Data-driven Economic NMPC using Reinforcement Learning

Sébastien Gros, Mario Zanon

***Abstract*—Reinforcement Learning (RL) is a powerful tool to perform data-driven optimal control without relying on a model of the system. However, RL struggles to provide hard guarantees on the behavior of the resulting control scheme. In contrast, Nonlinear Model Predictive Control (NMPC) and Economic NMPC (ENMPC) are standard tools for the closed-loop optimal control of complex systems with constraints and limitations, and benefit from a rich theory to assess their closed-loop behavior. Unfortunately, the performance of (E)NMPC hinges on the quality of the model underlying the control scheme. In this paper, we show that an (E)NMPC scheme can be tuned to deliver the optimal policy of the real system even when using a wrong model. This result also holds for real systems having stochastic dynamics. This entails that ENMPC can be used as a new type of function approximator within RL. Furthermore, we investigate our results in the context of ENMPC and formally connect them to the concept of dissipativity, which is central for the ENMPC stability. Finally, we detail how these results can be used to deploy classic RL tools for tuning (E)NMPC schemes. We apply these tools on both a classical linear MPC setting and a standard nonlinear example from the ENMPC literature.**

***Index Terms*—Adaptive NMPC, Reinforcement Learning, Economic NMPC, Strict Dissipativity**

I. Introduction

Reinforcement Learning (RL) is a powerful tool for tackling Markov Decision Processes (MDP) without depending on a model of the probability distributions underlying the state transitions. Indeed, most RL methods rely purely on observed state transitions, and realizations of the stage cost in order to increase the performance of the control policy. RL has drawn an increasing attention thanks to its striking accomplishments ranging from computers beating Chess and Go masters [31], to robots learning to walk or fly without supervision [40], [1].

Most RL methods are based on learning the optimal control policy for the real system either directly, or indirectly. Indirect methods rely on learning an approximation of the optimal action-value function underlying the MDP, typically using variants of Temporal-Difference learning [19]. Since the action-value function is in general unknown a priori, a generic function approximator is typically used to approximate it. A common choice in the RL community is to use a Deep Neural Network (DNN).

Direct RL methods seek to learn the optimal policy directly. Most direct RL methods are based on the stochastic or deterministic policy gradient methods, see e.g. [37], [32]. Both rely on carrying an approximation of the action-value function, or at least of the value function underlying the policy. Similarly to indirect methods, also direct RL methods typically use DNNs to approximate the optimal policy and the associated (action-) value function.

Unfortunately, the closed-loop behavior of a system subject to an approximate optimal policy supported by a DNN or a generic function approximation can be difficult to formally analyze. It can therefore be difficult to generate certificates of the behavior of a system controlled by a generic RL algorithm. This issue is especially salient when dealing with safety-critical systems. The development of safe RL methods, which aims at tackling this issue, is an open field of research [18].

Nonlinear Model Predictive Control (NMPC) is a formal control method based on solving at every time instant an optimal control problem to generate the optimal control policy. The optimal control problem seeks to minimize a sum of stage costs over a prediction horizon, subject to state trajectories provided by a model of the real system, the current observed state of the real system, and possibly state and input constraints to be respected. The optimal control problem then delivers an entire input and state sequence spanning the prediction horizon. However, only the first input is applied to the real system. At the next time instant, the entire optimal control problem is solved again using a new estimation of the state of the system.

If the system model underlying the NMPC scheme is perfect, and with the addition of an adequate terminal cost, the NMPC scheme delivers the optimal control policy. Classic NMPC is based on a stage cost lower-bounded by $\mathcal{K}_\infty$ functions, while Economic NMPC (ENMPC) accepts a generic stage cost [26], [12], [4]. A rich and mature theory exists in the literature to analyze the properties of classic NMPC when operating in closed-loop with a real system, establishing desirable key properties such as recursive feasibility and stability [23], [27], [16]. ENMPC has recently attracted the attention of the research community, and a stability theory has been developed fairly recently [26], [12], [4], [6], [24], but is arguably still under development. Since it can tackle the system constraints directly and since it benefits from a rich theory analyzing its behavior, (E)NMPC is arguably an ideal candidate for safety-critical applications.

Unfortunately, the performance of (E)NMPC schemes relies on having a good model of the system to be controlled. A data-driven adaption of the NMPC model to better fit the real system is a fairly obvious approach to tackle this issue. However, since the model does not necessarily match the real system, fitting the model to the data does not necessarily result in the (E)NMPC scheme delivering the optimal policy, and can even be counterproductive. Some attempts have been recently proposed to tackle this problem such as e.g. in [17].

S.Gros is with the Department of Signal & System, Chalmers University of Technology, Hörsalsvägen 9a, Göteborg, Sweden.

Mario Zanon is with the IMT School for Advanced Studies Lucca, Lucca 55100, Italy.

The problem of optimizing a system based on a model having the wrong structure is well known in the field of Real-Time Optimization [14], [13], [2], [3], [22], and has been addressed via the Modifier Approach [28], [29], [38], [15], [22], whereby the cost function of the optimization problem is adapted rather than the model.

In this paper, we propose to use (E)NMPC schemes instead of DNNs to support the parametrization to approximate the (action-)value functions and the policy. Similarly to the idea originated in RTO, we show that the NMPC scheme can deliver the optimal control policy even if the underlying model is incorrect, by adapting the stage cost, terminal cost and constraints only. This observation is applicable to any NMPC scheme such as e.g. classic NMPC, ENMPC, robust and stochastic ENMPC. Furthermore, we establish strong connections between this cost adaptation and the concept of strict dissipativity, which is fundamental to the stability theory of ENMPC.

One practical outcome of the theory proposed in this paper is that all RL techniques can be directly used to tune the NMPC scheme to increase its performance on the real system. Because the theory proposed is very generic, this observation holds e.g. for a stochastic system being controlled by an (E)NMPC scheme based on a deterministic model or a robust NMPC scheme using a scenario tree. Another practical outcome of the theory proposed in this paper is that if a stage cost attached to a given system yields a stabilizing optimal control policy, then an ENMPC scheme with a positive stage cost can in principle be tuned to deliver the optimal policy. A last practical outcome of the proposed theory is that using (E)NMPC as a parametrization for RL instead of DNN allows one to use the rich theory underlying (E)NMPC schemes in the context of RL, and e.g. deliver certificates on the behavior of the policy resulting from the learning process.

The combination of learning and control techniques has been proposed in e.g. [20], [7], [25], [8]. To the best of our knowledge, however, our paper is the first work (a) proposing to use NMPC as a function approximator in RL and (b) investigating the connection between RL and economic MPC.

The paper is structured as follows. Section II establishes the fundamental result of the paper, showing that an (E)NMPC scheme based on the wrong model can, under some conditions, nonetheless deliver the optimal control policy. Section III details how these results can be used in practice. Section IV details how some classic RL techniques can be deployed to adjust the (E)NMPC parameters. Section V further develops the theory and details its connection to the fundamental concept of strict dissipativity underlying the stability theory of ENMPC, and details its consequences for using ENMPC as a parametrization for RL. Section VI deploys the theory on the case of LQR with Gaussian noise, for which all the objects discussed in the theory can be built explicitly, and their practical meaning assessed. Section VII proposes some illustrative examples.

II. Optimal policy based on an inexact model

We will consider that the real system we want to control is described by a discrete Markov-Decision Process (MPD) having the (possibly) stochastic state transition dynamics

$$\mathbb{P}\left[s_+\,|\,s,a\right], \tag{1}$$

where s, a is the current state-input pair and s_+ is the subsequent one. We will label $L(s,a)$ the stage cost associated to the MDP, possibly infinite for some state-input pairs s, a, which we will assume can take the form:

$$L\left(s,a\right) = l\left(s,a\right) + \mathcal{I}_\infty\left(h\left(s,a\right)\right) + \mathcal{I}_\infty\left(g\left(a\right)\right) \tag{2}$$

where we use the indicator function:

$$\mathcal{I}_\infty(x) = \left\{ \begin{array}{cc} \infty & \text{if } x_i > 0 \text{ for some } i \\ 0 & \text{otherwise} \end{array} \right. . \tag{3}$$

In (2), function l captures the cost given to different state-input pairs, while the constraints

$$g(a) \leq 0, \quad h(s,a) \leq 0 \tag{4}$$

capture undesirable state and inputs, and infinite values are given to L when (4) is violated. Note that we have separated pure input constraints and mixed constraints for reasons that will be made cleared later on.

With the addition of a discount factor $0 < \gamma \leq 1$, (1) and (2) yield the optimal policy $\pi_\star(s)$. Note that the notation (1) is standard in the literature on MDPs, while the control literature typically uses the notation $s_+ = f(s,a,\zeta)$, where ζ is a stochastic variable and f a possibly nonlinear function. The action-value function $Q_\star$ and value function $V_\star$ associated with the MDP are defined by the Bellman equations [9]:

$$Q_\star\left(s,a\right) = L\left(s,a\right) + \gamma\mathbb{E}\left[V_\star(s_+)\,|\,s,a\right], \tag{5a}$$

$$V_\star\left(s\right) = Q_\star\left(s,\pi_\star\left(s\right)\right) = \min_a\, Q_\star\left(s,a\right). \tag{5b}$$

Throughout the paper we will assume that the MDP underlying the real system, the associated stage cost L and the discount factor γ yield a well-posed problem, i.e. the value functions defined by (5) are well-posed, and finite over some sets.

We then consider a model of the real system having the state transition dynamics

$$\mathbb{P}\left[\hat{s}_+\,|\,s,a\right], \tag{6}$$

which typically do not match (1) perfectly. Note that (6) trivially includes deterministic models as a special case. Consider a stage cost defined as

$$\hat{L}\left(s,a\right) = \left\{ \begin{array}{cc} Q_\star\left(s,a\right) - \gamma\mathcal{V}^+\left(s,a\right) & \text{if } |\,\mathcal{V}^+\left(s,a\right)| < \infty \\ \infty & \text{otherwise} \end{array} \right. , \tag{7}$$

where $\mathcal{V}^+\left(s,a\right) = \mathbb{E}\left[V_\star\left(\hat{s}_+\right)\,|\,s,a\right]$ and where the expectation is taken over the distribution (6). It is useful here to specify that the conditional formulation of the stage cost $\hat{L}$ proposed in (7) is a technicality dedicated to having a well-defined $\hat{L}$ even when both the action value function and value function take infinite values.

We establish next the central theorem of this paper, stating that under some conditions the optimal policy $\pi_\star$ that minimizes the stage cost L for the true dynamics (1) is also generated by using model (6) combined with the stage cost $\hat{L}$. Hence it is possible to generate the optimal policy based on a wrong model by modifying the stage cost. It may be useful to specify here that our approach further in the paper will be to bypass the possibly difficult evaluation of (7), and replace it by learning $\hat{L}$ directly from the data, see Section IV.

Theorem 1: Consider the optimal value function

$$\hat{V}_N(s) = \min_{\pi} \mathbb{E}\left[\gamma^N V_\star(\hat{s}_N^\pi) + \sum_{k=0}^{N-1} \gamma^k \hat{L}(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\right] \tag{8}$$

associated to the stage cost (7), the state transition model (6), and the terminal cost $V_\star(s)$ over an optimization horizon N. Here we define $\hat{s}^\pi_{0,\dots,N}$ as the (possibly stochastic) trajectories of the state transition model $\mathbb{P}\left[\hat{s}_+ \,|\, s, a\right]$ under a policy π, starting from $\hat{s}_0^\pi = s$. We will label $\hat{\pi}$ the optimal policy associated to $\hat{V}_N(s)$ and $\hat{Q}_N(s,a)$ the associated action-value functions. Consider the set $\mathcal{S}$ such that

$$|\,\mathbb{E}\left[V_\star\left(\hat{s}_k^{\pi_\star}\right)\right]| < \infty, \qquad \forall\, s \in \mathcal{S}, \qquad \forall\, k. \tag{9}$$

Then the following identities hold on $\mathcal{S}$:

(i) $\hat{V}_N(s) = V_\star(s)$

(ii) $\hat{\pi}\left(s\right) = \pi_\star\left(s\right)$

(iii) $\hat{Q}_N\left(s,a\right) = Q_\star\left(s,a\right)$ for the inputs a such that $|\,\mathbb{E}\left[V_\star\left(\hat{s}_+\right) \,|\, s,a\right]| < \infty$.

Proof: Let us consider the N-step value function $\hat{V}_N^\pi$ associated to the stage cost $\hat{L}$, the state transition model $\mathbb{P}\left[\hat{s}_+ \,|\, s, a\right]$ and a policy π, defined as

$$\hat{V}_N^\pi(s) = \mathbb{E}\left[\gamma^N V_\star(\hat{s}_N^\pi) + \sum_{k=0}^{N-1} \gamma^k \hat{L}(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\right]. \tag{10}$$

Assumption (9) ensures that, at least for $\pi = \pi_\star$, all the terms in the sum in (10) have a finite expected value for $s \in \mathcal{S}$, such that $\hat{V}_N^\pi(s)$ is well defined and finite over $\mathcal{S}$ for some policies π. Using a telescopic sum, we can rewrite (10) as:

$$\hat{V}_N^\pi(s) = Q_\star(s, \pi(s)) + \mathbb{E}\left[\sum_{k=1}^{N-1} \gamma^k A_\star(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\right], \tag{11}$$

where the advantage function $A_\star$ is defined as:

$$A_\star(s,a) = \left\{ \begin{array}{cc} Q_\star(s,a) - V_\star(s) & \text{if } |Q_\star(s,a)| < \infty \\ \infty & \text{otherwise} \end{array} \right. . \tag{12}$$

Using the Bellman equalities:

$$\min_\pi \, Q_\star(s, \pi(s)) = Q_\star(s, \pi_\star(s)) = V_\star(s), \tag{13a}$$

$$\min_\pi \, A_\star(s, \pi(s)) = A_\star(s, \pi_\star(s)) = 0, \tag{13b}$$

we observe that all terms in (11) are minimized by the policy $\pi_\star$, such that the following equalities hold on $\mathcal{S}$:

$$\hat{V}_N(s) = \min_\pi \hat{V}_N^\pi(s) = \hat{V}_N^{\pi_\star}(s) = V_\star(s), \tag{14}$$

where the first equality holds by definition, the second holds because $\pi_\star$ is the minimizer of $\hat{V}_N^\pi$, and the last equality holds from (13). It follows that $\hat{V}_N(s) = V_\star(s)$ holds on $\mathcal{S}$ for any N, hence the identities (i) and (ii) hold. We furthermore observe that on $\mathcal{S}$ and for any input a such that $|\mathbb{E}\left[V_\star\left(\hat{s}_+\right) \,|\, s,a\right]| < \infty$, the following equalities hold:

$$\begin{aligned} \hat{Q}_N\left(s,a\right) &= \hat{L}\left(s,a\right) + \gamma \mathbb{E}\left[\hat{V}_{N-1}(\hat{s}_+) \,|\, s, a\right] \\ &= Q_\star\left(s,a\right) + \gamma \mathbb{E}\left[\hat{V}_{N-1}(\hat{s}_+) - V_\star\left(\hat{s}_+\right) \,|\, s, a\right] \\ &= Q_\star\left(s,a\right), \end{aligned} \tag{15}$$

which yields statement (iii). ■

Note that if the transition model is exact, i.e. $\mathbb{P}\left[\hat{s}_+ \,|\, s, a\right] = \mathbb{P}\left[s_+ \,|\, s, a\right]$, then the stage cost defined in (7) satisfies $\hat{L}(s,a) = L(s,a)$, $\forall\, s \in \mathcal{S}$, with $\mathcal{S}$ defined in (9). It is interesting to discuss here the case $N \to \infty$ for which, under some conditions, the terminal cost can be dismissed. We detail this in the following Corollary.

Corollary 1: Under the assumptions of Theorem 1 and under the additional assumption:

$$\lim_{N\to\infty} \, \mathbb{E}\left[\gamma^N V_\star(\hat{s}_N^\pi)\right] = 0, \tag{16}$$

then

$$\hat{V}_\infty(s) = \lim_{N\to\infty} \, \min_\pi \mathbb{E}\left[\sum_{k=0}^{N-1} \gamma^k \hat{L}(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\right] = V_\star(s), \tag{17}$$

and the equalities $\hat{\pi}(s) = \pi_\star(s)$ and $\hat{Q}_\infty(s,a) = Q_\star(s,a)$ hold for $|\,\mathbb{E}\left[V_\star\left(\hat{s}_+\right) \,|\, s,a\right]| < \infty$.

Proof: Let us consider the N-step value function $\hat{V}_N^\pi$ associated to the stage cost $\hat{L}$, the state transition model $\mathbb{P}\left[\hat{s}_+ \,|\, s, a\right]$ and a policy π defined as:

$$\hat{V}_\infty^\pi(s) = \lim_{N\to\infty} \, \mathbb{E}\left[\sum_{k=0}^{N-1} \gamma^k \hat{L}(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\right] \tag{18a}$$

$$\begin{aligned} &= Q_\star(s, \pi(s)) + \lim_{N\to\infty} \, \mathbb{E}\Bigg[-\gamma^N V_\star(\hat{s}_N^\pi) \\ &\qquad + \sum_{k=1}^{N-1} \gamma^k A_\star(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi))\Bigg], \end{aligned} \tag{18b}$$

where (18b) holds as, by assumption, all terms in the summations in (18) have finite expected values. This entails that

$$\hat{V}_\infty^{\pi_\star}(s) = \lim_{N\to\infty} \, V_\star(s) + \mathbb{E}\left[-\gamma^N V_\star(\hat{s}_N^{\pi_\star})\right]. \tag{19}$$

Using (16), we finally observe that the following inequalities hold on $\mathcal{S}$:

$$\hat{V}_\infty(s) = \min_\pi \hat{V}_\infty^\pi(s) = \hat{V}_\infty^{\pi_\star}(s) = V_\star(s), \tag{20}$$

and are justified similarly to (14). ■

Let us make next a few observations regarding Theorem 1.

- Assumption (9) requires that the model trajectories under policy $\pi_\star$ are contained within the set where the value function $V_\star$ is finite with a unitary probability.
- Assumption (16) can be construed as some form of stability condition on the model dynamics under policy $\pi_\star$. This observation is especially clear in the

case $\gamma = 1$, which then imposes the condition $\lim_{N\to\infty} \mathbb{E}\left[V_\star(\hat{s}^\pi_N)\right] = 0$

- Condition (16) is not required in Theorem 1, however, it is highly desirable to fulfil it in practice when a finite-horizon is required, so that the terminal cost in (8) has a limited impact on the optimal policy $\hat{\pi}$.
- Though very similar, Theorem 1 and Corollary 1 cover different cases, since taking the limit for $N \to \infty$ in Theorem 1 does not require Assumption (16) to hold.
- Theorem 1 proposes a modified stage cost and a terminal cost such that the finite-horizon problem (8) delivers the optimal policy $\pi_\star$. It ought to be noted that problem (8) would actually use $\pi_\star(\hat{s}_k)$ to select the inputs at every stage k of the prediction $\hat{s}_{0,\dots,N-1}$. However, identities (i)-(iii) of Theorem 1 do not necessarily require this restriction, i.e. it is sufficient to find a stage cost and a terminal cost such that the resulting optimal control problem yields $\pi_\star$ as its *initial* policy (at stage $k = 0$) only to get identities (i)-(iii).

III. ENMPC as a Function Approximator

We will now detail how the theory presented above applies to using a parametrized ENMPC scheme to approximate the optimal policy and value functions $\pi_\star$ and $V_\star$, $Q_\star$, even if the model underlying the ENMPC scheme is not highly accurate. We will detail in Section IV how the ENMPC parameters can be adjusted to achieve this approximation.

The constraints (4) can be used in the ENMPC scheme to explicitly exclude undesirable states and inputs. Since the ENMPC scheme will be based on an imperfect model f_θ, it will seek the minimization of the modified stage cost $\hat{L}(s,a)$ rather than the original one $L(s,a)$. As a result, while the pure input constraints $g(a) \leq 0$ are arguably fixed, the mixed constraints used in the ENMPC scheme ought to be modified as well, in order to capture the domain where $\hat{L}(s,a)$ is finite. Since $\hat{L}$ is not known a priori, the domain where it is finite depends, among other things, on the discrepancy between f_θ and (1), and will have to be learned. We will therefore consider introducing the parametrized mixed constraints $h_\theta(s,a) \leq 0$ in the ENMPC scheme, where θ will be parameters that can be adjusted via RL tools. Equation (26) and Corollary 2 below provide a more formal explanation of these observations.

Since RL tools cannot handle infinite penalties, we will need to consider a relaxed version of L and of the mixed constraints h_θ. We can now formulate the parametrized ENMPC scheme that will serve as a function approximation in the RL tools.

We will consider a parametrization of the value function $V_\star$ using the following ENMPC scheme parametrized by θ:

$$\begin{aligned} V_\theta(s) = \min_{u,x,\sigma} \quad & \lambda_\theta(x_0) + \gamma^N \left(V^{\mathrm{f}}_\theta(x_N) + w_{\mathrm{f}}^\top \sigma_N\right) \\ & + \sum_{k=0}^{N-1} \gamma^k \left(l_\theta(x_k,u_k) + w^\top \sigma_k\right) && \text{(21a)} \\ \text{s.t.} \quad & x_{k+1} = f_\theta\left(x_k,u_k\right), \quad x_0 = s, && \text{(21b)} \\ & g\left(u_k\right) \leq 0, && \text{(21c)} \\ & h_\theta\left(x_k,u_k\right) \leq \sigma_k, \quad h^{\mathrm{f}}_\theta(x_N) \leq \sigma_N. && \text{(21d)} \end{aligned}$$

Problem (21) is a classic ENMPC formulation when $\gamma = 1$ and $\lambda_\theta = 0$ [27], [16]. Note that we have used x, u in order to clearly distinguish the ENMPC prediction from the actual closed-loop state and control trajectory.

We observe that the ENMPC scheme (21) holds a model parametrization f_θ, a constraint parametrization h_θ as discussed above, and a parametrization of the stage cost l_θ and terminal cost V^{f}_θ. The extra cost λ_θ is discussed in detail in Section (V-A). Reasonable guesses for these functions are to use $l_\theta = l$, $h_\theta = h$, and any classic heuristic to build the terminal cost approximation V^{f}_θ, such as e.g. a quadratic cost stemming from the LQR approximation of the ENMPC scheme. RL tools are then used to modify these initial guesses towards higher closed-loop performances.

The ℓ_1 relaxation of the mixed constraints (21d) relying on slack variables σ_k is fairly standard in practical implementation of (EN)MPC schemes. For w, w_{f} large enough, the solution to (21) is identical to the unrelaxed one whenever a feasible trajectory exists for the initial state s [30]. In this case, we refer to the relaxation as *exact*. The specific role of the relaxation will be detailed in Section IV-C. Function λ_θ in (21a) is not required anywhere in the following developments, but will play a central role in forming nominal stability guarantees of the ENMPC scheme (21), see Section V. We define the policy:

$$\pi_\theta(s) = u^\star_0, \tag{22}$$

where $u^\star_0$ is the first element of the input sequence $u^\star_0, \dots, u^\star_{N-1}$ solution of (21) for a given s. We additionally define the action-value function Q_θ:

$$\begin{aligned} Q_\theta(s,a) = \min_{u,x} \quad & \text{(21a)} && \text{(23a)} \\ \text{s.t.} \quad & \text{(21b)} - \text{(21d)}, && \text{(23b)} \\ & u_0 = a. && \text{(23c)} \end{aligned}$$

Note that the proposed parametrization trivially satisfies the fundamental equalities underlying the Bellman equations, i.e.:

$$\pi_\theta(s) = \arg\min_a Q_\theta(s,a), \quad V_\theta(s) = \min_a Q_\theta(s,a). \tag{24}$$

Let us make some key observations on (21)-(22). The deterministic model (21b) can be construed as a special case of the stochastic state transition (6), using:

$$\mathbb{P}\left[\hat{s}_+ \,|\, s,a\right] = \delta(\hat{s}_+ - f\left(s,a\right)), \tag{25}$$

where δ is the Dirac distribution. Moreover, suppose that we can select θ such that $\lambda_\theta = 0$ and such that the stage cost and constraints in (21) satisfy:

$$\begin{aligned} \hat{L}\left(s,a\right) &= l_\theta\left(s,a\right) + \mathcal{I}_\infty\left(h_\theta\left(s,a\right)\right) + \mathcal{I}_\infty\left(g_\theta\left(a\right)\right), && \text{(26a)} \\ V_\star\left(s\right) &= V^{\mathrm{f}}_\theta(s) + \mathcal{I}_\infty\left(h^{\mathrm{f}}_\theta(s)\right), && \text{(26b)} \end{aligned}$$

for $\hat{L}$ given by (7), where l_θ, $V^{\mathrm{f}}_\theta < \infty$.

Corollary 2 (of Theorem 1): Assume that the NMPC scheme (21) is parametrized using a rich enough parametrization with an exact relaxation (i.e. w, w_{f} large enough). Then, the NMPC scheme (21) delivers the optimal policy $\pi_\star$ and value functions $V_\star$, $Q_\star$ for any state s for which assumption (9) is satisfied.

Proof: By assumption, there exists θ such that (26) holds, then Theorem 1 directly yields the desired result. ■
Note that assumption (9) entails that $\mathcal{S}$ is the forward-invariant set of the dynamics (21b) under the policy $\pi_\star$, associated to the condition $|V_\star(s)| < \infty$.

Unfortunately, a parameter θ satisfying (26) is clearly not guaranteed to exist, and does not exist in most non-trivial practical cases. Indeed, the modified stage cost $\hat{L}$ defined by (7) can be highly intricate, and satisfying (26) can require a very elaborate parametrization of the ENMPC cost and constraints. Even if a θ satisfying (26) does exist, finding it is arguably highly difficult as evaluating (7) can be extremely demanding and requires the knowledge of the real stochastic state transition (6). In this paper, we propose to circumvent this difficulty by (i) relying on a limited parametrization of the ENMPC scheme, at the price of not achieving $\pi_\theta = \pi_\star$ exactly, and (ii) deploying RL techniques in order to adjust the ENMPC parameters θ, so as to avoid computing $\hat{L}$ altogether, see Section IV. In the RL context, we ought to consider the ENMPC scheme (21) as a function approximator for $V_\star$, $Q_\star$ and $\pi_\star$.

The choice of cost parametrization is clearly important, but beyond the scope of this paper. In the example below, see Section VII, we have made fairly trivial choices and obtained interesting results nonetheless. It is clear, however, that a rich parametrization is in theory desirable. It is also desirable that the stage cost and terminal cost approximations, l_θ and V_θ^{f}, always remain (quasi-)convex in order to facilitate the computation of the NMPC solution. In contrast, the parametrization of the initial cost λ_θ can be unrestricted. In the future, we will investigate rich function approximations, including e.g. positive sums of convex functions and sum-of-squares approaches.

A. Robust NMPC Using Scenario Trees

Robust NMPC implemented via scenario tree approaches also readily fits in the framework presented here. Indeed, the scenario tree can be construed as a stochastic process with a discrete probability distribution. More specifically, the stochastic model state transition for a scenario tree dynamics reads as:

$$\mathbb{P}\left[\hat{s}_+ \,|\, s, a\right] = \sum_{i=1}^{N_{\mathrm{s}}} W_i \delta\left(\hat{s}_+ - f_i(s,a)\right), \tag{27}$$

where $f_{1,\ldots,N_{\mathrm{s}}}$ are the N_{s} different dynamics underlying the scenario tree, and $W_{1,\ldots,N_{\mathrm{s}}}$ are the associated probabilities, with $W_i \geq 0$ and $\sum_{i=1}^{N_{\mathrm{s}}} W_i = 1$. All the observations made in Section III are then also valid in this context. One ought then to see the discrete probability distribution w_i as part of the parameters to be adjusted in the (E)NMPC scheme.

B. Model Parametrization

As detailed in Section III the ENMPC scheme (21) can in principle capture the optimal policy $\pi_\star$ without having to adjust the model (21b). This observation, however, does not preclude an adaptation of the model (21b) in order to drive the NMPC policy π_θ towards the optimal one $\pi_\star$, and one can easily argue that allowing such an adaptation introduces additional freedom for the NMPC scheme (21) to better approximate the optimal policy $\pi_\star$. The interplay between the cost and constraints adaptation and the model adaptation is the object of current research.

IV. Reinforcement-Learning for ENMPC

Theorem 1 guarantees that it is in theory possible to generate the optimal policy and value functions using an ENMPC scheme based on a possibly inaccurate model. In practice, one has to rely on ad-hoc parametrization θ of the NMPC scheme, yielding the value function $Q_\theta(s,a)$, $V_\theta(s)$ and the policy $\pi_\theta(s)$. The goal is then to adjust the parameters θ such that the ENMPC scheme policy fits the optimal policy as closely as possible.

Because computing $\hat{L}$ given by (7) is difficult and requires the knowledge of the true dynamics, we will rely on RL techniques to adjust the parameters θ. We will focus here on classical RL approaches. These techniques typically require the sensitivities of the value functions. We briefly detail next how to compute these sensitivities for (21), (22), and (23).

A. Sensitivities of the ENMPC scheme

We detail next how to evaluate the gradients of functions Q_θ, V_θ, π_θ. To that end, let us define the Lagrange function associated to the ENMPC Problem (23) as

$$\begin{aligned}\mathcal{L}_\theta(s,y) = \lambda_\theta(x_0) + \gamma^N V_\theta^{\mathrm{f}}(x_N) + \chi_0^\top\left(x_0 - s\right) + \mu_N^\top h_\theta^{\mathrm{f}}(x_N) \\ + \sum_{k=0}^{N-1} \chi_{k+1}^\top\left(f_\theta\left(x_k,u_k\right) - x_{k+1}\right) + \nu_k^\top g_\theta\left(u_k\right) \\ + \gamma^k L_\theta(x_k,u_k) + \mu_k^\top h_\theta\left(x_k,u_k\right) + \zeta^\top(u_0 - a),\end{aligned}$$

where χ, μ, ν, ζ are the multipliers associated to constraints (21b)-(21d) and (23b)-(23c) respectively, and we will note $y = (x, u, \chi, \mu, \nu, \zeta)$ the primal-dual variables associated to (23).

Note that, for $\zeta = 0$, $\mathcal{L}_\theta(s,y)$ is the Lagrange function associated to the NMPC problem (21). We observe that [10]

$$\nabla_\theta Q_\theta(s,a) = \nabla_\theta \mathcal{L}_\theta(s,y^\star) \tag{29}$$

holds for $y^\star$ given by the primal-dual solution of (23). The gradient (29) is therefore straightforward to build as a by-product of solving the NMPC problem (23). We additionally observe that

$$\nabla_\theta V_\theta(s) = \nabla_\theta \mathcal{L}(s,y^\star), \tag{30}$$

for $y^\star$ given by the primal-dual solution to (21) completed with $\zeta^\star = 0$. Finally, the gradient of the NMPC policy with respect to the parameters θ is given by [10]:

$$\nabla_\theta \pi_\theta(s) = -\nabla_\theta \xi_\theta(s,y^\star) \nabla_y \xi_\theta(s,y^\star)^{-1} \frac{\partial y}{\partial u_0}, \tag{31}$$

for $y^\star$ given by the primal-dual solution to (21) with $\zeta^\star = 0$, and where $\xi_\theta(s,y)$ gathers the primal-dual KKT conditions underlying the NMPC scheme (21).

We remark that (29) and (30) are well-defined for any s, θ such that no inequality constraint in the ENMPC schemes (21) and (23), respectively, is weakly active. When an inequality constraint is weakly active, (29) and (30) may be defined only up to the subgradients generated by the possible active sets. This technical issue is fairly straightforward to circumvent in practice by use of interior-point techniques when solving (21) and (23). Additionally, the policy gradient (31) is only valid if the ENMPC (21) satisfies the linear independence constraint qualification, and the second-order sufficient conditions [10], which are typically satisfied by properly formulated ENMPC schemes.

We detail next how TD-learning is deployed on the NMPC scheme (23), both in an on- and off-policy fashion.

B. Q-learning for (E)NMPC

A classical approach to Q-learning [36] is based on parameter updates driven by the temporal-difference with instantaneous policy updates

$$\tau_k = L_\theta(s_k, a_k) + \gamma\, V_\theta(s_{k+1}) - Q_\theta(s_k, a_k), \tag{32a}$$
$$\theta \leftarrow \theta + \alpha\tau_k \nabla_\theta Q_\theta(s_k, a_k), \tag{32b}$$

where

$$L_\theta(s_k, a_k) = l_\theta(x_k, a_k) + w^\top \max(0, h_\theta\,(x_k, a_k)), \tag{33}$$

and where a_k is selected according to the NMPC policy $\pi_\theta(s)$, with the possible addition of occasional random exploratory moves [36]. The scalar $\alpha > 0$ is a step-size commonly used in stochastic gradient-based approaches. A version of Q-learning with batch policy updates reads as:

$$\tau_k = L_{\tilde\theta}(s_k, a_k) + \gamma\, V_{\tilde\theta}(s_{k+1}) - Q_{\tilde\theta}(s_k, a_k), \tag{34a}$$
$$\tilde\theta \leftarrow \tilde\theta + \alpha\tau_k \nabla_{\tilde\theta} Q_{\tilde\theta}(s_k, a_k), \tag{34b}$$

where a_k is selected according to the fixed NMPC policy $\pi_\theta(s)$ while the learning is performed within an alternative NMPC scheme based on the parameters $\tilde\theta$, and not applied to the real system. Note that the on-policy approach (32) entails that changes in the NMPC parameters θ are readily applied in closed-loop after each update (32b), while the off-policy approach (34) allows one to learn a new set of policy parameters $\tilde\theta$ while deploying the original NMPC scheme, based on the parameters θ, on the real system. The learned parameters $\tilde\theta$ can then be introduced in closed-loop at convenience, by performing the replacement $\theta \leftarrow \tilde\theta$, e.g. after they have converged and after a formal verification of the corresponding NMPC scheme has been carried out, see e.g. [33], [21].

It ought to be made clear here that RL methods of the type (32) or (34) yield no guarantee to find the global optimum of the parameters. This limitation pertains to most applications of RL relying on nonlinear function approximators such as the commonly used DNN. In practice, however, RL improves the closed-loop performance over the one of the initial parameters.

C. Role of the Constraints Relaxation

We can now further discuss the ℓ_1 constraints relaxation (21d) in the light of the TD approaches (32) and (34). In the absence of constraints relaxation, the value functions take infinite values when a constraint violation occurs, and they are therefore meaningless in the context of RL, as only finite value functions can be used in (32)-(34). The proposed constraint relaxation ensures that the value functions retain finite value even in the presence of constraints violation, such that the RL updates (32)-(34) remain well-defined and meaningful. This technical observation has a fairly simple and generic interpretation: any form of learning is meaningless if infinite penalties are assigned to violating limitations.

In practice, violating some safety-critical constraints may be unacceptable. In that context, avoiding the violation of crucial constraints ought to be prevented from the formulation of the (E)NMPC scheme. Here, robust NMPC techniques are arguably an important tool to avoid such difficulties. The applicability of the proposed theory to the robust (E)NMPC formulations of Sec. III-A is therefore of crucial importance for safety-critical applications. The interplay of robust (E)NMPC with RL and the handling of safety-critical constraints will be the object of future publications.

D. Deterministic Policy Gradient Methods for ENMPC

It is useful to underline here that Q-learning techniques seek the fitting of Q_θ to $Q_\star$ under some norm, with the hope that $Q_\theta \approx Q_\star$ will result in $\pi_\theta \approx \pi_\star$. There is, however, no a priori guarantee that the latter approximation holds when the former does. In order to formally maximize the performance of policy π_θ, it is useful to turn to policy gradient methods. For the sake of brevity we propose to focus on deterministic policy gradient methods here [32], based on the policy gradient equation:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta(s) \nabla_a Q_{\pi_\theta}(s, a)\right], \tag{35}$$

where J is the expected closed-loop cost associated to running policy π_θ on the real system, and Q_{π_θ} the corresponding action-value function. Note that the expectation $\mathbb{E}$ is taken over trajectories of the real system subject to policy π_θ. A necessary condition of optimality for policy π_θ is then:

$$\nabla_\theta J(\pi_\theta) = 0. \tag{36}$$

Deterministic policy gradient methods are often built around the TD actor-critic approach, based on [32]:

$$\begin{aligned} &\tau_k = L(s_k, a_k) + \gamma\, Q_w(s_{k+1}, \pi_\theta(s_{k+1})) - Q_w(s_k, a_k) \\ &w \leftarrow w + \alpha_w \tau_k \nabla_w Q_w(s_k, a_k) \\ &\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \pi_\theta(s_k) \nabla_a Q_w(s_k, \pi_\theta(s_k)) \end{aligned} \tag{37}$$

for some $\alpha_w, \alpha_\theta > 0$ small enough, where $Q_w \approx Q_{\pi_\theta}$ is an approximation of the corresponding action-value function. Computationally efficient choices of action-value function parametrization Q_w in the ENMPC context will be the object of future publications.

V. RL and Stable Economic NMPC

The main idea in economic NMPC is to optimize performance (defined by a suitably chosen cost) rather than penalizing deviations from a given reference. Since the cost is generic, the value function is not guaranteed to be positive-definite and proving stability becomes challenging. The main idea for proving stability is to introduce a cost modification, called *rotation*, which does not modify the optimal solution, but yields a positive-definite value function, and hence is a Lyapunov function, such that nominal stability follows.

Section II focused on learning the optimal policy even in case a wrong model $\mathbb{P}\left[\hat{s}_+ \mid s, u\right]$ is used. In this section, we aim at enforcing nominal stability even if the corresponding stage cost $\hat{L}$ is indefinite[1]. To this end, in the NMPC cost we replace $\hat{L}$ by a newly-defined stage cost $\bar{L}$, which we force to be positive-definite. We then use a cost modification in order to recover the correct, possibly indefinite, value and action-value functions corresponding to $\hat{L}$. Throughout the section, we assume that the value and action-value functions are bounded on some set.

In order to show how the cost modification can be implemented by the term λ_θ in (21), we first introduce the proposed cost modification as a generalization of the standard one, prove that it does not modify the optimal policy and discuss how it is used to learn the optimal value and action-value functions. Then, we discuss how the cost modification relates to the cost rotation typically used in ENMPC. Finally, we illustrate how the cost modification can be used in a parametrized ENMPC scheme to be used as function approximator within RL.

A. Generalized Cost Rotation

Consider modifying the cost of the problem according to

$$\bar{L}\left(s,a\right) = \hat{L}\left(s,a\right) + \Lambda\left(s,a\right) - \gamma\mathbb{E}\left[\Lambda\left(\hat{s}_+, \hat{\pi}(\hat{s}_+)\right) \mid s, a\right], \tag{38a}$$

$$\bar{V}_{\mathrm{f}}(s) = V_\star(s) + \Lambda\left(s, \hat{\pi}(s)\right), \tag{38b}$$

where $\bar{V}_{\mathrm{f}}(s)$ denotes the rotated terminal cost and where

$$\Lambda\left(s,a\right) \geq \Lambda(s, \hat{\pi}(s)), \quad \forall\, s, a \tag{38c}$$

holds. Moreover, we require $\Lambda(s,a)$ to be such that $\bar{L}\left(s,a\right)$ is finite whenever $\hat{L}\left(s,a\right)$ is finite.

It is important to underline that defining a function $\Lambda(s,a)$ strictly satisfying condition (38c) is in general hard, as it requires knowledge of $\hat{\pi}$. A simpler choice satisfying (38c) with equality is $\Lambda(s,a) = \lambda(s)$, where λ is any function satisfying the boundedness assumption on $\bar{L}(s,a)$.

Theorem 2: The modification (38) preserves the optimal control policy $\hat{\pi}$ corresponding to $\hat{L}$ and the model $\mathbb{P}\left[\hat{s}_+ \mid s, u\right]$. Moreover, the optimal value and action-value functions satisfy

$$\bar{V}_N(s) = \hat{V}_N(s) + \Lambda(s, \hat{\pi}(s)), \tag{39a}$$

$$\bar{Q}_N(s,a) = \hat{Q}_N(s,a) + \Lambda(s,a). \tag{39b}$$

Proof: We will first prove the theorem for $N=1$ and then proceed by induction. The definition of action value function reads

$$\hat{Q}_1\left(s,a\right) = \hat{L}\left(s,a\right) + \gamma\mathbb{E}\left[V_\star\left(\hat{s}_+\right) \mid s, a\right]. \tag{40}$$

Then, we can write

$$\begin{aligned}
&\hat{Q}_1\left(s,a\right) + \Lambda\left(s,a\right) \\
&\qquad = \hat{L}\left(s,a\right) + \Lambda\left(s,a\right) + \gamma\mathbb{E}\left[V_\star\left(\hat{s}_+\right) \mid s, a\right] \\
&\qquad = \bar{L}\left(s,a\right) + \gamma\mathbb{E}\left[\bar{V}_{\mathrm{f}}\left(\hat{s}_+\right) \mid s, a\right] = \bar{Q}_1\left(s,a\right).
\end{aligned}$$

Since by definition $\hat{Q}_1\left(s,a\right) > \hat{Q}_1\left(s, \hat{\pi}(s_+)\right)$, $\forall\, a \notin \hat{\pi}(s_+)$ and by construction (38c) holds, we obtain that

$$\bar{Q}_1\left(s,a\right) > \bar{Q}_1\left(s, \hat{\pi}(s_+)\right), \quad \forall\, a \notin \hat{\pi}(s_+).$$

Therefore, the optimal policy is preserved and

$$\bar{V}_1(s) = \hat{V}_1(s) + \Lambda(s, \hat{\pi}(s)) = \hat{V}_\star(s) + \Lambda(s, \hat{\pi}(s)).$$

We now use bootstrapping to obtain

$$\begin{aligned}
&\hat{Q}_{k+1}\left(s,a\right) + \Lambda\left(s,a\right) \\
&\qquad = \hat{L}\left(s,a\right) + \Lambda\left(s,a\right) + \gamma\mathbb{E}\left[V_k\left(\hat{s}_+\right) \mid s, a\right] \\
&\qquad = \bar{L}\left(s,a\right) + \gamma\mathbb{E}\left[\hat{V}_k\left(\hat{s}_+\right) \mid s, a\right] = \bar{Q}_{k+1}\left(s,a\right).
\end{aligned}$$

Therefore, cost modification (38) preserves the policy over any horizon N, i.e. $\bar{\pi} = \hat{\pi}$, and (39) hold. ■

We will show next that, for any given policy, the modified NMPC scheme can yield any bounded value function, while this does not hold for the action-value function. Consequently any attempt at learning the optimal policy by solely relying on learning the value function and the proposed modification cannot succeed.

Consider a problem formulated using a stage cost $\check{L}(s,a)$, with the corresponding value functions associated to the optimal policy $\check{\pi}$

$$\check{V}_N(s) = \check{L}\left(s, \check{\pi}(s)\right) + \gamma\mathbb{E}\left[\check{V}_{N-1}\left(\check{s}_+\right) \mid s, \check{\pi}(s)\right], \tag{41a}$$

$$\check{Q}_N(s,a) = \check{L}\left(s,a\right) + \gamma\mathbb{E}\left[\check{V}_{N-1}\left(\check{s}_+\right) \mid s, a\right]. \tag{41b}$$

Lemma 1: Consider the set $\check{\mathcal{S}}$ such that for all $s \in \check{\mathcal{S}}$ both $\check{V}_N(s)$ and $\hat{V}_N(s)$ are bounded. Then, there exists a cost modification $\Lambda(s,a) = \lambda(s)$, $\forall\, a$ such that

$$\check{V}_N(s) + \lambda(s) = \hat{V}_N(s). \tag{42}$$

Proof: The result is obtained by choosing $\lambda(s) = \hat{V}_N(s) - \check{V}_N(s)$. ■

B. Strict Dissipativity

Since most of the literature on dissipativity-based ENMPC focuses on deterministic formulations, in this subsection we also restrict to the deterministic case and we adopt $\gamma = 1$, as is usual in the literature on ENMPC. Strict dissipativity is typically used in ENMPC in order to obtain a positive-definite cost to construct a Lyapunov function and prove closed-loop stability. Strict dissipativity holds if there exists a function λ such that

$$\lambda(s_+) - \lambda(s) \leq -\rho(\|s - s_{\mathrm{e}}\|) + L(s,a) - L(s_{\mathrm{e}}, a_{\mathrm{e}}), \tag{43}$$

with ρ a positive-definite function and

$$(s_{\mathrm{e}}, a_{\mathrm{e}}) = \underset{s,a}{\mathrm{argmin}} \quad L(s,a) \qquad \text{s.t.} \quad s = f(s,a).$$

[1] We remark that in this context, stability is obtained for the model used by NMPC for predictions. Future work will investigate obtaining stability guarantees for the real process $\mathbb{P}\left[s_+ \mid s, u\right]$.

In [4] it has been proven that strict dissipativity is sufficient for closed-loop stability, provided that λ is bounded and the constraint set is compact. In [24] it has been proven that, under a controllability assumption and if $(s_{\mathrm{e}}, a_{\mathrm{e}})$ lies in the interior of the constraint set, strict dissipativity with a bounded function λ is also necessary for closed-loop stability on a compact constraint set.

For simplicity and without loss of generality, we assume that $L(s_{\mathrm{e}}, a_{\mathrm{e}}) = 0$.

Lemma 2: If there exists a function $\Lambda(s, a)$ such that $\min_a \Lambda(s, a) = \Lambda(s, \bar{\pi}(s)) = \lambda(s)$, with $\lambda(s)$ satisfying the strict dissipativity inequality (43), then $\bar{L}(s, a) \geq \rho(\|s - s_{\mathrm{e}}\|) \geq 0$ holds.

Proof: The proof is obtained by noting that

$$\begin{aligned}\bar{L}(s, a) &= L(s, a) + \Lambda(s, a) - \Lambda(\hat{s}_+, \bar{\pi}(s_+)) \\ &\geq L(s, a) + \lambda(s) - \lambda(\hat{s}_+) \geq \rho(\|s - s_{\mathrm{e}}\|),\end{aligned}$$

where the first inequality follows from (38c), and the second inequality is a direct consequence of (43) and $L(s_{\mathrm{e}}, a_{\mathrm{e}}) = 0$. ■

Lemma 2 is of paramount importance in the context of ENMPC because it establishes that any stabilizing optimal policy $\pi_\star$ originating from any given stage cost $L(s, a)$ can be learned by using a parametrization that yields a positive-definite stage cost $\bar{L}(s, a)$. Note that the proposed cost modification is a generalization of the cost rotation used for ENMPC, as we replace $\lambda(s)$ with $\Lambda(s, a)$.

We ought to stress here that though the results presented in this section pertain to the deterministic case, we expect them to extend to the stochastic case as well. Unfortunately a mature dissipativity theory for stochastic ENMPC has not been developed yet. It would arguably not be surprising if a stochastic dissipativity criterion can be put in the form

$$\bar{L}(s, a) \geq \rho(\|s - s_{\mathrm{e}}\|),$$

with $\bar{L}$ defined in (38) in combination with appropriate terminal conditions, hence offering a generalization of [35]. These questions will be the object of future reasearch.

C. Economic Reward and NMPC Parametrization

In the following, we will first show that the proposed cost modification can be reduced to a cost on the initial state, which is introduced in the ENMPC scheme (21) as the term λ_θ. Second, we will explain how we can use a stability-enforcing positive-definite stage cost to approximate the value and action-value function of the economic cost.

For simplicity, we use the cost modification $\mathit{\Lambda}(s, a) = \lambda(s)$. Note that we drop the dependence on the control in $\mathit{\Lambda}$ for practical reasons: (a) keeping it would require knowledge of the optimal policy and (b) $\lambda(s)$ is a valid choice in the sense that it satisfies (38c).

We observe that the cost modification can be summarized by the initial cost $\lambda(x_0)$ by relying on the fact that for all predicted trajectories the modified cost reads as [12], [4]

$$\begin{aligned}&\gamma^N V^{\mathrm{f}}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k) + \gamma^k \lambda(x_k) - \gamma^{k+1}\lambda(x_{k+1}) \\ &= \lambda(x_0) + \gamma^N V^{\mathrm{f}}(x_N) + \sum_{k=0}^{N-1} \gamma^k l(x_k, u_k).\end{aligned}$$

In (21), we parametrize the cost modification as $\lambda_\theta(x_0)$.

We now turn to the question of enforcing stability in the parametrized ENMPC scheme. In the literature on EMPC the cost modification is used to prove stability in case the stage cost is not positive-definite. In our case we are interested in the opposite: we would like to use a positive-definite stage cost to enforce stability while the cost modification is used in order to obtain value and action-value functions corresponding to the economic (non positive-definite) cost.

Thanks to Theorem 1 and Lemma 2, introducing the term λ_θ in (21) guarantees that if the optimal policy $\pi_\star$ is stabilizing for the model dynamics (21b), then the ENMPC scheme (21) can deliver the optimal policy $\pi_\star$ and value functions $V_\star$, $Q_\star$ with a stage cost l_θ lower bounded by a $\mathcal{K}_\infty$ function.

VI. Analytical Case Study: The LQR Case

One of the few cases where (26) can be constructed and satisfied exactly is the LQR case. We use it as a simple illustration of the meaning of $\hat{L}$, Assumption (16) and the cost modification $\mathit{\Lambda}(s, a)$. Consider a centered linear-quadratic-gaussian control problem with the true dynamics and stage cost:

$$s_+ = As + Ba + e, \quad e \sim \mathcal{N}(0, \Sigma), \tag{44}$$

$$L(s, a) = \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} T & N \\ \star & R \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix}. \tag{45}$$

The associated value functions if they exist read as:

$$V_\star = s^\top S s + V_0, \tag{46}$$

$$Q_\star = \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} T + \gamma A^\top S A & N + \gamma A^\top S B \\ \star & R + \gamma B^\top S B \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix} + V_0,$$

where $V_0 = \gamma (1 - \gamma)^{-1} \operatorname{Tr}(S\Sigma)$. Matrix S and the associated optimal policy $K_\star$ are delivered by the Schur complement of the quadratic form in $Q_\star$, i.e. the discounted LQR equations:

$$T + \gamma A^\top S A = S + \left(N + \gamma A^\top S B\right) K_\star, \tag{47a}$$

$$\left(R + \gamma B^\top S B\right) K_\star = N^\top + \gamma B^\top S A. \tag{47b}$$

A. LQR with Imperfect Model and $\hat{L}$

We now consider the deterministic model:

$$\hat{s}_+ = \hat{A}s + \hat{B}a. \tag{48}$$

In order to verify Theorem 1, we consider the stage cost

$$\begin{aligned}\hat{L}(s, a) &= Q_\star(s, a) - \gamma V_\star(\hat{s}_+) \\ &= \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} \hat{T} & \hat{N} \\ \star & \hat{R} \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix} + (1 - \gamma) V_0.\end{aligned}$$

This implies that $\hat{T}$, $\hat{N}$, $\hat{R}$ must satisfiy:

$$\hat{T} + \gamma \hat{A}^\top S \hat{A} = T + \gamma A^\top S A, \tag{49a}$$

$$\hat{N} + \gamma \hat{A}^\top S \hat{B} = N + \gamma A^\top S B, \tag{49b}$$

$$\hat{R} + \gamma \hat{B}^\top S \hat{B} = R + \gamma B^\top S B. \tag{49c}$$

The resulting value function reads as $\hat{V}(s) = s^\top \hat{S} s + \hat{V}_0$, where $\hat{S}$ and the associated policy $\hat{K}$ satisfy:

$$\hat{T} + \gamma \hat{A}^\top \hat{S} \hat{A} = \hat{S} + \left(\hat{N} + \gamma \hat{A}^\top \hat{S} \hat{B}\right) \hat{K}, \tag{50a}$$

$$\left(\hat{R} + \gamma \hat{B}^\top \hat{S} \hat{B}\right) \hat{K} = \hat{N}^\top + \gamma \hat{B}^\top \hat{S} \hat{A}. \tag{50b}$$

Using (49), we obtain that matrices $\hat{S} = S$ and $\hat{K} = K_\star$ satisfy (50).

B. *Assumption* (16) *in the LQR case*

In order for the Discrete Algebraic Riccati Equation (DARE) (50) to deliver a valid LQR solution $\hat{S}$, $\hat{K}$ (in the sense of a Bellman optimality backup), Assumption (16) needs to be satisfied. In the case $\gamma = 1$, this requires that $\hat{A} - \hat{B} K_\star$ has all its eigenvalues inside the unit circle. We illustrate this fact by the following simple example:

$$A = 1, \qquad B = 1, \qquad T = 1, \qquad R = 2, \qquad N = 0,$$

which yields $S = 2$ and $K_\star = 0.5$. By using the model $\hat{A} = 2$, $\hat{B} = 1$, we obtain

$$\hat{T} = -5, \qquad \hat{R} = 2, \qquad \hat{N} = -2,$$

such that $\hat{A} - \hat{B} K_\star = 1.5$ and $K_\star$ corresponds to the non-stabilizing solution $\hat{S} = 2$ of the DARE. Note, however, that the DARE does have a stabilizing solution, which reads

$$\hat{S} = 7, \qquad \hat{K} = 4/3 \qquad \text{such that} \qquad \hat{A} - \hat{B}\hat{K} = 2/3.$$

C. *Economic LQR and Cost Modification* $\Lambda(s,a)$

We now consider an economic LQR, for which $\hat{L}(s,a)$ is indefinite. We introduce matrices δT, δN, δR to define

$$\Lambda(s,a) := \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} \delta T & \delta N \\ \delta N^\top & \delta R \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix},$$

and observe that $\delta N = K_\star^\top \delta R$, $\delta R \succeq 0$ must hold in order for (38c) to be fulfilled, where $K_\star$ is the LQR controller gain associated to the stage cost L for the true system (47b). By defining $\delta S := \delta T - K_\star^\top \delta R K_\star$, we obtain

$$\mathbb{E}\left[\Lambda\left(s_+, \hat{\pi}(\hat{s}_+)\right) | s, a\right] = \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} \hat{A}^\top \delta S \hat{A} & \hat{A}^\top \delta S \hat{B} \\ \hat{B}^\top \delta S \hat{A} & \hat{B}^\top \delta S \hat{B} \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix},$$

such that admissible stage cost modifiers are based on the quadratic forms:

$$\delta W_L = \begin{bmatrix} K_\star^\top \delta R K_\star & K_\star^\top \delta R \\ \delta R K_\star & \delta R \end{bmatrix} - \begin{bmatrix} \gamma \hat{A}^\top \delta S \hat{A} - \delta S & \gamma \hat{A}^\top \delta S \hat{B} \\ \gamma \hat{B}^\top \delta S \hat{A} & \gamma \hat{B}^\top \delta S \hat{B} \end{bmatrix}$$
$$= \delta W_L^0 + \delta W_L^1.$$

It follows that the modified action-value function reads

$$\hat{Q}(s,a) = \begin{bmatrix} s \\ a \end{bmatrix}^\top \left(W + \delta W_L\right) \begin{bmatrix} s \\ a \end{bmatrix}. \tag{51}$$

Since we are interested in stability-enforcing schemes, one needs to choose δR and δS such that $W + \delta W_L \succ 0$ holds. Since this expression is linear in δR and δS, the problem amounts to solving an LMI.

In the following, we state explicitly how the two contributions in δW_L relate to conditions (38c) and (43) respectively. The term δW_L^1 can be framed as a quadratic stage cost rotation and the condition

$$\delta W_L^1 = -\begin{bmatrix} \gamma \hat{A}^\top \delta S \hat{A} - \delta S & \gamma \hat{A}^\top \delta S \hat{B} \\ \gamma \hat{B}^\top \delta S \hat{A} & \gamma \hat{B}^\top \delta S \hat{B} \end{bmatrix} \succ 0,$$

is the strict dissipativity condition (43) for the linear-quadratic case [42]. For $\delta R = 0$, we obtain $\Lambda(s,a) = \lambda(s) = s^\top \delta S s$.

The term δW_L^0 resembles the stage cost used in [11], [41] and satisfies

$$\Lambda^0(s,a) = \begin{bmatrix} s \\ a \end{bmatrix}^\top \delta W_L^0 \begin{bmatrix} s \\ a \end{bmatrix} = (a + K_\star s)^\top \delta R (a + K_\star s),$$

such that $\Lambda^0(s,a) > \Lambda^0(s, -K_\star s)$, $\forall\, a \neq -K_\star s$, $\forall\, s$. As discussed in [11], [41], the use of $\Lambda^0(s,a)$ as stage cost with a zero terminal cost yields a scheme which delivers the optimal feedback for the nominal model. The related value function and action-value function, however, are zero.

VII. Numerical Examples

In this section, we propose two examples in order to illustrate the theoretical developments.

A. *Linear MPC*

We first consider a simple linear MPC example to illustrate the methods above. We consider the MPC scheme:

$$\min_{x,u} \; V_0 + \frac{\gamma^N}{2} x_N^\top S_N x_N + \sum_{k=0}^{N-1} f^\top \begin{bmatrix} x_k \\ u_k \end{bmatrix} \tag{52a}$$

$$\sum_{k=0}^{N-1} \frac{1}{2} \gamma^k \left(\|x_k\|^2 + \frac{1}{2} \|u_k\|^2 + w^\top s_k \right) \tag{52b}$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k + b, \tag{52c}$$

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} + \underline{x} - s_k \leq x_k \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \bar{x} + s_k, \tag{52d}$$

$$-1 \leq u_k \leq 1, \qquad x_0 = s, \tag{52e}$$

where the NMPC parameters subject to the RL scheme are

$$\theta = \left(V_0,\, \underline{x},\, \bar{x},\, b,\, f,\, A,\, B\right), \tag{53}$$

and where the ℓ_1 relaxation uses the weight $w^\top = \begin{bmatrix} 10^2 & 10^2 \end{bmatrix}$. We selected a discount factor $\gamma = 0.9$. The terminal cost matrix S_N is selected as the Riccati matrix underlying the LQR controller locally equivalent to the MPC scheme. The objective of the MPC scheme is to drive the states to the origin, such that the MPC reference is activating the lower bound of the first state. A horizon of $N = 10$ is used. The MPC model is initially chosen as

$$A = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0312 \\ 0.25 \end{bmatrix}, \tag{54}$$

while all other parameters are initialized with zero values.

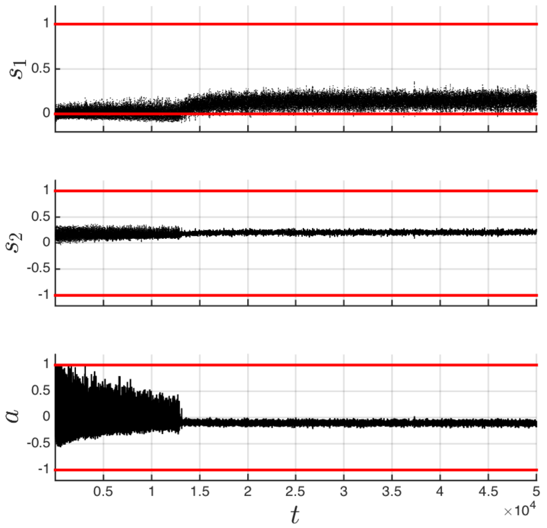

Fig. 1: State and input trajectories for the example detailed in Sec. VII-A.

We will consider that the "real" process is following the dynamics:

$$x_{k+1} = \begin{bmatrix} 0.9 & 0.35 \\ 0 & 1.1 \end{bmatrix} x_k + \begin{bmatrix} 0.0813 \\ 0.2 \end{bmatrix} u_k + \begin{bmatrix} e_k \\ 0 \end{bmatrix} \tag{55}$$

where e_k is a random, uncorrelated, uniformly distributed variable in the interval $[-10^{-1}, 0]$, and therefore drives the first state to violate its lower bound.

We use the on-policy algorithm (32) without introducing exploration. The step size was selected as $\alpha = 10^{-6}$. Figure 1 displays the resulting state and input trajectories. Figure 2 shows the adaptation of the MPC parameters via the on-policy algorithm (32). Figure 3 shows the evolution of the stage cost $L(s, a)$ (including the large ℓ_1 penalties for the constraints violations), and the evolution of the TD error (32a). It can be observed that the RL algorithm manages to reduce the TD error τ to small values, averaging to zero. The state trajectories often violate the state bound $x_1 \geq 0$ in the beginning, resulting in large control actions (see Figure 1), but the RL adjusts the MPC parameters in order to avoid these expensive violations. The adaptation of the parameters is a combination of modifying the stage gradient f and of introducing a model bias b (mostly on the first state, subject to the process noise).

We have also deployed this example without letting the RL scheme adapt the model parameters. The MPC performance is then increased mostly via tightening the bounds, and does not reach the performance displayed in Fig. 2. For the sake of brevity, we do not report these results here.

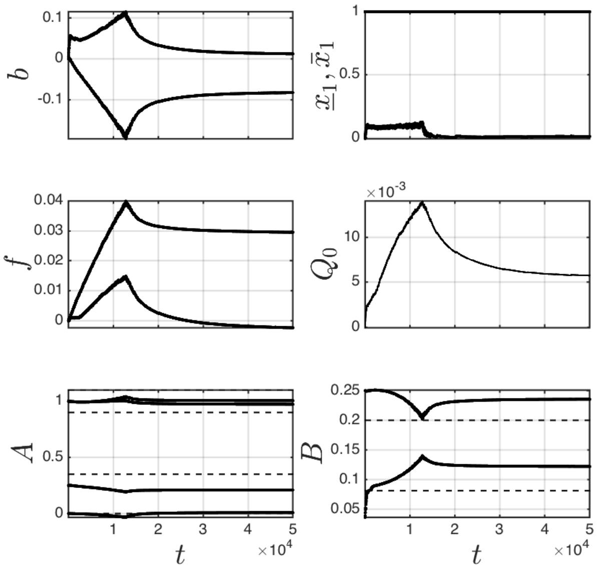

Fig. 2: Trajectory of the parameters θ for the example detailed in Sec. VII-A. The dashed lines report the "real system" A and B entries. One can observe that the RL scheme does not perform system identification, as the MPC model does not converge to the "real system".

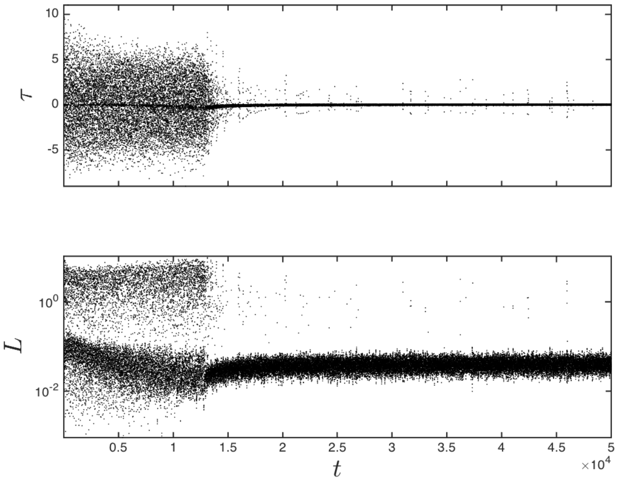

Fig. 3: Evolution of the stage cost (52b) and TD error (32a) achieved for the example detailed in Sec. VII-A.

B. Evaporation Process

We consider an example from the process industry, i.e. the evaporation process modeled in [39], [34] and used in [5], [42] in the context of economic MPC. The model equations include states (X_2, P_2) (concentration and pressure) and controls (P_{100}, F_{200}) (pressure and flow). The model further depends on concentration X_1, flow F_2 and temperatures T_1, T_{200}, which are assumed to be constant in the control model. In reality, these quantities are stochastic with variance $\sigma_{X_1} = 1$, $\sigma_{F_1} = 2$, $\sigma_{T_1} = 8$, $\sigma_{T_{200}} = 5$, and mean centered on the nominal value. Bounds $(25, 40) \leq (X_2, P_2) \leq (100, 80)$ on the states and $u_{\mathrm{l}} = (100, 100) \leq (P_{100}, F_{200}) \leq (400, 400) = u_{\mathrm{u}}$ on the controls are present. In particular, the bound $X_2 \geq 25$ is introduced in order to ensure sufficient quality in the product. All state bounds are relaxed and translated into cost terms as in Section IV-A. The system dynamics are given by [5]

$$M\dot{X}_2 = F_1 X_1 - F_2 X_2, \qquad C\dot{P}_2 = F_4 - F_5, \tag{56}$$

a	b	c	d	e	f	g
0.5616	0.3126	48.43	0.507	55	0.1538	55
h	M	C	U_{A_2}	C_p	λ	λ_s
0.16	20	4	6.84	0.07	38.5	36.6
F_1	X_1	F_3	T_1	T_{200}		
10	5 %	50	40	25		

TABLE I: Model Parameters. The units are omitted and are consistent with the physical quantities they correspond to.

where

$$\begin{aligned} T_2 &= aP_2 + bX_2 + c, & T_3 &= dP_2 + e, \\ \lambda F_4 &= Q_{100} - F_1 C_\mathrm{p}(T_2 - T_1), & T_{100} &= fP_{100} + g, \\ Q_{100} &= U_{A_1}(T_{100} - T_2), & U_{A_1} &= h(F_1 + F_3), \\ Q_{200} &= \frac{U_{A_2}(T_3 - T_{200})}{1 + U_{A_2}/(2C_\mathrm{p}F_{200})}, & F_{100} &= \frac{Q_{100}}{\lambda_\mathrm{s}}, \\ \lambda F_5 &= Q_{200}, & F_2 &= F_1 - F_4, \end{aligned}$$

and the model parameters are given in Table I. The economic objective is given by

$$L(x,u) = 10.09(F_2 + F_3) + 600F_{100} + 0.6F_{200}.$$

We introduce functions $\lambda_\theta, V_\theta^\mathrm{f}, l_\theta$ as fully parametrized quadratic functions defined by Hessian $H_\dagger$, gradient $h_\dagger$ and constant $c_\dagger$, $\dagger = \{\lambda, V^\mathrm{f}, l\}$ to formulate the ENMPC controller

$$\begin{aligned} \min_z \quad & \lambda_\theta(x_0) + \gamma^N \left(V_\theta^\mathrm{f}(x_N) + w^\top \sigma_\mathrm{f}\right) \\ & \qquad + \sum_{k=0}^{N-1} \gamma^k \left(l_\theta(x_k, u_k) + w^\top \sigma_k\right) \\ \text{s.t.} \quad & x_{k+1} = f_\theta\left(x_k, u_k\right), \qquad x_0 = s, \\ & g\left(u_k\right) \leq 0, \qquad h_\theta\left(x_k\right) \leq \sigma_k. \end{aligned}$$

The model is parametrized as the nominal model with the addition of a constant, i.e. $f_\theta(x,u) = f(x,u) + c_f$. The control constraints are fixed and the state constraints are parametrized as simple bounds, i.e. $h(x) = (x - x_\mathrm{l}, x_\mathrm{u} - x)$. The vector of parameter therefore reads as:

$$\theta = (H_\lambda, h_\lambda, c_\lambda, H_{V^\mathrm{f}}, h_{V^\mathrm{f}}, c_{V^\mathrm{f}}, H_l, h_l, c_l, c_f, x_\mathrm{l}, x_\mathrm{u}).$$

Constants $w = 1$ are fixed and assumed to reflect the known cost of violating the state constraints.

We use the batch policy update (32) with $\alpha = 10^{-4}$ and we update the parameters with the learned ones every $N_\mathrm{upd} = 2000$ time steps. In order to induce enough exploration, we use an ϵ-greedy policy which is greedy 90 % of the samples, while in the remaining 10 % we perturb the optimal feedback as

$$a = \mathrm{sat}(u_0^* + e, u_\mathrm{l}, u_\mathrm{u}), \qquad e \sim \mathcal{N}(0,1),$$

where $\mathrm{sat}(\cdot, u_\mathrm{l}, u_\mathrm{u})$ saturates the input between its lower and upper bounds $u_\mathrm{l} = (100, 100), u_\mathrm{u} = (400, 400)$ respectively. We initialize the ENMPC scheme by tuning the cost using the economic-based approach proposed in [42], which is based on the nominal model. For the model we use $c_f = 0$ and the bounds are initialized at their nominal values. While at every step we do check that $H_l, H_{V^\mathrm{f}} \succ 0$, during the learning phase the parameters never violate this constraint. As displayed in Figure 4, the algorithm converges to a constant parameter value while reducing the average TD-error.

We ran a closed-loop simulation to compare the performance of RL-tuned NMPC with the one using the economic-based tuning proposed in [42]. We display in Figure 5 the difference in cost and concentration X_2 between the two ENMPC schemes. It can be seen that RL keeps X_2 at higher values in order to reduce the violation of the quality constraint. This entails an improvement in the cost which is about 7% in the considered scenario.

It is important to stress here that both the cost and the model parametrization do not have the same structure as the real cost and model. Therefore, the action-value function and, consequently, the policy can be learned only approximately. Another important remark concerns the role of the storage function λ_θ. RL exploits this function to keep the stage cost positive-definite. If we invert the cost modification (38) in order to recover an approximation of the economic stage cost, we obtain an indefinite one.

We ran an additional simulation in which the model used in simulations is deterministic and coincides with the one used for predictions in NMPC. We initialize the learning phase with the nominally tuned parameters from [42]. In this case, RL keeps the parameters essentially unaltered, which suggests that the nominal tuning was already optimal.

Consider the naive initial guess $H_l = I$, $x_\mathrm{l} = (25, 40)$, $x_\mathrm{u} = (100, 80)$, while all other parameters are 0. In order to obtain parameters which are close to convergence in each batch, we set $N_\mathrm{upd} = 20000$. In this case, with 10^5 samples we did not yet obtain convergence. Even though convergence of the RL scheme is much slower, the parameters are approaching a steady value and the closed-loop performance is improved with respect to the initial guess.

While our simulation results are promising, we ought to specify here that we have observed some potential difficulties related - to the best of our knowledge - to (i) using a gradient method in adjusting the ENMPC parameters and (ii) using a Q-learning method as a proxy for learning the optimal policy. These observations arguably call for exploring the effectiveness of deploying 2$^\mathrm{nd}$-order methods, such as e.g. LSTD-type methods [36], and policy gradient approaches [32] to alleviate the difficulties observed in some cases.

VIII. Conclusions

In this paper, we propose to use Economic NMPC schemes to support the parametrization of the value functions and/or the policy which is an essential component of Reinforcement Learning. We show that the Economic NMPC schemes can generate the optimal policy for the real system even if the underlying model is wrong, by adjusting the stage and terminal cost alone. We also show how a positive stage and terminal cost can be used in the ENMPC scheme to learn the optimal control policy, resulting in an ENMPC scheme that is stable by construction. The resulting ENMPC delivers the optimal control policy for the real system if that policy is itself stabilizing. We additionally detail how some classic RL methods

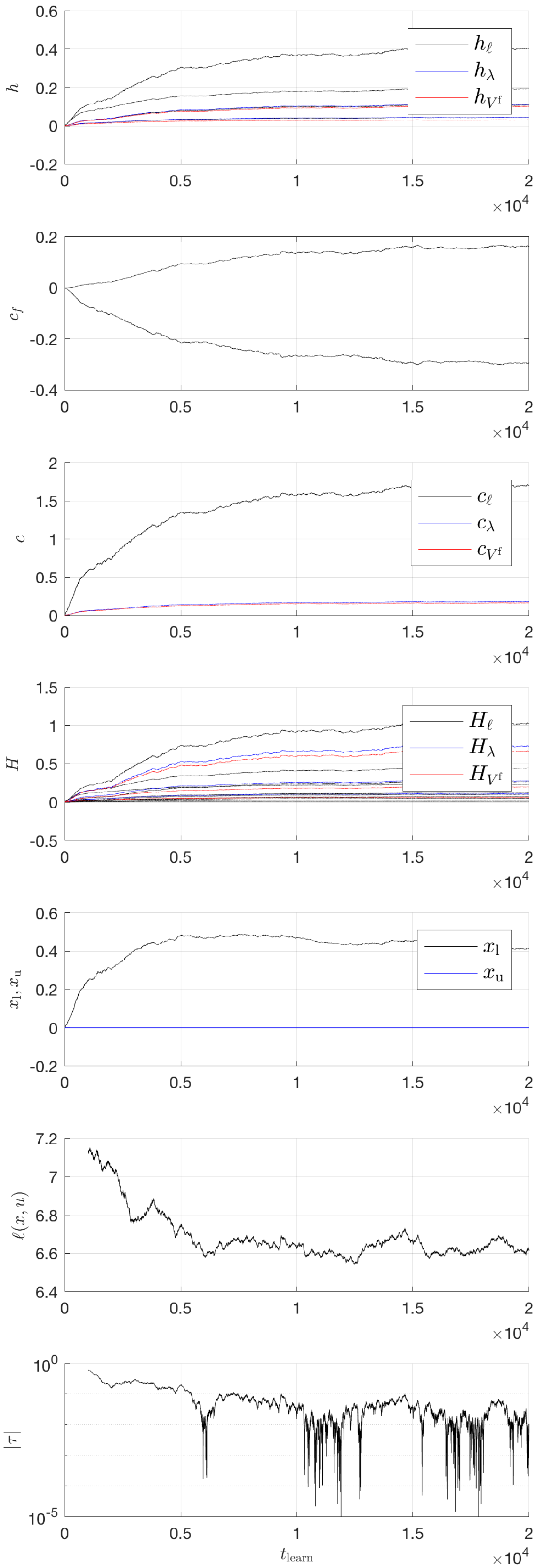

Fig. 4: Evolution of the parameters (increment w.r.t. the initial guess value) and of the TD-error (averaged over the preceding 1000 samples) during learning.

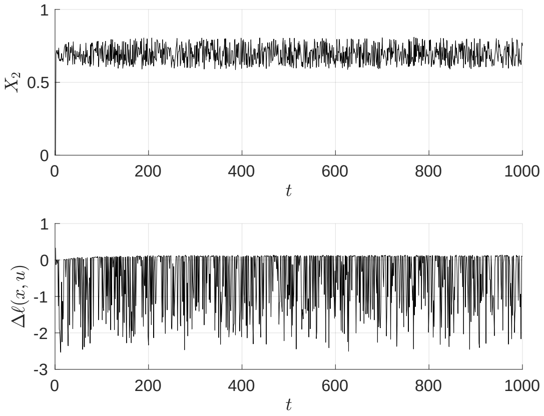

Fig. 5: Closed-loop simulations: difference between the RL-tuned NMPC, and the nominal economic tuning [42]. Top plot: difference in concentration X_2. Bottom plot: difference in economic cost.

can be deployed in practice to adjust the parameters of the ENMPC scheme. The methods are illustrated in simulations.

Future work will propose improvements in the Reinforcement Learning algorithms specific for ENMPC, and propose an efficient combination of the existing classic model-tuning techniques and the Reinforcement-Learning-based tuning.

References and Notes

[1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, page 2007. MIT Press, 2007.

[2] M. Agarwal. Feasibility of on-line reoptimization in batch processes. *Chemical Engineering Communications*, 158(1):19–29, 1997.

[3] M. Agarwal. Iterative set-point optimization of batch chromatography. *Comp. Chem. Eng.*, 29(6):1401–1409, 2005.

[4] R. Amrit, J. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control*, 35:178–186, 2011.

[5] Rishi Amrit, James B. Rawlings, and Lorenz T. Biegler. Optimizing process economics online using model predictive control. *Computers & Chemical Engineering*, 58:334 – 343, 2013.

[6] D. Angeli, R. Amrit, and J. Rawlings. On Average Performance and Stability of Economic Model Predictive Control. *IEEE Transactions on Automatic Control*, 57:1615 – 1626, 2012.

[7] Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216 – 1226, 2013.

[8] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe Model-based Reinforcement Learning with Stability Guarantees. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 908–918. Curran Associates, Inc., 2017.

[9] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3rd edition, 2005.

[10] C. Büskens and H. Maurer. *Online Optimization of Large Scale Systems*, chapter Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems, pages 3–16. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[11] L. Chisci, J.A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37:1019–1028, 2001.

[12] M. Diehl, R. Amrit, and J.B. Rawlings. A Lyapunov Function for Economic Optimizing Model Predictive Control. *IEEE Trans. of Automatic Control*, 56(3):703–707, March 2011.

[13] J.F. Forbes and T.E. Marlin. Design cost: a systematic approach to technology selection for model-based real-time optimization systems. volume 20(67), pages 717–734, 1996.

[14] J.F.. Forbes, T.E. Marlin, and J.F. MacGgregor. Model adequacy requirements for optimizing plant operations. *Comp. Chem. Eng.*, 18(6):497–510, 1994.

[15] W. Gao and S. Engell. Comparison of iterative set-point optimisation strategies under structural plant-model mismatch. volume 16, pages 401–401, 2005.

[16] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer, London, 2011.

[17] Lukas Hewing, Alexander Liniger, and Melanie N. Zeilinger. Cautious NMPC with gaussian process dynamics for miniature race cars. *CoRR*, abs/1711.06586, 2017.

[18] J. Fernandez J. Garcia. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2013.

[19] Jan Peters Jens Kober, J. Andrew Bagnell. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32, 2013.

[20] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based Model Predictive Control for Safe Exploration and Reinforcement Learning. Published on Arxiv, 2018.

[21] J. Löfberg. Oops! I cannot do it again: Testing for recursive feasibility in MPC. *Automatica*, 48(3):550–555, 2012.

[22] A. Marchetti, B. Chachuat, and D. Bonvin. Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res*, 48(13):6022–6033, 2009.

[23] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 26(6):789–814, 2000.

[24] M. A. Müller, D. Angeli, and F. Allgöwer. On necessity and robustness of dissipativity in economic model predictive control. *IEEE Transactions on Automatic Control*, 60(6):1671–1676, 2015.

[25] Chris J. Ostafew, Angela P. Schoellig, and Timothy D. Barfoot. Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking. *The International Journal of Robotics Research*, 35(13):1547–1563, 2016.

[26] James B. Rawlings and Rishi Amrit. Optimizing Process Economic Performance using Model Predictive Control. In *Proceedings of NMPC 08 Pavia*, pages 119–138. 2009.

[27] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.

[28] P.D. Roberts. An algorithm for steady-state system optimization and parameter estimation. *Int. J. Systems Sci.*, 10(7):719–734, 1979.

[29] P.D. Roberts. Coping with model-reality differences in industrial process optimi- sation, a review of integrated system optimisation and parameter estimation (isope). *Computers in Industry*, 26(3):281–290, 1995.

[30] P.O.M. Scokaert and J.B. Rawlings. Feasibility Issues in Linear Model Predictive Control. *AIChE Journal*, 45(8):1649–1659, 1999.

[31] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[32] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–387–I–395, 2014.

[33] D. Simon and J. Löfberg. Stability analysis of model predictive controllers using mixed integer linear programming. pages 7270–7275, 2016.

[34] C. Sonntag, O. Stursberg, and S. Engell. Dynamic Optimization of an Industrial Evaporator using Graph Search with Embedded Nonlinear Programming. In *Proc. 2nd IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS)*, pages 211–216, 2006.

[35] Pantelis Sopasakis, Domagoj Herceg, Panagiotis Patrinos, and Alberto Bemporad. Stochastic economic model predictive control for markovian switching systems. *IFAC-PapersOnLine*, 50(1):524 – 530, 2017. 20th IFAC World Congress.

[36] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[37] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pages 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

[38] P. Tatjewski. Iterative optimizing set-point control-the basic principle redesigned. pages 992–992, 2002.

[39] F. Y. Wang and I. T. Cameron. Control studies on a model evaporation process constrained state driving with conventional and higher relative degree systems. *Journal of Process Control*, 4:59–75, 1994.

[40] Shouyi Wang, Wanpracha Chaovalitwongse, and Robert Babuska. Machine learning algorithms in bipedal robot control. *Trans. Sys. Man Cyber Part C*, 42(5):728–743, September 2012.

[41] M. Zanon, T. Charalambous, H. Wymeersch, and P. Falcone. Optimal scheduling of downlink communication for a multi-agent system with a central observation post. *IEEE Control Systems Letters*, 2(1):37–42, Jan 2018.

[42] M. Zanon, S. Gros, and M. Diehl. A Tracking MPC Formulation that is Locally Equivalent to Economic MPC. *Journal of Process Control*, 2016.

Sébastien Gros received his Ph.D degree from EPFL, Switzerland, in 2007. After a journey by bicycle from Switzerland to the Everest base camp in full autonomy, he joined a R&D group hosted at Strathclyde University focusing on wind turbine control. In 2011, he joined the university of KU Leuven, where his main research focus was on optimal control and fast NMPC for complex mechanical systems. He joined the Department of Signals and Systems at Chalmers University of Technology, Göteborg in 2013, where he became associate Prof. in 2017. He is now full Prof. at NTNU, Norway and affiliate Prof. at Chalmers. His main research interests includes numerical methods, real-time optimal control, reinforcement learning, and the optimal control of energy-related applications.

Mario Zanon received the Master's degree in Mechatronics from the University of Trento, and the Diplôme d'Ingénieur from the Ecole Centrale Paris, in 2010. After research stays at the KU Leuven, University of Bayreuth, Chalmers University, and the University of Freiburg he received the Ph.D. degree in Electrical Engineering from the KU Leuven in November 2015. He held a Post-Doc researcher position at Chalmers University until the end of 2017 and is now Assistant Professor at the IMT School for Advanced Studies Lucca. His research interests include numerical methods for optimization, economic MPC, reinforcement learning, and the optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.