2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Model-free Data-driven Predictive Control Using Reinforcement Learning

Shambhuraj Sawant, Dirk Reinhardt, Arash Bahari Kordabad, Sebastien Gros

Abstract— **This paper proposes a novel approach for Predictive Control utilizing Reinforcement Learning (RL) and Data-Driven techniques to derive optimal control policies for real systems. Using pure input-output multi-step predictors based on Subspace Identification and RL techniques, the resulting predictive control scheme can approximate the optimal control policy of a system with high accuracy, even if the predictor cannot accurately capture the true system dynamics. One of the key contributions of the proposed approach is the extension of the framework connecting Model Predictive Control (MPC) and RL to one that does not require explicit state-space models, nor to define a notion of state at all. The paper demonstrates the efficacy of the proposed approach through an illustrative example, highlighting the ability of our approach to provide an optimal control policy for a real system without requiring any prior knowledge about its internal dynamics.**

I. Introduction

A. Motivation & Background

Model Predictive Control (MPC) is a widely-used optimal control strategy for multi-variable systems subject to constraints [1], with applications in diverse fields such as power system engineering, processing industry, autonomous vehicles, energy management and robotics. Recent research in MPC has sought to combine MPC with Machine Learning methods to design schemes that can leverage the real system data to enhance closed-loop performances amidst uncertainties, viz. Bayesian Optimization for the derivative-free adaptation of MPC parameters [2], Gaussian Processes for online model refinement [3], [4], and Neural Networks (NNs) to approximate complex MPC schemes via Deep Learning approaches [5]. All these methods share the objective of addressing uncertainties in learning-augmented MPC schemes. A recent survey on this topic can be found in [6].

The authors of [7] have introduced the theory of learning-based MPC schemes using Reinforcement Learning (RL) [8], which we refer to as MPC-based RL framework in this work. They propose an approach to modify MPC schemes to achieve optimal closed-loop performance without depending on the accuracy of the model and, in practice, employ RL techniques to learn these modifications. In this context, various MPC formulations have been explored in subsequent studies. Specifically, robust MPC is detailed in [9], [10], economic MPC in [7], [11], and tracking MPC in [12], [13]. Additionally, Mixed-integer MPC has been discussed in [14], and output-based MPC can be found in [15].

In the MPC-based RL framework, the theories and formulations studied thus far employed classic MPC schemes. These schemes rely on simulation-based predictions using a state-space model of the actual system. However, for some applications, modeling the systems in a state-space form is challenging. This is often because defining a state for the system is difficult, as seen in cases like modeling energy utilization in buildings or problems in soft robotics. In such cases, predictive control schemes that are based on predictions in the input-output space of the underlying system can offer a simpler solution compared to formulating its state-space model. We refer to these schemes as data-driven predictive control schemes in the following. Specifically, input-output predictors leveraging Subspace Identification (SI) [16] can deliver purely data-driven, reliable multi-step predictors. These predictors can be directly embedded within Subspace Predictive Control schemes (SPC) [17]. We use the term *model-free* to describe these predictive control schemes. Typically, a *model* refers to a dynamical system representation that predicts the next state based on the current state and action, continuously looping over for predicting states throughout the optimization horizon. In contrast, for SPC, the multi-step predictor is a static function that replaces the model and works without any defined notion of state. SPC tends to provide a more dependable method for developing purely data-driven predictive control schemes compared to other data-driven alternatives.

However, the performance of SPC relies on the accuracy of the predictions delivered by the multi-step predictor. Due to stochasticity or various uncertainties, it may deliver inaccurate predictions of the real system trajectories. Consequently, the resulting SPC scheme may often produce sub-optimal policies.

B. Contribution

In this paper, our objective is to extend the theory of the MPC-based RL framework, as presented in [7], to predictive control schemes that utilize purely data-driven linear multi-step predictors. We will show that these predictive control schemes can be tuned to deliver the optimal policy for the real system, even with a possibly inaccurate multi-step predictor. This central result facilitates the development of a purely data-driven predictive control framework. The framework does not require explicit knowledge of the physics governing the system dynamics or an understanding of the system's state space. It achieves optimal closed-loop control performance even with potential inaccuracies in the predictors. This result presents a significant step forward compared to alternative purely data-driven predictive control techniques.

The authors are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. E-mail:{shambhuraj.sawant, dirk.p.reinhardt, arash.b.kordabad and sebastien.gros}@ntnu.no

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

C. Structure of the paper

The paper is structured as follows. Section II-A provides background material on Markov Decision Processes using input-output information, Subspace Predictive Control and the MPC-based RL framework. Section III presents the extension of the learning-based MPC to SPC schemes while section IV details the implementation of RL techniques for SPC. Section V presents simulation results illustrating the functioning of the proposed formulation and Section VI provides conclusions.

II. Background

In this section, we detail the problem setup considered in this work and provide a brief overview of SPC [17], and the MPC-based RL framework presented in [7].

A. Markov Decision Process on Input-Output information

In this paper, we consider systems where the state is unknown, but the actions (or inputs) $\boldsymbol{a}$ applied to the system are available, as well as measurements (or outputs) $\boldsymbol{y}$. We will then consider stochastic output dynamics where at the discrete time t, the next output $\boldsymbol{y}_{t+1}$ depends on the past outputs observed in the system and the past actions of the system in a probabilistic sense.

To define the system more formally, let us consider finite sequences of past inputs and outputs at a given time t. These sequences serve as initial conditions for modelling the output dynamics of the system. Specifically, we define,

$$\boldsymbol{a}_t^{\text{ini}} = [\,\boldsymbol{a}_{t-T_{\text{ini}}}, \dots, \boldsymbol{a}_{t-1}\,]^T \tag{1}$$

$$\boldsymbol{y}_t^{\text{ini}} = [\,\boldsymbol{y}_{t-T_{\text{ini}}}, \dots, \boldsymbol{y}_{t}\,]^T \tag{2}$$

for a given length of history T_{ini}. We will then assume that the next output $\boldsymbol{y}_{t+1}$ is stochastic, but that $\{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\}$ provide a "complete statistics" to determine $\boldsymbol{y}_{t+1}$, in the formal sense provided below.

Assumption 1. T_{ini} *is such that the conditional distribution (or measure):*

$$\boldsymbol{y}_{t+1} \sim \varrho(\,\cdot\,|\,\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t\,) \tag{3}$$

remains unaffected for any $T'_{\text{ini}} > T_{\text{ini}}$.

Under assumption 1, the recent history $\{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\}$ of length T_{ini} is sufficient to determine the statistics of the next output $\boldsymbol{y}_{t+1}$ for a given action $\boldsymbol{a}_t$. We then consider Markov Decision Processes based on $\{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\}$ rather than on the state information of the system. More specifically, let us consider the problem of defining an optimal policy $\boldsymbol{\pi}^\star$ minimizing

$$J(\boldsymbol{\pi}) = \mathbb{E}_\varrho\left[\sum_{t=0}^{\infty} \gamma^t L\left(\boldsymbol{y}_t, \boldsymbol{a}_t\right) \,\middle|\, \boldsymbol{a}_t = \boldsymbol{\pi}\left(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\right)\right], \tag{4}$$

where L is a given stage cost that additionally captures any penalty for constraint violation, $\gamma \in (0,1]$ a discount factor, and the expected value $\mathbb{E}_\varrho[.]$ is taken over the closed-loop trajectories stemming from (3).

We are interested in using deterministic input-output predictive control techniques to provide policies approaching $\boldsymbol{\pi}^\star$ using data obtained from the real system. We detail next the predictive control methods investigated in this paper. To simplify the discussion, we will make the additional assumption that (3) is linear in expected value, i.e. that $\mathbb{E}_\varrho\left[\boldsymbol{y}_{t+1}\,|\,\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t\right]$ is linear in $\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t$, such that dynamics (3) can arguably be modelled using linear techniques. This assumption is in principle not crucial, but it will avoid a more complex presentation.

B. Data-Driven Predictive Control

This paper explores a data-driven predictive control scheme that originates from Subspace Identification [16], and Subspace Predictive Control (SPC) [17]. Our focus will be on using a predictive control scheme that takes the following form,

$$\min_{\boldsymbol{u},\boldsymbol{y}} \quad \gamma^N T(\boldsymbol{y}_{t+N}) + \sum_{k=0}^{N-1} \gamma^k L(\boldsymbol{y}_{t+k}, \boldsymbol{u}_{t+k}) \tag{5a}$$

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{y}_{t+1} \\ \vdots \\ \boldsymbol{y}_{t+N} \end{bmatrix} = \Phi \begin{bmatrix} \boldsymbol{a}_t^{\text{ini}} \\ \boldsymbol{y}_t^{\text{ini}} \\ \hdashline \boldsymbol{u}_t \\ \vdots \\ \boldsymbol{u}_{t+N-1} \end{bmatrix} \tag{5b}$$

where matrix Φ follows a specific structure and provides a prediction of the output trajectories for a given set of initial conditions, $\{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\}$, and a future input sequence $\boldsymbol{u}_{t,\dots,t+N-1}$ with $\gamma \in (0,1]$. Here, T represents a terminal cost that compensates for the finite horizon N. The first element of the control input sequence $\boldsymbol{u}^\star$ from the solution of (5), i.e. $\boldsymbol{u}_t^\star$, is used as action on the real system. As a result, the SPC scheme produces the following policy,

$$\boldsymbol{\pi}^{\text{SPC}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \boldsymbol{u}_t^\star\,. \tag{6}$$

In a data-driven context, Φ is typically provided by regression techniques applied to the input-output data available from the system. A least-squares regression, for instance, takes the form,

$$\min_{\Phi} \quad \frac{1}{2}\sum_{t\in D} \left\| \begin{bmatrix} \boldsymbol{y}_{t+1} \\ \vdots \\ \boldsymbol{y}_{t+N} \end{bmatrix} - \Phi \begin{bmatrix} \boldsymbol{a}_t^{\text{ini}} \\ \boldsymbol{y}_t^{\text{ini}} \\ \hdashline \boldsymbol{u}_t \\ \vdots \\ \boldsymbol{u}_{t+N-1} \end{bmatrix} \right\|^2 \tag{7a}$$

$$\text{s.t.} \quad \Phi = \left[\begin{array}{c:c} \Phi_{\text{P}} & \Phi_{\text{F}} \end{array}\right] \tag{7b}$$

$$\Phi_{\text{F}} \text{ lower-block triangular} \tag{7c}$$

where $\Phi_{\text{P}}, \Phi_{\text{F}}$ linearly maps history $\{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\}$ and and a future input sequence $\boldsymbol{u}_{t,\dots,t+N-1}$ to observed measurements $\boldsymbol{y}_{t+1,\dots,t+N}$ with Φ_{F} as a lower-block triangular matrix to preserve causality. Here, D is a set of time indices for which data are available up to time $t+N$. Although alternative regression techniques may be relevant for building Φ on stochastic systems, we focuse on (7) to build the predictor in the current work.

However, building the predictor Φ from (7) (or alternative regressions) does not guarantee that the resulting SPC scheme

(5) will deliver a good policy $\boldsymbol{\pi}_{\mathrm{SPC}}$, especially in the presence of stochasticity in the real system dynamics. In this paper, we tackle this issue using the recent developments in learning-based MPC using RL and extending them to the input-output predictive control context. These recent developments are tailored to state-space formulations, and require some careful modifications to apply them in the SPC context. The following section recalls the MPC-based RL formulation.

C. MPC-based Reinforcement Learning

A classic Markov Decision Process (MDP) operates over a given state space $\mathcal{S}$ and an action space $\mathcal{A}$. Similar to (3), the state transitions occur according to,

$$\boldsymbol{s}_{t+1} \sim \rho(\,\cdot\,|\,\boldsymbol{s}_t, \boldsymbol{a}_t)\,, \tag{8}$$

where the closed-loop performance is given by,

$$J(\boldsymbol{\pi}) = \mathbb{E}_\rho \left[\sum_{t=0}^{\infty} \gamma^t L\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) \,\middle|\, \boldsymbol{a}_t = \boldsymbol{\pi}\left(\boldsymbol{s}_t\right) \right], \tag{9}$$

and the aim is to minimize it using a policy $\boldsymbol{\pi}^\star$ operating over $\mathcal{S}$. Finding an optimal policy for an MDP typically involves computing the optimal value function $V^\star : \mathcal{S} \to \mathbb{R}$ and the optimal action-value function $Q^\star : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defined by the Bellman equations,

$$V^\star(\boldsymbol{s}_t) = \min_{\boldsymbol{a}_t} Q^\star(\boldsymbol{s}_t, \boldsymbol{a}_t)\ , \tag{10a}$$

$$\boldsymbol{\pi}^\star(\boldsymbol{s}_t) = \arg\min_{\boldsymbol{a}_t} Q^\star(\boldsymbol{s}_t, \boldsymbol{a}_t)\ , \tag{10b}$$

$$Q^\star(\boldsymbol{s}_t, \boldsymbol{a}_t) = L(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \mathbb{E}_\rho\left[V^\star(\boldsymbol{s}_{t+1})\,|\,\boldsymbol{s}_t, \boldsymbol{a}_t\right]\ . \tag{10c}$$

Computing a solution to the Bellman equations (10) for given L, ρ, and γ is notoriously difficult. When the transition probability distribution ρ is unknown, RL methods seek approximate solutions to (10) using data from the real system. This is often done by using generic function approximators for approximating $V^\star$, $Q^\star$ and $\boldsymbol{\pi}^\star$, such as neural networks.

In contrast to seeking approximate solutions to (10) using data, which can be challenging when ρ is unknown, an MPC scheme can deliver $V^\star$, $Q^\star$, and $\boldsymbol{\pi}^\star$ exactly, even when using an inaccurate model of the system dynamics, as demonstrated by [7]. To achieve this, modifications are made to the cost and constraints forming the MPC scheme, which can be learned using RL techniques. Specifically, the modified MPC scheme for a given state $\boldsymbol{s}_t$ is formulated as,

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad \gamma^N T_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t+N}\right) + \sum_{k=0}^{N-1} \gamma^k L_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}\right) \tag{11a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}\right) \tag{11b}$$

$$\boldsymbol{x}_t = \boldsymbol{s}_t \tag{11c}$$

All elements in the MPC scheme (11) are parameterized by a parameter vector $\boldsymbol{\theta}$. The model $\boldsymbol{f}_{\boldsymbol{\theta}}$ is a representation of the true system dynamics described in (8). Typically, $\boldsymbol{f}_{\boldsymbol{\theta}}$ fails to provide accurate predictions of the true system dynamics. In such cases, the parameterized stage and terminal cost functions $L_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$ can be modified to compensate for the model inaccuracy to deliver an optimal policy.

Similar to (5), the MPC scheme in (11) generates an input sequence and a predicted state sequence. Only the first element of the input sequence is applied, resulting in a policy of the form,

$$\boldsymbol{\pi}_\theta\left(\boldsymbol{s}_t\right) = \boldsymbol{u}_t^\star\,. \tag{12}$$

Additionally, (11) delivers an action-value function,

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \min_{\boldsymbol{x},\boldsymbol{u}} \quad \text{(11a)}, \tag{13a}$$

$$\text{s.t.} \quad \text{(11b)} - \text{(11c)}, \quad \boldsymbol{u}_t = \boldsymbol{a}_t \tag{13b}$$

as well as a value function,

$$V_{\boldsymbol{\theta}}(\boldsymbol{s}_t) = \min_{\boldsymbol{a}_t} Q_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{14}$$

which equates with the cost of (11) at its solution. In [7], it is shown that, under a mild assumption on $\boldsymbol{f}_{\boldsymbol{\theta}}$, if $T_{\boldsymbol{\theta}}$ and $L_{\boldsymbol{\theta}}$ are richly parametrized then there exists a $\boldsymbol{\theta}$ such that:

$$V_{\boldsymbol{\theta}} = V^\star, \quad Q_{\boldsymbol{\theta}} = Q^\star, \quad \boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}^\star. \tag{15}$$

Using these observations, [7] concluded that the MPC formulation in (11), where the parameters $\boldsymbol{\theta}$ are adjusted using RL techniques, allows reaching high closed-loop performance even if the MPC model (11b) cannot capture the true system dynamics accurately. The next section extends this framework to the SPC case, where the notion of state space is not used explicitly.

III. Optimal Policies from Data Driven Predictive Control

This section extends the framework summarized in Sec. II-C to the SPC case (5). The most straightforward approach to establish this extension is arguably to form a Markov state for the real system based on input-output sequences, and apply the framework summarized in Sec. II-C in that context. In the following section, we define such a state and elaborate on the background presented in Sec. II-A.

A. MDP with input-output information

Under assumption 1, forming a Markov state for a system of input-output dynamics (3) is fairly straightforward. We formalize that state in the following simple Lemma.

Lemma 1. *Under Assumption 1,*

$$\bar{\boldsymbol{s}}_t := \left[\,\boldsymbol{a}_t^{\mathrm{ini}}, \boldsymbol{y}_t^{\mathrm{ini}}\,\right]^T \tag{16}$$

is a Markov state.

Proof. State $\bar{\boldsymbol{s}}_t$ is a Markov state if and only if

$$\bar{\boldsymbol{s}}_{t+1} \sim \rho(\,.\,|\,\bar{\boldsymbol{s}}_t, \boldsymbol{a}_t\,) = \rho(\,.\,|\,\bar{\boldsymbol{s}}_t, \boldsymbol{a}_t, \bar{\boldsymbol{s}}_{t-1}, \boldsymbol{a}_{t-1}, \ldots\,)\,, \tag{17}$$

i.e. if $\bar{\boldsymbol{s}}_t$ contains the required information to provide complete statistics for the next state $\bar{\boldsymbol{s}}_{t+1}$. Using (16), $\bar{\boldsymbol{s}}_{t+1}$ reads as:

$$\bar{\boldsymbol{s}}_{t+1} = \left[\boldsymbol{a}_{t-T_{\mathrm{ini}}+1}, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{a}_t, \boldsymbol{y}_{t-T_{\mathrm{ini}}+1}, \ldots, \boldsymbol{y}_t, \boldsymbol{y}_{t+1}\right]^T. \tag{18}$$

We then observe that the first part of $\bar{\boldsymbol{s}}_{t+1}$, i.e.

$$\left[\boldsymbol{a}_{t-T_{\mathrm{ini}}+1}, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{a}_t, \boldsymbol{y}_{t-T_{\mathrm{ini}}+1}, \ldots, \boldsymbol{y}_t\right]^T. \tag{19}$$

is a trivial, deterministic function of $\bar{\boldsymbol{s}}_t$ and an input $\boldsymbol{a}_t$. Moreover, we observe that under assumption 1, $\bar{\boldsymbol{s}}_t$ provides a complete statistics for the last element $\boldsymbol{y}_{t+1}$ of $\bar{\boldsymbol{s}}_{t+1}$ for any input $\boldsymbol{a}_t$. Then, it follows that $\bar{\boldsymbol{s}}_t$ provides a complete statistics for $\bar{\boldsymbol{s}}_{t+1}$, such that $\bar{\boldsymbol{s}}_t$ is a Markov state. ■

Using $\bar{\boldsymbol{s}}_t$, one can trivially define an MDP equivalent to the input-output MDP defined in Sec. II-A, based on $\bar{\boldsymbol{s}}_t$ as a state, $\boldsymbol{a}_t$ as action, with a stage cost,

$$\bar{L}(\bar{\boldsymbol{s}}_t, \boldsymbol{a}_t) := L(\boldsymbol{y}_t, \boldsymbol{a}_t)\,, \tag{20}$$

and the state transition dynamics (17). Then the closed-loop performance criterion (9) and the associated Bellman equations (10) naturally apply. In particular, one can define optimal value functions associated with (4) as,

$$V^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \min_{\boldsymbol{a}_t} Q^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t), \tag{21a}$$

$$\boldsymbol{\pi}^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \arg\min_{\boldsymbol{a}_t} Q^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) \tag{21b}$$

$$\begin{aligned} Q^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) &= L(\boldsymbol{y}_t, \boldsymbol{a}_t) \\ &\quad + \gamma \mathbb{E}_\rho\left[V^\star(\boldsymbol{a}_{t+1}^{\text{ini}}, \boldsymbol{y}_{t+1}^{\text{ini}}) \,|\, \boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t\right] \end{aligned} \tag{21c}$$

where $\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}$ and $\bar{\boldsymbol{s}}_t$ can be loosely interchanged.

Using these observations, the framework summarized in Sec. II-C describing the use of RL for learning-based MPC can be extended to the SPC formulation (5). The next section elaborates more formally on this statement and points to some important details where the SPC framework will differ from the classic framework of Sec. II-C.

B. Optimal Policy using SPC

In this section we provide a modified SPC scheme capable of capturing the optimal policy and value functions associated with the input-output MDP introduced in Sec. II-A and further detailed in Sec. III-A. This SPC scheme will have a structure nearly similar to SPC (5), but use modifications of the stage and terminal costs, along the lines of Sec. II-C. However, unlike the MPC case (11), in the SPC case, the arguments of the modified costs need to be different than the original SPC (5). These observations are formally detailed in the following proposition.

Proposition 1. *Consider the SPC scheme*

$$\min_{\boldsymbol{u},\boldsymbol{y}} \quad \gamma^N T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+N}) + \sum_{k=0}^{N-1} \gamma^k L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+k}, \boldsymbol{u}_{t+k}) \tag{22a}$$

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{y}_{t+1} \\ \vdots \\ \boldsymbol{y}_{t+N} \end{bmatrix} = \Phi \left[\begin{array}{c} \boldsymbol{a}_t^{\text{ini}} \\ \boldsymbol{y}_t^{\text{ini}} \\ \hline \boldsymbol{u}_t \\ \vdots \\ \boldsymbol{u}_{t+N-1} \end{array}\right] \tag{22b}$$

where the predicted state $\hat{\boldsymbol{s}}_{t,\dots,t+N}$ *is constructed from the initial conditions* $\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}$*, the future inputs* $\boldsymbol{u}_{t,\dots,t+N-1}$ *and the predicted output* $\boldsymbol{y}_{t,\dots,t+N}$ *as follows:* $\hat{\boldsymbol{s}}_t = [\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}]^T$*, for* $0 < k < T_{\text{ini}}$

$$\begin{aligned} \hat{\boldsymbol{s}}_{t+k} =& [\,\boldsymbol{a}_{t-T_{\text{ini}}+k}, \dots, \boldsymbol{a}_{t-1}, \boldsymbol{u}_t, \dots, \boldsymbol{u}_{t+k-1}, \\ & \boldsymbol{y}_{t-T_{\text{ini}}+1}, \dots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_t, \dots, \boldsymbol{y}_{t+k}\,]^T \end{aligned} \tag{23}$$

while for $k \geq T_{\text{ini}}$

$$\hat{\boldsymbol{s}}_{t+k} = [\,\boldsymbol{u}_{t-T_{\text{ini}}+k}, \dots, \boldsymbol{u}_{t+k-1}, \boldsymbol{y}_{t-T_{\text{ini}}+1}, \dots, \boldsymbol{y}_{t+k}\,]^T. \tag{24}$$

Consider the set Ω *of initial conditions* $\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}$ *defined as:*

$$\Omega := \{\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}} \,|\, |V^\star(\hat{\boldsymbol{s}}_{t+k})| < \infty,\, k = 0, \dots, N-1\}\,. \tag{25}$$

Then for $T_{\boldsymbol{\theta}}, L_{\boldsymbol{\theta}}$ *richly parametrized, there exist a* $\boldsymbol{\theta}$ *and* Φ *such that the following identities hold over* Ω*:*

i. $V^{\text{SPC}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = V^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$.

ii. $\boldsymbol{\pi}^{\text{SPC}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \boldsymbol{\pi}^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$.

iii. $Q^{\text{SPC}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) = Q^\star(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t)$ *for an input* $\boldsymbol{a}_t$ *such that* $|V^\star(\hat{\boldsymbol{s}}_{t+1})| < \infty$.

Proof. Consider the set of Φ stemming from a state space representation of the dynamics of $\hat{\boldsymbol{s}}_t$, more specifically, consider that Φ takes the form:

$$\Phi = \begin{bmatrix} \tau_1(0) & 0 & 0 & \dots & 0 \\ \tau_1(1) & \tau_2(0) & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \tau_1(N-1) & \tau_2(N-2) & \dots & \dots & 0 \end{bmatrix} \tag{26}$$

where $\tau_1(k) = CA^k$ and $\tau_2(k) = CA^kB$ for some matrices A, B, C in the observable companion form associated with a deterministic output predictor model of (3) in the form,

$$\boldsymbol{y}_{t+1} = \phi_{a^{\text{ini}}}^\top \boldsymbol{a}_t^{\text{ini}} + \phi_{y^{\text{ini}}}^\top \boldsymbol{y}_t^{\text{ini}} + \phi_{\boldsymbol{a}}^\top \boldsymbol{a}_t. \tag{27}$$

for some vectors $\phi_{a^{\text{ini}}}$, $\phi_{y^{\text{ini}}}$, $\phi_{\boldsymbol{a}}$. Then SPC (22) is equivalent to the MPC scheme:

$$\min_{\boldsymbol{u},\hat{\boldsymbol{s}}} \quad \gamma^N T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+N}) + \sum_{k=0}^{N-1} \gamma^k L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+k}, \boldsymbol{u}_{t+k}) \tag{28a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{s}}_{t+k+1} = A\hat{\boldsymbol{s}}_{t+k} + B\boldsymbol{u}_{t+k} \tag{28b}$$

$$\hat{\boldsymbol{s}}_t = [\,\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}\,]^T \tag{28c}$$

Then [7, Theorem 1] applies to (28), i.e. for $T_{\boldsymbol{\theta}}$, $L_{\boldsymbol{\theta}}$ richly parametrized, there is a vector of parameters $\boldsymbol{\theta}$ such that (28) delivers the optimal value functions and policy $Q^\star$, $V^\star$, $\boldsymbol{\pi}^\star$ over Ω.

■

A few remarks are in order here regarding Preposition 1.

1) Matrix Φ in the SPC scheme (22) is arguably best done via regression methods on existing data, e.g. using (7) or alternative loss functions. However, a matrix Φ obtained from (22) is unlikely to have the structure (26) used in the proof of Proposition 1. We ought to observe, though, that this is not an issue in the context of the argumentation provided here. Indeed, Proposition 1 focuses on proving the existence of a $\boldsymbol{\theta}$, Φ such that SPC (22) delivers the optimal policy and value functions. Then the conclusions of Proposition 1 hold for an SPC (22) having a less restrictive structure for Φ.

2) The assumption that the parametrization of $T_{\boldsymbol{\theta}}, L_{\boldsymbol{\theta}}$ is rich typically does not hold in practice. Then in practice, Proposition 1 holds in the sense that the richer the parametrization of $T_{\boldsymbol{\theta}}, L_{\boldsymbol{\theta}}$, the closer SPC (22) can approach the optimal policy and value functions.
3) A crucial difference between modified MPC (11) and modified SPC (22) ought to be pointed out. Indeed, in modified MPC (11) the cost modification $L_{\boldsymbol{\theta}}$ operates on the system state stage-wise, i.e. it does not mix the states at different time instants. In contrast, because the Markovian state (17) blends present and past input-output information, the modified stage cost $L_{\boldsymbol{\theta}}$ in (22) combines input-output prediction across stages. Hence modified SPC (22) is structurally different than (5).
4) Computing a $\boldsymbol{\theta}$, Φ such that SPC (22) comes as close as possible to delivering the value function and policies is very difficult. Following the arguments of [7], $\boldsymbol{\theta}$, Φ are then best adjusted using RL techniques using data obtained from the real system. In that context, an initial Φ is computed from (7) and an initial vector of parameters $\boldsymbol{\theta}$ is selected such that the SPC scheme (22) corresponds to SPC (5), typically yielding a reasonably good yet sub-optimal closed-loop performance. RL is then used to adjust $\boldsymbol{\theta}$, Φ for performance improvement.

In the next section, we detail the use RL techniques for tuning SPC schemes.

IV. RL FOR DATA-DRIVEN PREDICTIVE CONTROL

Proposition 1 guarantees the existence of $\boldsymbol{\theta}$ and Φ such that the SPC scheme delivers an optimal policy and value functions. However, computing such $\boldsymbol{\theta}$ and Φ is difficult and requires knowledge of the true system dynamics. Thus, following the suggestions in [7], we make use of RL techniques to adjust them for better closed-loop performance. In the following section, we present a discussion on integrating the SPC scheme in (22) with classical RL techniques.

Consider the following parametric SPC scheme for approximating the value function,

$$V_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \min_{\boldsymbol{y},\boldsymbol{u}} \gamma^N T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+N}) + \sum_{k=0}^{N-1} \gamma^k L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+k}, \boldsymbol{u}_{t+k}) \tag{29a}$$

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{y}_{t+1} \\ \vdots \\ \boldsymbol{y}_{t+N} \end{bmatrix} = \Phi \begin{bmatrix} \boldsymbol{a}_t^{\text{ini}} \\ \boldsymbol{y}_t^{\text{ini}} \\ \boldsymbol{u}_t \\ \vdots \\ \boldsymbol{u}_{t+N-1} \end{bmatrix} \tag{29b}$$

where the predicted state $\hat{\boldsymbol{s}}_{t,\dots,t+N}$ is constructed as in (23) and (24), and the parameter $\boldsymbol{\Theta}$ is a vector consisting of $\boldsymbol{\theta}$ and the parameterization in Φ. Then, similar to (5), the resulting control policy is

$$\boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \boldsymbol{u}_t^\star \tag{30}$$

where $\boldsymbol{u}_t^\star$ is the first input element from solution of (29). The SPC-based action-value function is given as,

$$Q_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) = \min_{\boldsymbol{y},\boldsymbol{u}} \quad \text{(29a)}, \tag{31a}$$

$$\text{s.t.} \quad \text{(29b)}, \quad \boldsymbol{u}_t = \boldsymbol{a}_t \tag{31b}$$

For learning the parameter $\boldsymbol{\Theta}$, RL techniques typically require sensitivities of value functions and control policy. We next discuss how to evaluate the gradients of $V_{\boldsymbol{\Theta}}$, $Q_{\boldsymbol{\Theta}}$ and $\boldsymbol{\pi}_{\boldsymbol{\Theta}}$ with respect to $\boldsymbol{\Theta}$. To that end, consider the Lagrange function associated with SPC in (31),

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t, \boldsymbol{z}) =\ & \gamma^N T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+N}) + \boldsymbol{\zeta}^T(\boldsymbol{u}_t - \boldsymbol{a}_t) \\ & + \sum_{k=0}^{N-1} \gamma^k L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{t+k}, \boldsymbol{u}_{t+k}) \\ & + \boldsymbol{\mu}^T \left(\begin{bmatrix} \boldsymbol{y}_{t+1} \\ \vdots \\ \boldsymbol{y}_{t+N} \end{bmatrix} - \Phi \begin{bmatrix} \boldsymbol{a}_t^{\text{ini}} \\ \boldsymbol{y}_t^{\text{ini}} \\ \boldsymbol{u}_t \\ \vdots \\ \boldsymbol{u}_{t+N-1} \end{bmatrix} \right)\end{aligned} \tag{32}$$

where $\boldsymbol{\mu}, \boldsymbol{\zeta}$ are the multipliers for (29b) and (31b), respectively, and $\boldsymbol{z} = (\boldsymbol{u}_t, \dots, \boldsymbol{u}_{t+N-1}, \boldsymbol{y}_{t+1}, \dots, \boldsymbol{y}_{t+N}, \boldsymbol{\mu}, \boldsymbol{\zeta})$ represents all the primal-dual variables associated with (31). Then, with [18], we observe that,

$$\nabla_{\boldsymbol{\Theta}} Q_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) = \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t, \boldsymbol{z}^\star) \tag{33}$$

for $\boldsymbol{z}^\star$ as the primal-dual solution of (31). Similarly, the gradient of the value function $V_{\boldsymbol{\Theta}}$ with respect to $\boldsymbol{\Theta}$ is

$$\nabla_{\boldsymbol{\Theta}} V_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{u}_t^\star, \boldsymbol{z}^\star) \tag{34}$$

where $\boldsymbol{z}^\star$ is the primal-dual solution of (29) ($\boldsymbol{\zeta} = 0$). Finally, the gradient of $\boldsymbol{\pi}_{\boldsymbol{\Theta}}$ with respect to parameters $\boldsymbol{\Theta}$ is

$$\begin{aligned}\nabla_{\boldsymbol{\Theta}} \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) =\ & -\nabla_{\boldsymbol{\Theta}} \xi_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{z}^\star) \\ & \cdot \nabla_{\boldsymbol{z}} \xi_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{z}^\star)^{-1} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{u}_t}\end{aligned} \tag{35}$$

where $\boldsymbol{z}^\star$ is the primal-dual solution of (29) ($\boldsymbol{\zeta} = 0$) and $\boldsymbol{\xi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{z}^\star)$ gathers the primal-dual KKT conditions for (29). With the sensitivities of SPC-based value functions and policy defined, we next discuss the use of classical RL methods in learning $\boldsymbol{\Theta}$.

A. Q-learning for SPC

Q-learning [19] solves the following least square problem in order to achieve the best parameters $\boldsymbol{\Theta}^\star$ for describing the optimal action-value function $Q^\star$,

$$\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\Theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) - Q^\star(\boldsymbol{s}_t, \boldsymbol{a}_t) \right)^2 \right] . \tag{36}$$

Then the temporal difference learning update rule is,

$$\delta_t = L(\boldsymbol{y}_t, \boldsymbol{a}_t) + \gamma V_{\boldsymbol{\Theta}}(\bar{\boldsymbol{s}}_{t+1}) - Q_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t), \tag{37a}$$

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \alpha \delta_t \nabla_{\boldsymbol{\Theta}} Q_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t), \tag{37b}$$

where $\alpha > 0$ is the learning rate.

B. Deterministic Policy Gradient Methods for SPC

Policy gradient methods [20] directly optimize the closed-loop performance (4). With the Deterministic Policy Gradient (DPG) theorem [20], the parameter update rule is,

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \alpha \mathbb{E}\Big[\nabla_{\boldsymbol{\Theta}} \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\Theta}}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t)\Big] \tag{38}$$

with the action-value function approximated separately.

V. Simulations

In this section, we illustrate the efficacy of the proposed combination of SPC and RL using a Point Mass example.

A. Experimental Setup

1) Point Mass: We consider a Point Mass task to illustrate our approach to learning-based SPC on input-output data. The goal is to push a point mass to the origin of the state space with its state vector as $\boldsymbol{s}_t = [x, y, \dot{x}, \dot{y}]^T$ consisting of position and velocities of the point mass and an action vector $\boldsymbol{a}_t = [F_x, F_y]^T$ as the applied forces along x and y directions. The observation space is limited to position information, $\boldsymbol{y}_t = [x, y]^T$, for our setup. The true system dynamics for the task is given by,

$$\boldsymbol{s}_{t+1} = A\,\boldsymbol{s}_t + B\,\boldsymbol{a}_t + \mathcal{N}(0, 0.02)\,, \tag{39}$$

$$A = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

while the stage cost is given as

$$L(\boldsymbol{y}_t, \boldsymbol{a}_t) = 9\|\boldsymbol{y}_t\|^2 + 0.01\|\boldsymbol{a}_t\|^2 - 100(x)_-\,, \tag{40}$$

where function $(m)_-$ equals m if $m < 0$ and zero otherwise and it penalizes crossing to the negative half of the state space. However, the true state dynamics in (39) and the state space description are considered to be unknown, necessitating formulating a predictive control scheme over the input-output information. Note here, that for the output $\boldsymbol{y}_t$, the minimum length of history for having complete statistics of the next output $\boldsymbol{y}_{t+1}$ is 2. Finally, the discount factor γ is set to be 0.95, and the task length is 50.

2) SPC parameterization: For the illustrative example, we parameterized (29) as follows:

$$L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k, \boldsymbol{u}_k) = \boldsymbol{y}_k^T W W^T \boldsymbol{y}_k + \boldsymbol{a}_k^T R R^T \boldsymbol{a}_k \tag{41}$$

$$T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k, \boldsymbol{u}_k) = 0 \tag{42}$$

$$\Phi = \left[\begin{array}{c:c} \Phi_{\text{P}} & \Phi_{\text{F}} \end{array}\right] \tag{43}$$

where Φ_{P} and Φ_{F} are defined as in (7) and are fully parameterized while the stage cost weight matrices are

$$W = \begin{bmatrix} \theta_1 & 0 \\ \theta_2 & \theta_3 \end{bmatrix}, \quad R = \begin{bmatrix} \theta_4 & 0 \\ \theta_5 & \theta_6 \end{bmatrix}. \tag{44}$$

The parameters $\boldsymbol{\theta}$ are initiated according to (40) and Φ is initiated to the least squares solution of (7). Additionally, history length T_{ini} and horizon N are set to 5 and 10, respectively. Note that the SPC scheme is built with a longer length of history than required as the minimum required length is usually unknown.

3) DPG with LSTDQ learning: For learning $\boldsymbol{\Theta}$, we use a Temporal Difference actor-critic setup [21] based on DPG theorem [20] with an SPC-based actor and a critic using compatible function approximation. Specifically, the actor $\boldsymbol{\pi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$ is given by,

$$\boldsymbol{\pi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) = \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) + \mathcal{U}(-0.1, 0.1)\,, \tag{45}$$

while the action-value function $Q(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t)$ and value function $V(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$ are approximated as,

$$\begin{aligned} Q_{\boldsymbol{w}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) &= V(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) + (\boldsymbol{a}_t - \boldsymbol{\pi}_\theta(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}))^T \\ &\quad \cdot \nabla_\theta \boldsymbol{\pi}_\theta(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})^T \boldsymbol{w}\,, && (46) \\ V(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) &= \boldsymbol{\beta}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})^T \boldsymbol{v}\,. && (47) \end{aligned}$$

The feature vector $\boldsymbol{\beta}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$ is chosen to be all monomials of $(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})$ with degrees ≤ 2. The vectors $\boldsymbol{w}, \boldsymbol{v}$ are tuned using temporal difference learning, such that Q approximates the optimal action-value function $Q^\star$, as in (36). The least squares solutions for $\boldsymbol{w}, \boldsymbol{v}$ are,

$$\begin{aligned} \boldsymbol{v} &= \mathbb{E}\left[\boldsymbol{\beta}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})(\boldsymbol{\beta}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) - \gamma \boldsymbol{\beta}(\bar{\boldsymbol{s}}_{t+1}))^T\right]^{-1} \\ &\quad \mathbb{E}\left[\boldsymbol{\beta}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) L(\boldsymbol{y}_t, \boldsymbol{a}_t)\right], && (48) \\ \boldsymbol{w} &= \mathbb{E}\left[\boldsymbol{\psi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) \boldsymbol{\psi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t)^T\right]^{-1} \\ &\quad \mathbb{E}\Big[(L(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) + \gamma V_v(\bar{\boldsymbol{s}}_{t+1}) - V_v(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})) \\ &\quad \cdot \boldsymbol{\psi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t)\Big], && (49) \end{aligned}$$

where expectations are taken over on-policy data (20 episodes) and $\boldsymbol{\psi}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) = \nabla_{\boldsymbol{\Theta}} \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})(\boldsymbol{a}_t - \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}))$. For (46), we have,

$$\nabla_{\boldsymbol{a}} Q_{\boldsymbol{w}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}, \boldsymbol{a}_t) = \nabla_{\boldsymbol{\Theta}} \pi_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})^T \boldsymbol{w}\,. \tag{50}$$

Thus, the parameter update rule using (38) is,

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \alpha \nabla_{\boldsymbol{\Theta}} \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}}) \nabla_{\boldsymbol{\Theta}} \boldsymbol{\pi}_{\boldsymbol{\Theta}}(\boldsymbol{a}_t^{\text{ini}}, \boldsymbol{y}_t^{\text{ini}})^T \boldsymbol{w} \tag{51}$$

where $\alpha > 0$ is the small enough learning rate. We use $\alpha_{\boldsymbol{\theta}} = 1e-5$ and $\alpha_\Phi = 1e-6$ for the simulation study.

B. Results and Discussion

As the noise present in the true dynamics is Gaussian in nature, the least squares solution in (7) is the best fit to describe the transition dynamics. However, due to the presence of stochasticity in the transition dynamics and the stage cost (40) with the added penalty, the least squares Φ will not deliver the best closed-loop performance. With this experimental setup, we use RL to adjust $\boldsymbol{\Theta}$ for improving closed-loop performance and to illustrate the proposed theoretical developments. We refer to the SPC scheme with least squares Φ and initial $\boldsymbol{\theta}$ as SPC_0 while the learning-based SPC with RL is simply referred to as SPC.

In figure 1, we show the performance of SPC against that of SPC_0 (averaged over 5 seeds). As expected, with RL tuning, SPC improves closed-loop performance over SPC_0. Note that, the observed improvement was despite SPC being defined over a longer history than required which resulted in additional parameters in the learning process.

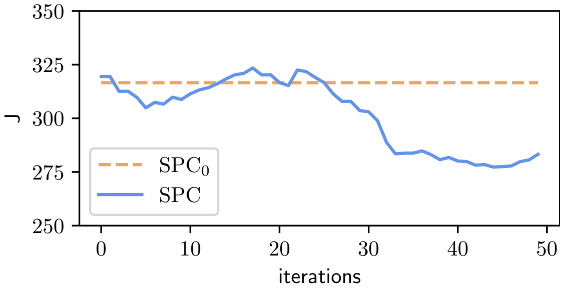
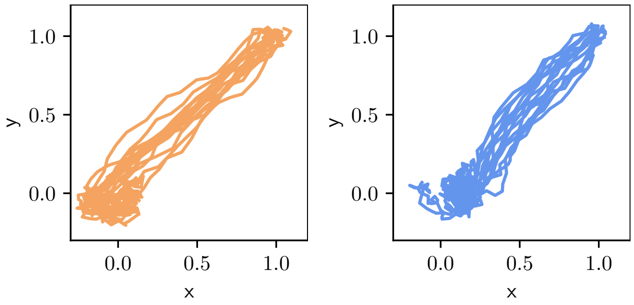


Fig. 1: The performance of SPC scheme adjusted with RL

Figure 2 visualizes the trajectories resulting from SPC as against from SPC, to further illustrate the parameter tuning carried out using RL. The trajectories under the policy SPC$_0$, shown in figure 2a, settle around the origin incurring extra cost due to added penalty. On the contrary, SPC yields trajectories that settle at an offset from the origin, achieving better closed-loop performance. Such a policy behaviour can only be attributed to better-tuned Φ for closed-loop performance as $L_{\boldsymbol{\theta}}$ is purely quadratic centred around the origin.



(a) SPC$_0$

(b) SPC

Fig. 2: Visualisation of trajectories under SPC$_0$ and SPC

VI. Conclusions

In this work, we extend the MPC-based RL framework to work without any explicit state-space models. The proposed predictive control scheme effectively combines RL and pure data-driven predictive control to deliver optimal closed-loop performance. We show that, under some assumptions, the data-driven predictive control scheme with an input-output multi-step predictor can generate optimal policy and value functions for the real system, even if the predictor underlying the predictive control scheme is not accurate. We then present the way to tune the data-driven predictive controller with RL techniques in practice to achieve better closed-loop performances. We illustrate the workings of the proposed predictive control scheme on a linear task.

As data-driven predictive control schemes based on multi-step predictors have a significantly larger parameter space than classic one-step prediction models, we expect them to be more generic and better suited to approximate complex value functions, especially in combination with rich control objective structures such as convex NNs [22]. With such combinations, future work will propose improvements to address applications wherein state-space model definition is difficult, especially for complex transition dynamics.

References

[1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[2] F. Sorourifar, G. Makrygirgos, A. Mesbah, and J. A. Paulson, "A data-driven automatic tuning method for mpc under uncertainty using constrained bayesian optimization," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 243–250, 2021, 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021.

[3] M. Maiworm, D. Limon, and R. Findeisen, "Online learning-based model predictive control with gaussian process models and stability guarantees," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8785–8812, 2021.

[4] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.

[5] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.

[6] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?" in *2022 American Control Conference (ACC)*. IEEE, jun 2022.

[7] S. Gros and M. Zanon, "Data-driven economic nmpc using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2019.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[9] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2021.

[10] A. B. Kordabad, R. Wisniewski, and S. Gros, "Safe reinforcement learning using wasserstein distributionally robust mpc and chance constraint," *IEEE Access*, vol. 10, pp. 130 058–130 067, 2022.

[11] A. Bahari Kordabad, W. Cai, and S. Gros, "Mpc-based reinforcement learning for economic problems with application to battery storage," in *2021 European Control Conference (ECC)*, June 2021, pp. 2573–2578.

[12] A. B. Martinsen, A. M. Lekkas, and S. Gros, "Reinforcement learning-based nmpc for tracking control of asvs: Theory and experiments," *Control Engineering Practice*, vol. 120, p. 105024, 2022.

[13] W. Cai, A. B. Kordabad, H. N. Esfahani, A. M. Lekkas, and S. Gros, "Mpc-based reinforcement learning for a simplified freight mission of autonomous surface vehicles," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2990–2995.

[14] S. Gros and M. Zanon, "Reinforcement learning for mixed-integer problems based on mpc," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5219–5224, 2020, 21st IFAC World Congress.

[15] H. N. Esfahani, A. B. Kordabad, and S. Gros, "Reinforcement learning based on mpc/mhe for unmodeled and partially observable dynamics," in *2021 American Control Conference (ACC)*, May 2021, pp. 2121–2126.

[16] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[17] W. Favoreel, B. D. Moor, and M. Gevers, "SPC: Subspace Predictive Control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, Jul. 1999.

[18] C. Büskens and H. Maurer, "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems," *Online Optimization of Large Scale Systems*, pp. 3–16, 2001.

[19] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[20] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. Pmlr, 2014, pp. 387–395.

[21] W. Cai, A. B. Kordabad, H. N. Esfahani, A. M. Lekkas, and S. Gros, "Mpc-based reinforcement learning for a simplified freight mission of autonomous surface vehicles," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2990–2995.

[22] K. Seel, A. B. Kordabad, S. Gros, and J. T. Gravdahl, "Convex neural network-based cost modifications for learning model predictive control," *IEEE Open Journal of Control Systems*, vol. 1, pp. 366–379, 2022.