Constrained LQR Using Online Decomposition Techniques*

L. Ferranti[1], G. Stathopoulos[2], C. N. Jones[2], and T. Keviczky[1]

***Abstract*— This paper presents an algorithm to solve the infinite horizon constrained linear quadratic regulator (CLQR) problem using operator splitting methods. First, the CLQR problem is reformulated as a (finite-time) model predictive control (MPC) problem without terminal constraints. Second, the MPC problem is decomposed into smaller subproblems of fixed dimension independent of the horizon length. Third, using the fast alternating minimization algorithm to solve the subproblems, the horizon length is estimated online, by adding or removing subproblems based on a periodic check on the state of the last subproblem to determine whether it belongs to a given control invariant set. We show that the estimated horizon length is bounded and that the control sequence computed using the proposed algorithm is an optimal solution of the CLQR problem. Compared to state-of-the-art algorithms proposed to solve the CLQR problem, our design solves at each iteration only unconstrained least-squares problems and simple gradient calculations. Furthermore, our technique allows the horizon length to decrease online (a useful feature if the initial guess on the horizon is too conservative). Numerical results on a planar system show the potential of our algorithm.**

I. INTRODUCTION

The linear quadratic regulator (LQR) proposed by [1] allows one, under mild assumptions on the system dynamics, to design an optimal state feedback to stabilize the plant in closed loop in the absence of constraints. When constrains are present the controller should be able to exploit as much as possible the actuator operating ranges to maximize the production. This observation motivated the study of model predictive control (MPC) in the late 70s [2], [3].

An MPC controller solves a constrained optimization problem (derived from the LQR formulation) over a finite time window (prediction horizon). The main advantage of MPC is its ability to handle constraints. The use of a finite time window, however, compromises the nice properties of the LQR controller in terms of optimality and closed-loop stability (more details can be found in [4], [5]). Closed-loop stability of the MPC controller can be preserved by including in the MPC problem formulation a terminal set (together with a terminal cost), as discussed in [6]. The main advantage is that this formulation is equivalent to solving the constrained infinite horizon LQR problem gaining tractability from a computational point of view. In particular, the terminal set is used to constrain the last predicted state to remain within a control invariant set. This set is usually selected to be the maximal positively invariant set of the closed loop obtained using the (unconstrained) LQR control law associated with the MPC controller. On one hand, the use of the terminal set (together with a terminal cost) allows one to prove closed-loop stability of the system controlled with MPC. On the other hand, the use of the terminal set reduces the region of attraction of the MPC controller leading, in general, to more conservative performance. For this reason, many practical MPC applications rely on the use of a *sufficiently long* prediction horizon tuned offline to ensure that the system will converge to the terminal set. Although this approach often works in practice, there are no guarantees that for all the possible initial conditions the closed-loop system is stable.

Contribution. We aim to solve the CLQR problem in a computationally tractable way, without relying on the use of the terminal set. In this respect, we propose the use of decomposition techniques to exploit the MPC problem structure and estimate the length of the prediction horizon online. The constrained LQR problem is reformulated as an MPC problem. The length of the horizon in the MPC problem, however, is not fixed, but it is decided online by our algorithm to guarantee closed-loop stability. By relying on the fast alternating minimization algorithm (FAMA) [8], our algorithm solves at each iteration unconstrained least-squares problems and simple gradient calculations of size independent of the length of the prediction horizon. We show, under mild assumptions on the cost and on the system dynamics, that the length of the horizon is bounded and the solution using our algorithm is an optimal solution of the CLQR problem. Finally, numerical results using the planar system of [12] are provided to show the potential of our proposed approach.

Related Work. The proposed approach relies on the work of [9] for the CLQR. We combined their technique with the decomposition along the length of the prediction horizon (*time-splitting* approach) proposed in [10], aiming to reduce the computational complexity of the algorithm, as detailed in Section IV. In [12], an algorithm to compute online the length of the horizon using operator-splitting techniques is also proposed. Compared to their approach, we use a different splitting technique that allows the length of the horizon to decrease online aiming to reduce the conservatism in the initial guess on the horizon length.

Outline. Section II introduces our problem formulation.

*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 European Union's Seventh Framework Programme (FP/2007-2013)/ ERC grant agreement n. 307608 (BuildNet), and 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).

[1]L. 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

[2]G. Stathopoulos and C. N. Jones are with the Laboratoire d'Automatique, École Polytecnique Fédérale de Lausanne (EPFL), Lausanne, CH-1015, Switzerland, {georgios.stathopoulos,colin.jones}@epfl.ch

Section III summarizes existing results from [8] and [9]. Section IV details our proposed approach. Section V presents numerical results. Section VI 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, $\mathbf{Pr}_{\mathbb{C}}(u)$ is the projection of u onto $\mathbb{C}$. 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, $\mathrm{eig}_{\max}(A)$ 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 $P = P^{\mathrm{T}} \in \mathbb{R}^{n \times n}$ is positive definite. Finally, details on the notions of strong convexity and Lipschitz continuity can be found in [11].

II. PROBLEM FORMULATION

This section presents the constrained LQR (CLQR) problem that we aim to solve using the decomposition techniques proposed in Section IV.

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

$$x(t+1) = Ax(t) + Bu(t), \tag{1}$$

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

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

and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ are known constant matrices.

Assumption 1. *The pair (A, B) is stabilizable.*

Our goal is to regulate to the origin the state of the system, starting from a feasible initial condition. In the absence of constraints, a natural choice to achieve this goal would be to design an optimal state feedback using the well known unconstrained linear quadratic regulator (ULQR) theory [1]. The design of the optimal state feedback, however, is complicated by the presence of the constraints (2). In particular, the computation of an optimal control law to regulate the system to the origin leads to the following infinite-dimensional optimization problem:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \ & \frac{1}{2}\sum_{t=0}^{\infty} x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t && \text{(3a)}\\
\text{subject to: } & x_{t+1} = Ax_t + Bu_t, \quad t \in \mathbb{N} && \text{(3b)}\\
& x_0 = x_{\text{init}}, && \text{(3c)}\\
& d \geq Cx_t + Du_t, \quad t \in \mathbb{N}, && \text{(3d)}
\end{aligned}$$

where x_t and u_t are the t-step-ahead state and control predictions, respectively. We refer to Problem 3 as the CLQR problem.

Assumption 2. *$Q \in \mathbb{S}^{n \times n}_{++}$ and $R \in \mathbb{S}^{m \times m}_{++}$.*

The CLQR control law obtained by solving Problem (3) is $u^{\infty} = u_0$, where $u_0(x_{\text{init}})$ is the first element of the infinite sequence of predicted control commands $\mathbf{u}^{\infty} := \{u_0, u_1, \ldots\}$.

Suppose that Assumptions 1 and 2 hold. Then, there exists an optimal stabilizing state-feedback gain $K \in \mathbb{R}^{m \times n}$ and $P \in \mathbb{S}^{n \times n}_{++}$ (solution of the algebraic Riccati equation) associated with the ULQR. Furthermore, starting from a given initial condition x_{init}, there exists a time instance N^{∞} in which the state will enter a polyhedral set $\mathcal{X}_f := \{x \in \mathbb{R}^n \,|\, C_f x \leq d_f\}$ such that $\forall\, x(t) \in \mathcal{X}_f \ \Rightarrow x(t+1) = (A + BK)x(t) \in \mathcal{X}_f \ \forall t \geq N^{\infty}$. The computation of N^{∞} is, in general, very challenging and in classical MPC it is common practice to reformulate Problem (3) as follows:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \ & \frac{1}{2}\sum_{t=0}^{N-1} x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t + V_f && \text{(4a)}\\
\text{subject to: } & x_{t+1} = Ax_t + Bu_t, \quad t = 0, \ldots, N && \text{(4b)}\\
& x_0 = x_{\text{init}}, && \text{(4c)}\\
& d \geq Cx_t + Du_t, \quad t = 0, \ldots, N-1, && \text{(4d)}\\
& x_N \in X_f, && \text{(4e)}
\end{aligned}$$

where the horizon N is fixed based on some heuristic, X_f is used as terminal set to constrain the N-step-ahead predicted state and $V_f := x_N^{\mathrm{T}} P x_N$ is used as terminal cost to replace $\sum_N^{\infty}(\cdot)$ in the cost (3a). We refer to Problem (4) as the finite-horizon MPC (FH-MPC) Problem.

The use of the terminal set allows one to prove closed-loop stability of the proposed MPC controller, but it also leads to conservatism in the performance. To avoid this conservatism, in practical applications, it is common practice to remove the terminal set and define a sufficiently long horizon N offline to guarantee that the predicted state at the end of the horizon is inside a positively invariant set, which is hard to compute.

Our work aims to solve Problem (3) by using an MPC approximation of the CLQR problem without explicitly relying on the terminal set. The proposed MPC controller relies on an online estimation strategy of the horizon length $N \geq N^{\infty}$ to ensure that the predicted terminal state enters the terminal set. In particular, we aim to solve the following problem:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \ & \frac{1}{2}\sum_{t=0}^{N-1} x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t + V_f && \text{(5a)}\\
\text{subject to: } & x_{t+1} = Ax_t + Bu_t, \quad t = 0, \ldots, N && \text{(5b)}\\
& x_0 = x_{\text{init}}, && \text{(5c)}\\
& d \geq Cx_t + Du_t, \quad t = 0, \ldots, N. && \text{(5d)}
\end{aligned}$$

In the remainder of the paper, we refer to Problem (5) as the adaptive-horizon MPC (AH-MPC), which differs from the FH-MPC Problem (4) in the definition of constraints (no terminal constraints) and length of $N \geq N^{\infty}$ adapted online.

Our approach is based on a similar idea to the one proposed in [9] combined with the use of splitting methods [8]. The next section summarizes these approaches.

III. PRELIMINARIES

In the following, we review interesting and closely related existing strategies to solve the CLQR Problem (3).

A. Constrained Linear Quadratic Regulator [9]

The design described in Section IV strongly relies on the results proposed by [9]. Hence, in the following, we

Algorithm 1 Constrained LQR [9].

Given $N = N^0$, K, $\mathcal{X}_f$, x_{init}.
1. Solve the AH-MPC Problem (5).
if $x_N \notin \mathcal{X}_f$ **then**
  2. $N = N + 1$.
  3. Return to Step 1.
end if
$\mathbf{u}^{\text{AH-MPC}} = \{u_0, \dots, u_{N-1}, Kx_N, Kx_{N+1}, \dots\}$.

summarize their algorithm and their main findings useful for the scope of the paper.

Under Assumptions 1 and 2 the following holds:

Theorem 1 (Theorem 3 in [9])**.** *Let V^∞ be the optimal cost obtained by solving the CLQR Problem* (3) *and let $\mathbf{u}^\infty$ be the associated optimal control sequence, for $x_0 = x_{init}$. Furthermore, let $V^{AH\text{-}MPC}$ be the optimal cost obtained by solving the AH-MPC Problem* (5) *using Algorithm 1 and let $\mathbf{u}^{AH\text{-}MPC}$ be the associated optimal control sequence. Then, for every $x_0 \in \mathcal{X}$, where $\mathcal{X}$ indicates the set of feasible states for which V^∞ is finite, there exists a **finite** positive integer $N^\infty(x_0)$ such that $V^\infty = V^{AH\text{-}MPC}$ and $\mathbf{u}^\infty = \mathbf{u}^{AH\text{-}MPC}$ for all $N \geq N^\infty$.*

By relying on the results of the theorem above, the authors in [9] propose Algorithm 1 to solve Problem (5) (and consequently Problem (3)). Note that step 1 of Algorithm 1 requires the computation of a solution for Problem (5) until the optimal length of the horizon has been computed. This might be very expensive from the computational viewpoint, if the initial estimate of the horizon is too conservative. In [9], the authors suggest to use $N^0 = 0$ to reduce the computational load. Section IV shows how to overcome this issue by exploiting the structure of Problem (5).

B. Fast Alternating Minimization Algorithm

Our design relies on the use of splitting methods. In particular, we exploit the fast alternating minimization algorithm (FAMA) [8]. FAMA solves the following problem:

$$\text{minimize} \;\; f(x) + g(y) \tag{6a}$$
$$\text{subject to:} \;\; H_x x + H_y y = d. \tag{6b}$$

The functions f and g satisfy the following assumptions:

Assumption 3. *f is strongly convex with convexity parameter σ_f.*

Assumption 4. *g is a convex function not necessarily smooth.*

FAMA (described in Algorithm 2) is equivalent to apply the fast proximal gradient method (e.g., FISTA [7]) on the dual function of Problem (6), as detailed in [8]. FAMA can handle problems in which $x \in \mathbb{C}$, i.e., when x belongs to a convex set $\mathbb{C}$. Both in the unconstrained and constrained scenarios, if Assumptions 3 and 4 are satisfied, it is possible to show that the FAMA has a theoretical convergence rate to the optimal solution for Problem (6) of $\mathcal{O}(\frac{1}{k^2})$ (refer to [8], [13] for more details). The following lemmas are useful for the remainder of the paper:

Algorithm 2 Fast alternating minimization algorithm [8].

Given H_x, H_y, c, μ_{init}, $\tau < \sigma_f / \text{eig}_{\max}(H_x)$.
Initialize $\hat{\mu}^1 = \mu^0 = \mu_{\text{init}}$, $\alpha^0 = 1$, $\alpha^1 = (1 + \sqrt{5})/2$.
for $k = 1, 2 \dots$ **do**
  1. $x^k = \text{argmin}_x \, f(x) + \langle \hat{\mu}^k, -H_x x \rangle$.
  2. $y^k = \text{argmin}_y \, g(y) + \langle \hat{\mu}^k, -H_y y \rangle + \frac{\tau}{2} \|d - H_x x^k - H_y y\|^2$.
  3. $\mu^k = \hat{\mu}^k + \tau(d - H_x x^k - H_y y^k)$.
  4. $\alpha^{k+1} = (1 + \sqrt{4\alpha^{k^2} + 1})/2$.
  5. $\hat{\mu}^{k+1} = \mu^k + (\alpha^k - 1)(\mu^k - \mu^{k-1})/\alpha^{k+1}$.
end for

Lemma 1. *[Lemma 3.4 in [13]] Let $\mathbb{C}$ be a convex cone. The conjugate function of the indicator function of the set $\mathbb{S} := \{v | -v \in \mathbb{C}\}$ is equal to the indicator function of the dual cone of $\mathbb{C}$, i.e., $\mathcal{I}^\star_{\mathbb{S}}(v) = \mathcal{I}_{\mathbb{C}^\star}(v)$.*

Lemma 2. *[Lemma 3.5 in [13]] Let $\mathbb{C}$ be the nonnegative orthant $\mathbb{C} := \{v \,|\, v \geq 0\}$. For any $v \in \mathbb{R}^{N_{\mathbb{C}}}$, the point $z \in \mathbf{Pr}_{\mathbb{C}}(v) - v$ satisfies $z \in \mathbb{C}$.*

IV. CONSTRAINED LQR WITH ADAPTIVE DECOMPOSITION ALONG THE HORIZON

This section presents our proposed approach to solve Problem (3). In particular, compared to the decomposition technique used in [12], our approach relies on the decomposition along the length of the prediction horizon proposed in [10] that allows one to solve $N + 1$ smaller subproblems (in place of Problem (3)) of size independent of the length of the horizon, which is an appealing quality when this quantity is unknown and potentially large.

According to the time-splitting strategy presented in [10], we introduce a new set of decision variables z_t ($t = 1, \dots, N$) to break up the dynamic coupling (3b). Furthermore, we add a new set of decision variables σ_t ($t = 0, \dots, N$) to handle the inequality constraints (5d). Then, we reformulate Problem (5) as follows:

$$\min_{x,u} \;\; \frac{1}{2} \sum_{t=0}^{N} x_t^{(t)^{\mathrm{T}}} Q_t x_t^{(t)} + u_t^{(t)^{\mathrm{T}}} R u_t^{(t)} + \mathcal{I}_{\mathbb{C}}(\sigma_t) \tag{7a}$$
$$\text{s.t.:} \; z_{t+1} = A x_t^{(t)} + B u_t^{(t)} \quad t = 0, \dots, N-1 \tag{7b}$$
$$z_{t+1} = x_{t+1}^{(t+1)} \quad t = 0, \dots, N-1 \tag{7c}$$
$$\sigma_t = d - C x_t^{(t)} - D u_t^{(t)} \quad t = 0, \dots, N \tag{7d}$$
$$x_0^{(0)} = x_{\text{init}}, \tag{7e}$$

where, defining $\mathbb{C} = \{\sigma \in \mathbb{R}^p \,|\, \sigma \geq 0\}$, the indicator functions $\mathcal{I}_{\mathbb{C}}(\sigma_t)$ has been added to the cost (5a) to penalize the inequality constraint violations (5d), and $Q_t = Q$ for $t = 0, \dots, N-1$ and $Q_t = P$ for $t = N$. Note that if the consensus constraints are satisfied, Problem (7) is equivalent to Problem (5). Hence, this implies that computing N for Problem (7) is equivalent to computing N for Problem (5). Consequently, the results of Theorem 1 hold and we can rely on the existence of a finite value $N^\infty \leq N$, such that x_{N^∞} is in the terminal set X_f.

In the following, first, we show that we can use FAMA to solve Problem (7). Second, we present our proposed algorithm to compute N and a solution for Problem (5).

Algorithm 3 FAMA for Problem (7).

Given x_{init}, N_{init}, H_1, H_2, $\mathcal{Q}$, G, g, τ, η, and $\bar{k}$, $\mathcal{X}_f$, K, and $\tilde{H}$.
Set $N^0 = N_{\text{init}}$, $\alpha^0 = 1$, $\alpha^1 = (1+\sqrt{5})/2$, $\mu_t^0 = \hat{\mu}_t^1 = \mu_t^{\text{start}}$.
for $k = 0, \ldots, \bar{k}$ **do**

1.a For $t = 0$, $u_0^{k+1} = \operatorname{argmin}_{u_0} \mathcal{L}_0$ and $y_0^{k+1} := \left[x_{\text{init}}^{\mathrm{T}} \; u_0^{k+1^{\mathrm{T}}}\right]^{\mathrm{T}}$.

1.b For $t = 1, \ldots, N^{k-1}-1$, $y_t^{k+1} = \operatorname{argmin}_{y_t} \mathcal{L}_t$.

1.c For $t = N^s$, $x_t^{k+1} = \operatorname{argmin}_{x_t} \mathcal{L}_N$ and $y_t^{k+1} := \left[x_t^{k+1^{\mathrm{T}}} \; 0\right]^{\mathrm{T}}$.

2. Compute $\alpha^{k+1} = \frac{1+\sqrt{4\alpha^k{}^2+1}}{2}$.

3. For $t = 0, \ldots, N^s$, $\sigma_t^{k+1} = \mathbf{Pr}_{\mathbb{C}}\left(G\hat{y}_t^{k+1} - d - \frac{1}{\tau}\hat{\lambda}_t^k\right)$.

4. For $t = 1, \ldots, N^s$, $z_t^{k+1} = \frac{H_1\hat{y}_t^{k+1} + H_2\hat{y}_{t-1}^{k+1}}{2} - \frac{\hat{w}_t^k + \hat{v}_t^k}{2\tau}$.

5 For $t = 0, \ldots, N^s$ compute

$$\mu_t^{k+1} = \hat{\mu}_t^k - \tau\tilde{H}\begin{bmatrix} y_{t-1}^{k+1} \\ y_t^{k+1} \\ z_t^{k+1} \\ \sigma_t^{k+1} \end{bmatrix} + \tau\begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix},$$

$$\hat{\mu}_t^{k+1} = \mu_t^{k+1} + \frac{\alpha^k - 1}{\alpha^{k+1}}\left(\mu_t^{k+1} - \mu_t^k\right).$$

end for

Finally, we show that the control sequence obtained using the proposed algorithm is an optimal solution of Problem (3).

Let $f(\mathbf{x}, \mathbf{u}) = \sum_{t=0}^{N} x_t^{(t)^{\mathrm{T}}} Q_t x_t^{(t)} + u_t^{(t)^{\mathrm{T}}} R u_t^{(t)}$ and $g(\boldsymbol{\sigma}, \boldsymbol{z}) = \sum_{t=0}^{N} \mathcal{I}_{\mathbb{C}}(\sigma_t)$. If Assumption 2 is satisfied, $f(\mathbf{x}, \mathbf{u})$ satisfies Assumption 3 with $\sigma_f = \operatorname{eig}_{\min}(\operatorname{blockdiag}\{Q, R\})$. In addition, given that $\mathbb{C}$ is a convex set and the associated indicator function is convex, $g(\boldsymbol{\sigma}, \boldsymbol{z})$ satisfies Assumption 4. Hence, we can rely on FAMA to solve Problem (7). In particular, given that FAMA operates on the dual space, we formulate the dual of Problem (7) as follows:

$$\underset{\boldsymbol{\mu}}{\text{maximize}}\; D(\boldsymbol{\mu}), \tag{8}$$

where the dual function is defined as follows:

$$\mathbf{D}(\boldsymbol{\mu}) = \min_{\mathbf{y},\boldsymbol{\sigma},\mathbf{z}} f(\boldsymbol{y}) + g(\boldsymbol{\sigma}, \boldsymbol{z}) + \sum_{t=0}^{N} \langle \lambda_t, -Gy_t - \sigma_t + d\rangle + \sum_{t=1}^{N} \langle w_t, z_t - H_1 y_t\rangle + \langle v_t, z_t - H_2 y_{t-1}\rangle,$$

$\boldsymbol{\mu}^{\mathrm{T}} = [w_1^{\mathrm{T}}, \ldots, w_N^{\mathrm{T}}, v_1^{\mathrm{T}}, \ldots, v_N^{\mathrm{T}}, \lambda_0^{\mathrm{T}}, \ldots, \lambda_N^{\mathrm{T}}]$, $\mathbf{y}^{\mathrm{T}} = [y_0^{\mathrm{T}}, \ldots, y_N^{\mathrm{T}}]$, $y_t^{\mathrm{T}} = [x_t^{(t)^{\mathrm{T}}}, u_t^{(t)^{\mathrm{T}}}]$, $\boldsymbol{\sigma}^{\mathrm{T}} = [\sigma_0^{\mathrm{T}}, \ldots, \sigma_N^{\mathrm{T}}]$, $\mathbf{z}^{\mathrm{T}} = [z_1^{\mathrm{T}}, \ldots, z_N^{\mathrm{T}}]$, $H_1 = [I_n \; 0]$, $H_2 = [A \; B]$, $G = [C \; D]$.

Algorithm 4 details our strategy to estimate the solution of Problem (7) and the length of the horizon N. In particular, Algorithm 4 relies on Algorithm 3, which is Algorithm 2 applied to Problem (7), to compute the primal and dual variables (step 2). In particular, (step 1 of Algorithm 3) $\mathcal{L}_t$ is the Lagrangian associated with Problem (7) and defined as follows:

$$\mathcal{L}_t = \min\; f(y_t) + g(\sigma_t, z_t) + \langle \lambda_t, -Gy_t - \sigma_t + d\rangle + \langle w_t, z_t - H_1 y_t\rangle + \langle v_{t+1}, z_{t+1} - H_2 y_t\rangle.$$

Furthermore, (step 5 of Algorithm 3) $\tilde{H}$ is the matrix associated with the multiplier update, i.e.,

$$\tilde{H} := \begin{bmatrix} 0 & H_1 & -I_n & 0 \\ H_2 & 0 & -I_n & 0 \\ 0 & -G & 0 & -I_p \end{bmatrix}.$$

For $t = 0$ only λ_0 is updated, given that z_t, w_t, and v_t are defined for $t = 1, \ldots, N$.

Algorithm 4 CLQR for Problem (7).

Given x_{init}, N_{init}, H_1, H_2, Q, R, P, G, g, τ, η, and $\bar{k}$, $\mathcal{X}_f$, K, and $\tilde{H}$.
Set $N^0 = N_{\text{init}}$, $\alpha^0 = 1$, $\alpha^1 = (1+\sqrt{5})/2$, $\mu_t^0 = \hat{\mu}_t^1 = \mu_t^{\text{start}}$, and $s = 0$.
if $x_{\text{init}} \in \mathcal{X}_f$ **then**
1. $u^\infty = Kx_{\text{init}}$, $N = 0$.
else
while Termination criteria are not met **do**
2. $\left[\mathbf{y}^{s+1}, \hat{\boldsymbol{\mu}}^{s+1}, \boldsymbol{\mu}^{s+1}, \alpha^{s+1}\right] = \texttt{fama}\left(\hat{\boldsymbol{\mu}}^s, \boldsymbol{\mu}^s, \alpha^s, N^s, \bar{k}\right)$.
if $x_N^{s+1} \in \mathcal{X}_f$ **then**
3. $N^{s+1} = N^s - 1$.
4. Remove Subproblem N^s.
else
5. $N^{s+1} = N^s + 1$.
6. Add Subproblem N^{s+1}.
end if
7. $s = s + 1$.
end while
8. $u^\infty = u_0$, $N = N^s$.
end if

For $t = 0$ only λ_0 is updated, given that z_t, w_t, and v_t are defined for $t = 1, \ldots, N$.

It is evident that compared to Algorithm 3, Algorithm 4 has an additional `if` condition used to check every $\bar{k} \geq 1$ iterations whether $x_{N^s} \in \mathcal{X}_f$. If $x_0 \in \mathcal{X}_f$ the algorithm terminates immediately (step 1). Otherwise, the algorithm terminates only when $x_{N^s} \in \mathcal{X}_f$ and $\{\mathbf{y}, \boldsymbol{\mu}\}$ returned by Algorithm 3 reaches a desired accuracy. If $x_{N^s} \in \mathcal{X}_f$, the algorithm decreases N^s. From the splitting perspective, this means that the last subproblem is removed (steps 3 and 4). Note that removing a subproblem with its associated dual variables does not compromise the future updates of the remaining subproblems. If $x_{N^s} \notin \mathcal{X}_f$, N^s increases by 1 with respect to the previous iterate (steps 5 and 6). From the splitting perspective, this means that a new subproblem (of the same dimension as the previous ones) is added.

Remark 1. In theory, we can set $\bar{k} = 1$, i.e., the algorithm checks the state of the last subproblem at every iteration. In practice, we noticed that checking the state of the last subproblem at every iteration affects the convergence of N^s to N^∞ given that N^s oscillates around N^∞ requiring an higher number of iterations. If we allow larger $\bar{k}$ the oscillations disappear and N^s converges faster to N^∞. From the FAMA perspective, a larger $\bar{k}$ means that at each outer iteration s of Algorithm 4, Problem (7) is solved up to a given accuracy (which depends on $\bar{k}$). Then, the quality of the estimates is refined every $\bar{k}$ iterations together with N^s.

Remark 2. Step 8 of Algorithm 4 can be modified to achieve a tighter upper bound on N^∞. In particular, by using $\mathbf{u}_N = \{u_0^{(0)}, \ldots, u_{N-1}^{(N-1)}, Kx_N\}$ we can compute $x_N^{(0)}$. Then, while $Cx_N^{(0)} + Du_N^{(0)} < d$, $N \leftarrow N-1$. The first time the constraints are active, the algorithm terminates. This does not affect the computational time of the algorithm (given that the solution

has been already computed), but can improve the initial guess on the length of the horizon for the next problem instance in a closed-loop implementation.

According to [13] the following result concerning the quality of the primal estimates holds:

Theorem 2. *Consider Problem* (7)*. Let* $\{\mathbf{y}^k\}$ *and* $\{\boldsymbol{\mu}^k\}$ *be generated by Algorithm 4. If Assumption 2 is satisfied, then, for any* $s \geq 0$ *and* $\bar{k} \geq 0$*, the following holds:*

$$\mathbf{D}(\boldsymbol{\mu}^*) - \mathbf{D}(\boldsymbol{\mu}^s) \leq \frac{2\,\mathrm{eig}_{\max}(H_{\mathbf{y}})}{\sigma_f(s\bar{k}+1)^2}\|\boldsymbol{\mu}^0 - \boldsymbol{\mu}^*\|^2,$$

where $\boldsymbol{\mu}^0$ *and* $\boldsymbol{\mu}^*$ *are the initial and optimal values of multipliers, respectively, and* $H_{\mathbf{y}}$ *is defined as follows:*

$$H_{\mathbf{y}} := \mathrm{blockdiag}\{\underbrace{H_1,\ldots,H_1}_{N},\underbrace{H_2,\ldots,H_2}_{N},\underbrace{-G,\ldots,-G}_{(N+1)}\},$$

If $\lambda_t^0 \in \mathbb{C}$ *(*$t = 0,\ldots,N$*) and* $\boldsymbol{y}^0$ *is such that the consensus constraints are satisfied, then the dual iterates will remain feasible for all* $k \geq 1$ *and*

$$\|\mathbf{y}^s - \mathbf{y}^*\|^2 \leq \frac{4\,\mathrm{eig}_{\max}(H_{\mathbf{y}})}{\sigma_f(s\bar{k}+1)^2}\|\boldsymbol{\mu}^0 - \boldsymbol{\mu}^*\|^2. \tag{9}$$

Proof. The proof follows from the one of Theorem 5.3 in [13] applied to Problem (7). □

Then, the following result holds:

Theorem 3. *Consider Problem* (7)*. Let* N *and* $\mathbf{u}_N = \{u_0^{(0)},\ldots,u_{N-1}^{(N-1)}, Kx_N\}$ *be generated by Algorithm 4. Under the same assumptions of Theorem 2 then, for all* $s \geq 1$ *and* $\bar{k} \geq 1$*, there exists* $N^\infty \leq N$ *finite such that*

$$(\mathbf{u}_{N^\infty}^s - \bar{\mathbf{u}}^\infty)^{\mathrm{T}}\tilde{B}(\mathbf{u}_{N^\infty}^s - \bar{\mathbf{u}}^\infty) \leq \frac{4\,\mathrm{eig}_{\max}(H_{\mathbf{y}})}{\sigma_f(s\bar{k}+1)^2}\|\boldsymbol{\mu}^0 - \boldsymbol{\mu}^*\|^2, \tag{10}$$

where $\bar{\mathbf{u}}^\infty = \{u_0,\ldots,u_{N^\infty}\}$ *is the (truncated) solution of the CLQR Problem and* $\tilde{B}$ *is defined as follows:*

$$\tilde{B} := \begin{bmatrix} I_m & 0 & \ldots & 0 \\ B & I_m & & 0 \\ \vdots & \vdots & \ddots & 0 \\ A^{N^\infty-1}B & A^{N^\infty-2}B & \ldots & I_m \end{bmatrix}.$$

Proof. Given that we initialize $\mathbf{y}^0$ to achieve consensus, according to Theorem 2, $\mathbf{z}^s$ will remain in consensus, and solving Problem (7) becomes equivalent to solving Problem (5). Hence, we can use the results of Theorem 1 to show that N^∞ is finite. Concerning the inequality above, first, note that $x_t^s - x_i^\infty = Ax_0 + A^{t-1}Bu_0^s + \ldots + Bu_{t-1}^s - Ax_0 - A^{t-1}Bu_0^\infty - \ldots - Bu_{i-1}^\infty = A^{t-1}B(u_0^s - u_0^\infty) + \ldots + B(u_{t-1}^s - u_{t-1}^\infty)$. Second, according to Theorem 1, $\mathbf{u}^* = \mathbf{u}^\infty$. Hence, the inequality (10) follows directly from (9). Third, for $s\bar{k} \to \infty$, we can conclude that the the control sequence obtained using Algorithm 4 converges to the CLQR control law obtained by solving Problem (3). □

Remark 3. Note that Theorem 3 considers the truncated sequences $\mathbf{u}_N$ and $\bar{\mathbf{u}}^\infty$ for practical reasons, given that, after N^∞ steps the control commands are obtained using the LQR gain K, i.e., are identical for both sequences.

Initializing the new values of the multipliers is important to satisfy the assumptions of the theorems above. In the estimation phase of the horizon length, when N^s increases then setting $z_{N^s}^s := H_2 y_{N^s-1}^s$ (i.e., to maintain consensus between the former last subproblem and the new last subproblem) allows one to initialize $w_{N^k}^{k-1} = v_{N^s}^{s-1} = 0$. Concerning $\lambda_{N^s}^{s-1}$, any value such that $\lambda_{N^s}^{s-1} \in \mathbb{C}$ (according to Lemmas 1 and 2) can be used (e.g., $\lambda_{N^s}^{s-1} = \lambda_{N^s-1}^{s-1}$).

In [12] splitting strategies are also used to estimate N. Compared to [12], we use a different strategy to compute the length of the prediction horizon online. First, we do not propagate the dynamics forward at each iterate (until a stable value of $N \geq N^\infty$ is reached), but we check whether the state of the last subproblem is in $\mathcal{X}_f$ (an inexpensive operation). Second, thanks to the time splitting, our algorithm allows one to decrease the length of the horizon online, while the approach proposed in [12] allows the horizon value only to increase from its initial guess. Specifically, if our initial guess is too conservative, Algorithm 4 starts removing the tail subproblems. Removing subproblems implies removing dual variables that could, in general, affect the future updates of the algorithm, such as in [12]. This is not the case for the time splitting. The dual variables of the subproblems are independent of each other and removing one of them (on the tail of the horizon) does not compromise the future updates of the others.

Our approach relies on the results of [9]. In [9], however, the steps of the algorithm are more involved from the computational point of view. Step 1 of Algorithm 1 requires the solution of a constrained QP of dimension proportional to the length of the prediction horizon. Recursively solving a constrained QP can be extremely time-consuming, especially when the computational resources are limited, such as, in embedded applications. Our design, thanks to the time splitting, only solves unconstrained least-squares problems and simple gradient calculations. Furthermore, an increase in the length of the horizon does not change the dimension of the subproblems, given that their size is independent of the length of the prediction horizon.

The proposed algorithm can be, in principle, fully parallelized. If N independent workers are available, each of them can be dedicated to a subproblem. The workers communicate with their neighbors only at given time instances to exchange information concerning the consensus variables. When the number of subproblems is large and the number of workers is smaller than N asynchronous update strategies can be beneficial to improve the performance of the algorithm. Investigation of asynchronous update strategies is part of our future work.

Remark 4. According to Theorem 3, Algorithm 4 returns an optimal solution for Problem (3) for $s\bar{k} \to \infty$. In practical implementations, the algorithm terminates after a fixed number of iterations. In this scenario, constraint tightening techniques can be used to enforce the feasibility of the primal

estimates (e.g., [14]).

V. NUMERICAL EXAMPLE

We tested our design on the system proposed in [12]. The system is described by the following matrices:

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix}. \qquad (11)$$

The state and the control input are constrained in $\mathcal{X} := \{x \in \mathbb{R}^n \,|\, \|x\|_\infty \leq 10\}$ and $\mathcal{U} := \{u \in \mathbb{R}^m \,|\, \|u\|_\infty \leq 1\}$, respectively. The matrices Q and R are the identity matrices to satisfy Assumption 2. We computed offline the maximal positively invariant set $\mathcal{X}_f^{\max}$ of the closed loop associated with the LQR controller with weighting matrices Q and R according to [15]. Then, we selected X_f to be a tightened subset of $\mathcal{X}_f^{\max}$ to take into account the early termination of the solver. In particular, we tightened the terminal set by a quantity $\epsilon = 10^{-3}$ proportional to the selected termination criterion for the algorithm $\|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2 \leq \epsilon$. Then, we tuned $\tau = 0.0726$ and set $\bar{k} = 1000$ for $s = 0$ and $\bar{k} = 1$ for $s > 0$.

We tested Algorithm 4 for 1592 different initial conditions uniformly sampled from $\mathcal{X}$. For each initial condition, we ran the proposed algorithm and, at the end of each simulation, we tested whether the state $x_N = A^N x_0 + A^{N-1} B u_0^{(0)} + \ldots + B u_{N-1}^{(N-1)}$ was in $\mathcal{X}_f$. For practical reasons, we terminated the simulation if convergence to the suboptimal solution was not achieved within $k_{\max} = 10^5$ iterations.

First, we compared the behavior of the algorithm without and with the backtracking of the horizon length described in Remark 2. Figure 1 compares the horizon length obtained in the two scenarios using $N^0 = 20$ as initial guess on the horizon length. The proposed backtracking strategy reduces the conservatism in the value of N. As part of our future work, we plan to investigate an online strategy to reduce the horizon length in steps 3-6 of Algorithm 4 (for example, by replacing the `if` condition with a `while`).

Second, we compared the behavior of Algorithm 4 with backtracking for three different initial guesses on the horizon length, i.e., $N^0 \in \{2, 8, 20\}$. Figure 2 shows the distributions of the estimated values of N for the three aforementioned scenarios. Furthermore, the vertical dashed lines in Figure 2 represent the mean value of the horizon length computed using the proposed algorithm. Note that we removed from the plots the initial conditions that were inside the terminal set and the initial conditions that lead the algorithm to terminate after $k_{\max}$ iterations. Hence, based on this selection on the initial states, we noticed that for $N^0 = 2$ Algorithm 4 terminates within $k_{\max}$ in 238 cases, for $N^0 = 8$ it terminates in 818 cases, and for $N^0 = 20$ it terminates in 1592 cases. Note that by using $N^0 = 2$ we are only able to converge within $k_{\max}$ iterations only when the optimal horizon length is close to 2. This observation confirms the benefits of warm-starting the horizon length. Figure 3 shows the average number of iterations $s\bar{k}$ needed to compute $N \geq N^\infty$ and an optimal solution for Problem (3). Note that, warm starting the length of the horizon does not compromise the number of

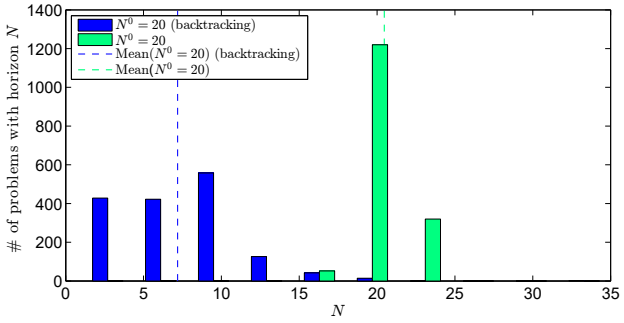

Fig. 1: Comparison of the values of $N \geq N^\infty$ obtained using Algorithm 4 without and with backtracking, respectively.

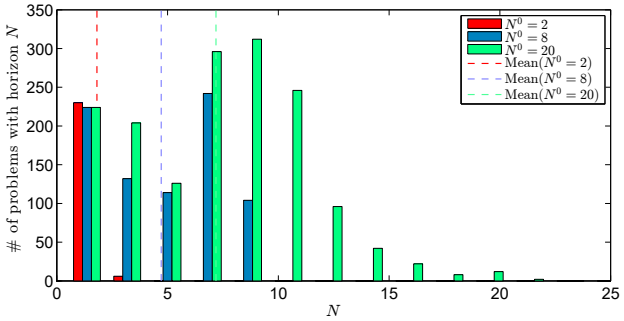

Fig. 2: Estimated horizon length N computed using Algorithm 4 with backtracking starting from different initial conditions in $\mathcal{X}$ using different values of N^0 to initialize the algorithm.

iteration needed by the algorithm to converge to $N \leq N^0$. We believe that the current values can be improved if the back tracking is implemented along with the computation of the primal and dual variables.

VI. CONCLUSIONS

This paper proposes an alternative approach to solve the constrained linear quadratic regulator (CLQR) problem using operator splitting techniques. The original CLQR problem is reformulated as an MPC problem, whose horizon length is estimated online for each initial condition. We show that the solution obtained using our proposed algorithm is an optimal solution of the CLQR problem and that the horizon length is bounded. Finally, we tested our design on a planar system to show the advantages of the proposed technique that allows to reduce the number of iterations needed to achieve an optimal solution for the CLQR problem thanks to the warm starting of the horizon length.

As part of our future work, we plan investigate the possibility of asynchronous updates. Furthermore, we plan to test the proposed algorithm on a practical application.

REFERENCES

[1] R. E. Kalman, "Contribution to the theory of optimal control", Boletin de la Sociedad Matematica Mexicana, 1960.

[2] J. Richalet et al, "Model predictive heuristic control: Applications to industrial processes", *Automatica*, vol. 14, n. 5, pp. 413–428, 1978.

[3] C.R. Cutler and B.L. Ramaker, "Dynamic matrix control–a computer control algorithm", Proc. of the *Joint ACC*, 1980.

[4] J. M. Maciejowski, "Predictive control: with constraints", *Pearson education*, 2002.

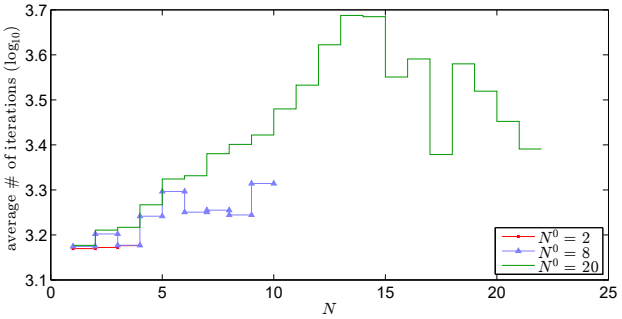

Fig. 3: Average number of iterations of Algorithm 4 obtained for each initial condition with respect to the length of the horizon N using different values of N^0 to initialize the algorithm.

[5] F. Borrelli, A. Bemporad, and M. Morari, "Predictive Control for linear and hybrid systems", 2015, `http://www.mpc.berkeley.edu/mpc-course-material`

[6] D. Q. Mayne et al, "Constrained model predictive control: Stability and optimality", *Automatica*, vol. 36, n. 6, pp. 789–814, 2000.

[7] 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

[8] 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.

[9] P. O. M. Scokaert and J. B. Rawlings, "Constrained Linear Quadratic Regulation", *IEEE TAC*, vol. 43, n. 8, pp. 1163–1169, 1998.

[10] 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.

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

[12] G. Stathopoulos, M. Korda, and C. N. Jones, "Solving the Infinite-horizon Constrained LQR Problem using Accelerated Dual Proximal Methods", accepted for publication in *IEEE TAC*.

[13] 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.

[14] M. Rubagotti, P. Patrinos, A. Bemporad, "Stabilizing linear model predictive control under inexact numerical optimization", *IEEE Transactions on Automatic Control*, vol. 59, n. 6, pp. 1660-1666, 2014.

[15] E. G. Gilbert and K. Tin Tan. "Linear systems with state and control constraints: The theory and application of maximal output admissible sets", *IEEE TAC*, vol. 36, n.9, pp. 1008–1020, 1991.