TWCCC ⋆ Texas – Wisconsin – California Control Consortium

Technical report number 2014-01

Whither discrete time model predictive control?*

Gabriele Pannocchia†
Dept. of Civil and Industrial Engineering (DICI)
Univ. of Pisa (Italy)
James B. Rawlings‡
Dept. of Chemical and Biological Engineering
Univ. of Wisconsin, Madison (USA)
David Q. Mayne§
Dept. of Electrical & Electronic Engineering
Imperial College London (UK)
Giulio M. Mancuso¶
United Technologies, Rome (Italy)

April 23, 2014

*This report is an expanded version of the journal paper [20] conditionally accepted as Technical Note in the IEEE Transactions on Automatic Control.

†Email: g.pannocchia@diccism.unipi.it. Author to whom correspondence should be addressed.
‡Email: rawlings@engr.wisc.edu
§Email: d.mayne@imperial.ac.uk
¶Email: giuliomancuso@gmail.com

Abstract

This paper proposes an efficient computational procedure for the continuous time, input constrained, infinite horizon, linear quadratic regulator problem (CLQR). To ensure satisfaction of the constraints, the input is approximated as a piecewise linear function on a finite time discretization. The solution of this approximate problem is a standard quadratic program. A novel lower bound on the infinite dimensional CLQR problem is developed, and the discretization is refined until a user supplied error tolerance on the CLQR cost is achieved.

The offline storage of the matrices required for the solution of the model and adjoint equations, integration of the cost function, and computation of the cost gradient at several levels of discretization tailor the method for online use as required in model predictive control (MPC). The performance of the proposed algorithm is then compared with the standard *discrete time* algorithms used in most industrial model predictive control implementations. The proposed method is shown on two examples to be significantly more efficient than standard discrete time MPC that uses a sample time short enough to generate a cost close to the CLQR solution.

1 INTRODUCTION

In this paper we are concerned with the infinite horizon, *continuous time* optimal control problem for a linear system subject to input bounds. For brevity, we refer to this as the CLQR (constrained linear quadratic regulator) problem. It is perhaps the *simplest* optimal control problem of significant interest after the classical, unconstrained LQR. One of the compelling features of both LQR and CLQR is the guarantee of nominal, closed-loop stability that they provide for unconstrained and input constrained linear systems, respectively. Model predictive control, which is based on implementing solutions to optimal control problems as state measurements (or state estimates) become available, is arguably the most important advanced industrial control design method in use today. Although theoretical research on *nonlinear* MPC has reached at least a respectable level of completeness, almost all industrial applications remain based on linear models. Research on linear MPC therefore remains useful and important. Besides linearity, however, another feature of almost all industrial MPC methods is the use of *discrete time* models. It is this use of discrete time that we would like to examine in this paper. Is it necessary? Is it convenient? Is it as good as, or even better than, continuous time? For what reasons and in what ways? To provide a sound basis for comparison, we first develop and present a new algorithm for solving the CLQR problem. The problem is *doubly* infinite dimensional, first because the input is a continuous time *function*, and second because the cost function is defined on an *infinite* horizon. We show that neither feature causes insurmountable computational difficulties, and we can solve this problem reasonably efficiently with a guarantee of proximity to optimality.

The paper is organized as follows. In Section 2, we develop the basic numerical discretization of the continuous time problem using a piecewise linear input parameterization. In Section 3, we present quadrature formulas, based on matrix exponentiation, that can be computed and stored offline for fast, repetitive online calculation, as is required in MPC. These formulas allow us to efficiently solve the model and adjoint differential equations, evaluate the cost function, and its gradient with respect to the input. Because the original CLQR problem is (strictly) convex, we are able to develop a novel lower bound on the optimal cost; this is presented in Section 4. This lower bound enables a stopping criterion that meets a user specified proximity to optimality. In Section 5, we propose an algorithm for refining the discretization to solve the CLQR problem and discuss stopping criteria. Next in Section 6, we show that this algorithm converges. In Section 7, we provide numerical examples that show the algorithm can be solved quickly, and we briefly explore how it scales with state dimension. Notice that the usual discrete time issue of scaling with respect to horizon length is rendered moot because of the infinite horizon. Finally in Section 8 we draw conclusions of the study. At the end of the conclusions, we enumerate a number of issues that should be examined, or perhaps reexamined, in light of these new results when choosing between discrete time and continuous time in formulating MPC algorithms for industrial applications with linear models.

An earlier, condensed version of this paper was presented at the 2013 European Control Conference [19]. Some of the changes in this paper include the following: (i) improved second order, instead of first order, lower bounding functions, (ii) a revised and improved adaptive interval partitioning scheme, (iii) a convergence proof that does not require bisecting the largest interval, (iv), closed-loop simulations and comparison with discrete time MPC algorithms, and (v) larger scale examples.

Notation The symbols $\mathbb{R}$ and $\mathbb{Z}$ denote the fields of reals and integers, respectively. Given two reals (integers) a, b with $a < b$, $\mathbb{R}_{a:b}$ ($\mathbb{Z}_{a:b}$) denotes all reals (integers) x such that $a \leq x \leq b$. The symbol $'$ is the transpose operator. All vector inequalities are meant component wise. Given two vectors $x, y \in \mathbb{R}^n$, $(x, y) \triangleq \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ denotes the composite vector, $\langle x, y \rangle$ denotes the inner product, $|x| \triangleq \sqrt{\langle x, x \rangle}$ denotes the Euclidean norm, and $|x|_Q^2 \triangleq \langle x, Qx \rangle$. Given a matrix A and positive integers a, b, c, d, the symbol $A_{a:b,c:d}$ denotes the submatrix with rows a to b and columns c to d, and $A_{a:b,:}$ denotes the submatrix of rows a to b and all columns. $\mathbf{I}_m$ is the identity matrix of dimension $m \times m$ (the dimension is omitted if not necessary). Given two symmetric matrices A, B of the same dimensions, the relation $A > B$ ($A \geq B$) means that $A - B$ is positive definite (positive semi-definite); $\lambda_{\min}(A)$ is the smallest eigenvalue of A. Given a set S, $\text{int}(S)$ denotes its interior.

2 PRELIMINARIES

2.1 Continuous time optimal control problem

In this paper we address the computation of the optimal solution to the continuous time, infinite horizon, input constrained, linear quadratic regulation problem:

$$\mathbb{P}_\infty(x): \quad \inf_{u(\cdot)} \left\{ V_\infty(x, u(\cdot)) \triangleq \int_0^\infty \ell(x(t), u(t))dt \right\}, \quad \text{s.t. } x(0) = x, \tag{1a}$$

$$\dot{x} = f(x, u) \triangleq Ax + Bu, \quad \text{for all } t \in [0, \infty), \text{ and} \tag{1b}$$

$$u(\cdot) \in \mathcal{U}_\infty, \tag{1c}$$

in which $\mathcal{U}_\infty$ is the class of measurable controls defined on $[0, \infty)$ and taking values in $\mathbb{U} = \prod_{i=1}^m \mathbb{U}_i$, where $\mathbb{U}_i \triangleq [u_i^{\min}, u_i^{\max}]$, $0 \in \text{int}(\mathbb{U})$, which is a compact, convex subset of $\mathbb{R}^m$. The state $x \in \mathbb{R}^n$, and the function $\ell(\cdot)$ is quadratic: $\ell(x, u) \triangleq \frac{1}{2}(x'Qx + u'Ru)$.

Assumption 1. *The pair (A, B) is stabilizable and $(Q^{1/2}, A)$ is observable. $Q \geq 0$ and $R > 0$.*

Remark 2. *Observability of $(Q^{1/2}, A)$ can be relaxed to detectability. To clarify this point, assume without loss of generality that matrices (A, B, C), with $C = Q^{1/2}$ are in observability canonical form, i.e., the system of problem $\mathbb{P}_\infty(x)$ evolves as:*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \qquad y = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{2}$$

in which (A_{11}, C_1) is observable, A_{22} is Hurwitz, and x_2 is the unobservable mode. In fact, the evolution of x_1 does not depend on x_2 and the cost function can be expressed as:

$$\ell(x, u) = \tfrac{1}{2}(y'y + u'Ru) = \tfrac{1}{2}(x_1'C_1'C_1x_1 + u'Ru),$$

which shows that the cost function is not affected by substate x_2. Therefore, it can be removed from (2), reducing the system matrices to (A_{11}, B_1) and $Q = C_1'C_1$ with $(Q^{1/2}, A_{11})$ observable, without altering problem $\mathbb{P}_\infty(x)$. All the results in the paper, such as closed-loop stability, then apply to the (x_1, u) system. But since x_2 satisfies

$$\dot{x}_2 = A_{22}x_2 + \begin{bmatrix} A_{21} & B_2 \end{bmatrix} \begin{bmatrix} x_1 \\ u \end{bmatrix}$$

and A_{22} is Hurwitz, stability of the (x_1, u) system implies x_2 converges to zero and, therefore, stability of the (x, u) system as well.

We define $\mathbb{X}_\infty$ as the set of initial states x for which there exists $u(\cdot) \in \mathcal{U}_\infty$ such that $V_\infty(x, u(\cdot))$ is finite. Thus $V_\infty : \mathbb{X}_\infty \times \mathcal{U}_\infty \to \mathbb{R}_{\geq 0}$. Existence and uniqueness of a solution to $\mathbb{P}_\infty(x)$ for each $x \in \mathbb{X}_\infty$ is established after (6).

In order to rewrite the *infinite horizon* problem $\mathbb{P}(x)$ as an equivalent *finite horizon* problem, we define a suitable ellipsoid invariant set as follows. Let P be the unique symmetric positive definite solution to the (continuous time) algebraic Riccati equation (which exists under Assumption 1):

$$0 = Q + A'P + PA - PBR^{-1}B'P. \tag{3}$$

Given a positive scalar α, we consider the following set:

$$\mathbb{X}_f \triangleq \{x \in \mathbb{R}^n \mid x'Px \leq \alpha\}. \tag{4}$$

Clearly, $\mathbb{X}_f$ is an invariant (ellipsoidal) set for the *unconstrained* closed-loop system:

$$\dot{x} = Ax + Bu, \qquad u = Kx,$$

with $K = -R^{-1}B'P$. Because $\mathbb{U}$ contains the origin in its interior, if α is sufficiently small, then for any $x \in \mathbb{X}_f$ there holds $Kx \in \mathbb{U}$. Hence, $u(t) = Kx(t)$ remains feasible at all times with respect to the constraint (1c) once $x(t)$ has entered $\mathbb{X}_f$. Let $\mathbb{U}$ be rewritten as $\{u \in \mathbb{R}^m \mid Du \leq d\}$. The largest admissible value of α can be found as follows.

Algorithm 3 (Invariant set)**.** ***Require:*** *P, K, D, d.*
1: *Perform eigenvalue decomposition of P, i.e. $P = S\Lambda S'$.*
2: *Define $M \triangleq DKS\Lambda^{-\frac{1}{2}}$.*
3: *Evaluate radius of largest sphere, $y'y \leq r^2$, s.t. $My \leq d$.*
4: ***return*** *$\alpha \triangleq r^2$.*

Given $T > 0$, we replace $\mathbb{P}(x)$ by the following finite horizon optimal control problem:

$$\mathbb{P}_T(x): \quad \min_{u(\cdot)} \left\{ V_T(x, u(\cdot)) \triangleq \int_0^T \ell(x(t), u(t))dt + V_f(x(T)) \right\}, \tag{5a}$$

subject to $x(0) = x$ and

$$\text{model (1b) and constraint (1c) for all } t \in [0, T], \tag{5b}$$

$V_f(x) \triangleq \frac{1}{2}x'Px$ with $P > 0$ computed from (3), $\mathcal{U}_T$ is the class of measurable controls defined on $[0, T]$ and taking values in the compact, convex set $\mathbb{U}$. Thus, $V_T : \mathbb{R}^n \times \mathcal{U}_T \to \mathbb{R}_{\geq 0}$. $\mathbb{P}_T(x)$ has a unique solution for any $x \in \mathbb{R}^n$ [16, Thm. 14, Chapter 3]. Let $u_T^0(\cdot)$ be the (finite time) input trajectory solution to $\mathbb{P}_T(x)$ and $x_T^0(\cdot)$ the associated (finite time) state trajectory.

Proposition 4. *For each $x \in \mathbb{X}_\infty$, there exists $\bar{T} \in \mathbb{R}_{>0}$ such that $x_T^0(T) \in \mathbb{X}_f$ $\forall T \geq \bar{T}$, and $\lim_{T\to\infty} x_T^0(T) = 0$.*

Proof: We prove the result in three parts.

(i) *There exists finite $\bar{T}$ such that $x^0_{\bar{T}}(\bar{T}) \in \mathbb{X}_f$.* Since $(A, Q^{1/2})$ is observable, there exists an L such that $(A - LC)$ is Hurwitz and the linear system $\dot{x} = Ax + Bu$ can be rewritten as

$$\dot{x} = (A - LC)x + \begin{bmatrix} B & L \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix}$$

with $y = Q^{1/2}x$. Observability of $(A, Q^{1/2})$ also implies that $\mathbb{X}_f$ is compact, contains the origin in its interior, and there exists $a > 0$ such that x not in $\mathbb{X}_f$ implies $|x| \geq a$.

Since $\mathbb{X}_f$ is forward invariant for $\dot{x} = (A + BK)x$, if $x^0_T(T)$ is not in $\mathbb{X}_f$, then neither is $x^0_T(t)$ for all $t \in [0, T]$. So we assume contrary to what we wish to prove that there is no finite T such that $x^0_T(t) \in \mathbb{X}_f$ for $t \in [0, T]$. Lemma 15 (reported in Appendix) then applies and we know that $\left\|(u^0_T(\cdot), y^0_T(\cdot))\right\|_2 \to \infty$ as $T \to \infty$. But this contradicts the fact that $\int_0^T \ell(x^0_T(t), u^0_T(t))dt = \int_0^T (\left|y^0_T(t)\right|^2 + \left|u^0_T(t)\right|^2_R)dt$ is uniformly bounded for all T, and the result is established.

(ii). *Given $\bar{T}$ with $x^0_{\bar{T}}(\bar{T}) \in \mathbb{X}_f$, then $x^0_T(T) \in \mathbb{X}_f$ for every $T \geq \bar{T}$.* From optimality and the fact that $\mathbb{X}_f$ is forward invariant for the unconstrained optimal control problem $\dot{x} = (A + BK)x$, we have that $x^0_T(\bar{T}) = x^0_{\bar{T}}(\bar{T}; x)$ for $T \geq \bar{T}$ and the result follows.

(iii). *Convergence: $x^0_T(T) \to 0$ as $T \to \infty$.* Let $T > \bar{T}$. From (i) we know that $x^0_{\bar{T}}(\bar{T}) \in \mathbb{X}_f$ and from (ii) that $x^0_T(T) \in \mathbb{X}_f$. From optimality of the unconstrained control law, the evolution after $\bar{T}$ is given by $x^0_T(T) = e^{(A+BK)(T-\bar{T})}x_{\bar{T}}(\bar{T})$. Since $(A + BK)$ is Hurwitz, $x^0_T(T) \to 0$ as $T \to \infty$. □

For $x \in \mathbb{X}_\infty$, if $T \in \mathbb{R}_{\geq 0}$ is large enough that $x^0_T(T) \in \mathbb{X}_f$, since $V_f(x)$ is the optimal infinite horizon cost for any $x \in \mathbb{X}_f$, by the principle of optimality it follows that the (infinite time) input and state trajectories defined as:

$$\left(u^0_\infty(\cdot), x^0_\infty(\cdot)\right) \triangleq \begin{cases} \left(u^0_T(t), x^0_T(t)\right) & \text{if } t \in [0, T], \\ \left(Ke^{(A+BK)(t-T)}x^0_T(T), e^{(A+BK)(t-T)}x^0_T(T)\right) & \text{if } t > T, \end{cases} \tag{6}$$

are, respectively, the minimizer of $\mathbb{P}_\infty(x)$ and its associated state trajectory. Thus, $\mathbb{P}_\infty(x)$ and $\mathbb{P}_T(x)$ yield the same solution, i.e. $V^0_T(x) \triangleq V_T(x, u^0_T(\cdot)) = V^0_\infty(x) \triangleq V_\infty(x, u^0_\infty)$. From the above discussion it follows that $\mathbb{P}_\infty(x)$ has a unique solution for all $x \in \mathbb{X}_\infty$. In discrete time, earlier but less general results on existence and uniqueness of solutions to $\mathbb{P}_\infty$ are available. For example, [7] treats only the case $Q > 0$, whose proof is only a few lines (see Assumption H1 of their paper). The authors of [27, 28] treat the general case, $(A, Q^{1/2})$ detectable, but do not include a proof of existence and uniqueness of the infinite horizon problem (see remark 4 in [28]).

There is a rich literature on solution methods for finite horizon nonlinear constrained optimal control. In most approaches a piecewise constant input parameterization is considered and numerical discretization is deployed to derive and solve (approximate) optimality conditions (see e.g. [12, 5, 2, 3, 26, 13] and references therein). On the other hand, methods specifically tailored to constrained linear systems are less common, but some interesting results and methods can be found in [6, 14, 24, 10, 17, 25, 1, 8]. We remark that one distinguishing feature of our method is that computes a solution to $\mathbb{P}_T(x)$ that is accurate to a user defined tolerance. Furthermore, our method is based on the solution of strictly convex Quadratic Programming problems, for which reliable (off-the-shelf or tailored) algorithms exist, and has no specific restriction on the system (state and input) dimensions.

Remark 5. *We are assuming that the controlled system evolves as in (1b), i.e. the actuator hardware, if digital, is able to implement the continuous time input solution to* $\mathbb{P}_\infty(x)$ *without introducing noticeable discretization effects.*

2.2 Input parameterizations

For all $T \in \mathbb{R}_{>0}$, let γ be a *partition* of the interval $[0, T]$, defined as a sequence of $J_\gamma \in \mathbb{Z}_{>0}$ intervals $\{I_j \triangleq [t_j, t_{j+1}] \mid j \in \mathbb{Z}_{0:J_\gamma-1}\}$ such that $0 = t_0 < t_1 < \cdots < t_{J_\gamma} = T$. Let $\Delta_j \triangleq t_{j+1} - t_j$ denote the length of I_j; we assume that each Δ_j satisfies $\Delta_j = 2^{q_j}\delta$ with $q_j \in \mathbb{Z}_{\geq 0}$ and $\delta > 0$, in which case we say that $\gamma \in \Gamma_\delta^T$. In order to consider a finite parameterization of the function $u : [0, T] \to \mathbb{R}^m$, it is customary in sampled data control of continuous time systems (see, e.g. [31]) to assume that the input is constant in each interval I_j, i.e.,

$$u(t) = u_j \qquad \text{for all } t \in I_j. \tag{7}$$

Formally, given a partition γ of $[0, T]$, we define $\mathcal{U}_T^{\gamma,\text{ZOH}}$ as the set of all functions $u(\cdot) = \mathcal{U}_T$ satisfying the zero-order hold (ZOH) parameterization (7) in which $u_j \in \mathbb{U}$ for all $j \in \mathbb{Z}_{0:J_\gamma-1}$. Since $u(\cdot)$ is piecewise constant it is measurable. Besides the fact that restricting $u(\cdot)$ to the set $\mathcal{U}_T^{\gamma,\text{ZOH}}$ makes $\mathbb{P}_T(x)$ finite dimensional, it also ensures that $u(t) \in \mathbb{U}$ for all $t \in [0, T]$. In [18] we argued that a *better* choice is to assume the input piecewise *linear* in each interval:

$$u(t) = (1 - \eta_j(t))u_j + \eta_j(t)v_j, \qquad \text{for all } t \in I_j, \qquad \text{with } \eta_j(t) \triangleq \frac{t - t_j}{\Delta_j}. \tag{8}$$

Formally, given a partition γ of $[0, T]$, we define $\mathcal{U}_T^{\gamma,\text{PWLH}}$ as the set of all functions $u(\cdot) \in \mathcal{U}_T$ satisfying the piecewise linear hold (PWLH) parameterization (8) in which $(u_j, v_j) \in \mathbb{U}^2$ for all $j \in \mathbb{Z}_{0:J_\gamma-1}$. Notice that for all $j \in \mathbb{Z}_{0:J_\gamma-1}$, we have that $\eta_j(t_j) = 0$ and $\eta_j(t_{j+1}) = 1$. Thus, if $(u_j, v_j) \in \mathbb{U}^2$, then $u(t) \in \mathbb{U}$ for all $t \in I_j$, all $j \in \mathbb{Z}_{0:J_\gamma-1}$.

A variant to PWLH (8), also discussed in [18] and called forward first order hold (FFOH), enforces continuity of $u(\cdot)$ for all $t \in [0, T]$ by adding to (8) the restriction:

$$v_j = u_{j+1} \text{ for all } j \in \mathbb{Z}_{0:J_\gamma-2}. \tag{9}$$

For the sake of brevity, this variant is not discussed in this paper.

2.3 Discretized optimal control problem

Given a discretization γ and choosing either ZOH or PWLH, i.e. defining $\mathcal{U}^\gamma \triangleq \mathcal{U}_{\text{ZOH}}^\gamma$ or $\mathcal{U}^\gamma \triangleq \mathcal{U}_{\text{PWLH}}^\gamma$, we can obtain a suboptimal solution to $\mathbb{P}_T(x)$ by solving the following discretized optimal control problem:

$$\mathbb{P}_T^\gamma(x): \quad \min_{u(\cdot)\in\mathcal{U}^\gamma} V_T(x, u(\cdot)) \qquad \text{subject to } x(0) = x \text{ and model (1b)}\ . \tag{10}$$

As discussed in the next section, we rewrite $\mathbb{P}_T^\gamma(x)$ as an equivalent discrete time CLQR problem and solve it via Quadratic Programming (QP). Then, under certain conditions we accept the achieved solution or we refine the discretization γ.

In most existing approaches to solve CLQR (and general nonlinear optimal control) problems, the time discretization is uniform, but a number of methods exist which take advantage of (offline predetermined) non-uniform discretization schemes [18, 17, 23]. Such schemes can be equivalently interpreted as move blocking constraints in conventional discrete time MPC formulations [11].

3 ODE SOLVER FREE DISCRETIZATION

3.1 LQR discretization for ZOH via matrix exponential

Given an interval I_j, assuming to use the ZOH parameterization (7), it is well known [15,31] that we can compute an equivalent discrete time system evolution as:

$$x_{j+1} = A_j x_j + B_j u_j, \tag{11}$$

in which $x_j \triangleq x(t_j)$ and:

$$A_j = e^{A\Delta_j}, \qquad B_j = \int_0^{\Delta_j} e^{As} B ds. \tag{12}$$

Moreover, there holds:

$$V_T(x, u(\cdot)) = \sum_{j=0}^{J_\gamma - 1} \ell_j(x_j, u_j) + V_f(x(T)), \tag{13}$$

where

$$\ell_j(x_j, u_j) \triangleq \int_{t_j}^{t_{j+1}} \ell(x, u) dt = \tfrac{1}{2}(x_j' Q_j x_j + u_j' R_j u_j + 2x_j' M_j u_j), \tag{14}$$

in which

$$\begin{bmatrix} Q_j & M_j \\ M_j' & R_j \end{bmatrix} = \int_0^{\Delta_j} e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}' s} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} s} ds. \tag{15}$$

The above formulas allow one to compute all matrices $(A_j, B_j, Q_j, R_j, M_j)$ by solving a system of ordinary differential equations (ODE). However, Van Loan [29] showed that all of these matrices can be found by means of a single matrix exponentiation as follows. First, define a block upper triangular matrix C and partition its exponential as follow:

$$C \triangleq \begin{bmatrix} -A' & I & 0 & 0 \\ & -A' & Q & 0 \\ & & A & B \\ & & & 0 \end{bmatrix}, \quad e^{C\tau} \triangleq \begin{bmatrix} F_1(\tau) & G_1(\tau) & H_1(\tau) & K_1(\tau) \\ & F_2(\tau) & G_2(\tau) & H_2(\tau) \\ & & F_3(\tau) & G_3(\tau) \\ & & & F_4(t) \end{bmatrix}. \tag{16}$$

Then, obtain:

$$A_j = F_3(\Delta_j), \quad B_j = G_3(\Delta_j), \quad Q_j = F_3'(\Delta_j) G_2(\Delta_j), \quad M_j = F_3'(\Delta_j) H_2(\Delta_j),$$
$$R_j = R\Delta_j + \left[B' F_3'(\Delta_j) K_1(\Delta_j)\right] + \left[B' F_3'(\Delta_j) K_1(\Delta_j)\right]'. \tag{17}$$

3.2 LQR discretization for PWLH via matrix exponential

Numerical experience shows that computation of $(A_j, B_j, Q_j, R_j, M_j)$ for ZOH via matrix exponential formulas (16)–(17) is faster and (typically) more accurate than via an ODE solver. We show here how a similar procedure can be implemented for PWLH. To this aim, in each interval I_j, we can consider an augmented system with state $z \triangleq (z^{(1)}, z^{(2)}) \in \mathbb{R}^{2n}$, in which $z^{(1)}(t) \triangleq x(t)$ and $z^{(2)}(t) \triangleq u(t) - u_j = \eta_j(t)(v_j - u_j)$, and constant input $w_j = (u_j, v_j) \in \mathbb{R}^{2m}$. This augmented system evolves in I_j as:

$$\dot{z} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} B & 0 \\ -\frac{\mathbf{I}_m}{\Delta_j} & \frac{\mathbf{I}_m}{\Delta_j} \end{bmatrix} w_j. \tag{18}$$

If we set $A^* \triangleq \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}$, $B^* \triangleq \begin{bmatrix} B & 0 \\ -\frac{\mathbf{I}_m}{\Delta_j} & \frac{\mathbf{I}_m}{\Delta_j} \end{bmatrix}$, $Q^* \triangleq \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$ and we define C and its partitioned exponential as in (16) with (A, B, Q) replaced by (A^*, B^*, Q^*), under PWLH (8) we obtain:

$$z_{j+1} = A_j^* z_j + B_j^* w_j, \qquad \ell_j^*(z_j, w_j) \triangleq \int_{t_j}^{t_{j+1}} \ell(x, u)dt = \tfrac{1}{2}(z_j' Q_j^* z_j + w_j' R_j w_j + 2 z_j' M_j^* w_j), \tag{19}$$

where

$$\begin{aligned} A_j^* = F_3(\Delta_j), \quad B_j^* = G_3(\Delta_j), \quad Q_j^* = F_3'(\Delta_j) G_2(\Delta_j), \quad M_j^* = F_3'(\Delta_j) H_2(\Delta_j), \\ R_j = (1/6) \begin{bmatrix} 2R & R \\ R & 2R \end{bmatrix} \Delta_j + \left[B' F_3'(\Delta_j) K_1(\Delta_j)\right] + \left[B' F_3'(\Delta_j) K_1(\Delta_j)\right]'. \end{aligned} \tag{20}$$

Finally, by noticing that $z^{(2)}(t_j) = 0$, in the discrete time evolution and cost function we can remove the component $z^{(2)}$ to obtain:

$$x_{j+1} = A_j x_j + B_j w_j, \tag{21}$$

$$V_T(x, u(\cdot)) = \sum_{j=0}^{J_\gamma - 1} \ell_j(x_j, w_j) + V_f(x(T)), \tag{22}$$

where $A_j = A^*_{j\,1:n,1:n}$, $B_j = B^*_{j\,1:n,:}$ and

$$\ell_j(x_j, w_j) = \tfrac{1}{2}(x_j' Q_j x_j + w_j' R_j w_j + 2 x_j' M_j w_j), \tag{23}$$

in which , $Q_j = Q^*_{j\,1:n,1:n}$, $M_j = Q^*_{j\,1:n,:}$, and R_j is defined in (20).

We observe that in (21) the discrete time evolution of the system under PWLH is still described by a linear system with the original state x_j and an augmented input $w_j = (u_j, v_j)$.

For the sake of brevity, from now on we focus solely on PWLH, but all derivations and results will apply directly to ZOH, which can be seen as a PWLH in which $w_j = (u_j, u_j)$.

Given the above premises, problem $\mathbb{P}_T^\gamma(x)$ can be rewritten as a conventional discrete time CLQR problem. Let $\mathbf{u} \triangleq (w_0, w_1, \ldots, w_{J_\gamma - 1})$ be an augmented input sequence of

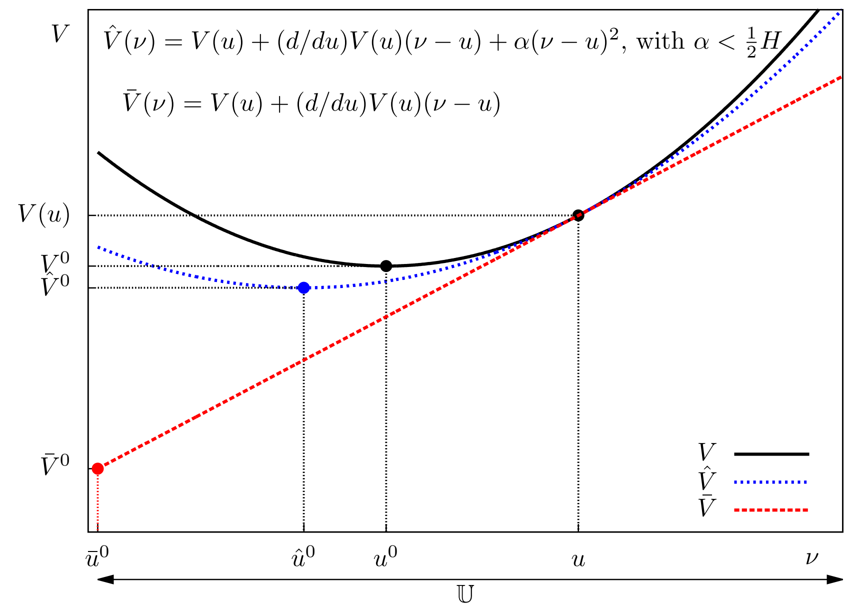

Figure 1: Illustrating the main idea of computing lower bounds on the optimal cost of the scalar function $V(\nu) = V(u) + (d/du)V(u)(\nu - u) + \frac{1}{2}H(\nu - u)^2$ in $\mathbb{U}$, given a feasible point u. $\hat{V}(\cdot)$ is the approximate second order lower bounding function, $\hat{u}^0$ is its minimizer and $\hat{V}^0$ its minimum value. $\bar{V}(\cdot)$ is the first order lower bounding function, $\bar{u}^0$ is its minimizer and $\bar{V}^0$ its minimum value.

length J_γ. Then, the discretized problem $\mathbb{P}_T^\gamma(x)$ can be equivalently written as:

$$\mathbb{P}_T^\gamma(x): \quad \min_{\mathbf{u} \in \mathbb{U}^{2J_\gamma}} \left\{ V_T^\gamma(x, \mathbf{u}) \triangleq \sum_{j=0}^{J_\gamma - 1} \ell_j(x_j, w_j) + V_f(x_{J_\gamma}) \right\},$$
$$\text{subject to } x_0 = x \text{ and model (21)} . \quad (24)$$

4 LOWER BOUNDS ON OPTIMAL COST OF $\mathbb{P}_T(x)$ AND $\mathbb{P}_T^\gamma(x)$

Next we show how, exploiting the convexity of both $\mathbb{P}_T(x)$ and $\mathbb{P}_T^\gamma(x)$, we can obtain a lower bound of the optimal cost of each problem, given any feasible input $u(\cdot)$. The main idea is graphically explained in Figure 1.

4.1 Lower bound of the continuous time optimal cost

$V_T : \mathbb{R}^n \times \mathcal{U}_T \to \mathbb{R}_{\geq 0}$ is defined by: $V_T(x, u(\cdot)) \triangleq \int_0^T \ell(x^u(t;x)), u(t)dt + V_f(x^u(T;x))$, in which $x^u(t;x)$ is the solution of (1b) at time t given that the initial state is x at time

0 and the control is $u(\cdot) \in \mathcal{U}_T$. Similarly, the cost due to another control $\nu(\cdot) \in \mathcal{U}_T$ is $V_T(x, \nu(\cdot)) \triangleq \int_0^T \ell(x^\nu(t;x)), \nu(t)dt + V_f(x^\nu(T;x))$. Let $\Delta u(\cdot) \triangleq \nu(\cdot) - u(\cdot)$ and let $\Delta x(\cdot) \triangleq x^\nu(\,\cdot\,;x) - x^u(\,\cdot\,;x)$ for all $t \in [0,T]$. Because $\ell(x,u) = \frac{1}{2}(x'Qx + u'Ru)$ and $V_f(x) = \frac{1}{2}x'Px$, we can write:

$$V_T(x,\nu(\cdot)) - V_T(x,u(\cdot)) = \int_0^T (\langle \Delta x(t), Qx^u(t;x)\rangle + \langle \Delta u(t), Ru(t)\rangle)dt + \langle \Delta x(T), Px^u(T)\rangle$$
$$+ \tfrac{1}{2}\int_0^T (\langle \Delta x(t), Q\Delta x(t)\rangle + \langle \Delta u(t), R\Delta u(t)\rangle)dt + \tfrac{1}{2}\langle \Delta x(T), P\Delta x(T)\rangle. \quad (25)$$

The first order terms may be computed in the usual way [4, pp. 148-149]:

$$\int_0^T \langle \Delta x(t), Qx^u(t;x)\rangle + \langle \Delta u(t), Ru(t)\rangle dt + \langle \Delta x(T), Px^u(T;x)\rangle =$$
$$\int_0^T \langle \nabla_u \mathcal{H}(x^u(t;x), u(t), \lambda^u(t;x)), \Delta u(t)\rangle dt,$$

in which the Hamiltonian $\mathcal{H} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is defined by $\mathcal{H}(x,u,\lambda) \triangleq \ell(x,u) + \lambda'(Ax + Bu)$, and $\lambda^u(t;x)$ is the solution at time t of the adjoint system:

$$-\dot{\lambda}(t) = A'\lambda(t) + Qx^u(t;x), \quad \lambda(T) = Px^u(T;x).$$

Since $Q \geq 0$ and $P > 0$, for any $R^* \in \mathcal{R} \triangleq \{S \mid 0 \leq R^* \leq R\}$, we have:

$$V_T(x,\nu(\cdot)) - V_T(x,u(\cdot)) \geq \int_0^T \langle g(x,u(\cdot))(t), (\nu(t) - u(t))\rangle + \tfrac{1}{2}|\nu(t) - u(t)|^2_{R^*} dt,$$

for all $\nu(\cdot) \in \mathcal{U}_T$, in which $g(\cdot)$ is defined by: $g(x,u(\cdot))(t) \triangleq \nabla_u \mathcal{H}(x^u(t;x), u(t), \lambda^u(t;x))$.

We now define the optimality function [21] $\theta : \mathbb{R}^n \times \mathcal{U}_T \to \mathbb{R}_{\leq 0}$ for problem $\mathbb{P}_T(x)$ as:

$$\begin{aligned}\theta(x,u(\cdot)) &\triangleq \inf_{\nu(\cdot)\in\mathcal{U}_T} \int_0^T \langle g(x,u(\cdot))(t), \nu(t) - u(t)\rangle + \tfrac{1}{2}|\nu(t) - u(t)|^2_{R^*} dt \\ &= \int_0^T \min_{v\in\mathbb{U}} \{\langle \nabla_u \mathcal{H}(x^u(t;x), u(t), \lambda^u(t;x)), v - u(t)\rangle + \tfrac{1}{2}|v - u(t)|^2_{R^*}\} dt,\end{aligned} \quad (26)$$

where the last equality is established in Proposition 16 in the Appendix if the function $L : [0,T] \times \mathbb{R}^m \to \mathbb{R}$ in Proposition 16 is defined by

$$L(t,v) \triangleq \langle \nabla_u \mathcal{H}(x^u(t;x), u(t), \lambda^u(t;x), v - u(t)\rangle + \tfrac{1}{2}|v - u(t)|^2_R$$

Thus, for any $\nu(\cdot) \in \mathcal{U}_T$, we have: $V_T(x,\nu(\cdot)) - V_T(x,u(\cdot)) \geq \theta(x,u(\cdot))$. Hence we have proved:

Proposition 6. *For any $(x, u(\cdot), T) \in \mathbb{R}^n \times \mathcal{U}_T \times \mathbb{R}_{>0}$, the following inequality holds:*

$$V_T^0(x) \geq V_T(x,u(\cdot)) + \theta(x,u(\cdot)).$$

4.2 Lower bound of the optimal cost for a given partition

In various stages of the algorithm described in § 5, it is useful to have a lower bound on the optimal cost of the discretized problem $\mathbb{P}_T^\gamma(x)$. Given an input $u(\cdot) \in \mathcal{U}_T^\gamma$ defined on a partition γ and its associated parameterization vector $\mathbf{u} = (w_0, w_1, \ldots, w_{J_\gamma - 1}) \in \mathbb{U}^{2J_\gamma}$, then:

$$V_T(x, u(\cdot)) = V_T^\gamma(x, \mathbf{u}) \triangleq \sum_{j=0}^{J_\gamma - 1} \left(\tfrac{1}{2}\langle x_j, Q_j x_j\rangle + \tfrac{1}{2}\langle w_j, R_j w_j\rangle + \langle x_j, M_j w_j\rangle \right) + \tfrac{1}{2}\langle x_{J_\gamma}, P x_{J_\gamma}\rangle.$$

For another $\nu(\cdot) \in \mathcal{U}_T^\gamma$, with associated parameterization vector $\boldsymbol{\nu} = (\nu_0, \nu_1, \ldots, \nu_{J_\gamma - 1})$ then:

$$V_T^\gamma(x, \boldsymbol{\nu}) - V_T^\gamma(x, \mathbf{u}) = \sum_{j=0}^{J_\gamma - 1} \left(\langle \Delta x_j, Q_j x_j\rangle + \langle \Delta w_j, R_j w_j\rangle + \langle \Delta x_j, M_j w_j\rangle \right) + \langle \Delta x_{J_\gamma}, P x_{J_\gamma}\rangle +$$
$$\tfrac{1}{2}\sum_{j=0}^{J_\gamma - 1} \left(\langle \Delta x_j, Q_j \Delta x_j\rangle + \langle \Delta w_j, R_j \Delta w_j\rangle + 2\langle \Delta x_j, M_j \Delta w_j\rangle \right) + \tfrac{1}{2}\langle \Delta x_{J_\gamma}, P \Delta x_{J_\gamma}\rangle. \quad (27)$$

The first order terms may be computed in the usual way [4, pp. 43-47]:

$$\sum_{j=0}^{J_\gamma - 1} \left(\langle \Delta x_j, Q_j x_j\rangle + \langle \Delta w_j, R_j w_j\rangle + \langle \Delta x_j, M_j w_j\rangle \right) + \langle \Delta x_{J_\gamma}, P x_{J_\gamma}\rangle = \langle g^\gamma(x, \mathbf{u}), \boldsymbol{\nu} - \mathbf{u}\rangle,$$

in which $g^\gamma(x, \mathbf{u}) \triangleq (g_0^\gamma(x, \mathbf{u}), g_1^\gamma(x, \mathbf{u}), \ldots, g_{J_{\gamma-1}}^\gamma(x, \mathbf{u}))$ with:

$$g_j^\gamma(x, \mathbf{u}) \triangleq \nabla_{w_j} \mathcal{H}_j(x_j, w_j, \lambda_{j+1}) = M_j' x_j + R_j w_j + B_j' \lambda_{j+1}, \quad j \in \mathbb{Z}_{0:J_\gamma - 1}, \quad (28)$$

where $\mathcal{H}_j : \mathbb{R}^n \times \mathbb{R}^{2m} \times \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian defined by $\mathcal{H}_j(x, w, \lambda) \triangleq \ell_j(x, w) + \lambda'(A_j x + B_j w)$ and $\{\lambda_0, \lambda_1, \ldots, \lambda_{J_\gamma}\}$ is the solution of the discrete time adjoint system:

$$\lambda_j = A_j' \lambda_{j+1} + M_j' w_j + Q_j x_j, \qquad \lambda_{J_\gamma} = P x_{J_\gamma}.$$

The second row of (27) consists of the second order terms. Forming the Schur complement, we note that: $(\langle \Delta x_j, Q_j \Delta x_j\rangle + \langle \Delta w_j, R_j \Delta w_j\rangle + 2\langle \Delta x_j, M_j \Delta w_j\rangle) \geq \langle \Delta w_j, (R_j - M_j' Q_j^{-1} M_j)\Delta w_j\rangle$. Since $P > 0$, it follows from (27) and (28) that, for all $\nu(\cdot)$ and $u(\cdot)$ in $\mathcal{U}_T^\gamma$:

$$V_T^\gamma(x, \boldsymbol{\nu}) - V_T^\gamma(x, \mathbf{u}) \geq \langle g^\gamma(x, \mathbf{u}), \boldsymbol{\nu} - \mathbf{u}\rangle + \tfrac{1}{2}\langle \boldsymbol{\nu} - \mathbf{u}, \mathbf{R}^*(\boldsymbol{\nu} - \mathbf{u})\rangle,$$

with $\mathbf{R}^*$ a block diagonal matrix formed by matrices $R_j^* \in \mathcal{R}_j \triangleq \{S \mid 0 \leq S \leq R_j - M_j' Q_j^{-1} M_j\}$, $j \in \mathbb{Z}_{0:J_\gamma - 1}$. We now define the optimality function $\theta^\gamma : \mathbb{R}^n \times \mathcal{U}_T^\gamma \to \mathbb{R}_{\leq 0}$ for $\mathbb{P}_T^\gamma(x)$ as:

$$\theta^\gamma(x, u(\cdot)) \triangleq \min_{\boldsymbol{\nu} \in \mathbb{U}^{2J_\gamma}} \langle g^\gamma(x, \mathbf{u}), \boldsymbol{\nu} - \mathbf{u}\rangle + \tfrac{1}{2}\langle \boldsymbol{\nu} - \mathbf{u}, \mathbf{R}^*(\boldsymbol{\nu} - \mathbf{u})\rangle = \sum_{j=0}^{J_\gamma - 1} \theta_j^\gamma(x, u(\cdot)), \quad (29)$$

$$\theta_j^\gamma(x, u(\cdot)) \triangleq \min_{w \in \mathbb{U}^2} \langle g_j^\gamma(x, \mathbf{u}), w - w_j\rangle + \tfrac{1}{2}\langle w - w_j, R_j^*(w - w_j)\rangle. \quad (30)$$

Thus, for any $\boldsymbol{\nu} \in \mathbb{U}^{2J_\gamma}$, we have: $V_T^\gamma(x, \boldsymbol{\nu}) - V_T^\gamma(x, \mathbf{u}) \geq \theta^\gamma(x, u(\cdot))$. Hence we have proved:

Proposition 7. *For any $(x, u(\cdot), T) \in \mathbb{R}^n \times \mathcal{U}_T^\gamma \times \mathbb{R}_{>0}$, the following inequality holds:*

$$V_T^{\gamma,0}(x) \geq V_T(x, u(\cdot)) + \theta^\gamma(x, u(\cdot)). \tag{31}$$

Finally, let $\theta^\delta(x, u(\cdot))$ denote $\theta^\gamma(x, u(\cdot))$ for the special *uniform* partition $\gamma^\delta \in \Gamma_\delta^T$ in which each constituent interval has length δ. Recalling that for any partition in Γ_δ^T, all intervals I_j have a length that is a multiple of δ (or at most equal to δ), we will refer to γ^δ as the *finest* partition.

5 ALGORITHM: CONCEPTUAL DESIGN AND PRACTICAL IMPLEMENTATION

5.1 Conceptual algorithm

As anticipated, we solve $\mathbb{P}_T^\gamma(x)$ repeatedly, refining γ at each iteration, until we obtain a satisfactory solution of $\mathbb{P}_\infty(x)$. We refer to $\tilde{\gamma} \in \Gamma_\delta^T$ as a *refinement* of $\gamma \in \Gamma_\delta^T$ if some of the intervals $\{\tilde{I}_j\}$ defining $\tilde{\gamma}$ are obtained by bisecting one or more intervals in the set $\{I_j\}$ that defines γ and if the remaining intervals in $\tilde{\gamma}$ are the same as the corresponding ones in γ. If $V_T^0(x)$ and $V_T^{\gamma,0}(x)$ are, respectively, the optimal value functions of $\mathbb{P}_T(x)$ and $\mathbb{P}_T^\gamma(x)$ then, clearly $V_T^{\gamma,0}(x) \geq V_T^0(x)$, for all $x \in \mathbb{R}^n$, all $\gamma \in \Gamma_\delta^T$, all permissible $\delta \in (0, T)$. Moreover, if $\tilde{\gamma}$ is a refinement of γ, it follows that $V_T^{\gamma,0}(x) \geq V_T^{\tilde{\gamma},0}(x)$. We now state the (conceptual) optimization algorithm to solve $\mathbb{P}_\infty(x)$.

Algorithm 8 (Conceptual algorithm)**.** ***Require:*** *$\delta, \epsilon > 0$, $\gamma \in \Gamma_\delta^T$, $c \in (0,1)$, $T, \Delta T > 0$.*

1. *Solve $\mathbb{P}_T^\gamma(x)$ yielding control $u(\cdot) \in \mathcal{U}_T^\gamma$ and state trajectory $x(\cdot)$. Compute $\theta^\delta(x, u(\cdot))$.*
2. *Refine γ (repeatedly if necessary) until $\theta^\gamma(x, u(\cdot)) \leq c\theta^\delta(x, u(\cdot))$.*
3. *If $\theta^\delta(x, u(\cdot)) \leq -\epsilon$, go to Step 1. Else, go to Step 4.*
4. *If $x(T) \notin \mathbb{X}_f$, define $I_{J_\gamma} = [T, T + \Delta T]$, and $\gamma \leftarrow \{\gamma, I_{J_\gamma}\}$, $T \leftarrow T + \Delta T$, $J_\gamma \leftarrow J_\gamma + 1$.*
5. *Replace $\epsilon \leftarrow \epsilon/2$, $\delta \leftarrow \delta/2$. Go to Step 1.*

A procedure for refining γ (repeatedly if necessary) is given in Section 5.2. In Step 5, $\epsilon \leftarrow \epsilon/2$ and $\delta \leftarrow \delta/2$ may be replaced, respectively, by $\epsilon \leftarrow c_1\epsilon$ and $\delta \leftarrow c_2\delta$, with $c_1 \in (0,1)$ and $c_2 = (1/2)^q$ (with $q \in \mathbb{Z}_{>0}$).

Remark 9. *The control $u(\cdot) \in \mathcal{U}_T^\gamma$ obtained in Step 1 satisfies $\theta^\gamma(x, u(\cdot)) = 0$; if $\tilde{\gamma}$ is the refined partition obtained in Step 2, and $u(\cdot)$ is not optimal for $\mathbb{P}_T^{\tilde{\gamma}}(x)$, then $\theta^{\tilde{\gamma}}(x, u(\cdot)) < 0$.*

Remark 10. *T is increased in Step 4 if the (implicit) terminal constraint $x(T) \in \mathbb{X}_f$ is not satisfied. As shown later by Theorem 13, this step occurs only a finite number of iterations.*

5.2 Refinement strategy

Since the length of each interval in the current partition γ is an even multiple of the current δ and since the length of all intervals in the refined partition should also be a multiple of δ, the refinement strategy of Step 2 consists of bisecting each interval with length greater than or equal to 2δ and selecting a subset whose bisection satisfies the condition in Step 2.

Suppose the current partition γ consists of the intervals $\{I_0, I_1, ..., I_{J_\gamma - 1}\}$. Because the current $u(\cdot)$ is optimal for $\mathbb{P}_T^\gamma(x)$, then $\theta_j^\gamma(x, u(\cdot)) = 0$ for all $j \in \mathcal{J}_\gamma \triangleq \mathbb{Z}_{0:J_\gamma - 1}$. If I_j is bisected, yielding $I_{j1} = [t_j, t_{j1}]$ and $I_{j2} = [t_{j1}, t_{j+1}]$ with $t_{j1} = \frac{t_j + t_{j+1}}{2}$, let $w_j \triangleq (u_j, v_j)$ be replaced by $w_{j1} = (u_j, \frac{u_j + v_j}{2})$ in I_{j1} and $w_{j2} = (\frac{u_j + v_j}{2}, v_j)$ in I_{j2}, and let x_{j1} and λ_{j1} denote the value of $x(\cdot)$ (the current state trajectory) and $\lambda(\cdot)$ at time t_{j1}. Then the gradients $g_{j1}^\gamma(x, u(\cdot))$ and $g_{j2}^\gamma(x, u(\cdot))$ of the cost with respect to w_{j1} and w_{j2} may be computed from (28) yielding:

$$\theta_j^{\tilde{\gamma}}(x, u(\cdot)) \triangleq \theta_{j1}^{\tilde{\gamma}}(x, u(\cdot)) + \theta_{j2}^{\tilde{\gamma}}(x, u(\cdot)), \tag{32}$$

where $\theta_{j1}^{\tilde{\gamma}}(x, u(\cdot))$ and $\theta_{j2}^{\tilde{\gamma}}(x, u(\cdot))$ are defined as in (30), respectively, for I_{j1} and I_{j2}. Notice that $\theta_j^{\tilde{\gamma}}(x, u(\cdot)) \leq 0$ is a lower bound on the cost reduction obtainable by bisecting I_j. Given a candidate set of intervals to be bisected, $\mathcal{J} \subseteq \mathcal{J}_\gamma$, we obtain: $\theta^{\tilde{\gamma}}(x, u(\cdot)) = \sum_{j \in \mathcal{J}} \theta_j^{\tilde{\gamma}}(x, u(\cdot))$. By ordering $\theta_j^{\tilde{\gamma}}(x, u(\cdot))$ in ascending manner, i.e., from the most negative to the least negative, $\mathcal{J}$ is chosen as the subset of $\mathcal{J}_\gamma$ with smallest cardinality such that the condition in Step 2 is satisfied by $\theta^{\tilde{\gamma}}(x, u(\cdot))$. If no such $\mathcal{J}$ can be found even if all intervals I_j are bisected, i.e., if $\mathcal{J} = \mathcal{J}_\gamma$, the procedure is repeated with γ replaced by the partition with *every* I_j bisected.

5.3 Practical considerations and algorithm with stopping conditions

The discrete time matrices appearing in the various steps of Algorithm 8 can be computed and stored offline for a (finite) number of possible interval sizes, in geometric sequence of ratio 2, using the formulas of § 3. The minimization in (30) is analytic if R_j^* are chosen diagonal, due to the fact that $\mathbb{U}$ (and hence $\mathbb{U}^2$ also) is a box constraint set. The choice of R_j^* diagonal is always possible, e.g. $R_j^* \triangleq \lambda_{\min}(R_j - M_j' Q_j^{-1} M_j)\mathbf{I}_{2m}$ is a valid choice because $0 < R_j^* \leq R_j - M_j' Q_j^{-1} M_j$. For a general polytopic set $\mathbb{U}$, the minimization in (30) is a small dimensional convex QP, namely in $2m$ decision variables.

For a given δ, the loop in Steps 1-3 is always exited in a finite number of iterations because, otherwise, the refinement of γ would reach γ^δ and then we would have $\theta^\gamma(x, u(\cdot)) = \theta^\delta(x, u(\cdot)) = 0$, which makes the condition to proceed to Steps 4-5 true. However, as written, Algorithm 8 never terminates because it would keep entering Step 5, reducing δ (and ϵ) and then going to Step 1.

A practical variant could terminate after Step 1 and return the computed solution $u(\cdot)$ when $\theta(x, u(\cdot)) \geq -\rho$, for a given $\rho > 0$. By doing so, there is a guarantee that the achieved cost $V_T(x, u(\cdot))$ satisfies: $V_T(x, u(\cdot)) - V_T^0(x) \leq \rho$. However, evaluation of $\theta(x, u(\cdot))$ from (26) requires computing a numerical integral of the scalar function $\psi : [0, T] \to \mathbb{R}_{\leq 0}$ whose value at any time $t \in [0, T]$ is

$$\psi(t) \triangleq \min_{v \in \mathbb{U}} \{ \langle \nabla_u \mathcal{H}(x^u(t; x), u(t), \lambda^u(t; x)), v - u(t) \rangle + \tfrac{1}{2} |v - u(t)|^2_{R^*},$$

i.e., $\theta(x, u(\cdot)) \triangleq \int_0^T \psi(t) dt$. Notice that if $R^* \leq R$ is chosen as a diagonal matrix, the previous minimization can be performed analytically.

Thus, evaluating $\theta(x, u(\cdot))$ and ensuring an exact bound on the termination error is achievable, but for fast closed-loop implementations a simpler alternative is to terminate after Step 1 when

$$\theta^{\delta}(x, u(\cdot)) \geq -\rho. \tag{33}$$

In this way the computed solution is guaranteed to satisfy $V_T(x, u(\cdot)) - V_T^{\delta,0}(x) \leq \rho$, i.e., the solution to $\mathbb{P}_T^{\gamma}(x)$ is a ρ-close approximation to the problem $\mathbb{P}_T^{\gamma^{\delta}}(x)$ at the *current* finest partition γ^{δ}. Notice that ρ should be chosen (significantly) smaller than the initial value of ϵ.

6 PROPERTIES OF THE ALGORITHM

In this section we discuss the main properties of Algorithm 8.

6.1 The space of control and state trajectories

In order to analyze the convergence properties, we notice that each $u(\cdot) \in \mathcal{U}_T$ lies in $L_p(T) \triangleq \mathcal{L}_p([0,T], \mathbb{R}^m)$ for all $p \in \{1, 2, \ldots, \infty\}$, in which $\mathcal{L}_p([0,T], \mathbb{R}^m) = \{u : [0,T] \to \mathbb{R}^m \mid u(\cdot) \text{ measurable}, \ ||u(\cdot)||_p < \infty\}$, with $||u(\cdot)||_p \triangleq \left[\int_0^T |u(t)|^p dt\right]^{1/p}$ and $||u(\cdot)||_\infty \triangleq$ ess $\sup_{[0,T]} |u(t)|$.

The space $L_p(T)$ is a Banach space. Moreover, the spaces $L_p(T)$, $p = 1, 2, \ldots, \infty$ are nested; i.e. $p < q$ implies $L_q(T) \subset L_p(T)$. In fact, since $[0,T]$ has finite measure T, for all $u(\cdot) \in \mathcal{U}_T$

$$\|u(\cdot)\|_p \leq T^{(1/p - 1/q)} \|u(\cdot)\|_q, \tag{34}$$

so that $\|u(\cdot)\|_q \to 0$ implies $\|u(\cdot)\|_p \to 0$ for all $p, q \in \mathbb{I}_{\geq 1}$, $p < q$ and all $u \in \mathcal{U}_T$. It is also possible to show that, for all $u(\cdot) \in \mathcal{U}_T$ and all $p, q \in \mathbb{I}_{\geq 1}$, $\|u(\cdot)\|_p \to 0$, $u(\cdot) \in \mathcal{U}_T$, implies $\|u(\cdot)\|_q \to 0$.

The next result follows from Theorem 3.1 in [9].

Theorem 11. *For all $(x, T) \in \mathbb{R}^n \times \mathbb{R}_{>0}$, $u_T^0 : [0,T] \to \mathbb{U}$ is Lipschitz continuous.*

Since $u(\cdot)$ is bounded and T is finite, it follows that the solution $x(\cdot) = \phi(\cdot; x, u(\cdot))$ of (1b) is absolutely continuous for any $(x, u(\cdot)) \in \mathbb{R}^n \times \mathcal{U}_T$.

6.2 Convergence of Algorithm 8

Let $u_i(\cdot)$, $x_i(\cdot)$, ϵ_i, γ_i, δ_i and T_i denote, respectively, the values of $u(\cdot)$, $x(\cdot)$, ϵ, γ, δ and T at iteration i of Algorithm 8, where $i \in \mathbb{Z}_{\geq 1}$ increases each time Step 1 is executed.

Proposition 12. *The loop in Steps 1–3 is always exited in a finite number of iterations.*

We can now state the main result.

Theorem 13. *For each $x \in \mathbb{X}_\infty$, there exists an $i^* \in \mathbb{Z}_{\geq 1}$ and a $T^* \in \mathbb{R}_{>0}$ such that $T_i = T^*$ and $x_i(T_i) \in \mathbb{X}_f$ for all $i \geq i^*$. Also $V_{T_i}(x, u_i(\cdot)) \to V_{T^*}^0(x) = V_\infty^0(x)$ and $u_i(\cdot) \to u_{T^*}^0(\cdot)$ in $L_p(T^*)$ for all $p \in \mathbb{Z}_{\geq 1}$ as $i \to \infty$ (with $u_{T^*}^0(\cdot) = u_\infty^0(\cdot)$ restricted to $[0, T^*]$.)*

Proof: Since the loop Steps 1–3 is always exited in a finite number of iterations, both ϵ_i and δ_i tend to 0 as $i \to \infty$. For each $i \in \mathbb{Z}_{\geq 1}$, let $\tilde{u}_i(\cdot) \in \mathcal{U}_{T_i}$ denote the sample-hold version of $u^0_{T_i}(\cdot)$ in a PWLH sense obtained by sampling $u^0_{T_i}(\cdot)$ at times $j\delta_i$, $j \in \mathbb{Z}_{0:(T_i/\delta_i-1)}$; thus $\tilde{u}_i(\cdot)$ is defined as in (8) with $u_j \triangleq u^0_{T_i}(j\delta_i)$ and $v_j \triangleq u^0_{T_i}(j+1)\delta_i)$. Let $T^\dagger$ denote the smallest time T such that $x^0_T(T) \in \mathbb{X}_f$. By Theorem 11, $u^0_{T^\dagger}(\cdot)$ is Lipschitz continuous in $[0, T^\dagger]$ with Lipschitz constant κ_1 and is equal to $u^0_\infty(\cdot)$ (the optimal control for $\mathbb{P}_\infty(x)$) restricted to $[0, T^\dagger]$. In $[T^\dagger, \infty)$, $u^0_\infty(\cdot)$ is as defined in (6) with $T = T^\dagger$; since $x^0_\infty(\cdot)$ is bounded in $[T^\dagger, \infty)$ (because $x^0_\infty(t) \in \mathbb{X}_f$, $t \in [T^\dagger, \infty)$), so is $\dot{u}^0_\infty(\cdot)$ in $[T^\dagger, \infty)$. Hence $u^0_\infty(\cdot)$ is Lipschitz continuous with constant κ_2 in this interval. It follows that $u^0_T(\cdot)$ is Lipschitz continuous for all $T \in [0, \infty)$ with Lipschitz constant $\kappa = \max\{\kappa_1, \kappa_2\}$.

Since $\delta_i \to 0$ as $i \to \infty$, $\{\tilde{u}_i(\cdot) \in \mathcal{U}^{\delta_i}_{T_i} \mid i \in \mathbb{Z}_{\geq 1}\}$ is a sequence of controls satisfying $||\tilde{u}_i(\cdot) - u^0_{T_i}(\cdot)||_\infty \to 0$ as $i \to \infty$ so that $|V^0_{T_i}(x) - V_{T_i}(x, \tilde{u}_i(\cdot))| \to 0$ as $i \to \infty$. Let $V^{\delta_i,0}_{T_i}(\cdot)$ be the optimal value function for $\mathbb{P}^{\gamma^{\delta_i}}_{T_i}(x)$. Then $V^0_{T_i}(x) \leq V^{\delta_i,0}_{T_i}(x) \leq V_{T_i}(x, \tilde{u}_i(\cdot)))$ for all i so that $|V^{\delta_i,0}_{T_i}(x) - V^0_{T_i}(x)| \to 0$ as $i \to \infty$. The algorithm ensures $V_{T_i}(x, u_i(\cdot)) = V^{\gamma_i,0}_{T_i}(x) \geq V^{\delta_i,0}_{T_i}(x)$ for all i. Hence

$$V^0_{T_i}(x) \leq V^{\delta_i,0}_{T_i}(x) \leq V_{T_i}(x, u_i(\cdot)) \leq V^{\delta_i,0}_{T_i}(x) - \theta^{\delta_i}(x, u_i(\cdot))$$

where the last inequality follows from (31) with γ chosen as γ^{δ_i}. It follows from the convergence of $|V^{\delta_i,0}_{T_i}(x) - V^0_{T_i}(x)|$ and of $\theta^{\delta_i}(x, u_i(\cdot))$ to zero, that

$$|V_{T_i}(x, u_i(\cdot)) - V^0_{T_i}(x)| \to 0$$

as $i \to \infty$.

In (25) let $\nu(\cdot) = u_i(\cdot)$, $u(\cdot) = u^0_{T_i}(\cdot)$ and $T = T_i$ noting that $V^0_{T_i}(x) = V_{T_i}(x, u^0_i(\cdot))$. The first order component (first line) of $V_{T_i}(x, \nu(\cdot)) - V_{T_i}(x, u(\cdot))$ is non-negative since both $\nu(\cdot) = u_i(\cdot)$ and $u(\cdot) = u^0_{T_i}(\cdot)$ lie in $\mathcal{U}_{T_i}$ and $u^0_{T_i}(\cdot)$ is optimal. Since $|V_{T_i}(x, u_i(\cdot)) - V^0_{T_i}(x)| \to 0$, each term on the right hand side of (25), in particular, $(1/2)|x_i(T_i) - x^0_{T_i}(T_i)|_P$, tends to zero as $i \to \infty$. Hence $|x_i(T_i) - x^0_{T_i}(T_i)| \to 0$ as $i \to \infty$. From Proposition 4, it follows that $x^0_{T_i}(T_i) \to 0$ as $T_i \to \infty$. Thus $x_i(T_i) \to 0$ as $i \to \infty$ if Step 4 is entered at every iteration of the algorithm. However, since T_i increases by a finite amount each time Step 4 is entered, there exists a finite integer i^* such that $x_i(T_i) \in \mathbb{X}_f$ and $T_i = T^*$ for all $i \geq i^*$ (T_i stops increasing only after $x_i(T_i)$ enters $\mathbb{X}_f$). This also implies that $x^0_{T_i}(T_i) \in \mathbb{X}_f$ for all $i \geq i^*$. Since $x^0_{T_i}(T_i) \in \mathbb{X}_f$ implies $V^0_{T_i}(x) = V^0_\infty(x)$, it follows that $V_{T_i}(x, u_i(\cdot)) \to V^0_{T^*}(x) = V^0_\infty(x)$ as $i \to \infty$.

It follows from (25), with $\nu(\cdot) = u_i(\cdot)$ and $u(\cdot) = u^0_{T^*}(\cdot)$, and the non-optimality of $\nu(\cdot) = u_i(\cdot)$ that

$$(1/2)\int_0^{T^*} |u_i(t) - u^0_{T^*}(t)|^2_R dt \leq V_{T^*}(x, u_i(\cdot)) - V^0_{T^*}(x)$$

for all $i \geq i^*$ so that $u_i(\cdot) \to u^0_{T^*}(\cdot)$ in $L_2(T^*)$ (hence, in $L_p(T^*)$ for any $p \in \mathbb{Z}_{>0}$) as $i \to \infty$, where $u^0_{T^*}(\cdot) = u^0_\infty(\cdot)$ restricted to $[0, T^*]$. □

Remark 14. *To prove this theorem, we do not have to assume that the largest interval in the partition goes to zero as $i \to \infty$.*

7 APPLICATION EXAMPLES

7.1 Introduction and performance indicators

We present in this section some illustrative simulation results. Computations are performed in Matlab (R2012b) on a MacBook Air (1.8 GHz Intel Core i7, 4 GB of RAM). The discretized CLQR problems $\mathbb{P}_T^\gamma(x)$ are solved using the function `quadprog.m` [1], in which both (augmented) input and state sequences ($\{w_j\}$ and $\{x_j\}$) are the QP decision variables (see, e.g., [22]). Timing is measured with the functions `tic` and `toc`. We remark that with this formulation and solver, the solution time approximately scales linearly with the number of intervals J_γ. All required discretized matrices are precomputed offline and stored in order to speed up the online computations. The following performance indicators are considered during the execution of Algorithm 8:

- number of intervals, J_γ, at a given iteration;
- continuous time cost error bound, $-\theta(x, u(\cdot))$, at a given iteration;
- cumulative solution time (in ms) of all past and current iterations.

In the computation of $\theta^\delta(x, u(\cdot))$ given in (29)–(30) we use $R_j^* \triangleq \lambda_{\min}(R_j - M_j' Q_j^{-1} M_j)$, and in the computation of $\theta(x, u(\cdot))$ given in (26) we use $R^* \triangleq R$ (because in all examples we use continuous time R diagonal).

7.2 Example # 1: SISO open-loop stable system

The first example is a three-state one-input system defined by the (continuous time) matrices:

$$A = \begin{bmatrix} -0.1 & 0 & 0 \\ 0 & -2.0 & -6.25 \\ 0 & 4.0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 \\ 2.0 \\ 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R = 0.1.$$

We remark that the system matrix A has three stable eigenvalues ($-0.1,\ -1 \pm 4.899i$). Given the initial state $x(0) = \begin{bmatrix} 1.3440 & -4.5850 & 5.6470 \end{bmatrix}'$, we consider the first five iterations of Algorithm 8 in three different variants:

- using PWLH parameterization (8) and adaptive refinement as in § 5.2;
- using PWLH parameterization (8) and fixed refinement in which all intervals are bisected;
- using ZOH parameterization (7) and fixed refinement in which all intervals are bisected.

[1] With options: `'interior-point-convex'` algorithm, 'function tolerance' of 10^{-12} and 'variable tolerance' of 10^{-10}.

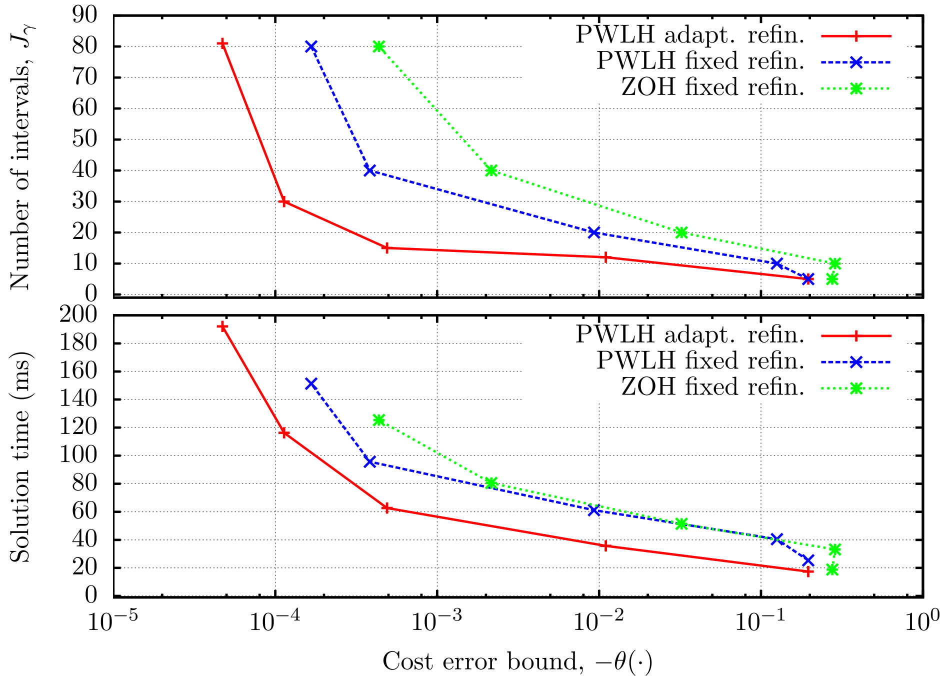

Figure 2: Example 1: Performance indicators during the first five iterations of Algorithm 8 using PWLH with adaptive discretization refinement (red), PWLH with fixed discretization refinement (blue), ZOH with fixed discretization refinement (green).

In all cases we use the following fixed parameters $T = 10$, $c = 0.8$, and initial values of $\delta = 0.125$ and $\epsilon = 0.1$. The storage of the required discretized matrices, for 11 different interval sizes, takes 7 kB. Comparative results are depicted in Figure 2 in which number of intervals and solution time are plotted against the continuous time cost error bound. By comparing adaptive and fixed refinement strategies (using the same PWLH parametrization) we immediately observe that the adaptive refinement allows Algorithm 8 to achieve smaller errors with far fewer intervals (top plot), and in turns this reduces the required computation time (bottom plot). By comparing PWLH and ZOH (using the same fixed refinement procedure, i.e. the same discretization) we observe that PWLH grants a much lower error than ZOH for the same number of intervals (top plot).

In order to further examine the efficiency of the adaptive refinement procedure we depict in Figure 3 the input $u_i(\cdot)$ achieved during each iteration of Algorithm 8 using PWLH. In this picture the discretization intervals at each iteration are also reported. From this picture we notice that the devised adaptive procedure is able to detect, at each iteration, which intervals require (possibly repeated) bisection. Hence, the proposed algorithm appears parsimonious in the usage of intervals and hence of decision variables in the discretized CLQR problems $\mathbb{P}_T^\gamma(x)$.

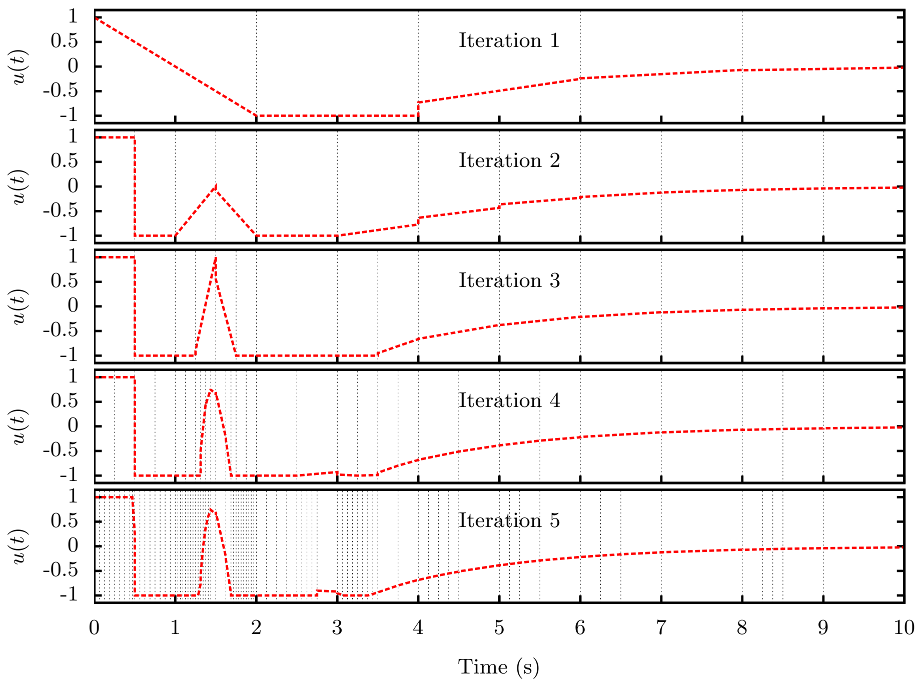

Figure 3: Example 1: Input $u^\gamma(\cdot)$ computed during the first five iterations of Algorithm 8 using PWLH, and associated (adaptive) discretization intervals.

Next, we discuss the closed-loop performance of the proposed continuous time CLQR and compare it with that of *standard* discrete time MPC, formulated as in [22] and solved using the function `quadprog.m` with same options and tolerances used in solving problems $\mathbb{P}_T^\gamma(x)$. Every sampling time T_s, given the current state x, we solve $\mathbb{P}_T(x)$ and inject the first portion, $t \in [0, T_s)$, of the computed input $u(\cdot)$. Three controllers, which use a sampling time of $T_s = 1$ s, are compared: CTCLQR 1 and CTCLQR 2 use Algorithm 8 with PWLH parameterization and stopping condition (33) for $\rho = 5 \cdot 10^{-4}$ and $\rho = 5 \cdot 10^{-3}$, respectively; DTMPC 1 is a discrete time MPC with horizon $N = 10$ (hence same finite time as CTCLQR 1 and CTCLQR 2 of $T = T_s N = 10$ s). DTMPC 2 instead uses a horizon of $N = 200$ and hence a sampling time of $T_s = T/N = 0.05$ s. In Table 1 we report: the ratio between computation time T_C and sampling time T_s, the closed-loop suboptimality with respect to CTCLQR 1 defined as $(V_{CL} - V_{CL}^0)/V_{CL}^0$ in which $V_{CL} = \int_0^{T_f} \ell(x, u) dt$, $T_f = 20$ s, is the closed-loop cost achieved with a given controller and V_{CL}^0 is that achieved with CTCLQR 1. We can observe that CTCLQR 2 has a small relative suboptimality of (up to 0.53%), whereas DTMPC 1 has a suboptimality (up to 28.2%). By decreasing the sample time, DTMPC 2 achieves a small suboptimality (up to 0.12%). However, CTCLQR 1, CTCLQR 2 and DTMPC 1 have computation times significantly smaller than their sampling time. DTMPC 2, instead, have computation times similar and even larger than its sampling time.

Table 1: Example 1: closed-loop performance comparison of continuous time CLQR and discrete time MPC. Results are averaged over 50 closed-loop simulations of 20 s, each starting from a different random initial state.

Controller	(T_C/T_s)		$(V_{CL} - V^0_{CL})/V^0_{CL}$	
	Mean	Max	Mean	Max
CTCLQR 1 ($\rho = 5 \cdot 10^{-4}$, $T_s = 1\,\text{s}$)	0.0229	0.1144	–	–
CTCLQR 2 ($\rho = 5 \cdot 10^{-3}$, $T_s = 1\,\text{s}$)	0.0178	0.0585	0.0018	0.0053
DTMPC 1 ($N = 10$, $T_s = 1\,\text{s}$)	0.0096	0.0141	0.1223	0.2820
DTMPC 2 ($N = 200$, $T_s = 0.05\,\text{s}$)	0.8141	1.3879	0.0002	0.0012

7.3 Example #2: MIMO open-loop unstable system

The second example is an open-loop unstable three-input three-output system defined by the transfer function matrix:

$$G(s) = \begin{bmatrix} \frac{-5s+1}{36s^2+6s+1} & \frac{0.5}{8s+1} & 0 \\ 0 & \frac{0.1(-10s+1)}{(8s+1)s} & \frac{-0.1}{(64s^2+6s+1)s} \\ \frac{-2s+1}{12s^2+3s+1} & 0 & \frac{2(-5s+1)}{16s^2+2s+1} \end{bmatrix}.$$

A minimal realization of the system has $n = 10$ states. The input constraint set is $\mathbb{U} = [-1,\ 1]^3$, and we consider continuous time LQR penalties of $Q = \mathbf{I}_{10}$ and $R = 0.25\,\mathbf{I}_3$.

As in the first example, we first focus on the solution of $\mathbb{P}_T(x)$ for a given (random) initial state, chosen in a way that inputs constraints become active for some time in $[0,\ T]$. We consider the same three variants of Algorithm 8 discussed in the first example (namely, PWLH with adaptive refinement, PWLH with fixed refinement and ZOH with fixed refinement), and in all cases we use the following parameters $T = 60$, $c = 0.85$, and initial values of $\delta = 0.75$ and $\epsilon = 0.1$. The storage of the required discretized matrices, for 11 different interval sizes, takes 82 kB. Comparative results are depicted in Figure 4 in which number of intervals and solution time are plotted against the continuous time cost error bound during the algorithm's iterations. Also in this multivariable example we observe that the adaptive refining strategy is much more effective than a brute force bisection approach. For instance, at the second iteration the adaptive refining strategy (red curves) uses about 20 intervals and the cost error bound is less than 10^{-3}, value that is achieved by the fixed refinement strategy (blue curves) at the fourth iteration using 40 intervals. Consequently, the cumulative computation time is less than 100 ms using adaptive bisection and is about 220 ms using fixed bisection. Similar considerations apply to other iterations.

In Table 2 we report the performance indicators, as discussed for the first example, using four alternative controllers in closed-loop implementation (sampling time $T_s = 5\,\text{s}$, for all controllers except DTMPC 2). In particular, CTCLQR 1 and CTCLQR 2 use Algorithm 8 with PWLH parameterization and stopping condition (33) with $\rho = 10^{-4}$ and $\rho = 10^{-3}$, respectively; DTMPC 1 is a discrete time MPC with horizon $N = 12$ (hence same finite time as CTCLQR 1 and CTCLQR 2 of $T = T_sN = 60\,\text{s}$). DTMPC 2 instead

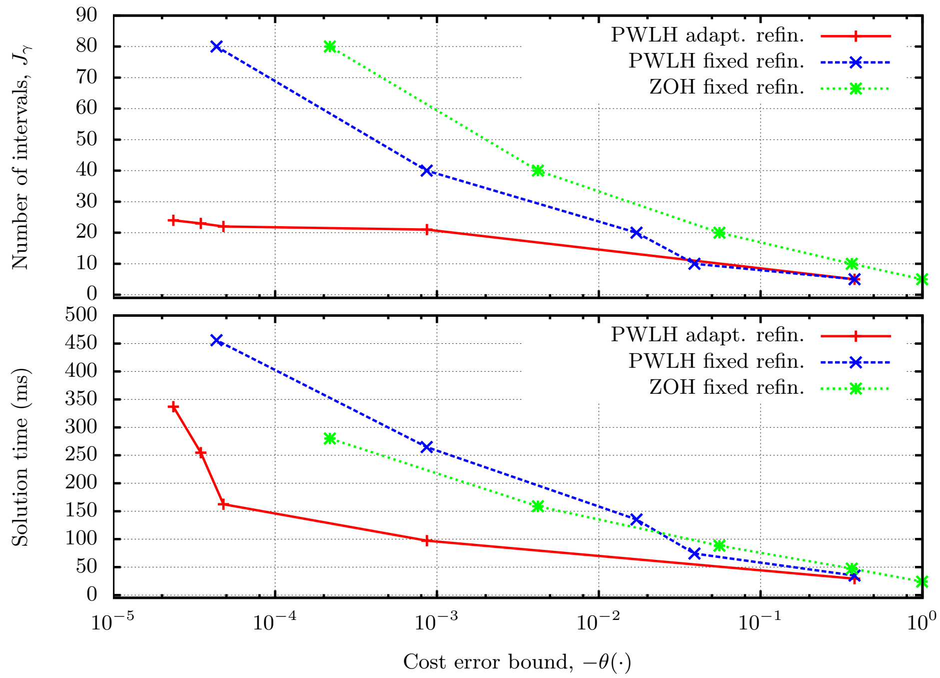

Figure 4: Example 2: Performance indicators during the first five iterations of Algorithm 8 using PWLH with adaptive discretization refinement (red), PWLH with fixed discretization refinement (blue), ZOH with fixed discretization refinement (green).

uses a horizon of $N = 240$ and hence a sampling time of $T_s = T/N = 0.25$ s. We notice, as in the first example, that CTCLQR 1, CTCLQR 2 and DTMPC 1 have (average and maximum) computation times much smaller than the sampling time. However, DTMPC 1 is rather suboptimal (3.67% on average, up to 11.7%). By decreasing the sampling time, DTMPC 2 is much less suboptimal than DTMPC 1 (0.76% on average, up to 3.04%), but DTMPC 2 has an average computation time more than twice its sampling time (almost five times in worst case). From these results it is clear than CTCLQR 1 and CTCLQR 2 provide much better results than standard discrete-time MPC algorithms.

8 CONCLUSIONS

The method presented in this paper solves the input constrained, infinite horizon, continuous time linear quadratic regulator problem to a user specified accuracy. The doubly infinite dimensional nature of the problem is addressed without undue online computation. The algorithm is efficient due to the storage of matrix exponentials for exact solution of the model, cost, and gradients at several levels of discretization. The storage requirement for these matrices is minor. The issue of *implementation* of this continuous time input in a standard industrial control system has been separated from the *numerical solution* of

Table 2: Example 2: closed-loop performance comparison of continuous time CLQR and discrete time MPC. Results are averaged over 50 closed-loop simulations of 60 s, each starting from a different random initial state.

Controller	(T_C/T_s)		$(V_{CL} - V_{CL}^0)/V_{CL}^0$	
	Mean	Max	Mean	Max
CTCLQR 1 ($\rho = 1 \cdot 10^{-4}$, $T_s = 5\,\mathrm{s}$)	0.0069	0.0288	–	–
CTCLQR 2 ($\rho = 1 \cdot 10^{-3}$, $T_s = 5\,\mathrm{s}$)	0.0061	0.0224	0.0009	0.0233
DTMPC 1 ($N = 12$, $T_s = 5\,\mathrm{s}$)	0.0033	0.0045	0.0367	0.1174
DTMPC 2 ($N = 240$, $T_s = 0.25\,\mathrm{s}$)	2.3335	4.9087	0.0076	0.0304

the CLQR problem. The small interface code required to deliver a piecewise linear input to the actuator hardware, when it is digital, can evolve as the capabilities of the actuator hardware improve; discretization effects are negligible given the relatively small sampling time of the digital actuator. With this approach, the MPC problem and solution method are independent of sample time and actuator hardware.

The advantages of *discrete time* linear MPC, and the reasons for its dominant position in industrial implementations, are mainly computational. All that is required for implementation is the solution of a strictly convex, finite dimensional quadratic program, and standard software exists for solving the QP to near machine precision in a finite number of iterations. But if one wants to implement an algorithm with a guarantee of even nominal recursive feasibility and closed-loop stability, more is required. Current theory requires a terminal penalty and an implicit or explicit terminal constraint. The terminal constraint restricts the set of feasible initial states that can be handled. One method to offset this reduction in the feasible set is to increase the horizon length. But computational cost increases at least linearly with horizon length, so this creates an unpleasant tradeoff between the size of the feasible set and computational efficiency. The same difficult tradeoff applies to the choice of sample time. If chosen too large, closed-loop performance and robustness to disturbances degrade; if chosen too small, the horizon and computation time are excessively large or the feasible set of initial states is excessively small. For online solutions as in MPC, it is often intractable to solve the infinite horizon *discrete time* problem because the input parameterization as a zero-order hold with a fixed and short sample time is inefficient. (See Figure 3, for example.)

The use of the infinite horizon, continuous time CLQR in MPC removes the need for terminal sets and terminal constraints. Of course we made liberal use of both in designing the CLQR algorithm, but those details can be hidden in the algorithm itself. The higher level controller design problem is free from these considerations. The simplicity of the resulting CLQR theory may be its most significant advantage. Nominal recursive feasibility and closed-loop stability follow directly from the optimal control problem's design. The CLQR feasible set is the largest set possible, i.e., the set of states for which there exists an input trajectory with a *finite* infinite horizon open-loop cost.

It remains to be seen whether this kind of approach can handle the largest industrial applications, which currently consist of hundreds or thousands of state variables. Because all notions of sample time are removed from the regulation problem, sample time can be chosen as a design parameter relevant to sensor hardware and robustness to disturbances, without consideration of the underlying regulation problem. This comment, of course, presumes that the regulation computation is at least as fast as the desired sampling rate. Research directed at further improving the online efficiency of solving the CLQR is therefore *always* relevant.

Finally, the issue of *identification* of the underlying linear model remains relevant to the choice of continuous versus discrete time in the regulation problem. In the identification field also, we notice the rise and fall in popularity of the time description, from continuous time parametric model identification to discrete time subspace model identification. It is not entirely clear what new challenges using continuous time linear models for MPC may present for the model identification task. From the regulation side of the problem, certainly future research should be directed towards handling multiple time delays, which change the underlying model from differential equations to delay-differential equations, a not insignificant increase in complexity.

References

[1] F. Bayat and T. A. Johansen. Multi-resolution explicit model predictive control: Delta-model formulation and approximation. *IEEE Trans. Auto. Contr.*, 2013. DOI: 10.1109/TAC.2013.2259982.

[2] T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, and O. V. Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 295–339. Springer-Verlag Berlin Heidelberg, 2001.

[3] T. Binder, L. Blank, W. Dahmen, and W. Marquardt. Multiscale concepts for moving horizon optimization. In J. Rambau M. Grötschel, S.O. Krumke, editor, *Online Optimization of Large Scale Systems*, pages 341–361. Springer-Verlag Berlin Heidelberg, 2001.

[4] A. E. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere Publishing, New York, 1975.

[5] Christof Büskens and Helmut Maurer. SQP-methods for solving optimal control problems with control and state constraints: Adjoint variables, sensitivity analysis and real-time control. *J. Comput. Appl. Math.*, 120:85–108, 2000.

[6] M. Cannon and B. Kouvaritakis. Infinite horizon predictive control of constrained linear systems. *Automatica*, 36:943–955, 2000.

[7] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Sys. Cont. Let.*, 29:121–129, 1996.

[8] V. Costanza, P. S. Rivadeneira, and A. H. González. Minimizing control energy in a class of bounded-control linear-quadratic regulator problems. *Opt. Contr. Appl. and Meth.*, 2013. DOI: 10.1002/oca.2072.

[9] G. N. Galbraith and R. B. Vinter. Lipschitz continuity of optimal controls for state constrained problems. *SIAM J. Control Optim.*, 42(5):1727–1744, 2003.

[10] R. Goebel and M. Subbotin. Continuous time linear quadratic regulator with control constraints via convex duality. *IEEE Trans. Auto. Contr.*, 52(5):886–892, 2007.

[11] R. Gondhalekar and J. Imura. Least-restrictive move-blocking model predictive control. *Automatica*, 46(7):1234–1240, 2010.

[12] William W. Hager. Runge-Kutta methods in optimal control and transformed adjoint system. *Numer. Math*, 87:247–282, 2000.

[13] A. Hartwich, M. Schlegel, L. Würth, and W. Marquardt. Adaptive control vector parameterization for nonlinear model-predictive control. *Int. J. Rob. Nonlin. Contr.*, 18(8):845–861, 2008.

[14] A. Kojima and M. Morari. LQ control of constrained continuous-time systems. *Automatica*, 40:1143–1155, 2004.

[15] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems.* John Wiley & Sons, 1972.

[16] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory.* John Wiley & Sons, New York, 1967.

[17] S. Longo, E. C. Kerrigan, K. V. Ling, and G. A. Constantinides. A parallel formulation for predictive control with nonuniform hold constraints. *Annual Reviews in Control*, 35(2):207–214, 2011.

[18] G. Pannocchia, J. B. Rawlings, D. Q. Mayne, and W. Marquardt. On computing solutions to the continuous time constrained linear quadratic regulator. *IEEE Trans. Auto. Contr.*, 55(9):2192–2198, 2010.

[19] Gabriele Pannocchia, David Q. Mayne, James B. Rawlings, and Guilio Mancuso. A parsimonious algorithm for the solution of continuous-time constrained LQR problems with guaranteed convergence. In *Proceedings of 2013 European Control Conference*, Zurich, Switzerland, 2013.

[20] Gabriele Pannocchia, James B. Rawlings, David Q. Mayne, and Giulio Mancuso. Whither discrete time model predictive control? *IEEE Trans. Auto. Contr.*, Conditionally accepted as Technical Note, 2014.

[21] E. Polak. *Optimization: Algorithms and Consistent Approximations.* Springer Verlag, 1997.

[22] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *J. Optim. Theory Appl.*, 99:723–757, 1998.

[23] J. W. C. Robinson and P. Ögren. On the use of gradual dense–sparse discretizations in receding horizon control. *Opt. Contr. Appl. and Meth.*, 2013. DOI: 10.1002/oca.2065.

[24] V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos. Explicit solutions to optimal contol problems for contrained continuous-time linear systems. *IEE Proc.-Control Theory Appl.*, 152(4):443–452, 2005.

[25] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *J. Proc. Contr.*, 21(5):715–728, 2011.

[26] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt. Dynamic optimization using adaptive control vector parameterization. *Comput. Chem. Eng.*, 29(8):1731–1751, 2005.

[27] P. O. M. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. In *13th Triennial IFAC World Congress*, pages 109–114, San Francisco, CA (USA), 1996.

[28] P. O. M. Scokaert and James B. Rawlings. Constrained linear quadratic regulation. *IEEE Trans. Auto. Contr.*, 43(8):1163–1169, 1998.

[29] C. F. Van Loan. Computing integrals involving the matrix exponential. *IEEE Trans. Auto. Contr.*, 23(3):395–404, 1978.

[30] R. B. Vinter. *Optimal Control*. Birkhaüser, Boston, 2000.

[31] J. Yuz, G. Goodwin, A. Feuer, and J. De Doná. Control of constrained linear systems using fast sampling rates. *Syst. Contr. Lett.*, 54:981–990, 2005.

A Complementary material

The following Lemma is used in the proof of Proposition 4.

Lemma 15 (An instability result)**.** *Consider an asymptotically stable linear system $\dot{x} = Ax + Bu$, a sequence of times $\{T_i\}$ with $T_i \to \infty$ as $i \to \infty$, and a sequence of measurable inputs $u_i : [0, T_i] \to \mathbb{U}$ with corresponding solutions $x_i : [0, T_i] \to \mathbb{R}^n$. Then if $|x_i(t)| \geq a > 0$ for all $t \in [0, T_i]$ and all i,*

$$\|u_i(\cdot)\|_2 \to \infty \text{ as } i \to \infty$$

Proof: Since A is Hurwitz, there exists $c > 0$ and $\lambda < 0$ such that $\left|e^{At}\right| \leq ce^{\lambda t}$ for all $t \geq 0$. Using this bound and the Cauchy-Schwartz inequality in the solution of the linear system then gives the following bound

$$|x(t)|^2 \leq c_1 e^{2\lambda t} |x(0)|^2 + c_2 \int_0^t |u(\tau)|_2^2 \, d\tau$$

with $c_1 = 2c^2$ and $c_2 = 2c^2 |B|^2 / (-2\lambda)$, where $|B|$ is the induced 2-norm of matrix B. Choose time $\Delta > 0$ such that $c_1 e^{2\lambda\Delta} = \rho < 1$. We then have

$$c_2 \left\|u_{[0,\Delta]}\right\|_2^2 \geq |x(\Delta)|^2 - \rho |x(0)|^2$$
$$c_2 \left\|u_{[\Delta,2\Delta]}\right\|_2^2 \geq |x(2\Delta)|^2 - \rho |x(\Delta)|^2$$
$$c_2 \left\|u_{[2\Delta,3\Delta]}\right\|_2^2 \geq |x(3\Delta)|^2 - \rho |x(2\Delta)|^2$$

and so on. Adding these up to time $j\Delta$ gives

$$c_2 \left\|u_{[0,j\Delta]}\right\|_2^2 \geq (1 - \rho) \sum_{k=1}^{j} |x(k\Delta)|^2 - \rho |x(0)|^2$$

If $|x_i(t)| \geq a$ for all $t \in [0, T_i]$, we have that $c_2 \|u_i(\cdot)\|_2^2 \geq a^2(1 - \rho)\lfloor T_i/\Delta \rfloor - \rho |x(0)|^2$, and therefore $\|u_i(\cdot)\|_2 \to \infty$ as $i \to \infty$. □

The following Proposition and proof were kindly supplied by R. B. Vinter:

Proposition 16. *Consider a function $L : [0, T] \times \mathbb{R}^m \to \mathbb{R}$ and set $\mathbb{U} \subset \mathbb{R}^m$ that satisfy the conditions:*

1. *$L(\cdot, v)$ is a Lebesgue measurable function for each v and $L(t, \cdot)$ is continuous for each t*
2. *$\mathbb{U}$ is compact*
3. *$L(\cdot, \cdot)$ is bounded on bounded sets*

Then:

$$\inf \left\{ \int_0^T L(t, v(t))dt \mid v(\cdot) \text{ is a selector of } \mathbb{U} \right\} = \int_0^T \min_{v \in \mathbb{U}} L(t, v)dt$$

Here 'selector' means 'measurable function $v(\cdot)$ such that $v(t) \in \mathbb{U}$ a.e. in $[0, T]$'.

Proof: Let $\hat{L}(t) \triangleq \min_{v\in\mathbb{U}} L(t,v)$. By Theorem 2.3.14 in [30], $\hat{L}(\cdot)$ is measurable. Take any selector $v(\cdot)$ of $\mathbb{U}$. Then $t \to L(t,v(t))$ is measurable and $\hat{L}(t) - L(t,v(t)) \le 0$ a.e. by definition of $\hat{L}(\cdot)$. Since, under the assumptions, $\hat{L}(\cdot)$ and $t \mapsto L(t,v(t))$ are integrable, we have

$$\int_0^T \hat{L}(t)dt - \int_0^T L(t,v(t))dt = \int_0^T \Big(\hat{L}(t)dt - L(t,v(t))\Big)\, dt \le 0.$$

But $v(\cdot)$ was chosen arbitrarily. So

$$\int_0^T \hat{L}(t)dt \;\le\; \inf\left\{\int_0^T L(t,v(t))dt \,|\, v(\cdot) \text{ is a selector of } \mathbb{U}\right\}. \tag{35}$$

On the other hand, it follows from Filippov's Generalized Selection Theorem as stated, for example, in Theorem 2.3.13 in [30] that there exists a selector $\bar{v}(\cdot)$ of $\mathbb{U}$ such that

$$L(t,\bar{v}(t)) = \hat{L}(t) \text{ a.e.}$$

(To see this, identify the scalar valued function $g(\cdot,\cdot)$ and the measurable function $v(\cdot)$ in this reference with $L(\cdot,\cdot)$ and $\hat{L}(\cdot)$ respectively). But then, since the relevant integrands are integrable functions,

$$\inf\left\{\int_0^T L(t,v(t))dt \,|\; v(\cdot) \text{ is a selector of } \mathbb{U}\right\} \;\le\; \int_0^T L(t,\bar{v}(t))dt \;=\; \int_0^T \hat{L}(t)dt. \tag{36}$$

(35) and (36) combine to give the required relationship. □