Learning solutions to hybrid control problems using Benders cuts

Sandeep Menta SMENTA@CONTROL.EE.ETHZ.CH
Joseph Warrington WARRINGTON@CONTROL.EE.ETHZ.CH
John Lygeros LYGEROS@CONTROL.EE.ETHZ.CH
Automatic Control Lab, ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland
Manfred Morari MORARI@SEAS.UPENN.EDU
Elec. and Systems Engineering, Univ. of Pennsylvania, 220 S. 33rd St, Philadelphia, PA 19104, United States

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

Hybrid control problems are complicated by the need to make a suitable sequence of discrete decisions related to future modes of operation of the system. Model predictive control (MPC) encodes a finite-horizon truncation of such problems as a mixed-integer program, and then imposes a cost and/or constraints on the terminal state intended to reflect all post-horizon behaviour. However, these are often *ad hoc* choices tuned by hand after empirically observing performance. We present a learning method that sidesteps this problem, in which the so-called N-step Q-function of the problem is approximated from below, based on experience evaluating the policy. The function takes a state and a sequence of N control decisions as arguments, and therefore extends the traditional notion of a Q-function from reinforcement learning. After learning it from a training process exploring the state-input space, we use it in place of the usual MPC objective. We take an example hybrid control task and show that it can be completed successfully with a shorter planning horizon than conventional hybrid MPC thanks to our proposed method. Furthermore, we report that Q-functions trained with long horizons can be truncated to a shorter horizon for online use, yielding simpler control laws with apparently little loss of performance.

Keywords: Hybrid systems, reinforcement learning, approximate dynamic programming

1. Introduction

Hybrid control problems arise frequently in applications as diverse as robotic motion planning (Kuindersma et al., 2016) and power electronic converters (Geyer and Quevedo, 2014), and feature systems that must be steered between various discrete modes of operation. In discrete time, it is typical to use binary decision variables to encode choices of mode at each step of a planning horizon; see Marcucci and Tedrake (2019) for an up-to-date review of hybrid system descriptions.

Even with a perfect model, hybrid problems are made difficult by the need to plan a potentially long sequence of discrete decisions in advance, only a small fraction of which correspond to feasible state trajectories, and fewer still achieve the control objective. A common approach is hybrid MPC (Borrelli et al., 2017), in which an N-step decision problem is encoded as a mixed-integer convex program (MICP), and a terminal cost $V(x_N)$ and/or constraint $x_N \in \mathcal{X}_f$ is used to account for the evolution of the system after x_N, the N^{th} state in the planned sequence. However, one is often forced to make very conservative choices, for example $x_N = 0$, because subsequent safe behaviour is otherwise hard to guarantee. In some applications, physical insights can be used to tailor the terminal cost in order to bring about desired long-term behaviour, e.g. Stellato et al. (2016) for power converter control. However, there is no general approach for this, with the exception of highly specialized and computationally expensive *explicit hybrid MPC* methods (Beccuti et al., 2007; Axehill et al., 2014). Alternatively one can increase N to limit the impact of a poorly chosen $V(x_N)$ or $\mathcal{X}_f$, however this may often render real-time control impractically expensive.

1.1. Contributions

We propose instead to change the objective function of the N-step decision problem to a direct approximation of the optimal Q-function of the problem, also known as the state-action value function (Sutton and Barto, 2018), and learn this function using reinforcement learning (RL) techniques. The Q-function we employ is in fact an N-step extension of the traditional definition, taking as its arguments the same N inputs as used in hybrid MPC. We propose a hybrid extension of Warrington (2019), which generates an increasingly tight lower-approximation of the optimal Q-function, in the form of a pointwise maximum of lower-bounding functions. These lower bounds are model based, in that they feature the (known) model dynamics, stage cost, and constraints in their parameterization. They are constructed via an exploration of the state-action space during which greedy N-step policies are used to choose the actions. They are convex, and yet induce complex N-step predictive control policies as we only evaluate them along feasible hybrid system trajectories. This is far more expressive than the method of Bouchat and Jungers (2019), which also uses Benders cuts but fits convex value functions only to system modes in isolation and limits the resulting control performance.

Numerical experiments show that our controllers outperform naive implementations of hybrid MPC, without having to choose terminal costs or constraints. Although we only demonstrate and visualize our approach on a simple problem, our results represent a step towards a general model-based learning method for hybrid control problems that have no other tractable solution.

2. The N-step Q-function

We consider infinite-horizon hybrid control problems for a sub-class of standard mixed logical dynamical (MLD) systems, given in Bemporad and Morari (1999), that are time-invariant and do not have binary states or control inputs. We denote the optimal infinite-horizon cost, or *optimal value function*, $V^\star$:

$$
\begin{aligned}
V^\star(x) := \min_{\{x_t\}_{t=0}^\infty, \{u_t\}_{t=0}^\infty, \{\delta_t\}_{t=0}^\infty, \{z_t\}_{t=0}^\infty} \quad & \sum_{t=0}^{\infty} \gamma^t \left(\tfrac{1}{2} x_t^\top Q x_t + \tfrac{1}{2} u_t^\top R u_t \right) && \text{(1a)} \\
\text{s. t.} \quad & x_{t+1} = A x_t + B_1 u_t + B_2 \delta_t + B_3 z_t \,, \quad t = 0, 1, \ldots \,, && \text{(1b)} \\
& E_2 \delta_t + E_3 z_t \leq E_4 x_t + E_1 u_t + E_5 \,, \quad t = 0, 1, \ldots \,, && \text{(1c)} \\
& \delta_t \in \{0,1\}^{n_\delta} \,, \quad t = 0, 1, \ldots \,, && \text{(1d)} \\
& x_0 = x \,, && \text{(1e)}
\end{aligned}
$$

in which $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ is the input, and $z_t \in \mathbb{R}^{n_z}$ and δ_t are auxiliary continuous and binary variables required in the MLD framework. We assume $Q \succ 0$ and $R \succ 0$. Constraints (1b)-(1e) jointly bind the evolution of the system to feasible trajectories, including mode switches parameterized by δ_t. Constraints (1c) encode both the MLD dynamics, and state and input constraints that only involve x_t and u_t. We assume the system is "well-posed" in the sense that δ_t and

z_t are uniquely fixed by x_t and u_t. The discount factor $\gamma \in (0,1]$ does not appear in Bemporad and Morari (1999), but is common in the RL literature, and is used here to accommodate, for example, problems with periodic solutions, where $V^\star(x) = +\infty$ unless we allow $\gamma < 1$.

The notion of an optimal value function readily extends to the case where the first N control inputs are also arguments (as are the associated δ and z values), resulting in what we call the *optimal N-step state-action value function* $Q^{(N)\star}$:

$$Q^{(N)\star}(x, \{u\}_0^{N-1}, \{\delta\}_0^{N-1}, \{z\}_0^{N-1}) := \min_{\{x\}_0^\infty, \{u\}_N^\infty, \{\delta\}_N^\infty, \{z\}_N^\infty} \textstyle\sum_{t=0}^\infty \gamma^t \left(\frac{1}{2}x_t^\top Q x_t + \frac{1}{2}u_t^\top R u_t\right) \quad \text{s. t. } \text{(1b)-(1e)}. \qquad (2)$$

We use the shorthands $\{u\}_0^{N-1}$ etc. to avoid writing out full argument lists $u_0, u_1, \dots, u_{N-1}$, etc. Note that the first N inputs and auxiliary variables are parameters rather than optimization variables in (2). The conventional "optimal Q-function" in the usual RL sense (Sutton and Barto, 2018) is just a special case of the above with $N = 1$. It is trivial to show that

$$Q^{(N)\star}(x, \{u\}_0^{N-1}, \{\delta\}_0^{N-1}, \{z\}_0^{N-1}) = \textstyle\sum_{t=0}^{N-1} \gamma^t \left(\frac{1}{2}x_t^\top Q x_t + \frac{1}{2}u_t^\top R u_t\right) + \gamma^N V^\star(x_N),$$

in which the state sequence $x_1, \dots, x_N$ is generated by the dynamics (1b) and satisfies (1c)-(1e).

For later use we denote triples of inputs and auxiliary variables $s = (u, \delta, z)$. Triples that are instantaneously compatible with the state x are described by the set $\mathcal{S}(x) := \{s = (u,\delta,z) \in \mathbb{R}^m \times \{0,1\}^{n_\delta} \times \mathbb{R}^{n_z} \ : \ E_2\delta + E_3 z \le E_1 u + E_4 x + E_5\}$. We also define the set of feasible N-step trajectories with initial state x,

$$\mathcal{S}^{(N)}(x) := \left\{ \begin{array}{l} \{s\}_0^{N-1} = (\{u\}_0^{N-1}, \{\delta\}_0^{N-1}, \{z\}_0^{N-1}) \ : \ x_0 = x,\ s_k \in \mathcal{S}(x_k) \text{ and} \\ \quad x_k = A x_{k-1} + B_1 u_{k-1} + B_2 \delta_{k-1} + B_3 z_{k-1} \text{ for } k = 1, \dots, N-1 \end{array} \right\}. \qquad (3)$$

2.1. Control via N-step approximate Q-functions

An approximation of $Q^{(N)\star}$, denoted $Q^{(N)}$, can be used to generate a control policy:

$$\pi(x; Q^{(N)}) \in \left[\operatorname*{arg\,min}_{\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x)} Q^{(N)}(x, \{s\}_0^{N-1}) \right]_{u_0} \qquad (4)$$

where the outer $[\cdots]_{u_0}$ indicates that the initial control decision u_0 is taken from the optimal $\{s\}_0^{N-1}$, assuming an optimum is attained. In the RL literature, this would be called a *greedy policy* with respect to $Q^{(N)}$. From (2) it follows that $\pi(x; Q^{(N)\star})$ is an optimal policy bringing about the optimal infinite-horizon "closed-loop" cost $V^\star(x)$ when applied recursively. By the definition of $V^\star$ in (1), any other $Q^{(N)}$ will generally induce closed-loop costs greater than $V^\star(x)$.

2.2. Hybrid MPC

In hybrid MPC, the infinite-horizon problem (1) is approximated by an N-step MICP, in which a terminal cost $V(x_N)$ replaces all terms for time steps $t \ge N$ in the objective function. A terminal constraint $x_N \in \mathcal{X}_f$ may also be imposed to guarantee certain stability or recursive feasibility properties. See §17.4 of Borrelli et al. (2017) for the formulation, which truncates (1) to N steps.

According to DP principles, one would ideally like to use $V^\star(x_N)$ as the terminal cost to ensure optimal performance for any N. However, this function is generally very complicated for hybrid systems, as it is the parametric minimum of infinite-dimensional problem (1). Convex approximations of $V^\star$ amenable to MICP solvers may all be inherently poor for some hybrid control tasks.

3. Learning N-step Q-functions

The central idea of our approach is to approximate $Q^{(N)\star}$ directly and use policy (4) to control the system. This is the same as replacing the whole of the MPC cost function with an approximation $Q^{(N)}$, rather than trying to use an approximation of $V^\star$ as the terminal cost. We find that $Q^{(N)\star}$ can be approximated more accurately with a simple function parameterization than $V^\star$. This is attractive, given that the optimization variables and constraints governing policy (4) are the same as in hybrid MPC. Surprisingly, our proposed approximation $Q^{(N)}$ is convex in its arguments, but yields rich closed-loop behaviour and high-quality approximations thanks to the fact that we only use it in the context of input sequences satisfying the hybrid dynamics. This extends the Benders-based learning approach of Warrington (2019) to hybrid systems, and to the required N-step horizon.

3.1. Bellman operator for N-step Q-functions

We define a Bellman operator $\mathcal{T}_Q^{(N)}$ for a generic function $Q^{(N)}$ taking the same arguments, in a manner similar to $\mathcal{T}_Q$ in Warrington (2019) for any $x_0 \in \mathbb{R}^n$ and $\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)$:

$$\mathcal{T}_Q^{(N)} Q^{(N)}(x_0, \{s\}_0^{N-1}) = \tfrac{1}{2} x_0^\top Q x_0 + \tfrac{1}{2} u_0^\top R u_0 + \inf_{s_N \in \mathcal{S}(x_{N-1})} \gamma Q^{(N)}(x_1, \{s\}_1^N)\,. \tag{5}$$

From the definition (2) one can see that $\mathcal{T}_Q^{(N)} Q^{(N)\star}(x_0, \{s\}_0^{N-1}) = Q^{(N)\star}(x_0, \{s\}_0^{N-1})$ for all $x_0 \in \mathbb{R}^n$ and $\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)$. The $\mathcal{T}_Q^{(N)}$ operator is monotonic: if $Q_a^{(N)}(x_0, \{s\}_0^{N-1}) \leq Q_b^{(N)}(x_0, \{s\}_0^{N-1})$ for all $x_0 \in \mathbb{R}^n$ and $\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)$, then we have:

$$\mathcal{T}_Q^{(N)} Q_a^{(N)}(x_0, \{s\}_0^{N-1}) \leq \mathcal{T}_Q^{(N)} Q_b^{(N)}(x_0, \{s\}_0^{N-1}), \forall x_0 \in \mathbb{R}^n\,, \{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)\,.$$

3.2. Benders cut

Suppose we have an approximation of $Q^{(N)\star}$, denoted $Q_I^{(N)}$, taking the form of a point-wise maximum of functions $q_0, q_1, \ldots, q_I$:

$$Q_I^{(N)}(x_0, \{s\}_0^{N-1}) := \max_{i=0,\ldots,I}\{q_i(x_0, \{s\}_0^{N-1})\}, \tag{6}$$

where the functions each satisfy a lower bounding property,

$$q_i(x_0, \{s\}_0^{N-1}) \leq Q^{(N)\star}(x_0, \{s\}_0^{N-1})\,, \quad \forall (x_0, \{s\}_0^{N-1})\,. \tag{7}$$

Thus $Q_I^{(N)} \leq Q^{(N)\star}$. We now show that if the functions q_i are parameterized in a particular way, then applying the Bellman operator to $Q_I^{(N)}$ at some $(x_0, \{s\}_0^{N-1})$ generates a new valid lower bound q_{I+1}. This new bound tightens our approximation of $Q^{(N)\star}$ under certain conditions.

The Bellman operator $\mathcal{T}_Q^{(N)}$ in (5) can be written out for N-step Q-functions of the form (6) in an equivalent manner:

$$
\begin{aligned}
\mathcal{T}_{\mathcal{Q}}^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1}) = \inf_{\{x\}_1^N, s_N, \alpha} \quad & \tfrac{1}{2} x_0^\top Q x_0 + \tfrac{1}{2} u_0^\top R u_0 + \gamma \alpha && \text{(8a)} \\
\text{s. t.} \quad & x_{t+1} = A x_t + B_1 u_t + B_2 \delta_t + B_3 z_t, \ t = 0, \dots, N-1 && \text{(8b)} \\
& E_2 \delta_N + E_3 z_N \leq E_1 u_N + E_4 x_N + E_5, && \text{(8c)} \\
& q_i(x_1, \{s\}_1^N) \leq \alpha, \quad i = 0, \dots, I && \text{(8d)}
\end{aligned}
$$

where α is an epigraph variable, which is bounded from below by each function q_i, and therefore models the value of $Q_I^{(N)}(x_1, \{s\}_1^N)$. For parameters $\{s\}_k^{k+N-1} \in \mathcal{S}^{(N)}(x_k)$, the linearity of the system dynamics and constraints motivates us to impose a separable form of q_i-functions for $i \geq 1$,

$$
q_i(x_k, \{s\}_k^{k+N-1}) = q_i^{x_0}(x_k) + \textstyle\sum_{j=0}^{N-1} q_i^{u_j}(u_{k+j}) + \sum_{j=0}^{N-1} q_i^{\delta_j}(\delta_{k+j}) + \sum_{j=0}^{N-1} q_i^{z_j}(z_{k+j}) + c_i \tag{9}
$$

where the terms $q_i^{x_0}$ etc. are convex linear and quadratic functions built up recursively starting from an initial lower bound q_0:

$$
q_0(x_k, \{s\}_k^{k+N-1}) = \textstyle\sum_{t=k}^{k+N-1} \frac{1}{2} \gamma^{t-k} x_t^\top Q x_t + \frac{1}{2} \gamma^{t-k} u_t^\top R u_t. \tag{10}
$$

As q_0 represents the sum of N stage costs, it is clearly a valid lower bound on $Q^{(N)\star}$ from definition (2). Note q_0 includes the x costs directly, whereas $q_1, q_2, \dots$ do not. Forms (9) and (10) are choices we have made in order to be able to define a tractable recursive algorithm for approximating $Q^{(N)\star}$. The treatment of these is made clear in Lemma 3.1.

We now form the dual of (8) by assigning the dual multipliers $\nu_t \in \mathbb{R}^n$, $\mu \in \mathbb{R}^{n_c}$, and $\lambda \in \mathbb{R}^{I+1}$ to constraints (8b), (8c), and (8d) respectively, and using the forms (9)-(10) for the lower bounding functions. This can be viewed as a new operator $\mathcal{D}^{(N)}$ acting on $Q_I^{(N)}$:

$$
\begin{aligned}
\mathcal{D}^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1}) := \sup_{\nu, \mu, \lambda} \quad & \tfrac{1}{2} x_0^\top Q x_0 + \tfrac{1}{2} u_0^\top R u_0 + \nu_0^\top (A x_0 + B_1 u_0 + B_2 \delta_0 + B_3 z_0) \\
& + \textstyle\sum_{t=1}^{N-1} \nu_t^\top (B_1 u_t + B_2 \delta_t + B_3 z_t) + \sum_{i=1}^{I} \lambda_i c_i - \mu^\top E_5 \\
& + \textstyle\sum_{i=1}^{I} \lambda_i \sum_{t=0}^{N-2} \Big(q_i^{u_t}(u_{t+1}) + q_i^{\delta_t}(\delta_{t+1}) + q_i^{z_t}(z_{t+1}) \Big) \\
& + \textstyle\sum_{t=1}^{N-1} \lambda_0 \frac{\gamma^{t-1}}{2} u_t^\top R u_t + \xi(\nu, \mu, \lambda) && \text{(11a)} \\
\text{s. t.} \quad & \mu \geq 0, \quad \lambda \geq 0, \quad \mathbb{1}^\top \lambda = \gamma, && \text{(11b)}
\end{aligned}
$$

where

$$
\xi(\nu, \mu, \lambda) :=
$$

$$
\inf_{\{x\}_1^N, s_N} \left\{ \begin{array}{c}
(\nu_1^\top A - \nu_0^\top) x_1 + \sum_{i=1}^{I} \lambda_i q_i^{x_0}(x_1) + \sum_{t=2}^{N-1} \left(\nu_t^\top A - \nu_{t-1}^\top \right) x_t + \sum_{t=1}^{N-1} \lambda_0 \frac{\gamma^{t-1}}{2} x_t^\top Q x_t \\
- \left(\mu^\top E_4 + \nu_{N-1}^\top \right) x_N + \lambda_0 \frac{\gamma^{N-1}}{2} x_N^\top Q x_N - \mu^\top E_1 u_N + \sum_{i=1}^{I} \lambda_i q_i^{u_{N-1}}(u_N) \\
+ \lambda_0 \frac{\gamma^{N-1}}{2} u_N^\top R u_N + \mu^\top E_3 z_N + \sum_{i=1}^{I} \lambda_i q_i^{z_{N-1}}(z_N) + \mu^\top E_2 \delta_N + \sum_{i=1}^{I} \lambda_i q_i^{\delta_{N-1}}(\delta_N)
\end{array} \right\}.
$$

For an approximate Q-function $Q_I^{(N)}$ of the form (6) constructed from lower bounding functions q_i satisfying (7), a new lower bounding function q_{I+1} for $Q^{(N)\star}$ can be formed using the optimal multipliers $(\nu^\star, \mu^\star, \lambda^\star)$ from the solution of (11):

Lemma 3.1 *The function*

$$q_{I+1}(x_0, \{s\}_0^{N-1}) := q_{I+1}^{x_0}(x_0) + \textstyle\sum_{t=0}^{N-1} q_{I+1}^{u_t}(u_t) + \sum_{t=0}^{N-1} q_{I+1}^{\delta_t}(\delta_t) + \sum_{t=0}^{N-1} q_{I+1}^{z_t}(z_t) + c_{I+1}\,, \tag{12}$$

$$\begin{array}{lll}
\textit{where} \quad q_{I+1}^{x_0}(x_0) = \frac{1}{2}x_0^\top Q x_0 + \nu_0^{\star\top} A x_0, & q_{I+1}^{\delta_0}(\delta_0) = \nu_0^{\star\top} B_2 \delta_0, & \\
q_{I+1}^{u_0}(u_0) = \frac{1}{2}u_0^\top R u_0 + \nu_0^{\star\top} B_1 u_0, & q_{I+1}^{z_0}(z_0) = \nu_0^{\star\top} B_3 z_0, & \\
q_{I+1}^{u_t}(u_t) = \sum_{i=1}^{I} \lambda_i^\star\, q_i^{u_{t-1}}(u_t) + \lambda_0^\star \frac{\gamma^{t-1}}{2} u_t^\top R u_t + \nu_t^{\star\top} B_1 u_t, & & t = 1, \dots, N-1 \\
q_{I+1}^{\delta_t}(\delta_t) = \sum_{i=1}^{I} \lambda_i^\star\, q_i^{\delta_{t-1}}(\delta_t) + \nu_t^{\star\top} B_2 \delta_t, & & t = 1, \dots, N-1 \\
q_{I+1}^{z_t}(z_t) = \sum_{i=1}^{I} \lambda_i^\star\, q_i^{z_{t-1}}(z_t) + \nu_t^{\star\top} B_3 z_t, & & t = 1, \dots, N-1 \\
c_{I+1} = \sum_{i=1}^{I} \lambda_i^\star c_i - \mu^{\star\top} E_5 + \xi(\nu^\star, \mu^\star, \lambda^\star) & &
\end{array}$$

and the triplet $(\nu^\star, \mu^\star, \lambda^\star)$ *solves problem* (11) *for a particular parameter* $(\hat{x}_0, \{\hat{s}\}_0^{N-1})$*, satisfies the global lower bounding property*

$$q_{I+1}(x_0, \{s\}_0^{N-1}) \le Q^{(N)\star}(x_0, \{s\}_0^{N-1})\,, \quad \forall x_0 \in \mathbb{R}^n,\ \{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)\,.$$

Proof The optimal dual solution $(\nu^\star, \mu^\star, \lambda^\star)$ of (11) with parameter $(\hat{x}_0, \{\hat{s}\}_0^{N-1})$ is in general a sub-optimal solution for any other parameter $(x_0, \{s\}_0^{N-1})$ where $\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x_0)$. Let

$$\begin{aligned}
\mathcal{H}(x_0, \{s\}_0^{N-1}) = \tfrac{1}{2}x_0^\top Q x_0 + \tfrac{1}{2}u_0^\top R u_0 + \nu_0^{\star\top}(A x_0 + B_1 u_0 + B_2 \delta_0 + B_3 z_0) + \\
\textstyle\sum_{t=1}^{N-1} \nu_t^{\star\top}(B_1 u_t + B_2 \delta_t + B_3 z_t) + \sum_{i=1}^{I} \lambda_i^\star \sum_{t=0}^{N-2} \Big(q_i^{u_t}(u_{t+1}) + q_i^{\delta_t}(\delta_{t+1}) + q_i^{z_t}(z_{t+1}) \Big) + \\
\textstyle\sum_{t=1}^{N-1} \lambda_0^\star \frac{\gamma^{t-1}}{2} u_t^\top R u_t + \sum_{i=1}^{I} \lambda_i^\star c_i + \xi(\nu^\star, \mu^\star, \lambda^\star) - \mu^{\star\top} E_5
\end{aligned}$$

Hence, $\mathcal{H}(x_0, \{s\}_0^{N-1}) \le \mathcal{D}^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1})$. Note that $(\nu^\star, \mu^\star, \lambda^\star)$ is feasible in (11) for all parameters $(x_0, \{s\}_0^{N-1})$, as the feasible set is independent of the parameter. From weak duality,

$$\mathcal{D}^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1}) \le \mathcal{T}_Q^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1}).$$

We get $\mathcal{T}_Q^{(N)} Q_I^{(N)} \le \mathcal{T}_Q^{(N)} Q^{(N)\star}$ from the fact that $Q_I^{(N)} \le Q^{(N)\star}$ and the monotonicity property of $\mathcal{T}_Q^{(N)}$. Combining this with $\mathcal{T}_Q^{(N)} Q^{(N)\star}(x_0, \{s\}_0^{N-1}) = Q^{(N)\star}(x_0, \{s\}_0^{N-1})$, the Bellman optimality condition, we obtain $\mathcal{H}(x_0, \{s\}_0^{N-1}) \le Q^{(N)\star}(x_0, \{s\}_0^{N-1})$. The result follows as $\mathcal{H}$ is equal to q_{I+1} in (12). ■

Lemmas III.2-4 of Warrington (2019) carry over to the present setting but have been omitted for brevity. In particular, we note that adding a new lower bounding function q_{I+1} at the parameter $(x_0, \{s\}_0^{N-1})$ raises the lower bound there by $\mathcal{D}^{(N)} Q_I^{(N)}(x_0, \{s\}_0^{N-1}) - Q_I^{(N)}(x_0, \{s\}_0^{N-1})$.

3.3. Learning algorithm

The training procedure for generating $Q^{(N)}$ is listed in Algorithm 1. Given a set of points $\mathcal{X}_{\text{Alg}}$ we evaluate the N-step greedy control policy, *i.e.*, the sequence $\{s\}_0^{N-1}$ that minimises $Q_I^{(N)}(x, \{s\}_0^{N-1})$,

Algorithm 1 Modified Q-Benders algorithm for MLD systems

```
**Inputs:** System model; training points $\mathcal{X}_{\text{Alg}} := \{x^1, \dots, x^M\}$, $I_{\text{max}}$
Set $I = 0$, $\beta^\star = \infty$ and $q_0(x, \{s\}_0^{N-1}) = \sum_{t=0}^{N-1} \gamma^t \left(\frac{1}{2} {x_t}^\top Q x_t + \frac{1}{2} {u_t}^\top R u_t\right)$
**while** $I \leq I_{\text{max}}$ and $\beta^\star \geq \beta_{\text{min}}$ **do**
    $Q_I^{(N)}(\cdot, \cdot, \cdot, \cdot) \leftarrow \max_{i=0,\dots,I} q_i(\cdot, \cdot, \cdot, \cdot)$
    **for each** $x^a \in \mathcal{X}_{\text{Alg}}$ **do**
        $\{s^a\}_0^{N-1} \leftarrow \arg\min_{\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x^a)} \left\{ Q_I^{(N)}(x^a, \{s\}_0^{N-1}) \right\}$
        $\beta(x^a, \{s^a\}_0^{N-1}; Q_I^{(N)}) \leftarrow \mathcal{D}^{(N)} Q_I^{(N)}(x^a, \{s^a\}_0^{N-1}) - Q_I^{(N)}(x^a, \{s^a\}_0^{N-1})$
    **end for**
    $x^\star \leftarrow \arg\max_{x^a \in \mathcal{X}_{\text{Alg}}} \{\beta(x^a, \{s^a\}_0^{N-1}; Q_I^{(N)})\}$
    **if** $(\nu^\star, \mu^\star, \lambda^\star)$ is optimal for problem (11) with parameter $(x^\star, \{s^\star\}_0^{N-1})$ **and** $\beta^\star = \beta(x^\star, \{s^\star\}_0^{N-1}; Q_I^{(N)}) \geq \beta_{\text{min}}$ **then**
        Add $q_{I+1}(\cdot, \cdot, \cdot, \cdot)$ parameterized by $(\nu^\star, \mu^\star, \lambda^\star)$ to the set of $q_i(\cdot, \cdot, \cdot, \cdot)$ functions
    **end if**
    $I \leftarrow I + 1$
**end while**
**Output:** $Q_I^{(N)}(\cdot, \cdot, \cdot, \cdot) = \max_{i=0,\dots,I} q_i(\cdot, \cdot, \cdot, \cdot)$
```

for each $x \in \mathcal{X}_{\text{Alg}}$. We then evaluate the improvement (or equivalently, increase) to the Q-function approximation, β, that would result if a cut were made there as in Lemma 3.1:

$$\beta(x^a, \{s^a\}_0^{N-1}; Q_I^{(N)}) := \mathcal{D}^{(N)} Q_I^{(N)}(x^a, \{s^a\}_0^{N-1}) - Q_I^{(N)}(x^a, \{s^a\}_0^{N-1}) \,.$$

We then find the $x^\star \in \mathcal{X}_{\text{Alg}}$ that brings about the largest β, denoted $\beta^\star$, and construct q_{I+1} by solving (11) at $(x^\star, \{s^\star\}_0^{N-1})$ as long as $\beta^\star \geq \beta_{\text{min}}$. This is repeated until $I > I_{\text{max}}$ or $\beta^\star < \beta_{\text{min}}$.

4. Numerical example

In this section we present the results of using Algorithm 1 on a scalar example with hybrid dynamics to learn the optimal $Q^{(N)}$-function. The system evolves as $x_{t+1} = x_t + u_t - 5\delta_{a,t} + 2\delta_{b,t}$, where $\delta_{a,t} = 1 \Leftrightarrow x_t \in [4.5, 5.5]$, and $\delta_{b,t} = 1 \Leftrightarrow x_t \in [1, 2]$. Thus $\delta_{a,t}$ and $\delta_{b,t}$ encode "jumps" of -5 and $+2$ respectively. The input is bounded as $|u_t| \leq 0.25$, and these bounds are encoded in the MLD constraints (1c) along with "big-M" conditions governing $\delta_{a,t}$ and $\delta_{b,t}$; see Bemporad and Morari (1999) for an explanation of this procedure. The stage cost is $\frac{1}{2}x_t^2 + \frac{1}{20}u_t^2$ and $\gamma = 1$, thus the task is to reach the origin. In our simulations we say the controller *fails to complete the task* if the trajectories never reach $|x_t| < 0.005$ for some t. For initial conditions $x_0 \in [1, 4.5)$, the control task is non-trivial as the only path to the origin uses the $\delta_{a,t}$ jump, and it is often optimal to use the $\delta_{b,t}$ jump once or even twice before this.

Consider the function

$$\underline{Q}^{(N)}(x) = \inf_{\{s\}_0^{N-1} \in \mathcal{S}^{(N)}(x)} Q^{(N)}(x, \{s\}_0^{N-1}).$$

It is readily shown that $\underline{Q}^{(N)\star}(x) = V^\star(x)$, thus evaluating $\underline{Q}^{(N)}$ for any suboptimal $Q^{(N)}$ gives an indication of approximation quality. This is shown in the top half of Fig. 1. Note that even this simple problem has a complicated $V^\star(x)$. The approximations are generated by running Algorithm 1 over the discretised range $x \in [-0.5, 6.5]$, $I_{\text{max}} = 50$ and $\beta_{\text{min}} = 10^{-5}$. Our approximations capture the shape and values of $V^\star(x)$ well, and improve with the horizon length N; see the upper plot of Fig. 1. Algorithm 1 terminates after $I = (50, 40, 15)$ cuts for $N = (10, 13, 15)$ respectively. As N increases, each iteration of the algorithm takes longer, but the number of iterations decreases.

The closed-loop cost of recursively applying the control policy (4) induced by $Q^{(N)}$ is compared against the optimal cost function $V^\star(x)$, for all initial conditions, in the lower plot of Fig. 1, along with the cost using a hybrid MPC controller (denoted HMPC) with the LQR terminal cost derived from the linear dynamics omitting the δ variables. In region $\mathcal{X}_h \coloneqq [1, 1.75] \cup [2, 3.75]$, short-horizon controllers fail at the task as the system trajectories get stuck just above $x = 2$. The costs of such failed trajectories have been capped at 100 in the plot. For $N < 10$, both controllers fail at the task for all $x \in \mathcal{X}_h$ and for $N > 10$ both techniques succeed everywhere. For $N = 10$ the policy induced by $Q^{(N)}$ succeeds everywhere, whereas HMPC fails in some places.

A surprising benefit of our proposed q-function parameterization is that for the trained $Q^{(N)}$-function, using the sum of just $q_i^{x_0}, q_i^{u_t}, q_i^{\delta_t}$ and $q_i^{z_t}$ for $t = \{0, 1, \dots, N_t\}$ to construct an approximation that uses a "policy horizon" length of $N_t < N$ is often enough for the controller to succeed for all initial conditions. Moreover,er with increasing training horizon N we find the task can be completed everywhere with a shorter policy horizon N_t. For $N = (11, 13, 15)$ the lowest N_t that leads to success everywhere is $N_t = (8, 5, 4)$ respectively and the corresponding results are shown in Fig. 2. The average policy computation time for HMPC with $N = 11$, the lowest N that leads to success everywhere, is 35.5 ms, and that for using $Q^{(15)}$ truncated to $N_t = 4$ is 29.0 ms.

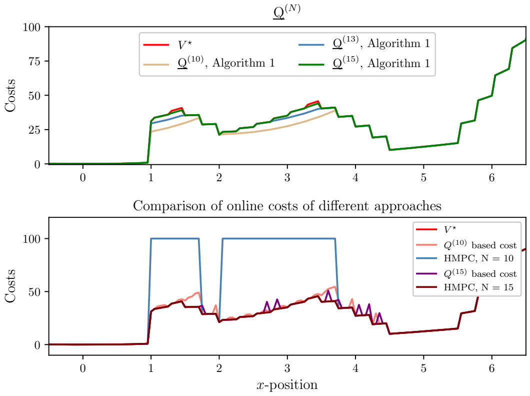

Figure 1: *Upper:* $V^\star$ vs. $\underline{Q}^{(N)}$.
Lower: Closed-loop costs of (4) compared to HMPC.

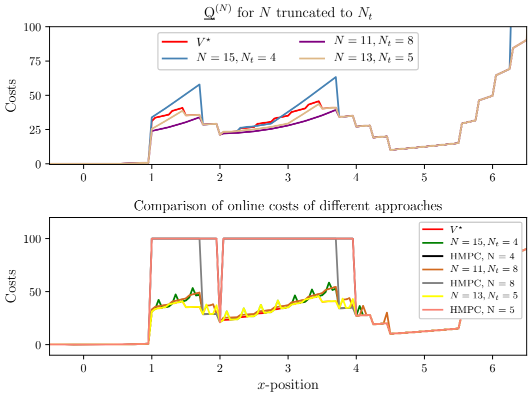

Figure 2: *Upper:* $V^\star$ vs. $\underline{Q}^{(N)}$ truncated to $N_t < N$.
Lower: Online costs with truncations of $Q^{(N)}$ in (4).

5. Conclusion

We have demonstrated a learning algorithm in which the optimal N-step state-action value function is approximated directly, as an improvement on the naive MPC cost function for hybrid systems. In particular, we showed that good control performance can be obtained with a shorter decision horizon, and without the need to find a suitable terminal cost or state constraint: instead the optimal N-step Q-function is approximated based on training experience evaluating the control policy. The method readily extends to higher dimensional systems, and future work will focus on more elaborate problems of practical relevance, characterizing the training process in more detail, and making safety guarantees for sub-optimal Q-functions.

References

Daniel Axehill, Thomas Besselmann, Davide Martino Raimondo, and Manfred Morari. A parametric branch and bound approach to suboptimal explicit hybrid mpc. *Automatica*, 50(1):240–246, 2014.

AG Beccuti, G Papafotiou, Roberto Frasca, and M Morari. Explicit hybrid model predictive control of the DC-DC boost converter. In *2007 IEEE Power Electronics Specialists Conference*, pages 2503–2509. IEEE, 2007.

Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

Jean Bouchat and Raphaël M Jungers. *Reinforcement learning for the optimal control of hybrid systems*. MSc Thesis, UC Louvain, 2019.

Tobias Geyer and Daniel E Quevedo. Multistep finite control set model predictive control for power electronics. *IEEE Transactions on power electronics*, 29(12):6836–6846, 2014.

Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

Tobia Marcucci and Russ Tedrake. Mixed-integer formulations for optimal control of piecewise-affine systems. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 230–239. ACM, 2019.

Bartolomeo Stellato, Tobias Geyer, and Paul J Goulart. High-speed finite control set model predictive control for power electronics. *IEEE Transactions on power electronics*, 32(5):4007–4020, 2016.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Joseph Warrington. Learning continuous Q-functions using generalized benders cuts. In *Proceedings of the European Control Conference, Naples, Italy*, 2019.