



Receding-Horizon Control of Constrained Switched Systems with Neural Networks as Parametric Function Approximators

Lukas Markolf¹ · Olaf Stursberg¹

Received: 17 December 2021 / Accepted: 7 October 2022
© The Author(s) 2022

Abstract

This work studies receding-horizon control of discrete-time switched linear systems subject to polytopic constraints for the continuous states and inputs. The objective is to approximate the optimal receding-horizon control strategy for cases in which the online computation is intractable due to the necessity of solving mixed-integer quadratic programs in each discrete time instant. The proposed approach builds upon an approximated optimal finite-horizon control law in closed-loop form with guaranteed constraint satisfaction. The paper derives the properties of recursive feasibility and asymptotic stability for the proposed approach. A numerical example is provided for illustration and evaluation of the approach.

Keywords Model approximation · Neural networks · Optimization · Online control · Reinforcement learning · Stability

Introduction

Applications from many domains, such as flexible production systems, smart grids, autonomous vehicles, logistic systems, smart cities, etc., are nowadays termed as *cyber-physical systems* (CPS), see e.g. [1]. These systems are characterized by the integration of physical objects with digital components of information processing and control interacting over communication networks. A typical property of CPS is the co-existence and interaction of continuous dynamics (most often stemming from the physical objects) and discrete-event dynamics (mostly arising from the digital components). Mixed continuous-discrete dynamics, also referred to as *hybrid dynamics*, has been subject of intensive research over the last decades [2].

A special class of hybrid dynamic systems are *switched systems*, for which the transition between different

continuous dynamics is defined by a switching logic. With respect to this logic, one can distinguish between internally forced and externally forced switching [3]. The earlier type typically depends on the continuous state, and a frequently considered example is piecewise affine systems [4]. In contrast, the paper on hand focuses on switched systems with externally forced switching, where an external decision unit decides on when to execute a transition to a new mode of continuous dynamics. This class of switched systems is relevant for applications in which, e.g., a discrete or supervisory controller (such as a programmable logic controller) coexists with continuous control loops. More precisely, this paper studies the control of discrete-time switched linear systems (DSLQ) with mixed-inputs, where discrete control inputs have to be selected to determine the continuous dynamics, and continuous inputs serve to achieve control goals for the continuous state, in particular the satisfaction of state constraints.

Discrete-time linear quadratic regulation problems for switched linear systems without constraints, here referred to as DSLQR problems, have been intensively studied before. It is shown in [5] that any finite-horizon value function of the DSLQR problem is a pointwise minimum of a finite number of quadratic functions. The quadratic functions are exactly described by a set of positive definite matrices obtained from the dynamic programming (DP) solution of the DSLQR problem. Even for a finite time horizon, the computation of the exact solution to the DSLQR problem

This article is part of the topical collection “Informatics in Control, Automation and Robotics” guest edited by Kurosh Madani, Oleg Gusikhin and Henk Nijmeijer.

✉ Lukas Markolf
lukas.markolf@uni-kassel.de
Olaf Stursberg
stursberg@uni-kassel.de

¹ Control and System Theory, Department of Electrical Engineering and Computer Science, University of Kassel, Wilhelmshöher Allee 71-73, 34121 Kassel, Germany

is an \mathcal{NP} -hard [6] problem. The critical point is that the number of possible discrete input sequences, and by that also the number of quadratic functions, grows exponentially with the time horizon. DP with pruning [7] and relaxed DP [8, 9] have been shown to reduce complexity drastically, as is also described in the generalizing work [4, 10].

It is proven in [5] that the finite-horizon value function converges under certain conditions exponentially fast to the infinite-horizon value function. The work proposes a relaxation framework to solve the infinite-horizon DSLQR problem with guaranteed closed-loop stability but suboptimal performance. The use of a stabilizing base policy and the concept of *rollout* are exploited in [11] to address the problem of finding low complexity policies with (preferably tight) performance bounds for the infinite-horizon DSLQR problem. Recently, a Q -learning algorithm with customized Q -function approximation has been proposed in [12]. The approach is based on analytic results for the value function, and it addresses the infinite-horizon DSLQR problem for higher dimensional cases—these are currently intractable to be solved by state-of-the-art methods. In general, receding horizon control (RHC) constitutes an approach to approximating infinite-horizon control problems [13]. According to [10], an RHC strategy can be expressed explicitly as a piecewise linear state feedback control law defined over (usually non-convex) regions.

The DSLQR problem for switched linear systems with polytopic constraints for the continuous states and inputs, here referred to as DCSLQR, is less studied. In [14], the infinite-horizon DCSLQR problem is addressed by splitting the problem into an unconstrained DSLQR problem and a finite-horizon DCSLQR problem. The finite-horizon DCSLQR part is then formulated as a mixed-integer quadratic programming (MIQP) problem. In general, MIQP problems are known to be \mathcal{NP} -hard problems. The online solution of such MIQP problems for the control of switched systems is addressed in [15], by determining a trade-off between performance and computation time through a tailored tree search with cost bounds and search heuristics. The finite-horizon DCSLQR problem has been recently considered in a previous paper of the authors [16]. There, the optimal closed-loop control law has been approximated by neural networks (NN) which are trained offline. This allows one a fast determination of guaranteed admissible and (preferably optimal) continuous and discrete inputs for any state of the DSLs. In general, NN have the attractive property of being able to approximate functions arbitrarily close when used with general activation functions [17–19], and thus have contributed significantly to the recent success of machine learning applications [20–23]. The use of NN as function approximators for policies (control laws) and value (cost-to-go) functions, as in DeepMind’s popular computer program AlphaZero [23], is the core of numerous approximate

dynamic programming and reinforcement learning algorithms [24–30]. Moreover, NN have also been considered recently for approximating receding-horizon (or model predictive) control laws [31–34].

Missing so far are techniques that efficiently solve and represent the result of the DLSQR problem for the case with constraints over infinite horizons—proposing a corresponding technique by approximating the solution by NN while ensuring all constraints is the goal and contribution of this paper. Note that for the case without constraints, the work in [10] has shown that solving an MIQP problem at each time instant is necessary, leading to high computation times for larger problem instances. However, the MIQP problem to be solved at each time step can be transformed into one for DSLs with finite time horizons, as proposed in [16]. This motivates the concept of approximating the optimal RHC strategy by use of NN, and by exploiting the results from [16] for efficient computation. The objective thus is to suggest an approach which makes the computation of the control inputs (based on the approximating RHC strategy) significantly faster while guaranteeing constraint satisfaction. Thus, the present paper extends the work in [16] from a single optimization over a finite horizon to a setting of receding (finite) horizons in order to cover infinite time spans. As a consequence of this extension, recursive feasibility and asymptotic stability have to be considered as additional aspects, and this paper shows how these properties can be proven, requiring in addition, of course, that all state and input constraints are guaranteed to be satisfied throughout.

The paper is structured such that Section “[Problem Formulation and Preliminaries](#)” first introduces the RHC problem, and analyzes the properties and challenges of its solution. Motivated by the challenges, a simplified RHC problem is introduced, which can be transformed into the finite-horizon control problem considered in [16]. Section “[Finite-Horizon Control with Parametric Function Approximators](#)” reminds the reader of fundamental results from [16], which are then used to develop the new approach in Section “[Receding-Horizon Control with Parametric Function Approximators](#)”. A numerical example is provided for illustration in Section “[Numerical Example](#)”, and the paper is concluded in Section “[Conclusion](#)”. The “[Appendix A. Proofs of the Propositions](#)” contains all proofs for the results established in the mentioned sections.

Problem Formulation and Preliminaries

For defining discrete-time switched linear systems (DSLs), let $x_k \in \mathbb{R}^{n_x}$ denote the continuous state, $u_k \in \mathbb{R}^{n_u}$ the continuous control input, and $v_k \in \mathbb{M} = \{1, \dots, M\}$ the discrete control input, all for time $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$. The latter selects for any k the parameterization (A_i, B_i) of a linear

dynamics with system matrix $A_i \in \mathbb{R}^{n_x \times n_x}$ and input matrix $B_i \in \mathbb{R}^{n_x \times n_u}$. The DSLS is then written as:

$$x_{k+1} = A_{v_k} x_k + B_{v_k} u_k =: f(x_k, u_k, v_k), \tag{1}$$

with initial state $x_0 \in \mathbb{R}^{n_x}$. The states and inputs are subject to constraints:

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \in \mathbb{N}_0, \tag{2}$$

where it is required that \mathcal{X} and \mathcal{U} are polytopic and contain the origin in their interior. Subsequently, let $x_{j|k}$ denote a continuous state at time $k + j$ predicted at time k . According to (1), the predicted state at time $k + j + 1$ is given by:

$$x_{j+1|k} = f(x_{j|k}, u_{j|k}, v_{j|k}),$$

$$x_{0|k} := x_k, \quad j \in \mathbb{N}_0,$$

where $u_{j|k}$ and $v_{j|k}$ denote the continuous and discrete inputs at time $k + j$ predicted at time k , respectively. Predicted input sequences at k over a time span $\{k + j, \dots, k + N - 1\}$ are written as:

$$\phi_{j \rightarrow N-1|k}^u := (u_{j|k}, \dots, u_{N-1|k}),$$

$$\phi_{j \rightarrow N-1|k}^v := (v_{j|k}, \dots, v_{N-1|k}).$$

Consider the quadratic cost-to-go over the horizon from j to N :

$$J_{j \rightarrow N}(x_{j|k}, \phi_{j \rightarrow N-1|k}^u, \phi_{j \rightarrow N-1|k}^v)$$

$$= g_N(x_{N|k}) + \sum_{i=j}^{N-1} g(x_{i|k}, u_{i|k}, v_{i|k}) \tag{3}$$

subject to (1) with terminal cost function:

$$g_N(x_{N|k}) = x_{N|k}^T P x_{N|k}, \tag{4}$$

and stage cost function:

$$g(x_{j|k}, u_{j|k}, v_{j|k}) = x_{j|k}^T Q_{v_{j|k}} x_{j|k} + u_{j|k}^T R_{v_{j|k}} u_{j|k}, \tag{5}$$

where $P = P^T > 0$, $Q_{v_k} = Q_{v_k}^T > 0$, and $R_{v_k} = R_{v_k}^T > 0$ with $v_k \in \mathbb{M}$ are (switched) weighting matrices. For the sake of simplicity, the shorter notation J_j instead of $J_{j \rightarrow N}$, $\phi_{j|k}^u$ instead of $\phi_{j \rightarrow N-1|k}^u$, and $\phi_{j|k}^v$ instead of $\phi_{j \rightarrow N-1|k}^v$ is used when appropriate.

This work aims at determining online a control strategy based on a receding-horizon principle for steering the DSLS from the initial state x_0 into the origin.

Problem 1 (Receding-Horizon Control Problem) For the current state x_k at time k and the DSLS (1) subject to (2), find

a continuous input sequence $\phi_{0|k}^{u*} := (u_{0|k}^*, \dots, u_{N-1|k}^*)$ and a discrete input sequence $\phi_{0|k}^{v*} := (v_{0|k}^*, \dots, v_{N-1|k}^*)$ with prediction horizon N , such that the system state reaches the origin within N time steps, while the cost function (3) is minimized:

$$(\phi_{0|k}^{u*}, \phi_{0|k}^{v*}) \in \arg \min_{(\phi_{0|k}^u, \phi_{0|k}^v)} J_0(x_{0|k}, \phi_{0|k}^u, \phi_{0|k}^v)$$

subject to :

$$x_{0|k} = x_k, \quad x_{j+1|k} = f(x_{j|k}, u_{j|k}, v_{j|k}),$$

$$x_{j|k} \in \mathcal{X}, \quad u_{j|k} \in \mathcal{U}, \quad v_{j|k} \in \mathbb{M},$$

$$x_{N|k} = 0, \quad j \in \{0, \dots, N - 1\}. \tag{6}$$

If a feasible solution to the problem exists and an optimal receding-horizon control (RHC) strategy is obtained, it can be applied by imposing the first element $u_{0|k}^*$ of the continuous input sequence $\phi_{0|k}^{u*}$ and the first element $v_{0|k}^*$ of the discrete input sequence $\phi_{0|k}^{v*}$ to (1) at k :

$$(u_k, v_k) = (u_{0|k}^*, v_{0|k}^*) =: (\mu_0^{u*}(x_k), \mu_0^{v*}(x_k)). \tag{7}$$

The closed-loop dynamics for the DSLS controlled by the strategy is then:

$$x_{k+1} = f(x_k, \mu_0^{u*}(x_k), \mu_0^{v*}(x_k)) =: f_{cl}^*(x_k), \quad k \in \mathbb{N}_0. \tag{8}$$

Subsequently, let $J_0^*(x_k) = J_0(x_k, \phi_{0|k}^{u*}, \phi_{0|k}^{v*})$ denote the optimal cost-to-go for steering x_k into the origin within N steps.

Recursive Feasibility and Stability

The symbol $\mathcal{X}_{j \rightarrow N}$ is introduced to denote the set of states that can be steered into the origin within $N - j$ steps. A recursive definition of $\mathcal{X}_{j \rightarrow N}$ is given by:

$$\mathcal{X}_i = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U}, \exists v \in \mathbb{M} \text{ such that } f(x, u, v) \in \mathcal{X}_{i+1}\},$$

$$\mathcal{X}_N := \{0\},$$

with $i \in \{N - 1, \dots, j\}$. When appropriate, the shorter notation \mathcal{X}_j is used instead of $\mathcal{X}_{j \rightarrow N}$. The Problem 1 has a feasible solution if and only if x_k is an element of \mathcal{X}_0 .

An RHC strategy is called recursively feasible if $x_0 \in \mathcal{X}_0$ implies feasibility of Problem 1 for all future states $x_k, k > 0$. From the definition of the RHC strategy follows that $x_k \in \mathcal{X}_0$ implies $x_{k+1} \in \mathcal{X}_1$. Hence, a sufficient condition for recursive feasibility is that $\mathcal{X}_0 \supseteq \mathcal{X}_1$.

Let $\mathcal{X}_j^{(v_j, \dots, v_{N-1})}$ be the set of states that can be steered into the origin within $N - j$ steps for a fixed discrete input sequence (v_j, \dots, v_{N-1}) :

$$\mathcal{X}_j^{(v_j, v_{j+1}, \dots, v_{N-1})} = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} \text{ such that } f(x, u, v_j) \in \mathcal{X}_{j+1}^{(v_{j+1}, \dots, v_{N-1})}\},$$

$$\mathcal{X}_{N-1}^{(v_{N-1})} = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} \text{ such that } f(x, u, v_{N-1}) \in \mathcal{X}_N\}.$$

For brevity, the more compact notation:

$$\mathcal{X}_j^{(v_j, v_{j+1}, \dots, v_{N-1})} = \text{Pre}^{(v_j)}(\mathcal{X}_j^{(v_{j+1}, \dots, v_{N-1})}) \cap \mathcal{X},$$

$$\mathcal{X}_{N-1}^{(v_{N-1})} = \text{Pre}^{(v_{N-1})}(\mathcal{X}_N) \cap \mathcal{X} \tag{9}$$

is used, where the operator $\text{Pre}^{(v_j)}(\mathcal{S})$ returns the set of predecessor states to the set $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ for a fixed discrete input v_j :

$$\text{Pre}^{(v_j)}(\mathcal{S}) = \{x \in \mathbb{R}^{n_x} \mid \exists u \in \mathcal{U} \text{ such that } f(x, u, v_j) \in \mathcal{S}\}. \tag{10}$$

It follows that \mathcal{X}_j is the union of all possible sets $\mathcal{X}_j^{(v_j, \dots, v_{N-1})}$:

$$\mathcal{X}_j = \bigcup_{(v_j, \dots, v_{N-1}) \in \mathbb{M}^{N-j}} \mathcal{X}_j^{(v_j, \dots, v_{N-1})}, \tag{11}$$

and the following propositions can be established (see the Appendix for the corresponding proofs).

Proposition 1 For a given prediction horizon N and a fixed discrete input sequence $(v_j, \dots, v_{N-1}) \in \mathbb{M}^{N-j}$, the sets $\mathcal{X}_j^{(v_j, \dots, v_{N-1})}$ are polytopes. The sets \mathcal{X}_j are non-convex in the general case, with the property that:

$$\mathcal{X}_j \supseteq \mathcal{X}_{j+1}, \quad j \in \{0, \dots, N - 1\}.$$

Proposition 2 If $x_0 \in \mathcal{X}_0$, then the optimal RHC strategy (7) is recursively feasible.

Proposition 3 The origin of the closed-loop system (8) is asymptotically stable with domain of attraction \mathcal{X}_0 .

The Propositions 1–3 are extensions of results known from literature on RHC for systems without switching, as e.g. in [35]. More particularly, the sets \mathcal{X}_j are extensions (of the feasible sets considered there) with respect to the discrete inputs. If \mathcal{X}_N is control-invariant (as defined in the proof of Proposition 1), then the sets \mathcal{X}_j share the property (with the sets of the case without switching) that \mathcal{X}_j grows as j decreases, and stops growing when reaching the maximal control-invariant set (see e.g. [35, Remark 11.3]). Note that \mathcal{X}_N is a singleton here, which contains the origin and thus is control-invariant.

Properties of the RHC Problem

Let V_j^* denote the optimal cost-to-go for steering $x_{j|k}$ into the origin within $N - j$ steps for a chosen discrete input sequence $\phi_{j|k}^v$:

$$V_j^*(x_{j|k}, \phi_{j|k}^v) = \min_{\phi_{j|k}^u} J_j(x_{j|k}, \phi_{j|k}^u, \phi_{j|k}^v) \tag{12a}$$

subject to: $x_{i+1|k} = f(x_{i|k}, u_{i|k}, v_{i|k}),$
 $x_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U}, \quad v_{i|k} \in \mathbb{M},$ (12b)
 $x_{N|k} = 0, \quad i \in \{j, \dots, N - 1\}.$

The optimization problem (12) is a quadratic program (QP), and has a feasible solution if and only if $x_{j|k} \in \mathcal{X}_j^{(v_j, \dots, v_{N-1|k})}$.

As a convention, $V_j^*(x_{j|k}, \phi_{j|k}^v)$ is set to infinity for the case that (12) has no feasible solution.

The optimal cost-to-go $J_j^*(x_{j|k})$ for steering $x_{j|k}$ into the origin within $N - j$ steps, as well as the corresponding input sequences $\phi_{j|k}^{u*} = \Phi_j^{u*}(x_{j|k})$ and $\phi_{j|k}^{v*} = \Phi_j^{v*}(x_{j|k})$ may be obtained by solving a QP for each possible discrete input sequence $\phi_{j|k}^v$:

$$J_j^*(x_{j|k}) = \min_{\phi_{j|k}^v} V_j^*(x_{j|k}, \phi_{j|k}^v) \text{ subject to: } \phi_{j|k}^v \in \mathbb{M}^{N-j}, \tag{13}$$

$$\Phi_j^{v*}(x_{j|k}) \in \arg \min_{\phi_{j|k}^v} V_j^*(x_{j|k}, \phi_{j|k}^v) \text{ subject to: } \phi_{j|k}^v \in \mathbb{M}^{N-j}, \tag{14}$$

$$\Phi_j^{u*}(x_{j|k}) \in \arg \min_{\phi_{j|k}^u} J_j(x_{j|k}, \phi_{j|k}^u, \Phi_j^{v*}(x_{j|k})) \text{ subject to: 12b.} \tag{15}$$

Denote as $\mathcal{U}_j^v(x_{j|k}, v_{j|k})$ the set that contains the admissible continuous inputs for a state $x_{j|k}$ and a discrete input $v_{j|k}$ to reach the state set \mathcal{X}_{j+1} :

$$\mathcal{U}_j^v(x_{j|k}, v_{j|k}) = \{u \in \mathcal{U} \mid f(x_{j|k}, u, v_{j|k}) \in \mathcal{X}_{j+1}\}. \tag{16}$$

Instead of solving a QP problem for each possible discrete input sequence, the optimal cost-to-go $J_0^*(x_k)$ and the optimal RHC strategy (7) may be computed, in principle, by setting $j = 0$ and solving the optimization problem:

$$\mathcal{Q}_j^*(x_{j|k}, v_{j|k}) = \min_{u_{j|k}} g(x_{j|k}, u_{j|k}, v_{j|k})$$

$$+ J_{j+1}^*(f(x_{j|k}, u_{j|k}, v_{j|k})) \tag{17}$$

subject to: $u_{j|k} \in \mathcal{U}_j^v(x_{j|k}, v_{j|k}),$

for each possible discrete input $v_{j|k} \in \mathbb{M}$ and just one step, leading to:

$$J_j^*(x_{j|k}) = \min_{v_{j|k}} Q_j^*(x_{j|k}, v_{j|k}) \text{ subject to: } v_{j|k} \in \mathbb{M}, \tag{18}$$

$$\mu_j^{v^*}(x_{j|k}) \in \arg \min_{v_{j|k}} Q_j^*(x_{j|k}, v_{j|k}) \text{ subject to: } v_{j|k} \in \mathbb{M}, \tag{19}$$

and:

$$\begin{aligned} \mu_j^{u^*}(x_{j|k}) \in \arg \min_{u_{j|k}} & g(x_{j|k}, u_{j|k}, \mu_j^{v^*}(x_{j|k})) \\ & + J_{j+1}^*(f(x_{j|k}, u_{j|k}, \mu_j^{v^*}(x_{j|k}))) \end{aligned} \tag{20}$$

subject to: $u_{j|k} \in \mathcal{U}_j^v(x_{j|k}, \mu_j^{v^*}(x_{j|k}))$.

This requires, of course, that J_{j+1}^* and $\mathcal{U}_j^v(x_{j|k}, v_{j|k})$ are already known and that a globally optimal solution is found. By definition, J_N^* is equal to the terminal cost (4), i.e. $J_N^*(x_{N|k}) := g_N(x_{N|k})$. The following propositions establish that the optimization problem (17) is a difficult one, in the general case, since it constitutes a nonlinear program with non-convex objective function J_{j+1}^* and non-convex constraints $\mathcal{U}_j^v(x_{j|k}, v_{j|k})$. (The pointwise minimum of two functions J_1 and J_2 is defined as $J(x) = \min\{J_1(x), J_2(x)\}$, in analog to the definition of the pointwise maximum in [36].)

Proposition 4 *The optimal cost-to-go function J_j^* , $j \in \{0, \dots, N - 1\}$ is in the general case a pointwise minimum of functions that are convex and piecewise quadratic on polyhedra.*

Proposition 5 *The sets $\mathcal{U}_j^v(x_{j|k}, v_{j|k})$, $j \in \{0, \dots, N - 2\}$ are non-convex in general.*

Challenges and Objective

Problem 1 is an MIQP and known to be \mathcal{NP} -hard. The number of possible discrete input sequences, given by $|\mathbb{M}^N| = M^N$, grows exponentially with the prediction horizon N . Thus, the trivial approach of solving a QP for each possible discrete input sequence to solve Problem 1 is computationally intractable in almost all cases. While more efficient approaches to solve MIQPs exist (such as branch-and-bound or branch-and-cut techniques), these approaches typical still require too much time for the online optimization of RHC.

On first sight, the approach of computing the optimal RHC strategy (7) according to (19) and (20) requires only the solution of M optimization problems. However, these optimization problems are in general nonlinear programs with non-convex objective function J_1^* and non-convex constraints $\mathcal{U}_0^v(x_k, v_k)$, and thus challenging to solve. Moreover, J_1^* has a complicated form (pointwise minimum of functions

that are convex and piecewise quadratic on polyhedra, see Proposition 4), and the derivation of analytic solution is usually not possible. Last but not least, the proof of Proposition 5 provides the insight that the determination of $\mathcal{U}_0^v(x_k, v_k)$ relies on the computation of the union of M^{N-1} polytopes, and requires the expensive offline computation of M^{N-1} many polytopes $\mathcal{X}_1^{(v_1, \dots, v_{N-1})}$.

Thus, the objective of the further derivations of this work is to efficiently approximate the optimal RHC strategy to make the computation of the control inputs faster, while guaranteeing properties like constraint satisfaction, recursive feasibility, and asymptotic stability.

Simplified RHC Problem

As discussed above, the optimal RHC strategy could be computed by solving (19) and (20). The problem is, however, that the set of admissible continuous inputs $\mathcal{U}_0^v(x_k, v_k)$ is non-convex. While established methods exist for nonlinear programs with convex constraints, the solution of a nonlinear program with non-convex constraints (arising for each discrete input sequence) is computationally intractable for online application in most cases. Moreover, the determination of $\mathcal{U}_0^v(x_k, v_k)$ is computationally demanding. The objective of this section is to introduce a simplified RHC problem based on convex control-invariant subsets $\tilde{\mathcal{X}}_j$ of the (generally) non-convex sets \mathcal{X}_j . By doing so, the set of admissible continuous inputs $\tilde{\mathcal{U}}_0^v(x_k, v_k)$ is convex as well, and the computation of $\tilde{\mathcal{U}}_0^v(x_k, v_k)$ is less demanding.

Let the sets $\tilde{\mathcal{X}}_j$ be (recursively) defined by:

$$\begin{aligned} \tilde{\mathcal{X}}_j &= \{x \in \mathcal{X} \mid \forall v \in \mathbb{M} : \exists u \in \mathcal{U} \text{ such that } f(x, u, v) \in \tilde{\mathcal{X}}_{j+1}\}, \\ \tilde{\mathcal{X}}_N &= \mathcal{X}_N. \end{aligned}$$

Thus, for any state $x_{j|k} \in \tilde{\mathcal{X}}_j$ and an arbitrary choice of the discrete input $v_{j|k} \in \mathbb{M}$, at least one admissible continuous input $u_{j|k} \in \mathcal{U}$ exists such that $f(x_{j|k}, u_{j|k}, v_{j|k}) \in \tilde{\mathcal{X}}_{j+1}$. It follows from the definition that $\tilde{\mathcal{X}}_j$ is the intersection of all possible polytopes $\mathcal{X}_j^{(v_j, \dots, v_{N-1})}$:

$$\tilde{\mathcal{X}}_j = \bigcap_{(v_j, \dots, v_{N-1}) \in \mathbb{M}^{N-j}} \mathcal{X}_j^{(v_j, \dots, v_{N-1})}. \tag{21}$$

Hence, $\tilde{\mathcal{X}}_j$ is a polytope as well with the property that $\tilde{\mathcal{X}}_j \subseteq \mathcal{X}_j$. It is worth mentioning that it is possible, in principle, to determine a further polytopic inner approximation of $\tilde{\mathcal{X}}_j$ with smaller number of facets, if it is necessary to reduce complexity. Algorithm 1 provides a method for computing the sets $\tilde{\mathcal{X}}_j$ recursively.

Algorithm 1 Controllable Set Computation [16].**Require:** $M, N, \mathcal{X}, \tilde{\mathcal{X}}_N$ **Ensure:** $\tilde{\mathcal{X}}_0, \dots, \tilde{\mathcal{X}}_{N-1}$

```

1: for  $j = N - 1$  to  $0$  do
2:    $\tilde{\mathcal{X}}_j = \mathcal{X}$ 
3:   for  $v = 1$  to  $M$  do
4:      $\tilde{\mathcal{X}}_j \leftarrow \text{Pre}^{(v)}(\tilde{\mathcal{X}}_{j+1}) \cap \tilde{\mathcal{X}}_j$ 
5:   end for
6: end for

```

Problem 2 (Simplified RHC Problem) For a current state x_k at time k and the DSLS (1) subject to (2), find a continuous input sequence $\phi_{0|k}^{\tilde{u}^*} := (\tilde{u}_{0|k}^*, \dots, \tilde{u}_{N-1|k}^*)$ and a discrete input sequence $\phi_{0|k}^{\tilde{v}^*} := (\tilde{v}_{0|k}^*, \dots, \tilde{v}_{N-1|k}^*)$ for a prediction horizon N that steers the state into the origin within N time steps, while satisfying $x_{j|k} \in \tilde{\mathcal{X}}_j$ and minimizing the quadratic cost function (3):

$$\begin{aligned}
 (\phi_{0|k}^{\tilde{u}^*}, \phi_{0|k}^{\tilde{v}^*}) \in \arg \min_{(\phi_{0|k}^{\tilde{u}}, \phi_{0|k}^{\tilde{v}})} J_0(x_{0|k}, \phi_{0|k}^{\tilde{u}}, \phi_{0|k}^{\tilde{v}}) \\
 \text{subject to: } & x_{0|k} = x_k, \quad x_{j+1|k} = f(x_{j|k}, u_{j|k}, v_{j|k}), \\
 & x_{j|k} \in \tilde{\mathcal{X}}_j, \quad u_{j|k} \in \mathcal{U}, \quad v_{j|k} \in \mathbb{M}, \\
 & x_{N|k} = 0, \quad j \in \{0, \dots, N-1\}.
 \end{aligned} \quad (22)$$

In case a feasible solution exists, the application of the first elements of the input sequences to (1) at time k :

$$(\tilde{u}_k^*, \tilde{v}_k^*) = (\tilde{u}_{0|k}^*, \tilde{v}_{0|k}^*) =: (\mu_0^{\tilde{u}^*}(x_k), \mu_0^{\tilde{v}^*}(x_k)) \quad (23)$$

leads to the closed-loop dynamics:

$$x_{k+1} = f(x_k, \mu_0^{\tilde{u}^*}(x_k), \mu_0^{\tilde{v}^*}(x_k)) =: \tilde{f}_{\text{cl}}^*(x_k). \quad (24)$$

Recursive feasibility of the RHC strategy (23) and asymptotic stability of the origin of the closed-loop system (24) with a domain of attraction $\tilde{\mathcal{X}}_0$ can be proven in accordance with Propositions 2 and 3. Again, the optimal cost-to-go $\tilde{J}_j^*(x_{j|k})$ and the corresponding input sequences $\phi_{j|k}^{\tilde{u}^*} = \Phi_j^{\tilde{u}^*}(x_{j|k})$ and $\phi_{j|k}^{\tilde{v}^*} = \Phi_j^{\tilde{v}^*}(x_{j|k})$ may be obtained by solving a QP:

$$\tilde{V}_j^*(x_{j|k}, \phi_{j|k}^{\tilde{v}^*}) = \min_{\phi_{j|k}^{\tilde{u}}} J_j(x_{j|k}, \phi_{j|k}^{\tilde{u}}, \phi_{j|k}^{\tilde{v}^*}) \quad (25a)$$

$$\begin{aligned}
 \text{subject to: } & x_{i+1|k} = f(x_{i|k}, u_{i|k}, v_{i|k}), \\
 & x_{i|k} \in \tilde{\mathcal{X}}_i, \quad u_{i|k} \in \mathcal{U}, \quad v_{i|k} \in \mathbb{M}, \\
 & x_{N|k} = 0, \quad i \in \{j, \dots, N-1\}
 \end{aligned} \quad (25b)$$

for each possible discrete input sequence:

$$\tilde{J}_j^*(x_{j|k}) = \min_{\phi_{j|k}^{\tilde{v}}} \tilde{V}_j^*(x_{j|k}, \phi_{j|k}^{\tilde{v}}) \text{ subject to: } \phi_{j|k}^{\tilde{v}} \in \mathbb{M}^{N-j}, \quad (26)$$

$$\Phi_j^{\tilde{v}^*}(x_{j|k}) \in \arg \min_{\phi_{j|k}^{\tilde{v}}} \tilde{V}_j^*(x_{j|k}, \phi_{j|k}^{\tilde{v}}) \text{ subject to: } \phi_{j|k}^{\tilde{v}} \in \mathbb{M}^{N-j}, \quad (27)$$

$$\Phi_j^{\tilde{u}^*}(x_{j|k}) \in \arg \min_{\phi_{j|k}^{\tilde{u}}} J_j(x_{j|k}, \phi_{j|k}^{\tilde{u}}, \Phi_j^{\tilde{v}^*}(x_{j|k})) \text{ subject to: 25b.} \quad (28)$$

The polytopes \mathcal{U} and $\tilde{\mathcal{X}}_j$ can be written in half-space representation as:

$$\mathcal{U} = \{u \in \mathbb{R}^{n_u} \mid H^{\mathcal{U}}u \leq h^{\mathcal{U}}\}, \quad \tilde{\mathcal{X}}_j = \{x \in \mathbb{R}^{n_x} \mid H^{\tilde{\mathcal{X}}_j}x \leq h^{\tilde{\mathcal{X}}_j}\},$$

with matrices $H^{\mathcal{U}}, H^{\tilde{\mathcal{X}}_j}$ and vectors $h^{\mathcal{U}}, h^{\tilde{\mathcal{X}}_j}$ of appropriate dimensions. By use of these sets, the set of admissible continuous inputs for a state $x_{j|k}$ and a discrete input $v_{j|k}$ at prediction time $k+j$ can be written here as:

$$\begin{aligned}
 \tilde{\mathcal{U}}_j^{\tilde{v}}(x_{j|k}, v_{j|k}) &= \{u \in \mathcal{U} \mid f(x_{j|k}, u, v_{j|k}) \in \tilde{\mathcal{X}}_{j+1}\} \\
 &= \left\{ u \in \mathbb{R}^{n_u} \mid \begin{bmatrix} H^{\tilde{\mathcal{X}}_{j+1}} B_{v_{j|k}} \\ H^{\mathcal{U}} \end{bmatrix} u \right. \\
 &\quad \left. \leq \begin{bmatrix} h^{\tilde{\mathcal{X}}_{j+1}} - H^{\tilde{\mathcal{X}}_{j+1}} A_{v_{j|k}} x_{j|k} \\ h^{\mathcal{U}} \end{bmatrix} \right\}.
 \end{aligned} \quad (29)$$

Obviously, $\tilde{\mathcal{U}}_j^{\tilde{v}}(x_{j|k}, v_{j|k})$ is a polytope. The optimal cost-to-go $\tilde{J}_0^*(x_k)$ and the optimal RHC strategy (23) of the simplified RHC problem may alternatively be computed by setting $j = 0$ and solving the nonlinear program:

$$\begin{aligned}
 \tilde{Q}_j^*(x_{j|k}, v_{j|k}) &= \min_{u_{j|k}} g(x_{j|k}, u_{j|k}, v_{j|k}) \\
 &\quad + \tilde{J}_{j+1}^*(f(x_{j|k}, u_{j|k}, v_{j|k})) \\
 \text{subject to: } & u_{j|k} \in \tilde{\mathcal{U}}_j^{\tilde{v}}(x_{j|k}, v_{j|k}).
 \end{aligned} \quad (30)$$

The constraints $\tilde{\mathcal{U}}_j^{\tilde{v}}(x_{j|k}, v_{j|k})$ are convex in this case, and it applies that:

$$\tilde{J}_j^*(x_{j|k}) = \min_{v_{j|k}} \tilde{Q}_j^*(x_{j|k}, v_{j|k}) \text{ subject to: } v_{j|k} \in \mathbb{M}, \quad (31)$$

$$\mu_j^{\tilde{v}^*}(x_{j|k}) \in \arg \min_{v_{j|k}} \tilde{Q}_j^*(x_{j|k}, v_{j|k}) \text{ subject to: } v_{j|k} \in \mathbb{M}, \quad (32)$$

and:

$$\begin{aligned} \mu_j^{\tilde{u}^*}(x_{j|k}) \in \arg \min_{u_{j|k}} & g(x_{j|k}, u_{j|k}, \mu_j^{\tilde{v}^*}(x_{j|k})) \\ & + \tilde{J}_{j+1}^* \left(f(x_{j|k}, u_{j|k}, \mu_j^{\tilde{v}^*}(x_{j|k})) \right) \quad (33) \\ \text{subject to: } & u_{j|k} \in \tilde{\mathcal{U}}_j^v(x_{j|k}, \mu_j^{\tilde{v}^*}(x_{j|k})). \end{aligned}$$

Again, \tilde{J}_N^* is equal to the terminal cost (4) by definition, i.e. $\tilde{J}_N^*(x_{N|k}) := g_N(x_{N|k})$. The optimal cost-to-go \tilde{J}_j^* is still a pointwise minimum of functions that are convex and piecewise quadratic on polyhedra, such that the challenge of deriving an analytical expression remains.

The Problem 2 can be transformed into the finite-horizon control problem considered in previous work of the authors [16]. There, the optimal finite-horizon control laws have been approximated by (deep) neural networks. These finite-horizon control laws are fundamental for the RHC approach presented in Section “Receding-Horizon Control with Parametric Function Approximators”, thus the relevant results from [16] are summarized in Section “Finite-Horizon Control with Parametric Function Approximators” and tailored to the problem formulated before.

Finite-Horizon Control with Parametric Function Approximators

For the DSLS with finite-horizon N :

$$x_{j+1} = A_v x_j + B_v u_j =: f(x_j, u_j, v_j), \quad j \in \{0, \dots, N-1\} \quad (34)$$

subject to the constraints:

$$x_j \in \tilde{\mathcal{X}}_j, \quad x_N = 0, \quad u_j \in \mathcal{U}, \quad j \in \{0, \dots, N-1\}, \quad (35)$$

consider the following problem, into which the simplified RHC problem 2 can be transferred with x_k as initial state x_0 .

Problem 3 (Finite-Horizon Control Problem) For a given initial state x_0 at time $j = 0$, the DSLS (34) subject to (35), and a finite time horizon N , find input sequences $\phi_0^{\tilde{u}^*} := (\tilde{u}_0^*, \dots, \tilde{u}_{N-1}^*)$ and $\phi_0^{\tilde{v}^*} := (\tilde{v}_0^*, \dots, \tilde{v}_{N-1}^*)$ that steer x_0 into the origin within N time steps, while minimizing (3):

$$\begin{aligned} (\phi_0^{\tilde{u}^*}, \phi_0^{\tilde{v}^*}) \in \arg \min_{(\phi_0^{\tilde{u}}, \phi_0^{\tilde{v}})} & J_0(x_0, \phi_0^{\tilde{u}}, \phi_0^{\tilde{v}}) \\ \text{subject to: } & x_{j+1} = f(x_j, u_j, v_j), \quad (36) \\ & x_j \in \tilde{\mathcal{X}}_j, \quad u_j \in \mathcal{U}, \quad v_j \in \mathbb{M}, \\ & x_N = 0, \quad j \in \{0, \dots, N-1\}. \end{aligned}$$

The optimal finite-horizon control law:

$$\tilde{\pi}^* = \left\{ (\mu_0^{\tilde{u}^*}, \mu_0^{\tilde{v}^*}), \dots, (\mu_{N-1}^{\tilde{u}^*}, \mu_{N-1}^{\tilde{v}^*}) \right\} \quad (37)$$

with $v_j = \mu_j^{\tilde{v}^*}(x_j)$ and $u_j = \mu_j^{\tilde{u}^*}(x_j)$ as defined in (32) and (33) does not only produce the optimal sequences $\phi_0^{\tilde{v}^*}$ and $\phi_0^{\tilde{u}^*}$ for a single, but also for all initial states $x_0 \in \tilde{\mathcal{X}}_0$. This control law is, however, not readily applicable, as discussed in Section “Problem Formulation and Preliminaries”.

In [16], the functions $\mu_j^{\tilde{u}^*}$ and $\mu_j^{\tilde{v}^*}$ are approximated with the help of neural networks. The main ideas required for the further method development in Section Receding-Horizon Control with Parametric Function Approximators are briefly repeated here.

The approximation of the cost-to-go functions \tilde{J}_j^* by parametric functions \tilde{J}_j with real-valued parameter vectors r_j^j constitutes a so-called *approximation in value space* [30]. This allows to approximate the function \tilde{Q}_j^* defined in (30) by solving the following one-step look-ahead optimization problem with convex constraints:

$$\begin{aligned} \tilde{Q}_{\text{VS},j}(x_j, v_j) = \min_{u_j} & g(x_j, u_j, v_j) + \tilde{J}_{j+1}(f(x_j, u_j, v_j); r_j^j) \\ \text{subject to: } & u_j \in \tilde{\mathcal{U}}_j^v(x_j, v_j), \quad (38) \end{aligned}$$

where $\tilde{J}_N(x_N) := g_N(x_N)$. Let $\xi_{\text{VS},j}^{\tilde{u}}(x_j, v_j)$ denote a solution of the optimization problem (38):

$$\begin{aligned} \xi_{\text{VS},j}^{\tilde{u}}(x_j, v_j) \in \arg \min_{u_j} & g(x_j, u_j, v_j) + \tilde{J}_{j+1}(f(x_j, u_j, v_j); r_j^j) \\ \text{subject to: } & u_j \in \tilde{\mathcal{U}}_j^v(x_j, v_j). \quad (39) \end{aligned}$$

The finite-horizon control law (37) can then be approximated by:

$$\tilde{\pi}_{\text{VS}} = \left\{ (\mu_{\text{VS},0}^{\tilde{u}}, \mu_{\text{VS},0}^{\tilde{v}}), \dots, (\mu_{\text{VS},N-1}^{\tilde{u}}, \mu_{\text{VS},N-1}^{\tilde{v}}) \right\}, \quad (40)$$

with:

$$\mu_{\text{VS},j}^{\tilde{v}}(x_j) \in \arg \min_{v_j} \tilde{Q}_j(x_j, v_j) \text{ subject to: } v_j \in \mathbb{M}, \quad (41)$$

$$\mu_{\text{VS},j}^{\tilde{u}}(x_j) = \xi_{\text{VS},j}^{\tilde{u}}(x_j, \mu_{\text{VS},j}^{\tilde{v}}(x_j)). \quad (42)$$

If a closed-form expression for the partial derivative $[\partial \tilde{J}_{j+1} / \partial x_{j+1}]$ is available, well established gradient methods can be used to solve (38). The satisfaction of the convex constraints $u_j \in \tilde{\mathcal{U}}_j^v(x_j, v_j)$ in methods of this type is not a problem, see e.g. [37, Chapter 3]. This approach with neural networks to approximate the cost-to-go has been proposed in [38] to guarantee constraint satisfaction for systems without switching. Note that satisfaction of the constraints (35) is even guaranteed in case of imperfect approximations of

the optimal cost-to-go functions, and if the iterative procedure of gradient methods is stopped before finding a local minimum.

The alternative approach of approximating the functions (μ_j^u, μ_j^v) directly by parametric functions $(\mu_{PS,j}^u, \mu_{PS,j}^v)$ with real-valued parameter vectors (r_j^u, r_j^v) constitutes a so-called *approximation in policy space* [30]. In what follows, a possible realization of $\mu_{PS,j}^v$ is presented.

Motivated by classification tasks, parametric functions:

$$p_j(x_j; r_j^v) = [p_{1,j}(x_j) \dots p_{M,j}(x_j)]^T, \\ \sum_{i=1}^M p_{i,j}(x_j) = 1, \quad p_{i,j}(x_j) \geq 0$$

are introduced, which are trained to predict the probability $p_{v_j,j}$ of a discrete input v_j being optimal for state x_j at time j . Note that $p_j(x_j)$ represents by definition a valid probability distribution. The function $\mu_{PS,j}^v$ can be defined as the one which assigns to each state x_j at time j the discrete input v_j with the highest predicted probability to be optimal. The procedure of establishing p_j as neural network is described in Section [Neural Networks as Parametric Approximators](#).

The finite-horizon control law (37) can be approximated on the basis of *approximation in policy space* by:

$$\tilde{\pi}_{PS} = \left\{ \left(\mu_{PS,0}^u, \mu_{PS,0}^v \right), \dots, \left(\mu_{PS,N-1}^u, \mu_{PS,N-1}^v \right) \right\}, \quad (43)$$

with:

$$\mu_{PS,j}^v(x_j) \in \arg \max_{v_j} p_{v_j,j}(x_j) \text{ subject to: } v_j \in \mathbb{M}, \quad (44)$$

$$\mu_{PS,j}^u(x_j) \in \arg \min_{u_j} \left\| u_j - \mu_{PS,j}^u(x_j) \right\|_2^2 \text{ s.t.: } u_j \in \tilde{\mathcal{U}}_j^u(x_j, \mu_{PS,j}^v(x_j)). \quad (45)$$

The projection of an inadmissible input $\mu_{PS,j}^u(x_j) \notin \tilde{\mathcal{U}}_j^u(x_j, \mu_{PS,j}^v(x_j))$ onto the polytope $\tilde{\mathcal{U}}_j^u(x_j, \mu_{PS,j}^v(x_j))$ in (45) can be solved efficiently as QP. This is proposed in [31] to guarantee constraint satisfaction of neural network controllers—here, the approach guarantees the satisfaction of the constraints (35).

Training and Training Data Generation

The prediction of the optimal cost-to-go \tilde{J}_j^* by \tilde{J}_j for a state x_j at time j can be solved as a regression task. Assume for the moment that a parametric function \tilde{J}_j and a data set consisting of state-cost pairs $(x_j^s, J_j^s), s \in \{1, \dots, q_j^s\}$ are available. Each value J_j^s denotes a regression target that represents a cost sample for the corresponding sample state x_j^s . The parameter vector r_j^l can then be adapted with the aim to improve the performance on the considered regression task by *learning* from the data set. Of course, a performance measure is required hereto, and the mean-squared error is a typical choice. The adaption procedure, typically named *training*, is an instance of supervised learning, for which several established algorithms exist, see e.g. [39]. The parameter vectors r_j^u and r_j^v can be adapted by supervised learning, too, requiring that data sets $(x_j^s, u_j^s), s \in \{1, \dots, q_j^u\}$ and $(x_j^s, v_j^s), s \in \{1, \dots, q_j^v\}$ are available.

Algorithm 2 Sequential Dynamic Programming for Constrained DSLS [16]

Require: q_0, \dots, q_{N-1}

Ensure: r_j^J, r_j^u, r_j^v

- 1: **for** $j = N - 1$ to 0 **do**
 - 2: Generate a large number of states $x_j^s, s \in \{1, \dots, q_j\}$
 - 3: **for** $s = 1$ to q_j **do**
 - 4: $J_j^s \leftarrow \min_{v_j} \tilde{Q}_{VS,j}(x_j^s, v_j)$ subject to: $v_j \in \mathbb{M}$
 - 5: $u_j^s \leftarrow \mu_{VS,j}^u(x_j^s), \quad v_j^s \leftarrow \mu_{VS,j}^v(x_j^s)$
 - 6: **end for**
 - 7: Adapt r_j^\bullet by training with $(x_j^s, \bullet_j^s), s \in \{1, \dots, q_j\}$
 - 8: **end for**
-

The training data may originate from offline solutions of the considered MIQP problem. This approach, however, may take too much time due to the exponential growth of the number of possible discrete input sequences with N . An alternative is the use of approximate dynamic programming or reinforcement learning methods. The offline procedure in Algorithm 2 constitutes an approximate dynamic programming example that extends the sequential dynamic programming procedure from [30] to DSLS.

Neural Networks as Parametric Approximators

Figure 1 illustrates a feed-forward neural network that is characterized by a chain structure of the form:

$$h(x_j) = (h^{(L)} \circ \dots \circ h^{(2)} \circ h^{(1)})(x_j), \tag{46}$$

where $h^{(L)}$ denotes the final layer and $h^{(l)}$ the hidden layer $l \in \{1, \dots, L - 1\}$.

Further, $\eta^{(l)}$ denotes the output of layer l , and $\eta^{(0)}$ constitutes the input of the overall network:

$$\eta^{(0)}(x_j) = x_j, \tag{47}$$

$$\eta^{(l)}(x_j) = (h^{(l)} \circ \dots \circ h^{(1)})(x_j). \tag{48}$$

The hidden layers in Fig. 1 are vector-to-vector functions of the form

$$h^{(l)}(\eta^{(l-1)}) = (\phi^{(l)} \circ \psi^{(l)})(\eta^{(l-1)}), \tag{49}$$

with affine and nonlinear transformations $\psi^{(l)}$ and $\phi^{(l)}$, respectively. The affine transformation is affected by the choice of the weight matrix $W^{(l)}$ and the bias vector $b^{(l)}$:

$$\psi^{(l)}(\eta^{(l-1)}) = W^{(l)}\eta^{(l-1)} + b^{(l)}. \tag{50}$$

Each layer consists of parallel acting units, and a positive integer $S^{(l)}$ describes the number of units in layer l . Each unit i in layer l defines a vector-to-scalar function, which is the i -th component of $h^{(l)}$. For the hidden layers, $h_i^{(l)}(\eta^{(l-1)}) = \phi_i^{(l)}(W^{(l)}\eta^{(l-1)} + b^{(l)})$ with $\phi_i^{(l)}$ denoting an activation function. Typical choices are rectified linear units or sigmoid functions. For the purposes of this work, linear and softmax output units are considered. For a neural network with linear output units, the function $h^{(L)}$ is an affine transformation:

$$\psi^{(L)}(\eta^{(L-1)}) = W^{(L)}\eta^{(L-1)} + b^{(L)}. \tag{51}$$

Such an affine transformation arises also in softmax output units, in which $h_i^{(L)}$ is set to:

$$\text{softmax}_i(\psi^{(L)}(\eta^{(L-1)})) = \frac{\exp(\psi_i^{(L)}(\eta^{(L-1)}))}{\sum_{j=1}^{S^{(L)}} \exp(\psi_j^{(L)}(\eta^{(L-1)}))}. \tag{52}$$

The neural network (46) belongs to the family of parametric functions, whose shape is formed by the parameter vector that consists of the weights and biases:

$$r = [W_{1,1}^{(1)} \dots W_{S^{(L)},S^{(L-1)}}^{(L)} \ b_1^{(1)} \dots b_{S^{(L)}}^{(L)}]^T. \tag{53}$$

Subsequently, the neural network structure (46) is considered as basis for parametric approximators. For the cost-to-go function approximators \tilde{J}_j , the use of continuous and continuously differentiable activation functions (such as sigmoid functions) and linear output units is proposed. As shown in [38], this allows one to derive closed-form expressions for the partial derivatives of h with respect to its arguments:

$$\begin{aligned} \frac{\partial h(x_j)}{\partial x_j} &= \prod_{i=0}^{L-1} \frac{\partial h^{(L-i)}(\eta^{(L-(i+1))}(x_j))}{\partial \eta^{(L-(i+1))}}, \\ \frac{\partial h^{(l)}(\eta^{(l-1)}(x_j))}{\partial \eta^{(l-1)}} &= \frac{\partial \phi^{(l)}(\psi^{(l)}(\eta^{(l-1)}(x_j)))}{\partial \psi^{(l)}} \cdot W^{(l)}, \\ \frac{\partial h^{(l)}(\eta^{(l-1)}(x_j))}{\partial \eta^{(L-1)}} &= W^{(L)}, \quad l \in \{1, \dots, L - 1\}. \end{aligned} \tag{54}$$

Linear output units are further proposed for establishing $\mu_{PS_j}^{\tilde{u}}$. Here, it is not necessarily required that the activation functions are continuous and continuously differentiable. The softmax output unit, on the other hand, is proposed as output unit for p_j . It is common to use softmax units as output units to represent probability distributions over different classes [39]. According to (52), each output of the neural network with softmax output units is in between 0 and 1, and all outputs sum up to 1, leading to a valid probability distribution.

Receding-Horizon Control with Parametric Function Approximators

The RHC strategy (23) can be computed by solving a QP for each possible discrete input sequence. As already mentioned, this procedure, however, becomes computationally intractable rapidly for increasing N , due to the exponential growth of the possible number of discrete input sequences. The approach presented in this section aims at approximating the RHC strategy to make the online application possible, and is based on the idea to solve a QP only for a small number of discrete input sequences. Of course, a procedure is desirable which selects those discrete input sequences which are promising candidates for being the true optimal

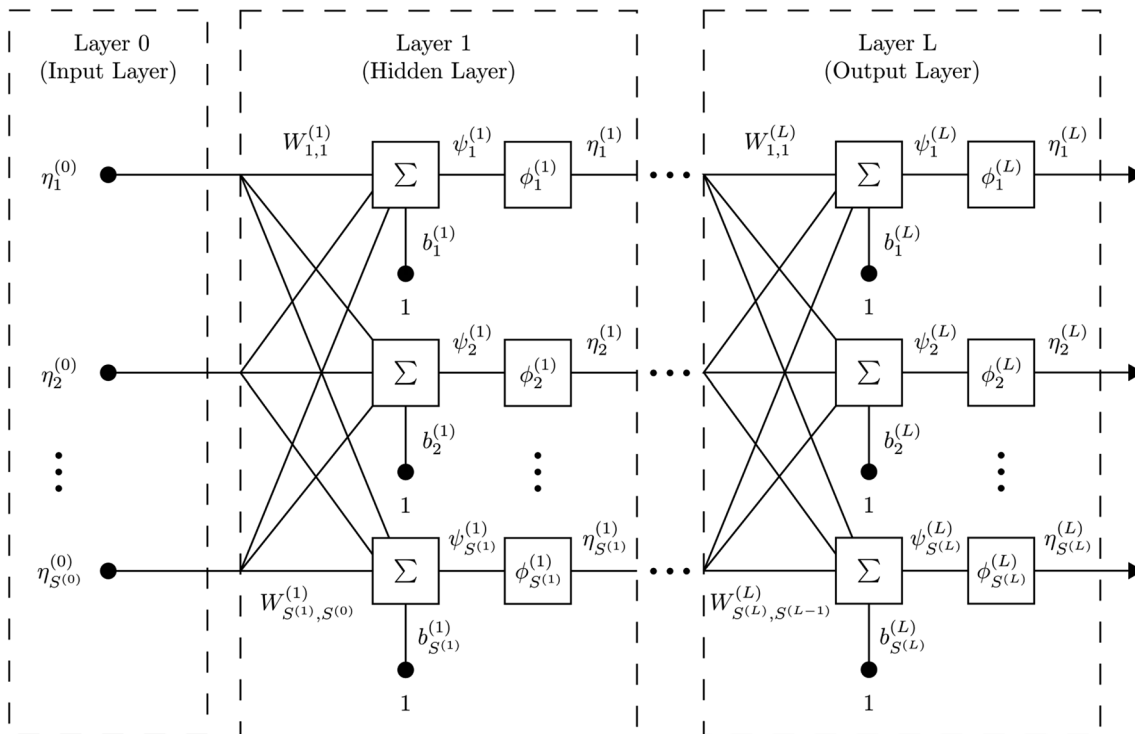


Fig. 1 Architecture of a feed-forward neural network

one(s). The procedure proposed in this section is based on the ideas for approximating the finite-horizon control law by neural networks, as presented in the previous section.

Let $\mathcal{V}(k)$ denote a small set of selected discrete input sequences at time k —suppose for a moment that this set was available. For a given state x_k , the approach computes the input sequences $\phi_{0|k}^{\tilde{v}} = (\tilde{v}_{0|k}, \dots, \tilde{v}_{N-1|k})$ and $\phi_{0|k}^{\tilde{u}} = (\tilde{u}_{0|k}, \dots, \tilde{u}_{N-1|k})$ by solving the QP defined in (25) for each discrete input sequence $\phi_{0|k}^v \in \mathcal{V}(k)$:

$$\phi_{0|k}^{\tilde{v}} = \Phi_0^{\tilde{v}}(x_k) \in \arg \min_{\phi_{0|k}^v} \tilde{V}_0^*(x_k, \phi_{0|k}^v) \text{ subject to: } \phi_{0|k}^v \in \mathcal{V}(k) \quad (55)$$

$$\phi_{0|k}^{\tilde{u}} = \Phi_0^{\tilde{u}}(x_k) \in \arg \min_{\phi_{0|k}^u} J_0(x_k, \phi_{0|k}^u, \Phi_0^{\tilde{v}}(x_k)) \text{ subject to: 25b.} \quad (56)$$

The approximated RHC strategy is obtained by applying the first element $\tilde{u}_{0|k}$ of the continuous input sequence $\phi_{0|k}^{\tilde{u}}$ and the first element $\tilde{v}_{0|k}$ of the discrete input sequence $\phi_{0|k}^{\tilde{v}}$ to the DSLS (1) at time k :

$$(u_k, v_k) = (\tilde{u}_{0|k}, \tilde{v}_{0|k}) =: (\mu_0^{\tilde{u}}(x_k), \mu_0^{\tilde{v}}(x_k)). \quad (57)$$

The closed-loop dynamics for the DSLS (1) controlled by the approximated RHC strategy is then:

$$x_{k+1} = f(x_k, \mu_0^{\tilde{u}}(x_k), \mu_0^{\tilde{v}}(x_k)) =: \tilde{f}_{\text{cl}}(x_k), \quad k \in \mathbb{N}_0. \quad (58)$$

Subsequently, $\tilde{J}_{\text{RHC},0}$ is defined as:

$$\tilde{J}_{\text{RHC},0}(x_k) := \tilde{V}_0^*(x_k, \Phi_0^{\tilde{v}}(x_k)) = J_0(x_k, \Phi_0^{\tilde{u}}(x_k), \Phi_0^{\tilde{v}}(x_k)), \quad (59)$$

and constitutes, obviously, an upper bound to the optimal cost-to-go J_0^* .

For the determination of $\mathcal{V}(k)$ at time k , M^ℓ different discrete input sequences $(v_{0|k}^{[i]}, \dots, v_{\ell-1|k}^{[i]}, \dots, v_{N-1|k}^{[i]})$, $i \in \{1, \dots, M^\ell\}$ are generated by a combination of *approximation in value space* and *approximation in policy space*, as described next. First, for each possible subsequence $(v_{0|k}^{[i]}, \dots, v_{\ell-1|k}^{[i]}) \in \mathbb{M}^\ell$, the state $x_{\ell|k}^{[i]}$ is determined recursively as illustrated in Fig. 2:

$$x_{j+1|k}^{[i]} = f(x_{j|k}^{[i]}, \xi_{\text{VS},j}^{\tilde{u}}(x_{j|k}^{[i]}, v_{j|k}^{[i]}), v_{j|k}^{[i]}), \quad j \in \{0, \dots, \ell-1\}. \quad (60)$$

Here, $x_{0|k}^{[i]}$ is the current state x_k at time k , i.e. $x_{0|k}^{[i]} = x_k$. Recall that the value of function $\xi_{\text{VS},j}^{\tilde{u}}$ for state $x_{j|k}^{[i]}$ and discrete input $v_{j|k}^{[i]}$ results according to (39) from the solution of the nonlinear program (38) with convex constraints. The application of well-established gradient methods is possible here due to the availability of the closed-form expression for

the gradient of the neural network \tilde{J}_{j+1} , as specified in (54). The remaining subsequence $(v_{\ell+1|k}^{[i]}, \dots, v_{N-1|k}^{[i]})$ follows from the approximated finite-horizon control law specified in (43):

$$x_{j+1|k}^{[i]} = f(x_{j|k}^{[i]}, \mu_{\text{PS},j}^{\tilde{u}}(x_{j|k}^{[i]}), \mu_{\text{PS},j}^{\tilde{v}}(x_{j|k}^{[i]})), \quad j \in \{\ell, \dots, N-1\}. \tag{61}$$

In addition, one further discrete input sequence $(v_{0|k}^{[0]}, \dots, v_{N-1|k}^{[0]})$ is selected, and chosen to guarantee asymptotic stability. Motivated by the proof of Proposition 3, $(v_{0|k}^{[0]}, \dots, v_{N-2|k}^{[0]}, v_{N-1|k}^{[0]})$ is set to $(\tilde{v}_{1|k-1}, \dots, \tilde{v}_{N-1|k-1}, 1)$ for $k > 0$. The discrete input sequence $(v_{0|k}^{[0]}, \dots, v_{N-1|k}^{[0]})$ for $k = 0$, on the other hand, is arbitrarily selected from \mathbb{M}^N . Algorithm 3 summarizes the procedure of determining the set $\mathcal{V}(k)$.

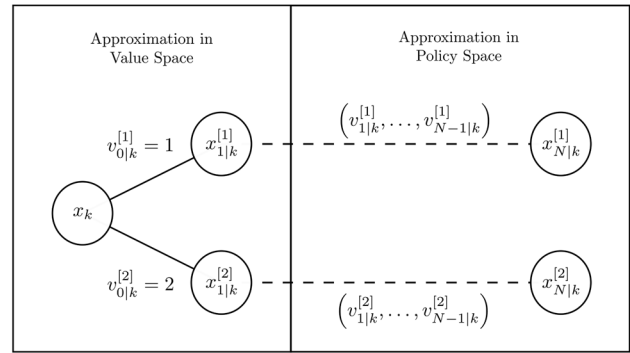


Fig. 2 Generation of discrete input sequences for the determination of $\mathcal{V}(k)$ with $\ell = 1$

$\phi_{0|k}^v \in \mathcal{V}(k)$. The definition of $\tilde{\mathcal{X}}_0$ ensures that the QP has a feasible solution for any $\phi_{0|k}^v \in \mathbb{M}^N \supseteq \mathcal{V}(k)$ if $x_k \in \tilde{\mathcal{X}}_0$, such that $x_k \in \tilde{\mathcal{X}}_0$ ensures feasibility of the approximated RHC strategy, too. If feasible, the constraints of the QP enforce

Algorithm 3 Determination of $\mathcal{V}(k)$

```

Require:  $\ell, x_k, (v_{0|k}^{[0]}, \dots, v_{N-1|k}^{[0]})$ 
Ensure:  $\mathcal{V}(k)$ 
1:  $\mathcal{V}(k) \leftarrow \emptyset, \quad \mathcal{V}(k) \leftarrow \mathcal{V}(k) \cup (v_{0|k}^{[0]}, \dots, v_{N-1|k}^{[0]})$ ,  $i \leftarrow 1$ 
2: for all  $(v_{0|k}, \dots, v_{\ell-1|k}) \in \mathbb{M}^\ell$  do
3:    $x_{0|k}^{[i]} \leftarrow x_k, \quad (v_{0|k}^{[i]}, \dots, v_{\ell-1|k}^{[i]}) \leftarrow (v_{0|k}, \dots, v_{\ell-1|k})$ 
4:   for  $j = 0$  to  $N - 1$  do
5:     if  $j < \ell$  then
6:        $u_{j|k}^{[i]} \leftarrow \xi_{\text{VS},j}^{\tilde{u}}(x_{j|k}^{[i]}, v_{j|k}^{[i]})$ 
7:     else if  $j \geq \ell$  then
8:        $u_{j|k}^{[i]} \leftarrow \mu_{\text{PS},j}^{\tilde{u}}(x_{j|k}^{[i]})$ 
9:        $v_{j|k}^{[i]} \leftarrow \mu_{\text{PS},j}^{\tilde{v}}(x_{j|k}^{[i]})$ 
10:    end if
11:     $x_{j+1|k}^{[i]} \leftarrow f(x_{j|k}^{[i]}, u_{j|k}^{[i]}, v_{j|k}^{[i]})$ 
12:  end for
13:   $\mathcal{V}(k) \leftarrow \mathcal{V}(k) \cup (v_{0|k}^{[i]}, \dots, v_{N-1|k}^{[i]})$ ,  $i \leftarrow i + 1$ 
14: end for
    
```

Theorem 1 For the DSLS (1) with constraints (2), let $\mathcal{V}(k)$ be determined by Algorithm 3. Then, the approximated RHC strategy (57) is recursively feasible for $x_0 \in \tilde{\mathcal{X}}_0$, and the origin of the closed-loop system (58) is asymptotically stable with domain of attraction $\tilde{\mathcal{X}}_0$.

Proof of Theorem 1 Given x_k at time k , the approximated RHC strategy solves the QP defined in (25) for all

that $\tilde{f}_{c1}(x_k) \subseteq \tilde{\mathcal{X}}_1$. It follows immediately from Propositions 1 and (21) that $\tilde{\mathcal{X}}_0 \supseteq \tilde{\mathcal{X}}_1$, such that recursive feasibility of the approximated RHC strategy is guaranteed for $x_0 \in \tilde{\mathcal{X}}_0$.

Now consider an $x_0 \in \tilde{\mathcal{X}}_0$, and let $\phi_{0|0}^{\tilde{x}} = (\tilde{x}_{0|0}, \tilde{x}_{1|0}, \dots, \tilde{x}_{N|0})$ be the state sequence with $\tilde{x}_{N|0} = 0$ and $\tilde{x}_{0|0} := x_0$ that results from the input sequences $\phi_{0|0}^{\tilde{u}} = \Phi_0^{\tilde{u}}(x_0)$ and $\phi_{0|0}^{\tilde{v}} = \Phi_0^{\tilde{v}}(x_0)$. Hence, one gets:

$$\begin{aligned} \tilde{J}_{\text{RHC},0}(x_0) &= g_N(\tilde{x}_{N|0}) + \sum_{j=0}^{N-1} g(\tilde{x}_{j|0}, \tilde{u}_{j|0}, \tilde{v}_{j|0}) \\ &= \sum_{j=0}^{N-1} g(\tilde{x}_{j|0}, \tilde{u}_{j|0}, \tilde{v}_{j|0}). \end{aligned}$$

Due to $u_0 = \tilde{u}_{0|0}$ and $v_0 = \tilde{v}_{0|0}$, it follows from (1) that $x_1 = f(x_0, \tilde{u}_{0|0}, \tilde{v}_{0|0})$, such that $x_1 = \tilde{x}_{1|0}$.

The state sequence $\phi_{0|1}^x = (x_{0|1}, \dots, x_{N-1|1}, x_{N|1}) := (\tilde{x}_{1|0}, \dots, \tilde{x}_{N|0}, 0)$ corresponds to the continuous input sequence $\phi_{0|1}^u = (u_{0|1}, \dots, u_{N-2|1}, u_{N-1|1}) := (\tilde{u}_{1|0}, \dots, \tilde{u}_{N-1|0}, 0)$ and the discrete input sequence $\phi_{0|1}^v = (v_{0|1}, \dots, v_{N-1|1}) := (v_{0|1}^{[0]}, \dots, v_{N-2|1}^{[0]}, v_{N-1|1}^{[0]}) = (\tilde{v}_{1|0}, \dots, \tilde{v}_{N-1|0}, 1)$. Since the sequences $\phi_{0|1}^x$ and $\phi_{0|1}^v$ satisfy $x_{j|1} \in \tilde{\mathcal{X}}_j$ and $u_{j|1} \in \mathcal{U}$, respectively, they are admissible, such that the cost:

$$\begin{aligned} J_0(x_1, \phi_{0|1}^u, \phi_{0|1}^v) &= g_N(x_{N|1}) + \sum_{j=0}^{N-1} g(x_{j+1}, u_{j+1}, v_{j+1}) \\ &= \sum_{j=0}^{N-2} g(x_{j+1}, u_{j+1}, v_{j+1}) \\ &= \sum_{j=1}^{N-1} g(\tilde{x}_{j|0}, \tilde{u}_{j|0}, \tilde{v}_{j|0}) \\ &= \tilde{J}_{\text{RHC},0}(x_0) - g(\tilde{x}_{0|0}, \tilde{u}_{0|0}, \tilde{v}_{0|0}) \end{aligned}$$

constitutes an upper bound on $\tilde{V}^*(x_1, \phi_{0|1}^v)$. On the other hand, $\phi_{0|1}^v \in \mathcal{V}(1)$, such that $\tilde{V}^*(x_1, \phi_{0|1}^v)$ constitutes an upper bound on $\tilde{J}_{\text{RHC},0}(x_1) = \tilde{V}^*(x_1, \phi_{0|1}^v)$. Hence:

$$\begin{aligned} \tilde{J}_{\text{RHC},0}(x_1) \leq J_0(x_1, \phi_{0|1}^u, \phi_{0|1}^v) &= \tilde{J}_{\text{RHC},0}(x_0) \\ &\quad - g(\tilde{x}_{0|0}, \tilde{u}_{0|0}, \tilde{v}_{0|0}), \end{aligned}$$

and it follows from induction that:

$$\begin{aligned} \tilde{J}_{\text{RHC},0}(\tilde{f}_{\text{cl}}(x_k)) - \tilde{J}_{\text{RHC},0}(x_k) &\leq -g(x_k, \mu_0^u(x_k), \mu_0^v(x_k)), \\ \forall x_k \in \tilde{\mathcal{X}}_0. \end{aligned} \tag{62}$$

Since $x_k \in \tilde{\mathcal{X}}_0$ implies $\tilde{f}_{\text{cl}}(x_k) \in \tilde{\mathcal{X}}_0$, the state sequence of the closed-loop system (58) lies within $\tilde{\mathcal{X}}_0$ for any $x_0 \in \tilde{\mathcal{X}}_0$. Note that the stage cost g and the terminal cost g_N are continuous and positive definite functions, and that $\tilde{J}_{\text{RHC},0}(x_k)$ is lower bounded by zero. Hence, $\tilde{J}_{\text{RHC},0}(x_k)$ decreases according to (62) along any state sequence that starts from $\tilde{\mathcal{X}}_0$, i.e., convergence to the origin without leaving $\tilde{\mathcal{X}}_0$ is guaranteed as $k \rightarrow \infty$. \square

Numerical Example

This section provides a numerical example for the illustration and the evaluation of the proposed approach, inspired by the numerical example considered in [16] for the finite-horizon case.

The switched system (1) is parameterized by the matrices:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 \\ -0.8 & 2.4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ -1.8 & 3.6 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 \\ -0.56 & 1.8 \end{bmatrix}, \\ A_4 &= \begin{bmatrix} 0 & 1 \\ -8 & 6 \end{bmatrix}, \quad B_1 = B_2 = B_3 = B_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \end{aligned}$$

and is subject to polytopic constraints (2) with $\mathcal{X} = \{x \in \mathbb{R}^2 \mid |x_i| \leq 1\}$ and $\mathcal{U} = \{u \in \mathbb{R} \mid |u| \leq 4\}$. Furthermore, a quadratic cost function of type (3) is chosen with prediction horizon $N = 6$ and:

$$\begin{aligned} P &= Q_1 = Q_2 = Q_3 = Q_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ R_1 &= R_2 = R_3 = R_4 = 1. \end{aligned}$$

As in [16], all neural networks required for the proposed approach are chosen to consist of one hidden layer with 50 units (i.e. $S^{(1)} = 50$). In each hidden unit, hyperbolic tangents have been chosen as activation function. The neural networks have been trained according to Algorithm 2 offline by choosing $q_j = 1000$.

To evaluate the approximation quality of the approximated RHC strategy, 1000 states x^p have been generated by gridding the set $\tilde{\mathcal{X}}_0$. The latter has been determined with Algorithm 1 and is marked by the shaded polytope in Fig. 3. For each x^p , the optimal cost-to-go $\tilde{J}_0^*(x^p)$ and its approximation $\tilde{J}_{\text{RHC},0}(x^p)$ for $\ell = 1$ have been computed. The distribution of the optimal cost $\tilde{J}_0^*(x^p)$ for the different states is shown in Fig. 4. The comparison of the optimal cost $\tilde{J}_0^*(x^p)$ with its approximation $\tilde{J}_{\text{RHC},0}(x^p)$ yield a mean-squared error of only 6.99×10^{-5} .

The average online computation time for the determination of the optimal costs by complete enumeration over all 4^N possible discrete input sequences (and solving one QP each) was 19.8s on a standard notebook (Intel® Core™ i5 – 7200U Processor), where the CPLEXQP solver from the IBM® ILOG® CPLEX® Optimization Studio has been used for the solution of the QPs. In contrast, when applying the proposed scheme, the average online computation time for determining $\tilde{J}_{\text{RHC},0}(x^p)$ (and thus the control inputs) was only 0.96s.

Figure 3 shows a state sequence obtained from the approximated RHC strategy for an exemplary initial state x_0 , and demonstrates the asymptotic stability of the origin as proven in Theorem 1.

Fig. 3 Example of a state sequence obtained from the approximated RHC strategy for $x_0 = [0.125 \ 1]^T$. The shaded polytope marks X_0

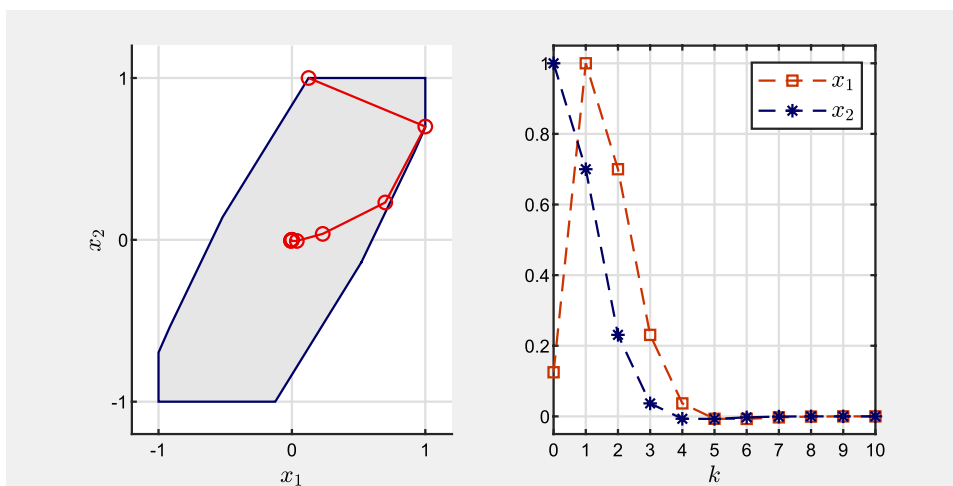
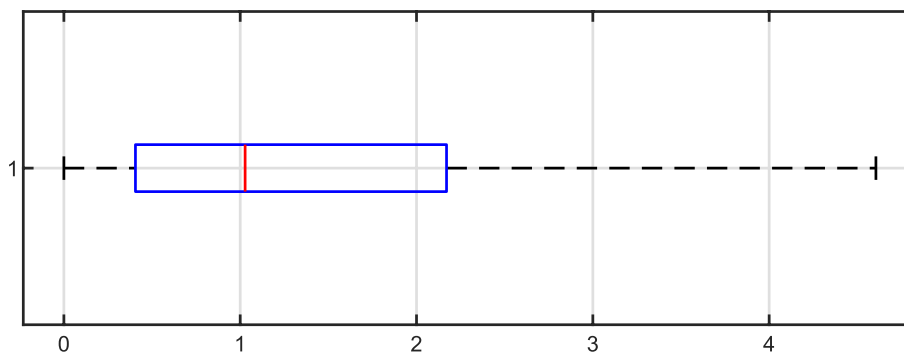


Fig. 4 Box plot diagram showing the distribution of the optimal costs $\tilde{J}_0^*(x^p)$ for 1000 states x^p obtained by gridding of $\tilde{\mathcal{X}}_0$



Conclusion

This paper has considered the optimal control of discrete-time constrained switched linear systems (with externally forced switching) using the principle of the receding-horizon control. Building on previous work of the authors, an approach for approximating the optimal receding-horizon control strategy with neural networks as parametric function approximators has been developed. Important properties such as the guaranteed satisfaction of the polytopic constraints, recursive feasibility, and asymptotic stability of the origin of the closed-loop system under the approximated receding-horizon control strategy have been proven. The numerical example has shown that the proposed approach allows for the computation of approximating but close-to-optimal receding-horizon control strategies, which is considerable faster than solving the same problem by mixed-integer quadratic programming in any step.

The focus of this work has been to provide theoretical guarantees for the proposed approach, without putting the focus on reducing the speed of computation through efficient implementation. A streamlined implementation of the proposed algorithm certainly allows for further reduction of the computation. Furthermore, an in-depth treatment of an

efficient training data generation has been out of the scope of this work. Investigating alternative schemes to sequential dynamic programming for efficient generation of training data is a worthwhile point of future work for improving the applicability of the proposed scheme.

Appendix 1

Proofs of the Propositions

Proof of Proposition 1 Suppose that $\mathcal{X}_{j+1}^{(v_{j+1}, \dots, v_{N-1})}$ is a polytope. Then, $\text{Pre}^{(v_j)}(\mathcal{X}_{j+1}^{(v_{j+1}, \dots, v_{N-1})})$ is the result of linear operations on the polytopes $\mathcal{X}_j^{(v_{j+1}, \dots, v_{N-1})}$ and \mathcal{U} (see e.g. [35, Chapter 10]), and hence a polytope, too. Moreover, since \mathcal{X} is a polytope by definition, $\mathcal{X}_j^{(v_j, \dots, v_{N-1})} = \text{Pre}^{(v_j)}(\mathcal{X}_{j+1}^{(v_{j+1}, \dots, v_{N-1})}) \cap \mathcal{X}$ is polytopic as well. Here, the terminal set $\mathcal{X}_N \subseteq \mathcal{X}$ is a singleton that contains only the origin. Hence, $\mathcal{X}_{N-1}^{(v_{N-1})} = \text{Pre}^{(v_{N-1})}(\mathcal{X}_N) \cap \mathcal{X}$ is a polytope, and the fact that the sets $\mathcal{X}_j^{(v_{j+1}, \dots, v_{N-1})}$ are polytopes, too, follows by induction. The union of convex sets (including polytopes) may be

non-convex, however, such that the sets \mathcal{X}_j are non-convex in the general case.

Let $\mathcal{S} \subseteq \mathcal{X}$ be a *control-invariant set* for the DSLS (1) subject to the constraints (2) if:

$$x_k \in \mathcal{S} \Rightarrow \exists u_k \in \mathcal{U}, \exists v_k \in \mathbb{M} \text{ such that } f(x_k, u_k, v_k) \in \mathcal{S} \text{ for all } k \in \mathbb{N}_0.$$

Suppose that \mathcal{X}_{j+1} is control-invariant. According to the definition, \mathcal{X}_j is the set of states $x_{jk} \in \mathcal{X}_j$ for which at least one $u_{jk} \in \mathcal{U}$ and at least one $v_{jk} \in \mathbb{M}$ exist such that $f(x_{jk}, u_{jk}, v_{jk}) \in \mathcal{X}_{j+1}$. If \mathcal{X}_{j+1} is control-invariant, then for each $x_{jk} \in \mathcal{X}_{j+1}$ at least one $u_{jk} \in \mathcal{U}$ and at least one $v_{jk} \in \mathbb{M}$ exist such that $f(x_{jk}, u_{jk}, v_{jk})$ is in \mathcal{X}_{j+1} again. Consequently, \mathcal{X}_{j+1} is a subset of \mathcal{X}_j , i.e. $\mathcal{X}_j \supseteq \mathcal{X}_{j+1}$. Moreover, \mathcal{X}_j is control-invariant, since it is always possible to reach the control-invariant set \mathcal{X}_{j+1} in one time-step and to remain in there up to step N . Here, \mathcal{X}_N is a singleton that contains only the origin. Furthermore, the origin is in the interior of \mathcal{U} . Hence, $\bar{x} = f(\bar{x}, \bar{u}, \bar{v})$ for $\bar{x} = 0, \bar{u} = 0$, and arbitrary $\bar{v} \in \mathbb{M}$, such that \mathcal{X}_N is control-invariant according to the definition above. The fact that $\mathcal{X}_j \supseteq \mathcal{X}_{j+1}$ follows thus by induction. \square

Proof of Proposition 2 Problem 1 is feasible for $x_k \in \mathcal{X}_0$. From the definition of the RHC strategy follows that $x_k \in \mathcal{X}_0$ implies $x_{k+1} \in \mathcal{X}_1$. Hence, a sufficient condition for recursive feasibility is that $\mathcal{X}_0 \supseteq \mathcal{X}_1$. It follows from Proposition 1 and (11) that $\mathcal{X}_0 \supseteq \mathcal{X}_1$. \square

Proof of Proposition 3 (The proof is similar to the one for systems without switching, as can be found in [35, Chapter 12].) Let x_0 be an element of \mathcal{X}_0 , such that recursive feasibility of the RHC strategy is guaranteed by Proposition 2. Further, let $\phi_{0|0}^{u^*} = (u_{0|0}^*, \dots, u_{N-1|0}^*)$ and $\phi_{0|0}^{v^*} = (v_{0|0}^*, \dots, v_{N-1|0}^*)$ be the optimal solution of Problem 1 for the current state x_0 at time 0, and $(x_{0|0}^*, x_{1|0}^*, \dots, x_{N|0}^*)$ the corresponding state sequence with $x_{0|0}^* = x_0$ and $x_{N|0}^* = 0$. Hence, the optimal cost $J_0^*(x_0)$ is:

$$J_0^*(x_0) = g_N(x_{N|0}^*) + \sum_{j=0}^{N-1} g(x_{j|0}^*, u_{j|0}^*, v_{j|0}^*) \\ = \sum_{j=0}^{N-1} g(x_{j|0}^*, u_{j|0}^*, v_{j|0}^*).$$

Since $u_0 = u_{0|0}^*$ and $v_0 = v_{0|0}^*$, it follows from (1) that $x_1 = f(x_0, u_{0|0}^*, v_{0|0}^*)$, such that $x_1 = x_{1|0}^*$.

The state sequence $(x_{0|1}, \dots, x_{N-1|1}, x_{N|1}) := (x_{1|0}^*, \dots, x_{N|0}^*, 0)$ corresponds to the input sequences $\phi_{0|1}^u = (u_{0|1}, \dots, u_{N-1|1}) := (u_{1|0}^*, \dots, u_{N-1|0}^*, 0)$ and $\phi_{0|1}^v =$

$(v_{0|1}, \dots, v_{N-1|1}) := (v_{1|0}^*, \dots, v_{N-1|0}^*, 1)$. The sequences $\phi_{0|1}^u$ and $\phi_{0|1}^v$ are feasible, but generally not an optimal solution of Problem 1 for the current state $x_1 = x_{0|1}$ at time 1, such that:

$$J_0(x_1, \phi_{0|1}^u, \phi_{0|1}^v) = g_N(x_{N|1}) + \sum_{j=0}^{N-1} g(x_{j|1}, u_{j|1}, v_{j|1}) \\ = \sum_{j=0}^{N-2} g(x_{j|1}, u_{j|1}, v_{j|1}) \\ = \sum_{j=1}^{N-1} g(x_{j|0}^*, u_{j|0}^*, v_{j|0}^*) \\ = J_0^*(x_0) - g(x_{0|0}^*, u_{0|0}^*, v_{0|0}^*)$$

is an upper bound on $J_0^*(x_1)$. Hence:

$$J_0^*(x_1) \leq J_0(x_1, \phi_{0|1}^u, \phi_{0|1}^v) = J_0^*(x_0) - g(x_{0|0}^*, u_{0|0}^*, v_{0|0}^*),$$

and it follows from induction that

$$J_0^*(f_{cl}^*(x_k)) - J_0^*(x_k) \leq -g(x_k, \mu_0^{u^*}(x_k), \mu_0^{v^*}(x_k)), \quad \forall x_k \in \mathcal{X}_0. \quad (\text{A1})$$

Since $x_k \in \mathcal{X}_0$ implies $f_{cl}^*(x_k) \in \mathcal{X}_0$, the state sequence of the closed-loop system (8) lies within \mathcal{X}_0 for any $x_0 \in \mathcal{X}_0$. Note that the stage cost g and the terminal cost g_N are continuous and positive definite functions, and $J_0^*(x_k)$ is lower bounded by zero. Hence, $J_0^*(x_k)$ decreases according to (A1) along any state sequence that starts from \mathcal{X}_0 , such that convergence to the origin without leaving \mathcal{X}_0 is guaranteed as $t \rightarrow \infty$. \square

Proof of Proposition 4 Let $J_j^{*(v_j, \dots, v_{N-1})}$ be the value function V_j^* for a fixed discrete input sequence $(v_j, \dots, v_{N-1}) \in \mathbb{M}^{N-j}$. The optimization problem in (12) subject to the fixed discrete input sequence can then be transformed into a multi-parametric QP (mp-QP) problem of the form:

$$J_j^{*(v_j, \dots, v_{N-1})}(x_{jk}) = \min_{\phi_{jk}^u} \left(\begin{bmatrix} u_{j|k}^T & \dots & u_{N-1|k}^T \end{bmatrix} H_j \begin{bmatrix} u_{j|k}^T & \dots & u_{N-1|k}^T \end{bmatrix}^T \right. \\ \left. + 2x_{jk}^T F_j \begin{bmatrix} u_{j|k}^T & \dots & u_{N-1|k}^T \end{bmatrix}^T + x_{jk}^T Y_j x_{jk} \right) \\ \text{subject to: } G_j \begin{bmatrix} u_{j|k}^T & \dots & u_{N-1|k}^T \end{bmatrix}^T \leq w_j + E_j x_{jk}.$$

In here, $H_j = H_j^T > 0$, F_j , Y_j , G_j , w_j , and E_j are similarly defined as in [35, Chapter 11]. It is proven there that the value function obtained as solution of the mp-QP problem is convex and piecewise quadratic on polyhedra. This transfers here to the function $J_j^{*(v_j, \dots, v_{N-1})}$. According to (13), $J_{j \rightarrow N}^*$ is the pointwise minimum over the functions $J_j^{*(v_j, \dots, v_{N-1})}$,

$(v_j, \dots, v_{N-1}) \in \mathbb{M}^{N-j}$, and hence the pointwise minimum over functions that are convex and piecewise quadratic on polyhedra. \square

Proof of Proposition 5 Denote by $\mathcal{U}_j^{\phi^v}$ the mapping which assigns the set of admissible continuous inputs to state $x_{j|k}$ and the input sequence $\phi_{j|k}^v$:

$$\mathcal{U}_j^{\phi^v}(x_{j|k}, \phi_{j|k}^v) = \left\{ u \in \mathcal{U} \mid f(x_{j|k}, u, v_{j|k}) \in \mathcal{X}_{j+1}^{(v_{j+1|k}, \dots, v_{N-1|k})} \right\}.$$

Since \mathcal{U} and $\mathcal{X}_{j+1}^{(v_{j+1|k}, \dots, v_{N-1|k})}$ are polytopes, $\mathcal{U}_j^{\phi^v}(x_{j|k}, \phi_{j|k}^v)$ is a polytope as well. The set $\mathcal{U}_j^v(x_{j|k}, v_{j|k})$ is given by:

$$\mathcal{U}_j^v(x_{j|k}, v_{j|k}) = \bigcup_{(v_{j+1|k}, \dots, v_{N-1|k}) \in \mathbb{M}^{N-(j+1)}} \mathcal{U}_j^{\phi^v}(x_{j|k}, (v_{j|k}, v_{j+1|k}, \dots, v_{N-1|k})),$$

and thus constitutes the union of polytopes. Polytopes are convex sets, and the union of convex sets is non-convex in the general case. \square

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Tarraf DC. Control of cyber-physical systems. In: Lecture Notes in Control and Information Sciences, vol. 449. Heidelberg: Springer; 2013.
2. Lunze J, Lamnabhi-Lagarigue F. Handbook of hybrid systems control: theory, tools, applications. Cambridge: Cambridge University Press; 2009.
3. Zhu F, Antsaklis PJ. Optimal control of hybrid switched systems: a brief survey. *Discrete Event Dyn Syst*. 2015;25(3):345–64.
4. Görges D. Optimal control of switched systems with application to networked embedded control systems. Berlin: Logos Verlag; 2012.

5. Zhang W, Hu J, Abate A. On the value functions of the discrete-time switched lqr problem. *IEEE Trans Autom Control*. 2009;54(11):2669–74.
6. Zhang W, Hu J, Abate A. Infinite-horizon switched lqr problems in discrete time: a suboptimal algorithm with performance analysis. *IEEE Trans Autom Control*. 2012;57(7):1815–21.
7. Zhang W, Hu J. On optimal quadratic regulation for discrete-time switched linear systems. In: Proceedings of the 11th International Conference on hybrid systems: computation and control, 2008; pp. 584–97.
8. Lincoln B, Rantzer A. Relaxed dynamic programming. *IEEE Trans Autom Control*. 2006;51(8):1249–60.
9. Rantzer A. Relaxed dynamic programming in switching systems. *IEE Proc Control Theory Appl*. 2006;153(5):567–74.
10. Görges D, Izak M, Liu S. Optimal control and scheduling of switched systems. *IEEE Trans Autom Control*. 2011;56(1):135–40.
11. Antunes D, Heemels WPMH. Performance analysis of a class of linear quadratic regulators for switched linear systems. In: Proceedings of the 53rd IEEE Conference on decision and control, 2014; pp. 5475–80.
12. Chen H, Zheng L, Zhang W. Optimal control inspired q-learning for switched linear systems. In: Proceedings of the 2020 American Control Conference, 2020; pp. 4003–10.
13. Grüne L, Rantzer A. On the infinite horizon performance of receding horizon controllers. *IEEE Trans Autom Control*. 2008;53(9):2100–11.
14. Balandat M, Zhang W, Abate A. On infinite horizon switched lqr problems with state and control constraints. *Syst Control Lett*. 2012;61(4):464–71.
15. Liu Z, Stursberg O. Optimizing online control of constrained systems with switched dynamics. In: Proceedings of the 2018 European Control Conference, 2018; pp. 788–94.
16. Markolf L, Stursberg O. Learning-based optimal control of constrained switched linear systems using neural networks. In: Proceedings of the 18th International Conference on informatics in control, automation and robotics, 2021; pp. 90–98.
17. Cybenko G. Approximation by superpositions of a sigmoidal function. *Math Control Signals Syst*. 1989;2(4):303–14.
18. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Netw*. 1989;2(5):359–66.
19. Leshno M, Lin VY, Pinkus A, Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw*. 1993;6(6):861–7.
20. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540):529–33.
21. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529(7587):484–9.
22. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D. Mastering the game of Go without human knowledge. *Nature*. 2017;550(7676):354–9.
23. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, Lillicrap T, Simonyan K, Hassabis D. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*. 2018;362(6419):1140–4.

24. Bertsekas DP, Tsitsiklis JN. *Neuro-dynamic programming*. Belmont: Athena Scientific; 1996.
25. Lendaris GG. A retrospective on adaptive dynamic programming for control. In: *Proceedings of the International Joint Conference on neural networks, 2009*; pp. 1750–57.
26. Lewis FL, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circ Syst Mag*. 2009;9(3):32–50.
27. Gaggero M, Gnecco G, Sanguineti M. Dynamic programming and value-function approximation in sequential decision problems: error analysis and numerical results. *J Optim Theory Appl*. 2013;156(2):380–416.
28. Gaggero M, Gnecco G, Sanguineti M. Approximate dynamic programming for stochastic N-stage optimization with application to optimal consumption under uncertainty. *Comput Optim Appl*. 2014;58(1):31–85.
29. Sutton RS, Barto A. *Reinforcement learning: an introduction*. Cambridge: MIT Press; 2018.
30. Bertsekas DP. *Reinforcement learning and optimal control*. Belmont: Athena Scientific; 2019.
31. Chen S, Saulnier K, Atanasov N, Lee DD, Kumar V, Pappas GJ, Morari M. Approximating explicit model predictive control using constrained neural networks. In: *Proceedings of the 2018 Annual American Control Conference, 2018*; pp. 1520–27.
32. Cervellera C, Maccio D, Parisini T. Learning robustly stabilizing explicit model predictive controllers: a non-regular sampling approach. *IEEE Control Syst Lett*. 2020;4(3):737–42.
33. Karg B, Lucia S. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Trans Cybern*. 2020;50(9):3866–78.
34. Paulson JA, Mesbah A. Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction. *IEEE Control Syst Lett*. 2020;4(3):719–24.
35. Borrelli F, Bemporad A, Morari M. *Predictive control for linear and hybrid systems*. Cambridge: Cambridge University Press; 2017.
36. Boyd S, Vandenberghe L. *Convex optimization*. Cambridge: Cambridge University Press; 2004.
37. Bertsekas DP. *Nonlinear programming*. Belmont: Athena Scientific; 2016.
38. Markolf L, Stursberg O. Polytopic input constraints in learning-based optimal control using neural networks. In: *Proceedings of the 2021 European Control Conference, 2021*; pp. 1018–23.
39. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge: MIT Press; 2016.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.