Chapter 6
Integrating Learning-Based and MPC-Based Control for PWA Systems: Challenges and Opportunities

Azita Dabiri, Kanghui He, Shengling Shi, Dingshan Sun, Jesus Lago, and Bart De Schutter

Abstract Learning-based control, in particular reinforcement learning, and optimization-based control, in particular model predictive control, each have their advantages and disadvantages for online, real-time optimal control of systems with complex dynamics. However, both approaches are highly complementary and therefore there is an increased interest in combining their advantages in an integrated approach. In this chapter, we provide an overview of recent results, challenges, and opportunities on an integrated learning-based and optimization-based control approach. We focus in particular on piecewise affine systems as they are an extension of linear systems that can model or approximate hybrid or nonlinear behavior and as they still allow for effective numerical solution approaches.

Jesus Lago contributed to this work as an outside activity and not as part of his role at Amazon.

A. Dabiri (✉) · K. He · S. Shi · B. De Schutter
Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands
e-mail: a.dabiri@tudelft.nl

K. He
e-mail: K.He@tudelft.nl

S. Shi
e-mail: S.Shi-3@tudelft.nl

B. De Schutter
e-mail: B.DeSchutter@tudelft.nl

D. Sun
Department of Transport and Planning, Delft University of Technology, Delft, The Netherlands
e-mail: D.Sun-1@tudelft.nl

J. Lago
Amazon Inc., Amsterdam, The Netherlands

E. Garone et al. (eds.), *Nonlinear and Constrained Control*, Lecture Notes in Control and Information Sciences 496, https://doi.org/10.1007/978-3-031-82681-8_6

6.1 Challenges and Opportunities in Nonlinear Control

Controlling nonlinear systems is crucial in various fields due to the existence of nonlinearities in numerous real-world systems. Efficient control of these nonlinear systems, such as intelligent transportation systems, renewable energy, and smart grids is of paramount significance. These applications serve as the backbone of our society, making the reliable control of their nonlinear dynamics crucial for their efficient operation. Several bottlenecks make the design of an efficient control methodology for nonlinear systems challenging. What follows briefly examines the primary challenges associated with controlling these systems.

- **Complex Dynamics:** Most control methodologies rely on the existence of an accurate model describing the system's behavior. However, accurate modeling of nonlinear systems is often a complex task and may require sophisticated mathematical tools. Controlling these complex dynamics would call for advanced control methodologies. However, this itself will lead to the next major challenge in the control of nonlinear systems, namely computational complexity.
- **Computational Complexity:** Due to the complex dynamics exhibited by nonlinear systems, conventional and computationally cheap linear control methodologies will likely fail to deliver high performance. Most advanced model-based control techniques designed for nonlinear systems are computationally intensive in general as they result in nonlinear non-convex optimization problems possibly with mixed continuous and integer variables, hence NP-hard. Moreover, most of these problems should, in general, be solved online/ in real time, and the computational complexity of the advanced controller makes their implementation even more challenging.
- **Uncertainty in the Model, and External Disturbances:** Mathematically formulated models describing nonlinear systems, result in models that are often subject to uncertainty in parameters, external disturbances, or unmodeled dynamics. Achieving a reliable controller would call for the explicit consideration of the effects of these sources of uncertainties in the design.
- **Stability Analysis of the Closed-Loop System:** A critical aspect of a control system is its closed-loop stability guarantee. The task is inherently challenging when the underlying system behaves nonlinearly.
- **Constraints Satisfaction on States and Input:** In practice, ensuring the safe operation of a system involves defining safe sets, wherein states and control inputs are permitted and adhering to this set during the operation. Designing a control strategy for a nonlinear system that respects state-input constraints is crucial, albeit non-trivial.
- **Dealing with Large-Scale Systems:** Large-scale systems appear in many applications like water networks, sensor networks, and smart grids. As centralized control methods cannot deal with large-scale systems due to the tremendous computational complexity of the centralized control task, scalability issues, and bandwidth limitations, real-time control of large-scale systems often necessitates the use of distributed control techniques. Most optimization-based distributed control methods

either impose additional assumptions (such as linearity or convexity) or in some cases, do not result in intelligent controllers that can efficiently and effectively negotiate and coordinate their actions.

Hence, the big scientific challenge in controlling nonlinear systems is to develop control methods that can determine control signals within the limited available computation time such that the controlled system operates in an efficient, high performance, and reliable way, while at the same time making sure all operational constraints are satisfied. In the following discussion, we examine two mainstream approaches in the field, namely Model Predictive Control (MPC) and Reinforcement Learning (RL). As elaborated below, the two approaches exhibit similarities in various aspects but are significantly different in some others. Our attention is in general directed toward a distinct class of nonlinear systems, namely those with hybrid dynamics which are systems with a combination of continuous dynamics (typically described by differential or difference equations) and switching (between different modes of continuous operation) or topology changes. Later on, we will focus on a special subclass of systems called Piewise Affine systems (PWA). It is worth noting that the presented arguments are mostly valid with minor deviation for other classes of nonlinear systems.

6.2 Control of Systems with Hybrid Dynamics

Complex large-scale transportation and infrastructure systems such as road, railway, electricity, gas, and water networks, are typical examples of (large scale) networks with hybrid dynamics. At an abstract level, they can be represented by a network consisting of links (e.g., roads, rails, power lines, pipes) and nodes (intersections, joints) through which some commodity (e.g., vehicles, trains, electrons, molecules) flows. Despite their importance, efficient control of these systems is beyond the reach of state-of-the-art control methods. In addition to the aforementioned challenges, the hybrid dynamics introduce discrete decisions in addition to continuous decisions, which adds an extra level of complexity. Improper control and insufficient coordination of these types of systems can lead to highly inefficient performance, constraint violations, instability, or even worse, to severe malfunctions or failures, such as the regular breakdowns of road traffic networks in case of incidents, network-wide accumulation and propagation of huge delays in railway networks, or blackouts in power grids.

6.2.1 Model Predictive Control

Model Predictive Control (MPC) [19, 57] is an optimization-based receding-horizon control approach in which a model is used to predict the future behavior of the system

for a given input sequence and in which a cost criterion is optimized over a finite horizon subject to constraints on the inputs, states, and outputs.

Thanks to its optimization-based approach, MPC enables the integration of nonlinear systems with multiple inputs and outputs, and provides a systematic consideration of multiple objectives in the synthesis. The receding horizon nature of MPC allows a degree of robustness, allowing the scheme to adapt to some extent to changes in the system and environment. There exists a strong theoretical foundation providing guarantees in terms of performance, recursive feasibility, and stability of nominal, robust, and stochastic MPC. With these favorable features, MPC has found application in various industries, e.g., process control, the automotive industry, and energy management. However, MPC suffers from a few disadvantages:

- Since MPC is an optimization-based scheme, real-time implementation of it, especially for large-scale systems or systems with fast dynamics is a computationally challenging task.
- The performance of MPC relies heavily on the accuracy of the underlying system model. Model mismatch may highly deteriorate the controller's performance.
- In case an approximation of the original problem in terms of, e.g., approximated model or cost function is used in the MPC formulation, hyperparameter tuning of MPC parameters is an important step to assure system performance, which in general, is not a trivial task.

For hybrid systems, in general, optimization-based control synthesis is a challenging task characterized by a huge computational complexity due to the presence of discrete variables resulting from the hybrid dynamics. Control of hybrid systems is a topic of intensive ongoing research [6, 16, 28, 48, 70, 74]. In [7], MPC has been extended to a class of hybrid systems with real-valued and binary decision variables and logic relations. The systems are called Mixed Logical Dynamical (MLD). This allows to apply MPC to small-scale piecewise affine (PWA) systems, resulting in linear or quadratic mixed-integer optimization problems. The main challenge for MPC of hybrid dynamics however lies in the fact that a nonlinear non-convex optimization problem has to be solved, which results in a huge computational burden, in particular for large-scale systems.

6.2.2 Reinforcement Learning

In the context of learning-based control, a prominent approach is Reinforcement Learning (RL) [63]. In reinforcement learning, the control agent iteratively learns directly or indirectly an optimal policy (i.e., a mapping from states to control actions) that minimizes an infinite-horizon discounted reward. Several RL algorithms have been developed, including but certainly not limited to fitted Q-iteration, least-squares policy iteration, approximate SARSA, and policy gradient [10, 17, 63].

The training can either be done in a model-free way, by interacting with the environment, or in a model-based way. This feature makes the RL algorithms suitable

for cases in which the system model is unknown, or highly nonlinear, and hence hard to capture. The trial-and-error, as well as the exploratory nature of the scheme, allows adaptation and learning of strategies in an evolving environment. Moreover, RL algorithms can adapt their strategies in real time and hence, continuously improve their control policy. The computational burden of evaluating a learned policy in the implementation phase is negligible. However, designing an RL-based controller comes with the following challenges:

- Except for systems with a considerably small number of states and actions, RL algorithms require a sort of function approximator to approximate, e.g., the underlying value or action-value functions. Neural networks are one of the most widely used function approximators for this purpose. However, especially deep RL algorithms, which employ deep neural networks as function approximators, can be susceptible to instability and convergence issues [63].
- The lack of sample efficiency is a major drawback of the RL algorithms. The learning phase of an RL-based controller often calls for a large number of interactions with the environment (the real system or a simulation). Moreover, training sophisticated deep RL controllers can be computationally intensive and requires substantial computational resources.
- Constraint satisfaction on state and input is not a trivial task. RL algorithms may need to try unsafe actions before learning the safe set of input and state.
- When the function approximator is used, choosing the right parametrization is generally not straightforward. Overfitting may result in poor generalizability of the value function or policy, while underfitting may sacrifice accuracy and hence the performance of the algorithm.
- While the use of neural networks as function approximators in RL schemes has shown success across various applications [36, 38, 53, 61, 71], the lack of interpretability of the results remains a concern.
- Due to the high dimensionality of the state and action space, RL algorithms are not easily extendable for use in large-scale systems without substantial computational resources at hand. Moreover, in the case of multi-agent settings, convergence of RL algorithms may suffer due to the non-stationarity issue [18].
- The performance of RL algorithms can be sensitive to the hyperparameters, e.g., learning rate, or the architecture of the underlying neural network. Tuning hyperparameters is a time-consuming task, and the inherent lack of interpretability while using neural networks further adds to the difficulty of this tuning process.
- When the function approximator is used, choosing the right parametrization is generally not straightforward. Overfitting may result in poor generalizability of the value function or policy, while underfitting may sacrifice accuracy and hence the performance of the algorithm.
- Achieving a desired behavior using an RL-based controller hinges on the careful selection of a proper reward/penalty, which can be a challenging task.

6.3 Integration of MPC and RL: General Case

The combination of MPC and RL enables a hybrid approach that leverages the strengths of both methods while overcoming the weaknesses of each. Such a hybrid scheme is characterized by the following:

- The advantages of optimization-based control (ability to deal with complex mixed state-input constraints and system changes, and providing some guarantees on performance, safety, feasibility, and stability) and learning-based control (low online computational burden, inherent ability to deal with uncertainty) are fused in a unified framework. Hence, MPC can contribute to sample efficiency, interpretability, and safety, as it provides a structured optimization-based framework in which all the knowledge from the system can be integrated into the control synthesis. On the other hand, RL contributes adaptability and learning capabilities through its exploratory nature.
- The major disadvantages of the respective approaches (such as heavy online computational burden for optimization-based control versus inability to deal with complex constraints and heavy offline computation burden for learning-based control) can be addressed.
- In the case of large-scale systems, a multi-agent control framework can be adopted in which solution approaches based on distributed control and multi-agent learning have demonstrated promising results.

In such a setting, the use of models allows the controllers and the learning algorithms to predict the effects of control actions on the future evolution of the system. In this chapter, we propose to consider the class of piecewise affine (PWA) models [62]. The rationale for this choice is the tractability feature that the PWA formulation may offer. In PWA state space models, the state-input space is divided into polyhedral regions where in each region the state update and output equations are affine. PWA models, although nonlinear, are the simplest extension of linear/affine systems that can model hybrid phenomena. Moreover, they can approximate nonlinear behavior with a tunable trade-off between complexity and accuracy; this tuning knob can be used to develop control methods that combine the required speed with the required optimality. Using PWA models will also enable us to use the rich variety of results in optimization-based control for this class of systems.

Given the various trade-offs that play a role in control of hybrid systems, there will not be a single method in which MPC and RL are combined that performs best under all circumstances and for all systems. We present one solution method in detail in Sect. 6.4 and propose several alternative methods in Sect. 6.5. However, we defer this in-depth discussion until after providing a brief overview of the available methods in the literature that integrate learning-based and model-based control in a single framework. It is worth mentioning that there is limited relevant research specifically focused on hybrid systems. Therefore, the following literature review will consider the research on more general classes of systems.

The connection between reinforcement learning and optimal control for unconstrained problems has been elaborated in [43, 44]. Reference [33] provides an overview of the research on the integration of learning into MPC. In essence, there are three main directions: using online learning to improve the prediction model, using MPC to augment learning-based controllers with constraint satisfaction properties, and using learning to infer the cost function and constraints for MPC that lead to the best closed-loop performance. In particular, [4, 46] consider learning-based MPC where a model of the system is learned online and combined with robust MPC. In [35, 37, 39, 40, 55], the system is modeled as a Gaussian process, and an MPC controller is designed that optimizes the expected value of the cost subject to chance constraints. In [65, 66], robust linear MPC is used to modify a proposed learning-based input aimed at constraint satisfaction properties. The integration of MPC with reinforcement learning is more explicitly addressed in [2, 3, 24–26], which uses differential MPC or parameterized MPC as policy function approximator for reinforcement learning. Within the same framework, closed-loop stability of MDPs is explicitly considered in [73] and constraint satisfaction in RL is accounted for via robust MPC in [72]. Moreover, reinforcement learning can be used to train a neural network offline to approximate a robust MPC controller using samples generated by that controller [32]. Robust optimal control has been combined with reinforcement learning in [8, 64], but for linear systems only. In [27] RL algorithms are used to update a parameterized nonlinear MPC scheme. Different from [27], which parameterizes all terms including the terminal function, the prediction model, and the constraints, [54] only parameterizes the terminal function and uses approximate value iteration to learn it offline. Learning a convex terminal cost of MPC is also adopted in [1]. Similarly, in [47] approximate policy iteration is used to learn the terminal function offline. It is proven in [47, 54] that the resulting MPC controller makes the system safe and asymptotically stable if the approximation error is bounded and the MPC horizon is large enough. However, the aforementioned combinations are unable to address the online computational challenges encountered by MPC. This limitation arises because their policies are still dictated by online MPC schemes with an unchanged horizon.

Several of the aforementioned approaches consider the unconstrained case [43, 44], assume linear models (with bounded uncertainty) [4, 8, 64–66], or require online model learning (which requires targeted exploration to reduce uncertainty, while such exploration may conflict with the optimality of the control performance) [35, 39, 40, 46]. The approaches listed above do not apply to PWA systems and do not take the hybrid dynamics of systems into account. In what follows, we provide a solution method for this particular class of systems.

6.4 Control of PWA Systems by Integrating MPC and RL

There has been an increasing interest in control of piecewise affine systems due to their capability of representing hybrid systems and approximating nonlinear dynamics. In PWA systems, the state and input spaces are jointly partitioned into polyhedral regions, where an affine subsystem is defined in each region. Control of PWA systems has a broad range of applications, including emergency evasive maneuvers [21], robotic manipulation that has multi-contact behaviors [58], transportation networks [52], and power systems [51]. For PWA systems with constraints, MPC is widely applied. However, MPC for PWA systems still faces challenges in computational complexity [15], because it involves solving a mixed-integer linear or quadratic programming (MILP or MIQP) problem, which is NP-hard in general and the complexity of which grows rapidly with the number of subsystems. Explicit MPC [15], an offline version of MPC, requires solving a parametric MILP or MIQP problem, which also suffers from computational complexity issues. These issues make MPC-only suitable for slow PWA processes or low-dimensional problems with short prediction horizons [58].

In contrast to MPC, reinforcement learning can learn a policy that minimizes a finite-/infinite-horizon cost and has a much lower online computational burden than MPC [23]. In RL, two different methodologies can be distinguished: policy search and dynamic programming [17]. Policy search directly searches for an optimal policy, while dynamic programming aims to solve the Bellman equation first and then determines the policy. Dynamic programming has an advantage over policy search in that it reduces the policy optimization problem to a one-step look-ahead problem. When applied to systems with continuous state and input spaces, approximate dynamic programming (ADP), which uses sample-based methods accompanied by sophisticated function approximators, has been developed [34, 68]. However, RL-based controllers may provide unsafe or unstable control actions that result in undesirable or even destructive effects on the system.

In this chapter, we focus on PWA systems with union-of-polyhedra (UoP) state constraints and polyhedral input constraints. We employ RL to learn an online computationally efficient MPC controller. To this end, we develop two formulations of including RL in the MPC paradigm. The first one relaxes the state constraint by adding a soft penalty to the stage cost and then uses RL to learn a terminal function for MPC. Consequently, the horizon of MPC can be significantly reduced to one, and thereby the online computational cost is reduced significantly as well. The second one exploits RL to approximate the maximal control-invariant set, which serves as the terminal constraint for MPC. In the absence of approximation errors, this constraint ensures both the recursive feasibility of the MPC problem and the safety of the MPC policy.

6.4.1 Optimal Control of PWA Systems

We consider the following infinite-horizon constrained optimal control problem for PWA systems

$$
\begin{aligned}
J^*(x) = \min_{\{u_t, x_t\}_{t=0}^{\infty}} \lim_{N\to\infty} & \left\{ J_N(x, u_1, u_2, \ldots, u_N) = \sum_{t=0}^{N} \alpha^t l(x_t, u_t) \right\} \\
& \text{s.t. } x_{t+1} = f_{\text{PWA}}(x_t, u_t) = A_i x_t + B_i u_t + f_i, \quad \text{if } \begin{bmatrix} x_t \\ u_t \end{bmatrix} \in C_i, \\
& x_t \in X, \ u_t \in U, \ t = 0, 1, \ldots N, x = x_0,
\end{aligned} \tag{6.1}
$$

where $\alpha \in (0, 1]$, $\{C_i\}_{i=1}^{s}$ is a polyhedral partition of the state-input space $\mathcal{X} \times \mathcal{U}$, and the function $l : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is the stage cost. Throughout the remainder of this section, we assume that the dynamics, constraints, and stage costs satisfy the following assumption:

Assumption 6.1
- *Dynamics*: The function $f_{\text{PWA}}(\cdot, \cdot) : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is a continuous PWA function. Besides, $f_{\text{PWA}}(0, 0) = 0$.
- *Constraints*: The state constraint set $X = \bigcup_{i=1}^{r_0} X^{(i)}$ is a UoP, where $X^{(i)}$ is a polytope for each $i \in 1\{, \ldots, r_0\}$ and r_0 is the number of polytopes. The input constraint set U is a polytope. Moreover, $X \times U \subset \bigcup_{i=1}^{s} C_i = \mathcal{X} \times \mathcal{U}$.
- *Stage cost*: The stage cost is continuous PWA w.r.t. x and u and satisfies $l(0, 0) = 0$ and $l(x, u) > 0$ if $x \neq 0$ or $u \neq 0$.

We denote by $\bar{X}$ the set of feasible initial states x_0 that make the problem (6.1) feasible. In some literature [16], $\bar{X}$ is called the maximal control-invariant set. Control-invariant sets play a crucial role in the synthesis of safe control policies. The methods of synthesizing control-invariant sets for PWA systems include the LMI approach [42], polyhedra iteration [16], and polytope trees [58].

Note that the first and second parts of Assumption 6.1 are common in literature. The assumption on the PWA property of the stage cost will be explained in Sect. 6.4.3.

Assumption 6.2 The set $\bar{X}$ is non-empty. Furthermore, for any $x_0 \in \bar{X}$, there exists a policy $\pi(\cdot)$ such that the system $x_{t+1} = f_{\text{PWA}}(x_t, \pi(x_t))$, starting from x_0, reaches the origin in a finite number of time steps.

The u_0^* in the optimal solution $\{u_t^*, x_t^*\}_{t=0}^{\infty}$ is a PWA function of the initial state x_0 [5]. We denote this function as the optimal policy $\pi^*(\cdot) : \bar{X} \to U$ for the optimal control problem (6.1). For any $x \in \mathcal{X}$, according to Bellman's Principle of Optimality [14], J^* and π^* satisfy the following equations:

$$
\begin{aligned}
J^*(x) &= T_{\bar{X}} J^*(x) := \min_{u \in U, \ f_{\text{PWA}}(x,u) \in \bar{X}} l(x, u) + \alpha J^*(f_{\text{PWA}}(x, u)), \\
\pi^*(x) &\in \operatorname*{arg\,min}_{u \in U, \ f_{\text{PWA}}(x,u) \in \bar{X}} l(x, u) + \alpha J^*(f_{\text{PWA}}(x, u)), \ x \in \bar{X},
\end{aligned} \tag{6.2}
$$

where T_S is a Bellman operator under the constraints $f_{\text{PWA}}(x,u) \in S$ and $u \in U$, with S being any subset of $\mathcal{X}$. To solve (6.1), we should identify $\bar{X}$ and then compute $J^*(\cdot)$ and $\pi^*(\cdot)$. However, the computation of J^*, $\bar{X}$, and π^* is in general intractable for high-dimensional linear and PWA systems. Usually, an implicit MPC controller is computed online with a finite N. To approximate the infinite-horizon solution and maintain stability, a terminal cost function is added to estimate the neglected stage costs $\sum_{t=N}^{\infty} \alpha^t l(x_t, u_t)$. Besides, a terminal constraint is imposed to guarantee the feasibility of MPC.

The design of the terminal cost and terminal constraint usually relies on solving linear matrix inequalities problems [42]. Such design, however, can result in very conservative MPC controllers. Namely, the MPC solution is very suboptimal for problem (6.1). To reduce conservatism, one needs to increase the horizon, which will, on the other hand, impose a substantial online computational burden.

Observe from (6.2) that if J^* and $\bar{X}$ can be exactly computed, the optimal control input can be obtained by solving an MPC problem with $N = 1$, the terminal cost $J^*(x_1)$ and the terminal constraint $x_1 \in \bar{X}$. In the following section, we will show how RL can be applied to approximate J^* and $\bar{X}$ so that the horizon of MPC can be reduced to one.

6.4.2 *Learning the Terminal Cost*

Value iteration (VI) [17], the most basic RL algorithm, computes J^* by solving the following parametric optimization problems iteratively:

$$\begin{aligned} J_k(x) &= T_{X_{k-1}} J_{k-1}(x) = \min_{u \in U,\ f_{\text{PWA}}(x,u) \in X_{k-1}} l(x,u) + \alpha J_{k-1}(f_{\text{PWA}}(x,u)), \\ X_k &= \text{Pre}(X_{k-1}) \cap X, \end{aligned} \tag{6.3}$$

with the initialization $X_0 = X_{\text{CI}}$ and $J_0(\cdot) = J_{\text{CL}}$, where $J_{\text{CL}}(\cdot) : X_{\text{CI}} \to \mathbb{R}_{\geq 0}$ is a control Lyapunov function defined in a polyhedral control-invariant set X_{CI}. In (6.3), $\text{Pre}(S) = \{x \in \mathcal{X} | \exists u \in U \text{ s.t. } f_{\text{PWA}}(x,u) \in S\}$. Under Assumptions 6.1 and 6.2, as $k \to \infty$, J_k converges to J^* and X_k converges to $\bar{X}$ [9]. An optimization-based method that can compute X_{CI} and J_{CL} for PWA systems is reported in [42]. Although J_k can be exactly computed by solving parametric mixed-integer linear programs, this approach is limited to small-scale PWA systems. Therefore, it is necessary to simplify both the procedure of solving (6.3) and the optimal value function, by using some approximation methods.

However, the VI formulation (6.3) is not suitable for approximation because the probably non-convex constraint $f_{\text{PWA}}(x,u) \in X_{k-1}$ leads to complex optimization problems when using sample-based approaches. To deal with this issue, we consider soft state constraints by defining a penalty function $P(\cdot,\cdot)$. Suppose that each $X^{(i)}$ of the UoP constraint set X is characterized by the linear inequality $E_X^{(i)} x \leq g_X^{(i)}$,

where $E_X^{(i)} \in \mathbb{R}^{m_x^{(i)} \times n_x}$ and $g_X^{(i)} \in \mathbb{R}^{m_x^{(i)}}$. Then we design the penalty function $P(\cdot, \cdot)$ as the following min-max form:

$$P(x, X) = p \min_{i \in \{1,\ldots,r_0\}} \max_{j \in \{1,\ldots,m_x^{(i)}\}} \left\{ (E_X^{(i)})_{j\cdot} x - (g_X^{(i)})_{j\cdot} \right\},$$

where $(E_X^{(i)})_{j\cdot}$ is the jth row of $E_X^{(i)}$, $m_x^{(i)}$ is the number of rows of $E_X^{(i)}$, and $p > 0$ is the constraint violation penalty weight. With P, we modify the VI (6.3) to the following state-unconstrained version:

$$J_k^{\text{soft}}(x) = \min_{u \in U} \left(l(x, u) + \max\{0,\ P(x, X)\} + \alpha J_{k-1}^{\text{soft}}\left(f_{\text{PWA}}(x, u)\right) \right), x \in \mathcal{X}, \tag{6.4}$$

with the initialization $J_0^{\text{soft}}(x) = \max\{0,\ P(x, X_{\text{CI}})\}$. In the following lemma, we analyze the PWA property and continuity of each $J_k^{\text{soft}}(\cdot)$ as well as the convergence of the sequence $\{J_k^{\text{soft}}(\cdot)\}_{k=0}^{\infty}$ to a fixed optimal value function.

Lemma 6.1 ([30]) *Considering the VI (6.4), if Assumptions 6.1–6.2 hold, each $J_k^{\text{soft}}(\cdot)$ is a continuous PWA function on $\mathcal{X}$ and the value function sequence $\{J_k^{\text{soft}}(\cdot)\}_{k=0}^{\infty}$ converges pointwise to*

$$\begin{aligned} J^{\text{soft}*}(x) = \min_{\{u_t, x_t\}_{t=0}^{\infty}} & \sum_{t=0}^{\infty} \alpha^t \left(l(x_t, u_t) + \max\{0,\ P(x_t, X)\} \right) \\ \text{s.t.} \quad & x_{t+1} = f_{PWA}(x_t, u_t),\ u_t \in U,\ t = 0, 1, \ldots,\ x_0 = x. \end{aligned}$$

Continuity of the value functions is desired since it enables a universal approximation capability [29]. Since (6.4) has no state constraints, a tractable ADP approach can be developed to approximate each J_k^{soft} by a function approximator $\hat{J}$. The convergence of the ADP algorithm to $J^{\text{soft}*}$ is guaranteed if $\alpha < 1$ and if the approximation error in each iteration is upper bounded [11, Proposition 2.3.2]. See, e.g., [14, Sect. 5.2] and [17, Sect. 3.4.1] for the detailed implementation of ADP.

With $\hat{J}(\cdot)$ available, a suboptimal control policy $\hat{\pi}(\cdot) : \mathcal{X} \to \mathcal{U}$ can be implicitly determined by

$$\hat{\pi}(x) \in \arg\min_{u \in U}\ l(x, u) + \alpha \hat{J}(f_{\text{PWA}}(x, u)),\ \forall x \in \mathcal{X}. \tag{6.5}$$

Convergence of ADP is influenced by approximation errors introduced by value function approximation. Due to the iterative nature of the VI process, these approximation errors tend to accumulate with each iteration. Introducing a discount factor $\alpha < 1$ can mitigate this issue by making the mapping from J_k^{soft} to J_{k+1}^{soft} a contraction mapping, thus constraining the growth of errors. Consequently, the convergence of the ADP algorithm to $J^{\text{soft}*}$ is guaranteed if $\alpha < 1$ and if the approximation error in each iteration is upper bounded [11, Proposition 2.3.2]. However, it is worth noting that a discounted factor can have adverse effects on the closed-loop stability of the system controlled by $\hat{\pi}$ [56] and hence $\alpha = 1$ can be a better choice for stability.

Therefore, when selecting α, one must carefully balance considerations of expected stability performance and approximation capability.

Note that the error between J_k^{soft} and $J^{\text{soft}*}$ can be analyzed by the contraction property of the mapping from J_k^{soft} to J_{k+1}^{soft}. In particular, when $\mathcal{X}$ is compact and $\alpha < 1$, the error bound of J_k^{soft} w.r.t. $J^{\text{soft}*}$ is $O(\alpha^k)$ [11], which vanishes as $k \to \infty$. The error between $J^{\text{soft}*}$ and J^* is caused by the soft penalty. For general PWA systems, this error cannot be eliminated because the penalty weight p cannot be infinity. However, if the PWA system reduces to a linear time-invariant system and X is a polytope, it is possible to make $J^{\text{soft}*} = J^*$ by choosing a sufficiently large p, according to the exact penalty theorem [13, Proposition 5.4.5].

6.4.3 *Learning the Terminal Constraint*

It is observed from Lemma 6.1 that J_k^{soft} converges to $J^{\text{soft}*}$, the optimal value function for the soft constrained optimal control problem, instead of J^*. Besides, there is no constraint on the terminal state in (6.5). Therefore, the policy $\hat{\pi}$ can only ensure state constraint satisfaction in a small subset of $\bar{X}$. In this section, we improve the safety of the RL policy by enlarging the safe set. To this end, we develop an ADP algorithm to approximate each reachable set X_k in (6.3), which will converge to $\bar{X}$ as k goes to $+\infty$ [16].

We consider each B_k as the value function of the following optimization problem:

$$B_k(x) \triangleq \min_{\{u_t, x_t\}_{t=0}^{k}} \max \left\{ \max_{t \in \mathbb{I}(k-1)} \alpha^t P(x_t, X), \alpha^k P(x_k, X_{\text{CI}}) \right\}$$
$$\text{s.t. } x_0 = x,\ x_{t+1} = f_{\text{PWA}}(x_t, u_t),\quad u_t \in U,\ t = 0, 1, \ldots, k-1. \tag{6.6}$$

In (6.6), $\mathbb{I}(k-1)$ denotes the set of nonnegative integers less than or equal to $k-1$. The following lemma shows the equivalence between each X_k and the zero sub-level set $\mathcal{S}_{B_k} := \{x \in \mathcal{X} | B_k(x) \leq 0\}$ of each B_k.

Lemma 6.2 *Consider B_k from (6.6). Suppose that Assumptions 6.1–6.2 hold. Then (i) $B_k(\cdot)$ is a continuous PWA function of the state, and (ii) $\mathcal{S}_{B_k} = X_k,\ \forall k \in \mathbb{I}(\infty)$.*

Proof The continuity of $B_k(\cdot)$ is proven in [31]. The PWA property of $B_k(\cdot)$ follows directly from the facts that $P(\cdot, X)$, $P(\cdot, X_{CI})$, and $f_P W A(\cdot, \cdot)$ are PWA and that U is a polytope. To prove $\mathcal{S}_{B_k} = X_k,\ \forall k \in \mathbb{I}(\infty)$, we see that similar to J_k, B_k can also be computed recursively via the following dynamic programming technique:

$$B_k(x) = \Gamma B_{k-1}(x) := \max\{P(x, X),\ \alpha \min_{u \in U} B_{k-1}(f_{\text{PWA}}(x, u))\},\ \forall x \in \mathcal{X}. \tag{6.7}$$

Suppose that $\mathcal{S}_{B_{k-1}} = X_{k-1}$ for a $k \in \{1, 2, \ldots, \infty\}$. As a result, for any $x \in \mathcal{S}_{B_k}$, it follows from (6.8) that there exists a $u \in U$, such that $f_{\text{PWA}}(x, u)) \in \mathcal{S}_{B_{k-1}}$. Therefore, $\mathcal{S}_{B_k} \subseteq \text{Pre}(\mathcal{S}_{B_{k-1}}) \cap X = \text{Pre}(X_{k-1}) \cap X = X_k$. Meanwhile, for any $x \notin \mathcal{S}_{B_k}$, according to (6.8) we have $x \notin X$ or that there does not exist any $u \in U$ such that $f_{\text{PWA}}(x, u)) \in \mathcal{S}_{B_{k-1}}$. This implies that $\text{Pre}(\mathcal{S}_{B_{k-1}}) \cap X \subseteq \mathcal{S}_{B_k}$. Therefore, we have $\mathcal{S}_{B_k} = X_k$. Finally, as $\mathcal{S}_{B_0} = X_0$, by induction, we can conclude the proof. □

Lemma 6.2 indicates that if an approximation $\hat{B}_{\omega_k}$ of each B_k is available, the constraint $f_{\text{PWA}}(x, u) \in X_{k-1}$ in (6.3) can be replaced by its approximation $\hat{B}_{\omega_{k-1}}(f_{\text{PWA}}(x, u)) \leq 0$, which in general has a much simpler representation than the original constraint. Here, $\hat{B}_{\omega_k}(\cdot)$ is a function of x, parameterized by ω_k. Similar to J_k, B_k can also be computed recursively via the following dynamic programming technique:

$$B_k(x) = \Gamma B_{k-1}(x) := \max\{P(x, X),\ \alpha \min_{u \in U} B_{k-1}(f_{\text{PWA}}(x, u))\},\ \forall x \in \mathcal{X}. \quad (6.8)$$

Together with the continuity of B_k, (6.8) allows for applying an ADP approach to estimate each B_k. Combining (6.2), (6.3), and (6.8), we develop Algorithm 6.1 to synthesize a policy. Algorithm 6.1 includes two ADP updates in parallel. One is for updating the estimation of J_k, the other is performed to estimate the reachable set X_k. In Algorithm 6.1, the sampling strategy can be sampling from a uniform grid or random sampling. The Bellman operators T_S and Γ are defined in (6.2) and (6.8). After Algorithm 6.1 terminates, a policy is generated by solving the following MPC problem with $N = 1$

$$\hat{\pi}'(x) \in \underset{u \in U,\ \hat{B}_{\omega_{k-1}}(f_{\text{PWA}}(x,u)) \leq 0}{\arg\min} \quad l(x, u) + \alpha \hat{J}_{\theta_{k-1}}(f_{\text{PWA}}(x, u)),\ \forall x \in \mathcal{X}. \quad (6.9)$$

Different from (6.5), (6.9) contains a terminal constraint. If $\hat{B}_{\omega_{k-1}}$ is a good upper approximation of B_{k-1} and problem (6.9) is recursively feasible for the closed-loop system $x_{t+1} = f_{\text{PWA}}(x_t, \hat{\pi}'(x_t))$ starting from x_0, such a constraint will make $\hat{\pi}'$ a safe policy for x_0. In an ideal situation when k approaches $+\infty$ and there is no approximation error, $\hat{\pi}'$ is a safe policy in $\bar{X}$. However, recursive feasibility of (6.9) is in general not guaranteed and difficult to verify in advance due to the approximation error. Therefore, it is recommended to add a slack variable to relax this constraint when infeasibility is detected.

Since both B_k and J_k are PWA, it is preferable that the candidate approximator can also output a PWA function. Suitable choices include neural networks with (leaky) rectifier linear units as activation functions, difference of two max-affine functions, and min-max neural networks. The benefit of using PWA approximation is that one can convert the optimization problems $\Gamma \hat{B}_{\omega_{k-1}}(x_i^{\text{s}})$ and $T_{\mathcal{S}_{\hat{B}_{\omega_k}}} \hat{J}_{\theta_{k-1}}(x_i^{\text{s}})$ in Algorithm 6.1 into MILP problems so that the global optimum can be obtained by branch and bound [69].

Algorithm 6.1 Constrained ADP algorithm

1: **Output:** Value function approximations $\hat{J}_{\theta_{k-1}}$ and $\hat{B}_{\omega_{k-1}}$.
2: Compute J_{CL} and X_{CI} via the method in [42]. Collect N_{s} state samples $\{x_i^{\mathrm{s}}\}_{i=1}^{N_{\mathrm{s}}}$ in $\mathcal{X}$. Initialize the value function $\hat{J}_{\theta_0}(\cdot)$ and $\hat{B}_{\omega_0}(\cdot)$ by

$$\theta_0 \leftarrow \arg\min_{\theta} \sum_{x_i^{\mathrm{s}} \in X_{\mathrm{CI}}} \left(J_{\mathrm{CL}}(x_i^{\mathrm{s}}) - \hat{J}_{\theta}(x_i^{\mathrm{s}})\right)^2, \ \omega_0 \leftarrow \arg\min_{\omega} \sum_{x_i^{\mathrm{s}} \in \mathcal{X}} \left(P(x_i^{\mathrm{s}}, X_{\mathrm{CI}}) - \hat{B}_{\omega}(x_i^{\mathrm{s}})\right)^2$$

3: **for** $k = 1, 2, \ldots$ **do** the updates for $\hat{J}_{\theta_{k-1}}(\cdot)$ and $\hat{B}_{\omega_{k-1}}(\cdot)$ by

$$\omega_k \leftarrow \arg\min_{\omega} \sum_{x_i^{\mathrm{s}} \in \mathcal{X}} \left(\Gamma \hat{B}_{\omega_{k-1}}(x_i^{\mathrm{s}}) - \hat{B}_{\omega}(x_i^{\mathrm{s}})\right)^2, \ \theta_k \leftarrow \arg\min_{\theta} \sum_{\hat{B}_{\omega_k}(x_i^{\mathrm{s}}) \leq 0} \left(T_{S_{\hat{B}_{\omega_k}}} \hat{J}_{\theta_{k-1}}(x_i^{\mathrm{s}}) - \hat{J}_{\theta}(x_i^{\mathrm{s}})\right)^2$$

4: **if** $||\theta_k - \theta_{k-1}|| \leq \epsilon$ and $||\omega_k - \omega_{k-1}|| \leq \epsilon$ **then break**.
5: **end if**
6: **end for**

Assuming the stage cost to be PWA allows us to analyze the PWA property of the value functions, facilitating the utilization of PWA approximation. This further contributes to globally solving the optimization problems in ADP by MILP. This assumption is not restrictive because any continuous nonlinear cost function defined on a compact set can be fitted by a continuous PWA function with arbitrary accuracy [60]. Conversely, one can simply assume a general nonlinear and continuous stage cost and use nonlinear programming techniques to implement Algorithm 6.1.

6.5 Possible Future Research Directions

In the previous section, two solution methods have been presented in which RL and MPC are combined to control a PWA system, assuming perfect knowledge of the model. There remain several interesting directions for developing RL-MPC in a data-driven paradigm, e.g., analyzing the robustness of the generated policies to model uncertainties, or directly creating model-free RL algorithms. Moreover, there is inherently a trade-off between the conservatism of the policy $\hat{\pi}'$ and the satisfaction of state constraints. The conservatism comes from the inner approximation of $\bar{X}$ while the constraint violation results from the outer approximation. Significant advancements in addressing this challenge necessitate the pursuit of a delicate balance between computational efficiency and approximation accuracy. This can be achieved by incorporating concepts from various sources, such as predictive safety filters, learning control barrier functions, Hamilton–Jacobi reachability [67], and advanced classification techniques [20]. Furthermore, practical challenges persist in real-world applications. These include enhancing sampling efficiency for large-scale PWA systems, devising specific structures for approximation models to preserve key properties of real models, and investigating additional benefits derived from the

consideration of PWA systems. Moreover, further research is needed to explore the impact of approximation errors on the performance of the one-step lookahead policy in terms of both infinite-horizon cost and constraint satisfaction.

As mentioned earlier, there exist several other solution methods to explore for combining optimization-based and learning-based control of PWA systems. In what follows, some of these possibilities are listed. Note that several of these approaches can also be combined into one framework, depending on the underlying application.

- The discrete optimization part is one of the main causes for the computational complexity of the MPC optimization problem for PWA systems, while learning-based approaches by their nature can more easily deal with discrete control actions (and less easily with continuous control actions). A promising research direction is to use learning-based approaches for determining discrete actions and MPC for continuous actions.
- Another important factor determining the complexity of the MPC optimization problem is the control horizon (N_c) as it determines the number of optimization variables. A setting can be adopted in which we use MPC with a shorter control horizon ($N_s < N_c$), followed by a learning-based controller (for the control actions from N_{s+1} to N_c). The rationale behind this decision is based on the general principle that control actions in the near future exert the most significant influence. Therefore, ensuring the optimality of these early control actions is crucial. In contrast, the optimality of later control actions becomes less critical due to the receding-horizon strategy.
- An alternative approach consists of starting with MPC and gradually, through online learning, building up the policy map, where the MPC controller provides the control actions to be learned. Once the policy map has been learned sufficiently well, the MPC controller's role would gradually be shifted away. The advantage of this approach is that-for a limited computation time-at first, the MPC controller will generate feasible but (due to the restricted CPU time) suboptimal solutions, while gradually the policy map will be learned and more optimal solutions can be generated quickly (as once the map has been learned the online computation takes negligible time).
- Another alternative is to use learning-based approaches to adapt online the control (and prediction) horizon and tuning weights of MPC based on the available CPU time, and the reaction (both in performance and constraint violation) of the system to the MPC control actions.
- In the case of large-scale or multi-agent systems, to obtain coordination among the multiple control agents an alternative is to develop an agreement-based approach that addresses the problem of convergence of the interconnecting variables at the boundaries of the subsystems, by first imposing agreement on these values. This can be done via appropriate parameterizations of the control policies that guarantee that the boundary values are always equal. Next, optimization-based, learning-based, or hybrid methods can be used to optimize the overall control objectives. Another promising research direction is to adapt and fuse existing MPC methods for small-scale (PWA) systems with recent advances [12, 49] in distributed

optimization and distributed control as well as newly developed methods for distributed learning-based MPC for linear systems [50] and extend them to PWA systems.

- In order to deal with uncertainty, adopting scenario-based approaches is a promising avenue of research. For the optimization-based control part, scenario-based chance-constrained MPC [35, 45, 59] can be explored, while deep learning approaches [22, 41] can be adopted to construct sets of representative scenarios that are rich enough so that performance guarantees can be given, and that can be extracted efficiently from the possibly huge amount of historical data that is available. In an alternative approach, scenario-based chance-constrained MPC can be approximated by a nominal MPC approach where the nominal disturbance scenario is determined online, as well as an approach based on a collection of scenario classes with pre-trained controllers for each class, where then the most probable scenario class for the current situation and the corresponding controller are selected online.

6.6 Conclusions

Nonlinearities are pervasive in various real-world systems. This chapter offers an overview of numerous challenges associated with ensuring reliable and safe control of these nonlinear systems. Specifically, the focus is on two key control methodologies: MPC and RL. The paper has reviewed the advantages and disadvantages of each method and explored the potential benefits of integrating the two to mitigate their respective weaknesses. The discussion includes a proposed solution method that combines MPC and RL for controlling piecewise affine systems. The chapter has concluded by suggesting several potential research directions where these two control methodologies can be integrated within a unified framework to obtain a more efficient and reliable controller.

Acknowledgements This chapter is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101018826 - CLariNet).

References

1. Abdufattokhov, S., Zanon, M., Bemporad, A.: Learning convex terminal costs for complexity reduction in MPC. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 2163–2168 (2021)
2. Airaldi, F., De Schutter, B., Dabiri, A.: Reinforcement learning with model predictive control for highway ramp metering (2023). arXiv:2311.08820

3. Amos, B., Jimenez, I., Sacks, J., Boots, B., Kolter, J.: Differentiable MPC for end-to-end planning and control. In: Advances in Neural Information Processing Systems, vol. 31, pp. 8289–8300 (2018)
4. Aswani, A., Gonzalez, H., Sastry, S., Tomlin, C.: Provably safe and robust learning-based model predictive control. Automatica **49**, 1216–1226 (2013)
5. Baoti, M., Christophersen, F.J., Morari, M.: Constrained optimal control of hybrid systems with a linear performance index. IEEE Trans. Autom. Control **51**(12), 1903–1919 (2006)
6. Bemporad, A., Heemels, W., De Schutter, B.: On hybrid systems and closed-loop MPC systems. IEEE Trans. Autom. Control **47**(5), 863–869 (2002). https://doi.org/10.1109/TAC.2002.1000287
7. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints **35**(3), 407–427 (1999)
8. Berberich, J., Koch, A., Scherer, C.W., Allgöwer, F.: Robust data-driven state-feedback design. In: 2020 American Control Conference (ACC), pp. 1532–1538 (2020)
9. Bertsekas, D.: Infinite time reachability of state-space regions by using feedback control. IEEE Trans. Autom. Control **17**(5), 604–613 (1972)
10. Bertsekas, D.: Dynamic Programming and Optimal Control—vol. II, 4th edn. Athena Scientific (2012)
11. Bertsekas, D.: Abstract Dynamic Programming. Athena Scientific (2022)
12. Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific (2015)
13. Bertsekas, D.P.: Nonlinear programming. J. Oper. Res. Soc. **48**(3), 334–334 (1997)
14. Bertsekas, D.P.: Reinforcement Learning and Optimal Control. Athena Scientific (2019)
15. Borrelli, F., Baotić, M., Bemporad, A., Morari, M.: Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. Automatica **41**(10), 1709–1721 (2005)
16. Borrelli, F., Bemporad, A., Morari, M.: Predictive Control for Linear and Hybrid Systems. Cambridge University Press (2017)
17. Busoniu, L., Babuska, R., De Schutter, B., Ernst, D.: Reinforcement Learning and Dynamic Programming Using Function Approximators. CRC Press (2017)
18. Busoniu, L., Babuska, R., Schutter, B.D.: A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) **38**(2), 156–172 (2008)
19. Camacho, E., Bordons, C.: Model Predictive Control, 2nd edn. Springer, Berlin (2007)
20. Chakrabarty, A., Dinh, V., Corless, M.J., Rundell, A.E., Żak, S.H., Buzzard, G.T.: Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences. IEEE Trans. Autom. Control **62**(1), 135–148 (2016)
21. Gharavi, L., De Schutter, B., Baldi, S.: Efficient MPC for emergency evasive maneuvers, Part I: hybridization of the nonlinear problem (2023). arXiv:2310.00715
22. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
23. Görges, D.: Relations between model predictive control and reinforcement learning. IFAC-PapersOnLine **50**(1), 4920–4928 (2017)
24. Gros, S., Zanon, M.: Towards safe reinforcement learning using NMPC and policy gradients: part I—stochastic case (2019). ArXiv:1906.04057v1
25. Gros, S., Zanon, M.: Towards safe reinforcement learning using NMPC and policy gradients: part II—deterministic case (2019). ArXiv:1906.04034v1
26. Gros, S., Zanon, M.: Data-driven economic NMPC using reinforcement learning. IEEE Trans. Autom. Control **65**(2), 636–648 (2020)
27. Gros, S., Zanon, M.: Learning for MPC with stability & safety guarantees. Automatica **146**, 110598 (2022)
28. Hajiahmadi, M., De Schutter, B., Hellendoorn, H.: Design of stabilizing switching laws for mixed switched affine systems. IEEE Trans. Autom. Control **61**(6), 1676–1681 (2016). https://doi.org/10.1109/TAC.2015.2480216
29. Hanin, B.: Universal function approximation by deep neural nets with bounded width and ReLU activations. Mathematics **7**(10), 992 (2019)

30. He, K., Shi, S., Boom, T.V.D., De Schutter, B.: Approximate dynamic programming for constrained piecewise affine systems with stability and safety guarantees (2023). arXiv:2306.15723
31. He, K., Shi, S., Boom, T.V.D., De Schutter, B.: State-action control barrier functions: imposing safety on learning-based control with low online computational costs (2023). arXiv:2312.11255
32. Hertneck, M., Köhler, J., Trimpe, S., Allgöwer, F.: Learning an approximate model predictive controller with guarantees. IEEE Control Syst. Lett. **2**(3), 543–548 (2018)
33. Hewing, L., Wabersich, K., Menner, M., Zeilinger, M.: Learning-based model predictive control: toward safe learning in control. Annu. Rev. Control, Robot., Auton. Syst. **3**(1), 269–296 (2020)
34. Heydari, A.: Revisiting approximate dynamic programming and its convergence. IEEE Trans. Cybern. **44**(12), 2733–2743 (2014)
35. Jain, A., Nghiem, T., Morari, M., Mangharam, R.: Learning and control using Gaussian processes. In: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), pp. 140–149 (2018)
36. Jin, J., Song, C.N., Li, H., Gai, K., Wang, J., Zhang, W.: Real-time bidding with multi-agent reinforcement learning in display advertising. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (2018)
37. Kabzan, J., Hewing, L., Liniger, A., Zeilinger, M.: Learning-based model predictive control for autonomous racing. IEEE Robot. Autom. Lett. **4**(4), 3363–3370 (2019)
38. Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., Levine, S.: Scalable deep reinforcement learning for vision-based robotic manipulation. In: Billard, A. Dragan, A., Peters, J., Morimoto, J. (eds.) Proceedings of The 2nd Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 87, pp. 651–673. PMLR (2018)
39. Kamthe, S., Deisenroth, M.: Data-efficient reinforcement learning with probabilistic model predictive control. In: Twenty-First International Conference on Artificial Intelligence and Statistics, pp. 1701–1710. Playa Blanca, Lanzarote, Canary Islands (2018)
40. Koller, T., Berkenkamp, F., Turchetta, M., Boedecker, J., Krause, A.: Learning-based model predictive control for safe exploration and reinforcement learning (2019). ArXiv:1906.12189v1
41. Lago, J., De Ridder, F., De Schutter, B.: Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms. Appl. Energy **221**, 386–405 (2018)
42. Lazar, M., Heemels, W., Weiland, S., Bemporad, A.: Stabilizing model predictive control of hybrid systems. IEEE Trans. Autom. Control **51**(11), 1813–1818 (2006)
43. Lewis, F., Vrabie, D.: Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst. Mag. **9**(3), 32–50 (2009)
44. Lewis, F., Vrabie, D., Vamvoudakis, K.: Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. IEEE Control Syst. Mag. **32**(6), 76–105 (2012)
45. Li, P., Arellano-Garcia, H., Wozny, G.: Chance constrained programming approach to process optimization under uncertainty. Comput. Chem. Eng. **32**(1–2), 25–45 (2008)
46. Limon, D., Calliess, J., Maciejowski, J.: Learning-based nonlinear model predictive control. IFAC-PapersOnLine **50**(1), 7769–7776 (2017)
47. Lin, M., Sun, Z., Xia, Y., Zhang, J.: Reinforcement learning-based model predictive control for discrete-time systems. IEEE Trans. Neural Netw. Learn. Syst., Early Access (2023)
48. Lunze, J., Lamnabhi-Lagarrigue, F. (eds.): Handbook of Hybrid Systems Control—Theory, Tools, Applications. Cambridge University Press (2009)
49. Maestre, J., Negenborn, R. (eds.): Distributed Model Predictive Control Made Easy. Springer, Berlin (2014)
50. Mallick, S., Airaldi, F., Dabiri, A., De Schutter, B.: Multi-agent reinforcement learning via distributed MPC as a function approximator (2023). arXiv:2312.05166
51. Masti, D., Pippia, T., Bemporad, A., De Schutter, B.: Learning approximate semi-explicit hybrid MPC with an application to microgrids. IFAC-PapersOnLine **53**(2), 5207–5212 (2020)

52. Mehr, N., Sadigh, D., Horowitz, R., Sastry, S.S., Seshia, S.A.: Stochastic predictive freeway ramp metering from signal temporal logic specifications. In: 2017 American Control Conference (ACC), pp. 4884–4889 (2017)
53. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., 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 **518**(7540), 529–533 (2015)
54. Moreno-Mora, F., Beckenbach, L., Streif, S.: Predictive control with learning-based terminal costs using approximate value iteration. IFAC-PapersOnLine **56**(2), 3874–3879 (2023)
55. Ostafew, C., Schoellig, A., Barfoot, T.: Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. Int. J. Robot. Res. **35**(13), 1547–1563 (2016)
56. Postoyan, R., Buşoniu, L., Nešić, D., Daafouz, J.: Stability analysis of discrete-time infinite-horizon optimal control with discounted cost. IEEE Trans. Autom. Control **62**(6), 2736–2749 (2016)
57. Rawlings, J., Mayne, D., Diehl, M.: Model Predictive Control: Theory, Computation, and Design. Nob Hill Publishing (2017)
58. Sadraddini, S., Tedrake, R.: Sampling-based polytopic trees for approximate optimal control of piecewise affine systems. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 7690–7696 (2019)
59. Schildbach, G., Fagiano, L., Frei, C., Morari, M.: The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. Automatica **50**(12), 3009–3018 (2014)
60. Shen, Z., Yang, H., Zhang, S.: Optimal approximation rate of ReLU networks in terms of width and depth. J. de Mathématiques Pures et Appliquées **157**, 101–135 (2022)
61. Silver, D., Huang, A., Maddison, C.J., 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 **529**(7587), 484–489 (2016)
62. Sontag, E.: Nonlinear regulation: the piecewise linear approach **26**(2), 346–358 (1981)
63. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (2018)
64. Valadbeigi, A.P., Sedigh, A.K., Lewis, F.L.: H_∞ static output-feedback control design for discrete-time systems using reinforcement learning. IEEE Trans. Neural Netw. Learn. Syst. **31**(2), 396–406 (2020)
65. Wabersich, K., Hewing, L., Carron, A., Zeilinger, M.: Probabilistic model predictive safety certification for learning-based control. IEEE Trans. Autom. Control **67**, 176–188 (2019)
66. Wabersich, K., Zeilinger, M.: Linear model predictive safety certification for learning-based control. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 7130–7135 (2018)
67. Wabersich, K.P., Taylor, A.J., Choi, J.J., Sreenath, K., Tomlin, C.J., Ames, A.D., Zeilinger, M.N.: Data-driven safety filters: Hamilton–Jacobi reachability, control barrier functions, and predictive methods for uncertain systems. IEEE Control Syst. Mag. **43**(5), 137–177 (2023)
68. Wei, Q., Liu, D., Lin, H.: Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. IEEE Trans. Cybern. **46**(3), 840–853 (2015)
69. Wolsey, L.A., Nemhauser, G.L.: Integer and Combinatorial Optimization. Wiley (1999)
70. Xu, J., van den Boom, T., De Schutter, B.: Optimistic optimization for continuous nonconvex piecewise affine functions. Automatica **125**, 109476 (2021)
71. Yu, C., Liu, J., Nemati, S.: Reinforcement learning in healthcare: a survey. ACM Comput. Surv. (CSUR) **55**, 1–36 (2019)
72. Zanon, M., Gros, S.: Safe reinforcement learning using robust MPC. IEEE Trans. Autom. Control **66**(8), 3638–3652 (2021)
73. Zanon, M., Gros, S., Palladino, M.: Stability-constrained Markov decision processes using MPC. Automatica **143**, 110399 (2022)
74. Zhu, F., Antsaklis, P.: Optimal control of hybrid switched systems: a brief survey. Discret. Event Dyn. Syst. **25**(3), 345–364 (2015)