# Computation of the Infinite Horizon Continuous Time Constrained Linear Quadratic Regulator ⋆

**Gabriele Pannocchia** * **James B. Rawlings** ** **David Q. Mayne** ***
**Wolfgang Marquardt** ****

* *Dept. Chem. Eng., Ind. Chem. & Sc. Mat. – Univ. of Pisa, Italy (email: g.pannocchia@ing.unipi.it)*
** *Dept. Chem. & Biol. Eng. – Univ. of Wisconsin. Madison (WI), USA (emai: rawlings@eng.wisc.edu)*
*** *Dept. Elec. & Electronic Eng. – Imperial College London, UK (email: d.mayne@imperial.ac.uk)*
**** *AVT–Process Systems Engineering, RWTH Aachen University, Germany (email: Wolfgang.marquardt@avt.rwth-aachen.de)*

**Abstract:** We present a method for computing the solution to the infinite horizon continuous-time constrained linear quadratic regulator (CLQR). The method relies on two main features: a multi-grid method for placing a finite number of time intervals, and a piece-wise linear parameterization of the input within the intervals. The input values at the grid points and slopes within the time intervals are computed via quadratic programs (QPs). The grids are gradually refined to efficiently improve the accuracy of the solution, and the required matrices and vectors for all QPs are computed offline and stored to improve the online efficiency. We present two examples, a single-input single-output unstable system and a three-input three-output stable system, to show the main characteristics of the proposed computation method.

*Keywords:* Constrained Linear Quadratic Regulation, Continuous Time Systems, Model Predictive Control, Optimal Control

## 1. MOTIVATIONS FOR CONTINUOUS TIME MODEL PREDICTIVE CONTROL

The scope of applications of model predictive control (MPC) has expanded well beyond its original starting point in the process industries. With this increased scope has come the need to evaluate in real time the solution to the MPC optimal control problem for systems with fast open-loop dynamics. It is reasonable to anticipate that this trend to faster applications may culminate with a return to the continuous time description of the system model. The previous widespread adoption of discrete time models to represent the system dynamics made perfect sense. The typical sample time in earlier applications was small compared to the closed-loop dynamic response of the system (seconds compared to minutes) so there was essentially no loss in model accuracy. Moreover, the earlier analysis of the closed-loop properties and computational strategies to approximate infinite horizon control laws was simplified using discrete time models (Mayne et al., 2000).

In today's application environment, it is no longer safe to assume that some fixed sample rate can be chosen very small compared to the closed-loop system dynamics. Next, given the rapid development of MPC theory for discrete time models over the last 15 years, there is no real difficulty in establishing properties of interest in a continuous time setting. Finally, the actuator hardware has become "smarter" and it is now becoming appropriate to assume that any reasonable time function may be sent to the actuator, and it is actuator hardware's job to accurately track this signal. If the application has fast dynamics, obviously a requirement of the process design is the selection of sensors and actuators that are fast enough to keep up.

In this paper we would like to remove all issues of sampling and address directly the MPC problem for the continuous-time model (Yuz et al., 2005). The job of the MPC controller in this context is to send its solution as a time signal to the actuators until a measurement becomes available and a new state initial condition is available to the controller. This context has become popular in the nonlinear MPC area, where nonlinear models from physical principles are almost always continuous time nonlinear differential equations [see (Diehl et al., 2008) and references therein]. In this paper we would like to explore what efficiencies can be gained when we restrict attention to *linear* continuous time models.

## 2. PROBLEM DEFINITION

We consider linear time-invariant continuous-time systems

$$\dot{x} = Ax + Bu\,, \tag{1}$$

in which $x \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}^m$ is the input. We define the following cost function for a given initial state $x_0 \in \mathbb{R}^n$ and infinite-time input $\mathbf{u}$:

$$V(x_0, \mathbf{u}) = \frac{1}{2}\int_0^\infty (x'Qx + u'Ru)\,dt, \quad \text{s.t. (1) and } x(0) = x_0\,. \tag{2}$$

in which we use the following notation. Given a function $u : \overline{\mathbb{R}} \to \mathbb{R}^m$, we define $\mathbf{u} = \{u(t) | t \geq 0\}$. The aim of this work is to compute, given the current initial state $x_0$, the solution of the

⋆ This research was supported by National Science Foundation (Grant CTS-0456694).

following infinite horizon optimal constrained linear quadratic regulation (CLQR) problem

$$\mathbf{u}^*(x_0) = \underset{\mathbf{u}}{\operatorname{argmin}}\ V(x_0, \mathbf{u}), \quad \text{s.t. } Du(t) \leq d\,, \qquad (3)$$

in which $D = \begin{bmatrix} D_1' & D_2' \end{bmatrix}'$ and $d = \begin{bmatrix} d_1' & d_2' \end{bmatrix}'$ with $d_1 > 0$ and $d_2 = 0$ (in element-wise sense). Notice that we allow the possibility of either $(D_1, d_1)$ or $(D_2, d_2)$ being empty.

We make the following assumptions.

*Assumption 1.* Given a matrix $T \in \mathbb{R}^{m \times r}$ with rank $r$ such that $D_2 T = 0$, the pair $(A, BT)$ is stabilizable, $R$ is positive definite, $Q$ is positive semi-definite and $(A, Q)$ is detectable.

It is important to point out that the constraint $D_2 u \leq 0$ is active at the equilibrium point ($u = 0$). Assumption 1 states that the system must be stabilizable under the restricted control $D_2 u = 0$. We can write the restricted optimal control as $u = \bar{K}x = -(T'RT)^{-1}(BT)'\bar{P}x$, with $\bar{P}$ solution of the continuous-time Riccati equation for the system matrices $(A, BT)$ with penalties $(Q, T'RT)$ [see (Rao and Rawlings, 1999; Pannocchia et al., 2003) for further details]. Such control is feasible for all $x$ in the positively invariant set $\mathcal{X} = \{x | D_1 \bar{K} e^{(A+B\bar{K})t} x \leq d_1 \text{ for all } t\}$.

Compared to the discrete-time counterpart, the continuous-time CLQR problem has received much less attention, although several results are available. Cannon and Kouvaritakis (2000) proposed a method for single-input single-output systems, using basis functions, in which input constraint satisfaction is ensured by a backoff strategy. Kojima and Morari (2004) propose a design method that is based on the singular value decomposition (SVD) of the finite horizon linear system and that guarantees in the limit constraint satisfaction and convergence to the optimal solution. For low dimensional linear systems, Sakizlis et al. (2005) present an approach for computing the explicit solution to the finite horizon continuous-time CLQR problem, by merging variational analysis with parametric optimization tools. Goebel and Subbotin (2007) present an approach based on the solution of the backward Hamiltonian system; optimal trajectories are stored for subsequent on-line suboptimal evaluation.

In this work, we propose a novel approach based on the direct evaluation of a suboptimal solution to the infinite horizon CLQR problem that requires, like the discrete-time case, the solution of only quadratic programs (QPs). Thus, we compute the control input in a number of grid points that can be efficiently adapted on-line to improve the accuracy of the solution. Two appropriate continuous-time input parameterizations (holds) are proposed, which show markedly improved convergence towards the optimal continuous-time solution compared to the usual piece-wise constant parameterization, and still guarantees satisfaction of the input constraints. Other related details, including proofs of the results, can be found in (Pannocchia et al., 2009).

## 3. INPUT PARAMETERIZATION, EXACT COST EVALUATION AND CONVERGENCE ANALYSIS

We consider three different input parameterization methods (also referred to as holds), and for each hold type a discrete-time system realization is obtained in a way that the continuous-time and the discrete-time state and input match at given grid points $(t_0, t_1, \ldots)$ assumed, in general, not evenly spaced. Moreover,

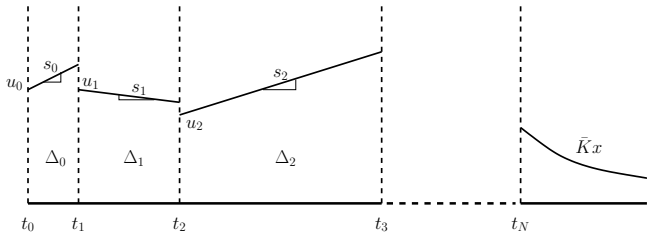


Fig. 1. Piece-wise linear input parameterization and time gridding scheme.

for each hold type we compute the corresponding discrete-time LQR cost matrices in a way that the cost in (2) is exactly evaluated. We recall that the solution to (1) is given by:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau\,. \qquad (4)$$

### 3.1 Three input parameterization methods

First we summarize the properties of the well known zero-order hold (ZOH), defined as:

$$u(t) = u_k, \qquad t_k \leq t < t_{k+1}\,, \qquad (5)$$

with $t_{k+1} = t_k + \Delta_k$. Notice that we do not assume, however, that $\Delta_k$ is constant. From (4) and (5), we obtain the time-varying discrete-time system: $x_{k+1} = A_k^0 x_k + B_k^0 u_k$, in which the matrices $(A_k^0, B_k^0)$ are defined as: $A_k^0 = e^{A\Delta_k}$, $B_k^0 = I_0(\Delta_k)B$, with $I_0(t) = \int_0^t e^{A\tau}d\tau$.

We consider a second input parameterization referred to as "piece-wise linear" hold (PWLH), and defined as:

$$u(t) = u_k + s_k(t - t_k), \qquad t_k \leq t < t_{k+1}\,, \qquad (6)$$

in which $s \in \mathbb{R}^m$ defines the "slope" of $u$ between grid points. In the sake of clarity, we sketch the PWLH input parameterization in Figure 1, in which we also emphasize the uneven time gridding scheme that we consider later in Section 4. From (4) and (6), the time-varying discrete-time system $x_{k+1} = A_k^{\mathrm{I}} x_k + B_k^{\mathrm{I}} u_k^{\mathrm{I}}$ is obtained, in which $u^{\mathrm{I}} \in \mathbb{R}^{2m} = \begin{bmatrix} u' & s' \end{bmatrix}'$ is the augmented input, and the matrices $(A_k^{\mathrm{I}}, B_k^{\mathrm{I}})$ are: $A_k^{\mathrm{I}} = A_k^0 = e^{A\Delta_k}$, $B_k^{\mathrm{I}} = [I_0(\Delta_k)B \quad I_1(\Delta_k)B]$, with $I_0(t) = \int_0^t e^{A\tau}d\tau$ and $I_1(t) = \int_0^t e^{A(t-\tau)}\tau d\tau$.

Finally, we consider a third input parameterization, which also assumes that the input varies linearly between discrete times, but with a slope equal to forward finite input difference, i.e.

$$u(t) = u_k + \left(\frac{u_{k+1} - u_k}{\Delta_k}\right)(t - t_k), \qquad t_k \leq t < t_{k+1}\,. \qquad (7)$$

In the sequel, this input parameterization is referred to as "forward first-order" hold (FFOH). From (4) and (7), we obtain the time-varying discrete-time system: $x_{k+1}^{\mathrm{II}} = A_k^{\mathrm{II}} x_k^{\mathrm{II}} + B_k^{\mathrm{II}} u_k^{\mathrm{II}}$, in which the augmented state $x^{\mathrm{II}} \in \mathbb{R}^{n+m}$, the shifted input $u^{\mathrm{II}} \in \mathbb{R}^m$, and $(A_k^{\mathrm{II}}, B_k^{\mathrm{II}})$ are:

$$x_k^{\mathrm{II}} = \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \quad u_k^{\mathrm{II}} = u_{k+1},$$

$$A_k^{\mathrm{II}} = \begin{bmatrix} A_k^0 & I_0(\Delta_k)B - \dfrac{I_1(\Delta_k)}{\Delta_k}B \\ 0 & 0 \end{bmatrix}, \quad B_k^{\mathrm{II}} = \begin{bmatrix} \dfrac{I_1(\Delta_k)}{\Delta_k}B \\ I \end{bmatrix}.$$

We emphasize that ZOH generates a $u(t)$ constant between discrete times and discontinuous at the discrete times, PWLH generates a $u(t)$ linear between discrete times and discontinuous at the discrete times, FFOH generates a $u(t)$ continuous at all times and linear between discrete times.

### 3.2 Exact continuous time cost computation

*Lemma 2.* (Exact cost matrices for ZOH). If the continuous-time input is given by (5), then

$$\int_{t_k}^{t_{k+1}} (x'Qx + u'Ru)dt = x_k'Q_k^0 x_k + u_k'R_k^0 u_k + 2x_k'M_k^0 u_k \ ,$$

with: $Q_k^0 = \int_0^{\Delta_k} (e^{At})'Qe^{At}dt$, $R_k^0 = \int_0^{\Delta_k} (R + (I_0B)'QI_0B)\, dt$, $M_k^0 = \int_0^{\Delta_k} (e^{At})'QI_0Bdt$.

It is interesting to notice that this quadrature result is known (Kwakernaak and Sivan, 1972, p.549), (Yuz et al., 2005, Sec.2.2), but most of the literature on optimal control of continuous-time systems usually ignores the mixed state-input cost term and assumes $Q_k^0 = Q\Delta_k$, $R_k^0 = R\Delta_k$, thus introducing an inherent quadrature error.

From the previous result it immediately follows that, given an infinite discrete-time input sequence $(u_0, u_1, u_2, \ldots)$, assuming that the continuous-time input $\mathbf{u}$ is defined in (5), then

$$V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{\infty} x_k'Q_k^0 x_k + u_k'R_k^0 u_k + 2x_k'M_k^0 u_k \ .$$

*Lemma 3.* (Exact cost matrices for PWLH). If the continuous-time input is given by (6), then $\int_{t_k}^{t_{k+1}} (x'Qx + u'Ru)dt = x_k'Q_k^{\mathrm{I}} x_k + (u_k^{\mathrm{I}})'R_k^{\mathrm{I}} u_k^{\mathrm{I}} + 2x_k'M_k^{\mathrm{I}} u_k^{\mathrm{I}}$, with:
$Q_k^{\mathrm{I}} = \int_0^{\Delta_k} (e^{At})'Qe^{At}dt$,
$R_k^{\mathrm{I}} = \int_0^{\Delta_k} \begin{bmatrix} R + (I_0B)'Q(I_0B) & (I_0B)'Q(I_1B) + Rt \\ (I_1B)'Q(I_0B) + Rt & (I_1B)'Q(I_1B) + Rt^2 \end{bmatrix} dt$,
$M_k^{\mathrm{I}} = \int_0^{\Delta_k} \left[(e^{At})'Q(I_0B)\ \ (e^{At})'Q(I_1B)\right] dt$.

From this results it follows that, given infinite discrete-time input sequence and slope sequence $(u_0, u_1, u_2, \ldots)$, $(s_0, s_1, s_2, \ldots)$, assuming that the continuous-time input $\mathbf{u}$ is defined in (6), then

$$V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{\infty} x_k'Q_k^{\mathrm{I}} x_k + (u_k^{\mathrm{I}})'R_k^{\mathrm{I}} u_k^{\mathrm{I}} + 2x_k'M_k^{\mathrm{I}} u_k^{\mathrm{I}} \ .$$

*Lemma 4.* (Exact cost matrices for FFOH). If the continuous-time input is given by (7), then $\int_{t_k}^{t_{k+1}} (x'Qx + u'Ru)dt = (x_k^{\mathrm{II}})'Q_k^{\mathrm{II}} x_k^{\mathrm{II}} + (u_k^{\mathrm{II}})'R_k^{\mathrm{II}} u_k^{\mathrm{II}} + 2(x_k^{\mathrm{II}})'M_k^{\mathrm{II}} u_k^{\mathrm{II}}$, with:
$Q_k^{\mathrm{II}} = \int_0^{\Delta_k} \begin{bmatrix} (e^{At})'Qe^{At} & (e^{At})'Q(I_0 - \frac{I_1}{\Delta_k})B \\ ((I_0 - \frac{I_1}{\Delta_k})B)'Qe^{At} & ((I_0 - \frac{I_1}{\Delta_k})B)'Q(I_0 - \frac{I_1}{\Delta_k})B + R\left(1 - \frac{t}{\Delta_k}\right)^2 \end{bmatrix} dt$,
$R_k^{\mathrm{II}} = \int_0^{\Delta_k} \left(R\left(\frac{t}{\Delta_k}\right)^2 + \left(\frac{I_1}{\Delta_k}B\right)'Q\left(\frac{I_1}{\Delta_k}B\right)\right) dt$ ,
$M_k^{\mathrm{II}} = \int_0^{\Delta_k} \begin{bmatrix} (e^{At})'Q\frac{I_1}{\Delta_k}B \\ ((I_0 - \frac{I_1}{\Delta_k})B)'Q\frac{I_1}{\Delta_k}B + R\frac{t}{\Delta_k}\left(1 - \frac{t}{\Delta_k}\right) \end{bmatrix} dt$.

Clearly, it follows that given an infinite discrete-time input sequence $(u_0, u_1, u_2, \ldots)$, assuming that the continuous-time input $\mathbf{u}$ is defined in (7), then

$$V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{\infty} (x_k^{\mathrm{II}})'Q_k^{\mathrm{II}} x_k^{\mathrm{II}} + (u_k^{\mathrm{II}})'R_k^{\mathrm{II}} u_k^{\mathrm{II}} + 2(x_k^{\mathrm{II}})'M_k^{\mathrm{II}} u_k^{\mathrm{II}} \ .$$

### 3.3 Unconstrained convergence analysis for the three holds

In this section, we evaluate the unconstrained optimal cost that is achieved by using the three different input parameterizations, in the case of evenly spaced points, i.e. $\Delta_0 = \Delta_1 = \cdots = \Delta$, and we compare the order of convergence to the optimal continuous-time cost as $\Delta$ goes to zero. [1] We first recall the following well-known results for unconstrained LQR problems.

*Lemma 5.* The optimal cost-function value for the unconstrained continuous-time LQR problem $\min_{\mathbf{u}} V(x_0, \mathbf{u})$ is given by $\frac{1}{2}x_0'Px_0$ in which $P$ is the positive semi-definite solution of the Riccati equation:

$$0 = Q + A'P + PA - PBR^{-1}B'P \ . \tag{8}$$

*Lemma 6.* The following discrete-time LQR problem with mixed state-input terms:

$$\min_{(u_0,u_1,\ldots)} \frac{1}{2}\sum_{k=0}^{\infty} x_k'\bar{Q}x_k + u_k'\bar{R}u_k + 2x_k'\bar{M}u_k \ , \quad \text{s.t.}$$
$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \ ,$$

is equivalent to the following discrete-time LQR problem without mixed state-input terms:

$$\min_{(u_0,u_1,\ldots)} \frac{1}{2}\sum_{k=0}^{\infty} x_k'\bar{Q}x_k + u_k'\bar{R}u_k, \quad \text{s.t.}$$
$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \ , \tag{9}$$

in which the following change of variables is considered: $u_k \leftarrow u_k - \bar{R}^{-1}\bar{M}'x_k$, $\bar{A} \leftarrow \bar{A} - \bar{B}\bar{R}^{-1}\bar{M}'$, $\bar{Q} \leftarrow \bar{Q} - \bar{M}\bar{R}^{-1}\bar{M}'$.

*Lemma 7.* The optimal cost-function value for the unconstrained discrete-time LQR problem (9) is $\frac{1}{2}x_0'\Pi x_0$, in which $\Pi$ is the positive semi-definite solution of the Riccati equation:

$$0 = -\Pi + \bar{Q} + \bar{A}'\Pi\bar{A} - \bar{A}'\Pi\bar{B}(\bar{R} + \bar{B}'\Pi\bar{B})^{-1}\bar{B}\Pi\bar{A} \ . \tag{10}$$

If $P$ is the solution of the continuous-time Riccati equation (8) and $\Pi(\Delta)$ is the solution of the discrete-time Riccati equation (10) using a given hold and a fixed discrete-time interval $\Delta$, it is straightforward to show that $P \preccurlyeq \Pi(\Delta)$, which is equivalent to saying $\Pi(\Delta) - P$ is positive semidefinite. Clearly, it is desirable for an input parameterization to have convergence $\Pi(\Delta) \to P$ as $\Delta \to 0$. We define the order of convergence $\ell$ for a given hold implementation as the smallest non-negative integer for which $\lim_{\Delta\to 0} \frac{\Pi(\Delta)-P}{\Delta^\ell} \neq 0$.

We next establish the following results about the convergence order the LQR cost using different holds [2] . Notice that each hold defines a discrete-time optimal control problem, in which the decision variables are $\{u_k\}_{k=0}^{\infty}$ for ZOH and FFOH, and are $\{(u_k, s_k)\}_{k=0}^{\infty}$ for PWLH.

*Theorem 8.* (Second order convergence of ZOH). The convergence of $\Pi^0(\Delta)$ to $P$ is second order for system matrices $\bar{A} = A^0 - B^0(R^0)^{-1}(M^0)'$, $\bar{B} = B^0$ and cost matrices $\bar{Q} = Q^0 - M^0(R^0)^{-1}(M^0)'$, $\bar{R} = R^0$.

It is interesting to notice that if inexact cost matrices are used in ZOH, the order of convergence is less than 2. For instance, if one simply chooses $Q^0 = Q\Delta$, $R^0 = R\Delta$, $M^0 = 0$, the convergence order is 1.

*Theorem 9.* (Fourth order convergence of PWLH). The convergence of $\Pi^{\mathrm{I}}(\Delta)$ to $P$ is fourth order for system matrices $\bar{A} = A^{\mathrm{I}} - B^{\mathrm{I}}(R^{\mathrm{I}})^{-1}(M^{\mathrm{I}})'$, $\bar{B} = B^{\mathrm{I}}$ and cost matrices $\bar{Q} = Q^{\mathrm{I}} - M^{\mathrm{I}}(R^{\mathrm{I}})^{-1}(M^{\mathrm{I}})'$, $\bar{R} = R^{\mathrm{I}}$.

Before presenting the convergence result for FFOH, it is important to recall that, in system (3.1), the state is aug-

[1] The proof for ZOH is reported in (Pannocchia et al., 2009), and follows Taylor expansions of all terms in the discrete algebraic Riccati equation. For PWLH and FFOH, symbolic manipulation software may be useful.

[2] Since the time interval is fixed, all the discrete-time matrices are time-invariant. Thus, we drop the subscript $k$ in this section.

mented. Thus, the corresponding solution of (10) is in the form $\Pi = \begin{bmatrix} \Pi_{xx} & \Pi_{xu} \\ \Pi'_{xu} & \Pi_{uu} \end{bmatrix}$, and, hence, the unconstrained cost for any given initial state $x_0$ is given by $\frac{1}{2}(x_0'\Pi_{xx}x_0 + 2x_0'\Pi_{xu}u_0 + u_0'\Pi_{uu}u_0)$. Since the input $u_0$ is a decision variable, the optimal unconstrained cost for any given initial state $x_0$ using FFOH is easily obtained as $\frac{1}{2}x_0'(\Pi_{xx} - \Pi_{xu}\Pi_{uu}^{-1}\Pi'_{xu})x_0$.

*Theorem 10.* (Fourth order convergence of FFOH). The convergence of $\Pi^{\rm II}(\Delta) = \Pi_{xx} - \Pi_{xu}\Pi_{uu}^{-1}\Pi'_{xu}$ to $P$ is fourth order for system matrices $\bar{A} = A^{\rm II} - B^{\rm II}(R^{\rm II})^{-1}(M^{\rm II})'$, $\bar{B} = B^{\rm II}$ and cost matrices $\bar{Q} = Q^{\rm II} - M^{\rm II}(R^{\rm II})^{-1}(M^{\rm II})'$, $\bar{R} = R^{\rm II}$.

We now show how the optimal discrete-time cost matrices for different holds are "ordered".

*Theorem 11.* (Cost comparison). The following linear matrix inequalities hold: $\Pi^{\rm I}(\Delta) \preccurlyeq \Pi^{0}(\Delta)$, $\Pi^{\rm I}(\Delta) \preccurlyeq \Pi^{\rm II}(\Delta)$.

The reader may naturally expect the following ordering also to hold, $\Pi^{\rm II}(\Delta) \preccurlyeq \Pi^{0}(\Delta)$, but this is not valid for arbitrary $\Delta$. The discontinuities allowed in ZOH may provide better performance than the continuous FFOH for large $\Delta$. Of course, due to the different convergence orders, the ordering does hold for sufficiently small $\Delta$.

## 4. ALGORITHM FOR COMPUTATION OF THE CONTINUOUS-TIME CLQR

### *4.1 Introduction and main definitions*

Motivated by the nice convergence results of the PWLH and FFOH input parameterizations, we propose computing a suboptimal solution to problem (3) in terms of an appropriate finite number of decision variables, namely the inputs $(u_0, \ldots, u_{N-1})$, and for the PWLH case also the slopes $(s_0, \ldots, s_{N-1})$. As shown in this section, we can write the infinite-horizon continuous-time CLQR problem as a finite dimensional Quadratic Program, whose complexity is the same as that of the discrete-time CLQR problem. Furthermore, we define a procedure for placing the grid points $(t_0, \ldots, t_N)$ in a way that the number of decision variables is kept small while the accuracy of the solution is improved.

Let $(t_0, \ldots, t_N)$ be a sequence of $N+1$ grid points with $t_0 = 0$, and consider the following suboptimal CLQR problems:

$$\mathbf{u}^{\rm I} = \underset{\mathbf{u}}{\operatorname{argmin}}\ V(x_0, \mathbf{u}) \quad \text{s.t. } Du(t) \le d, \text{ (6), and } D_2u(t) = 0, \quad \text{for } t \ge t_N\,, \quad (11)$$

$$\mathbf{u}^{\rm II} = \underset{\mathbf{u}}{\operatorname{argmin}}\ V(x_0, \mathbf{u}) \quad \text{s.t. } Du(t) \le d, \text{ (7), and } D_2u(t) = 0, \quad \text{for } t \ge t_N\,, \quad (12)$$

From what presented so far, we can rewrite the problem (11) as follows:

$$\min_{(u_0^{\rm I}, u_1^{\rm I}, \ldots, u_{N-1}^{\rm I})} \frac{1}{2}x_N'P^{\rm I}x_N + \frac{1}{2}\sum_{k=0}^{N-1} x_k'Q_k^{\rm I}x_k + (u_k^{\rm I})'R_k^{\rm I}u_k^{\rm I} + 2x_k'M_k^{\rm I}u_k^{\rm I}, \quad \text{s.t.} \quad (13a)$$

$$x_{k+1} = A_k^{\rm I}x_k + B_k^{\rm I}u_k^{\rm I},\ x_0 = x(0),\ \begin{bmatrix} D & 0 \\ D & D\Delta_k \end{bmatrix} u_k^{\rm I} \le \begin{bmatrix} d \\ d \end{bmatrix}\,, \quad (13b)$$

in which $P^{\rm I} = \bar{P}$, provided that $x_N \in \mathcal{X}$.

Similarly, the problem (12) can be rewritten as follows:

$$\min_{(x_0^{\rm II}, u_0^{\rm II}, u_1^{\rm II}, \ldots, u_{N-1}^{\rm II})} \frac{1}{2}(x_N^{\rm II})'P^{\rm II}x_N^{\rm II} + \frac{1}{2}\sum_{k=0}^{N-1} (x_k^{\rm II})'Q_k^{\rm II}x_k^{\rm II} + (u_k^{\rm II})'R_k^{\rm II}u_k^{\rm II} + 2(x_k^{\rm II})'M_k^{\rm II}u_k^{\rm II}, \quad \text{s.t.} \quad (14a)$$

$$x_{k+1}^{\rm II} = A_k^{\rm II}x_k^{\rm II} + B_k^{\rm II}u_k^{\rm II},\ \mathcal{I}x_0^{\rm II} = x(0),\ D[0,\ I]x_0^{\rm II} \le d, \quad Du_k^{\rm II} \le d\,, \quad (14b)$$

in which $P^{\rm II} = \mathcal{I}'\bar{P}\mathcal{I}$ and $\mathcal{I} = [I,\ 0]$, provided that $\mathcal{I}x_N^{\rm II} \in \mathcal{X}$.

After elimination of the state variables using (13b), problem (13) can be written as a convex QP in $2mN$ decision variables. Similarly, using (14b), the problem (14) can be written as a convex QP in $m(N+1)$ decision variables. Notice that $x_0^{\rm II} = [x_0',\ u_0']'$ is a decision variable in (14), but the initial constraint in (14b) implies that only $u_0$ is a free variable.

Notice that if $V^{\rm I}(x_0) = V(x_0, \mathbf{u}^{\rm I})$ and $V^{\rm II}(x_0) = V(x_0, \mathbf{u}^{\rm II})$ denote the optimal cost-function values for the problems (11) and (12), respectively, obtained with the same grid points $(t_0, \ldots, t_N)$, then it follows that $V^{\rm II}(x_0) \ge V^{\rm I}(x_0) \ge V^*(x_0)$.

### *4.2 Offline and online computations*

The degree of suboptimality of the solution depends on the number of grid points, where it is clear that a larger number of intervals would result in a more accurate solution. However, it is important to observe that, in order to improve the solution, more grid points need to be placed where input and states are far away from the origin, whereas when they approach the origin, many intervals are unnecessary. During this work we tested several strategies for deciding the number and size of the intervals, with two desired goals in mind: (i) at each iteration the solution accuracy is improved, i.e. the computed objective function is decreased; (ii) the total number of intervals, and hence the number of QP decision variables, is kept small. We next present the simplest algorithm that is used offline to generate the QP associated with (13).

*Algorithm 1.* Data: maximum time $t_N$, initial number of intervals $\Theta$, number of halving loops $\Theta_s$.

(1) Compute the $\Theta$ intervals $(\Delta_0, \ldots, \Delta_{\Theta-1})$ such that $\Delta_k/\Delta_{k+1} = 0.5$ and $\sum_{k=0}^{\Theta-1}\Delta_k = t_N$.
(2) Define the initial sequence of $N_1 = \Theta$ intervals as $\mathcal{P}_1 = (\Delta_0^1, \ldots, \Delta_{N_1-1}^1) = (\Delta_0, \ldots, \Delta_{\Theta-1})$.
(3) Set $j \leftarrow j+1$ and define the next sequence $\mathcal{P}_j$ of $N_j = 2N_{j-1}$ intervals by halving each interval of $\mathcal{P}_{j-1}$.
(4) If $j < \Theta_s + 1$ go to 3. Otherwise, for each interval sequence $\mathcal{P}_j$ with $j = 1, \ldots, \Theta_s+1$, compute the matrices $(A_k^{\rm I}, B_k^{\rm I}, Q_k^{\rm I}, R_K^{\rm I}, M_k^{\rm I})$ in (13), build the associated QP in the form:

$$\mathbb{QP}_j: \qquad \min_{\mathbf{v}} \frac{1}{2}\mathbf{v}'\mathbf{H}\mathbf{v} + \mathbf{v}'\mathbf{Q}x_0, \quad \text{s.t. } \mathbf{A}\mathbf{v} \le \mathbf{b} \quad (15)$$

with $\mathbf{v} = (u_0^{\rm I}, \ldots, u_{N_j-1}^{\rm I})$, storing $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{A}$, $\mathbf{b}$.

For open-loop unstable systems, in order to avoid ill-conditioning of $\mathbf{H}$, the input variable re-parameterization discussed in (Rossiter et al., 1998) is recommended.

The next algorithm describes the operations that are performed online to compute $\mathbf{u}^I(x_0)$.

*Algorithm 2.* Given $x_0$, the relative cost decrease tolerance $\mu > 0$. Initialize $j = 1$.

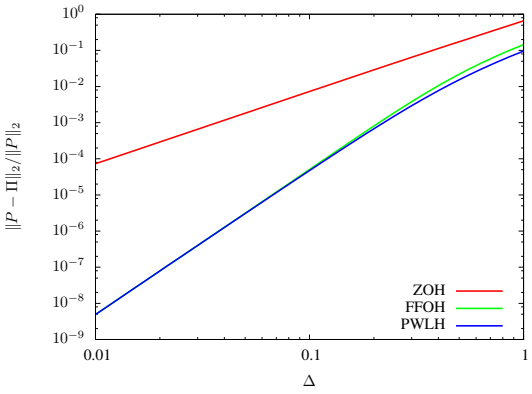


Fig. 2. System 1: Relative error between $P$ and discrete-time cost $\Pi$ for ZOH, FFOH, PWLH *vs* $\Delta$.

(1) Solve $\mathbb{QP}_j$, let $V_j^{\mathrm{I}}(x_0)$ be the associated optimal cost and $\mathbf{u}_j^{\mathrm{I}}(x_0)$ the optimal input.
(2) If $j = 1$, set $j \leftarrow j+1$ and go to 1. Otherwise,
(3) If the relative cost decrease satisfies $\frac{V_{j-1}^{\mathrm{I}}(x_0) - V_j^{\mathrm{I}}(x_0)}{V_{j-1}^{\mathrm{I}}(x_0)} < \mu$ or $j = \Theta_s + 1$, set $\mathbf{u}^{\mathrm{I}}(x_0) = \mathbf{u}_j^{\mathrm{I}}(x_0)$ and stop. Otherwise, set $j \leftarrow j+1$ and go to 1.

Notice that if at any iteration $j$, the solution to $\mathbb{QP}_j$ is such that $x_N \notin \mathcal{X}$, then an additional interval of the largest size is added, i.e. $t_N \leftarrow t_N + \Delta_{N-1}$ and $\mathbb{QP}_j$ is solved again $x_N \notin \mathcal{X}$. Such additional intervals are retained also for subsequent iterations. For efficient online computation it is advised that the matrices associated with each $\mathbb{QP}_j$ are built and stored for different increasing $t_N$, so that the online CPU time is required only for solving the QPs.

We have the following important result.

*Theorem 12.* For each iteration $j > 1$ of Algorithm 2, we have that:

$$V_j^{\mathrm{I}}(x_0) \leq V_{j-1}^{\mathrm{I}}(x_0), \quad \text{for all } j > 0\,. \tag{16}$$

It is important to remark that the same cost decrease property holds true if Algorithm 2 is applied to compute $\mathbf{u}^{\mathrm{II}}(x_0)$.

## 5. CASE STUDIES

To illustrate the main features of the proposed method, we present two examples. The first example is a SISO unstable system, whose transfer function is shown below:

$$g_1(s) = -\frac{6.512s + 1.628}{-2.4390s^2 + 3.9756s + 1}\,.$$

The second example is the 3 input, 3 output Shell Control Problem (Prett and Morari, 1987), for which we use a 10 state continuous-time model. For both examples we use $Q = I$, $R = 0.1I$.

In Figure 2 we show, for the first example, the relative error (evaluated using the 2-norm) between the continuous-time LQR cost matrix $P$ and the corresponding discrete-time cost matrix $\Pi$ obtained with ZOH, FFOH, PWLH as a function of $\Delta$. As expected from Theorems 8, 9, 10, the orders of convergence for ZOH is 2, whereas it is 4 for PWLH and FFOH.

Unless otherwise specified, in the subsequent studies we consider input constraints $-1 \leq u \leq 1$, the optimal input $\mathbf{u}^{\mathrm{II}}(x_0)$ is computed with Algorithm 2 using FFOH input parameterization and using a relative tolerance of $\mu = 10^{-4}$. For

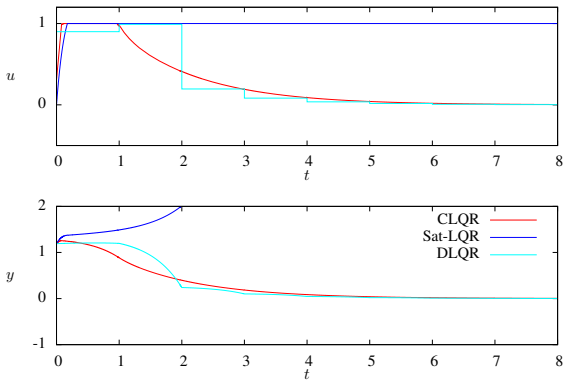


Fig. 3. System 1. Input and output closed-loop response using CLQR, "saturated" LQR and DLQR.

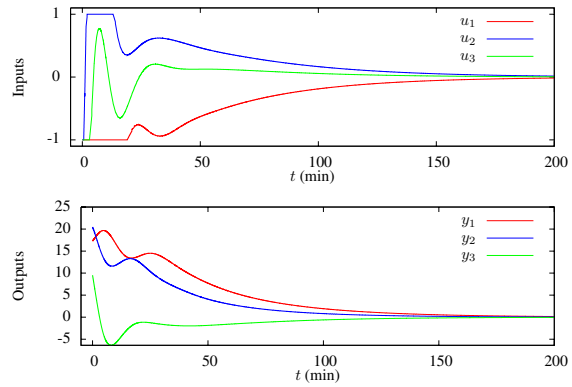


Fig. 4. System 2. Closed-loop inputs and outputs using CLQR at decision times $0, 1, \ldots$.

System 1 we show in the top plot of Figure 3 the optimal closed-loop input and output, computed by solving the optimal control problem with the proposed algorithm at decision times $0, 1, \ldots, 7$, and implementing the computed infinite horizon input in a receding horizon fashion. For comparison, we also show the results obtained by: (i) infinite horizon discrete-time constrained LQR (DLQR), (ii) "saturated" continuous-time LQR law $u = \mathrm{sat}\,(Kx)$. We observe that DLQR generates a stable closed-loop response that is fairly different from the optimal one obtained with CLQR. We also note that $Kx_0$ is feasible, but nonetheless the saturated LQR makes the closed-loop system unstable, whereas CLQR and DLQR stabilize the system in closed-loop (due to the infinite horizon formulation). Inputs and outputs for System 2 computed with constrained LQR at the decision times $0, 1, \ldots, 200$ are shown in Figure 4.

Finally, for System 1, we report in Table 1 the relative cost error, the number of required intervals, and CPU-time for solving all the QPs[3] for different values of $\mu$. In computing the relative cost error, we approximate $V^*(x)$ with the value obtained using PWLH-CLQR with $\mu = 10^{-8}$. The data reported in Table 1 refer to the computation of $\mathbf{u}(x_0)$ for the same initial state considered in Figure 3. We report in Table 2 the same computational study for System 2. We can observe that, when the number of intervals is the same, PWLH-CLQR achieves a slightly lower cost than FFOH-CLQR. However, this comes at the expense of a higher CPU time because PWLH-CLQR

[3] Using GNU Octave on an AMD Athlon$^{\mathrm{TM}}$ 64 X2 Dual Core Processor 4400+ running Debian Linux.

Table 1. System 1. Comparison of cost relative error, number of intervals of the final QP and overall CPU-time for solving all QPs vs. relative tolerance $\mu$ for PWLH-CLQR and FFOH-CLQR

| | PWLH-CLQR Algorithm | | | FFOH-CLQR Algorithm | | |
|---|---|---|---|---|---|---|
| $\mu$ | $\frac{V^{\mathrm{I}}(x_0)-V^*(x_0)}{V^*(x_0)}$ | $N$ | CPU-time (s) | $\frac{V^{\mathrm{II}}(x_0)-V^*(x_0)}{V^*(x_0)}$ | $N$ | CPU-time (s) |
| $10^{-2}$ | $1.755 \cdot 10^{-3}$ | 6 | 0.00012 | $7.689 \cdot 10^{-4}$ | 12 | 0.00020 |
| $10^{-3}$ | $1.841 \cdot 10^{-4}$ | 24 | 0.00120 | $1.883 \cdot 10^{-4}$ | 24 | 0.00040 |
| $10^{-4}$ | $4.495 \cdot 10^{-8}$ | 96 | 0.0376 | $4.926 \cdot 10^{-8}$ | 96 | 0.0075 |
| $10^{-5}$ | $4.495 \cdot 10^{-8}$ | 96 | 0.0376 | $4.926 \cdot 10^{-8}$ | 96 | 0.0075 |
| $10^{-6}$ | $4.495 \cdot 10^{-8}$ | 96 | 0.0376 | $4.926 \cdot 10^{-8}$ | 96 | 0.0075 |
| $10^{-7}$ | $6.408 \cdot 10^{-9}$ | 192 | 0.360 | $8.841 \cdot 10^{-9}$ | 192 | 0.0440 |
| $10^{-8}$ | — | 384 | 3.264 | $1.468 \cdot 10^{-9}$ | 384 | 0.444 |

Table 2. System 2. Comparison of cost relative error, number of intervals of the final QP and overall CPU-time for solving all QPs vs. relative tolerance $\mu$ for PWLH-CLQR and FFOH-CLQR

| | PWLH-CLQR Algorithm | | | FFOH-CLQR Algorithm | | |
|---|---|---|---|---|---|---|
| $\mu$ | $\frac{V^{\mathrm{I}}(x_0)-V^*(x_0)}{V^*(x_0)}$ | $N$ | CPU-time (s) | $\frac{V^{\mathrm{II}}(x_0)-V^*(x_0)}{V^*(x_0)}$ | $N$ | CPU-time (s) |
| $10^{-2}$ | $3.3681 \cdot 10^{-3}$ | 6 | 0.00040 | $6.4084 \cdot 10^{-3}$ | 6 | 0.00027 |
| $10^{-3}$ | $5.0657 \cdot 10^{-5}$ | 48 | 0.334 | $1.1411 \cdot 10^{-4}$ | 48 | 0.0267 |
| $10^{-4}$ | $8.8931 \cdot 10^{-7}$ | 96 | 2.40 | $1.0786 \cdot 10^{-6}$ | 192 | 2.29 |
| $10^{-5}$ | 0 | 192 | 15.8 | $1.0786 \cdot 10^{-6}$ | 192 | 2.29 |

optimizes over the input and the slope in each interval, while FFOH-CLQR optimizes only over the input with the slope fixed by continuity at the end point of the interval. For instance, for System 1, with $\mu = 10^{-3}$ PWLH-CLQR optimizes (in the last QP) over $2mN = 48$ decision variables while FFOH-CLQR over $m(N+1) = 25$ variables. Also notice that a relative tolerance between $10^{-2}$ and $10^{-3}$ results in small suboptimality. For these reasons, the most effective algorithm appears to be FFOH-CLQR, which achieves a solution in 0.40 ms with a relative cost error of about $1 \cdot 10^{-4}$. For System 2, a reasonable value for $\mu$ is also between $10^{-2}$ and $10^{-3}$. For instance using FFOH-CLQR with $\mu = 10^{-3}$, we compute a solution in 27 ms with a relative cost error less than $2 \cdot 10^{-4}$. Such a computational time is clearly negligible compared to the systems dynamics. The resolution of the actuator in the application also implies a fairly loose tolerance ($\sim 10^{-3}$) on the solution should be used. It obviously makes little sense to compute an optimal input more accurately than the actuator hardware can resolve.

## 6. CONCLUSIONS

In this paper we presented a method for solving the infinite horizon continuous-time constrained linear quadratic regulator, by solving a finite number of finite dimensional quadratic programs. A number of unevenly spaced grid points are selected and adapted on-line to achieve a tolerance specification in the controller cost. The input parameterization is piecewise linear on the chosen time intervals; we examined both continuous and discontinuous parameterizations. The parameterizations guarantee exact input constraint satisfaction. Both of these input parameterizations converge to the optimal solution much more quickly than piecewise constant inputs, allowing a reduction in the final number of intervals and, therefore, decision variables. Furthermore, we derived exact discrete-time matrices and penalties to avoid quadrature errors, and moved offline all the computation required for solving ODEs and creating the Hessian and linear terms in the QPs solved online. Finally, we presented simulation results for two examples to illustrate the main ideas. The online complexity of solving the infinite horizon continuous-time CLQR with the proposed method is no larger than that required for solving an infinite horizon discrete-time CLQR problem of the same size.

## REFERENCES


Cannon, M. and Kouvaritakis, B. (2000). Infinite horizon predictive control of constrained linear systems. *Automatica*, 36, 943–955.

Diehl, M., Ferreau, H.J., and Haverbeke, N. (2008). Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *Proceedings of the International Workshop on Assessment and Future Directions of NMPC*. Pavia, Italy.

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

Kojima, A. and Morari, M. (2004). LQ control of constrained continuous-time systems. *Automatica*, 40, 1143–1155.

Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems*. John Wiley & Sons.

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

Pannocchia, G., Wright, S.J., and Rawlings, J.B. (2003). Existence and computation of infinite horizon model predictive control with active steady-state input constraints. *IEEE Trans. Auto. Contr.*, 48(6), 1002–1006.

Pannocchia, G., Rawlings, J.B., Mayne, D.Q., and Marquardt, W. (2009). On computing the solutions to the continuous time constrained linear quadratic regulator. Submitted for publication in IEEE Trans. Auto. Cont.

Prett, D.M. and Morari, M. (1987). *The Shell Process Control Workshop*. Butterworth Publishers.

Rao, C.V. and Rawlings, J.B. (1999). Steady states and constraints in model predictive control. *AIChE J.*, 45, 1266–1278.

Rossiter, J.A., Kouvaritakis, B., and Rice, M.J. (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34, 65–73.

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

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