Solving the Infinite-horizon Constrained LQR Problem using Accelerated Dual Proximal Methods

Giorgos Stathopoulos, *Member, IEEE,* Milan Korda, *Member, IEEE,* and Colin N. Jones, *Member, IEEE.*

***Abstract*—This work presents an algorithmic scheme for solving the infinite-time constrained linear quadratic regulation problem. We employ an accelerated version of a popular proximal gradient scheme, commonly known as the Forward-Backward Splitting (FBS), and prove its convergence to the optimal solution in our infinite-dimensional setting. Each iteration of the algorithm requires only finite memory, is computationally cheap, and makes no use of terminal invariant sets; hence, the algorithm can be applied to systems of very large dimensions. The acceleration brings in 'optimal' convergence rates $O(1/k^2)$ for function values and $O(1/k)$ for primal iterates and renders the proposed method a practical alternative to model predictive control schemes for setpoint tracking. In addition, for the case when the true system is subject to disturbances or modelling errors, we propose an efficient warm-starting procedure, which significantly reduces the number of iterations when the algorithm is applied in closed-loop. Numerical examples demonstrate the approach.**

***Index Terms*—Constrained LQR, Alternating minimization, Operator splitting**

I. Introduction

An important extension of the famous result of [1] on the closed form solution of the infinite-horizon linear quadratic regulation (LQR) problem is the case where input and state variables are constrained. This problem is computationally significantly more difficult and has been by and large addressed only approximately. A prime example of an approximation scheme is *model predictive control (MPC)*, which approximates the infinite-time constrained problem by a finite-time one. Stability of such MPC controllers is then typically enforced by adding a suitable terminal constraint and a terminal penalty. The inclusion of a terminal constraint limits the feasible region of the MPC controller, and, consequently, the region of attraction of the closed-loop system. In practical applications, this problem is typically overcome by simply choosing a "sufficiently" long horizon based on process insight (e.g., dominant time constant). Closed-loop behavior is then analyzed a posteriori, for instance by exhaustive simulation.

There have been few results addressing directly the *infinite-horizon constrained LQR (CLQR)* problem. Among the most well-known efforts are the works [2], [3] and [4]. The authors of [2] suggest a scheme for offline computation of a sufficient horizon length. The solution of the corresponding quadratic program (QP) is then equivalent to the original infinite dimensional problem. The reported results are somewhat conservative, while the offline part of the proposed algorithm can be computationally prohibitive since it involves the solution of a possibly nonconvex problem, the computation of a positively invariant set as well as a vertex enumeration problem. The authors of [3] extend the work of [5] by solving a sequence of QPs of finite horizon length, which is monotonically non-decreasing. After each QP has been solved, the inclusion of the final state in a positively invariant set associated to the optimal unconstrained LQ controller is checked; if the final state is not included in the set, the horizon was insufficient and has to be increased. Finally, the authors in [4] employ parametric quadratic programming and a reachability analysis approach to compute the least conservative horizon length that ensures optimal infinite horizon performance. Although this work provides the exact necessary horizon length for the feasible set of initial states, it suffers from tractability issues since it is based on state-space partitioning and thus can only be applied to small systems.

Our approach is inspired by the framework of *proximal gradient methods*, a class of algorithms known for their simplicity when it comes to solving convex optimization problems with simple constraints [6]. From this family of algorithms, we use the *(accelerated) version of the Forward-Backward Splitting method ((A)FBS)* (see, *e.g.* [7, Chapter 25] and [8]). The idea is to condense the problem and describe it in terms of its input variables, then formulate the dual of the corresponding (infinite-dimensional) QP, and apply the (A)FBS method to solve it. More specifically, the method decomposes the QP into two subproblems, the first one being an infinite-dimensional least squares problem and the second one a simple clipping of an infinite sequence to the non-positive orthant. The subproblems are solved repeatedly (with the solution of one influencing the cost function of the other) until convergence to the solution of the original problem. This is in contrast to the approach of [3], which requires the solution of a sequence of *constrained* QPs. We show that both subproblems of the proposed algorithm can be solved tractably (which is not a priori obvious since we are working with infinite sequences), the first one by solving a finite-dimensional system of linear equations (with the possibility to pre-factorize the matrices) and the second one by simple clipping of finitely many real numbers on the non-positive real line. Convergence of the scheme (with rate $O(1/k^2)$ for function values and $O(1/k)$ for primal iterates) to the optimal infinite-horizon sequence is guaranteed under mild assumptions. Therefore the proposed algorithmic scheme provides a means to compute the solution of the infinite-horizon constrained LQR problem with guaranteed convergence.

This work is based on our recent result in [9], where the same problem is solved by employing a primal-dual scheme, the *Alternating Minimization Algorithm (AMA)* [10], which is equivalent to FBS. The approach is equivalent to splitting

The authors are with the the Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. E-mail: {georgios.stathopoulos,milan.korda,colin.jones}@epfl.ch .

the infinite-horizon constrained LQR problem into an *unconstrained* LQR problem and a proximal minimization problem. In the current work, the result is developed in both theoretical and practical directions. From the theoretical perspective, we (i) drop the limiting assumption for positive-definiteness of the state penalty matrix Q that existed in [9], and, more importantly, we (ii) prove convergence of an accelerated counterpart of the original FBS method, a fact that enables a (significantly) faster convergence rate. From the practical perspective, we provide a fully implementable method, competitive for real-time control. We (i) eliminate the need for knowledge of uncomputable quantities, (ii) propose computationally efficient ways to solve the optimization subproblems and (iii) propose a warm-starting scheme that performs well in practice.

The paper is organized as follows: In Section II we introduce the problem of interest. In Section III the problem is expressed in terms of its dual variables by means of the proximal splitting framework. In Section IV we present the accelerated forward-backward splitting algorithm to solve the problem and show that each iteration of the algorithm can be carried out tractably. The convergence proofs for this scheme are given in Section V, while Section VI discusses the computational aspects that make the algorithm practical to use. Section VII presents two numerical examples: A toy example of an unstable system with two states and one control input illustrates the main features of the algorithm. Subsequently, we demonstrate the practical applicability of the algorithm on a linearized model of a quadcopter with 12 states, 4 inputs and polytopic constraints. Finally, Appendices A and B provide the proofs for the results presented in Section V.

II. Problem statement

The constrained regulation problem for a linear time invariant (LTI) system can be written in the general form

$$\begin{array}{ll} \text{minimize} & (1/2)\sum_{i=0}^{\infty} x_i^\top Q x_i + u_i^\top R u_i \\ \text{subject to} & x_{i+1} = Ax_i + Bu_i, \quad i \in \mathbb{N} \\ & x_0 = x_{\text{init}} \\ & C_x x_i \leq c_x \\ & C_u u_i \leq c_u \ , \end{array} \tag{1}$$

with variables $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$, and data $c_x \in \mathbb{R}^{p_x}$ and $c_u \in \mathbb{R}^{p_u}$.

We make the following standing assumptions:

Assumption 1: The pair (A, B) is stabilizable, the optimal value of problem (1) is finite, the set

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid C_x x \leq c_x\}$$

contains the origin in its interior, the matrix C_u has full column rank, the matrix Q is positive semidefinite and R is positive definite.

Remark 1 (Stability): Clearly, under Assumption 1, if the problem (1) is feasible and the pair $(A, \sqrt{Q})$ detectable, then the control sequence optimal in (1) is stabilizing. Therefore, there is no need to enforce stability in an ad hoc way as is commonly done when the infinite-time problem (1) is approximated by a finite-time one solved in a receding horizon fashion.

From now on we write infinite sequences and infinite-dimensional operators in bold font.

The problem can be rewritten in the dense form, *i.e.*, by writing the states as functions of the inputs. This is done by defining the operators

$$\boldsymbol{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} B & 0 & 0 & \cdots \\ AB & B & 0 & \cdots \\ A^2B & AB & B & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} ,$$

$$\boldsymbol{Q} = \mathbf{diag}(Q, Q, \ldots) \ , \quad \boldsymbol{R} = \mathbf{diag}(R, R, \ldots) \ ,$$

$$\boldsymbol{C_x} = \mathbf{diag}(C_x, C_x, \ldots) \ , \quad \boldsymbol{c_x} = (c_x, c_x, \ldots) \ ,$$

$$\boldsymbol{C}_i = \begin{bmatrix} \boldsymbol{e}_i \otimes C_u \\ [\boldsymbol{C_x B}]_i \end{bmatrix}, \quad c_i = \begin{bmatrix} c_u \\ c_x - C_x A^i x_{\text{init}} \end{bmatrix} ,$$

$$\boldsymbol{H} = \boldsymbol{B}^\star \boldsymbol{Q} \boldsymbol{B} + \boldsymbol{R}, \quad \boldsymbol{G} = \boldsymbol{A}^\star \boldsymbol{Q} \boldsymbol{B}, \quad \bar{F} = \boldsymbol{A}^\star \boldsymbol{Q} \boldsymbol{A} + Q \ ,$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_1^\star & \boldsymbol{C}_2^\star & \ldots \end{bmatrix}^\star, \quad \boldsymbol{c} = [c_1^\star, c_2^\star, \ldots]^\star \ , \tag{2}$$

where we denote by $\mathbf{diag}(\cdot, \cdot, \ldots)$ the operator that forms a block diagonal matrix of the provided arguments, $\boldsymbol{e}_i$ the (infinite dimensional) row vector with only one nonzero element equal to one at position i, by $[\cdot]_i$ the i^{th} block row of size $p_x \times \infty$ of the corresponding operator and by * the adjoint of an operator (i.e., the infinite-dimensional analogue of transpose; see Appendix A for a brief introduction to operators). Using the above, (1) can be written in the form

$$\begin{array}{ll} \text{minimize} & (1/2)\boldsymbol{u}^\star \boldsymbol{H} \boldsymbol{u} + \boldsymbol{h}^\star \boldsymbol{u} + r \\ \text{subject to} & \boldsymbol{C}\boldsymbol{u} \leq \boldsymbol{c} \ , \end{array} \tag{3}$$

where $\boldsymbol{h} = \boldsymbol{G}^\star x_{\text{init}}$, $r = (1/2) x_{\text{init}}^\star \bar{F} x_{\text{init}}$, $\boldsymbol{u}$ is the infinite sequence $\boldsymbol{u} := [u_0^\star, u_1^\star, \ldots]^\star$ and $\boldsymbol{H} \colon \mathcal{H}_{\boldsymbol{u}} \to \mathcal{H}_{\boldsymbol{u}}$, $\boldsymbol{C} \colon \mathcal{H}_{\boldsymbol{\lambda}} \to \mathcal{H}_{\boldsymbol{u}}$, $\boldsymbol{c} \in \mathcal{H}_{\boldsymbol{\lambda}}$, where $\mathcal{H}$'s are suitable Hilbert spaces specified next: Any sequence

$$\boldsymbol{z} := (z_0, z_1, \ldots)$$

is viewed as an element of an l^2-weighted (or l_w^2) real Hilbert space $\mathcal{H}_{\boldsymbol{z}}$ (see Appendix A, Definition 1) induced by the inner product

$$\langle \boldsymbol{z}, \boldsymbol{y} \rangle = \sum_{i=0}^{\infty} w^i z_i^\top y_i \ , \quad \forall \, \boldsymbol{y} \in \mathcal{H}_{\boldsymbol{z}}, \, \boldsymbol{z} \in \mathcal{H}_{\boldsymbol{z}} \ , \tag{4}$$

where w is an appropriately chosen weight (see Appendix A, Definition 1). The norm of any $\boldsymbol{z} \in \mathcal{H}_{\boldsymbol{z}}$ is thus given by

$$\|\boldsymbol{z}\|_{\mathcal{H}_{\boldsymbol{z}}} := \sqrt{\langle \boldsymbol{z}, \boldsymbol{z} \rangle} = \sqrt{\sum_{i=0}^{\infty} w^i \|z_i\|_2^2} \ .$$

Unless stated otherwise, for the rest of the paper by a Hilbert space we mean the l_w^2 real Hilbert space as just introduced.

III. DUALIZATION

Before proceeding to the reformulation of problem (3), we revisit some material related to proximal algorithms and duality. To this end, we denote the family of functions $F : \mathcal{H}_{\boldsymbol{z}} \to \overline{\mathbb{R}}$ that are closed, proper and convex by $F \in \Gamma_0(\mathcal{H}_{\boldsymbol{z}})$, where $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$. For $F \in \Gamma_0(\mathcal{H}_{\boldsymbol{z}})$, its *proximal operator* $\mathbf{prox}_F : \mathcal{H}_{\boldsymbol{z}} \to \mathcal{H}_{\boldsymbol{z}}$ is defined as

$$\mathbf{prox}_F(y) := \operatorname*{argmin}_z \left\{ f(z) + (1/2)\|z - y\|^2 \right\} \quad . \tag{5}$$

The operator is evaluated at a given point y and looks for a minimizer that makes a compromise between the minimizer of the function F and the point y.

Consider, now, the case that we want to minimize $F(z) = f(z) + g(z)$, $f \in \Gamma_0(\mathcal{H}_{\boldsymbol{z}})$ is *differentiable* and $g \in \Gamma_0(\mathcal{H}_{\boldsymbol{z}})$. The proximal gradient method is the iteration

$$z^{k+1} := \mathbf{prox}_{\rho^k g}\left(z^k - \rho^k \nabla f(z^k)\right) \quad . \tag{6}$$

The algorithm converges for $\rho^k > 0$, either constant or determined by line search methods. The proximal gradient method is one among many proximal algorithms that enable the solution of a composite convex optimization problem using only function and gradient evaluations. These algorithms have recently gained more ground in the control community due to their simplicity (see, *e.g.*, [11]). The work [6] provides a comprehensive treatment of these methods.

Our purpose is to use (an accelerated variant of) the proximal gradient method on the dual of problem (3). This can be derived by making use of conjugate duality (see, *e.g.*, Chapter 7, [12]). By defining

$$f(\boldsymbol{u}) = (1/2)\boldsymbol{u}^\star \boldsymbol{H}\boldsymbol{u} + \boldsymbol{h}^\star \boldsymbol{u} + r, \quad g(\boldsymbol{C}\boldsymbol{u} - \boldsymbol{c}) = \delta_-(\boldsymbol{C}\boldsymbol{u} - \boldsymbol{c}) \tag{7}$$

and $\delta_-(\cdot)$ being the indicator function for the nonpositive orthant, we can rewrite (3) as

$$\min_{\boldsymbol{u}} \{ f(\boldsymbol{u}) + g(\boldsymbol{C}\boldsymbol{u} - \boldsymbol{c}) \}$$

and then express the Lagrange dual problem by using the dual variable $\boldsymbol{\lambda}$ as

$$\begin{aligned}
&\max_{\boldsymbol{\lambda}} \left\{ \min_{\boldsymbol{u}} \{ f(\boldsymbol{u}) + \langle \boldsymbol{c} - \boldsymbol{C}\boldsymbol{u}, \boldsymbol{\lambda} \rangle \} - g(\boldsymbol{\lambda}) \right\} \Leftrightarrow \\
&\max_{\boldsymbol{\lambda}} \left\{ -\max_{\boldsymbol{u}} \{ \langle \boldsymbol{C}^\star \boldsymbol{\lambda}, \boldsymbol{u} \rangle - f(\boldsymbol{u}) \} - g(\boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \right\} \Leftrightarrow \\
&\max_{\boldsymbol{\lambda}} \{ -f^\star(\boldsymbol{C}^\star \boldsymbol{\lambda}) - g(\boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \} \quad ,
\end{aligned}$$

The solution can be derived by solving

$$\min_{\boldsymbol{\lambda}} \{ f^\star(\boldsymbol{C}^\star \boldsymbol{\lambda}) + g(\boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle \} \quad . \tag{8}$$

The function involved in the minimization problem (8) is

$$\begin{aligned}
f^\star(\boldsymbol{C}^\star \boldsymbol{\lambda}) = (1/2)\langle \boldsymbol{C}^\star \boldsymbol{\lambda}, \boldsymbol{H}^{-1} \boldsymbol{C}^\star \boldsymbol{\lambda} \rangle \\
- \langle \boldsymbol{C}^\star \boldsymbol{\lambda}, \boldsymbol{H}^{-1} \boldsymbol{h} \rangle + (1/2)\langle \boldsymbol{h}, \boldsymbol{H}^{-1} \boldsymbol{h} \rangle - r \quad .
\end{aligned} \tag{9}$$

Finally, the problem we are interested in solving can be cast in a more compact form as:

$$\text{minimize} \quad F(\boldsymbol{\lambda}) := h^\star(\boldsymbol{\lambda}) + \delta_-(\boldsymbol{\lambda}), \tag{10}$$

with variables $\boldsymbol{\lambda}$, $h^\star(\boldsymbol{\lambda}) = f^\star(\boldsymbol{C}^\star \boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{c} \rangle$ a differentiable function in $\Gamma_0(\mathcal{H}_{\boldsymbol{\lambda}})$, and $\delta_-(\boldsymbol{\lambda}) \in \Gamma_0(\mathcal{H}_{\boldsymbol{\lambda}})$. Now (6) can be applied to solve (10).

Before proceeding, we elaborate on the reasons why the original problem (1) had to be reformulated in order to solve it. There are two reformulations, namely, posing the problem as a function of the input sequences only, resulting in (3), and dualization of (3), resulting in (10). The reason for considering the condensed formulation is the need for *strong convexity* of the primal objective, which implies the Lipschitz continuity of $\nabla h^\star(\cdot)$ [7, Corollary 18.16]. By using the condensed form we avoid the restrictive assumption of $Q \succ 0$ required in [9].

The reason for considering the dual problem is simplicity in the evaluation of the proximal operator of the function $\delta_-(\cdot)$, which is a simple projection on the non-positive orthant (*i.e.*, componentwise clipping) as opposed to the primal case where one would have to project on a generic polytope of the form $\boldsymbol{C}\boldsymbol{u} \leq \boldsymbol{c}$.

Both steps are crucial for successfully applying the *Forward-Backward Splitting (FBS)* method, as presented and analyzed in [7, Theorem 25.8]

IV. SOLUTION USING AFBS

Problem (10) is a composite minimization problem (i.e., a minimization of a sum of a smooth and a non-smooth function) and will be solved using an *accelerated forward-backward splitting (AFBS)* method, which is an accelerated variant of the forward-backward splitting proposed in [7, Theorem 25.8]. The acceleration comes from a Nesterov-like momentum sequence from [13].

This modification allows acceleration of the FBS method in a very simple manner and adds practically zero computational complexity per iteration, in comparison to the original version. The idea can be traced back to Polyak and the so called heavy ball methods [14] for minimizing a smooth convex function $f(\cdot)$:

$$\begin{aligned}
\hat{\lambda}^k &= \lambda^k + \alpha^k(\lambda^k - \lambda^{k-1}) \\
\lambda^{k+1} &= \hat{\lambda}^k - \rho^k \nabla f(\lambda^k) \ ,
\end{aligned}$$

where $\alpha^k \in [0, 1)$ is an extrapolation factor and ρ^k a stepsize parameter. This seemingly small change of updating the new iterate as a linear combination of the two previous iterates greatly improves the performance of the original gradient scheme.

In his seminal paper [15], Nesterov modified the heavy ball method by simply evaluating the gradient at the extrapolated point $\hat{\lambda}^k$ instead of λ^k. In addition, he proposed a special formula for computing the relaxation sequence $(\alpha^k)_{k \in \mathbb{N}}$, resulting in an optimal convergence rate for the scheme. Subsequently,Nesterov's results were extended in several ways, lately handling the minimization of the sum of a smooth and a nonsmooth function [16].

The acceleration that optimal first order methods enjoy comes solely from the additional momentum of the minimizer sequence due to the relaxation sequence α^k. Among the many works that derived optimal relaxation sequences similar to that of Nesterov, a distinguished one is that of Beck and Teboulle in [17], who used Nesterov's framework to accelerate the FBS

method. The resulting algorithm, commonly known as *FISTA (Fast Iterative Shrinkage Thresholding Algorithm)* gives an optimal, $1/k^2$ convergence rate in terms of the function values. This result has valuable practical and theoretical implications, however, convergence of the iterates of the FISTA algorithm has not yet been proven. Building upon FISTA, Lorenz and Pock suggest in [8] a modified version of the method that achieves weak convergence of the iterates to the solution of the optimization problem in a Hilbert space setting.

The several 'optimal' relaxation sequences have been put under a common framework in the recent work [13]. The authors showed that any sequence $(t^k)_{k\in\mathbb{N}}$ of the form $t^k = \frac{k+a-1}{a}$, with $a \geq 2$ satisfies the inequality $(t^k)^2 - t^k \leq (t^{k-1})^2,\ k \geq 2$. Then the sequence defined as $\alpha^k = \frac{t^k-1}{t^{k+1}}$ allows for the optimal $O(1/k^2)$ convergence rate in terms of the function values, starting from $t_1 = 1$, and using a constant stepsize $\rho \in (0, 1/L_f]$, where L_f is the Lipschitz constant of $\nabla f(\cdot)$. In addition, *weak convergence of the iterates* is achievable for any choice $a > 2$. Any such scheme is optimal in the sense that, for every iterate k, there exists a problem which has a lower complexity bound of the same order. We denote the algorithm emanating from this scheme the Accelerated Forward-Backward Splitting (AFBS) algorithm, and write it down for problem (10) as follows:

Algorithm 1 AFBS for Problem (10)

0: Initialize $\boldsymbol{\lambda}^0 = \mathbf{0}$, $a > 2$, $\alpha^0 = 0$,
$L_{h^\star} \leftarrow$ a Lipschitz constant of $\nabla h^\star$.
repeat
1: $\alpha^k = \frac{k-1}{k+a},\ k \geq 1$
2: $\hat{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \alpha^k(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1})$
3: $\boldsymbol{\lambda}^{k+1} = \min\left\{\hat{\boldsymbol{\lambda}}^k - (1/L_{h^\star})\nabla h^\star(\hat{\boldsymbol{\lambda}}^k), 0\right\}$
until termination condition is satisfied.

The iterative scheme above is very simple: in this case the AFBS boils down to a variant of the fast projected gradient method as proposed by Nesterov [18]. In order to apply algorithm (1), we need to be able to

- evaluate the gradient of $h^\star(\cdot)$
- represent $\boldsymbol{\lambda}^k$ and $\hat{\boldsymbol{\lambda}}^k$ using a finite amount of memory.

The remaining steps of the algorithm are simple scalar or vector updates or componentwise clipping on the non-positive orthant (Step 3), both of which can be carried out inexpensively provided that $\boldsymbol{\lambda}^k$ can be represented using finite amount of memory.

In the rest of the text P_{LQ} and K_{LQ} will denote the positive-semidefinite solution to the discrete-time algebraic Riccati equation and the corresponding LQ optimal state feedback matrix associated with the matrices (A, B, Q, R).

We start by finding the gradient of $h^\star(\cdot)$. To this end, define the Lagrangian of the (infinite dimensional) problem (3), written in its equivalent sparse form and truncated at any $T \geq 0$, as

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \hat{\boldsymbol{\lambda}}|T) \ &= x_T^\top P_{\mathrm{LQ}} x_T \\ &+ \sum_{i=0}^{T-1} x_i^\top Q x_i + u_i^\top R u_i - w^i \begin{bmatrix} C_x x_i - c_x \\ C_u u_i - c_u \end{bmatrix}^\top \hat{\lambda}_i.\end{aligned} \tag{11}$$

Lemma 1: If $\hat{\boldsymbol{\lambda}}$ is such that $\hat{\lambda}_i = 0$ for all $i \geq T$, then

$$[\nabla h^\star(\hat{\boldsymbol{\lambda}})]_i = w^i \begin{bmatrix} C_x \hat{x}_i - c_x \\ C_u \hat{u}_i - c_u \end{bmatrix}, \tag{12}$$

where

$$\begin{aligned}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = \underset{\boldsymbol{u},\boldsymbol{x}}{\operatorname{argmin}}\{\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \hat{\boldsymbol{\lambda}}|T) \mid \\ x_{i+1} = Ax_i + Bu_i,\ x_0 = x_{\mathrm{init}}\}\end{aligned} \tag{13}$$

for $i \in \{0, \ldots, T\}$, and

$$\hat{x}_i = (A + BK_{\mathrm{LQ}})^{i-T}\hat{x}_T, \quad \hat{u}_i = K_{\mathrm{LQ}}\hat{x}_i$$

for $i > T$.

Proof: This is a standard result from duality (see, e.g., [19, Chapter 5]), noticing that the minimization in (13) is equivalent to the minimization of the Lagrangian of (3) under the assumption that $\hat{\lambda}_i = 0$ for all $i \geq T$. ∎

Lemma 1 gives us a way to compute the gradient of $h^\star$. Clearly, this gradient is an infinite sequence and therefore cannot be stored directly, but it is available to us explicitly for $i \in \{0, \ldots T\}$ and implicitly, through the dynamics of the system $\hat{x}_+ = (A + BK_{\mathrm{LQ}})\hat{x}$, for $i > T$.

Now we show that $\boldsymbol{\lambda}^k$ and $\hat{\boldsymbol{\lambda}}^k$ can be represented using a finite amount of memory.

Lemma 2: If $\boldsymbol{\lambda}^k$ and $\hat{\boldsymbol{\lambda}}^k$ are generated by Algorithm 1, then for each k there exists a $T^k < \infty$ such that $\lambda_i^k = 0$ and $\hat{\lambda}_i^k = 0$ for all $i \geq T^k$.

Proof: We have $\lambda_i^0 = 0$ and $\hat{\lambda}_i^0 = 0$ for all $i \geq 0$ and hence $T^0 = 0$. Assume now $k > 0$ and $\lambda_i^k = 0$ and $\hat{\lambda}_i^k = 0$ for all $i \geq T^k$. Then, evaluating $\nabla h^\star(\boldsymbol{\lambda}^k)$ using Lemma 1 with $T = T^k$, we see that the sequence $(\hat{x}_i, \hat{u}_i)$ is generated by the unconstrained LQ controller for $i > T^k$ and hence converges to the origin. Since the set $\mathcal{X}$ has the origin in the interior we conclude that there exists a time $\hat{T}^{k+1} < \infty$ such that $C_x\hat{x}_i \leq c_x$ and $C_u\hat{u}_i \leq c_u$ for all $i \geq \hat{T}^{k+1}$. We define $T^{k+1} = \max\{T^k, \hat{T}^{k+1}\}$. In view of (12), we conclude that $\hat{\boldsymbol{\lambda}}^k - \frac{1}{L_{h^\star}}\nabla h^\star(\hat{\boldsymbol{\lambda}}^k) \geq 0$ for all $i \geq T^{k+1}$. Therefore, in view of Step 3 of Algorithm 1, we have $\lambda_i^{k+1} = 0$ for all $i \geq T^{k+1}$. Finally, $\hat{\boldsymbol{\lambda}}^{k+1}$ is a linear combination of $\boldsymbol{\lambda}^k$ and $\boldsymbol{\lambda}^{k+1}$ and hence $\hat{\lambda}_i^{k+1} = 0$ for all $i \geq T^{k+1}$. ∎

To determine T^{k+1} computationally (given T^k and $\hat{\boldsymbol{x}}^{k+1}$ and $\hat{\boldsymbol{u}}^{k+1}$) we simply find the first time $T^{\mathcal{S}}$ that $\hat{x}_i^{k+1}$ enters a given subset $\mathcal{S}$, with $0 \in \mathbf{int}\,\mathcal{S}$, of the maximal positively invariant set of the system $\hat{x}_+ = (A + BK_{\mathrm{LQ}})\hat{x}$ subject to the constraint $\begin{bmatrix} C_uK \\ C_x \end{bmatrix}\hat{x} \leq \begin{bmatrix} c_u \\ c_x \end{bmatrix}$. The time T^{k+1} is then equal to the first time no less than T^k such that $C_x\hat{x}_{i+1}^{k+1} \leq c_x$ and $C_u\hat{u}_i^{k+1} \leq c_u$ simultaneously hold for all $i \in \{T^{k+1}, \ldots, T^{\mathcal{S}}\}$. More formally, we have the equality

$$\begin{aligned}T^{k+1} = \min\big\{T \geq T^k \mid \exists\, T^{\mathcal{S}} \text{ s.t. } C_x\hat{x}_{i+1}^{k+1} \leq c_x, \\ C_u\hat{u}_i^{k+1} \leq c_u\ \forall\, i \in \{T, \ldots, T^{\mathcal{S}}\}, \\ \text{and } \hat{x}_{T^{\mathcal{S}}}^{k+1} \in \mathcal{S}\big\}.\end{aligned} \tag{14}$$

Remark 2 (Computation of T^k): In practice, to determine T^{k+1} after solving (13), we iterate forward the system dynamics $\hat{x}_+ = (A + BK_{\mathrm{LQ}})\hat{x}$ starting from the initial condition $\hat{x}_{T^k+1}^{k+1}$ until $\hat{x}_i^{k+1} \in \mathcal{S}$.

Remark 3 (Set $\mathcal{S}$): The set $\mathcal{S}$ is determined offline and is *not* required to be invariant. A good candidate is the set $\{x \mid x^\top P_{\rm LQ} x \leq 1\}$ scaled such that it is included in $\left\{x \mid \begin{bmatrix} C_u K \\ C_x \end{bmatrix} x \leq \begin{bmatrix} c_u \\ c_x \end{bmatrix}\right\}$, or any subset of this set containing the origin in the interior.

Now we are ready to formulate an implementable version of the abstract algorithm 1:

Algorithm 2 AFBS for the CLQR problem

Require: $x_{\rm init}$, $Q \succeq 0, R \succ 0$, C_u of full column rank

a: Determine $P_{\rm LQ}$, $K_{\rm LQ}$ solving the unconstrained LQR problem associated with the matrices (A, B, Q, R).

b: Determine a set $\mathcal{S}$, with $0 \in \mathbf{int}\,\mathcal{S}$, included in any positively invariant set for the system $x_+ = (A + BK_{\rm LQ})x$ subject to the constraint $\begin{bmatrix} C_u K \\ C_x \end{bmatrix} x \leq \begin{bmatrix} c_u \\ c_x \end{bmatrix}$. See Remark 3.

c: *Initialize* $\boldsymbol{\lambda}^0 = \mathbf{0}$, $T^0 = 0$, $w = \min\left\{1, \frac{1}{\lambda^2_{\max}(A)}\right\}$, $a > 2$, $\alpha^0 = 0$ $L^0 > 0$ *or* $L_{h^\star} \leftarrow$ *a Lipschitz constant of* $\nabla h^\star$ *(optional).*

for $k = 0, \ldots$ **do**

1: $\alpha^k = \frac{k-1}{k+a}$, $k \geq 1$

2: $\hat{\lambda}_i^k = \lambda_i^k + \alpha^k(\lambda_i^k - \lambda_i^{k-1}), \quad i = 1, \ldots, T^k$

3: Set

$$(\hat{\boldsymbol{x}}^{k+1}, \hat{\boldsymbol{u}}^{k+1}) = \underset{\boldsymbol{u},\boldsymbol{x}}{\operatorname{argmin}}\{\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \hat{\boldsymbol{\lambda}}^k | T^k) \mid x_{i+1} = Ax_i + Bu_i,\, x_0 = x_{\rm init}\},\ i = 0, \ldots, T^k,$$

$\hat{x}_{i+1}^{k+1} = (A + BK_{\rm LQ})\hat{x}_i^{k+1}, \quad i > T^k$

4: Determine T^{k+1} (see Remark 2)

6: Choose stepsize ρ (see Remark 4)

7: Set $\lambda_i^{k+1} = \min\left(\hat{\lambda}_i^k - \rho^{k+1} w^i \begin{bmatrix} C_x \hat{x}_i^{k+1} - c_x \\ C_u \hat{u}_i^{k+1} - c_u \end{bmatrix}, 0\right)$.

8: If termination condition is satisfied, solve KKT system (see Remark 6)

end for

Remark 4 (Stepsize): In Step 6 of the algorithm, a stepsize is selected. One option is to fix a constant stepsize $\rho = 1/L_{h^\star}$, which needs a global Lipschitz constant of $\nabla h^\star(\cdot)$. This can be computed offline (see Section VI-A). Alternatively, one can use a *backtracking stepsize rule*, inspired from [17], which can be used in combination with the global estimate. The procedure is analyzed in Section VI.

Remark 5 (Role of the weight w): It is worth mentioning that working in the weighted Hilbert space l_w^2 is much more than a mathematical formalism and has serious practical implications. In the case of unstable systems, a nontrivial sequence of weights has to be chosen such that the growth of the largest unstable eigenvalue of the state matrix A is bounded by a faster decaying sequence so that the operator $\boldsymbol{C}$ remains bounded, an assumption necessary for applying the proposed method. At the same time the sequence of weights will act as a left preconditioner of $\boldsymbol{C}$, hence a left and right preconditioner on the Hessian operator of the quadratic form in (10). This scaling can seriously affect the numerical performance of the proposed algorithms, as we will see in subsequent sections. Note that for stable systems the sequence can be trivially set to 1, hence no scaling occurs. These claims are explained in more detail in Appendix B.

Remark 6 (Termination): Algorithm 2 terminates when a prespecified accuracy is reached in terms of the progress of the dual sequence. The extracted primal sequence is given by

$$(\boldsymbol{x}^k, \boldsymbol{u}^k) = \underset{\boldsymbol{u},\boldsymbol{x}}{\operatorname{argmin}}\{\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}^k | T^k) \mid x_{i+1} = Ax_i + Bu_i,\, x_0 = x_{\rm init}\} \tag{15}$$

for $i \in \{0, T^k\}$ and $x_i^k = (A + BK_{\rm LQ})^{i-T^k} x_{T^k}$ for $i > T^k$. In Theorem 1 below it is proven that $(\boldsymbol{x}^k, \boldsymbol{u}^k)$ tends to the optimal constrained LQ solution. At any finite iterate, however, the sequence $(\boldsymbol{x}^k, \boldsymbol{u}^k)$ may violate the constraints. In order to remedy this we solve upon termination an equality-constrained QP where we minimize the objective function subject to the active constraints at optimality. The active constraints can be (approximately) detected by looking at the nonzero values of the dual vector $\boldsymbol{\lambda}^k$ at termination. This step comes at a very small cost since it involves one solution of a KKT system of linear equations.

V. Convergence results

In the previous section we gave an implementable algorithmic scheme that computes the solution to the CLQR problem. Here we provide all the necessary proofs which allow us to assert that the solution generated by Algorithm 2 via (15) indeed converges to the true optimizer of the CLQR problem. In what follows $\boldsymbol{\lambda}^\infty$ denotes any optimal solution to the dual problem (10) (which exists under Assumption 1 but may not be unique) and $(\boldsymbol{u}^\infty, \boldsymbol{x}^\infty)$ the optimal solution to the primal problem (1). Our main result is:

Theorem 1 (Main Theorem): Suppose Assumption 1 holds and let $\boldsymbol{\lambda}^k$ be a sequence of iterates generated by Algorithm 1 and $(\boldsymbol{x}^k, \boldsymbol{u}^k)$ the associated primal sequence given by (15) and let L be a Lipschitz constant of $\nabla h^\star(\cdot)$. The following statements hold:

(i) The composite function $F(\boldsymbol{\lambda}) = h^\star(\boldsymbol{\lambda}) + \delta_-(\boldsymbol{\lambda})$ as defined in (10) converges as

$$F(\boldsymbol{\lambda}^k) - F(\boldsymbol{\lambda}^\infty) \leq \frac{a^2 L}{2(k+a-1)^2} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|^2_{\mathcal{H}_{\boldsymbol{\lambda}}} \ .$$

(ii) The sequence of the dual iterates $(\boldsymbol{\lambda}^k)_{k\in\mathbb{N}}$ converges weakly (see Definition 3 in Appendix A) to an optimizer, that is,

$$\boldsymbol{\lambda}^k \rightharpoonup \boldsymbol{\lambda}^\infty$$

for some $\boldsymbol{\lambda}^\infty \in \operatorname{argmin}(F)$.

(iii) The input sequence $(\boldsymbol{u}^k)_{k\in\mathbb{N}}$ converges strongly to the unique minimizer as

$$\|\boldsymbol{u}^k - \boldsymbol{u}^\infty\|_{\mathcal{H}_{\boldsymbol{u}}} \leq a\sqrt{\frac{L}{\mu}} \frac{\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|_{\mathcal{H}_{\boldsymbol{\lambda}}}}{(k+a-1)} \ ,$$

where $\mu > 0$ is the strong convexity modulus of $f(\boldsymbol{u})$.

(iv) The state sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ converges strongly to the unique minimizer as

$$\|\boldsymbol{x}^k - \boldsymbol{x}^\infty\|_{\mathcal{H}_{\boldsymbol{x}}} \leq a\sqrt{\frac{\|\boldsymbol{B}\|^2 L}{\mu}} \frac{\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|_{\mathcal{H}_{\boldsymbol{\lambda}}}}{(k+a-1)} \ .$$

(v) The sequence $(T^k)_{k\in\mathbb{N}}$ is bounded.

Proof:

(i) Convergence of $F(\boldsymbol{\lambda}^k)$ with a constant stepsize is proven in [13, Theorem 1]. Convergence at the same rate with an adaptive stepsize generated from the backtracking Algorithm 3 is proven in Lemma 10, Appendix C.

(ii) The proof is stated in [13, Theorem 3].

(iii) The idea is to upper bound the input sequence's convergence rate making use of the result in point (i). In order to do so we make use of strong duality. The proof is inspired from [20, Theorem 4.1] and is as follows:
Let $\boldsymbol{\lambda}^k \leq 0$ generated from Step 7 of Algorithm 2. Denote

$$\boldsymbol{u}^k = \underset{\boldsymbol{u}\in\mathcal{H}_{\boldsymbol{u}}}{\operatorname{argmin}} \left\{ f'(\boldsymbol{u}) := f(\boldsymbol{u}) + \langle \boldsymbol{\lambda}^k, \boldsymbol{c} - \boldsymbol{C}\boldsymbol{u}\rangle \right\} \ , \tag{16}$$

where $f(\boldsymbol{u}) = (1/2)\boldsymbol{u}^\star \boldsymbol{H}\boldsymbol{u} + \boldsymbol{h}^\star \boldsymbol{u} + r$ as defined in Section II. Then we have that the Lagrangian of (3) evaluated at $\boldsymbol{\lambda}^k$ is $\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}^k) = f'(\boldsymbol{u})$. The function $f(\boldsymbol{u})$ is strongly convex with modulus $\mu \geq \lambda_{\min}(R) > 0$, where $\lambda_{\min}(R)$ denotes the smallest eigenvalue of R. Strong convexity of $f'(\boldsymbol{u})$ with modulus μ follows directly. Using (16), it holds that

$$f'(\boldsymbol{u}) - f'(\boldsymbol{u}^k) \geq \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{u}^k\|^2_{\mathcal{H}_{\boldsymbol{u}}}, \quad \forall \boldsymbol{u} \in \mathcal{H}_{\boldsymbol{u}} \ ,$$

or, equivalently,

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}^k) - \mathcal{L}(\boldsymbol{u}^k, \boldsymbol{\lambda}^k) \geq \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{u}^k\|^2_{\mathcal{H}_{\boldsymbol{u}}}, \quad \forall \boldsymbol{u} \in \mathcal{H}_{\boldsymbol{u}} \ . \tag{17}$$

Substituting $\boldsymbol{u} = \boldsymbol{u}^\infty$ in (17) and by observing that $\max_{\boldsymbol{\lambda}\leq 0} \mathcal{L}(\boldsymbol{u}^\infty, \boldsymbol{\lambda}) \geq \mathcal{L}(\boldsymbol{u}^\infty, \boldsymbol{\lambda}^k)$, we have that

$$\mathcal{L}(\boldsymbol{u}^\infty, \boldsymbol{\lambda}^\infty) - \mathcal{L}(\boldsymbol{u}^k, \boldsymbol{\lambda}^k) \geq \frac{\mu}{2}\|\boldsymbol{u}^\infty - \boldsymbol{u}^k\|^2_{\mathcal{H}_{\boldsymbol{u}}}, \quad \forall \boldsymbol{u} \in \mathcal{H}_{\boldsymbol{u}} \ . \tag{18}$$

We have managed to derive an upper bound for the distance of the generated sequence of primal minimizers $(\boldsymbol{u}^k)_{k\in\mathbb{N}}$ from the optimal one. The last step is to show that the Lagrangian $\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda})$ is associated to the composite objective $F(\boldsymbol{\lambda})$. This can be easily shown as follows:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{u}^k, \boldsymbol{\lambda}^k) &= \min_{\boldsymbol{u}\in\mathcal{H}_{\boldsymbol{u}}} \left\{ f(\boldsymbol{u}) + \langle \boldsymbol{\lambda}^k, \boldsymbol{c} - \boldsymbol{C}\boldsymbol{u}\rangle \right\} \\
&= -\max_{\boldsymbol{u}\in\mathcal{H}_{\boldsymbol{u}}} \left\{ -f(\boldsymbol{u}) + \langle \boldsymbol{\lambda}^k, \boldsymbol{C}\boldsymbol{u}\rangle \right\} + \langle \boldsymbol{\lambda}^k, \boldsymbol{c}\rangle \\
&= -f^\star(\boldsymbol{C}^\star \boldsymbol{\lambda}^k) + \langle \boldsymbol{\lambda}^k, \boldsymbol{c}\rangle \\
&= -F(\boldsymbol{\lambda}^k), \quad \text{by (10)} \ .
\end{aligned}$$

From strong duality and the fact that $-F(\boldsymbol{\lambda}^k)$ converges to the optimal dual value (point (i)), we have that the optimal value of the dual function $-F(\boldsymbol{\lambda})$ coincides with that of the Lagrangian evaluated at the saddle point $(\boldsymbol{u}^\infty, \boldsymbol{\lambda}^\infty)$, *i.e.*, $\mathcal{L}(\boldsymbol{u}^\infty, \boldsymbol{\lambda}^\infty) = \max_{\boldsymbol{\lambda}\in\mathcal{H}_{\boldsymbol{\lambda}}} \{-F(\boldsymbol{\lambda})\} = -F(\boldsymbol{\lambda}^\infty)$ (see [19, Section 5.5.5]). Making use of point (i), inequality (18) becomes

$$\frac{\mu}{2}\|\boldsymbol{u}^k - \boldsymbol{u}^\infty\|^2_{\mathcal{H}_{\boldsymbol{u}}} \leq F(\boldsymbol{\lambda}^k) - F(\boldsymbol{\lambda}^\infty) \leq \frac{a^2 L \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|^2_{\mathcal{H}_{\boldsymbol{\lambda}}}}{2(k+a-1)^2}, \tag{19}$$

which concludes the proof.

(iv) The state sequence is generated by

$$\boldsymbol{x}^k = \boldsymbol{A}x_{\text{init}} + \boldsymbol{B}\boldsymbol{u}^k \ . \tag{20}$$

Strong convergence of the input sequence $(\boldsymbol{u}^k)_{k\in\mathbb{N}}$, along with the facts that $\boldsymbol{B}\colon \mathcal{H}_{\boldsymbol{x}} \to \mathcal{H}_{\boldsymbol{u}}$ is bounded (follows directly from Lemma 6 in Appendix C) and the uniqueness of $\boldsymbol{u}^\infty$ prove strong convergence of the state sequence with rate $1/k$, *i.e.*,

$$\begin{aligned}
\|\boldsymbol{x}^k - \boldsymbol{x}^\infty\|_{\mathcal{H}_{\boldsymbol{x}}} &= \|\boldsymbol{B}(\boldsymbol{u}^k - \boldsymbol{u}^\infty)\|_{\mathcal{H}_{\boldsymbol{x}}} \\
&\leq \|\boldsymbol{B}\|\|\boldsymbol{u}^k - \boldsymbol{u}^\infty\|_{\mathcal{H}_{\boldsymbol{x}}} \\
&\leq a\|\boldsymbol{B}\|\sqrt{\frac{L}{\mu}}\frac{\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|_{\mathcal{H}_{\boldsymbol{\lambda}}}}{(k+a-1)} \ ,
\end{aligned}$$

with the last inequality following directly from point (iii).

(v) Although the proof for the unaccelerated case was presented in [9], the appearance of the over-relaxed sequences $(\hat{\boldsymbol{x}}^k)_{k\in\mathbb{N}}$, $(\hat{\boldsymbol{u}}^k)_{k\in\mathbb{N}}$ renders the derivation of the result more challenging.
The key piece for the proof is the weak convergence of $\hat{\boldsymbol{x}}^k$ to $\boldsymbol{x}^\infty$. This claim is subsequently proven as a sequence of intermediate results. We show that:

a) The relaxed dual sequence $(\hat{\boldsymbol{\lambda}}^k)_{k\in\mathbb{N}}$ converges weakly to a dual minimizer $\boldsymbol{\lambda}^\infty$.

b) Provided that the operator $\boldsymbol{C}$ is bounded, the sequence $(\boldsymbol{H}\hat{\boldsymbol{u}}^k)_{k\in\mathbb{N}}$ converges weakly to $(\boldsymbol{H}\boldsymbol{u}^k)_{k\in\mathbb{N}}$, and $(\hat{\boldsymbol{u}}^k)_{k\in\mathbb{N}}$ converges weakly to $(\boldsymbol{u}^k)_{k\in\mathbb{N}}$. From strong duality, $(\boldsymbol{u}^k)_{k\in\mathbb{N}}$ converges to the primal optimizer $\boldsymbol{u}^\infty$.

c) Weak convergence of the accelerated state sequence $(\hat{\boldsymbol{x}}^k)_{k\in\mathbb{N}}$ to $\boldsymbol{x}^\infty$ follows directly.

Weak convergence of the relaxed sequence $(\hat{\boldsymbol{\lambda}}^k)_{k\in\mathbb{N}}$ follows from Corollary 2 of [13], which states that the error sequence $(\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2)_{k\in\mathbb{N}}$ converges to zero with rate $1/k^2$. We state the result below.

Lemma 3: The relaxed sequence $(\hat{\boldsymbol{\lambda}}^k)_{k\in\mathbb{N}}$ converges weakly to $\boldsymbol{\lambda}^\infty$.

Proof: Since $\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \to 0$ and α_k is bounded we also have $\boldsymbol{\nu}^k = \sqrt{\alpha^k}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) \to 0$. Since strong convergence implies weak convergence we have that $\langle \boldsymbol{\nu}^k, \boldsymbol{y}\rangle \xrightarrow{k\to\infty} 0$, $\forall \boldsymbol{y} \in \mathcal{H}_{\boldsymbol{\lambda}}$. The relaxed sequence of duals $\hat{\boldsymbol{\lambda}}$ can be written as $\hat{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \sqrt{\alpha^k}\boldsymbol{\nu}^k$. Consequently, since $\boldsymbol{\lambda}^k \rightharpoonup \boldsymbol{\lambda}^\infty$, we have that

$$\begin{aligned}
\langle \hat{\boldsymbol{\lambda}}^k, \boldsymbol{y}\rangle &= \langle \boldsymbol{\lambda}^k + \sqrt{\alpha^k}\boldsymbol{\nu}^k, \boldsymbol{y}\rangle \\
&= \langle \boldsymbol{\lambda}^k, \boldsymbol{y}\rangle + \sqrt{\alpha^k}\langle \boldsymbol{\nu}^k, \boldsymbol{y}\rangle \to \boldsymbol{\lambda}^\infty
\end{aligned}$$

for all $y \in \mathcal{H}_{\boldsymbol{\lambda}}$ and hence $\hat{\boldsymbol{\lambda}}^k \rightharpoonup \boldsymbol{\lambda}^\infty$. ■

Lemma 4: The sequence $(\hat{\boldsymbol{u}}^k)_{k\in\mathbb{N}}$ converges weakly to $\boldsymbol{u}^\infty$.

Proof: Writing down the relation between $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\lambda}}$ from Lemma 1 in terms of the operators, we get $\hat{\boldsymbol{u}}^k = \boldsymbol{H}^{-1}(\boldsymbol{C}^\star \boldsymbol{W}\hat{\boldsymbol{\lambda}}^k - \boldsymbol{h})$. Similarly we have $\boldsymbol{u}^k = \boldsymbol{H}^{-1}(\boldsymbol{C}^\star \boldsymbol{W}\boldsymbol{\lambda}^k - \boldsymbol{h})$. Now, from Theorem 1 we have $\boldsymbol{u}^k \to \boldsymbol{u}^\infty$ and from Lemma 3 we have $\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^k \rightharpoonup 0$.

Therefore, since $\boldsymbol{C}^\star$, $\boldsymbol{W}$ and $\boldsymbol{H}^{-1}$ are bounded operators (see Lemmas 6 and 7 in Appendix C) and since weak convergence is preserved under bounded linear mappings, we conclude that $\hat{\boldsymbol{u}}^k \rightharpoonup \boldsymbol{u}^\infty$. ∎

Lemma 5: The sequence $(\hat{\boldsymbol{x}}^k)_{k\in\mathbb{N}}$ converges weakly to $\boldsymbol{x}^\infty$.

Proof: Exactly as we did at point (iv) of Theorem 1, the accelerated state sequence can be written as

$$\hat{\boldsymbol{x}}^k = \boldsymbol{A}x_{\text{init}} + \boldsymbol{B}\hat{\boldsymbol{u}}^k \ .$$

Weak convergence $\hat{\boldsymbol{u}}^k \rightharpoonup \boldsymbol{u}^\infty$ and boundedness of $\boldsymbol{B}$ prove weak convergence of the accelerated state sequence to $\boldsymbol{x}^\infty$. ∎

We have, hence, proven the weak convergence of the accelerated state sequence to the optimal one. For $(T^k)_{k\in\mathbb{N}}$ to be bounded, it is sufficient to show that

$$\limsup_{k\to\infty} T^k < \infty. \tag{21}$$

To prove (21), define the sequence of the first hitting times of the interior of $\mathcal{S}$ as

$$\tau^k := \inf\{i \geq T^k \mid x_i^k \in \mathbf{int}\,\mathcal{S}\}, \ \ k \in \mathbb{N}\cup\{+\infty\},$$

where $\tau^\infty < \infty$ is the hitting time of the optimal state sequence $\boldsymbol{x}^\infty$. Clearly, $\tau^k \geq T^k$ and $\tau^k < \infty$ since the origin is in the interior of $\mathcal{S}$ and for each $k \in \mathbb{N}$ the sequence $(\hat{x}_i^k)_{i\in\mathbb{N}}$ generated by Algorithm 2 converges to the origin as $i \to \infty$. We shall prove that $\limsup_{k\to\infty} \tau^k \leq \tau^\infty < \infty$, which implies (21). For the purpose of contradiction assume that there exists a subsequence τ^{k_j}, $j \in \mathbb{N}$, with $\lim_{j\to\infty} \tau^{k_j} \geq \tau^\infty + 1$. Since the sequence of hitting times τ^k is integer valued, this implies that there exists a $j^\star \in \mathbb{N}$ such that $\tau^{k_j} \geq \tau^\infty + 1$ for all $j \geq j^\star$. We now use this to contradict the weak convergence of $\hat{\boldsymbol{x}}^k$ to $\boldsymbol{x}^\infty$. To this end, observe that $x_{\tau^\infty}^\infty \in \mathbf{int}\,\mathcal{S}$ whereas $\hat{x}_{\tau^\infty}^{k_j} \notin \mathbf{int}\,\mathcal{S}$ for all $j \geq j^\star$. By the definition of the interior there exists an $\epsilon > 0$ such that $z \in \mathbf{int}\,\mathcal{S}$ for all z with $\|z - x_{\tau^\infty}^\infty\|_2 < \epsilon$. Therefore $\|\hat{x}_{\tau^\infty}^{k_j} - x_{\tau^\infty}^\infty\|_2 \geq \epsilon$ for all $j \geq j^\star$, and consequently

$$\begin{aligned}\langle \hat{\boldsymbol{x}}^{k_j} - \boldsymbol{x}^\infty, \boldsymbol{z}\rangle &= \sum_{i=0}^{\infty} w^i (\hat{x}_i^{k_j} - x_i^\infty)^\top z_i \\ &\geq w^{\tau^\infty}(\hat{x}_{\tau^\infty}^{k_j} - x_{\tau^\infty}^\infty)^\top z_{\tau^\infty} \\ &\geq w^{\tau^\infty}\epsilon^2 > 0 \ ,\end{aligned}$$

for a sequence $\boldsymbol{z}$ with $z_{\tau^\infty} = \hat{x}_{\tau^\infty}^{k_j} - x_{\tau^\infty}^\infty$ and for all $j \geq j^\star$, contradicting the weak convergence of $\hat{\boldsymbol{x}}^k$ to $\boldsymbol{x}^\infty$ asserted by Lemma 5.

∎

VI. Computational aspects and warm-starting

Having presented the algorithm and its convergence results, we now focus on the computational aspects that render the algorithm a practical option to alternatives such as MPC. We start with explaining how no prior knowledge of a fixed stepsize is needed, give some references concerning the solution of the linear system, which is the most expensive operation of the method, and we conclude the section by suggesting a warm-starting scheme.

A. Stepsize selection

The stepsize used in Algorithm 2 is computed as the reciprocal of the Lipschitz constant of $h^\star$. For the problem discussed here this can be explicitly computed. We have that

$$\begin{aligned}\|\nabla h^\star(\boldsymbol{\lambda}_1) - \nabla h^\star(\boldsymbol{\lambda}_2)\| &= \|\boldsymbol{C}\boldsymbol{H}^{-1}\boldsymbol{C}^\star\boldsymbol{W}(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)\| \\ &\leq \|\boldsymbol{C}\boldsymbol{H}^{-1}\boldsymbol{C}^\star\boldsymbol{W}\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\| \\ &= \|\boldsymbol{H}^{-1}\|\|\boldsymbol{C}\|^2\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\| \ ,\end{aligned}$$

where $\boldsymbol{W}$ is the diagonal operator constituted of the decaying weighting sequence, *i.e.*, $\boldsymbol{W} = \mathbf{diag}(I, wI, w^2I, \ldots)$. The last equality follows from the fact that $\boldsymbol{W}$ contains a non-increasing sequence with the largest element being one. Hence $L_{h^\star} = \|\boldsymbol{H}^{-1}\|\|\boldsymbol{C}\|^2$, which requires computation of the operator norm $\|\boldsymbol{H}^{-1}\|$ and $\boldsymbol{C}$. The proofs for boundedness of the two operators, as well as the computations of their bounds are derived in Appendix B, Lemmas 6 and 7.

Although valid, this offline computation of the stepsize tends to be conservative in many cases, due to the conservativeness of the computed upper bounds. An elegant and practical method to achieve faster convergence is to employ an algorithm that locally estimates the Lipschitz constant online, at every iteration of Algorithm 2. In order to do so, we use the *backtracking stepsize rule* suggested in [17]. The idea is simple: after each iteration of the algorithm, we make a quadratic approximation model of the function around the successor point, making use of the knowledge of the exact point and its gradient value. A quadratic term with varying curvature is added on top of the linear (first order Taylor) approximation, and the curvature is adapted recursively until our quadratic approximant upper bounds the original function, centered around the given point. Thus, the quadratic model's curvature is an estimate of the Lipschitz constant of the gradient of the original function. It is proven in [17] that the locally evaluated Lipschitz constant L^k is related to the global one by $\beta L_{h^\star} \leq L^k \leq \gamma L_{h^\star}$, where $\beta = \frac{L^0}{L_{h^\star}}$ and $\gamma > 1$, $L^0 > 0$ being an initial estimate. Consequently the rule allows for smaller L's and hence larger stepsizes, *i.e.*, faster convergence in practice.

Remark 7 (Backtracking): Although points (i), (iii) and (iv) of Theorem 1 can be easily shown to hold for a stepsize generated from the backtracking procedure described above, the same does not hold for point (ii), and, consequently, for point (v). More specifically, weak convergence of the dual sequence is based on the assumption that no stepsize larger than $2/L_{h^\star}$ is allowed at any iteration [13, Theorem 3].

B. Complexity

The most expensive operation of Algorithm 2 is the linear system solve in Step 3. There is a variety of ways to perform this step, *i.e.*, solving the KKT system or perform the Riccati recursion when both states and inputs are considered, invert the dense Hessian when only the inputs are considered. In the first case, a sparse (permuted) $\mathrm{LDL}^\top$ factorization can be performed with cost $T(n+m)^3$ flops, followed by a forward-backward solve at $T(n+m)^2$ flops. This approach has been followed in [21]. A discussion on the KKT system solve and the Riccati recursion approach is contained in [22], where

the corresponding complexities are analyzed and compared in detail.

In the case of the condensed formulation, the linear system solve can be efficiently performed by first applying a Cholesky factorization on H (being a finite truncation of the $\boldsymbol{H}$ operator in (3)), followed by a forward-backward substitution, (see [19, Appendix C]). Although the condensed form of the optimal control problem that is used in the derivations is, generally, not advised for long horizons, recent advancements can render this approach very efficient [23], [24]. More specifically, the two proposed algorithms that perform factorization and solve of the condensed system in [24] come with a reduced complexity of $O(Tn^3)$ and $O(T^2n^2)$, respectively.

Whether considering the sparse or the dense formulation, note that the factorization steps would have to be performed several times until the 'correct' horizon T^∞ has been identified, since the size of the corresponding matrices (KKT or Hessian H) increase as the algorithm progresses. This typically happens within the first few tens of iterations. A valid alternative to this is the Riccati recursion, which completely eliminates the need for factorization, at the expense of forward (or backward) simulating the trajectories for the gain, the inputs and the states.

The backtracking scheme contributes to the complexity of Algorithm 2 by requiring a number of function evaluations per iteration, both for the quadratic model and for the original smooth function $h^\star(\cdot)$ (see [17] for more details). The rest of the steps are simple vector updates of negligible cost.

C. Warm-starting

In the nominal case, *i.e.*, when no noise and no model uncertainty are present, the open loop infinite-horizon control sequence generated from Algorithm 2 coincides with the control sequence generated by the optimal closed-loop feedback controller. Consequently, there is no need to re-optimize in a receding horizon fashion. Solving the CLQR problem just once is sufficient.

In the more realistic scenario where the predicted initial state differs from the measured one, the algorithm has to be re-applied. Provided that the prediction is not very different from the actual state, a good strategy is to initialize the decision variables (states and inputs) of the new problem with the values predicted from the previous one. This is commonly known as warm-starting. In our case, warm-starting has to be performed in the dual variables.

Note that once Algorithm 2 has run once, a hitting time T^∞ has been generated, along with an optimal dual sequence λ^∞ of corresponding length. It is expected that when computing the control law the hitting time should decrease by one at each solve, provided warm-starting from the optimal dual sequence that was generated once in the beginning. Hence, we suggest a heuristic scheme where the 'constrained' (nonzero) part of the preceding shifted dual sequence is used to initialize each subsequent CLQR problem. The computation time thus reduces significantly, with the horizon practically shrinking to zero once the initial state is identified to be inside the maximal positively invariant set of the LQ controller.

Application of the scheme is presented in Section VII. It is observed that it behaves particularly well for small perturbations of the initial state.

VII. Examples

For illustrative purposes, we run the algorithm on two systems, a small unstable system with two states and one input and a linearized model of a quadcopter with 12 states and 4 inputs. We use the small example as a benchmark for graphical illustrations, while the larger one exhibits the computational efficiency of the proposed scheme. The comparison is performed against the same implementation of the AFBS algorithm for finite horizon lengths. It is of course understood that there exist several methods capable of solving a finite horizon MPC problems (see interior point, active set, etc.), among which, optimal first order methods have gained considerable attention over the last few years, rendering them a competitive alternative [25], [26]. Consequently, comparing against an optimal first order method provides a valid basis for evaluating the potential of our scheme. In the two examples the termination criterion is simply set as $\|\boldsymbol{\lambda}^k - \hat{\boldsymbol{\lambda}}^{k+1}\| \leq 10^{-4}$.

A. Toy system

Consider the following system defined as

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix},$$

$$x_{i+1} = Ax_i + Bu_i,$$

with constraints

$$\|x\|_\infty \leq 10, \quad \|u\|_\infty \leq 1$$

and $Q = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$, $R = 2I$. Note that the system is unstable, hence a nonzero sequence of weights has to be chosen in order to ensure boundedness of the $\boldsymbol{C}$ operator. We choose $w = 1/1.1^2$ and the value of a in Step 1 of Algorithm 2 is set to 4.

The system is simulated for 750 different initial conditions x_0. In Figure 1 the distribution of $T^\infty = \max_k\{T^k\}$ is depicted. We see that T^∞ goes up to 30, while the mean value is 9.

We compare our method to the MPC approach both from the control and the algorithmic performance perspective. Regarding the former, we perform a comparison of the feasible sets for finite horizon implementations versus the maximal control invariant set in the case of CLQR, as well as the optimal value of the cost function. Regarding the latter, we perform comparisons in terms of the average number of iterations needed for convergence for several horizon length values in MPC versus the CLQR case. We also evaluate the conservativeness of our approach by computing the actual 'optimal' horizon length for each initial condition we simulate. For both CLQR and MPC we make use of the same AFBS algorithm with backtracking employed and termination tolerance set to 10^{-4} as stated before.

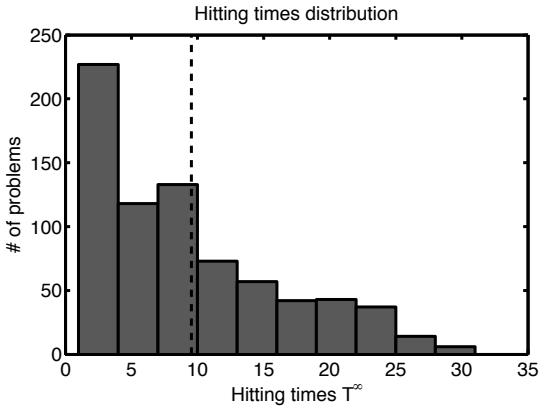


Fig. 1. Histogram of $T^\infty = \max_k\{T^k\}$ for 750 initial conditions of the 2 state system sampled from a normal distribution around $(-3, 0.3)$ with covariance matrix $\mathbf{diag}(4, 0.4)$.

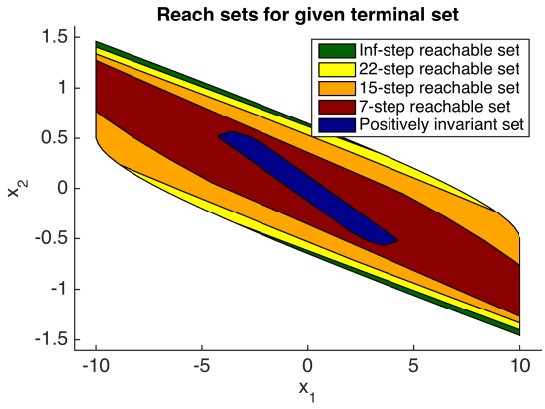


Fig. 2. Reachable sets for several horizon lengths and the LQ terminal set. The computation was done using the MPT3 toolbox [27]. It is apparent that a short horizon length reduces significantly the feasible region of the problem.

We perform the following simulations: we sample 243 initial conditions $x_{i,0},\ i = 1, \dots, 243$ from the maximal control invariant set (31-step) of the aforementioned constrained system (see Figure 2). For each point we compute:

1) The minimum horizon length $T_{\min}$, such that $x^*_{T_{\min}}(x_{i,0})$ resides in the maximal positively invariant set of the autonomous system $x^+ = (A + BK_{\mathrm{LQ}})x$, used as a terminal set.
2) The hitting time $T = T^\infty$ generated by our proposed scheme.

The following scenarios are generated: Firstly, an MPC problem with terminal set and horizon length $T_{\min}$ is solved, as described above. Subsequently we remove the terminal constraint and solve the MPC problem again for the same horizon length $T_{\min}$. We repeat the procedure described above (MPC with and without terminal set) for horizons $T = 2T_{\min}$ and $T = T^\star$, where $T^\star$ is specified by identifying the first horizon length for which the feasible solution of the MPC problem had the terminal constraint inactive. Finally, a comparison with the proposed CLQR approach is performed, with horizon $T = T^\infty$. In this way, six MPC problems with finite horizon as well as the CLQR problem are solved, for each of the 243 initial conditions. The total number of iterations is averaged by the number of corresponding initial conditions with the same minimum horizon length $T_{\min}$. The results are presented in Figure 3.

The first observation from the plot is that inclusion of the terminal constraint generally increases the number of required iterations since more constraints become active at optimality. This is especially the case when the horizon is relatively short and the terminal constraint satisfaction is imposed as we see in the blue curves. As the horizon increases (red and magenta color), the terminal constraint might be inactive and there is no significant difference in the number of iterations in the two cases.

A trend of increase in the number of iterations in all methods as $T^\star$ increases can be observed. This is expected since increase of $T^\star$ amounts to sampling of the initial state from regions of the feasible set that are further away from the origin. A consequence of the latter is that more constraints become active at optimality.

Short horizons in combination with a terminal set increase significantly the iteration count, as well as unnecessarily long horizons (*e.g.*, $2T_{\min}$). In that sense, one can observe that MPC with $T = T^\star$ behaves better than the other lengths. The curve corresponding to CLQR is denoted with black lines with asterisks. The method is comparable to the MPC approaches where $T = T_{\min}$ and $T = T^\star$ with terminal set, performing relatively better in small to medium horizon lengths and worse for larger horizon lengths. The reason for the latter is that the weighting operator $\boldsymbol{W}$ becomes quickly ill-conditioned as T grows, since its diagonal elements decay exponentially. This leads to ill-conditioning of the finite-dimensional truncation of the dual problem (10) since $\boldsymbol{W}$ preconditions (through the weighted l_2 inner product (4)) the objective of (10). As is commonly known first-order methods struggle with ill-conditioning. Efficient preconditioning of the operators should improve this and is a topic of further investigation. As a matter of fact, when the weights are set to one, the CLQR approach (black line with circles) clearly outperforms all MPC approaches with terminal set, as well as most of those instances without terminal set. Further simulations suggest that the proposed method performs very well, were we to drop the weights or when dealing with stable systems, when no scaling has to be performed.

Another interesting point is that the average hitting times T^∞ generated from Algorithm 2 are almost identical to the averaged optimal ones $T^\star$, with a very slight increase. This fact is depicted in Figure 4, where the ratio is always very close to one. On the contrary, the minimum required horizon length $T_{\min}$ is observed to be up to 45% smaller than the optimal length $T^\star$ in some cases, leading to a significant increase in the objective's cost when compared to the infinite horizon approach.

B. Quadcopter system

The next system we consider is a quadcopter linearized in a hovering equilibrium. The system has 12 states which correspond to position, angle and the corresponding velocities.

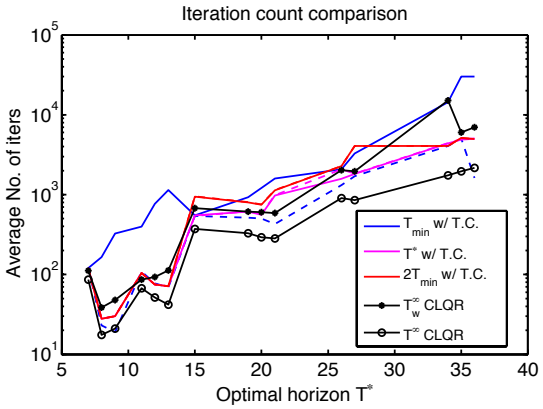


Fig. 3. Comparison of MPC with finite horizon length, for several horizons, with CLQR. The x-axis corresponds to the optimal horizon length $T^\star$ per initial condition. As $T^\star$ increases, the corresponding states are sampled further from the origin. MPC with terminal set is depicted with the solid lines, while without terminal set with dashed lines. CLQR is performed both with the weight sequence, denoted as T_w^∞ CLQR, and without the weights, denoted as T^∞ CLQR.

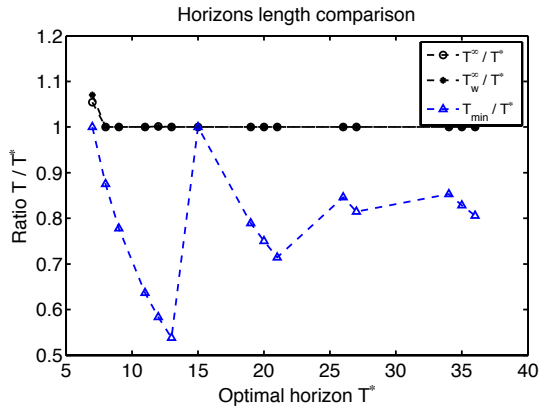


Fig. 4. Evolution of the ratios $\frac{T_{\min}}{T^\star}$, $\frac{T_w^\infty}{T^\star}$ and $\frac{T^\infty}{T^\star}$ for the sampled initial conditions.

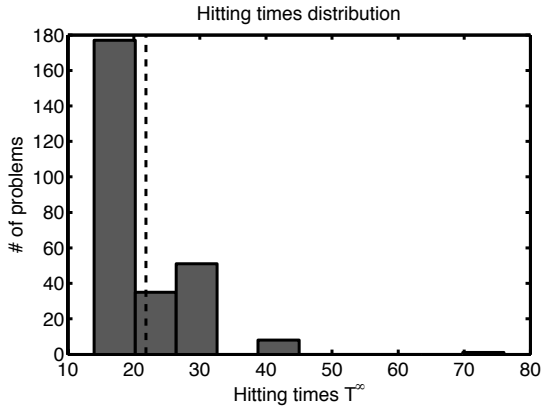


Fig. 5. Histogram of $T^\infty = \max_k\{T^k\}$ for 272 initial conditions sampled with a Hit-And-Run algorithm.

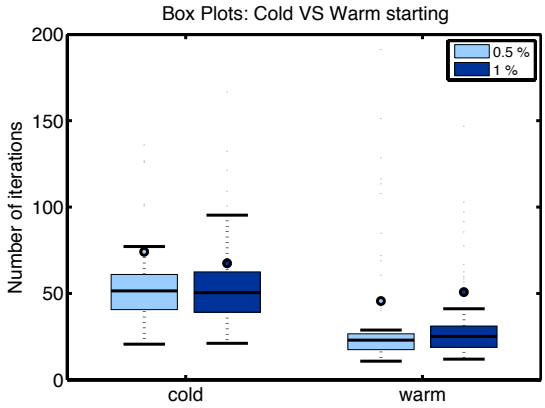


Fig. 6. Box plot for the average number of iterations of the warm-starting policy, given 0.5% and 1% uniform perturbations of the initial state. The horizontal line inside the box corresponds to the median, the edges of the box are the 25^{th} and 75^{th} percentiles while the horizontal lines outside the boxes correspond to the most extreme data points not considered outliers. Finally, the colored dots correspond to mean values. Warm-starting improves the performance of the suggested method in both cases, in all the depicted statistical measures.

There are four inputs corresponding to the four propellers. There are box constraints on all states and inputs, mainly ensuring the validity of the linearized model. The system is marginally stable; thus the weight w is set to one.

We simulate the algorithm starting from initial conditions randomly selected as follows: starting from a random feasible initial condition, we generate random directions on a unit ball centered around it and sample points along each of them. The points are generated from a normal distribution with standard deviation 0.15. Finally, we keep the initial conditions that result in feasible closed loop problems. The result of this step is 272 feasible initial conditions for the CLQR Algorithm 2. A histogram of $T^\infty = \max_k\{T^k\}$ is presented in Figure 5.

We conclude the section by applying the warm-starting heuristic scheme suggested in Section VI. We consider two scenarios; in the first one we uniformly perturb the initial state by 0.5% of its nominal value and run Algorithm 2 in closed loop for 78 different initial conditions. In the second scenario we perturb the state by 1%, for 68 different initial conditions. We solve 15 consecutive problems per initial condition and subsequently compare the average number of iterations as well as the generated hitting times per problem solve with and without warm-starting the dual variables. The results are summarized in Figure 6. It is evident that warm-starting consistently reduces the number of iterations in both cases.

Finally, the active reduction in the horizon length thanks to warm-starting is illustrated in Figure 7 for four different initial states in the case of the 1% perturbation. There is an (almost monotonic) decreasing trend, with the horizon finally shrinking to zero when the initial state resides in the positively invariant set.

C. Timings

We conclude this section with a numerical evaluation. For this purpose, we compare an implementation of the proposed algorithm with several modern convex optimization solvers, employed to solve the corresponding finite horizon MPC problems. In addition, we compare our proposed algorithm with

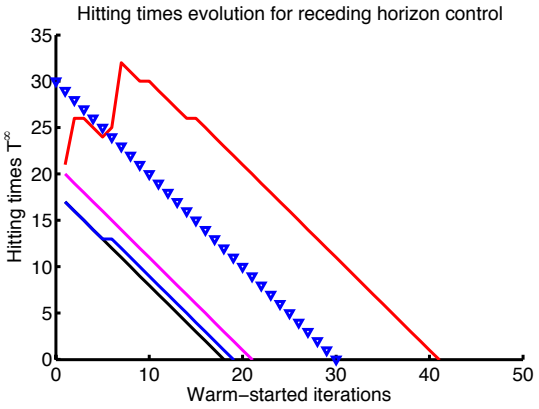


Fig. 7. Hitting times evolution for 50 subsequent solves, warm-started at the previous (shifted) dual optimizer. Four different initial conditions are depicted with the solid lines. The linear rate is depicted (for comparison) with the blue triangles. Note that the horizon decreases almost linearly, resulting from the relatively small perturbations of the initial state. The case depicted in red corresponds to an outlier where the perturbed initial condition rendered the previously computed horizon an infeasible option, leading to an increase of its length.

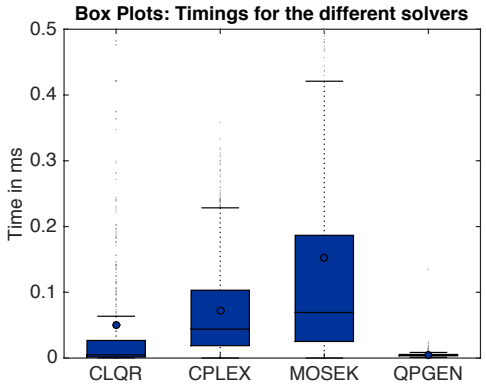


Fig. 8. Box plot for the solve times of the CLQR solver versus CPLEX, MOSEK and QPgen for the toy system, sampled from 750 initial conditions. CLQR generally requires less time than solving a series of finite horizon optimal control problems, leading to the same optimal solution. Only the solver QPgen clearly outperforms our proposed approach, which makes use of the same, in principle, splitting method, enhanced by several add-on's that enable speedup.

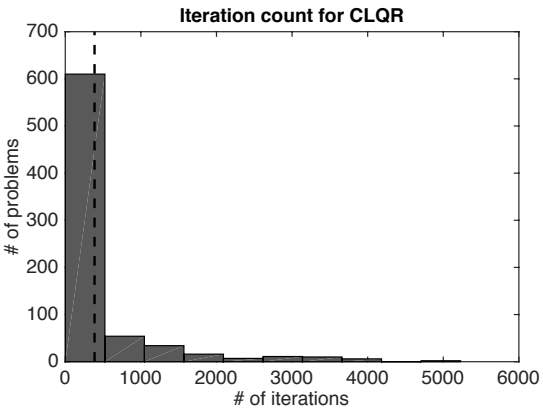


Fig. 9. Histogram for the number of iterations required over 750 problems. Note that more than 80% of the problems needed less than 400 iterations in order to converge, while the maximum number observed is around 5000 iterations.

its finite horizon implementation, solving the same optimal control problems in a receding horizon mode. Given that MPC is an approximation to the solution of the infinite horizon problem, a fair comparison between the two approaches cannot be easily performed. Hence, we consider a 'criterion of fairness' *the time required by Algorithm 2 versus a rolling horizon approach so that the infinite-horizon optimal solution to a given regulation problem is achieved*.

The difficulty with the aforementioned approach is to construct a rolling horizon scheme that can recover the infinite-horizon optimal solution. This can be achieved only if a sufficiently long horizon is used, so that the state sequence converges leisurely to the origin. Such a horizon is the output of our scheme. Thus, we perform the following simulation with the toy example presented above: We solve a series of optimal control problems for each of the 750 sampled initial conditions, the hitting times of which were depicted in Figure 1, using the resulting T^∞ from the CLQR approach, without imposing any terminal set. The problems are solved in a receding horizon fashion until the initial state resides inside the maximal positively invariant set of the LQ controller, depicted in Figure 2 in dark blue. Subsequently, we measure the time required to compute the closed loop solution with the MPC algorithms and we compare against the time required by Algorithm 2.

The algorithm is implemented in C++. The algorithmic values a and L^0 are initialized to $a = 5$ and $L^0 = 0.01$. The linear algebra library 'Armadillo' [28] is used for the operations, which is a wrapper for BLAS and LAPACK. The implementation is tested against the (commercial) solvers CPLEX (IBM) and MOSEK, as well as the open solver QPgen [29], [30].

The results regarding the first experiment are depicted in Figure 8. Since no noise is injected in the state, the CLQR controller is computed only once, in contrast to the MPC controller where multiple problems have to be solved in order to acquire the closed loop solution. As a consequence, the total time needed to solve the problem with CLQR is smaller than what most of the solvers in use can achieve. The relative accuracy of the generated solutions is of the order 10^{-5}, while the times are given in ms. In addition, Figures 9 and 10 elaborate on the iteration count, as well as the number of factorizations that had to be performed while running CLQR, as another computational complexity measure for the algorithm.

Subsequently, the closed loop MPC times are averaged over the number of optimal control problems solved, and are compared to the timings of Algorithm 2 in Figure 11. It is clear that in this case the proposed approach is not as competitive as MPC. However, it should be mentioned that the cases depicted in Figures 8 and 11 are two extremes. Figures 8 assumes that no noise is present, while Figure 11 implicitly assumes that Algorithm 2 is cold-started at every subsequent problem, making no use of the prior information gained from

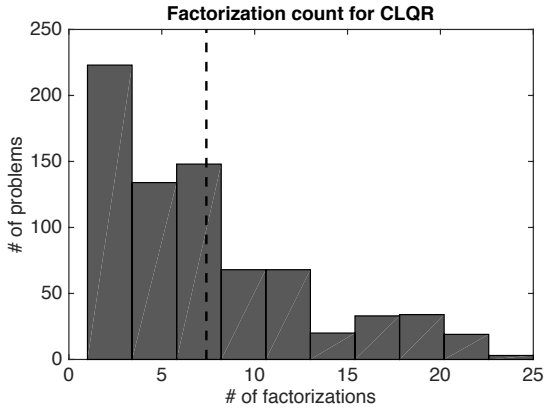


Fig. 10. Number of factorizations required over 750 problems. The mean value is 7 while no problem instance needed more than 25 factorizations.

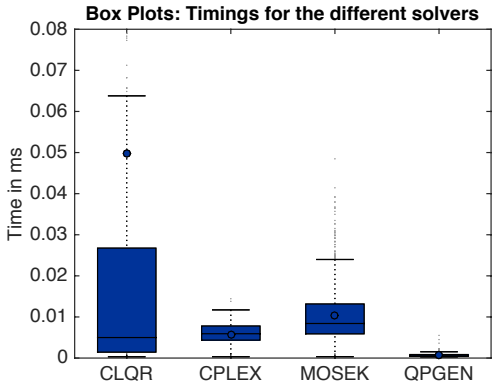


Fig. 11. Average time per optimal control problem solved with MPC, compared against one CLQR solve. In this case the three solvers are faster. The large variance of the CLQR approach in comparison to the others is a typical characteristic of first order methods, and can be significantly reduced with appropriately conditioning of the problem.

the convergence to the optimal multiplier sequence $\boldsymbol{\lambda}^\infty$.

In view of the facts that (i) the above packages run optimized C code and (ii) the employed algorithms differ quite significantly from our proximal method, we devise a simulation where we use *the exact same C++ code that we have developed for Algorithm 2, but rather in order to solve a series of finite horizon MPC problems. Ultimately, we apply the dual proximal gradient method to do MPC*. In this way we control for most of the irrelevant factors and focus on the comparison between CLQR and MPC. Once again, we consider the toy example and sample randomly 200 initial conditions from the set of the 750 ones utilized above. We inject disturbances with magnitude up to 3% of the predicted initial state and solve in closed loop until the initial state enters the terminal invariant set. We compare CLQR to MPC with horizon length $T = \lceil \frac{2}{3} T^\infty \rceil$ and terminal set constraint, as well as with $T = 2T^\infty$, without terminal set constraint, exactly as we did before. Regarding the algorithmic parameters, a is set to 5 and the stopping criterion is set to a very low tolerance ($\|\boldsymbol{\lambda}^k - \hat{\boldsymbol{\lambda}}^{k+1}\|/\|\boldsymbol{\lambda}^k\| \leq 10^{-12}$) so as to eliminate any numerical discrepancies resulting from early termination. We solve the Lagrangian minimization (Step 3 in Algorithm 2) by means of a Riccati recursion. Note that the complexity is similar to solving a linear system since only the affine parts of the Lagrangian function vary per iteration, allowing us to precompute the Riccati matrix and the feedback gain. For the same reasons, the same complexity holds when solving the MPC problem (see, *e.g.*, [22] for the time-varying Riccati recursion). We are interested in four metrics: The number of infeasible problems when doing MPC (indicator of the volume of the region of attraction), the number of iterations, the execution times and the quality of the controller (evaluated via the objective function value that the two schemes achieve). The results are summarized in Table I. Out of the 200 sampled initial conditions, 19 resulted in infeasible closed loop problems due to the additive state disturbance. The short-viewed MPC policy (3^{d} column) resulted in 69 infeasible problems. The times achieved are significantly higher than the CLQR times. The deterioration of the objective function value is negligible. When the horizon increases (4^{th} column), the average and median times go down, being comparable, but still not as good as the ones achieved with CLQR. The reduced iteration count is a result of the warm-starting policy which is more efficient when applying CLQR instead of MPC, in the sense that, firstly, the multipliers are initialized at their optimal values, and secondly, that the horizon (hitting time T^∞) shrinks as time goes by, essentially resulting in smaller QPs.

TABLE I
CLOSED LOOP COMPARISON BETWEEN CLQR AND MPC

	CLQR	**MPC** $\lceil \frac{2}{3} T^\infty \rceil$ **w/ T.S.**	**MPC** $2T^\infty$ **w/o T.S.**
Infeasible Problems	19	69	19
Times (ms)			
Average	124.8	169.9	92.1
Median	30.6	64.6	50.3
Iterations			
Average	1118.0	1537.8	1461.9
Median	290.0	591.0	317.0
Average Objective Increase	-	0.04%	-

VIII. CONCLUSION

This work presented an algorithmic scheme capable of solving the constrained linear quadratic regulator problem in real time. The algorithm is an accelerated version of the proximal gradient algorithm and belongs to the wider family of forward-backward splitting schemes. The approach is to write the problem in its condensed form and dualize, which leads to the minimization of an infinite-dimensional quadratic functional subject to non-positivity constraints. The resulting infinite dimensional problem can be tackled in finite dimensions by observing that the dual sequence has always only finitely many non-zero elements. The proposed algorithm makes no use of terminal invariant sets and provably converges to the optimal solution of the infinite-horizon problem. Regarding the implementation aspects, the algorithm can be highly competitive since it enjoys the convergence properties of optimal first order methods whose each iteration is computationally cheap. In

addition, it requires minimal a priori information since the most crucial quantities are computed online, and there are no unreasonable or conservative assumptions on the problem's structure.

The employed approach requires that the objective function is written as the sum of a smooth and a (possibly) nonsmooth function whose proximal operator is inexpensive to compute. Smoothness of the dual problem is recovered from strong convexity of the primal. The polytopic constraint set becomes a nonnegativity constraint when looking at the dual variables, resulting in a simple proximal operator.

There are mainly two ways to extend the constrained infinite horizon regulation result: Computational and theoretical enhancements. From a computational viewpoint, the number of iterations needed for convergence can be quite large for very unstable (large) systems, or for ill-conditioned problem data. To this end, preconditioning of the problem has to be considered. The several recent advancements regarding proximal gradient methods [31], [30] should be directly applicable to our approach, with the possible difficulty of adapting these results to infinite dimensional spaces. Theoretical extensions would include the tracking case, handling more general non-polytopic constraints, as well as the interesting possibility of applying the method to continuous time infinite horizon problems.

APPENDIX A
REQUIRED OPERATOR THEORY

The subsequent results hold for general real Hilbert spaces, including the special case of l^2_w we consider. For an in-depth treatment of (monotone) operator theory, the interested reader is referred to [7] and [32]. We write variables in normal font, and we use the bold font to describe the infinite-dimensional variables we are manipulating in our problem description.

Definition 1: The l^2-weighted (or l^2_w) real Hilbert space $\mathcal{H}$ is defined by

$$\mathcal{H} = \left\{ \boldsymbol{z} = (z_i)_{i\in\mathbb{N}} \ : \ \sum_{i=0}^{\infty} \|z_i\|_2^2 w^i < \infty \right\}, \quad w > 0 \ .$$

Definition 2: A linear operator (mapping) $F\colon \mathcal{H}_{\mathbf{1}} \to \mathcal{H}_{\mathbf{2}}$ between two Hilbert spaces is said to be bounded if the operator norm $\|F\|$ of F, defined as

$$\|F\| \coloneqq \sup_{\|x\|_{\mathcal{H}_{\mathbf{1}}}=1} \|Fx\|_{\mathcal{H}_{\mathbf{2}}} \ ,$$

satisfies $\|F\| < \infty$. The set of bounded operators between two Hilbert spaces $\mathcal{H}_{\mathbf{1}}$ and $\mathcal{H}_{\mathbf{2}}$ is denoted as $\mathcal{B}(\mathcal{H}_{\mathbf{1}}, \mathcal{H}_{\mathbf{2}})$.

Theorem 2: Let $\mathcal{H}_{\mathbf{1}}, \mathcal{H}_{\mathbf{2}}$ be real Hilbert spaces and $F \in \mathcal{B}(\mathcal{H}_{\mathbf{1}}, \mathcal{H}_{\mathbf{2}})$. The adjoint of F is the unique operator $F^\star \in \mathcal{B}(\mathcal{H}_{\mathbf{2}}, \mathcal{H}_{\mathbf{1}})$ that satisfies

$$\langle Fx, y\rangle = \langle x, F^\star y\rangle \quad \forall x \in \mathcal{H}_{\mathbf{1}}, \ \forall y \in \mathcal{H}_{\mathbf{2}} \ .$$

Moreover, $\|F\| = \|F^\star\|$.

Subsequently, we introduce the notions of weak and strong convergence.

Definition 3: Let $\mathcal{H}$ be a Hilbert space. We say that $(x^k)_{k\in\mathbb{N}}$ converges weakly to x if $\forall y \in \mathcal{H}$ $\langle y, x^k\rangle \xrightarrow{k\to\infty} \langle y, x\rangle$. We denote weak convergence as $x^k \rightharpoonup x$.

Definition 4: Let $(x^k)_{k\in\mathbb{N}}$ be a sequence in $\mathcal{H}$. Then $(x^k)_{k\in\mathbb{N}}$ converges strongly to x if $\|x^k - x\| \xrightarrow{k\to\infty} 0$. We denote strong convergence as $x^k \to x$.

Definition 5: An operator $F : \mathcal{H} \to \mathcal{H}$ is *positive-definite* if it is bounded, $F = F^\star$ and $\langle Fx, x\rangle \geq \alpha\|x\|_{\mathcal{H}}$ for some $\alpha > 0$, for all $x \in \mathcal{H}$.

We recall that a positive definite operator F is invertible and the inverse operator F^{-1} is bounded.

APPENDIX B
BOUNDEDNESS OF SEVERAL OPERATORS

Lemma 6: The operators $\boldsymbol{C}$ and $\boldsymbol{W}$ are bounded.

Proof: Boundedness of $\boldsymbol{W}$ is trivial since it is a diagonal operator with non-increasing elements on the diagonal.

The operator $\boldsymbol{C}$ can be expressed as the following sum:

$$\boldsymbol{C} = \begin{bmatrix} C_u & 0 & \cdots \\ 0 & 0 & \cdots \\ 0 & C_u & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & \cdots \\ C_xB & 0 & \cdots \\ 0 & 0 & \cdots \\ C_xAB & C_xB & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\boldsymbol{C_x}} ,$$

and by using the triangle inequality, we have

$$\|\boldsymbol{C}\| \leq \sigma_{\max}(C_u) + \|\boldsymbol{C_x}\|. \tag{22}$$

We thus have to show that $\boldsymbol{y} = \boldsymbol{C_x}\boldsymbol{u}$ is bounded, *i.e.*,

$$\sup_{\|\boldsymbol{u}\|_{\mathcal{H}_{\boldsymbol{u}}}=1} \|\boldsymbol{y}\|_{\mathcal{H}_{\boldsymbol{\lambda}}} < \infty.$$

In order not to carry the zero rows of $\boldsymbol{C_x}$, we define $\bar{\boldsymbol{y}} \in \bar{\mathcal{H}}_{\boldsymbol{\lambda}}$ by dropping the zero elements of $\boldsymbol{y}$. This infinite-dimensional vector consists of the elements $\bar{y}_i = [\boldsymbol{C_x}]_i\boldsymbol{u} = \sum_{j=0}^{i-1} C_xA^{i-j-1}Bu_j \in \mathbb{R}^{p_x}$. Note that

$$\sup_{\|\boldsymbol{u}\|_{\mathcal{H}_{\boldsymbol{u}}}=1} \|\bar{\boldsymbol{y}}\|_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} = \sup_{\|\boldsymbol{u}\|_{\mathcal{H}_{\boldsymbol{u}}}=1} \|\boldsymbol{y}\|_{\mathcal{H}_{\boldsymbol{\lambda}}} \ .$$

Focusing on the operator of interest, we have that

$$\|\bar{\boldsymbol{y}}\|_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} = \sqrt{\sum_{i=1}^{\infty} \|\bar{y}_i\|_2^2 w^i} = \sqrt{\sum_{i=1}^{\infty} w^i \| \sum_{j=0}^{i-1} C_xA^{i-j-1}Bu_j\|_2^2} \ . \tag{23}$$

Then

$$\begin{aligned} \|\bar{\boldsymbol{y}}\|^2_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} &= \sum_{i=1}^{\infty} w^i \| \sum_{j=0}^{i-1} C_xA^{i-j-1}Bu_j\|_2^2 \\ &= \sum_{i=1}^{\infty} \| \sum_{j=0}^{i-1} C_x(A\hat{w})^{i-j-1}Bu_j\hat{w}^{j+1}\|_2^2 \\ &= \hat{w}^2 \sum_{i=1}^{\infty} \| \sum_{j=0}^{i-1} C_x\hat{A}^{i-j-1}B\hat{u}_j\|_2^2 \ , \end{aligned}$$

where we introduced $\hat{w} = w^{1/2}$, $\hat{u}_j = u_j\hat{w}^j$ with $\hat{\boldsymbol{u}} \in l^2$ and $\hat{A} = A\hat{w}$. Observing the above expression, one can identify that $\sum_{j=0}^{i-1} C_x\hat{A}^{i-j-1}B\hat{u}_j$ is the *convolution sum* of the impulse response of the system

$$\Sigma \coloneqq \left(\begin{array}{c|c} \hat{A} & B \\ \hline C_x & 0 \end{array} \right)$$

with an input $\hat{u}$. More specifically, borrowing the notation from [33] we denote the impulse response of Σ as $h_{\Sigma,i} = C_x \hat{A}^{i-1} B s_i$, where $s_i = 1,\ i \geq 0$ is the unit step function. Then the convolution operator is defined as the linear map $S_\Sigma \colon \hat{\boldsymbol{u}} \rightarrow \bar{\boldsymbol{y}}$ with $\bar{y}_i = (h_\Sigma * \hat{\boldsymbol{u}})_i = \sum_{j=0}^{i-1} h_{\Sigma,i-j}\hat{u}_j = (S_\Sigma \hat{\boldsymbol{u}})_i$. Thus we have that

$$\|\bar{\boldsymbol{y}}\|^2_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} = \hat{w}^2 \sum_{i=1}^{\infty} \| \left(S_\Sigma \hat{\boldsymbol{u}}\right)_i \|_2^2 = \hat{w}^2 \|S_\Sigma \hat{\boldsymbol{u}}\|_2^2 \Rightarrow$$
$$\|\bar{\boldsymbol{y}}\|_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} = \hat{w}\|S_\Sigma \hat{\boldsymbol{u}}\|_2 \leq \hat{w} \sup_{\|\hat{\boldsymbol{u}}\|_2 \leq 1} \|S_\Sigma \hat{\boldsymbol{u}}\|_2 \ .$$

From the definition of the induced 2-norm of Σ, denoted here as $\|\Sigma\|_2$, we have that $\sup \|S_\Sigma \hat{\boldsymbol{u}}\|_2 = \|\Sigma\|_2 \|\hat{\boldsymbol{u}}\|_2$ and by assuming (without loss of generality) that $\|\hat{\boldsymbol{u}}\|_2 = \|\boldsymbol{u}\|_{\mathcal{H}_{\boldsymbol{u}}} = 1$, we end up having that $\|\bar{\boldsymbol{y}}\|_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} \leq \hat{w}\|\Sigma\|_2$. Finally, the operator $\boldsymbol{C_x}$ is bounded by the $\mathcal{H}_\infty$ norm of the transfer matrix $H_\Sigma(z)$, or $\|\bar{\boldsymbol{y}}\|_{\bar{\mathcal{H}}_{\boldsymbol{\lambda}}} \leq \hat{w}\mathcal{H}_\infty(\Sigma)$. Subsequently, we have from (22) that the operator $\boldsymbol{C}$ is bounded by $\sigma_{\max}(C_u) + \hat{w}\mathcal{H}_\infty(\Sigma)$. ■

Remark 8: Following the discussion from Section II, the weight $\hat{w}$ can be chosen to render any unstable system stable by shrinking the eigenvalues of the matrix A ($\hat{A} = \hat{w}A$).

Lemma 7: The operator $\boldsymbol{H}^{-1}$ is bounded and $\|\boldsymbol{H}\| \leq 1/\lambda_{\min}(R)$.

Proof: The operator $\boldsymbol{H}$ is given by $\boldsymbol{H} = \boldsymbol{B}^\star \boldsymbol{Q} \boldsymbol{B} + \boldsymbol{R}$; see (2). Since $\boldsymbol{B}^\star \boldsymbol{Q} \boldsymbol{B}$ is positive semidefinite (i.e., $\langle \boldsymbol{B}^\star \boldsymbol{Q} \boldsymbol{B} \boldsymbol{u}, \boldsymbol{u} \rangle \geq 0$ for all $\boldsymbol{u} \in \mathcal{H}_{\boldsymbol{u}}$) and $\boldsymbol{R}$ is positive definite according to Definition 5, we conclude that $\boldsymbol{H}$ is positive definite and hence has a bounded inverse.

In order to compute the bound for $\boldsymbol{H}^{-1}$ we observe that $\|(\boldsymbol{B}^\star \boldsymbol{Q} \boldsymbol{B} + \boldsymbol{R})^{-1}\| \leq \|\boldsymbol{R}^{-1}\| \leq 1/\lambda_{\min}(R)$, which concludes the proof. ■

APPENDIX C
BACKTRACKING STEPSIZE RULE

In this appendix we briefly revise the backtracking stepsize rule that allows for local estimates of the curvature of $h^\star$ and show that points (i), (iii) and (iv) of Theorem 1 also hold in this case. The arguments are in line with [17].

We denote as $f(\cdot)$ the smooth and $g(\cdot)$ the nonsmooth convex functions of interest. Note that in our case $f = h^\star$ and $g = \delta_-$. For any $L > 0$ consider the quadratic approximation of $F(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}) + g(\boldsymbol{\lambda})$ at a point $\boldsymbol{y}$:

$$Q_L(\boldsymbol{\lambda}, \boldsymbol{y}) := f(\boldsymbol{y}) + \langle \boldsymbol{\lambda} - \boldsymbol{y}, \nabla f(\boldsymbol{y}) \rangle + \frac{L}{2}\|\boldsymbol{\lambda} - \boldsymbol{y}\|^2 + g(\boldsymbol{\lambda}) \ . \tag{24}$$

We also define the unique minimizer parametrized by the point $\boldsymbol{y}$ as

$$p_L(\boldsymbol{y}) := \operatorname*{argmin}_{\boldsymbol{\lambda}} \{Q_L(\boldsymbol{\lambda}, \boldsymbol{y})\} \tag{25}$$
$$= \operatorname*{argmin}_{\boldsymbol{\lambda}} \left\{ g(\boldsymbol{\lambda}) + \frac{L}{2}\|\boldsymbol{\lambda} - \left(\boldsymbol{y} - \frac{1}{L}\nabla f(\boldsymbol{y})\right)\|^2 \right\} \tag{26}$$
$$= \mathbf{prox}_g \left(\boldsymbol{y} - \frac{1}{L}\nabla f(\boldsymbol{y})\right) \ , \tag{27}$$

which is the basic step of Algorithm 1, *i.e.*, Step 3. The following holds:

Lemma 8 (Lemma 2.3 [17]): Let $\boldsymbol{y} \in \mathcal{H}_{\boldsymbol{\lambda}}$ and $L > 0$ be such that

$$F(p_L(\boldsymbol{y})) \leq Q_L(p_L(\boldsymbol{y}), \boldsymbol{y}) \ .$$

Then for any $\boldsymbol{\lambda} \in \mathcal{H}_{\boldsymbol{\lambda}}$,

$$F(\boldsymbol{\lambda}) - F(p_L(\boldsymbol{y})) \geq \frac{L}{2}\|p_L(\boldsymbol{y}) - \boldsymbol{y}\|^2 + L\langle \boldsymbol{y} - \boldsymbol{\lambda}, p_L(\boldsymbol{y}) - \boldsymbol{y} \rangle \ .$$

The backtracking procedure as described in [17] is as follows:

Algorithm 3 Backtracking for stepsize computation

0: Take $L^0 > 0$, some $\eta > 1$, and $\boldsymbol{\lambda}^0 \in \mathcal{H}_{\boldsymbol{\lambda}}$.
repeat
1: Find the smallest nonnegative integer i^k such that with $\bar{L} = \eta^{i^k} L^{k-1}$
until $F(p_{\bar{L}}(\boldsymbol{y}^k)) \leq Q_{\bar{L}}(p_{\bar{L}}(\boldsymbol{y}), \boldsymbol{y})$

Note that for any $L^k = \bar{L}$ generated by Algorithm 3 Lemma 8 holds.

We also have the following instrumental Lemma from [13]:

Lemma 9 (Lemma 1 [13]): Let $L \geq L(f)$, $\boldsymbol{\lambda}, \boldsymbol{y} \in \mathcal{H}_{\boldsymbol{\lambda}}$ and $p_L(\boldsymbol{y}) := \mathbf{prox}_g(\boldsymbol{y} - \frac{1}{L}\nabla f(\boldsymbol{y}))$. Then for all $\boldsymbol{\lambda}$

$$F(p_L(\boldsymbol{y})) + \frac{L}{2}\|p_L(\boldsymbol{y}) - \boldsymbol{\lambda}\|^2 \leq F(\boldsymbol{y}) + \frac{L}{2}\|\boldsymbol{\lambda} - \boldsymbol{y}\|^2 \ .$$

Supposing that Lemma 9 holds, convergence of the function values with rate $1/k^2$ can be proven under no further assumptions using Theorem 2 and Corollary 1 in [13]. We will not repeat the aforementioned theorems here due to space limitations. What we are going to show instead is that all stepsizes $\rho^k = 1/L^k$ generated from Algorithm 3 satisfy Lemma 9.

Lemma 10: Consider $F = f + g$, with $f = h^\star$ and $g = \delta_-$ as defined in Section III. The iterates $\boldsymbol{\lambda}^k$ generated from Algorithm 1 with a backtracking stepsize rule generated from Algorithm 3 satisfy:

$$F(\boldsymbol{\lambda}^k) - F(\boldsymbol{\lambda}^\infty) \leq \frac{a^2 \bar{L}}{2(k + a - 1)^2}\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^\infty\|^2 \ .$$

Proof: All local Lipschitz estimates $\bar{L}$ generated from Algorithm 3 satisfy $F(p_{\bar{L}}(\boldsymbol{y}^k)) \leq Q_{\bar{L}}(p_{\bar{L}}(\boldsymbol{y}), \boldsymbol{y})$ and, consequently, from Lemma 8,

$$F(\boldsymbol{\lambda}) - F(p_{\bar{L}}(\boldsymbol{y})) \geq \frac{\bar{L}}{2}\|p_{\bar{L}}(\boldsymbol{y}) - \boldsymbol{y}\|^2 + \bar{L}\langle \boldsymbol{y} - \boldsymbol{\lambda}, p_{\bar{L}}(\boldsymbol{y}) - \boldsymbol{y} \rangle \ ,$$

or

$$\frac{2}{\bar{L}}\left(F(\boldsymbol{\lambda}) - F(p_{\bar{L}}(\boldsymbol{y}))\right) \geq \|p_{\bar{L}}(\boldsymbol{y}) - \boldsymbol{y}\|^2 + 2\langle \boldsymbol{y} - \boldsymbol{\lambda}, p_{\bar{L}}(\boldsymbol{y}) - \boldsymbol{y} \rangle \ .$$

It follows from the Pythagorean theorem that

$$\|b - a\|^2 + 2\langle b - a, a - c \rangle = \|b - c\|^2 - \|a - c\|^2 \ ,$$

and hence

$$\frac{2}{\bar{L}}\left(F(\boldsymbol{\lambda}) - F(p_{\bar{L}}(\boldsymbol{y}))\right) \geq \|p_{\bar{L}}(\boldsymbol{y}) - \boldsymbol{\lambda}\|^2 - \|\boldsymbol{y} - \boldsymbol{\lambda}\|^2 \ ,$$

from which Lemma 9 follows. Theorem 2 and Corollary 1 of [13] then lead to the desired result. ■

ACKNOWLEDGEMENTS

The authors would like to thank J.H. Hours for many helpful discussions about the resulting optimization problems.

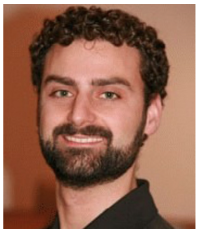
The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ ERC Grant Agreement n. 307608.

REFERENCES

[1] R. Kalman, "Contributions to the theory of optimal control," *Boletin de la Sociedad Matematica Mexicana*, 1960.
[2] D. Chmielewski and V. Manousiouthakis, "On constrained infinite-time linear quadratic optimal control," *Systems & Control Letters*, vol. 29, no. 3, pp. 121 – 129, 1996.
[3] P. Scokaert and J. B. Rawlings, "Constrained Linear Quadratic Regulation," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1163–1169, 1998.
[4] P. Grieder, F. Borrelli, F. Torrisi, and M. Morari, "Computation of the constrained infinite time linear quadratic regulator," *Automatica*, vol. 40, no. 4, pp. 701 – 708, 2004.
[5] M. Sznaier and M. Damborg, "Suboptimal control of linear systems with state and control inequality constraints," in *IEEE Conference on Decision and Control*, 1987.
[6] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, 2014.
[7] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.
[8] D. A. Lorenz and T. Pock, "An accelerated forward-backward algorithm for monotone inclusions," *arXiv preprint arXiv:1403.3522*, 2014.
[9] G. Stathopoulos, M. Korda, and C. N. Jones, "Solving the infinite-horizon constrained LQR problem using splitting techniques," in *19th IFAC World Congress*, 2014.
[10] P. Tseng, "Applications of splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim.*, vol. 29, no. 1, pp. 119–138, 1991.
[11] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *Automatic Control, IEEE Transactions on*, vol. 59, no. 1, pp. 18–33, 2014.
[12] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.
[13] A. Chambolle and C. Dossal, "On the convergence of the iterates of "FISTA"," 2014. [Online]. Available: https://hal.inria.fr/hal-01060130
[14] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics* , vol. 4, no. 5, pp. 1 – 17, 1964.
[15] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Dokl. Akad. Nouk SSSR*, 1983.
[16] ——, "Gradient methods for minimizing composite objective function," CORE, Catholic University of Louvain, Tech. Rep, Tech. Rep., 2007.
[17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, 2009.
[18] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004, vol. 87.
[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[20] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, vol. 42, no. 1, pp. 1–6, 2014.
[21] B. O'Donoghue, G. Stathopoulos, and S. P. Boyd, "A splitting method for optimal control." *IEEE Trans. Contr. Sys. Techn.*, vol. 21, no. 6, pp. 2432–2442, 2013.
[22] G. Frison and J. B. Jørgensen, "Efficient implementation of the Riccati recursion for solving linear-quadratic control problems." in *CCA*. IEEE, 2013, pp. 1117–1122.
[23] D. Axehill and M. Morari, "An alternative use of the riccati recursion for efficient optimization," *Systems & Control Letters*, vol. 61, no. 1, pp. 37 – 40, 2012.
[24] G. Frison and J. B. Jørgensen, "A fast condensing method for solution of linear-quadratic control problems," in *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy*, 2013, pp. 7715–7720.
[25] S. Richter, C. N. Jones, and M. Morari, "Real-time input-constrained MPC using fast gradient methods," in *Proceedings of the 48th IEEE Conference on Decision and Control, CDC 2009, combined withe the 28th Chinese Control Conference, December 16-18, 2009, Shanghai, China*, 2009, pp. 7387–7393.
[26] G. Stathopoulos, A. Szucs, Y. Pu, and C. N. Jones, "Splitting methods in control," in *Proceedings of the 13th IEEE European Control Conference, ECC 2014, Strasbourg, France*, 2014, pp. 2478 – 2483.
[27] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/ mpt.
[28] C. Sanderson, "Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments," NICTA, Tech. Rep., 2010.
[29] P. Giselsson and S. Boyd, "Diagonal scaling in Douglas-Rachford splitting and ADMM," *53rd IEEE Conference on Decision and Control*, 2014.
[30] ——, "Metric selection in fast dual forward backward splitting," 2014, manuscript submitted for publication to Automatica.
[31] ——, "Monotonicity and restart in fast gradient methods," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 2014, pp. 5058–5063.
[32] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, 1992.
[33] A. C. Antoulas, "Approximation of linear dynamical systems," in *Encyclopedia of Electrical and Electronics Engineering*. John Wiley and Sons, 1998, pp. 403–422.

Giorgos Stathopoulos Giorgos graduated with a Diploma in Electrical and Computer Engineering from University of Patras, Greece in 2009. In 2012 he received an MSc degree in Systems and Control from TU Delft in the Netherlands. During that period he was also a visitor at Stanford University, where he conducted his MSc thesis. He is currently interested in optimisation and control with application on networks of smart buildings and is pursuing a Ph.D. degree at the Automatic Control Laboratory, EPFL, Lausanne, Switzerland.

Milan Korda Milan Korda received the B.S. degree in electrical engineering and the M.S. degree in cybernetics and robotics from the Czech Technical University, Prague, in 2009 and 2011, respectively, and is currently pursuing the Ph.D. degree in polynomial optimization and stochastic model predictive control at the Automatic Control Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.



Colin N. Jones Colin N. Jones received the Bachelors degree in Electrical Engineering and the Masters degree in Mathematics from the University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree from the University of Cambridge, Cambridge, UK in 2005. He is an Assistant Professor in the Automatic Control Laboratory at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. He was a Senior Researcher at the Automatic Control Laboratory of the Swiss Federal Institute of Technology Zurich until 2010. His current research interests are in the areas of high-speed predictive control and optimization, as well as green energy generation, distribution and management.