

Asynchronous Splitting Design for Model Predictive Control*

L. Ferranti¹, Y. Pu², C. N. Jones², and T. Keviczky¹

Abstract—This paper focuses on the design of an asynchronous dual solver suitable for embedded 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 stochastic proximal gradient methods, and on the alternating minimization algorithm (AMA). The resultant algorithm, a stochastic AMA with VR, 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 to tune the probability of the asynchronous updates to improve the quality of the estimates. We apply the proposed algorithm to a specific class of splitting methods, i.e., the decomposition along the length of the prediction horizon, and provide preliminary numerical results on a practical application, the longitudinal control of an Airbus passenger aircraft.

I. INTRODUCTION

Model Predictive Control (MPC) applications to systems with fast dynamics are still relatively limited [1], [2]. Applications in fields such as automotive and aerospace have to deal very often with embedded legacy systems. These systems usually run on certified (for safety purposes) hardware architectures with limited availability, for example, of parallel computation units and 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).

Growing attention has been recently dedicated to the design of *simple* first-order solvers for MPC ([3], [4], [5], [6], [7]). These solvers are relatively easy to certify (in terms of level of suboptimality of the solution), use only simple algebraic operations, and require little memory. In [5]–[7], operator-splitting methods, such as the alternating minimization method of multipliers (ADMM) [8] and the fast alternating minimization algorithm (FAMA) [9], have been used to exploit the MPC problem structure and speed-up the computation of the solution. These algorithms most of the time require frequent exchanges of information at given synchronization points. To reduce the bottlenecks at the synchronization points, a solver that can offer more *flexibility*

in the way the solutions are computed (for example, by allowing asynchronous updates) would be attractive. In this work, we are interested in extending the use of splitting methods, such as AMA, to the asynchronous framework.

Contribution. The contribution of the paper is threefold. First, we propose a novel algorithm, a stochastic alternating minimization algorithm 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 scheme proposed in [10] for the proximal stochastic gradient method with the alternating minimization algorithm [11]. 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 quality of the estimates, as our numerical results show. Finally, the algorithm relies on simple algebraic operations, an appealing quality for embedded MPC applications.

Second, we show how we can use SVR-AMA for a specific splitting technique, i.e., the decomposition along the length of the prediction horizon (or *time splitting* [5]).

Third, we present simulation results on a practical aerospace application, i.e., the longitudinal control of an Airbus passenger aircraft [12]. The results show that the proposed method is 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.

Related Work. SVR-AMA derives from the application to the dual problem of the proximal stochastic gradient method with variance reduction (Prox-SVRG) proposed in [10].

The investigation of asynchronous dual algorithms for MPC is gaining more attention recently. In [13], for example, an asynchronous dual algorithm is proposed. Compared to [13], 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.

The idea of the time splitting has been previously proposed in [5]. Their work relies on a synchronous ADMM algorithm. In this context, we reformulate the approach for AMA to exploit SVR-AMA.

Outline. The paper is structured as follows. Section II introduces the MPC problem formulation. Section III sum-

*This research is supported by the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement n. AAT-2012-RTD-2314544 (RECONFIGURE), by the TU Delft Space Institute, by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n. 607957 (TEMPO) and by the Swiss National Science Foundation under Grand P2ELP2_165155.

¹L. Ferranti and T. Keviczky are with the Delft Center for Systems and Control, Delft University of Technology, Delft, 2628 CD, The Netherlands, {l.ferranti,t.keviczky}@tudelft.nl

²Y. Pu and C. N. Jones are with the Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, CH-1015, Switzerland, {y.pu,colin.jones}@epfl.ch

marizes AMA and Prox-SVRG. Section IV describes SVR-AMA and show convergence results. Section V shows how to reformulate the proposed MPC problem for SVR-AMA using the time splitting. Section VI presents preliminary numerical results using an aerospace example. Finally, Section VII concludes the paper.

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, $\text{Pr}_{\mathbb{C}}(u)$ is the projection of u onto \mathbb{C} . Let $f: \mathcal{D} \rightarrow \mathcal{C}$ be a function. Then, $f^*(y) = \sup_x (y^T x - f(x))$ and $\nabla f(x)$ are the conjugate function and the gradient of $f(x)$, respectively. Furthermore, $\mathcal{I}_{\mathbb{C}}(\sigma)$ 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, $\text{eig}_{\max}(A)$ and $\text{eig}_{\min}(A)$ are the largest and the smallest (modulus) eigenvalues of $A^T A$. $P \in \mathbb{S}_+^{n \times n}$ 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 [14].

II. PROBLEM FORMULATION

Consider the discrete linear time-invariant (LTI) system described by the following equation:

$$x(k+1) = Ax(k) + Bu(k), \quad k = 0, 1, 2, \dots \quad (1)$$

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

$$Cx(k) + Du(k) \leq d, \quad (2)$$

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

$$\min_{x, u} \frac{1}{2} \sum_{t=0}^N (x_t^T Q x_t + u_t^T R u_t) \quad (3a)$$

$$\text{s.t.}: x_{t+1} = Ax_t + Bu_t \quad t = 0, \dots, N-1 \quad (3b)$$

$$Cx_t + Du_t \leq d \quad t = 0, \dots, N \quad (3c)$$

$$x_0 = x_{\text{init}}, \quad (3d)$$

where x_t and u_t represent the t -step-ahead state and control predictions, respectively, N indicates the length of the prediction horizon, $Q \in \mathbb{S}_+^{n \times n}$, $R \in \mathbb{S}_+^{m \times m}$, and x_{init} is the initial (measured) state vector. The MPC law implemented in closed loop is given by the first element of the optimal control sequence obtained by solving Problem (3), i.e., $u_{\text{MPC}} = u_0^*$.

Our goal is to solve Problem (3) in an embedded environment. In particular, we assume that explicit MPC [15] cannot be used due to the problem size and that the computational resources are limited, i.e., parallel architectures are not available, memory resources are limited, and only simple algebraic operations are supported. With this framework in mind, in the following, we focus on the design of a simple solver for Problem (3) that relies on

Algorithm 1 AMA [11].

Given μ^0 , T , and $\tau < \sigma_f / \text{eig}_{\max}(H_y)$.
while $k = 1, \dots, T$ **do**
 1a. $y^k = \text{argmin}_y f(y) + \langle \mu^{k-1}, -H_y y \rangle$.
 1b. $z^k = \text{argmin}_z g(z) + \langle \mu^{k-1}, -H_z z \rangle + \frac{\tau}{2} \|d - H_y y^k - H_z z\|^2$.
 2. $\mu^k = \mu^{k-1} + \tau(d - H_y y^k - H_z z^k)$
end while

operator-splitting methods (which, for example, usually rely on parallel hardware architectures) and asynchronicity (which allows one to perform updates of a randomly selected subset of variables to reduce the computational effort). The next section introduces the techniques we rely on to solve Problem (3), i.e., AMA proposed by [11] and the proximal stochastic gradient descent method with variance reduction (Prox-SVRG) proposed by [10].

III. PRELIMINARIES

AMA and Prox-SVRG are the main techniques we rely on for our design. In the following, Section III-A briefly describes AMA developed by [11], while Section III-B describes Prox-SVRG [10].

A. Alternating Minimization Algorithm

Consider the following problem:

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

$$\text{subject to } H_y y + H_z z = d, \quad (4b)$$

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

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

Assumption 2. f_t has a Lipschitz continuous gradient with modulus L_{f_t} ($t = 0, \dots, N$) and $\text{eig}_{\min}(H_y) > 0$.

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

Furthermore, recall the following properties of the conjugate function f^* :

Lemma 1 (Thm. 4.2.1 [16]). *If f is strongly convex with convexity parameter σ_f , then f^* has a Lipschitz continuous gradient with modulus $L(\nabla f^*) = \sigma_f^{-1}$.*

Lemma 2 (Thm. 4.2.2 [16]). *If f is convex and has a Lipschitz continuous gradient with modulus L_f , then f^* is strongly convex with convexity parameter L_f^{-1} .*

A state-of-the-art algorithm to solve Problem (4) is AMA [11]. AMA operates as a proximal gradient algorithm (such as, ISTA [17]) on the dual of Problem (4). Specifically, given the dual of Problem (4) (under the assumptions above), described as follows:

$$\text{maximize}_{\mu \in \mathbb{R}^{n_\mu}} D(\mu) \{ := -F(\mu) - G(\mu) \}, \quad (5)$$

where $F(\mu) := \sum_{t=0}^N f_t^*(H_{y_t}^T \mu)$ and $G(\mu) := g^*(H_z^T \mu) - d^T \mu$, the following holds:

Lemma 3. *If Assumptions 1-3 are satisfied, $F(\mu)$ is strongly convex with Lipschitz continuous gradient characterized by Lipschitz constant $L(\nabla F) := \text{eig}_{\max}(H_y) \sigma_f^{-1}$. Furthermore $G(\mu)$ is convex.*

Proof. We can use Lemmas 1 and 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 $d^T \mu$ is a linear function. \square

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 that can lead to bottlenecks in the computation of the problem solution). Hence, it would be better to have some flexibility on the update strategy. Motivated by this observation, the following section introduces Prox-SVRG used to derive our proposed asynchronous AMA, as described in Section IV.

B. Stochastic Gradient Method with Variance Reduction

Consider the following primal problem:

$$\underset{y \in \mathbb{R}^{n_y}}{\text{minimize}} \ P(y) \ \{ := F(y) + G(y) \}, \quad (6)$$

where $F(y) := \sum_{t=0}^N F_t(y)$ 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:

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

The main idea behind Prox-SVRG [10] is to eliminate the dependency of the number of iterations (typical of stochastic gradient methods) in the definition of the step size and reduce the burden in the computation of $\nabla F(y)$ (typical of classical gradient methods). As pointed out in [10], proximal stochastic gradient methods (such as, [18], [19]) suffer of 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 below and described in Algorithm 2.

Algorithm 2 Prox-SVRG [10].

Given \tilde{y}^0 , N , $\mathcal{I}_N := \{0, \dots, N\}$, τ , 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. $y^k = \text{prox}_{\tau G}(y^{k-1} - \tau \beta^k)$

end for

4. $\tilde{y}^s = \sum_{k=1}^T y^k$.

end while

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 $\mathbb{E} \|\nabla F_i(y^{k-1}) - \nabla F(y^{k-1})\|^2$ ($i \in \mathcal{I}_N$), where $\nabla F_i(y^{k-1})$ is used as update direction (refer to [10] 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, i.e., 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 1 (Thm. 3.1 in [10]). *Suppose Assumptions 4 and 5 hold. Let $y^* = \underset{y}{\text{argmin}} P(y)$ and $L_\Pi := \max_t L_t / \pi_t$. Assume that $0 < \tau < 1/(4L_\Pi)$ and T is sufficiently large so that:*

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

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

$$\mathbb{E} P(\tilde{y}^{\bar{s}}) - P(y^*) \leq \rho^{\bar{s}} [P(\tilde{y}^0) - P(y^*)]. \quad (9)$$

Remark 1. The dependency on the probability π_t in the choice of the step size τ can be problematic when using probability distributions with $\pi_t \rightarrow 0$ or when $N \rightarrow \infty$ (e.g., the constrained infinite horizon LQR). Nevertheless, this dependency can be removed in the special case in which $F(y) = \sum_{t=1}^N F_t(y_t)$, i.e., when the cost is separable in y_t . From the MPC perspective, this is very often the case when the dual formulation is used, as it is shown in Section V for the decomposition along the length of the prediction horizon. Hence, this observation, when the algorithm is used in the dual framework, can be very beneficial to improve the choice

of the step size and the quality of the MPC solution.

IV. STOCHASTIC AMA WITH VARIANCE REDUCTION

Our goal is to solve Problem (4) in an asynchronous fashion, i.e. by allowing updates of a randomly selected subset of the 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, i.e., we apply Algorithm 2 to the dual of Problem (4). The resultant algorithm (SVR-AMA) is described by Algorithm 3.

Theorem 2. *Suppose Assumptions 1-3 hold. Let $\mu^* = \operatorname{argmax}_{\mu} D(\mu)$, where $D(\mu)$ is the dual cost defined in (5). In addition, let $L_{\Pi}^* := \max_t (\pi_t \sigma_f)^{-1} \operatorname{eig}_{\min}(H_y)$, $\pi_t \in \Pi$. Assume that $0 < \tau < 1/(4L_{\Pi}^*)$ and $T \geq 1$ such that:*

$$\rho := \frac{L_f}{\tau T(1 - 4\tau L_{\Pi}^*)} + \frac{4\tau L_{\Pi}^*(T + 1)}{T(1 - 4\tau L_{\Pi}^*)} < 1. \quad (10)$$

Then, for $\bar{s} > 0$, Algorithm 3 has geometric convergence in expectation:

$$D(\mu^*) - \mathbb{E}D(\bar{\mu}^{\bar{s}}) \leq \rho^{\bar{s}} [D(\mu^*) - D(\bar{\mu}^0)]. \quad (11)$$

Proof. If the assumptions of the theorem are satisfied, exploiting a similar argument to the one in Theorem 1 in [9] for AMA, we can show that Algorithm 3 is equivalent to applying Algorithm 2 to Problem (5). Consequently, by using Lemmas 1 and 2, the results follows from Theorem 1. \square

Remark 2. Note that an acceleration step such as the one used in FAMA [9] can be added to improve the convergence of the algorithm. The proof of convergence for the acceleration is part of our future work, but note that the simulation results in Section VI rely on the acceleration step of FAMA, showing that, in practice, it can be applied to the proposed algorithm. In addition, note that an acceleration strategy for Prox-SVRG [10] has been recently proposed in [20] and can be extended to the dual framework to accelerate Algorithm 3.

V. ASYNCHRONOUS MPC

Our aim is to solve the MPC Problem (3) presented in Section II using Algorithm 3. Hence, we must show that the MPC Problem (3) is a particular case of Problem (4).

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 [5]. This results is achieved thanks to the introduction of N consensus variables $z_t \in \mathbb{R}^n$ ($t = 1, \dots, N$) used to break up the dynamic coupling (3b). This decomposition allows to

Algorithm 3 SVR-AMA.

Given $\bar{\mu}^0$, N , $\mathcal{I}_N := \{0, \dots, N\}$, τ , 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. $z^k = \operatorname{argmin}_z g(z) + \langle \mu^{k-1}, -H_z z \rangle + \frac{\tau}{2} \|d - H_y y^k - H_z z\|^2$.

3. $\beta^k = \tilde{\beta} + (y^k - \tilde{y})^T H_{y_i}^T \pi_i^{-1}$

4. $\mu^k = \mu^{k-1} - \tau(\beta^k + H_z z^k - d)$

end for

5. $\tilde{\mu}^s = \frac{1}{T} \sum_{k=1}^T \mu^k$, $\tilde{y}^s = \frac{1}{T} \sum_{k=1}^T y^k$.

end while

reformulate Problem (3) as follows:

$$\min_{x,u} \frac{1}{2} \sum_{t=0}^N \left(x_t^{(t)T} Q x_t^{(t)} + u_t^{(t)T} R u_t^{(t)} \right) \quad (12a)$$

$$\text{s.t.}: z_{t+1} = A x_t^{(t)} + B u_t^{(t)} \quad t = 0, \dots, N-1 \quad (12b)$$

$$z_{t+1} = x_{t+1}^{(t+1)} \quad t = 0, \dots, N-1 \quad (12c)$$

$$C x_t^{(t)} + D u_t^{(t)} \leq d \quad t = 0, \dots, N \quad (12d)$$

$$x_0^{(0)} = x_{\text{init}}, \quad (12e)$$

where the original dynamic coupling (3b) has been replaced by the consensus constraints (12b) and (12c). Note that we introduced the superscript t to underline that the x_t and u_t are local variables of the subproblems obtained after the time splitting [5]. Finally, if we introduce $N + 1$ additional slack variables $\sigma_t \in \mathbb{R}^p$ to remove the inequality constraints (12d) and define $\mathbb{C} := \{\sigma_t \in \mathbb{R}^p \mid \sigma_t \geq 0\}$, Problem (12) can be written as the sum of the following subproblems:

$$\min_{y_t} f_t(y_t) + \sum_{i=1}^p \mathcal{I}_{\mathbb{C}}(\sigma_{t_i}) \quad (13a)$$

$$\text{s.t.}: w_t : z_t = H_1 y_t, \quad (13b)$$

$$v_{t+1} : z_{t+1} = H_2 y_t, \quad (13c)$$

$$\lambda_t : \sigma_t = d - G y_t, \quad (13d)$$

where we define $y_t^T := [x_t^{(t)T} \ u_t^{(t)T}]$, $f_t(y_t) := y_t^T Q y_t$, $Q := \operatorname{diag}\{Q, R\}$, $G := [C \ D]$, $H_1 := [I_n \ 0_{n \times m}]$, $H_2 := [A \ B]$. Furthermore, for each equality constraint in Problem (13), the corresponding Lagrange multipliers have been highlighted.

If we define $\mathbf{y}^T = [y_0^T \ \dots \ y_N^T]$, $f(\mathbf{y}) = \mathbf{y}^T \mathbf{Q} \mathbf{y} = \sum_{t=0}^N f_t(y_t)$, $\mathbf{z}^T = [z_1^T \ \dots \ z_N^T \ \sigma_0^T \ \dots \ \sigma_N^T]$, $\mathbf{Q} = \operatorname{diag}\{Q \ \dots \ Q\}$, $g(\mathbf{z}) = \sum_{t=0}^N \mathcal{I}_{\mathbb{C}}(\sigma_t)$, $h_{y_0}^T = [H_2^T \mid -G^T]$, $h_y^T = [H_1^T \ H_2^T \mid -G^T]$, $h_{y_0}^T = [H_1^T \mid -G^T]$, $h_{d_0}^T = h_{d_N}^T = [0_n \mid -d^T]$,

$$h_d^T = [0_n \ 0_n \ | \ -d^T],$$

$$H_y := \begin{bmatrix} h_{y_0} & 0 & \cdots & \cdots & 0 \\ 0 & h_y & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & h_y & 0 \\ 0 & 0 & \cdots & 0 & h_{y_N} \end{bmatrix}, \mathbf{d} := \begin{bmatrix} h_{d_0} \\ h_d \\ \vdots \\ h_d \\ h_{d_N} \end{bmatrix},$$

$$H_z := \begin{bmatrix} I_n & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & I_p & 0 & \cdots & 0 \\ \hline I_n & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & I_n & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & I_p & \cdots & 0 \\ \vdots & \ddots & & \vdots & \ddots & & & 0 \\ \hline 0 & 0 & \cdots & I_n & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I_p \end{bmatrix},$$

Problem (3) can be rewritten as follows:

$$\text{minimize } f(\mathbf{y}) + g(\mathbf{z}) \quad (14a)$$

$$H_y \mathbf{y} + H_z \mathbf{z} = \mathbf{d}, \quad (14b)$$

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

For the proposed splitting, Assumptions 2 and 3 are satisfied. Concerning Assumption 1, note that $L(\nabla F) := \text{eig}_{\max}(H_y)\sigma_f^{-1} = \max_t(\text{eig}_{\max}(H_{y_t})\sigma_{f_t}^{-1}) = \max_t(L_t(\nabla F_t)) = L_t(\nabla F_t) \leq \sum_{t=0}^N L_t(\nabla F_t)$, where the last equality follows from the fact that we deal with LTI systems ($L_0 = L_1 = \dots = L_N$). 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 (14) is detailed in Algorithm 4. In particular, defining $\boldsymbol{\mu}^T := [v_1^T \ \lambda_0^T \ | \ w_1^T \ \dots \ w_{N-1}^T \ v_N^T \ \lambda_{N-1}^T \ | \ w_N^T \ \lambda_N^T]$, according to the partitioning of H_y and H_z , $F(\boldsymbol{\mu}) = f^*(H_y^T \boldsymbol{\mu})$. 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 $H_y \mathbf{y}$, that can be performed efficiently by exploiting the structure of the matrix H_y . Finally, note that, given the structure of $F(\boldsymbol{\mu})$ the probability π_t does not affect the choice of the step size τ , according to Remark 1.

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

Theorem 3. Consider Problem (14). Let $\{\mathbf{y}^k\}$ and $\{\boldsymbol{\mu}^k\}$ be the sequence of primal and dual variables, respectively, generated by Algorithm 4. If Assumptions 1-3 are satisfied, given $\tilde{\boldsymbol{\mu}}^0 \in \text{dom}(G)$, where $G := g^*(H_z^T \boldsymbol{\mu}) - d^T \boldsymbol{\mu}$, then, the following holds:

$$\mathbb{E}\|\tilde{\mathbf{y}}^s - \mathbf{y}^*\|^2 \leq \frac{2}{\sigma_f} (D(\boldsymbol{\mu}^*) - \mathbb{E}D(\tilde{\boldsymbol{\mu}})). \quad (15)$$

Proof. We summarize the logic of the proof. The inequality

Algorithm 4 SVR-AMA for Problem (14).

Given $\tilde{\boldsymbol{\mu}}^0$, N , $\mathcal{I}_N := \{0, \dots, N\}$, $L^* := (\sigma_f)^{-1} \text{eig}_{\max}(H_y)$, $0 < \tau < 1/(4L^*)$, 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 $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}^{s-1}$.

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. $\mathbf{y}_i^k = \text{argmin}_y f_i(y_i) + \langle w_i, H_1 y_i \rangle + \langle v_{i+1}, H_2 y_i \rangle + \langle \lambda_i, -G y_i \rangle$.

2b. $\sigma_i^k = \text{Pr}_{\mathbb{C}}(G y_i^k - d - \frac{1}{\tau} \lambda_i)$.

2c. $z_i^k = \frac{1}{2} [H_1 y_i^k + H_2 y_{i-1} - \frac{1}{\tau} (w_i + v_i)]$.

3a. $\beta_{w_i}^k = \tilde{\beta}_{w_i} + \frac{(y_i^k - \tilde{y}_i)^T H_1^T}{\pi_i}$.

3b. $\beta_{v_i}^k = \tilde{\beta}_{v_i} + \frac{(y_{i-1} - \tilde{y}_{i-1})^T H_2^T}{\pi_i}$.

3c. $\beta_{\lambda_i}^k = \tilde{\beta}_{\lambda_i} - \frac{(y_i^k - \tilde{y}_i)^T G^T}{\pi_i}$.

4a. $w_i^k = w_i + \tau (z_i^k - \beta_{w_i}^k)$.

4b. $v_i^k = v_i + \tau (z_i^k - \beta_{v_i}^k)$.

4c. $\lambda_i^k = \lambda_i + \tau (\beta_{\lambda_i}^k + d - \sigma_i^k)$.

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 $\tilde{\mathbf{y}}^s = \frac{1}{T} \sum_{k=1}^T \mathbf{y}^k$.

end while

can be derived by the results of Theorem 5.3 in [7] by noticing that the primal updates in the inner loop are the same as AMA. Then, we have to take into account for Algorithm 4 that the primal variables are stochastic variables and that we must consider their expected values. These observations combined with the results of Theorem 2 lead to (15). \square

Remark 3. 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.

The decomposition along the length of the prediction horizon offers several advantages. First, the size of the subproblems (13) is fixed and independent from the length of the prediction horizon. Second, the resultant subproblems have improved numerical properties (in terms of condition number, for example), compared to solving Problem (3). Third, this decomposition allows one to fully parallelize the solution of Problem (3), thanks to the introduction of the consensus variables. In theory, 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 [5]. 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 3 plays a fundamental role to compute a suboptimal solution of Problem (3).

Algorithm 3 applied to Problem (12) 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 (e.g., [13]), Algorithm 3 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 next section.

VI. NUMERICAL EXAMPLE

This section considers the linearized model (at a given trim condition) of an Airbus passenger aircraft 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 states have different magnitudes (ranging from few degrees to thousands of feet) affecting the conditioning of the MPC problem. Third, some of the open-loop system eigenvalues are, in general, complex conjugate close to the imaginary axis. 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/sec], roll rate [deg/sec], ground speed [knots], 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 $T_s = 0.04$ sec 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

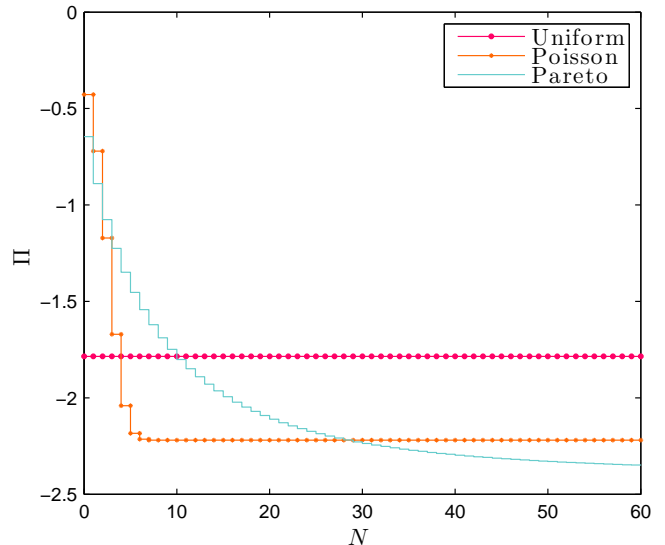


Fig. 1: Probability distributions used to test the performance of Algorithm 4 plotted in \log_{10} scale.

to the origin starting from a nonzero initial condition close to the saturation limits of the system.

We compared the behavior of Algorithm 4 to a synchronous one (a state-of-the-art AMA [11], [9]). The baseline for the comparison is the trajectory obtained using the MPC functions of MPT3 [21]. In particular, we are interested in showing that the possibility of tuning the probability distribution that the algorithm offers can lead to significant 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 four different probability distributions (depicted in Figure 1): (i) uniform, (ii) Poisson, (iii) generalized Pareto, and (iv) adaptive. Scenario (iv) is obtained as follows. We initialize the Π to be the uniform distribution. Then, every T iterations, we check, for each $t = 0, \dots, N$ whether the following condition is verified $\|\lambda^T - \lambda^{T-1}\|^2 < 0.01$. 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 $\pi_{t-1} \leftarrow \pi_{t-1} + 0.25\pi_t$.

To compare the different probability tuning strategies in practice, we use the same batch size $T = 10 < N$, step size $\tau = 0.9$, and number of outer iterations $\bar{s} = 15,000$ for all the numerical experiments. From the practical point of view, we noticed that the asynchronous algorithm would allow one to select a larger step size (compared to the synchronous algorithm), despite the large condition number of the problem ($\kappa \approx 10^5$). The tuning of the parameters is made to compare with AMA. In particular, we selected the same τ and the number of iteration of AMA is set to match the number of iterations of SVR-AMA ($\bar{s} = 15,000 \times T/N$). This choice is made to compare, given the same computation time, the quality of the solution obtained using the synchronous and the asynchronous algorithms.

Figures 2 and 4 show the open-loop predictions obtained

for the elevator command and the angle of attack using different probability distributions. As expected, the predictions are suboptimal. Nevertheless, it is interesting to note that the results obtained using the adaptive distribution, given the same number of iterations and step size of the other setups. The beginning of the horizon is solved with a higher accuracy, while the tail of the horizon is almost never updated. This does not affect the closed-loop performance, as Figures 3 and 5 show. In particular, the behavior obtained using the adaptive distribution outperforms the other setups, given the same number of iterations. In particular, compared to AMA and SVR-AMA with uniform distribution, within the same number of iterations, SVR-AMA with adaptive distribution reaches a higher level of suboptimality, which can be a useful feature if the system has hard real-time constraints and the number of iterations of the solver allowed within the sampling time is small.

This simple, but illustrative, example clearly shows the advantages of tuning the probability distribution of the random updates, in order to improve the quality of the estimates and speed up the computation of the MPC problem solution.

VII. CONCLUSIONS

We presented an asynchronous alternating minimization algorithm with variance reduction scheme suitable for model predictive control (MPC) applications. As our numerical example showed, the proposed algorithm outperforms a state-of-the-art solver (the alternating minimization algorithm) in terms of number of iterations needed to reach a desired level of suboptimality (i.e., when the solver terminates after a fixed number of iterations). Furthermore, the possibility of tuning the probability distribution of the random updates is an additional benefit for MPC applications. In particular, compared to other state-of-the-art asynchronous dual solvers that only performs random updates according to a uniform distribution, the proposed algorithm allows 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.

As part of our future work, we aim to extend these results to different splitting strategies and to formulate its accelerated version.

REFERENCES

- [1] T. Keviczky and G. J. Balas, "Receding horizon control of an F-16 aircraft: a comparative study", *Control Engineering Practice*, vol. 12, n.9, pp. 1023–1033, 2006, DOI:10.1016/j.conengprac.2005.06.003.
- [2] D. Hrovat et al, "The development of model predictive control in automotive industry: A survey", Proc. of the *IEEE International Conference on Control Applications*, pp. 295–302, 2012, DOI:10.1109/CCA.2012.6402735.
- [3] S. Richter, C. N. Jones, and M. Morari, "Real-time input-constrained MPC using fast gradient methods", Proc. of the *48th IEEE CDC* held jointly with the *28th CCC*, pp. 7387–7393, 2009, DOI:10.1109/CDC.2009.5400619.
- [4] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control", *IEEE Transactions on Automatic Control*, vol. 59, n. 1, pp. 18–33, 2014, DOI: 10.1109/TAC.2013.2275667.
- [5] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems", Proc. of the *ECC*, pp. 3089–3094, 2013.

- [6] L. Ferranti and T. Keviczky, "A Parallel Dual Fast Gradient Method for MPC Applications", Proc. of the *54th IEEE CDC*, pp. 2406–2413, 2015, DOI:10.1109/CDC.2015.7402568.
- [7] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Fast Alternating Minimization Algorithm for Model Predictive Control", Proc. of the *19th IFAC World Congress*, pp. 11980–11986, 2014, DOI:10.3182/20140824-6-ZA-1003.01432.
- [8] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Foundations and Trends[®] in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011, DOI:10.1561/22000000016.
- [9] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast Alternating Direction Optimization Methods", *SIAM Journal on Imaging Sciences*, vol. 7, n. 3, pp. 1588–1623, 2014, DOI:10.1137/120896219.
- [10] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction", *SIAM Journal on Optimization*, vol. 24, n. 4, pp. 2057–2075, 2014, DOI:10.1137/140961791.
- [11] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities", *SIAM Journal on Control and Optimization*, vol. 29, n.1, pp. 119–138, 1991, DOI:10.1137/0329006.
- [12] P. Goupil et al, "An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives", Proc. of the *9th IFAC Symposium on SAFEPROCESS*, pp 976–981, 2015.
- [13] I. Notarnicola and G. Notarstefano, "Randomized dual proximal gradient for large-scale distributed optimization", Proc. of the *54th IEEE CDC*, pp. 712–717, 2015, DOI:10.1109/CDC.2015.7402313.
- [14] S. Boyd and L. Vandenberghe, "Convex Optimization", *Cambridge University Press*, 2004.
- [15] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", *Automatica*, vol. 38, n. 1, pp. 3–20, 2002, DOI:10.1016/S0005-1098(01)00174-1.
- [16] J.-B. Hiriart-Urruty and C. Lemaréchal, "Convex Analysis and Minimization Algorithms", *Springer-Verlag*, 1993.
- [17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", *SIAM Journal on Imaging Sciences*, vol. 2, n. 1, pp. 183–202, 2009, DOI:10.1137/080716542.
- [18] Y. Singer, J.C. Duchi, "Efficient learning using forward-backward splitting", Proc. of *Advances in Neural Information Processing Systems*, pp. 495–503, 2009.
- [19] J. Langford, L. Li, T. Zhang, "Sparse online learning via truncated gradient", Proc. of *Advances in neural information processing systems*, pp. 905–912, 2009.
- [20] A. Nitanda, "Stochastic proximal gradient descent with acceleration techniques", Proc. of *Advances in Neural Information Processing Systems*, vol. 27, pp. 1574–1582, 2014.
- [21] M. Herceg et al, "Multi-Parametric Toolbox 3.0", Proc. of the *European Control Conference*, pp. 502–510, 2013, <http://control.ee.ethz.ch/~mpt>.

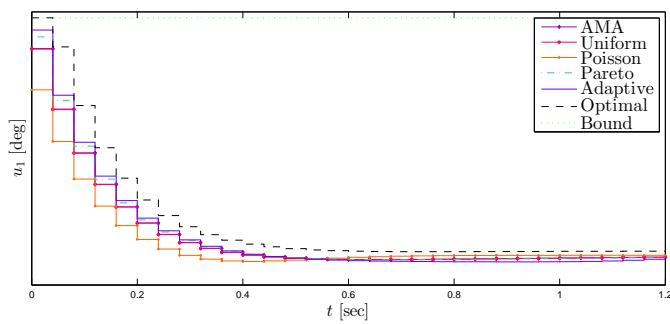


Fig. 2: Control trajectories obtained using different probability distributions Π in Algorithm 4 in open loop.

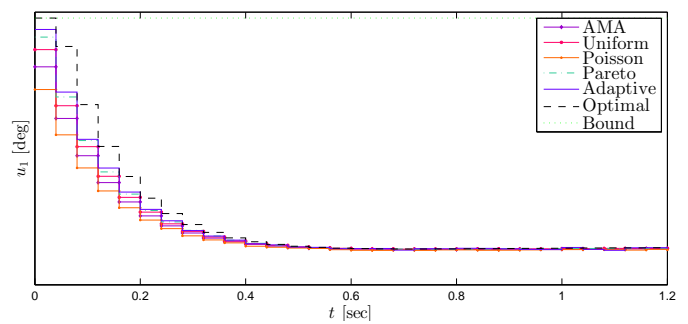


Fig. 3: Control trajectories obtained using different probability distributions Π in Algorithm 4 in closed loop.

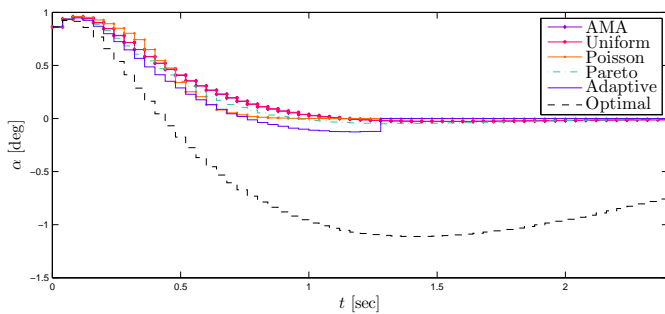


Fig. 4: Angle of attack trajectories obtained using different probability distributions Π in Algorithm 4 in open loop.

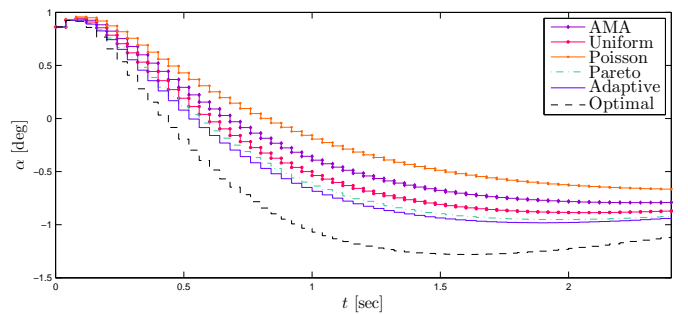


Fig. 5: Angle of attack trajectories obtained using different probability distributions Π in Algorithm 4 in closed loop.