Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 5213–5218

Reinforcement Learning Based on Real-Time Iteration NMPC

Mario Zanon * **Vyacheslav Kungurtsev** ** **Sébastien Gros** ***

* *IMT School for Advanced Studies Lucca, Italy*
** *Czech Technical University in Prague, Czech Republic*
*** *Norwegian University of Technology, NTNU*

Abstract: Reinforcement Learning (RL) has proven a stunning ability to learn optimal policies from data without any prior knowledge on the process. The main drawback of RL is that it is typically very difficult to guarantee stability and safety. On the other hand, Nonlinear Model Predictive Control (NMPC) is an advanced model-based control technique which does guarantee safety and stability, but only yields optimality for the nominal model. Therefore, it has been recently proposed to use NMPC as a function approximator within RL. While the ability of this approach to yield good performance has been demonstrated, the main drawback hindering its applicability is related to the computational burden of NMPC, which has to be solved to full convergence. In practice, however, computationally efficient algorithms such as the Real-Time Iteration (RTI) scheme are deployed in order to return an approximate NMPC solution in very short time. In this paper we bridge this gap by extending the existing theoretical framework to also cover RL based on RTI NMPC. We demonstrate the effectiveness of this new RL approach with a nontrivial example modeling a challenging nonlinear system subject to stochastic perturbations with the objective of optimizing an economic cost.

Keywords: Reinforcement Learning, Model Predictive Control

1. INTRODUCTION

Reinforcement Learning (RL) is a powerful data-driven technique which aims at optimizing the performance of a Markov Decision Process (MDP), typically without relying on a model of the state transition probability. Most RL methods rely purely on the observed stage cost and state transition in order to estimate and optimize the closed-loop performance. The potential of RL has been recently demonstrated by several successful implementations including, e.g., robots learning to walk or fly (Wang et al., 2012; Abbeel et al., 2007), or computers beating chess and go champions (Silver et al., 2016).

Some RL methods optimize the performance indirectly, by parametrizing the action-value function and learning a good approximation of the optimal action-value function underlying the MDP. The optimal policy is then obtained by optimizing the action-value function with respect to the action. Other RL methods, instead, directly parametrize the policy and estimate its gradient in order to optimize the policy parameter. Both direct and indirect approaches typically rely on derivative-based stochastic optimization, such that the function approximator must be differentiable in the parameter.

Deep Neural Networks (DNN) are a common choice to support the parametrization of either the action-value function or the policy. While DNN can be very effective in practice, they pose difficulties in the analysis of closed-loop stability or in imposing hard constraints on the evolution of the state of the real system. An alternative to DNNs has been proposed in (Gros and Zanon, 2020), where it has been suggested to use nonlinear Model Predictive Control (MPC) as a function approximator. Strategies to provide strict constraint satisfaction guarantees have been investigated in (Gros et al., 2020a) by using projection approaches and in (Zanon and Gros, 2019; Gros and Zanon, 2020b) by using robust MPC schemes. The use of mixed-integer MPC formulations has been investigated in (Gros and Zanon, 2020a). Furthermore, the use of MPC as a function approximator gives one the possibility to easily introduce any available information on the system dynamics. Which MPC parameter to adapt using RL is a design decision and can include, e.g., the cost Hessian and gradient, some model or constraint parameter, etc.

One of the main drawbacks of MPC, however, is the need to solve an optimal control problem in real time. While this issue is less significant for linear systems, for nonlinear MPC (NMPC) tailored algorithms are required in order to limit the computational burden and guarantee real-time feasibility for many applications of interest. In particular, the real-time iteration (RTI) scheme (Diehl et al., 2005) obtains the complexity reduction by solving an individual quadratic program (QP), i.e., one step of sequential quadratic programming (SQP). By exploiting the similarity between two consecutive MPC problems, the solution of the first can be used to construct a good initial guess for the second. As a result, each OCP solve is reduced to one (or a small number) of QP solves, which can be performed quickly using contemporary solvers.

The use of RTI-based NMPC as a function approximator for RL is a natural means of making MPC-based RL applicable in real time to a wide class of systems. In order to be

10.1016/j.ifacol.2020.12.1195

able to deploy standard RL techniques, one must be able to compute the sensitivities of RTI-based NMPC with respect to the RL parameter. The necessary sensitivity analysis for fully converged NMPC has been proposed in (Gros and Zanon, 2020), by relying on results on parametric optimization. However, the analysis of (Gros and Zanon, 2020) does not apply to the RTI scheme, since it assumes that the NMPC problem is solved to full convergence. In this paper, we set the theoretical foundations which allow one to deploy the RTI scheme within RL and demonstrate the effectiveness of our approach in simulations by minimizing the economic cost of a nonlinear system subject to stochastic perturbations.

The paper is structured as follows. In Section 2, we provide some background on Q-learning and OCPs as function approximation. In Section 3 we present nonlinear MPC and the RTI scheme; we provide the sensitivity analysis for the RTI scheme in Section 4; and in Section 5 we discuss the adaptation of classic RL methods when deploying them in combination with NMPC. We report on some numerical simulations in Section 6. In Section 7 we conclude the paper and outline future research directions.

2. REINFORCEMENT LEARNING BACKGROUND

Consider a system whose dynamics are described by a Markov Process (MP) with with continuous state $\mathbf{s}$ and action (or control) $\mathbf{a}$, and state transition $\mathbf{s}, \mathbf{a} \rightarrow \mathbf{s}_+$ described by probability density

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

Note that often system (1) is written as $\mathbf{s}_+ = \mathbf{f}(\mathbf{s}, \mathbf{a}, \mathbf{w})$, with $\mathbf{w}$ stochastic process noise described by probability density $\mathbb{P}_{\mathbf{w}}$. We Associate with system (1) the stage cost $\ell(\mathbf{s}, \mathbf{a})$ and the discount factor $\gamma \in [0, 1]$ to define a Markov Decision Process (MDP).

Consider a deterministic policy delivering the control input $\mathbf{a} = \boldsymbol{\pi}(\mathbf{s})$, resulting in state distribution $\tau^{\boldsymbol{\pi}}$. Reinforcement Learning aims at finding the best policy $\boldsymbol{\pi}_\star$, i.e., at solving

$$\boldsymbol{\pi}_\star := \arg\min_{\boldsymbol{\pi}} \; J(\boldsymbol{\pi}) := \mathbb{E}_{\tau^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{\infty} \gamma^k \ell\left(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right)\right]. \tag{2}$$

Important quantities in RL are the action-value function $Q^\star(s, a)$ and value function $V^\star(s)$ associated with the optimal policy $\pi^\star(s)$, defined by the Bellman equations:

$$Q^\star(s, a) = \ell(s, a) + \gamma \mathbb{E}\left[V^\star(s_+) \,|\, s, a\right], \tag{3a}$$

$$V^\star(s) = Q^\star\left(s, \pi_\star(s)\right) = \min_a Q^\star(s, a). \tag{3b}$$

Several RL algorithms have been proposed in the literature and two popular approaches are Q-learning and policy gradient methods (Sutton and Barto, 2018). We provide next a brief description of these two approaches.

2.1 Q-learning

Q-learning parametrizes the action value function as $Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$, where $\boldsymbol{\theta}$ is a vector of parameters whose values have to be learned, and aims at minimizing $\|Q^\star(\mathbf{s}, \mathbf{a}) - Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})\|_2^2$. Standard algorithms rely on the recursive parameter update (Sutton and Barto, 2018)

$$\delta_k = \ell(\mathbf{s}_k, \mathbf{a}_k) + \gamma \min_{\mathbf{a}_{k+1}} Q_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k), \tag{4a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k). \tag{4b}$$

Q-learning has been successfully applied in, e.g., (Watkins, 1989; Mnih et al., 2015; Theocharous et al., 2015).

2.2 Actor-Critic Policy Gradient

Policy gradient approaches parametrize the policy as $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ and directly aim at approximately solving (2) as

$$\boldsymbol{\theta}_\star = \arg\min_{\boldsymbol{\theta}} \; J(\boldsymbol{\pi}_{\boldsymbol{\theta}}). \tag{5}$$

This typically involves some form of stochastic descent algorithm where, in a deterministic policy gradient framework based on actor-critic methods, we have

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right], \tag{6a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{6b}$$

with advantage function $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) = Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) - V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s})$.

2.3 Function Approximation based on MPC

The use of NMPC to parametrize the action-value function has been first investigated in (Gros and Zanon, 2020), where it has been proven that NMPC is a universal function approximator in the sense that it can be used to support the action value function $Q_{\boldsymbol{\theta}}$, value function $V_{\boldsymbol{\theta}}$ and corresponding policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ at once. We remark that NMPC is a nonlinear function approximator which typically requires to be formulated using a positive-definite cost. In general, however, the cost to be optimized by RL can be indefinite.

As proven in (Gros and Zanon, 2020) this does not limit the applicability of NMPC in the RL context, provided that an initial cost term is added to the problem formulation to perform a so-called *cost rotation*. This rotation makes it possible to approximate an indefinite Q while using a positive-definite cost in NMPC.

We ought to stress that the RL-NMPC framework proposed in (Gros and Zanon, 2020) and further investigated in (Zanon et al., 2019; Zanon and Gros, 2019; Gros and Zanon, 2020b,a) does not provide strict stability guarantees. Such guarantees would be given in case the discount factor were $\gamma = 1$ and suitable terminal conditions were formulated (Rawlings and Mayne, 2009; Grüne and Pannek, 2011). Clearly, one can formulate the NMPC problem by using $\gamma = 1$ even when the MDP has $\gamma < 1$. However, the ability of NMPC to approximate $Q_\star, V_\star, \boldsymbol{\pi}_\star$ has not been thoroughly investigated. This is the subject of ongoing research, but beyond the scope of this paper.

3. NMPC

In this paper, we consider function approximators based on OCP parametrized by $\boldsymbol{\theta}$ of the form

$$Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \min_{\mathbf{z}} \; \lambda_{\boldsymbol{\theta}}(\mathbf{s}) + \gamma^N V^{\mathrm{f}}_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} \gamma^k \ell_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k) \tag{7a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{s}, \quad \mathbf{u}_0 = \mathbf{a}, \tag{7b}$$

$$\mathbf{x}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right), \tag{7c}$$

$$\mathbf{g}\left(\mathbf{u}_k\right) \leq 0, \tag{7d}$$

$$\mathbf{h}_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \leq 0, \quad \mathbf{h}^{\mathrm{f}}_{\boldsymbol{\theta}}(\mathbf{x}_N) \leq 0, \tag{7e}$$

where $\mathbf{z} = (\mathbf{x}_0, \mathbf{u}_0, \ldots, \mathbf{x}_N)$. The stage and terminal cost $\ell_{\boldsymbol{\theta}}, V^{\mathrm{f}}_{\boldsymbol{\theta}}$, the system dynamics and constraints $\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{h}_{\boldsymbol{\theta}}, \mathbf{h}^{\mathrm{f}}_{\boldsymbol{\theta}}$ and

the initial cost $\lambda_{\boldsymbol{\theta}}$ are parametric functions of $\boldsymbol{\theta}$. Note that in MPC the initial constraint (7b) typically only involves the state, i.e., $\mathbf{u}_0 = \mathbf{a}$ is not present, since the goal is to compute an optimal policy. The policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$ and value function $V_{\boldsymbol{\theta}}(\mathbf{s})$ are obtained by solving Problem (7) with constraint $\mathbf{u}_0 = \mathbf{a}$ removed. This is fully equivalent to

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}), \quad V_{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}). \tag{8}$$

In standard NMPC, parameter $\boldsymbol{\theta}$ is typically considered as follows. The system dynamics and constraints $\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{h}_{\boldsymbol{\theta}}, \mathbf{h}^{\mathrm{f}}_{\boldsymbol{\theta}}$, are derived as mathematical models of the physical process that needs to be controlled. System identification techniques are deployed to compute parameter $\boldsymbol{\theta}$ such that the model predictions fit experimental data sufficiently accurately. Concerning the cost, instead, parameter $\boldsymbol{\theta}$ is a tuning parameter which should be chosen by the control engineer to ensure that the closed-loop system performance is satisfactory, e.g., in terms of disturbance rejection and asymptotic stability. Alternatively, if—similarly to RL—a clear objective is available, so-called *economic MPC* schemes directly optimize the prescribed objective (Rawlings and Mayne, 2009; Grüne and Pannek, 2011).

In order to address feasibility issues in Problem (7) we propose to adopt an exact relaxation of state-dependent constraints, as proposed in (Scokaert and Rawlings, 1999) and used in the context of RL in (Gros and Zanon, 2020; Zanon et al., 2019; Zanon and Gros, 2019).

3.1 Real-Time NMPC

Problem (7) is a parametric nonlinear programming problem, the solution of which can be computationally demanding. Several approaches have been proposed in order to solve Problem (7) approximately but quickly, e.g., the Real-Time Iteration (RTI) scheme (Diehl et al., 2002), the Advanced Step NMPC Controller (Zavala and Biegler, 2009) and the continuation/GMRES approach (Ohtsuka, 2004). All these approaches are based on pathfollowing techniques for parametric NLPs, where the parameter of interest is the initial state $\mathbf{s}$, and essentially rely on the (approximate) solution at the previous time instant to build a good initial guess for the problem at the current time in a predictor-corrector framework. For the sake of simplicity, we focus on the RTI scheme, which is based on sequential quadratic programming and operates as follows.

We write Problem (7) in the compact form

$$\min_{\mathbf{z}} \; F_{\boldsymbol{\theta}}(\mathbf{z}) \qquad \text{s.t.} \;\; \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}) = 0, \qquad \mathbf{H}_{\boldsymbol{\theta}}(\mathbf{z}) \leq 0, \tag{9}$$

with Lagrange multipliers $\boldsymbol{\xi}, \boldsymbol{\upsilon}$ associated with $\mathbf{G}, \mathbf{H}$ respectively. Given an initial guess $\mathbf{y}^{(0)} = (\mathbf{z}^{(0)}, \boldsymbol{\xi}^{(0)}, \boldsymbol{\upsilon}^{(0)})$, Sequential Quadratic Programming (SQP) solves (9) by computing a suitable step size t and updating

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + t\mathbf{y}^{\mathrm{QP}},$$

where $\mathbf{y}^{\mathrm{QP}}$ is the primal-dual solution of the quadratic program (QP)

$$\min_{\mathbf{z}} \;\; \frac{1}{2}\mathbf{z}^\top L^{(i)}\mathbf{z} + \nabla_{\mathbf{z}}\mathcal{L}_{\boldsymbol{\theta}}\left(\mathbf{y}^{(i)}\right)^\top \mathbf{z} \tag{10a}$$

$$\text{s.t.} \;\; \nabla_{\mathbf{z}}\mathbf{G}_{\boldsymbol{\theta}}\left(\mathbf{z}^{(i)}\right)^\top \mathbf{z} + \mathbf{G}_{\boldsymbol{\theta}}\left(\mathbf{z}^{(i)}\right) = 0, \tag{10b}$$

$$\nabla_{\mathbf{z}}\mathbf{H}_{\boldsymbol{\theta}}\left(\mathbf{z}^{(i)}\right)^\top \mathbf{z} + \mathbf{H}_{\boldsymbol{\theta}}\left(\mathbf{z}^{(i)}\right) \geq 0, \tag{10c}$$

with $\mathcal{L}_{\boldsymbol{\theta}}\left(\mathbf{y}^{(i)}\right)$ the Lagrangian of Problem (7), and $L^{(i)}$ a suitable approximation of $\nabla^2_{\mathbf{zz}}\mathcal{L}_{\boldsymbol{\theta}}\left(\mathbf{y}^{(i)}\right)$.

The RTI scheme solves only one QP per time instant, uses $t = 1$, and relies on the *initial value embedding*, i.e., constraints (7b) are not eliminated from the problem. Additionally, computations are split in a *preparation phase* in which the sensitivities are evaluated and the QP KKT matrix is factorized. Once the initial state is available, the QP is solved and the control is applied to the system. All details on the RTI scheme can be found in (Diehl et al., 2005; Gros et al., 2020b) and references therein.

By considering $\mathbf{y}^\diamond$ as a function of $\mathbf{s}$, implicitly defined as the optimal primal-dual solution of Problem (8), the RTI scheme can be viewed as a pathfollowing predictor-corrector scheme. Since NMPC predicts the future evolution of the system, provided that the perturbations acting on the system are not excessively large, the state at the next time step i will satisfy $\mathbf{s}_{i+1} \approx \mathbf{x}_1^\star(\mathbf{s}_i)$. RTI then exploits the good model prediction ability and the fast contraction of Newton-type methods to closely track the optimal solution $\mathbf{y}^\diamond(\mathbf{s})$. Similar considerations apply to the primal-dual solution $\mathbf{y}^\star$ of Problem (7), provided that $\|\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}\|$ is small.

4. SENSITIVITY ANALYSIS

In order to be able to use NMPC as a function approximator for Q-learning or actor-critic methods, one needs to be able to compute the parametric sensitivities $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$, $\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$, $\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}$. In the following, we first recall the results for the case in which NMPC is solved to full convergence. Afterwards, we discuss the case of the RTI scheme, which is constructed using a similar reasoning.

Note that the parametric sensitivities of optimization problems exist under the assumption that linear independence constraint qualification and the strong second-order sufficient conditions hold (Nocedal and Wright, 2006; Büskens and Maurer, 2001). Therefore, extreme care must be taken to ensure that Problem (7) satisfies both conditions.

4.1 Sensitivities of Fully Converged NMPC

We detail next how to compute the derivatives of the action-value function with respect to the parameters, in the same way as in (Gros and Zanon, 2020). To this end, we define the Lagrange function underlying NMPC problem (7) as

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y}) = \lambda_{\boldsymbol{\theta}}(\mathbf{x}_0) + \gamma^N V^{\mathrm{f}}_{\boldsymbol{\theta}}(\mathbf{x}_N) + \boldsymbol{\chi}_0^\top(\mathbf{x}_0 - \mathbf{s}) + \boldsymbol{\mu}_N^\top \mathbf{h}^{\mathrm{f}}_{\boldsymbol{\theta}}(\mathbf{x}_N) \\ + \sum_{k=0}^{N-1} \boldsymbol{\chi}_{k+1}^\top\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k,\mathbf{u}_k) - \mathbf{x}_{k+1}\right) + \boldsymbol{\nu}_k^\top \mathbf{g}_\theta(\mathbf{u}_k) \\ + \gamma^k \ell_{\boldsymbol{\theta}}(\mathbf{x}_k,\mathbf{u}_k) + \boldsymbol{\mu}_k^\top \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_k,\mathbf{u}_k) + \boldsymbol{\zeta}^\top(\mathbf{u}_0 - \mathbf{a}),\end{aligned}$$

where $\boldsymbol{\chi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\zeta}$ are the multipliers associated to constraints (7b)-(7e) and $\mathbf{y} = (\mathbf{z}, \boldsymbol{\chi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\zeta})$. Note that, for $\boldsymbol{\zeta} = 0$, $\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y})$ is the Lagrange function associated to the NMPC problem defining the value function $\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})$. We observe that (Büskens and Maurer, 2001)

$$\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y}^\star) \tag{12}$$

holds for $\mathbf{y}^\star$ given by the primal-dual solution of (7). Note that this equality holds because constraints (7b) are not an explicit function of $\boldsymbol{\theta}$. The gradient (12) is therefore straightforward to build as a by-product of solving the NMPC problem (7). We additionally observe that

$$\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(\mathbf{s}) = \nabla_{\boldsymbol{\theta}} \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}^\diamond), \tag{13}$$

where $\mathbf{y}^\diamond$ is given by the primal-dual solution to (8), i.e., (7) with constraint $\mathbf{u}_0 = \mathbf{a}$ removed and $\boldsymbol{\zeta}^\diamond = 0$.

The derivative of the optimal primal-dual solution with respect to the parameters is given by

$$\nabla_{\boldsymbol{\theta}} \mathbf{y}^\diamond = -\nabla_{\mathbf{y}} \boldsymbol{\xi}_{\boldsymbol{\theta}} \left(\mathbf{y}^\diamond\right)^{-1} \nabla_{\boldsymbol{\theta}} \boldsymbol{\xi}_{\boldsymbol{\theta}} \left(\mathbf{y}^\diamond\right), \tag{14}$$

where $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{y})$ gathers the primal-dual KKT conditions underlying the NMPC scheme (7). We remind here that the component of $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{y})$ coincide with the gradient of the Lagrangian $\nabla_{\mathbf{y}} \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y})$ with the components corresponding to the inactive constraints removed. For a complete discussion on parametric sensitivity analysis of NLPs we refer to (Büskens and Maurer, 2001) and references therein.

4.2 RTI Sensitivities

The sensitivity equations provided in the previous subsection are only valid if the NMPC problem is solved to full convergence. In case RTI or another approximate algorithm is used instead, the sensitivities must be computed differently. To that end, rather than considering RTI as a scheme approximately solving Problem (7) or (8), it is best to view it as a scheme solving the QP Problem (10) to compute $Q_{\boldsymbol{\theta}}$, $V_{\boldsymbol{\theta}}$, and $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. Note that these function approximations depend on the initial guess used in the computation of the QP data. However, since RTI always uses a good initial guess which is close to the optimal solution, the impact of variations in the initial guess on the function approximations is small.

Analogously to the fully converged case, we denote the solution to (10) as $\mathbf{y}^\star_{\mathrm{QP}}$, $\mathbf{y}^\diamond_{\mathrm{QP}}$ to respectively refer to the cases in which $\mathbf{u}_0 = \mathbf{a}$ is enforced or not. Then, based on the results provided above, the sensitivities are given by

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) &= \nabla_{\boldsymbol{\theta}} \mathcal{L}^{\mathrm{QP}}_{\boldsymbol{\theta}}(\mathbf{y}^\star_{\mathrm{QP}}),\\
\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(\mathbf{s}) &= \nabla_{\boldsymbol{\theta}} \mathcal{L}^{\mathrm{QP}}_{\boldsymbol{\theta}}(\mathbf{y}^\diamond_{\mathrm{QP}}),\\
\nabla_{\boldsymbol{\theta}} \mathbf{y}^\diamond_{\mathrm{QP}} &= -\nabla_{\mathbf{y}} \boldsymbol{\xi}^{\mathrm{QP}}_{\boldsymbol{\theta}} \left(\mathbf{y}^\diamond_{\mathrm{QP}}\right)^{-1} \nabla_{\boldsymbol{\theta}} \boldsymbol{\xi}^{\mathrm{QP}}_{\boldsymbol{\theta}} \left(\mathbf{y}^\diamond_{\mathrm{QP}}\right).
\end{aligned}$$

We stress for completeness that, for $\mathbf{y} = \bar{\mathbf{y}} + \Delta\mathbf{y}$ and $L^{(i)} = \nabla^2_{\mathbf{zz}} \mathcal{L}_{\boldsymbol{\theta}}\left(\mathbf{y}^{(i)}\right)$, the QP Lagrangian satisfies

$$\begin{aligned}
\mathcal{L}^{\mathrm{QP}}_{\boldsymbol{\theta}}(\mathbf{y}) &= \mathcal{L}_{\boldsymbol{\theta}}(\bar{\mathbf{y}}) + \nabla_{\mathbf{y}} \mathcal{L}_{\boldsymbol{\theta}}(\bar{\mathbf{y}})^\top \Delta\mathbf{y} + \Delta\mathbf{y}^\top \nabla^2_{\mathbf{yy}} \mathcal{L}_{\boldsymbol{\theta}}(\bar{\mathbf{y}}) \Delta\mathbf{y}\\
&= \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y}) + \mathcal{O}\left(\|\Delta\mathbf{y}\|^3\right),
\end{aligned}$$

such that

$$\boldsymbol{\xi}^{\mathrm{QP}}_{\boldsymbol{\theta}}(\mathbf{y}) = \boldsymbol{\xi}_{\boldsymbol{\theta}}(\bar{\mathbf{y}}) + \nabla_{\mathbf{y}} \boldsymbol{\xi}_{\boldsymbol{\theta}}(\bar{\mathbf{y}})^\top \Delta\mathbf{y} + \mathcal{O}\left(\|\Delta\mathbf{y}\|^2\right).$$

5. RL BASED ON RTI NMPC

In this section, we revise the standard RL algorithms presented in Section 2 and introduce some adaptations in order to account for the peculiarities of using NMPC as a function approximator.

The update (4b) can also be written as the solution to a fitting problem. To that end, we introduce the linearization

$$Q^{\mathrm{lin}}_{\bar{\boldsymbol{\theta}}}(\mathbf{s}_k,\mathbf{a}_k) = Q_{\boldsymbol{\theta}}(\mathbf{s}_k,\mathbf{a}_k) - \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}_k,\mathbf{a}_k)(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}).$$

Then, (4b) is equivalent to $\boldsymbol{\theta} \leftarrow \bar{\boldsymbol{\theta}}^*$, with $\bar{\boldsymbol{\theta}}^*$ computed as the optimal solution of the fitting problem

$$\min_{\bar{\boldsymbol{\theta}}} \left(\ell(\mathbf{s}_k,\mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}) - Q^{\mathrm{lin}}_{\bar{\boldsymbol{\theta}}}(\mathbf{s}_k,\mathbf{a}_k)\right)^2 + \frac{1}{\alpha}\left\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|_2^2. \tag{15}$$

In order to guarantee that the NMPC cost is positive-definite, we propose to compute $\boldsymbol{\theta}^*$ as the solution of a slightly modified version of (15), i.e.,

$$\min_{\bar{\boldsymbol{\theta}}} \left(\ell(\mathbf{s}_k,\mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}) - Q^{\mathrm{lin}}_{\bar{\boldsymbol{\theta}}}(\mathbf{s}_k,\mathbf{a}_k)\right)^2 + \frac{1}{\alpha}\left\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|_2^2 \tag{16a}$$

$$\text{s.t. } \nabla^2 \ell_{\boldsymbol{\theta}} \succ 0, \quad \nabla^2 V^{\mathrm{f}}_{\boldsymbol{\theta}} \succ 0, \quad \forall \mathbf{s},\mathbf{a} \in \mathrm{dom}\{(7)\}. \tag{16b}$$

Constraint (16b) imposes that the Hessian of the stage and terminal cost is positive-definite everywhere. We left this formulation intentionally implicit, since it can be enforced in several ways. For quadratic functions it can, e.g., be formulated as an LMI. Alternatively, one can rely on parametrizations which deliver a positive-definite function by construction. Note that the positive-definiteness requirement is only necessary for those state-action pairs for which all NMPC problems have a solution. In principle, this set could be further restricted to the state-action pairs which will be visited when operating the system.

Additionally to the enforcement of positive-definiteness, as stressed in (Zanon et al., 2019), the other main advantage of formulating Q-learning as a fitting problem is the possiblity to introduce globalization strategies such as, e.g., line search. This feature provides at least the guarantee that the proposed update $\Delta\boldsymbol{\theta}$ reduces the TD-error for the current sample, which is not the case with (4b).

Similarly to Q-learning, for actor-critic methods, the parameter update (6) can be replaced by $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$ with $\boldsymbol{\theta}^*$ solution of

$$\min_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}_k))^\top (\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}) + \frac{1}{\alpha}\left\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|_2^2 \tag{17a}$$

$$\text{s.t. } \nabla^2 \ell_{\boldsymbol{\theta}} \succ 0, \quad \nabla^2 V^{\mathrm{f}}_{\boldsymbol{\theta}} \succ 0, \quad \forall \mathbf{s},\mathbf{a} \in \mathrm{dom}\{(7)\}. \tag{17b}$$

with $\Delta\boldsymbol{\theta}^* = \bar{\boldsymbol{\theta}}^* - \boldsymbol{\theta}$. The considerations made for Problem (16) also apply to Problem (17).

Remark 1. Both in Problem (16) and (17) one could use the nonlinear model of Q and J rather than the linear one. The thorough investigation of the effects of this choice are the subject of ongoing research.

6. SIMULATIONS

In this section we demonstrate the effectiveness of the proposed combination of RL and RTI-based NMPC with an example from the process industry, i.e., the evaporation process modelled in (Wang and Cameron, 1994; Sonntag et al., 2006) and used in (Amrit et al., 2013; Zanon et al., 2016) to demonstrate the potential of economic MPC in the nominal case. The model equations are given by

$$M\dot{X}_2 = F_1 X_1 - F_2 X_2, \qquad C\dot{P}_2 = F_4 - F_5, \tag{18}$$

where

$$\begin{aligned}
T_2 &= aP_2 + bX_2 + c, & T_3 &= dP_2 + e,\\
\lambda F_4 &= Q_{100} - F_1 C_{\mathrm{p}}(T_2 - T_1), & T_{100} &= fP_{100} + g,
\end{aligned}$$

$$Q_{100} = UA_1(T_{100} - T_2), \qquad UA_1 = h(F_1 + F_3),$$
$$Q_{200} = \frac{UA_2(T_3 - T_{200})}{1 + UA_2/(2C_\mathrm{p}F_{200})}, \qquad F_{100} = \frac{Q_{100}}{\lambda_\mathrm{s}},$$
$$\lambda F_5 = Q_{200}, \qquad F_2 = F_1 - F_4,$$

with states $\mathbf{x} = (X_2,\, P_2)$ (concentration and pressure) and controls $\mathbf{u} = (P_{100},\, F_{200})$ (pressure and flow). The model parameters are given in (Amrit et al., 2013). 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. In this example, we assume a uniform distribution around the nominal value with interval $\Delta_{X_1} = \pm 1$, $\Delta_{F_1} = \pm 2$, $\Delta_{T_1} = \pm 8$, $\Delta_{T_{200}} = \pm 5$. Additionally, the controller must satisfy bounds $(25, 40) \leq (X_2, P_2) \leq (100, 80)$ on the states and $100 \leq (P_{100}, F_{200}) \leq 400$ on the controls. In particular, the bound $X_2 \geq 25$ is introduced in order to ensure sufficient quality of the product. The state bounds are relaxed as $(\mathbf{x} - \mathbf{x}_\mathrm{l} - \boldsymbol{\sigma}, \mathbf{x}_\mathrm{u} - \mathbf{x} - \boldsymbol{\sigma})$, where $\boldsymbol{\sigma}$ is a slack variable introduced as a ficticious control and penalized in the cost in order to introduce an exact constraint relaxation (Scokaert and Rawlings, 1999). The stage cost is then given by

$$\ell(\mathbf{x}, \mathbf{u}) = 10.09(F_2 + F_3) + 600F_{100} + 0.6F_{200} + \boldsymbol{\sigma}^\top B_{\boldsymbol{\sigma}} \boldsymbol{\sigma} + \mathbf{b}_{\boldsymbol{\sigma}}^\top \boldsymbol{\sigma}.$$

For the given stage cost, the nominal model is optimally operated at the steady state $\mathbf{x}_\mathrm{s} = (25, 49.74)$, $\mathbf{u}_\mathrm{s} = (191.71, 215.89)$, with stage cost $\ell(\mathbf{x}_\mathrm{s}, \mathbf{u}_\mathrm{s}) = \ell_\mathrm{s}$.

We parametrize an NMPC controller as in (7), i.e., a nonlinear MPC formulation, with $N = 10$. Functions $\lambda_{\boldsymbol{\theta}}, V^\mathrm{f}_{\boldsymbol{\theta}}, \ell_{\boldsymbol{\theta}}$ are quadratic of the form

$$\dagger = \begin{bmatrix} \mathbf{x} - \mathbf{x}_\mathrm{s} \\ \mathbf{u} - \mathbf{u}_\mathrm{s} \end{bmatrix}^\top B_\dagger \begin{bmatrix} \mathbf{x} - \mathbf{x}_\mathrm{s} \\ \mathbf{u} - \mathbf{u}_\mathrm{s} \end{bmatrix} + \mathbf{b}_\dagger^\top \begin{bmatrix} \mathbf{x} - \mathbf{x}_\mathrm{s} \\ \mathbf{u} - \mathbf{u}_\mathrm{s} \end{bmatrix} + c_\dagger,$$

defined by Hessian $B_\dagger$, gradient $\mathbf{b}_\dagger$, and constant $c_\dagger$, with a minimum in x_s, u_s, and $\dagger = \{\lambda_{\boldsymbol{\theta}}, V^\mathrm{f}_{\boldsymbol{\theta}}, \ell_{\boldsymbol{\theta}}\}$. The model is parametrized as the nominal model with the addition of a constant, i.e., $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{c}_\mathbf{f}$. The control constraints are fixed and the state constraints are parametrized as simple bounds, i.e., $\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_\mathrm{l} - \boldsymbol{\sigma}, \mathbf{x}_\mathrm{u} - \mathbf{x} - \boldsymbol{\sigma})$. The parameter vector reads as:

$$\boldsymbol{\theta} = (B_\lambda, \mathbf{b}_\lambda, c_\lambda, B_{V^\mathrm{f}}, \mathbf{b}_{V^\mathrm{f}}, B_l, \mathbf{b}_l, \mathbf{c}_\mathbf{f}, \mathbf{x}_\mathrm{l}, \mathbf{x}_\mathrm{u}).$$

Constants $\mathbf{b}_{\boldsymbol{\sigma}} = 10^5 \mathbf{1}$, $B_{\boldsymbol{\sigma}} = I$ are fixed and assumed to reflect the known cost of violating the state constraints.

We apply Q-learning and use $\alpha = 10^{-3}$. 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 apply the action

$$\mathbf{a} = \mathrm{sat}(\mathbf{e}, \mathbf{u}_\mathrm{l}, \mathbf{u}_\mathrm{u}), \qquad \mathbf{e} \sim \mathcal{N}(0, \sqrt{10} I),$$

where $\mathrm{sat}(\cdot, \mathbf{u}_\mathrm{l}, \mathbf{u}_\mathrm{u})$ saturates the input between its lower and upper bounds $\mathbf{u}_\mathrm{l}, \mathbf{u}_\mathrm{u}$, respectively.

We initialize the ENMPC scheme by the naive initial guess $H_\ell = I$, $H_{V^\mathrm{f}} = I$, $c_\lambda = \ell_\mathrm{s}$, $\mathbf{x}_\mathrm{l} = (25, 40)$, $\mathbf{x}_\mathrm{u} = (100, 80)$, while all other parameters are 0. As displayed in Figure 1, the algorithm converges to a constant parameter value while reducing the average TD-error.

We performed a simulation to compare the RL-tuned RTI NMPC scheme to the same scheme using as parameter (a) the naive initial guess, and (b) the nominal model and the

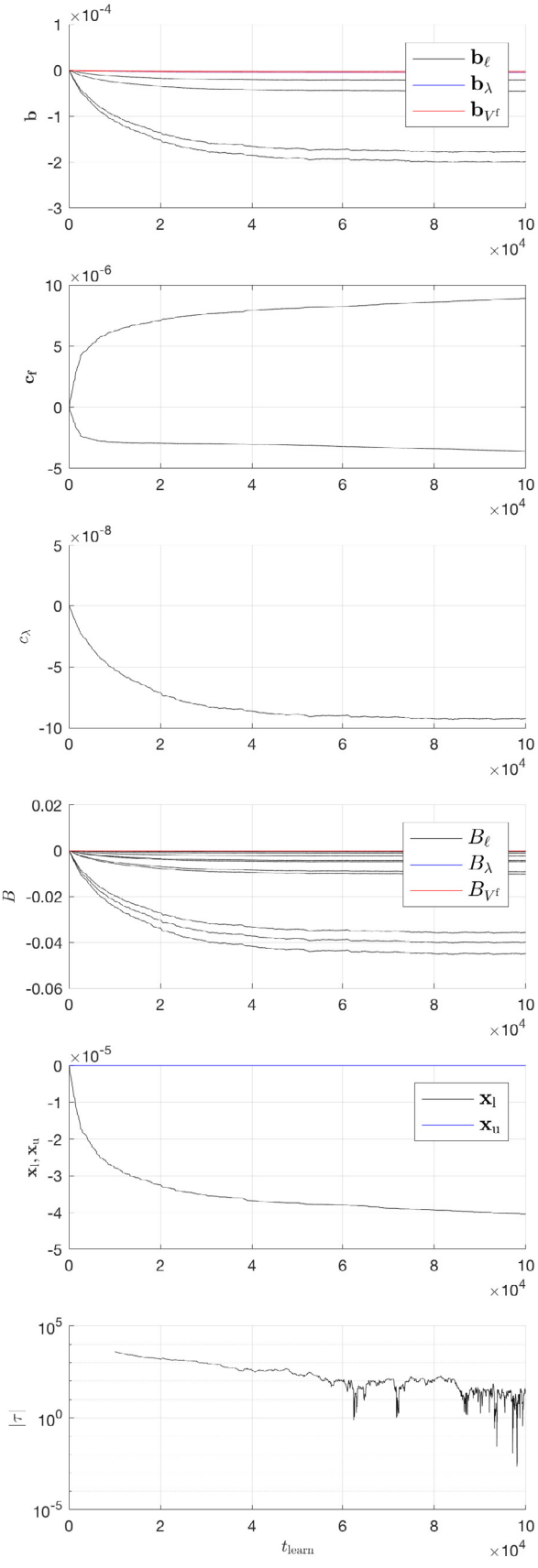

Fig. 1. Evolution of the parameters (increment w.r.t. the initial guess value) and of the TD error (averaged over the preceding 10000 samples).

cost tuned using the economic-based approach proposed in (Zanon et al., 2016). The economic gain obtained by RL is approximately 21 % and 15 % respectively.

Note that when RL is deployed with fully converged NMPC as function approximator, the obtained optimal parameters are different from those obtained with RTI NMPC as function approximator.

7. CONCLUSIONS AND OUTLOOK

In this paper we have extended the framework of RL based on NMPC in order to make it possible to deploy computationally efficient algorithms for NMPC such as the RTI scheme and we have demonstrated the effectiveness of the approach with an example from the process industry.

Future research will aim at further developing the framework of MPC-based RL by providing a thorough stability analysis, as well further extending the analysis on rigorous safety guarantees started in (Zanon and Gros, 2019; Gros and Zanon, 2020b).

REFERENCES

Abbeel, P., Coates, A., Quigley, M., and Ng, A.Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, 2007. MIT Press.

Amrit, R., Rawlings, J.B., and Biegler, L.T. (2013). Optimizing process economics online using model predictive control. *Computers & Chemical Engineering*, 58, 334 – 343.

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

Diehl, M., Bock, H., and Schlöder, J. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5), 1714–1736.

Diehl, M., Bock, H., Schlöder, J., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585.

Gros, S. and Zanon, M. (2020). Data-Driven Economic NMPC Using Reinforcement Learning. *IEEE Transactions on Automatic Control*, 65(2), 636–648.

Gros, S. and Zanon, M. (2020a). Reinforcement Learning for Mixed-Integer Problems Based on MPC. In *21st IFAC World Congress*. (accepted).

Gros, S. and Zanon, M. (2020b). Safe Reinforcement Learning Based on Robust MPC and Policy Gradient Methods. *IEEE Transactions on Automatic Control (submitted)*.

Gros, S., Zanon, M., and Bemporad, A. (2020a). Safe Reinforcement Learning via Projection on a Safe Set: How to Achieve Optimality? In *21st IFAC World Congress*. (accepted).

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2020b). From Linear to Nonlinear MPC: Bridging the Gap via Real-Time Iteration. *International Jounal of Control*.

Grüne, L. and Pannek, J. (2011). *Nonlinear Model Predictive Control*. Springer, London.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition.

Ohtsuka, T. (2004). A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. *Automatica*, 40(4), 563–574.

Rawlings, J. and Mayne, D. (2009). *Model Predictive Control: Theory and Design*. Nob Hill.

Scokaert, P. and Rawlings, J. (1999). Feasibility Issues in Linear Model Predictive Control. *AIChE Journal*, 45(8), 1649–1659.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529, 484–503.

Sonntag, C., Stursberg, O., and Engell, S. (2006). 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)*, 211–216.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction. Second Edition*. MIT press Cambridge.

Theocharous, G., Thomas, P.S., and Ghavamzadeh, M. (2015). Personalized Ad Recommendation Systems for Life-Time Value Optimization with Guarantees. In *IJCAI*, 1806–1812.

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

Wang, X., Stoev, J., Pinte, G., and Swevers, J. (2012). Energy optimal point-to-point motion using Model Predictive Control. In *Proceedings ASME 2012 5th Annual Dynamic Systems and Control Conference*.

Watkins, C. (1989). *Learning from delayed rewards*. Ph.D. thesis, King's College, Cambridge.

Zanon, M. and Gros, S. (2019). Safe Reinforcement Learning Using Robust MPC. *Transaction on Automatic Control*. Https://arxiv.org/abs/1906.04005, (submitted).

Zanon, M., Gros, S., and Bemporad, A. (2019). Practical Reinforcement Learning of Stabilizing Economic MPC. In *Proceedings of the European Control Conference*.

Zanon, M., Gros, S., and Diehl, M. (2016). A Tracking MPC Formulation that is Locally Equivalent to Economic MPC. *Journal of Process Control*, 45, 30 – 42.

Zavala, V.M. and Biegler, L. (2009). The Advanced Step NMPC Controller: Optimality, Stability and Robustness. *Automatica*, 45, 86–93.