 1993, doi: 10.1007/978-3-662-02796-7.
[30] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009, doi: 10.1137/080716542.
[31] J. D. Y. Singer, "Efficient learning using forward-backward splitting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 495–503.
[32] T. Z. J. Langford and L. Li, "Sparse online learning via truncated gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 905–912.
[33] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf.*, Zürich, Switzerland, Jul. 17–19, 2013, pp. 502–510. [Online]. Available: http://control.ee.ethz.ch/~mpt

Laura Ferranti received the M.Sc. degree in control engineering from the University of Rome "Tor Vergata," Rome, Italy, in 2012, and the Ph.D. degree in control engineering from the Delft University of Technology, Delft, The Netherlands, in 2017.

She is currently a Postdoctoral Researcher with the Intelligent Vehicles Section, Cognitive Robotics Department, Delft University of Technology. Her research interests include optimization and optimal control, model predictive control, embedded optimization-based control with application in flight control, maritime transportation, robotics, and automotive.

Ye Pu received the B.S. degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2008, the M.S. degree from the Department of Electrical Engineering and Computer Sciences, Technical University Berlin, Berlin, Germany, in 2011, and the Ph.D. degree in control engineering from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2016.

She is currently a Postdoctoral Scholar with the Hybrid Systems Laboratory EECS, University of California at Berkeley, Berkley, CA, USA. Her research interests include fast and distributed predictive control, optimization, and distributed algorithms.

Colin N. Jones (M'09) received the Bachelor's degree in electrical engineering and the Master's degree in mathematics from the University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree in control engineering from the University of Cambridge, Cambridge, U.K., in 2005.

He was a Senior Researcher with the Automatic Control Laboratory of the Swiss Federal Institute of Technology Zurich until 2010. He is currently an Associate Professor with the Automatic Control Laboratory at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. His current research interests include high-speed predictive control and optimization as well as green energy generation, distribution, and management.

Tamás Keviczky received the M.Sc. degree in electrical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2001, and the Ph.D. degree in control engineering from the Control Science and Dynamical Systems Center, University of Minnesota, Minneapolis, MN, USA, in 2005.

He was a Postdoctoral Scholar in control and dynamical systems with the California Institute of Technology, Pasadena. He is currently an Associate Professor with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. His research interests include distributed optimization and optimal control, model predictive control, embedded optimization-based control, and estimation of large-scale systems with applications in aerospace, automotive, mobile robotics, industrial processes, and infrastructure systems, such as water, heat, and power networks.

Dr. Keviczky was a co-recipient of the AACC O. Hugo Schuck Best Paper Award for Practice in 2005. He served as an Associate Editor for Automatica between 2011 and 2017.