2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

A hierarchical time-splitting approach for solving finite-time optimal control problems

Georgios Stathopoulos[1], Tamás Keviczky[2] and Yang Wang[3]

Abstract— **We present a hierarchical computation approach for solving finite-time optimal control problems using operator splitting methods. The first split is performed over the time index and leads to as many subproblems as the length of the prediction horizon. Each subproblem is solved in parallel and further split into three by separating the objective from the equality and inequality constraints respectively, such that an analytic solution can be achieved for each subproblem. The proposed approach leads to a highly parallelizable nested decomposition scheme. We present a numerical comparison with the standard state-of-the-art solver SDPT3, and provide analytic solutions to several elements of the algorithm, which enhances its applicability in fast large-scale applications.**

I. INTRODUCTION

Online optimization and optimal control methods are increasingly being considered for fast embedded applications, where efficient, reliable, and predictable computations involved in calculating the optimal solutions are a necessity. The potential use of optimal control in such embedded systems promises energy savings and more efficient resource usage, increased safety, and improved fault detection. The range of application areas that can benefit from embedded optimization include the mechatronics, automotive, process control and aerospace sectors [1]. The promise of unprecedented performance and capabilities in these applications, which typically rely on large-volume, real-time embedded control systems, has fueled recent research efforts towards fast and parallel optimization solvers.

One of the main research directions aim at developing special-purpose optimization solvers that target typical control or estimation problems arising in optimal control. Parallel solutions to systems of linear equations appearing in interior-point, and active set methods have been studied in [2]–[5]. In this work we consider a quadratic finite-time optimal control problem for discrete-time systems with constrained linear dynamics, which appears in typical model predictive control problems [6]. We investigate and develop different parallelizable algorithms using operator splitting techniques [7], [8] that have recently shown great promise for speeding up calculations involved in computing optimal solutions with medium accuracy [9]–[11]. Our approach relies on a hierarchical splitting up of the specially structured finite-time optimal control problem. The first split is performed over the time index and leads to as many subproblems as the length of the prediction horizon. Each subproblem can then be solved in parallel and further split into three by separating the objective from the equality and inequality constraints respectively, such that an analytic solution can be achieved for each subproblem. The proposed solution approach leads to a nested decomposition scheme, which is highly parallelizable. The proposed three-set splitting method does not only solve the particular quadratic programs (QPs) that appear in the update steps of the time-splitting algorithm efficiently, but also provides a compact, standalone alternative for solving generic QPs.

The paper is structured as follows. Section II presents the main idea behind the time-splitting optimal control approach for parallel computations, using the Alternating Direction Method of Multipliers and deriving the exact formulas required for each subproblem and update step. In Section III we propose an alternative scheme for solving the QPs with general polyhedral constraints that arise in the time-splitting update steps (or for any other generic QP). The two splitting schemes are combined in a hierarchical fashion in Section IV, and numerical experiments are performed in Section V to compare its performance with some advanced solvers in the literature. Section VI concludes the paper.

II. TIME-SPLITTING OPTIMAL CONTROL

A. Problem Formulation

We consider the following finite-time optimal control problem formulation that arises in typical model predictive control applications:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_{t=0}^{N}\left(x_t^T Q_t x_t + u_t^T R_t u_t\right) && \text{(1a)}\\ \text{subject to} \quad & (x_t,u_t)\in \mathscr{X}_t\times\mathscr{U}_t, \quad t=0,\ldots,N && \text{(1b)}\\ & x_{t+1}=A_t x_t + B_t u_t + c_t, t=0,\ldots,N-1 && \text{(1c)} \end{aligned}$$

where the decision variables are the states $x_t\in\mathbf{R}^n$, and the inputs $u_t\in\mathbf{R}^m$ of the system for $t=0,\ldots,N$. The index t denotes time, and the system evolves according to linear dynamics constraint (1c) where $c_t\in\mathbf{R}^n$ is considered to be a known disturbance. Here, N is the prediction horizon and $Q_t\in\mathbf{R}^{n\times n}, R_t\in\mathbf{R}^{m\times m}$ are symmetric matrices. The stage cost in (1a) is convex quadratic with $Q_t\succeq 0$ and $R_t\succ 0$. The stage-wise state-input pairs are constrained to reside within polyhedra (1b) denoted by $\mathscr{X}_t$ and $\mathscr{U}_t$, respectively. These are constraint sets defined by linear inequalities that involve states and inputs at the same sample time index.

[1]G. Stathopoulos is with Laboratoire d'Automatique, EPFL, CH-1015 Lausanne, Switzerland, georgios.stathopoulos@epfl.ch

[2]T. Keviczky is with the Delft Center for Systems and Control, Delft University of Technology, Delft, CD 2628, The Netherlands, t.keviczky@tudelft.nl

[3]Y. Wang is with Stanford University, Stanford, CA 94305, USA, yang1024@gmail.com

Motivated by the principles of operator splitting methods (see [12] for details and relevant references), we propose to split the problem (1) into $N+1$ smaller stage-wise subproblems that can be solved in parallel. This requires breaking the coupling that appears due to the dynamics. We introduce a copy of each variable that couples the dynamics equations in order to allow such a splitting into subproblems, and subsequently impose a consensus constraint on the associated complicating variables and their copies. This leads to the following equivalent formulation of (1), where the complicating variables that are used to perform the splitting are clearly highlighted:

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{t=0}^{N}\left(x_t^{(t)T}Q_t x_t^{(t)} + u_t^{(t)T}R_t u_t^{(t)}\right) && \text{(2a)}\\
\text{subject to} \quad & (x_t^{(t)}, u_t^{(t)}) \in \mathscr{X}_t \times \mathscr{U}_t, \quad t=0,\ldots,N && \text{(2b)}\\
& x_{t+1}^{(t)} = A_t x_t^{(t)} + B_t u_t^{(t)} + c_t, && \text{(2c)}\\
& x_0^{(0)} = x_{\text{init}} && \text{(2d)}\\
& \tilde{z}_{t+1} = x_{t+1}^{(t)} && \text{(2e)}\\
& \tilde{z}_{t+1} = x_{t+1}^{(t+1)}, \quad t=0,\ldots,N-1, && \text{(2f)}
\end{aligned}$$

where the subscript t of the decision variables $x_t^{(t)}, u_t^{(t)}$ indicates the time index and the superscript (t) denotes the group or subproblem where the variable belongs to. Hence, each subproblem contains three variables, the current state and input as well as a prediction of the state for the next time instant. The introduced complicating variable $\tilde{z}_t$ acts as a 'global' variable that brings the local copies $x_t^{(t-1)}$ and $x_t^{(t)}$ in agreement, i.e., $\tilde{z}_t = x_t^{(t-1)} = x_t^{(t)}$. The time-splitting idea is graphically depicted in Figure 1.

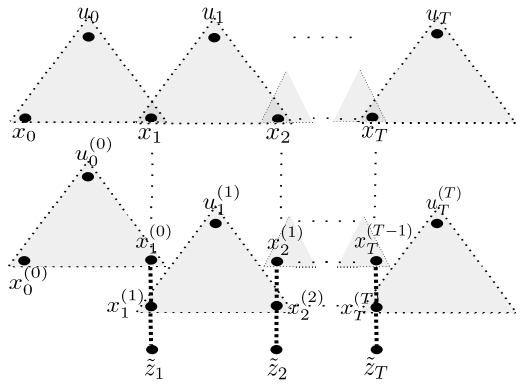

Fig. 1. The idea of the time splitting algorithm. Variables coupled via dynamics are decoupled using a slack variable $\tilde{z}_t = x_t^t = x_t^{(t-1)}$.

B. The time-splitting algorithm

In order to use a more compact formulation, we will denote the decision variables in (2) corresponding to each subproblem using

$$\tilde{x}_t = (x_t^{(t)}, u_t^{(t)}, x_{t+1}^{(t)}), \tag{3}$$

where $\tilde{x}_t \in \mathbf{R}^{2n+m}$. We also introduce dual variables to deal with the consensus equality constraints:

- $\tilde{w}_t$ associated with $x_t^{(t)} = \tilde{z}_t, \quad t=1,\ldots,N$
- $\tilde{v}_t$ associated with $x_t^{(t-1)} = \tilde{z}_t, \quad t=1,\ldots,N.$

In order to rewrite the finite-time optimal control problem in a more compact form, we define the following matrices:

$$\begin{aligned}
P_t &= \mathbf{diag}\begin{pmatrix} Q_t, R_t, 0 \end{pmatrix} \in \mathbf{R}^{(2n+m)\times(2n+m)}, && \text{(4a)}\\
F_t &= \begin{bmatrix} -A_t & -B_t & I \end{bmatrix} \in \mathbf{R}^{n\times(2n+m)}, && \text{(4b)}\\
G_0 &= \begin{bmatrix} I & 0 & 0 \end{bmatrix} \in \mathbf{R}^{n\times(2n+m)}, && \text{(4c)}\\
G_1 &= \begin{bmatrix} 0 & 0 & I \end{bmatrix} \in \mathbf{R}^{n\times(2n+m)}, && \text{(4d)}
\end{aligned}$$

where the operator $\mathbf{diag}(\cdot)$ indicates the block diagonal structure of the matrices involved.

We use the Alternating Direction Method of Multipliers (ADMM) [13], [14] in order to arrive at a solution approach that is amenable to parallel implementation. The updates involved in the ADMM algorithm include forming the augmented Lagrangian of the problem and minimizing over the primal variables $\tilde{x}_t$, $t=0,\ldots,N$ and $\tilde{z}_t$, $t=1,\ldots,N$, followed by updating the dual variables $\tilde{w}_t$ and $\tilde{v}_t$, $t=1,\ldots,N$. The three main steps of the algorithm are performed in an iterative fashion and are described next in detail. We use k to denote the algorithm's loop counter. The termination criterion based on primal and dual tolerances are provided. The analytic derivation of the formulas can be found in [15].

Step 1: Solving $N+1$ QP subproblems for the primal variables in $\tilde{x}_t$

Minimization of the augmented Lagrangian over the primal variables $\tilde{x}_t$ results in $N+1$ stage-wise quadratic programs (QPs):

- For the subproblem associated with the time instant $t=0$, we need to compute $\tilde{x}_0$ by solving

$$\begin{aligned}
\text{minimize} \quad & (1/2)\tilde{x}_0^T P_0 \tilde{x}_0 - \rho\tilde{v}_1^{kT}(G_1\tilde{x}_0 - \tilde{z}_1^k)\\
& +(\rho/2)\|G_1\tilde{x}_0 - \tilde{z}_1^k\|_2^2\\
\text{subject to} \quad & \tilde{x}_0 \in \mathscr{C}_0\\
& G_0\tilde{x}_0 = x_{\text{init}}\\
& F_0\tilde{x}_0 = c_0
\end{aligned} \tag{5}$$

- Similarly, we need to solve the following QPs for all the other groups of variables $\tilde{x}_t$, $t=1,\ldots,N-1$:

$$\begin{aligned}
\text{minimize} \quad & (1/2)\tilde{x}_t^T P_t \tilde{x}_t - \rho\tilde{v}_{t+1}^{kT}(G_1\tilde{x}_t - \tilde{z}_{t+1}^k)\\
& +(\rho/2)\|G_1\tilde{x}_t - \tilde{z}_{t+1}^k\|_2^2\\
& -\rho\tilde{w}_t^{kT}(G_0\tilde{x}_t - \tilde{z}_t^k)\\
& +(\rho/2)\|G_0\tilde{x}_t - \tilde{z}_t^k\|_2^2\\
\text{subject to} \quad & \tilde{x}_t \in \mathscr{C}_t\\
& F_t\tilde{x}_t = c_t
\end{aligned} \tag{6}$$

- For the subproblem associated with the final time instant $t=N$, we need to compute $\tilde{x}_N$ by solving

$$\begin{aligned}
\text{minimize} \quad & (1/2)\tilde{x}_N^T P_N \tilde{x}_N - \rho\tilde{w}_N^{kT}(G_0\tilde{x}_N - \tilde{z}_N^k)\\
& +(\rho/2)\|G_0\tilde{x}_N - \tilde{z}_N^k\|_2^2\\
\text{subject to} \quad & \tilde{x}_N \in \mathscr{C}_N
\end{aligned} \tag{7}$$

The polyhedral sets $\mathscr{C}_t$, $t = 0,\ldots,N$ are defined as

$$\mathscr{C}_t = \mathscr{X}_t \times \mathscr{U}_t \times \mathscr{X}_{t+1} \subseteq \mathbf{R}^{2n+m} \tag{8}$$

and the variable $\rho > 0$ is a parameter of the algorithm.

Remark 1: Notice that for the time instant N the decision variables of the QP actually simplify to $\tilde{x}_N = x_N$ and $\mathscr{C}_N = \mathscr{X}_N$, but we keep the same notation for simplicity (and without loss of generality).

Step 2: Averaging

The update of the 'global' primal variables $\tilde{z}_t$, $t = 1,\ldots,N$ is derived from a simple quadratic minimization problem, the solution of which turns out to be an average of the predicted ($x_t^{(t-1)}$) and current ($x_t^{(t)}$) state

$$\tilde{z}_t^{k+1} = \frac{G_0\tilde{x}_t^{k+1} + G_1\tilde{x}_{t-1}^{k+1}}{2}, \quad t = 1,\ldots,N. \tag{9}$$

This intuitively makes sense, since the global variable can be obtained by collecting the local (primal) ones and computing the best estimate based on their values.

Step 3: Dual update

The dual updates can be expressed as

$$\tilde{w}_t^{k+1} = \tilde{w}_t^k - G_0\tilde{x}_t^{k+1} + \tilde{z}_t^{k+1}, \tag{10a}$$
$$\tilde{v}_t^{k+1} = \tilde{v}_t^k - G_1\tilde{x}_{t-1}^{k+1} + \tilde{z}_t^{k+1}, \quad t = 1,\ldots,N. \tag{10b}$$

Termination criterion

The algorithm terminates when a set of *primal* and *dual* residuals are bounded by a specified threshold (primal and dual tolerances); see [12, §3.2]. The primal and dual residuals for the time-splitting algorithm are respectively defined as

$$r^k = A_{\text{res}}x_{\text{pri}}^k + B_{\text{res}}z_{\text{pri}}^k, \quad s^k = -\rho A_{\text{res}}^T B_{\text{res}}(z_{\text{pri}}^k - z_{\text{pri}}^{k-1}). \tag{11}$$

The termination criterion is activated when

$$\|r^k\|_2 \le \varepsilon^{\text{pri}}, \quad \|s^k\|_2 \le \varepsilon^{\text{dual}},$$

where the tolerances ε^{pri} and $\varepsilon^{\text{dual}}$ are defined as follows:

$$\begin{aligned}\varepsilon^{\text{pri}} &= \varepsilon^{\text{abs}}\sqrt{N2n} + \varepsilon^{\text{rel}}\max\{\|A_{\text{res}}x_{\text{pri}}\|_2, \|B_{\text{res}}z_{\text{pri}}\|_2, \|c_{\text{res}}\|_2\}\\ \varepsilon^{\text{dual}} &= \varepsilon^{\text{abs}}\sqrt{(2n+m)(N+1)} + \varepsilon^{\text{rel}}\|A_{\text{res}}^T v_{\text{dual}}\|_2\end{aligned} \tag{12}$$

and we defined the vectors

$$\begin{aligned}x_{\text{pri}} &= \begin{pmatrix} \tilde{x}_0, & \tilde{x}_1, & \ldots, & \tilde{x}_N \end{pmatrix},\\ z_{\text{pri}} &= \begin{pmatrix} \tilde{z}_1, & \tilde{z}_2, & \ldots, & \tilde{z}_N \end{pmatrix},\\ v_{\text{dual}} &= \begin{pmatrix} \tilde{v}_1, & \tilde{w}_1, & \tilde{v}_2, & \ldots, & \tilde{v}_N, & \tilde{w}_N \end{pmatrix}.\end{aligned}$$

Using $\otimes$ to denote the Kronecker product, the residual matrices A_{res} and B_{res} and the residual vector c_{res} are

$$\begin{aligned}A_{\text{res}} &= \mathbf{diag}\begin{pmatrix} G_1, I_{N-1}\otimes(G_0,G_1), G_0 \end{pmatrix} \in \mathbf{R}^{N2n\times(N+1)(2n+m)},\\ B_{\text{res}} &= \mathbf{diag}\begin{pmatrix} I_N\otimes(-I,-I) \end{pmatrix} \in \mathbf{R}^{N2n\times Nn},\\ c_{\text{res}} &= 0 \in \mathbf{R}^{N2n}.\end{aligned}$$

The three update steps described above are fully parallelizable at each iteration k. Assuming $N+1$ processors are available, then processor Π_t would need to execute the following actions for $t = 0,\ldots,N-1$:

1) Receive the estimate $\tilde{z}_{t+1}^k$ and $\tilde{v}_{t+1}^k$ from neighboring processor Π_{t+1} (6).
2) Compute $\tilde{x}_t^{k+1}$ and send to processor Π_{t+1}.
3) Receive the estimate $\tilde{x}_{t-1}^{k+1}$ from neighboring processor Π_{t-1} and compute $\tilde{z}_t^{k+1}$ (9).
4) Compute $\tilde{w}_t^{k+1}$ and $\tilde{v}_t^{k+1}$ (10), and send to Π_{t-1}.

The above scheme suggests that each processor Π_t, $t = 1,\ldots,N-1$ interacts with the two neighboring processors Π_{t-1} and Π_{t+1}. Processors Π_0 and Π_N communicate only with processors Π_1 and Π_{N-1}, respectively. After updating all variables, a gather operation follows in order to compute the residuals and check the termination criterion.

III. THREE-SET SPLITTING QP SOLVER

A. Motivation

The time-splitting algorithm presented in the previous section decomposes the centralized finite-time optimal control problem so that it can be solved using multiple parallel processors. However, the updates for the primal variables $\tilde{x}_t$, $t = 0,\ldots,N$ given in (5), (6) and (7) involve solving a QP at each iteration of the algorithm. Even though several fast interior point solvers exist for this purpose (see e.g., [16]), these are mostly suitable for only a limited number of variables. Although recently more computationally efficient schemes that scale better with the problem size have been developed, they are restricted to cases where simple box constraints are considered [8], [10], [17]. In order to achieve fast computations in an embedded control environment, other generic solution methods would be preferred.

In this section we propose an alternative scheme for solving the QPs with general polyhedral constraints that arise in the previous section. We propose to perform yet another type of splitting approach, which splits the state-input variables of the QP in three sets. One set involves the variables that appear in the objective function, another includes the ones in the dynamics equality constraints, and the last set contains variables from the inequality constraints. In this way, we solve three simpler subproblems instead of the single general QP. Since several variables are shared among the subproblems, their solutions must be in consensus again to ensure consistency.

An important element of the proposed method is the introduction of an extra slack variable, which allows to get an analytic solution for the subproblem associated with the inequality constraints. Using this variable, the projection on any polyhedral set can be practically rewritten as a projection onto the nonnegative orthant. Besides this feature, the proposed splitting exploits structure in the resulting matrices and thus modern numerical linear algebra methods can be employed for speeding up the computations.

B. Problem setup

We consider a QP of the form

$$\begin{aligned}\text{minimize} \quad & \tfrac{1}{2}x^T Mx + q^T x + r\\ \text{subject to} \quad & Ax = b\\ & Hx \preceq h,\end{aligned} \tag{13}$$

with decision variable $x \in \mathbf{R}^n$, where $M \in S_+^n$, $q \in \mathbf{R}^n$, $r \in \mathbf{R}$, $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $H \in \mathbf{R}^{p \times n}$, $h \in \mathbf{R}^p$ and S_+^n is the cone of positive semidefinite matrices of dimension n.

In order to apply a variable splitting idea for this problem we first replicate all variables appearing in (13) three times, introducing three different sets for which we must ensure consensus. Furthermore, we use a slack variable to remove the polyhedral constraint and transform it into a projection operation onto the nonnegative orthant. We define the following sets of variables:

- First set - objective: x^{I}
- Second set - equality constraints: x^{II}
- Third set - inequality constraints: x^{III}
- First global variable: $z = x^{\mathrm{I}} = x^{\mathrm{II}} = x^{\mathrm{III}}$
- Second global variable: $y = h - Hx^{(\mathrm{III})}$
- First set of dual variables: $\tilde{z}^{\mathrm{I}}, \tilde{z}^{\mathrm{II}}, \tilde{z}^{\mathrm{III}}$ associated with $z = x^{\mathrm{I}} = x^{\mathrm{II}} = x^{\mathrm{III}}$, respectively.
- Second set of dual variables: $\tilde{y}$ assoc. with $y = h - Hx^{\mathrm{III}}$

Using the above sets of variables problem (13) can be restated in the equivalent form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^{\mathrm{I}T} M x^{\mathrm{I}} + q^T x^{\mathrm{I}} + r && \text{(14a)} \\ \text{subject to} \quad & A x^{\mathrm{II}} = b && \text{(14b)} \\ & y = h - H x^{\mathrm{III}}, \quad y \succeq 0 && \text{(14c)} \\ & x^{\mathrm{I}} = x^{\mathrm{II}} = x^{\mathrm{III}} = z, && \text{(14d)} \end{aligned}$$

with variables $x^{\mathrm{I}}, x^{\mathrm{II}}, x^{\mathrm{III}}, z \in \mathbf{R}^n, y \in \mathbf{R}^p$. The dual variables are of dimensions $\tilde{z} \in \mathbf{R}^n, \tilde{y} \in \mathbf{R}^p$.

C. The proposed three-set splitting algorithm

The proposed algorithm consists of iterative updates to the three ADMM steps similarly to the case of the time-splitting approach in Section II-B, namely one for the (local) primal variables $x^{\mathrm{I}}, x^{\mathrm{II}}, x^{\mathrm{III}}$, one for the (global) primal variables z and y and one for the dual variables $\tilde{z}^{\mathrm{I}}, \tilde{z}^{\mathrm{II}}, \tilde{z}^{\mathrm{III}}$ and $\tilde{y}$. We provide the algorithm's steps below along with some clarifying comments. The analytic derivations are presented in the Appendix.

Step 1: Solving three subproblems for the primal variables $x^{\mathrm{I}}, x^{\mathrm{II}}, x^{\mathrm{III}}$

In all three cases we have to solve simple, unconstrained QPs. The updates are:

$$\left(x^{\mathrm{I}}\right)^{k+1} = (M + \rho I)^{-1} (\rho (z^k + \left(\tilde{z}^{\mathrm{I}}\right)^k) - q) \qquad (15)$$

$$\begin{bmatrix} \rho I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \left(x^{\mathrm{II}}\right)^{k+1} \\ \nu \end{bmatrix} = \begin{bmatrix} \rho (z^k + \left(\tilde{z}^{\mathrm{II}}\right)^k) \\ b \end{bmatrix}, \qquad (16)$$

where ν is the dual variable associated with the equality constraint $Ax^{\mathrm{II}} = b$, and

$$\begin{aligned} \left(x^{\mathrm{III}}\right)^{k+1} \quad = \quad & (H^T H + I)^{-1} \Big(H^T (h - \tilde{y}^k - y^k) + \\ & z^k + \left(\tilde{z}^{\mathrm{III}}\right)^k \Big). \end{aligned} \qquad (17)$$

The matrices $M + \rho I$ and $H^T H + I$ are symmetric positive definite due to the regularization terms. This means that, instead of directly inverting the matrices, we can save computational effort by taking the Cholesky factorization, i.e., write the matrix as a product of a lower triangular matrix and its transpose (see, e.g., [18]). Furthermore, the matrix $\begin{bmatrix} \rho I & A^T \\ A & 0 \end{bmatrix}$ is a KKT matrix and $\rho I \succ 0$. Hence we can exploit its structure and use block elimination to solve the KKT system (see [19, App. C]). The resulting matrices can be pre-factorized and then used in every solve step. The right-hand sides are the only parts that change in the update loop.

Step 2: Averaging and projection

The update for z is

$$z^{k+1} = \frac{1}{3} \sum_{i=\mathrm{I}}^{\mathrm{III}} \left(x^i\right)^{k+1}, \qquad (18)$$

while for y it is

$$y^{k+1} = \left(h - H \left(x_t^{\mathrm{III}}\right)^{k+1} - \tilde{y}^k \right)_+. \qquad (19)$$

The z-update is an averaging over the three sets of the primal variables $x^{\mathrm{I}}, x^{\mathrm{II}}, x^{\mathrm{III}}$, while the y-update is the solution of a proximal minimization problem (see Appendix), resulting in a projection onto the nonnegative orthant, denoted by $(\cdot)_+$.

Step 3: Dual update

The update for the dual variables $\tilde{z}^i$ is

$$\left(\tilde{z}^i\right)^{k+1} = \left(\tilde{z}^i\right)^k - \left(x^i\right)^{k+1} + z^{k+1}, \quad i = \mathrm{I}, \mathrm{II}, \mathrm{III}. \qquad (20)$$

Similarly, the update for $\tilde{y}$ is

$$\tilde{y}^{k+1} = \tilde{y}^k + y^{k+1} - h + H \left(x^{\mathrm{III}}\right)^{k+1}. \qquad (21)$$

Remark 2: The termination criterion used for the algorithm is the same as for the *time-splitting* one (see Section III). The formulas for the residuals and matrices are not explicitly reported here due to the limited space. The interested reader can find them in [15].

Remark 3: For the QP corresponding to the last sample time $t = N$ (7), the algorithm simplifies to splitting into two sets (objective and inequality constraints), since there are no dynamics equality constraints. The updates and residuals follow directly from the more generic case presented above.

IV. HIERARCHICAL TIME-SPLITTING OPTIMAL CONTROL

It is a natural idea to combine the two splitting algorithms (time-splitting and three-set splitting) that were introduced in the preceding two sections in order to speed up the solution of the finite-time optimal control problem (1). This can be accomplished via a nested decomposition scheme, where we employ the *three-set splitting* algorithm to solve the QPs (5), (6) and (7) appearing in Step 1 of the *time-splitting* algorithm. If we rewrite the generalized inequality constraints appearing in the problem formulation as

$$(x_t^{(t)}, u_t^{(t)}) \in \mathscr{C}_t \Leftrightarrow H_t \tilde{x}_t \preceq h_t, \quad t = 0, \ldots, N, \qquad (22)$$

then the QPs can be written in the form described by (13). The idea is graphically depicted in Figure 2. The exact expressions for the QPs are given in [15].

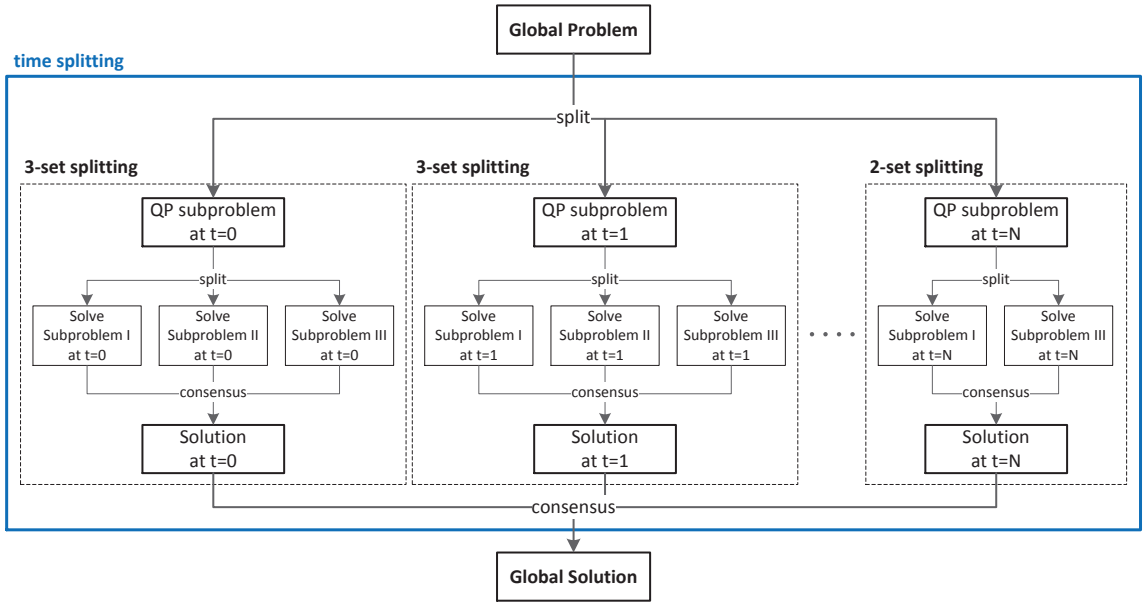

Fig. 2. Structural representation of the hierarchical time-splitting solution approach to the finite-time optimal control problem. The outer subproblems at $t = 0, 1, \ldots, N$ refer to (5), (6) and (7), respectively. The solutions of the nested subproblems enumerated by I, II and III correspond to (15), (16) and (17). Notice that only 2 subproblems have to be solved for the last time instant $t = N$.

For each iteration of the time-splitting algorithm, the three-set splitting algorithm runs in an inner loop until it converges. The quality of this convergence, i.e., the choice of the primal and dual tolerances of the inner loop (Remark 2) will affect the quality of the global solution.

A method that enables substantial speedup of the algorithm is warm starting. Since the three-set splitting algorithm will run for every iteration of the time-splitting algorithm, we can warm-start each QP with its previous solution. In this way, we can achieve a significant reduction in the number of iterations needed for the convergence of the inner loop.

V. NUMERICAL RESULTS

We consider three, randomly generated, numerical examples to illustrate the performance of the algorithm. The examples vary in terms of the number of decision variables involved. The systems considered are linear and time-invariant. We impose constraints on the difference between two consecutive states at each time instant of the form

$$x_{t,i} - x_{t,i-1} \leq dx,$$

where $x_{ti} \in \mathbf{R}$, $i = 1, \ldots, n$, $, t = 0, \ldots, N$ and box constraints on the inputs, i.e.,

$$\|u_t\|_\infty \leq u_{\max},\ t = 0, \ldots, N-1.$$

By adjusting the level of disturbance c_t in several time instances, we ensure activation of the constraints along the horizon.

For the simulations, we used an Intel Core i7 processor running at 1.7 GHz. We compared a C-implementation of our algorithm with using CVX [20], a parser-solver that uses SDPT3 [21]. For our method, the tolerances for both the outer and inner algorithms are set as $\varepsilon^{\text{pri}} = 10^{-4}$ and $\varepsilon^{\text{dual}} = 10^{-3}$. The parameter ρ was set after some simple tuning. The linear systems appearing in (15), (16) and (17) were solved by first factorizing the matrices off-line, using Tim Davis's sparse package [22]–[24] (see also [8]). The finite-time optimal control problem was solved only once and all the primal and dual variables were initialized at zero. However, the inner algorithm was warm-started at every iteration of the outer algorithm to the values acquired from the previous iteration. No relaxation or any other variance of the iterations was used. The numerical results in terms of computation times are summarized in Table I.

We can observe that, in the case of the small system, even when solving the problem on a single thread, the computation times are smaller than those of CVX. As the problem scales, the computations have to be parallelized in order to gain a significant advantage. More specifically, we expect the following speedup factors: 13 and 31 times faster in the case of the small problem (for the corresponding tolerances set to 10^{-4} and 10^{-3} respectively). For the medium-sized problem the speedups are by a factor of 10.5 and 24.6, and a factor of 5.4 and 11.7 for the large-scale problem, respectively.

In addition, we could observe that the factorization times are negligible in all cases, since the matrices being factorized are not large. Concerning the three-set splitting algorithm, only the average computational times are indicated over all iterations required to solve the problem. Warm starting was used when running the inner algorithm, as described in the previous section.

	small	medium	large
states n	10	20	50
inputs m	10	10	40
horizon length N	10	30	60
total variables	220	900	5400
ρ	15	25	50
active box constraints	5	6	20
active inequality constraints	2	2	4
CVX solve time	2430	3529	19420
factorization time	3.18	9.5	30
	Tolerance 10^{-4}		
3-set (average) iterations	21.80	17	15.95
3-set (average) solve time	0.75	1.38	9.15
time-split. iterations	250	241	389
time-split. solve time (single thread)	1880	10023	215888
time-split. solve time (N threads*)	188	334.1	3598
	Tolerance 10^{-3}		
3-set (average) iterations	13.14	13.27	12.32
3-set (average) solve time	0.49	1.18	7.34
time-split. iterations	156	128	224
time-split. solve time (single thread)	780	4304	99525
time-split. solve time (N threads*)	78	143.47	1659

* estimated parallel computation times

TABLE I

HIERARCHICAL TIME-SPLITTING OPTIMAL CONTROL: COMPUTATIONAL TIME RESULTS (MS) FOR DIFFERENT SIZE PROBLEMS.

VI. CONCLUSIONS

In this paper, we proposed an algorithm that solves a centralized convex finite-time optimal control problem making use of operator splitting methods, and, more specifically, the Alternating Direction Method of Multipliers. The initial problem is split into as many subproblems as the horizon length, which then can be solved in parallel.

The resulting algorithm is composed of three steps, including one where several QPs have to be solved. In this respect, we proposed another method, based again on operator splitting, that is applicable to QPs of any size, involving polyhedral constraints. This algorithm exploits the structure of the problem, leading to fast solutions.

The combination of the proposed algorithms results in a nested decomposition scheme for solving the aforementioned finite-time optimal control problems over several parallel processors.

Our numerical experiments suggest that the proposed hierarchical decomposition approach provides significant speed-up in computational time required for medium accuracy solutions for the class of problems considered. In our future work we intend to perform an even more extensive comparison with very recent tailor-made computational tools, and implement the algorithm on a parallel computing platform to obtain more accurate and representative computational time measurements.

ACKNOWLEDGMENTS

The main part of this work was carried out at Stanford. The authors would like to thank Stephen Boyd and Brendan O'Donoghue for helpful discussions. The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7-RECONFIGURE/2007–2013) under grant agreement no 314544.

REFERENCES

[1] *EMBOCON - Embedded Optimization for Resource Constrained Platforms*. EU project, 2010-2013. `http://www.embocon.org`.

[2] G. Constantinides, "Tutorial paper: Parallel architectures for model predictive control," in *European Control Conf.*, pp. 138–143, 2009.

[3] R. Gu, S. Bhattacharyya, and W. Levine, "Methods for efficient implementation of model predictive control on multiprocessor systems," in *IEEE Multi Systems Conf.*, pp. 1357–1364, 2010.

[4] J. Jerez, G. Constantinides, E. Kerrigan, and K. V. Ling, "Parallel MPC for Real-Time FPGA-based Implementation," in *IFAC World Congress*, pp. 1338–1343, 2011.

[5] A. Wills, A. Mills, and B. Ninness, "FPGA Implementation of an Interior-Point Solution for Linear Model Predictive Control," in *IFAC World Congress*, pp. 14527–14532, 2011.

[6] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control*. Cambridge University Press, 2012. In press.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, ch. 20.3.3. Operator Splitting Methods Generally. Cambridge Univ. Press, 2007.

[8] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Trans Control Sys Tech*, 2013. To appear.

[9] G. Stathopoulos, "Fast optimization-based control and estimation using operator splitting methods," Master's thesis, Delft Center for Systems and Control, Delft University of Technology, July 2012.

[10] M. Kögel and R. Findeisen, "Parallel solution of model predictive control using the alternating direction multiplier method," in *4th IFAC Nonlinear Model Predictive Control Conf.*, pp. 369–374, Aug. 2012.

[11] S. Boyd, M. Mueller, B. O'Donoghue, and Y. Wang, "Performance bounds and suboptimal policies for multi-period investment." `http://www.stanford.edu/~boyd/papers/port_opt_bound.html`, 2012.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 2011.

[13] R. Glowinski and A. Marrocco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de Dirichlet non lineares," *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, vol. 9, pp. 41–76, 1975.

[14] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Computers and Mathematics with App.*, vol. 2, pp. 17–40, 1976.

[15] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems." arXiv:1304.2152 [math.OC], 2013.

[16] J. Mattingley and S. Boyd, "CVXGEN: A Code Generator for Embedded Convex Optimization," *Optimization and Engineering*, vol. 13, pp. 1–27, 2012.

[17] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans Contr Sys Tech*, vol. 18, pp. 267–278, 2010.

[18] G. H. Golub and C. F. V. Loan, *Matrix computations*. Johns Hopkins University Press, third ed., 1966.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[20] G. M. Grant, S. Boyd, and Y. Ye, *Global Optimization: From Theory to Implementation*, ch. Disciplined Convex Programming, pp. 155–210. Nonconvex Optimization and its Applications, Springer, 2006.

[21] K. Toh, M. Todd, and R. Tütüncü, "SDPT3: A Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999.

[22] T. Davis, "Algorithm 849: A concise sparse Cholesky factorization package," *ACM Trans Math Software*, vol. 31, pp. 587–591, Dec. 2005.

[23] P. Amestoy, T. Davis, and I. Duff, "Algorithm 837: AMD, an approximate minimum degree ordering algorithm," *ACM Trans Math Software*, vol. 30, pp. 381–388, Sept. 2004.

[24] T. Davis, *Direct Methods for Sparse Linear Systems*. Fundamentals of Algorithms, SIAM, 2006.