Free finite horizon LQR: a bilevel perspective and its application to model predictive control

Alberto De Marchi* Matthias Gerdts

November 6, 2018

Abstract

We consider linear-quadratic optimal control problems with free final time and terminal state constraints and propose a solution procedure that is particularly useful for online feedback control in a model-predictive control (MPC) framework. The procedure avoids the standard time transformation, which transforms the problem into an equivalent but non-convex optimal control problem on a fixed time horizon. The transformed problem typically suffers from many local minima, which might cause instabilities in online optimization tasks like LQR or model-predictive control. To avoid this drawback of the time transformation we develop a method from the viewpoint of bilevel optimal control, which is beneficial especially in online control tasks. The novelty of the approach is the optimal final time tracking procedure, for which we exploit a property of the Hamiltonian function. To this end we show that within the continuous-time closed-loop controlled system the optimal final time linearly decreases in time, as one could intuitively expect. Finally, numerical experiments support the effectiveness of the proposed algorithm.

Keywords: Optimal control; Linear quadratic regulator; Free final time; Model predictive control.

1 Introduction

The linear quadratic regulator (LQR) (or Riccati controller) is a well-known and frequently used model-based feedback controller for tracking problems on a finite or infinite time horizon. Herein, the control law is derived by solving a linear-quadratic optimal control problem on a fixed and finite time horizon using a Riccati differential equation or on an infinite time horizon using the corresponding algebraic Riccati equation, compare [6, 10, 11]. Owing to its simplicity and since many standard tools for it exist, the LQR is often adopted within the (linear) model predictive control (MPC) framework, which is an advanced control technique, also known as moving or receding horizon control, compare [13, 4, 9]. Depending on the nonlinearities of the real system, a linear model might be not

*Bundeswehr University Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg/Munich, Germany. Corresponding author: alberto.demarchi@unibw.de.
DOI: 10.1016/j.automatica.2018.11.032.

accurate enough; in such cases, beyond the scope of this paper, one could adopt methods suitable for nonlinear optimization, e.g. full discretization and dynamic programming, see [16, 11, 7], or methods using sensitivity updates in combination with precomputed solutions in a database, see [14]. The MPC framework determines the system input by solving on-line, at every time-step, an open-loop optimal control problem (OCP) given the current state of the system. Typically, this optimization phase generates an input sequence on a specified finite time horizon (the prediction horizon), but only the first control input is actually commanded to the system. MPC allows to consider control or state constraints and, when applied on a finite horizon, even terminal state constraints. The latter may occur as hard constraints or as soft constraints using penalization. Often terminal constraints lead to improved stability properties of the control system, see [13, 2].

This work focuses on the free finite time-horizon LQR problem, referred to as FLQR, that is an OCP with free final time, i.e. with *a priori* unknown final time, and a cost for elapsed time. In contrast to LQR, FLQR is in general a nonlinear and non-convex problem, possibly with multiple local and global minima. It is possible to reformulate FLQR as a fixed final time OCP, e.g. through the widely adopted time transformation technique [7], but this introduces nonlinear dynamics not present in the original problem. Moreover, in case direct methods are chosen to solve FLQR, the numerical solution becomes very sensitive to the initial guess and it is likely to end up in a local minimum. In both cases, global optimality cannot be guaranteed and stability issues may arise, if the FLQR has to be solved many times, as it is the case in the MPC framework. There are also approaches, especially in automotive and robotic applications, that use a spatial transformation, e.g. through a curvilinear abscissa, and thus avoid time as the independent variable [3, 17]; these, however, are not of general applicability.

Within this work, we adopt a bilevel optimal control perspective in order to avoid the aforementioned time transformation and its drawbacks while exploiting the standard tools available for LQR. At the upper level of the bilevel optimal control problem, the final time is optimized, while satisfying constraints and system dynamics, enforced at the lower level. This idea proves to be particularly effective when FLQR is adopted within the MPC framework, in which case our approach can be interpreted as a tracking strategy for the optimal final time while solving standard LQR problems, at the upper and lower level respectively.

This approach recalls and takes inspiration from initial value embedding in real-time iteration scheme [5], sensitivity update [7] and predictor-corrector numerical methods. Also, it takes advantage of the relationship between value function and reduced cost function in order to tackle the original single level problem [15]. Van den Broeck et al. [4] introduced the Time Optimal MPC for mechatronic applications in which they also reformulate the problem as a two-level procedure, but the two levels have different objectives in there, being the time optimality the main goal. However, in order to avoid too nervous and aggressive maneuvers, this is combined with a quadratic cost function, which takes into account the control effort. The FLQR problem differs in that it always considers a trade-off between minimal time and input costs.

In the bilevel optimization setting, the cost function can be reduced, namely, in our case, it can be considered as a function of final time only, while constraints are taken care of and satisfied at the lower level. With this in mind, then, the bilevel perspective is supported by a novel result, presented in Theorem 1, that relates the gradient of the reduced cost function with the Hamilton function of the original problem. This allows to efficiently search for the optimal final time with gradient-based methods. This result also offers a new interpretation of the condition of vanishing Hamiltonian for autonomous free final time OCPs. Another novel outcome corroborates the suggested bilevel approach, especially for the MPC implementation. In fact, tracking the optimal final time turns out to be straightforward and effective because, owing to Theorem 2, it flows opposite to time.

This paper is organized as follows. It contains two major parts. Section 2 contains the derivation of a bilevel-optimization approach for free-time LQR problems, which we propose to overcome difficulties with local minima that might be introduced by standard time transformation techniques for problems with free final time. To this end the problem formulation and standing assumptions are introduced. The solution approach is briefly delineated and the bilevel problem is formulated. Specifically, Section 2.3 discusses about the fixed final time problem and its transformation into a linear system, while Section 2.5 analyses the free final time problem from the upper level point-of-view, providing a theoretical result supporting the bilevel perspective. The method derived in Section 2 has merits in its own rights as a generally applicable method for free-time LQR problems. In addition, we believe it is very well suited within free-time model predictive control, since the approach allows to track the final time in a more robust way than standard methods. The embedding of the approach into the MPC framework is detailed in Section 3. Section 3 examines how the bilevel problem structure can be exploited to fit the MPC framework. Section 4 validates the proposed approach numerically on a standard problem, showing effectiveness and limitations of the proposed algorithm. Section 5 concludes the paper and presents ideas for future research.

2 Free finite horizon LQR and the bilevel solution approach

The purpose of this section is to derive an algorithm for the solution of linear-quadratic optimal control problems with free final time. In contrast to standard techniques, we do not employ a standard time transformation to a fixed time interval, which leads to a nonlinear, nonconvex optimal control problem. Instead, we develop a method that considers the free-time LQR as a bilevel optimization problem. By this, the upper level problem becomes a scalar optimization problem aiming at finding the optimal final time as a root of the Hamilton function while the lower level problem is a standard LQR on a fixed time horizon. This splitting allows to exploit better the nice convexity properties of the lower level LQR problem.

2.1 Problem statement

Consider a time interval $[0, T]$, $T > 0$, and the linear time-invariant state differential equation

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

with boundary conditions

$$x(0) = x_0 \ , \qquad\qquad C_{\mathrm{T}} x(T) = y_{\mathrm{T}} \ , \tag{2}$$

where, for any time t, $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C_{\mathrm{T}} \in \mathbb{R}^{l\times n}$. Let

$$\Pi := \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix} \tag{3}$$

with $Q \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{m\times m}$. Let $Q_{\mathrm{T}} \in \mathbb{R}^{n\times n}$ be symmetric and $\xi_{\mathrm{T}} \in \mathbb{R}^n$. Let $\Gamma = \mathbb{R}_{>0}$ be the set of allowable final times. Finally, let $T \in \Gamma$ be the length of the time horizon, $X(T) := W^{1,\infty}([0, T], \mathbb{R}^n)$ the Sobolev space of vector-valued absolutely continuous functions with essentially bounded first derivative on $[0, T]$, $U(T) := L^\infty([0, T], \mathbb{R}^m)$ the vector space of vector-valued essentially bounded measurable functions on $[0, T]$, and consider the cost functional $\mathcal{J} : X(T) \times U(T) \times \Gamma \to \mathbb{R}$:

$$\mathcal{J}(x, u, T) := wT + \frac{1}{2}\int_0^T \begin{pmatrix} x(\tau) \\ u(\tau) \end{pmatrix}^\top \Pi \begin{pmatrix} x(\tau) \\ u(\tau) \end{pmatrix} \, \mathrm{d}\tau + \frac{1}{2}\left(x(T) - \xi_{\mathrm{T}}\right)^\top Q_{\mathrm{T}} \left(x(T) - \xi_{\mathrm{T}}.\right) \tag{4}$$

Herein and throughout the paper we assume $w > 0$, that is, the free final time contributes to the cost. We note that the case $w = 0$ in combination with the constraint $T \in [0, T_u]$, with a finite upper bound $T_u > 0$, can be handled as well. In fact, as derived in Appendix B, the latter case is included as a special case in our algorithm. For notational convenience we prefer to restrict the following discussion to the case $w > 0$. The OCP dealt with in this paper is the following:

Problem 1. *Find a final time $T \in \Gamma$, a control $u \in U(T)$, and a state $x \in X(T)$, minimizing the cost functional $\mathcal{J}$ in (4) while satisfying the constraints (1)–(2).*

In this paper it is assumed that

(A1) the pair (A, B) is controllable,

(A2) the matrix R is positive definite and the matrix Π is symmetric and positive semi-definite,

(A3) initial and final conditions do not match, namely $C_{\mathrm{T}} x_0 \neq y_{\mathrm{T}}$,

(A4) the set of optimal final times for Problem 1 has zero measure.

Remark 1. *The method proposed in this paper is particularly suitable and motivated by online control purposes. Hence, we restrict ourselves to this case, in which the initial state of the system is given and an affine constraint on the state might be imposed at the final time as in (2). However, the analysis can be easily extended to deal with the free final time version of more general finite-horizon LQR problems, e.g. [6].*

Remark 2. *Assumption (A3) is imposed in order to neglect the trivial case with zero final time. Hence, it is clear that the optimal final time T is within the open set Γ.*

Remark 3. *Problems arising from applications might require a solution with box-constrained final time, say $\Gamma = [T_{min}, T_{max}]$ with $0 \leq T_{min} < T_{max} < +\infty$. Since the final time is a scalar, a minimizer can be found by inspection of the boundaries T_{min} and T_{max} and the interior of Γ. The latter is the challenging task and thus the focus is on the interior of Γ, which is considered to be the set of positive real numbers for simplicity.*

Remark 4. *Assumption (A4) excludes the cases in which the optimal final times are not separated. Equivalently, the cost function should have nonzero total second derivative with respect to the final time in a neighborhood of each and every solution. To give an intuition behind this requirement, we can say that it is needed in order to never have a plateau of the optimal cost with respect to the final time in the vicinity of a solution. Instead, if (A4) is not satisfied, chattering of the solution may arise, because of the many equivalent optima. However, we argue that (A4) likely holds for real applications and that pathological cases, if possible, are rare. Anyway, a straightforward remedy is, e.g., to slightly modify the time cost $w > 0$.*

Problems with null time cost, i.e. $w = 0$, might be of interest; in these cases, existence of a solution is guaranteed only if Γ is a compact set. Appendix B briefly addresses this case, discussing necessary conditions and a suitable projection scheme.

2.2 Approach: bilevel perspective

Problem 1 can be casted into an equivalent bilevel optimization problem, whose upper level reads:

Problem 2. *Find a $T \in \Gamma$ minimizing the cost $\tilde{\mathcal{J}}(T) := \mathcal{J}(x_T, u_T, T)$, with $(x_T, u_T) \in \mathcal{M}(T)$,*

where $\mathcal{M}(T) \in X(T) \times U(T)$ is the set of minimizers of the lower level:

Problem 3. *Given $T \in \Gamma$, find a $u \in U(T)$ and a $x \in X(T)$ minimizing the cost $\mathcal{J}(x, u, T)$ under the constraints* (1)–(2).

Note that the Upper Level Problem 2 is a scalar optimization problem for the single optimization variable T, while the Lower Level Problem 3 is an almost standard linear-quadratic optimal control problem on a fixed and finite time horizon, but with a terminal state constraint.

Function $\tilde{\mathcal{J}}$ in Problem 2 is the aforementioned reduced cost function, that is the cost from the upper level point-of-view. This paper is mainly focused on exploiting the problem structure for online control in an MPC framework; nonetheless, the suggested bilevel perspective might be useful for solving FLQR offline, too. In particular, compared to standard techniques for free final time OCPs, e.g. the time transformation [7], its main advantage is that one has to solve standard LQR problems and to compare costs of their (unique and globally optimal) solution, instead of dealing with a nonlinear OCP and opening the door to multiple local optima.

Remark 5. *One should point out that, at least for the offline solution, the number of LQR problems to be solved might be very large, depending on the required accuracy. Indeed, a greedy optimization of the final time cannot guarantee* a priori *optimality. However, owing to the bilevel perspective, also off-the-shelves tools for scalar static optimization can be adopted to solve this kind of problems. Any of these methods can be adopted in order to get an offline solution of FLQR needed to initialize the online process within the MPC framework.*

Remark 6. *Let us consider the nonlinear OCP resulting from the transformation of a free final time OCP into one with fixed final time. In general, nonlinear optimization techniques cannot guarantee that, given an initial guess close enough to a global optimum, say $T^\star$, they will return an approximate solution close to $T^\star$. Instead, under mild assumptions for the local convergence of Newton-type methods, the bilevel-inspired algorithm presented in the following always returns an approximate solution close to $T^\star$. Thus, for an online execution with small enough disturbances, an initially global optimum remains a global optimum.*

2.3 Fixed time LQR: the lower level problem

Let us solve Problem 3 for a given $T \in \Gamma$, i.e. considering T known, fixed and thus not an optimization variable. The Hamilton function $\mathcal{H} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ for this problem is defined as [16, 7]:

$$\mathcal{H}(x, u, \lambda) := w + \frac{1}{2} \begin{pmatrix} x \\ u \end{pmatrix}^\top \Pi \begin{pmatrix} x \\ u \end{pmatrix} + \lambda^\top \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \tag{5}$$

Let us denote $(x^\star, u^\star, T^\star)$ a solution to Problem 1. Notably, the Hamilton function (5) does not depend on the final time T. This means that the optimal control $u^\star$ does not depend explicitly on $T^\star$. In fact, from the Pontryagin's Minimum Principle, the optimal control $u^\star$ minimizes the Hamiltonian $\mathcal{H}$ [16, 7], namely $u^\star(t) = \underset{v \in \mathbb{R}^m}{\operatorname{argmin}}\ \mathcal{H}(x^\star(t), v, \lambda^\star(t))$. Since the control is unconstrained and Problem 3 is convex owing to Assumption (A2), and thus the Hamiltonian is strictly convex w.r.t. u, the minimizer of the Hamiltonian exists and is unique, and it is found by solving the equation $\nabla_u \mathcal{H}(x^\star(t), u, \lambda^\star(t)) = 0$ with respect to u, compare Equation (6). In the following we might consider the optimal control expressed as a feedback control, $u_{\text{fb}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$, defined in (7), such that $u_{\text{fb}}(x^\star(t), \lambda^\star(t)) = u^\star(t)$ for a.e. $t \in [0, T]$:

$$u^\star(t) = -R^{-1} \left[S x^\star(t) + B^\top \lambda^\star(t) \right] , \tag{6}$$

$$u_{\text{fb}}(x, \lambda) := -R^{-1} \left[S x + B^\top \lambda \right] . \tag{7}$$

First-order necessary optimality conditions consist of system dynamics, adjoint equation and transversality conditions; these are obtained from first variation of the Lagrangian and Du Bois lemma [16, 7]. A solution to Problem 3 necessarily satisfies (1), (2) and (6) along with the following adjoint equation and transversality condition [16, 7]:

$$\dot{\lambda}(t) = -Qx(t) - S^\top u(t) - A^\top \lambda(t) \qquad t \in [0,T) \ , \tag{8}$$

$$\lambda(T) = Q_\mathrm{T} \left[x(T) - \xi_\mathrm{T}\right] + C_\mathrm{T}^\top \eta \ , \tag{9}$$

with costate (or adjoint) $\lambda : [0,T) \to \mathbb{R}^n$ and multiplier $\eta \in \mathbb{R}^l$. Plugging-in the optimal feedback control u_fb (7) in place of control u into state and costate dynamics (1),(8), one obtains the Hamiltonian system:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{pmatrix} = M \begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} \ , \tag{10}$$

where the matrix M and its $n \times n$ sub-blocks are defined by

$$M := \begin{bmatrix} M_{xx} & M_{x\lambda} \\ M_{\lambda x} & M_{\lambda\lambda} \end{bmatrix} := \begin{bmatrix} A - BR^{-1}S & -BR^{-1}B^\top \\ -Q + S^\top R^{-1} S & -A^\top + S^\top R^{-1} B^\top \end{bmatrix} \ . \tag{11}$$

Given the linear homogeneous autonomous dynamics (10) and initial conditions at time $t = 0$, $x(t)$ and $\lambda(t)$ can be evaluated through the matrix exponential [10], namely

$$\begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} = \mathrm{e}^{Mt} \begin{pmatrix} x(0) \\ \lambda(0) \end{pmatrix} \tag{12}$$

for any time $t > 0$. Initial conditions for the state are already provided by Problem 3. Instead, those for the costate can be obtained by exploiting boundary conditions (2), (9), that are necessary conditions. Let us denote λ_0 the value $\lambda^\star(0)$ and introduce matrix valued functions $E_{ij} : \mathbb{R} \to \mathbb{R}^{n\times n}$, $i, j \in \{x, \lambda\}$, such that

$$\begin{bmatrix} E_{xx}(t) & E_{x\lambda}(t) \\ E_{\lambda x}(t) & E_{\lambda\lambda}(t) \end{bmatrix} := \mathrm{e}^{Mt} \tag{13}$$

for any $t \in \mathbb{R}$. Substituting (12), (13) into boundary conditions (2), (9) and rearranging yields a linear system in unknowns λ_0, η:

$$\begin{bmatrix} C_\mathrm{T} E_{x\lambda}(T) & 0 \\ Q_\mathrm{T} E_{x\lambda}(T) - E_{\lambda\lambda}(T) & C_\mathrm{T}^\top \end{bmatrix} \begin{pmatrix} \lambda_0 \\ \eta \end{pmatrix} = \begin{pmatrix} y_\mathrm{T} - C_\mathrm{T} E_{xx}(T) x_0 \\ Q_\mathrm{T} \xi_\mathrm{T} + \left[E_{\lambda x}(T) - Q_\mathrm{T} E_{xx}(T)\right] x_0 \end{pmatrix} \ . \tag{14}$$

The linear system (14) is square of size $n + l$, thus independent on the number of controls m. Given a final time T and an initial state x_0, the corresponding initial costate λ_0 can be evaluated by solving (14) and then adopted to compute the optimal control at the initial time through the feedback law (7), namely $u^\star(0) = u_\mathrm{fb}(x_0, \lambda_0)$. Furthermore, it is possible to evaluate state, costate and control trajectory through (12) and (7).

In order to state explicitly the dependence of λ_0 on T and x_0, let us denote $\hat{A}(T)$ and $\hat{b}(T, x_0)$ the matrix and the vector, respectively, in (14). The (unique) solution to the linear equation system $\hat{A}(T)s = \hat{b}(T, x_0)$, denoted $\hat{s}(T, x_0)$, encapsulates the initial costate vector and depends on both final time and initial state. The vector y_T is here considered given and fixed, but it could also be easily accounted for. In the following, for compactness, we will consider the function $\hat{\lambda} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$, such that $\hat{\lambda}(T, x) := \begin{bmatrix} I_n & 0 \end{bmatrix} \hat{s}(T, x)$ for any T and x, to express the relationship between final time, initial state and initial costate. Thanks to the linearity in (14), function $\hat{s}$ and its derivatives can be explicitly represented and easily evaluated; see Algorithm 1. Notice that evaluating the d-th derivative of $\hat{A}$ and $\hat{b}$ with respect to final time require d matrix-multiplications with M, defined in (11). Also, solving linear systems to obtain derivatives of $\hat{s}$ at (T, x_0) can be sped up by adopting and re-using a suitable decomposition of the matrix $\hat{A}(T)$, e.g. LU factorization with pivoting. Then, from these observations, it turns out that there is little overhead for evaluating derivatives when the function $\hat{\lambda}$ is executed.

Algorithm 1 Pseudocode for evaluating initial costate and its derivatives.

```
Input: $T$, $x$, $M$
Output: $\lambda, \lambda'_T, \lambda''_{TT}, \lambda'_x$
  $E = \mathrm{e}^{MT}$                                              ▷ matrix exponential
  $\hat{A}, \hat{b} \leftarrow E, x$                                            ▷ Eq. 14
  $\hat{s} = \hat{A}^{-1}\hat{b}$
  $\lambda = \begin{bmatrix} I_n & 0 \end{bmatrix} \hat{s}$
  $\hat{A}'_T, \hat{b}'_T, \hat{A}''_{TT}, \hat{b}''_{TT}, \hat{b}'_x \leftarrow E, x$                      ▷ based on Eq. 14
  $\hat{s}'_T = \hat{A}^{-1}\left(\hat{b}'_T - \hat{A}'_T \hat{s}\right)$
  $\lambda'_T = \begin{bmatrix} I_n & 0 \end{bmatrix} \hat{s}'_T$
  $\lambda''_{TT} = \begin{bmatrix} I_n & 0 \end{bmatrix} \hat{A}^{-1}\left(\hat{b}''_{TT} - \hat{A}''_{TT}\hat{s} - 2\hat{A}'_T \hat{s}'_T\right)$
  $\lambda'_x = \begin{bmatrix} I_n & 0 \end{bmatrix} \hat{A}^{-1}\hat{b}'_x$
```

2.4 Reduced cost and Hamiltonian

An inspiring motivation for a bilevel approach comes from the following observation and result. It is known that for OCPs with free final time the Hamilton function $\mathcal{H}$ vanishes at the final time along a solution [16, 7]. Also, as a necessary condition for local optimality, we expect the reduced cost function $\tilde{\mathcal{J}}$ to have zero derivative. In the present setting, the reduced cost function can be assumed to be sufficiently smooth for our purposes, thanks to the LTI dynamics and linear-quadratic cost functional. A link between Hamilton function $\mathcal{H}$ and the reduced cost function $\tilde{\mathcal{J}}$ is established by Theorem 1 and exploited to solve the upper level Problem 2.

Theorem 1. *Consider the free finite time-horizon Problem 1 along with assumptions (A1)–(A4) and the reduced cost function $\tilde{\mathcal{J}}$ as defined in Problem 2. Let (x_T, u_T) be the unique solution to Problem 3 given final time T, and λ_T the associated multiplier. Then*

$$\tilde{\mathcal{J}}'(T) = \mathcal{H}\left(x_T(t), u_T(t), \lambda_T(t)\right) \tag{15}$$

for any $t \in [0, T]$ and $T \in \Gamma$.

Proof. Define a reduced cost function $\tilde{\mathcal{J}} : \Gamma \to \mathbb{R}$ through the solution of state and control for a given final time, such that function $\tilde{\mathcal{J}}$ associates to any feasible value assigned to the final time the corresponding optimal cost in the sense of Problem 1. For any $T \in \Gamma$, the reduced cost is formally given by

$$\tilde{\mathcal{J}}(T) := \mathcal{J}\left(\tilde{x}(\cdot,T), \tilde{u}(\cdot,T), T\right) \ , \tag{16}$$

where, for any given $T \in \Gamma$, functions $\tilde{x}(\cdot,T) \in X(T)$ and $\tilde{u}(\cdot,T) \in U(T)$ solve Problem 3 with given final time T. Let function $\tilde{\lambda}(\cdot,T) \in W^{1,\infty}([0,T],\mathbb{R}^n)$ and vectors $\tilde{\mu}(T) \in \mathbb{R}^n$ and $\tilde{\eta}(T) \in \mathbb{R}^l$ be the multipliers associated to the solution $(\tilde{x}(\cdot,T), \tilde{u}(\cdot,T))$. Then, Equations (1), (2), (6), (8), (9) are necessary optimality conditions (NOCs) [16, 7] and are satisfied by aforementioned functions, for any given $T \in \Gamma$, by definition. Let us define the auxiliary cost function $\tilde{\gamma} : \Gamma \to \mathbb{R}$ augmenting the reduced cost with constraints, namely as

$$\begin{aligned}\tilde{\gamma}\left(T\right) := \tilde{\mathcal{J}}(T) + \tilde{\mu}(T)^\top \left(x_0 - \tilde{x}(0,T)\right) + \tilde{\eta}(T)^\top \left(y_\mathrm{T} - C_\mathrm{T}\tilde{x}(T,T)\right) + \\ + \int_0^T \tilde{\lambda}(\tau,T)^\top \left[A\tilde{x}(\tau,T) + B\tilde{u}(\tau,T) - \tilde{x}'_t(\tau,T)\right] \mathrm{d}\tau\end{aligned} \tag{17}$$

for any $T \in \Gamma$. From (17), substitution of cost functional (4) and Hamilton function (5) and integration by parts yield

$$\begin{aligned}\tilde{\gamma}(T) = \frac{1}{2}\left(\tilde{x}(T,T) - \xi_\mathrm{T}\right)^\top Q_\mathrm{T}\left(\tilde{x}(T,T) - \xi_\mathrm{T}\right) + \\ + \tilde{\mu}(T)^\top \left(x_0 - \tilde{x}(0,T)\right) + \\ + \tilde{\eta}(T)^\top \left(y_\mathrm{T} - C_\mathrm{T}\tilde{x}(T,T)\right) + \\ + \int_0^T \left[\mathcal{H}[\tau,T] + \tilde{\lambda}'_t(\tau,T)^\top \tilde{x}(\tau,T)\right] \mathrm{d}\tau + \\ - \left[\tilde{\lambda}(\cdot,T)^\top \tilde{x}(\cdot,T)\right]_0^T\end{aligned} \tag{18}$$

with $[t,T] = \left(\tilde{x}(t,T), \tilde{u}(t,T), \tilde{\lambda}(t,T)\right)$ for any t and T. Notice that auxiliary function $\tilde{\gamma}$, as defined in (17), is identical to $\tilde{\mathcal{J}}$ because $\tilde{x}(\cdot,T)$ and $\tilde{u}(\cdot,T)$ are by definition solutions to Problem 3 for a given T and thus satisfy constraints therein, namely NOCs. Then, it holds $\tilde{\mathcal{J}}(T) = \tilde{\gamma}(T)$ for any $T \in \Gamma$ and, furthermore, it is also $\tilde{\mathcal{J}}'(T) = \tilde{\gamma}'(T)$, for any $T \in \Gamma$. Hence, formal differentiation of $\tilde{\gamma}$ at $T \in \Gamma$ from (18) yields the derivative $\tilde{\mathcal{J}}'(T)$. After substitution of NOCs (1), (2), (6), (8) and (9), Hamilton function (5) and some rearrangements, it reads:

$$\tilde{\gamma}'(T) = \mathcal{H}[T,T] - \left[\tilde{x}(\cdot,T)^\top \tilde{\lambda}'_T(\cdot,T)\right]_0^T + \int_0^T \left[\tilde{x}'_t(\tau,T)^\top \tilde{\lambda}'_T(\tau,T) + \tilde{x}(\tau,T)^\top \tilde{\lambda}''_{tT}(\tau,T)\right] \mathrm{d}\tau. \tag{19}$$

Since the partial second derivatives of $\tilde{\lambda}$ are continuous in a neighborhood of (τ,T), given $T \in \Gamma$, owing to Schwarz's theorem, it holds $\tilde{\lambda}''_{tT}(\tau,T) = \tilde{\lambda}''_{Tt}(\tau,T)$ for any $\tau \in [0,T]$.

Hence, substituting the last term, thanks to the fundamental theorem of calculus, Eq. (19) becomes

$$\begin{aligned}\tilde{\gamma}'(T) &= \mathcal{H}[T,T] - \Big[\tilde{x}(\cdot,T)^\top \tilde{\lambda}'_T(\cdot,T)\Big]_0^T + \int_0^T \frac{\partial}{\partial t}\Big[\tilde{x}(\tau,T)^\top \tilde{\lambda}'_T(\tau,T)\Big]\,\mathrm{d}\tau \\ &= \mathcal{H}[T,T]\ . \end{aligned} \tag{20}$$

The result in (15) directly follows from Eq. (20), the equivalence of $\tilde{\mathcal{J}}$ and $\tilde{\gamma}$ along solutions and noticing that, for autonomous dynamical systems, the Hamilton function attains a constant value along solutions [7, Thm. 7.1.6]. □

Remark 7. *Under mild assumptions on existence and uniqueness of solutions, Theorem 1 can be easily extended to a larger class of OCPs with nonlinear dynamics, cost and coupled boundary conditions.*

Remark 8. *The assumptions on existence, uniqueness and differentiability of these operators result from the implicit function theorem. This requires the mappings to be locally invertible, i.e. surjective and homeomorphisms. Intuitively, these conditions hold if and only if the final time influences the optimal control and state trajectory. We argue that meaningful free final time OCPs are those in which the final time affects the solution.*

Remark 9. *From the upper level perspective, as stated in Problem 2, the optimization problem consists of finding a minimizer $T^\star$ of $\tilde{\mathcal{J}}$. Assumption (A3) and continuous differentiability of the reduced cost function $\tilde{\mathcal{J}}$ at $T^\star$ make condition $\tilde{\mathcal{J}}'(T^\star) = 0$ necessary for (local) optimality. This condition can formally be adopted in order to find a solution to Problem 2. If a time transformation technique is applied, the evaluation of $\tilde{\mathcal{J}}'$, if needed, might be quite involved. Instead, by exploiting Theorem 1 within a bilevel approach, this evaluation turns out to be straightforward.*

2.5 Free time LQR: the upper level problem

For an autonomous-system fixed-final-time problem the Hamilton function (5) is constant with respect to time along a solution [7, Thm. 7.1.6]. Furthermore, for a free-final-time problem the Hamilton function (5) vanishes almost everywhere in time along a solution [7, Thm. 7.1.8]. Thus, considering the original free-final-time Problem 1, being a linear time-invariant system also autonomous, the following condition must hold along a solution:

$$\mathcal{H}\left(x^\star(t), u^\star(t), \lambda^\star(t)\right) = 0 \tag{21}$$

for a.e. $t \in [0, T^\star]$. At this point, the result established by Theorem 1 can be better appreciated with a broader view on the problem. After noticing that along a solution the optimal control is equivalent to the optimal feedback (7), i.e. $u^\star = u_{\mathrm{fb}}(x^\star, \lambda^\star)$, let us introduce the function $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, defined by $h\,(x, \lambda) := \mathcal{H}\,(x, u_{\mathrm{fb}}(x,\lambda), \lambda)$, for any

$x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^n$. Substituting (5), (7), (11) and rearranging terms yields

$$\begin{aligned} h(x,\lambda) &= w + \frac{1}{2}\begin{pmatrix} x \\ u_{\text{fb}}(x,\lambda) \end{pmatrix}^\top \Pi \begin{pmatrix} x \\ u_{\text{fb}}(x,\lambda) \end{pmatrix} + \\ &\quad + \lambda^\top \begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} x \\ u_{\text{fb}}(x,\lambda) \end{pmatrix} \\ &= w + \frac{1}{2}\begin{pmatrix} x \\ \lambda \end{pmatrix}^\top \underbrace{\begin{bmatrix} -M_{\lambda x} & -M_{\lambda\lambda} \\ M_{xx} & M_{x\lambda} \end{bmatrix}}_{=:W} \begin{pmatrix} x \\ \lambda \end{pmatrix} \end{aligned} \tag{22}$$

where $W = W^\top$, being $M_{x\lambda}$ and $M_{\lambda x}$ symmetric and $M_{xx} = -M_{\lambda\lambda}^\top$; see (11). From (21) and the definition of h, it follows that $h(x^\star(t), \lambda^\star(t)) = 0$ must hold for a.e. $t \in [0, T^\star]$. In particular, this necessary condition must hold at the initial time. Thus, an optimal final time $T^\star$ satisfies

$$h\left(x_0, \hat{\lambda}\left(T^\star, x_0\right)\right) = 0 \tag{23}$$

being (necessarily) $x^\star(0) = x_0$ and $\lambda^\star(0) = \hat{\lambda}\left(T^\star, x_0\right)$, as discussed in Section 2.3. We emphasize that condition (23) is necessary for optimality of a solution, but in general not sufficient. In fact, similarly to (21), it is satisfied by any final time corresponding to a stationary point of the reduced cost function $\tilde{\mathcal{J}}$; see Problem 2.

Given an initial state x_0 for Problem 1, let us define an auxiliary function $\tilde{h} : \mathbb{R} \to \mathbb{R}$, based on h, such that $\tilde{h}(T) := h\left(x_0, \hat{\lambda}(T, x_0)\right)$, for any $T \in \Gamma$. Then, necessary condition (23) can be rewritten as $\tilde{h}(T^\star) = 0$. Using the notation from Theorem 1, the equation $\tilde{h}(T) = \mathcal{H}[0, T]$ holds true, by definition, for any $T \in \Gamma$. We should remark now that, from the discussion in Section 2.3, evaluating $\hat{\lambda}(T, x_0)$ can be interpreted as solving Problem 3 with given final time T. Then, owing to the definitions, a root of $\tilde{h}$ is a root of the map $T \mapsto \mathcal{H}\left(x_0, u_{\text{fb}}(x_0, \hat{\lambda}(T, x_0)), \hat{\lambda}(T, x_0)\right)$ too. Furthermore, this is also a root of the map $T \mapsto \tilde{\mathcal{J}}'(T)$, as shown in Theorem 1.

In the following, we will also consider a function $\hat{T} : \mathbb{R}^n \to \Gamma$, such that, for any given x°, an optimal final time for Problem 1 with initial state x° is $T^\circ := \hat{T}(x^\circ)$. Function $\hat{T}$ can formally be evaluated but, in practice, it corresponds to solving a global optimization problem, whose solution is not straightforward (nor unique) in general. Nonetheless, an effective way of evaluating $\hat{T}$ at x° is possible when an (accurate enough) approximation of T° is available. We propose to exploit the necessary condition $\tilde{h}(T^\star) = 0$ in order to iteratively improve the estimation of $T^\star := \hat{T}(x_0)$. This root-finding approach is further discussed in the next part.

Remark 10. *The lack of sufficiency for condition* (23) *mirrors the presence of multiple stationary points for $\tilde{\mathcal{J}}$, in general. An optimal final time $T^\star$ for Problem 1 is a global minimizer of $\tilde{\mathcal{J}}$, and then global optimization techniques should be adopted to avoid local minima. We highlight that, in contrast to Problem 3, uniqueness of a solution to Problem 1 cannot be guaranteed, in general.*

Remark 11. *Given an initial state x_0 and the corresponding optimal final time $T^\star := \hat{T}(x_0)$, a sufficient condition for (local) optimality is given by the positivity of the second derivative of the reduced cost function, namely $\tilde{\mathcal{J}}''(T^\star) > 0$. By exploiting aforementioned definitions and Theorem 1, this sufficient condition can be rewritten as $\tilde{h}'(T^\star) > 0$.*

2.5.1 Final time refinement

Given an initial state x_0 and a guess $T_{[0]}$ for an optimal final time $T^\star := \hat{T}(x_0)$, we seek a numerical procedure capable of iteratively refining an estimate of $T^\star$, while taking advantage of the necessary condition $\tilde{h}(T^\star) = 0$. To this end, root-finding algorithms, e.g. gradient-based Householder's methods, can be adopted. Newton's and Halley's method belong to the just mentioned class of methods, and they are algorithms of order 1 and 2, with rate of convergence of 2 and 3, respectively. Halley's method, see [1], is suitable for scalar equations only and for $k = 0, 1, 2, \dots$ it generates iterates

$$T_{[k+1]} = T_{[k]} - \frac{\tilde{h}\left(T_{[k]}\right)}{\tilde{h}'\left(T_{[k]}\right) - \frac{1}{2}\tilde{h}''\left(T_{[k]}\right)\frac{\tilde{h}\left(T_{[k]}\right)}{\tilde{h}'\left(T_{[k]}\right)}}. \tag{24}$$

If the initial guess $T_{[0]}$ lies in the basin of attraction of $T^\star$ for a given method, then that method will return a sequence of estimates $\{T_{[k]}\}_{k\in\mathbb{N}}$ eventually converging to $T^\star$, i.e. such that $\lim_{k\to\infty} T_{[k]} = T^\star$. Recalling definitions given above, derivatives of the function $\tilde{h}$ can be evaluated and exploited in order to achieve high convergence rates and improved robustness. At the k-th iteration, the first two derivatives satisfy

$$\tilde{h}'\left(T_{[k]}\right) = h'_\lambda\left(x_0, \hat{\lambda}(\cdot)\right)\hat{\lambda}'_T(\cdot) \tag{25}$$

$$\begin{aligned}\tilde{h}''\left(T_{[k]}\right) = {} & h'_\lambda\left(x_0, \hat{\lambda}(\cdot)\right)\hat{\lambda}''_{TT}(\cdot) + \\ & + \hat{\lambda}'_T(\cdot)^\top h''_{\lambda\lambda}\left(x_0, \hat{\lambda}(\cdot)\right)\hat{\lambda}'_T(\cdot)\end{aligned} \tag{26}$$

where $(\cdot) := \left(T_{[k]}, x_0\right)$ for brevity. Expressions for h'_λ, $h''_{\lambda\lambda}$ and $\hat{\lambda}'_T$, $\hat{\lambda}''_{TT}$ are obtained from (22) and (14), respectively; see Appendix A. The refinement procedure is sketched in Algorithm 2.

Remark 12. *It is noticeable that every refinement iteration requires the solution of an instance of Problem 3. As discussed in Section 2.3, its solution exists and is unique for any given final time. Thus, in a sense, the lower level is nothing but a way to compute the cost function for any given final time, and the bilevel optimization problem is solved through an equivalent single-level problem.*

Remark 13. *Assumption (A4) is needed because of the root-finding approach for solving the upper level problem. Indeed, as discussed in Remark 11, it is $\tilde{\mathcal{J}}''(T) = \tilde{h}'(T)$ for any $T \in \Gamma$. In general, more sophisticated methods are needed to find a root $T^\star$ of $\tilde{h}$ when it also holds $\tilde{h}'(T^\star) = 0$.*

Algorithm 2 Pseudocode for refinement procedure.

```
Input: $T_{[0]}$, $x_0$, $w$, $M$, $\Gamma$, $\delta_h$, $\delta_{\Delta T}$
Output: $T$
  for $k = 0, 1, \dots$ do
      $\hat{\lambda}, \hat{\lambda}'_T, \hat{\lambda}''_{TT} \leftarrow T_{[k]}, x_0, M$                          ▷ Algorithm 1
      $\tilde{h}, \tilde{h}', \tilde{h}'' \leftarrow \hat{\lambda}, \hat{\lambda}'_T, \hat{\lambda}''_{TT}, x_0, w, M$          ▷ Eq. 22, 25, 26
      $\Delta T \leftarrow \tilde{h}, \tilde{h}', \tilde{h}''$                                      ▷ Eq. 24
      $T_{[k+1]} = \mathcal{P}_\Gamma\left(T_{[k]} + \Delta T\right)$                     ▷ projected update
      if $|\tilde{h}| < \delta_h$ or $|T_{[k+1]} - T_{[k]}| < \delta_{\Delta T}$ then
          break
      end if
  end for
  return $T = T_{[k+1]}$
```

The second derivative of $\tilde{h}$ can be exploited to speed up the final time refinement (and enlarge the convergence basin) through a higher-order root-finding Householder's method; hence, recalling (26), one has to compute $\hat{\lambda}''_{TT}$ with, e.g., Algorithm 1. In the case Γ is a compact set, a projection step, denoted $\mathcal{P}_\Gamma$, might affect the Halley's update in order to obtain a feasible final time, see Algorithm 2. In Appendix B we derive necessary complementarity conditions for optimality of the free final time and develop a simple yet effective projection scheme to deal with them.

3 Free time model predictive control and exploitation of the bilevel problem

This section discusses the implementation of free-time LQR within an MPC loop, that is the main application motivating this work.

Please note that a standard implementation of MPC would require a nonlinear programming (NLP) solver for the nonlinear and nonconvex free-final time subproblems in each MPC step. In contrast, the method developed in Section 2 does not require an NLP solver but merely a method for finding a root of a scalar equation involving the Hamilton function. With Halley's method a very fast third order method is available for this task. Note further, that the lower level LQR problems, which implictly enter in the root finding problem, can be solved exactly for a given final time.

Similarly to Section 2.1, let a continuous-time linear time-invariant system be given and described by matrices A, B and C_T and let vectors x_0 and y_T be the initial state and a desired final measurement, respectively. Also, let Π and Q_T be symmetric cost matrices and $w > 0$ a positive time cost. Let t denote time and $\{t_k\}_{k=0}^N$ a time grid, from initial time $t_0 = 0$ to a certain final time t_N, with $\Delta t_k := t_k - t_{k-1} > 0$ for $k = 1, \dots, N$. At each and every time point t_k, $k = 0, \dots, N$, a control command u_k has to be sent from the controller to the plant; control action u_k is applied during the time interval $[t_k, t_{k+1})$.

Meanwhile, an estimate of actual system state is given to the controller at some time points, possibly on a different time grid $\{t_k^x\}_{k=0}^N$, with $t_k^x \in (t_{k-1}, t_k]$ for $k = 1, \dots, N$ and, for simplicity, $t_0^x = t_0$. At time t_k^x, state estimate σ_k^x is made available to the controller, e.g. through a full-state observer, and represent a possibly noisy/perturbed estimate of the internal state of the system under control. Within the MPC framework, the control sequence $\{u_k\}_{k=0}^N$ is built as follows:

(0) set $k \leftarrow 0$;

(1) estimate state $\sigma_k := \sigma(t_k)$;

(2) solve OCP, i.e. Problem 1, from time t_k and initial state σ_k; denote $u_{k\to N} : [t_k, t_N) \to \mathbb{R}^m$ the resulting optimal control;

(3) send command $u_k := u_{k\to N}(t_k)$ to the plant;

(4) if $k < N$, set $k \leftarrow k + 1$ and repeat (1)–(4), otherwise quit;

There are many issues in this procedure when applied in this straightforward manner; for instance, the piecewise constant nature of the control action in time should be accounted for in the formulation of the OCP. Another subtle issue is due to finiteness of computational power (that cannot be overcome) and timing. Specifically, the control command u_k is indeed optimal in the sense of OCP if applied when the system state is exactly σ_k. Assuming the controller receives an error-free state estimate, estimate σ_k of system state at time t_k should be available to the controller before time t_k is eventually reached, and this anticipation should compensate for the time needed to execute step (2). On the other hand, predicting the future system state might introduce errors, e.g. due to modeling approximations; thus, the time horizon for prediction should be kept at a minimum. From these considerations, we conclude that the execution time for solving the OCP at step (2) should be as short as possible. Furthermore, we argue there exists a trade-off between time effort and solution accuracy, in the sense that, up to a certain degree, a rougher approximation of the solution might be compensated for by a shorter computational time, while still resulting in an effective control loop.

As a contribution of this work, we propose a three-steps procedure developed to face the aforementioned timing issue arising in online control tasks. As mentioned in the introduction, Section 1, this strategy is inspired by and shares some features with predictor-corrector methods from numerical analysis, sensitivity analysis [7] and real-time iteration scheme with initial value embedding [5]. In particular, considering the k-th time slot, after time t_{k-1} but before the next state estimate is available, at time t_k^x, both a *refinement* and a *prediction* phase take place. The latter generates estimates of future system states, specifically at time t_k^x and t_k, and corresponding sensitivities, based on quantities polished/made more accurate during the former refinement phase. Then, after estimate σ_k^x is read, a *correction* phase updates previous estimates based on discrepancy between those and the last measurement, through sensitivities. Finally, the control input can be calculated from these corrected quantities and commanded to the plant (hopefully, not after time t_k).

At the initial time t_0, the procedure requires an *initialization* phase, whose aim is to compute the optimal final time T_0 given initial condition σ_0 at t_0. Then, the optimal control command u_0 is given by the solution to Problem 3 with final time T_0. Hence, in practice, this initial step consists of solving Problem 1 without any good guess for the final time. We claim that, regarding this procedure, a method seeking global optimality should be preferred to one relying only on necessary optimality conditions. In fact, an inaccurate approximation of a global minimum might be more beneficial and effective compared to a precisely identified local minimum. This crucial observation stems from the local convergence properties of the developed method; see Remarks in Section 2.2 and 2.5.

The refinement phase has been discussed at the end of Section 2.5. In the following Sections 3.1–3.3 the main ingredients for prediction and correction steps are developed and analyzed. Finally, the overall procedure is summarized in Algorithm 3.

3.1 Final time prediction

Let x_k and T_k be system state and optimal final time, respectively, at a certain time t_k; final time T_k is meant to be optimal in the sense of Problem 1 with initial state x_k. In this Section we construct and discuss a way to predict future values of the optimal final time. In practice, given a future time point t_{k+1}, $t_{k+1} \geq t_k$, we aim at generating an *a priori* estimate of T_{k+1}; this, in principle, requires solving Problem 1 with initial state x_{k+1}.

Assuming that the mathematical model is an exact representation of the system behavior, we can proceed with the following reasoning, based on both intuition and Bellman's optimality principle: if the actual system evolution follows exactly the predicted one, the final time will evolve accordingly, and thus it will reduce of the same amount elapsed between two time points. From this observation, one could draw as a conclusion that

$$T_{k+1} = T_k - (t_{k+1} - t_k). \tag{27}$$

In order to substantiate this statement, let us consider the continuous-time system whose evolution satisfies the following initial value problem (IVP):

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ u(t) = u_{\text{fb}}(x(t), \lambda(t)) \\ \lambda(t) = \hat{\lambda}(T(t), x(t)) \\ T(t) = \hat{T}(x(t)) \\ x(0) = x_0 \end{cases} , \quad t \geq 0 \tag{28}$$

This system describes the closed-loop behavior of the linear system (1), with boundary conditions (2), when optimally-controlled through FLQR in an MPC scheme, i.e. by solving a different instance of Problem 1 at every time, in continuous-time, starting from initial state x_0. In (28), functions $\hat{T}$, $\hat{\lambda}$ and u_{fb} formally condense the solution to Problem 1. It is noticeable that, in this closed-loop model, the optimal final time is represented by a time-varying function. Let x_{CL}, u_{CL}, λ_{CL}, T_{CL} denote the state, control, costate and final time trajectory, respectively, solving IVP (28). This closed-loop state evolution x_{CL} should match the one predicted at the initial time by solving Problem 1, which is the best achievable, by definition, neglecting disturbances, being it a solution to FLQR starting from initial state x_0. As seen in Sections 2.3 and 2.5, the open-loop evolution can be interpreted as the solution to the following IVP:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ u(t) = u_{\mathrm{fb}}(x(t), \lambda(t)) \\ \dot{\lambda}(t) = -A^\top \lambda(t) - Qx(t) - S^\top u(t) \\ x(0) = x_0 \\ \lambda(0) = \hat{\lambda}(\hat{T}(x_0), x_0)) \end{cases} \quad , \ t \geq 0 \tag{29}$$

For consistency, let x_{OL}, u_{OL}, λ_{OL} denote the state, control and costate trajectory, respectively, from the open-loop dynamics in IVP (29), with optimal final time $T_{\mathrm{OL}} := \hat{T}(x_0)$. Aiming at the best performance, we would like the closed-loop evolution to match the open-loop one, i.e. to have $x_{\mathrm{OL}}(t) = x_{\mathrm{CL}}(t)$ for any $t \geq 0$ (at least $t \in [0, T_{\mathrm{OL}}]$), for any instance of Problem 1, i.e. for any cost functional, dynamics and boundary conditions. The following result establishes a link between this matching and a characterization useful for predicting the optimal final time.

Theorem 2. *Consider solutions to IVPs (28) and (29). Then, condition $x_{CL}(t) = x_{OL}(t)$ holds true for any $t \geq 0$, for any arbitrary boundary conditions, dynamics and cost functional, if and only if $\dot{T}_{CL}(t) = -1$ for any $t \geq 0$.*

The proof of Theorem 2 is reported in Appendix A. With this observation, finally, one can realize that (27) expresses an exact discrete-time version of the optimal final time time-derivative. □

Remark 14. *Optimal final time prediction, as discussed here, returns an* a priori *estimate by using information available up to the actual time. Then, disturbances and inaccuracies cannot be accounted for during this phase, because they are still unknown, but their effect should be compensated for through the correction step.*

Remark 15. *Considering a continuous-time closed-loop system might be an approximation of the underlying discrete-time nature of the controller, but it is adopted in order to easily obtain a prediction of future optimal control. Typically (digitally implemented) controllers apply piecewise constant-in-time control inputs, so continuous-time MPC as in (28) does not represent the closed-loop dynamics. However, the approximation error likely becomes negligible as sampling time/control frequency gets short/high enough. It should be highlighted also that this continuous-time approximation is commonly accepted, and most of the times it is present from the very beginning, see for instance Problem 1. These interesting modeling issues might heavily affect the analytical and computational aspects in simulation and optimization, but they are beyond the scope of this work and will not be further discussed here.*

3.2 State prediction

As mentioned at the beginning of Section 3, in order to minimize the delay between state measurement and control command, we aim at anticipating as much computation as possible. Then, we seek a prediction of both, the measurement itself and the state when control signal will be eventually sent.

Let us consider the linear dynamics (1) with a constant control input, say u_a, for time $t \geq t_a$, and given initial state x_a, namely

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu_a \\ x(t_a) = x_a \end{cases}, \quad t \geq t_a \tag{30}$$

Given time t_b, $t_b \geq t_a$, we are interested in $x_b := x(t_b)$ and $S_{x_a \to x_b} := \nabla_{x_a} x_b$. From LTI systems theory [10], we have that the solution to IVP (30) for $t \geq t_a$ is given by $x(t) = e^{A(t-t_a)} x_a + \int_{t_a}^{t} e^{A(t-\tau)} B u_a \, \mathrm{d}\tau$ and thus we can get x_b just by evaluating it at time t_b. It is also clear that the sensitivity matrix $S_{x_a \to x_b}$, i.e. the sensitivity of x_b with respect to x_a, is given by the matrix exponential, namely $S_{x_a \to x_b} = e^{A(t_b - t_a)}$. By substituting indexes a and b with those corresponding to previous control command, say time t_{k-1}, and next state measurement, at t_k^x, respectively, we can get an estimate for the state measurement x_k^x. Then, shifting ahead and substituting indexes a and b with those corresponding to state measurement, t_k^x, and next control command, t_k, respectively, we can get an estimate for state x_k at time t_k and sensitivity $S_{x_k^x \to x_k}$. These two quantities allow to update the prediction of system state x_k once the state measurement is available at t_k^x. The role of this sensitivity update, or predictor-corrector, strategy becomes clearer within Algorithm 3.

Remark 16. *Notice that x_b can also be computed by adopting standard integrators, e.g. the Runge-Kutta methods, instead of relying on the matrix exponential. Similarly, matrix $S_{x_a \to x_b}$ can be obtained through sensitivity analysis [7].*

Remark 17. *It is possible to predict/estimate state, and corresponding sensitivity, also considering a continuous-time MPC loop, as in Section 3.1. The reason why here we model system dynamics with piecewise constant-in-time control input is twofold: it is likely more realistic, as discussed in Remark 15, and leads to less involved expressions and computations.*

Remark 18. *In Section 3.1 the optimal final time is forecasted by considering a continuous-time MPC system as a model. Here, instead, the future state is predicted by means of a system with piecewise constant-in-time control input. Let us denote T_k and x_k the optimal final time and state estimates, respectively, corresponding to time t_k. It is a subtle but crucial point to understand that $T_k \neq \hat{T}(x_k)$ in general, because of the two different adopted models. This means that, even in a disturbance-free environment, final time estimate T_k has to be corrected in order to satisfy the necessary condition (23). The need for an* a posteriori *update is even more clear in view of possible noise and additional modeling inaccuracies. The basis for this correction are discussed in the next Section 3.3.*

3.3 Final time sensitivity

The idea is to estimate the sensitivity of the optimal final time $T^\star$ with respect to the initial state x_0 and then to update/correct the final time prediction, Section 3.1, based on the actual state estimate obtained through measurement and sensitivity update, Section 3.2. This takes inspiration from real-time iteration scheme, in particular from the initial value embedding [5]. More precisely, we seek the gradient of function $\hat{T}$ defined in Section 2.5. Owing to its definition, it turns out that necessary condition (23) is satisfied, in the form (31), for any $x \in \mathbb{R}^n$:

$$h\left(x, \hat{\lambda}\left(\hat{T}(x), x\right)\right) = 0. \tag{31}$$

Equation (31) implicitly defines (at least locally) the function $\hat{T}$ and hence, through the implicit function theorem, it is possible to find its gradient.

Let $x_0 \in \mathbb{R}^n$ be a given (predicted) initial state and $T^\star := \hat{T}(x_0)$, $\lambda^\circ := \hat{\lambda}(T^\star, x_0)$ the corresponding optimal final time and initial costate, respectively. By differentiating the implicit constraint (31) with respect to x at x_0 and recalling (25), one gets

$$\begin{aligned} 0 &= h'_x(x_0, \lambda^\circ) + h'_\lambda(x_0, \lambda^\circ)\left[\hat{\lambda}'_T(T^\star, x_0)\hat{T}'_x(x_0) + \hat{\lambda}'_x(T^\star, x_0)\right] \\ &= \tilde{h}'\left(T^\star\right)\hat{T}'_x(x_0) + \frac{\mathrm{d}h}{\mathrm{d}x}\left(x_0, \hat{\lambda}\left(T^\star, x_0\right)\right) \end{aligned}$$

and then

$$\tilde{h}'\left(T^\star\right)\hat{T}'_x(x_0) = -\frac{\mathrm{d}h}{\mathrm{d}x}\left(x_0, \hat{\lambda}\left(T^\star, x_0\right)\right). \tag{32}$$

Given prediction x_0, let us denote $S_{x\to T} := \hat{T}'_x(x_0)$ the final time sensitivity with respect to the initial state. Then, for any updated initial state x, hopefully in a neighbourhood of x_0, we correct the final time prediction by assuming a first-order Taylor approximation of $\hat{T}$ around x_0. Thus, the corrected final time T is chosen to be $T := T^\star + S_{x\to T}\,(x - x_0)$, as implemented in Algorithm 3.

Remark 19. *In order to evaluate the gradient $\hat{T}'_x$ at x_0 from* (31)*, the implicit function theorem requires $\tilde{h}'\left(T^\star\right) \neq 0$; compare* (32)*. This condition can be expressed also in terms of the reduced cost function, namely $\tilde{\mathcal{J}}''\left(T^\star\right) \neq 0$; see Section* 2.5 *and Theorem* 1*. As highlighted in Remark* 4*, though, Assumption (A4) allows only those cases in which that necessary condition holds and hence the implicit function theorem applies.*

3.4 Algorithm

The Predictor-Corrector-Refiner (PCR) procedure is reported in Algorithm 3. Therein, the control input is commanded before executing any refinement step. This reduces the time between the state measurement and the control command. Nevertheless, depeding on the time constraints, one could perform some degree of refinement, devoting more time but likely improving the optimal control estimate. As an initialization procedure, one can perform a refinement step, namely execute Algorithm 2, given the initial state and a guess for the optimal final time. Also, one can check the goodness of the initial guess by inspecting the sufficient optimality conditions, see Remark 11.

Algorithm 3 Pseudocode for PCR algorithm

```
Input: $T^k, x^k, \lambda^k, u^k, \tau^k, \tau_\sigma^{k+1}, \tau^{k+1}, \Gamma^{k+1}$
Output: $T^{k+1}, x^{k+1}, \lambda^{k+1}, u^{k+1}$
    PREDICTION
    $T^{k+1} = \left[T^k + \tau^k - \tau^{k+1}\right]_{\Gamma^{k+1}}$    $\triangleright$ final time
    $x_\sigma^{k+1}, x^{k+1} \leftarrow x^k, u^k, \tau^k, \tau_\sigma^{k+1}, \tau^{k+1}$    $\triangleright$ state
    $S_{x_\sigma \to x} \leftarrow \tau_\sigma^{k+1}, \tau^{k+1}$    $\triangleright$ state sensitivity
    $S_{x \to T} \leftarrow T^{k+1}, x^{k+1}$    $\triangleright$ time sensitivity
    read input $\sigma^{k+1}$    $\triangleright$ measured state
    CORRECTION
    $\Delta x = S_{x_\sigma \to x}\left(\sigma^{k+1} - x_\sigma^{k+1}\right)$
    $x^{k+1} \leftarrow x^{k+1} + \Delta x$    $\triangleright$ initial state
    $T^{k+1} \leftarrow \left[T^{k+1} + S_{x \to T}\Delta x\right]_{\Gamma^{k+1}}$    $\triangleright$ final time
    $\lambda^{k+1} = \hat{\lambda}\left(T^{k+1}, x^{k+1}\right)$    $\triangleright$ initial costate
    $u^{k+1} = u_{\mathrm{fb}}\left(x^{k+1}, \lambda^{k+1}\right)$    $\triangleright$ initial control
    write output $u^{k+1}$    $\triangleright$ commanded control
    REFINEMENT
    $T^{k+1} \leftarrow T^{k+1}, x^{k+1}, \Gamma^{k+1}$    $\triangleright$ final time
```

4 Numerical results

We consider an overhead crane as an example problem to test the effectiveness and robustness of the proposed approach and method, Fig. 1. This might also be a reasonable application for the free finite time-horizon LQR because, we argue, there might be a trade-off between minimizing required time, control effort and oscillations of the payload. For a comparison with a pure minimum time controller, see [4].

4.1 Linear overhead crane-trolley system

Let us consider a mathematical model of an overhead crane with fixed cable length $L > 0$ and let x, v, θ and ω denote the trolley's horizontal position, velocity, the load's angular position, and its angular velocity, respectively. Let $M \geq 0$ and $m > 0$ be trolley and load mass. A horizontal force applied onto the trolley is the input available to control the system dynamics. Also, gravity force, horizontal and angular linear viscous frictions are considered, with parameters $g > 0$, $c_x \geq 0$ and $c_\theta \geq 0$. By adopting, e.g., the Euler-Lagrange equation, one can obtain a nonlinear autonomous dynamical system with affine control input and state-dependent symmetric nonsingular mass matrix. In order to get a linear system, it suffices to assume small and slow oscillations of the load, namely $\cos\theta \approx 1$,

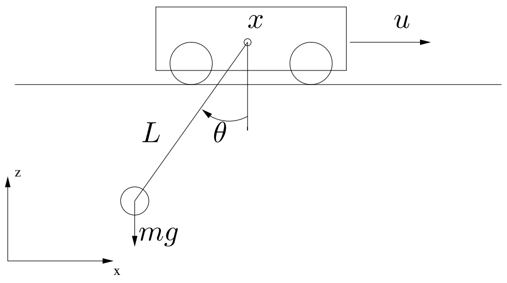

Figure 1: Schematic representation of the system under consideration.

Table 1: Model parameters.

$M = 10\,\mathrm{kg}$	$m = 1\,\mathrm{kg}$	$L = 1\,\mathrm{m}$
$c_x = 0.01\,\mathrm{Ns/m}$	$c_\theta = 0.01\,\mathrm{Ns/rad}$	$g = 9.81\,\mathrm{N/kg}$

$\sin\theta \approx \theta$ and $\omega^2 \sin\theta \approx 0$. Thus, it reads:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & M+m & 0 & mL \\ 0 & 0 & 1 & 0 \\ 0 & mL & 0 & mL^2 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -c_x & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -mgL & -c_\theta \end{bmatrix} \begin{pmatrix} x \\ v \\ \theta \\ \omega \end{pmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u \, .$$

Physical quantities with measurement units are expressed in terms of fundamental SI units, which are omitted. Model parameters are reported in Table 1. The control task consists of steering the system from an oscillating behaviour to rest in a different position. The initial state is set to be $x_0 = (1, 0, 0.1, -0.1)^\top$. A hard constraint is imposed on the final velocity, while terminal costs penalise the other states, through matrices $C_\mathrm{T} = [0, 1, 0, 0]$ and $Q_\mathrm{T} = \mathrm{diag}(84.4358, 0, 259.8704, 800.0685)$ respectively; all the states are subject to running costs. Other parameters related to the control problem are here reported: $w = 1$, $\xi_\mathrm{T} = 0$, $y_\mathrm{T} = 0$, $R = 0.1$, $S = [0, 0.9619, 0, 0]$, $Q = \mathrm{diag}(0.0046, 0.7749, 0.8173, 0.8687)$. Except for the control cost, cost matrices have been generated with uniformly distributed pseudo-random numbers. The sampling time is $T_s = 0.2\,\mathrm{s}$, whereas the optimal final time is about $3.7\,\mathrm{s}$, as found at the initial time, meaning that the control system is (on purpose) relatively slow compared to the system dynamics. State measurements are taken $\Delta T_\sigma = 0.05\,\mathrm{s}$ before the control input is to be commanded. Zero-mean Gaussian independent noise is adopted to simulate the measurement process, yielding $\sigma^k = x(t^k) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \Sigma)$ with covariance matrix $\Sigma = \mathrm{diag}(10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$. The tolerances for the refinement phase performed by Algorithm 2 are $\delta_h = 10^{-4}$ and $\delta_{\Delta T} = 10^{-4}$. Finally, as commonly adopted in the MPC setting, a piecewise constant control input is used, that is, the commanded control is kept constant until the next iteration updates its value. Instead, the underlying LQR considers and provides a continuous optimal control function. Introducing this approximation in the dynamics, the closed-loop control tasks is harder and more realistic.

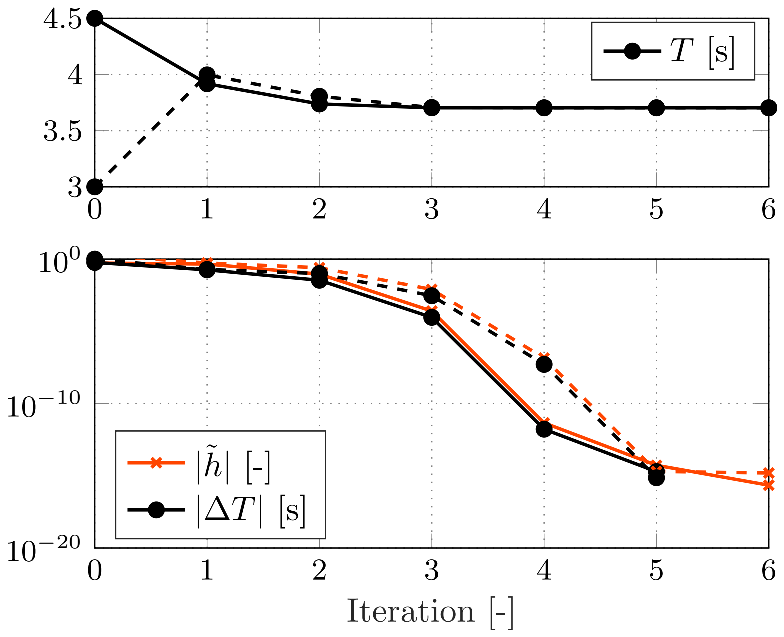

Figure 2: Refinement procedure: final time T (top), Hamilton function $\tilde{h}$ and time stepsize ΔT (bottom), starting from two guesses (solid and dashed); see Algorithm 2.

At the initial time, i.e. for the first MPC step, the optimal final time has to be estimated. To this end, the refinement procedure sketched in Algorithm 2 is executed, starting from a guided initial guess. In fact, the same routine is called at each and every MPC step, after the control input is commanded, see Section 3.4. Given an initial guess sufficiently close to the optimal point, the third-order root-finding method discussed in Section 2.5.1 ensures fast convergence, as depicted in Fig. 2 for two starting points. Therein, the stepsize $|\Delta T|$ and the residual $|\tilde{h}|$ reach the numerical precision after 5 iterations. For the sake of comparison with a standard approach, the following test was conducted. The free final time OCP was solved, through time transformation, by using the software package OCPID-DAE1 [8] (direct single shooting, 101 equidistant grid points, piecewise constant control parametrization), starting from a dummy initial guess of state and control. The solution matched the one obtained previously, with final time $T^\star = 3.703\,\mathrm{s}$. However, in order to have a fair comparison in the MPC context, a warm-starting must be considered. Hence, the latter numerical solution was stored and re-used as initial guess for the very same problem, just slightly changing the guess for the final time ($\pm 0.05\,\mathrm{s}$). Running OCPID-DAE1 with the modified initial guess resulted in the convergence to the same solution as before, but only after 12 iterations. Surely this result is not conclusive, but it suggests the merit of the proposed approach, which is tailored for the considered problem class.

The evolution of the system during MPC is depicted in Fig. 3, where also predicted and actual measurements are reported, along with the initially planned open-loop evolution.

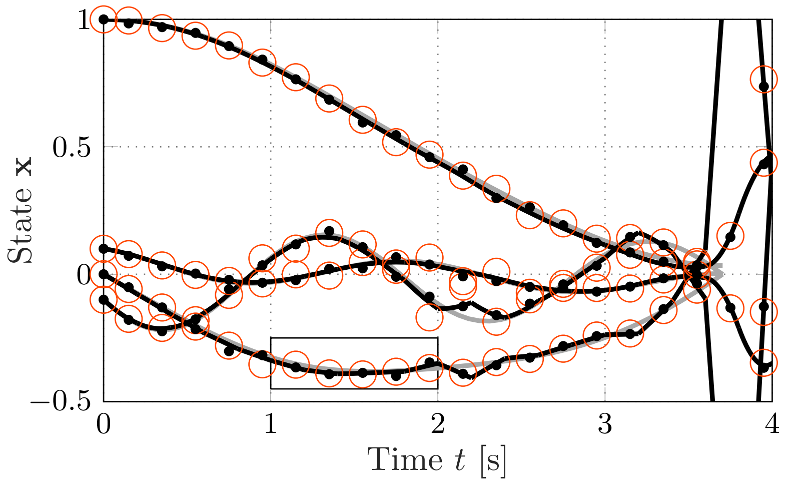

Figure 3: System state during MPC: state x (black solid line), open-loop evolution x_{OL} (grey solid line), predicted state measurements x_σ (red circle) and actual state measurements σ (black dot). The box is zoomed in and depicted in Fig. 4.

Fig. 4 shows a zoom from Fig. 3, where true, predicted, corrected and refined quantities are better displayed.

We point out that the presence of a hard terminal constraint makes it more difficult to steer the system toward the desired state, especially with noisy unfiltered measurements. In fact, at around $t = 3.6\,\mathrm{s}$, the state trajectory exhibits a jerky behaviour, see Fig. 3. We argue that, when close enough to the desired state, the optimal control becomes highly sensitive to disturbances; this explains the abrupt change in the state trajectory. Numerical tests suggest to substitute the hard constraint with a terminal penalty cost, in order to get a smoother evolution, when the distance from the desired state is comparable to the noise level. This issue is not further discussed within this paper.

The commanded control input u corresponds to corrected state and final time estimates, see Section 3.4. In fact, on purpose we have not allowed for refinement before the control input is commanded, but only the sensitivity update after the state estimate was available. Closed-loop control u_{CL} is computed with the same corrected state estimate but with the refined final time. Hence, the gap between these two curves in Fig. 5 highlights the effectiveness of the sensitivity update after the state measurement. It is noticeable that this gap is often much smaller than the discrepancy with predicted control input, see zoomed box in Fig. 6, proving the importance of the correction step. We point out also that closed-loop control trajectories approach the initially planned open-loop control input, Fig. 5. Nonetheless, some discrepancy arises after $t = 2\,\mathrm{s}$. We argue that the piecewise constant control approximation and the noisy readings steer the system away from the optimal state evolution. Hence, the actual state evolution is suboptimal but the (closed-loop) controlled system is still able to (approximately) reach the desired state, see Fig. 3.

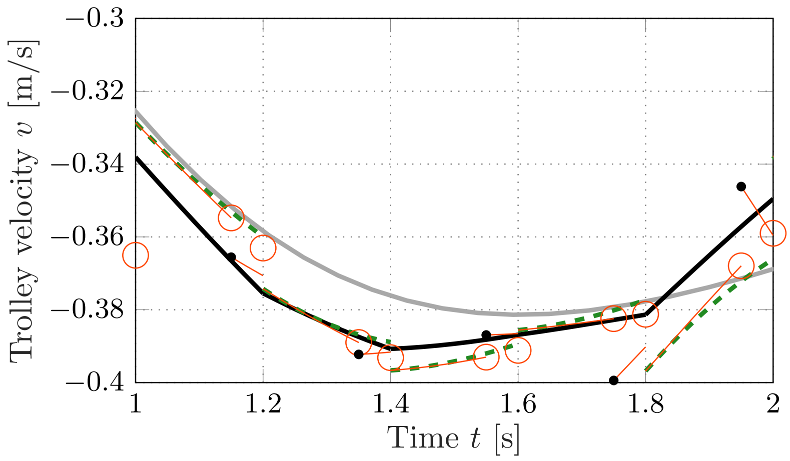

Figure 4: Trolley velocity zoomed in from Fig. 3: velocity v (black solid line), initially planned open-loop evolution v_{OL} (grey solid line), predicted measured velocity v_σ and next initial velocity (red circle), predicted evolution with piecewise constant control input (red solid thin line), measured velocity (black dot), closed-loop evolution v_{CL} from corrected state estimate and refined final time (green dashed line).

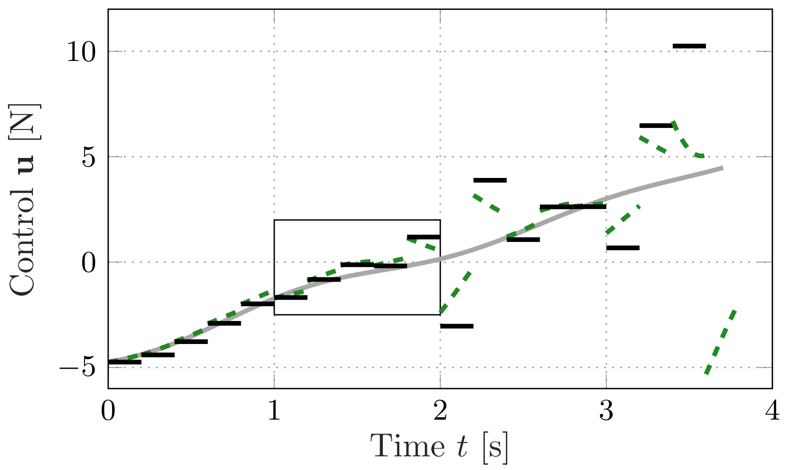

Figure 5: Control input during MPC: commanded control u (black solid line), initially planned open-loop control u_{OL} (grey solid line), closed-loop control u_{CL} given corrected state estimate and refined final time (green dashed line). The box is zoomed in and depicted in Fig. 6.

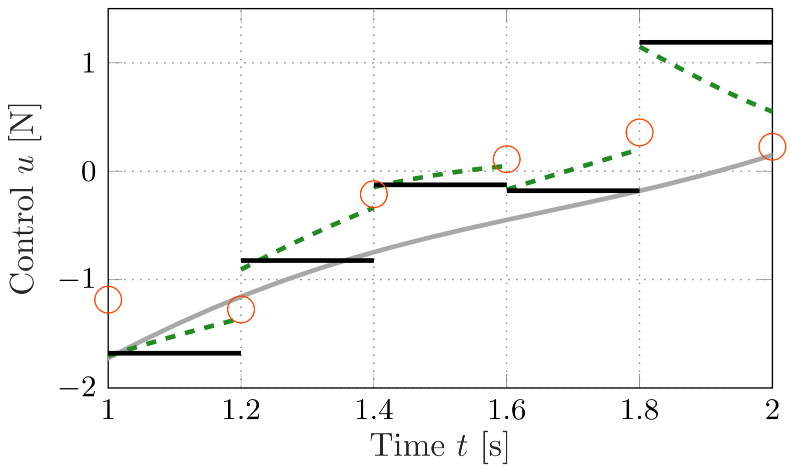

Figure 6: Control input zoomed in from Fig. 5: commanded control u (black solid line), initially planned open-loop control u_{OL} (grey solid line), closed-loop control u_{CL} given corrected state estimate and refined final time (green dashed line), predicted control input given state and final time predictions (red circle).

The expected final time t_{f} depends on time via the remaining time T, Fig. 7. Sub-optimality of the commanded final time is made visible by the gap between corrected and refined quantities. However, a single sensitivity update significantly improves the estimate reducing, if not eliminating, this gap. Notice also that Equation 27 is only approximately satisfied; it does not hold exactly because of the noise.

The relationships between final time, reduced cost and Hamilton function can be discussed based on Fig. 8. Quantities corresponding to six consecutive time steps, from $t = 1\,\mathrm{s}$ (black) to $t = 2\,\mathrm{s}$ (light grey), are plotted therein. As one might expect from Theorem 2 and Bellman's optimality principle, both the optimal final time and the optimal cost decrease in time, Fig. 8. While function $\tilde{\mathcal{J}}$ changes, the algorithm is able, within certain limits on disturbances, to track the optimal point of operation. Accordingly, as depicted in Fig. 9, the reduced Hamilton function h attains values close to zero when evaluated at this optimal final time. Notice that one can check (local) sufficient condition for optimality at every time step, by just looking at the slope of function h at the optimal final time, see Remark 11.

The developed method and the testing environment for these numerical tests have been programmed in plain MATLAB language. The computational effort in terms of elapsed time is approximately as follows: fixed-time LQR problem [1.4 ms], prediction step [5.1 ms], correction step [0.17 ms], refinement step [7.1 ms]. These valued are reported for completeness and should be compared against optimizers with similar implementations.

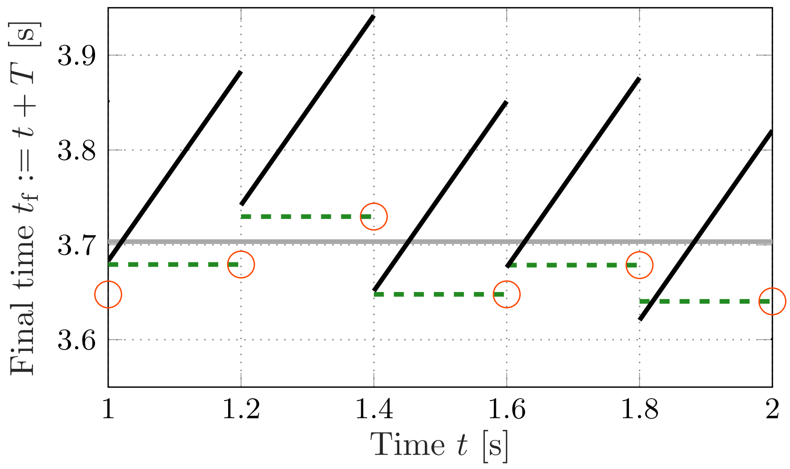

Figure 7: Optimal final time during MPC: piecewise constant commanded final time (black solid line), initially planned final time T_{OL} (grey solid line), closed-loop final time T_{CL} given corrected state estimate and refined final time (green dashed line), optimal final time prediction (red circle).

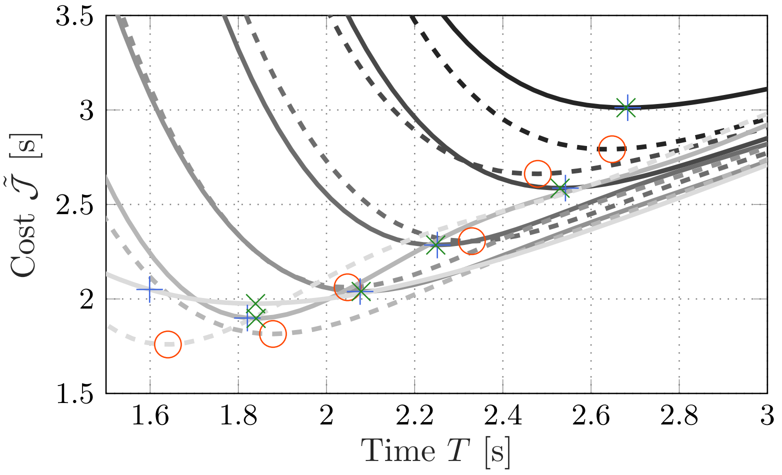

Figure 8: Reduced cost function $\tilde{\mathcal{J}}$ based on both predicted (dashed) and corrected (solid) state estimates, corresponding to six consecutive time steps, from $t = 1\,\mathrm{s}$ (black) to $t = 2\,\mathrm{s}$ (light grey). Predicted (red circle), corrected (blue plus) and refined (green cross) remaining time T.

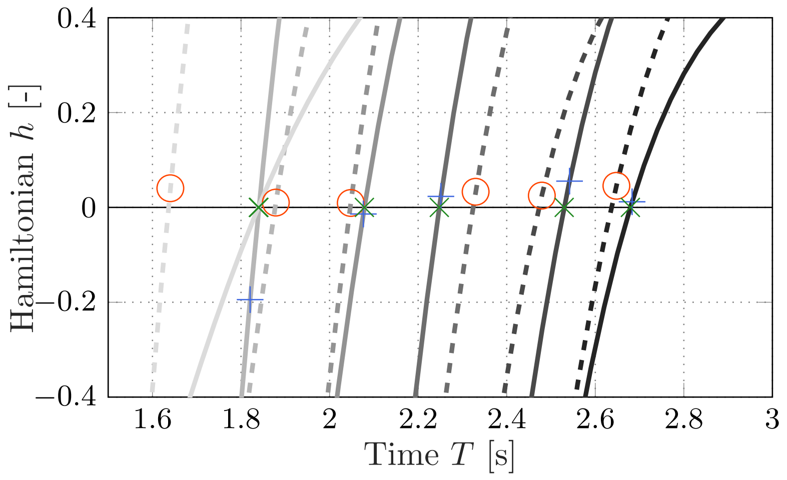

Figure 9: Reduced Hamilton function h based on both predicted (dashed) and corrected (solid) state estimates, corresponding to six consecutive time steps, from $t = 1\,\mathrm{s}$ (black) to $t = 2\,\mathrm{s}$ (light grey). Predicted (red circle), corrected (blue plus) and refined (green cross) remaining time T.

5 Conclusions

We presented a solution strategy for linear-quadratic optimal control problems with free final time and terminal state constraints. The proposed method exploits a bilevel reformulation and is effective especially if repeated solutions are required, e.g., within an MPC scheme. We established relations between the reduced objective functional of the upper level problem and the Hamiltonian function, which justify the proposed method.

Future work addresses the extension towards nonlinear and nonautonomous system dynamics. In the latter case, the condition of the Hamiltonian changes as it is zero only at the final time. Moreover, matrix exponentials have to be replaced by numerical integration methods. The extension towards nonlinear system dynamics requires numerical solution techniques to solve the lower level problem. Herein, issues with local minimizers and non-uniqueness arise.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under Award No, FA9550-14-1-0298.

A Proof of Theorem 2

The aim of this Section is to show that condition $\dot{T}_{\mathrm{CL}}(t) = -1$ for any $t \geq 0$ is necessary and sufficient to guarantee that open- and closed-loop evolutions match, in the sense suggested in Section 3.1, i.e. $x_{\mathrm{CL}}(t) = x_{\mathrm{OL}}(t)$ for any $t \geq 0$.

Owing to the state differential equations and initial conditions in IVPs (28) and (29), state trajectories match if and only if control trajectories do, i.e. $u_{\mathrm{CL}}(t) = u_{\mathrm{OL}}(t)$ for any $t \geq 0$. Then, because of the feedback control law (7), this is equivalent to $\lambda_{\mathrm{CL}}(t) = \lambda_{\mathrm{OL}}(t)$ for any $t \geq 0$, that is, costate trajectories match, too. Moreover, starting both from the value $\hat{\lambda}(\hat{T}(x_0), x_0)$, the latter condition is equivalent to $\dot{\lambda}_{\mathrm{CL}}(t) = \dot{\lambda}_{\mathrm{OL}}(t)$ for any $t \geq 0$. Starting from (28)–(29), and with results from Sections 2.3 and 2.5, we seek an expression of the latter condition in terms of $\dot{T}_{\mathrm{CL}}$. Omitting t for clarity, these read

$$\dot{\lambda}_{\mathrm{CL}} = \hat{\lambda}'_T(T_{\mathrm{CL}}, x_{\mathrm{CL}})\dot{T}_{\mathrm{CL}} + \hat{\lambda}'_x(T_{\mathrm{CL}}, x_{\mathrm{CL}})\dot{x}_{\mathrm{CL}} \tag{33}$$

$$\dot{\lambda}_{\mathrm{OL}} = M_{\lambda x} x_{\mathrm{OL}} + M_{\lambda\lambda}\lambda_{\mathrm{OL}} \tag{34}$$

Owing to linear system (14), we can express function $\hat{\lambda}$ at any $(T, x) \in \Gamma \times \mathbb{R}^n$ as $\hat{\lambda}(T, x) = \begin{bmatrix} I_n & 0_{n\times l}\end{bmatrix} \hat{s}(T, x)$, with $\hat{s}(T, x) := \hat{A}^{-1}(T)\hat{b}(T, x)$. Furthermore, one can evaluate the derivatives of $\hat{s}$ and then those of $\hat{\lambda}$; see Algorithm 1. Expressions for the derivatives of $\hat{A}$ and $\hat{b}$ can be analytically obtained from (13)–(14):

$$\hat{A}'_T(T) = \begin{bmatrix} C_{\mathrm{T}} E'_{x\lambda}(T) & 0_l \\ Q_{\mathrm{T}} E'_{x\lambda}(T) - E'_{\lambda\lambda}(T) & 0_{n\times l} \end{bmatrix}, \tag{35}$$

$$\hat{b}'_T(T, x) = \begin{pmatrix} -C_{\mathrm{T}} E'_{xx}(T) x \\ \left[E'_{\lambda x}(T) - Q_{\mathrm{T}} E'_{xx}(T)\right] x \end{pmatrix}, \tag{36}$$

$$\hat{b}'_x(T, x) = \begin{bmatrix} -C_{\mathrm{T}} E_{xx}(T) \\ E_{\lambda x}(T) - Q_{\mathrm{T}} E_{xx}(T) \end{bmatrix}. \tag{37}$$

Let us define, for future convenience, also $\Psi(T) \in \mathbb{R}^{n\times 2n}$, and its two $n \times n$ sub-blocks, for any $T \in \Gamma$, as

$$\Psi(T) := \begin{bmatrix} \Psi_1(T) & \Psi_2(T) \end{bmatrix} := \begin{bmatrix} I_n & 0_{n\times l} \end{bmatrix} \hat{A}^{-1}(T) \begin{bmatrix} -C_{\mathrm{T}} & 0_{l\times n} \\ -Q_{\mathrm{T}} & I_n \end{bmatrix} E(T) \tag{38}$$

Let us express terms in (33) using (11), (14), (28), matrix exponential $E(T)$ and (35)–(38). After lengthy calculations they read, omitting t:

$$\begin{aligned} &\hat{\lambda}'_T(T_{\mathrm{CL}}, x_{\mathrm{CL}}) \\ &= \begin{bmatrix} I_n & 0_{n\times l} \end{bmatrix} \hat{A}^{-1}(T_{\mathrm{CL}}) \cdot \left[\hat{b}'_T(T_{\mathrm{CL}}, x_{\mathrm{CL}}) - \hat{A}'_T(T_{\mathrm{CL}})\hat{s}(T_{\mathrm{CL}}, x_{\mathrm{CL}})\right] \\ &= \Psi(T_{\mathrm{CL}}) M \begin{pmatrix} x_{\mathrm{CL}} \\ \lambda_{\mathrm{CL}} \end{pmatrix} \end{aligned} \tag{39}$$

and

$$\begin{aligned} &\hat{\lambda}'_x(T_{\mathrm{CL}}, x_{\mathrm{CL}})\, \dot{x}_{\mathrm{CL}} \\ &= \begin{bmatrix} I_n & 0_{n\times l} \end{bmatrix} \hat{A}^{-1}(T_{\mathrm{CL}}) \hat{b}'_x(T_{\mathrm{CL}}, x_{\mathrm{CL}}) \cdot \left[A x_{\mathrm{CL}} + B u_{\mathrm{fb}}(x_{\mathrm{CL}}, \lambda_{\mathrm{CL}})\right] \\ &= \Psi(T_{\mathrm{CL}}) \begin{bmatrix} I_n & 0_n \\ 0_n & 0_n \end{bmatrix} M \begin{pmatrix} x_{\mathrm{CL}} \\ \lambda_{\mathrm{CL}} \end{pmatrix}. \end{aligned} \tag{40}$$

Owing to the work of Lu and Shiou [12], further expansion of matrix $\Psi_2(T)$ in (38), $T \in \Gamma$, after rearrangement, yields

$$\Psi_2(T) = -I_n \tag{41}$$

Substituting (39)–(41) into (33)–(34) and noticing that $x_{\mathrm{CL}}(t) = x_{\mathrm{OL}}(t)$ and $\lambda_{\mathrm{CL}}(t) = \lambda_{\mathrm{OL}}(t)$ hold for any $t \geq 0$, after rearrangement, the matching condition reads:

$$0 = \left[[\dot{T}_{\mathrm{CL}}(t) + 1]\Psi_1(T_{\mathrm{CL}}(t)) \quad -[\dot{T}_{\mathrm{CL}}(t) + 1]I_n\right] \cdot M \begin{pmatrix} x_{\mathrm{OL}}(t) \\ \lambda_{\mathrm{OL}}(t) \end{pmatrix} \tag{42}$$

for any $t \geq 0$. Since this equality holds true for any pair of IVPs (28)–(29), meaning that matrix M and vectors $x_{\mathrm{OL}}(t)$, $\lambda_{\mathrm{OL}}(t)$ can attain any value, an equivalent condition is that the matrix on the left is identically zero at any $t \geq 0$, namely, from (42):

$$\begin{cases} 0_n = [\dot{T}_{\mathrm{CL}}(t) + 1]\Psi_1(T_{\mathrm{CL}}(t)) \\ 0_n = -[\dot{T}_{\mathrm{CL}}(t) + 1]I_n \end{cases}, \ t \geq 0 \ . \tag{43}$$

Finally, the proof concludes because there exists an unique solution to (43), hence necessary and sufficient condition, that is $\dot{T}_{\mathrm{CL}}(t) = -1$ for any $t \geq 0$. $\square$

B Box-constrained final time and Projection

This section addresses necessary optimality conditions for the free box-constrained final time in the case $\Gamma := [T_l, T_u]$ with finite positive lower and upper bound T_l and T_u, $T_l < T_u$, namely Γ is a nonempty compact set. Lagrange multipliers ζ_l and ζ_u are introduced for the lower and upper bound, respectively. Recalling Hamilton function $\mathcal{H}$ from (5), the necessary conditions corresponding to the inequality constraints on T, satisfied by a solution, read

$$\begin{cases} \mathcal{H}[T^\star] + \zeta_u^\star - \zeta_l^\star = 0 \\ 0 \leq \zeta_u^\star \ \perp \ T^\star \leq T_u \\ 0 \leq \zeta_l^\star \ \perp \ T^\star \geq T_l \end{cases}, \tag{44}$$

where $\mathcal{H}[T]$ denotes the Hamilton function evaluated along the solution corresponding to final time T. After some rearrangements, condition (44) can be expressed equivalently as

$$\begin{cases} T^\star = T_l \\ \mathcal{H}[T^\star] \geq 0 \end{cases} \vee \begin{cases} T^\star \in (T_l, T_u) \\ \mathcal{H}[T^\star] = 0 \end{cases} \vee \begin{cases} T^\star = T_u \\ \mathcal{H}[T^\star] \leq 0 \end{cases} \tag{45}$$

The implemented projection scheme, consisting of the root-finding algorithm and a projected update step, see Algorithm 2, satisfies

$$\begin{cases} T^\star = \mathcal{P}_\Gamma\left(T^o\right) \\ \mathcal{H}[T^o] = 0 \end{cases} \tag{46}$$

where $\mathcal{P}_\Gamma$ denotes the projection operator onto Γ. In practice, the *unconstrained* optimal final time T^o is first computed, through the vanishing condition, as in Section 2.5, and then projected onto the feasible set Γ. When $T^o \in [T_l, T_u]$, conditions (45) and (46) are equivalent. When $T^o < T_l$ or $T^o > T_u$, instead, another condition is needed to make (45) and (46) equivalent:

$$\mathcal{H}'[T] > 0 \ , \quad T \text{ close to } T^\star \ . \tag{47}$$

In particular, first-order Taylor expansion of $\mathcal{H}$ at $T^\star$, along with (47), implies (45) from (46) and viceversa. Finally, from Remark 11, we stress that this property can be a posteriori checked and has an important meaning, being it related to the convexity of the reduced cost function. In fact, one can verify condition (47) every time $\tilde{h}'$ is evaluated within the refinement procedure, see Algorithm 2.

References

[1] G. Alefeld. On the convergence of Halley's method. *The American Mathematical Monthly*, 88(7):530–536, 1981.

[2] G. Bilardi and A. Ferrante. The role of terminal cost/reward in finite-horizon discrete-time LQ optimal control. *Linear Algebra and its Applications*, 425(2):323–344, 2007.

[3] P. Bosetti and F. Biral. Application of optimal control theory to milling process. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 4896–4901, Oct 2014.

[4] L. Van den Broeck, M. Diehl, and J. Swevers. Time optimal MPC for mechatronic applications. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pages 8040–8045, Shanghai, China, Dec 2009.

[5] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes.* PhD thesis, University of Heidelberg, 2001.

[6] A. Ferrante and L. Ntogramatzidis. A unified approach to the finite-horizon linear quadratic optimal control problem. *European Journal of Control*, 5:473–488, 2007.

[7] M. Gerdts. *Optimal Control of ODEs and DAEs.* De Gruyter, 2011.

[8] M. Gerdts. OCPID-DAE1 — optimal control and parameter identification with differential-algebraic equations of index 1. Technical report, University of the Federal Armed Forces at Munich, 2013. User's Guide, http://www.optimal-control.de.

[9] L. Grüne and J. Pannek. *Nonlinear model predictive control. Theory and algorithms.* London: Springer, 2011.

[10] J. P. Hespanha. *Linear Systems Theory.* Princeton Press, Princeton, New Jersey, Sep 2009. ISBN13: 978-0-691-14021-6.

[11] A. Locatelli. *Optimal control: An introduction.* Basel: Birkhäuser, 2001.

[12] T.-T. Lu and S.-H. Shiou. Inverses of 2×2 block matrices. *Computers & Mathematics with Applications*, 43(1):119–129, 2002.

[13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[14] J. Michael and M. Gerdts. Pro-active optimal control for semi-active vehicle suspension based on sensitivity updates. *Vehicle System Dynamics*, 53(12):1721–1741, 2015.

[15] K. D. Palagachev and M. Gerdts. Exploitation of the value function in a bilevel optimal control problem. In L. Bociu, J.-A. Désidéri, and A. Habbal, editors, *System Modeling and Optimization*, pages 410–419, Cham, 2016. Springer International Publishing.

[16] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. Mishchenko. *The mathematical theory of optimal processes.* Interscience Publishers, 1962. International series of monographs in pure and applied mathematics.

[17] R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl. Towards time-optimal race car driving using nonlinear MPC in real-time. In *53rd IEEE Conference on Decision and Control*, pages 2505–2510, Dec 2014.