Topics in Systems and Control Theory

Reinforcement Learning

Self study Course

Arash Bahari Kordabad

Topics in Systems and Control Theory,
TK8111, Department of Engineering Cybernetics,
Norwegian University of Science and Technology (NTNU)
The course topic is Reinforcement Learning and it was done under the supervision of Prof. Sebastien Gros in fall 2022.

Contents

head1.png

1. Introduction and Background

head1.png

Markov Decision Processes (MDPs) provide a standard framework for the optimal control of discrete-time stochastic processes, where the stage cost and transition probability depend only on the current state and the current input of the system [1]. MDPs also formally describe a (fully observable) environment for Reinforcement Learning (RL) [2]. The defining property of MDPs is the Markov property which says that the future is independent of the past given the current state.

1.1 Markov Decision Process

An MDP operates over given state and action (aka input) spaces S, A, respectively. These spaces can be discrete (i.e. integer sets), continuous, or mixed. We denoted p as a conditional probability (measure) defining the dynamics of the system considered, i.e. for a given state-action pair $\mathbf{s}, \mathbf{a} \in S \times A$, the successive state $\mathbf{s}_+$ is distributed according to

$$\mathbf{s}_+ \sim p(\cdot \,|\, \mathbf{s}, \mathbf{a}) \,. \tag{1.1}$$

The input $\mathbf{a}$ applied to the system for a given state $\mathbf{s}$ is selected by a deterministic policy $\boldsymbol{\pi} : S \rightarrow A$. Solving an MDP is then the problem of finding the optimal policy $\boldsymbol{\pi}^\star$, solution of:

$$\boldsymbol{\pi}^\star \in \arg\min_{\boldsymbol{\pi}} \; J(\boldsymbol{\pi}) \,, \tag{1.2}$$

where $J(\boldsymbol{\pi})$ is some form of cumulative cost, indicating the closed-loop performance of the system for policy $\boldsymbol{\pi}$, and depends on the optimality criteria describing the MDP.

1.1.1 Discounted setting

In the discounted setting, an MDP is defined by the triplet (L, γ, p), where $L : S \times A \rightarrow \mathbb{R}$ is a stage cost, $\gamma \in (0, 1]$ a discount factor and the performance function $J(\boldsymbol{\pi})$ is defined as follows:

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}\left(\mathbf{s}_k\right) \right] , \tag{1.3}$$

and the expected value operator $\mathbb{E}[.]$ is taken over the (possibly) stochastic closed loop trajectories of the system. Discussing the solution of MDPs is often best done via the Bellman equations defining implicitly the optimal value function $V^\star : S \to \mathbb{R}$, the optimal action-value function $Q^\star : S \times A \to \mathbb{R}$, and the optimal advantage function $A^\star : S \times A \to \mathbb{R}$ as follows:

$$V^\star(\mathbf{s}) = \min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}) \ , \tag{1.4a}$$

$$Q^\star(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}\left[V^\star(\mathbf{s}_+) \mid \mathbf{s}, \mathbf{a}\right] \ , \tag{1.4b}$$

$$A^\star(\mathbf{s}, \mathbf{a}) = Q^\star(\mathbf{s}, \mathbf{a}) - V^\star(\mathbf{s}) \ . \tag{1.4c}$$

The optimal policy then reads as:

$$\boldsymbol{\pi}^\star(\mathbf{s}) = \arg\min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}) \tag{1.5}$$

Then one can verify that

$$0 = \min_{\mathbf{a}} A^\star(\mathbf{s}, \mathbf{a}), \quad \boldsymbol{\pi}^\star(\mathbf{s}) \in \arg\min_{\mathbf{a}} A^\star(\mathbf{s}, \mathbf{a}) \, . \tag{1.6}$$

1.1.2 Undiscounted setting

Undiscounted MDPs refer to MDPs with a discount factor $\gamma = 1$. If using $\gamma = 1$ in (1.3), $V^\star$ is in general unbounded and the MDP ill-posed. In order to tackle this issue, alternative optimality criteria are needed. Gain optimality is one of the common criteria in the undiscounted setting. Gain optimality is defined based on the following average-cost problem:

$$\bar{V}^\star(\mathbf{s}) := \min_{\boldsymbol{\pi}} \lim_{N \to \infty} \frac{1}{N} \mathbb{E}\left[\sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k) \right] , \tag{1.7}$$

for all initial state $\mathbf{s}_0^{\boldsymbol{\pi}} = \mathbf{s}, \forall \boldsymbol{\pi}$, where $\bar{V}^\star$ is the optimal average cost. We denote the optimal policy solution of (1.7) as $\bar{\boldsymbol{\pi}}^\star$. This optimal policy is called *gain optimal*.

The gain optimal policy $\bar{\boldsymbol{\pi}}^\star$ may not be unique. Moreover, the optimal average cost $\bar{V}^\star$ is commonly assumed to be independent of the initial state $\mathbf{s}$ [3]. This assumption e.g. holds for *unichain* MDPs, in which under any policy any state can be reached in finite time from any other state.

Unfortunately, the gain optimality criterion only considers the optimal steady-state distribution and it overlooks transients. As an alternative, *bias optimality* considers the optimality of the transients. Precisely, bias optimality can be formulated through the following OCP:

$$\tilde{V}^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \mathbb{E}\left[\sum_{k=0}^{\infty} \left(L(\mathbf{s}_k, \mathbf{a}_k) - \bar{V}^\star\right) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k) \right] , \tag{1.8}$$

where $\tilde{V}^\star$ is the optimal value function associated with bias optimality. Note that (1.8) can be seen as a special case of the discounted setting in (1.3) when $\gamma = 1$ and the optimal average cost $\bar{V}^\star$ is subtracted from the stage cost in (1.3). Therefore, for the rest of the report, we will consider the discounted setting (1.3). Without loss of generality, we assume that $\bar{V}^\star = 0$ in the case $\gamma = 1$. This choice yields a well-posed optimal value function in the undiscounted setting. Clearly, if this does not hold, one can shift the stage cost to achieve $\bar{V}^\star = 0$.

In this report, we will investigate techniques that solve MDPs either exactly (using e.g., DP) or approximately (using RL). Moreover, we address recent developments in using Model Predictive Control (MPC) as a function approximator of RL to solve MDPs without having the exact system.

1.2 Outline

The rest of the report is structured as follows: chapter 2 details the Dynamic Programming (DP) method as an exact solution for MDPs with slightly low dimensions and less complexity. In chapter 3 we provide RL techniques as practical tools to tackle MDPs. In this section, we detail Q-learning and policy gradient method and provide recent developments on the Quasi-Newton policy gradient method. Finally, in chapter 4 we propose and justify the use of undiscounted Model Predictive Control (MPC) scheme as a function approximator of MDPs.

2. Dynamic Programming

head1.png

Dynamic programming (DP) is a well-known method to provide exact solutions for MDPs with fairly low-dimensional dynamics. Moreover, it assumes full knowledge of the MDP.

In this chapter, we consider MDPs with discounted cumulative costs. We define the value function and action-value function associated with a given policy $\boldsymbol{\pi}$, as follows:

$$\begin{aligned} Q^{\boldsymbol{\pi}}\left(\mathbf{s},\mathbf{a}\right) &= L\left(\mathbf{s},\mathbf{a}\right) + \gamma\mathbb{E}\left[V^{\boldsymbol{\pi}}\left(\mathbf{s}_{+}\right)\,|\,\mathbf{s},\mathbf{a}\right]\,, && \text{(2.1a)}\\ V^{\boldsymbol{\pi}}\left(\mathbf{s}\right) &= Q^{\boldsymbol{\pi}}\left(\mathbf{s},\boldsymbol{\pi}(\mathbf{s})\right)\,, && \text{(2.1b)} \end{aligned}$$

There are two main methods in the literature on DP, to tackle MDPs: policy iteration and value iteration. In the following we briefly formulate them.

2.1 Policy iteration

This method tries to generate a sequence of policies $\boldsymbol{\pi}, \boldsymbol{\pi}', \ldots$, starting from an arbitrary policy, to converge the optimal policy $\boldsymbol{\pi}^{\star}$. First, for a given policy, we need to evaluate the corresponding value function. This stage is called **policy evaluation**. More specifically, for a given policy $\boldsymbol{\pi}$, we generate a sequence of value functions $V_0^{\boldsymbol{\pi}}, V_1^{\boldsymbol{\pi}}, \ldots$ until convergence to the value policy associated with the policy $\boldsymbol{\pi}$. To this end, we use the Bellman equations in (2.1) as follows:

$$V_{k+1}^{\boldsymbol{\pi}}\left(\mathbf{s}\right) \leftarrow L\left(\mathbf{s},\boldsymbol{\pi}(\mathbf{s})\right) + \gamma\mathbb{E}\left[V_k^{\boldsymbol{\pi}}\left(\mathbf{s}_{+}\right)\,|\,\mathbf{s},\boldsymbol{\pi}(\mathbf{s})\right]\,, \tag{2.2}$$

The sequence will converge to $V_0^{\boldsymbol{\pi}}, V_1^{\boldsymbol{\pi}}, \ldots, V_\infty^{\boldsymbol{\pi}} = V^{\boldsymbol{\pi}}$.

Then we need to *improve* the policy $\boldsymbol{\pi}$ in order to achieve a better value function. Therefore, the stage of **policy improvement** is as follows:

$$\boldsymbol{\pi}'(\mathbf{s}) \in \min_{\mathbf{a}} L(\mathbf{s},\mathbf{a}) + \gamma\mathbb{E}\left[V_\infty^{\boldsymbol{\pi}}(\mathbf{s}_{+})\,|\,\mathbf{s},\mathbf{a}\right] \tag{2.3}$$

It is shown that these steps generate better policy $\boldsymbol{\pi}'(\mathbf{s})$, i.e., $V^{\boldsymbol{\pi}'}(\mathbf{s}) \leq V^{\boldsymbol{\pi}}(\mathbf{s})$.

2.2 Value iteration

Value iteration tries to capture the optimal value function directly, instead of evaluating the value function of a given policy. Therefore, value iteration updates the policy at every iteration and generates the sequence $V_0^\star, V_1^\star, \dots$ until convergence to the optimal value function $V_\infty^\star = V^\star$. More specifically, we use the following update rule:

$$V_{k+1}^\star(\mathbf{s}) = \min_{\mathbf{a}} L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}\left[V_k^\star(\mathbf{s}_+) \mid \mathbf{s}, \mathbf{a}\right], \tag{2.4}$$

and finally, we get the optimal policy by solving the following optimization:

$$\boldsymbol{\pi}^\star = \arg\min_{\mathbf{a}} L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}\left[V_\infty^\star(\mathbf{s}_+) \mid \mathbf{s}, \mathbf{a}\right] \tag{2.5}$$

■ **Example 2.1** A simple model for the battery storage reads as [4]:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \alpha\left(\Delta_k + \mathbf{a}_k\right), \tag{2.6}$$

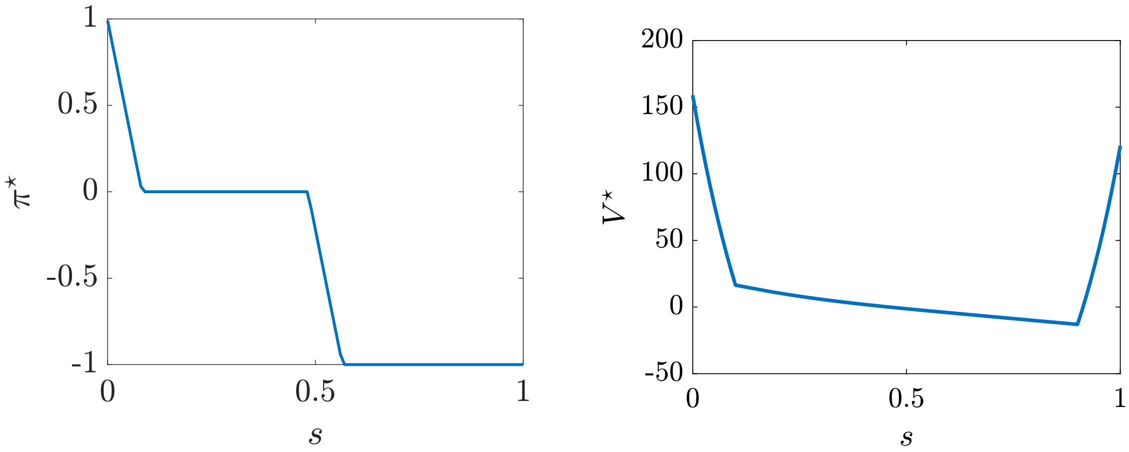
where $\mathbf{s}_k \in [0, 1]$ is the State-of-Charge (SOC) of the battery and the interval $[0, 1]$ represents the SOC levels considered as non-damaging for the battery (typically 20%-80% range of the physical SOC). Constant α is a positive value that reflects the battery size. Variable $\Delta_k \sim \mathcal{N}\left(\bar{\delta}^X, \sigma^X\right)$ is the difference between the local power production and demand, which–for the sake of simplicity–is considered as a Normal centered random variable here, where $\bar{\delta}^X$ and σ^X are the mean and variance of the Gaussian distribution. Input $\mathbf{a}_k \in [-\bar{U}, \bar{U}]$ is the power bought from (for $\mathbf{a}_k > 0$) and sold to (for $\mathbf{a}_k < 0$) the power grid. The economic stage cost can be written as follows:

$$L(\mathbf{s}_k, \mathbf{a}_k) = \begin{cases} \phi_b \mathbf{a}_k & \text{if} \quad \mathbf{a}_k \geq 0 \\ \phi_s \mathbf{a}_k & \text{if} \quad \mathbf{a}_k < 0 \end{cases}, \tag{2.7}$$

where $\phi_b \geq 0$ is the buying price and $\phi_s \geq 0$ is the selling price, and we assume that $\phi_b \geq \phi_s$. For the sake of simplicity, we consider the prices ϕ_b and ϕ_s as constants. As can be seen in fig. 2.1, the optimal policy has a bang-bang-like structure. When the battery is at $\mathbf{s} \approx 0$, maximum buying is the optimal policy. Then for a fairly large subset of the states ($\mathbf{s} \approx [0.05, 0.5]$), no exchange with the grid is the optimal policy. For a high SOC ($\mathbf{s} \approx [0.55, 1]$), maximum selling is optimum. ■

Although DP algorithms can provide exact solutions to the optimal policy and value functions of MDPs, they can only be applied for exceptional cases, especially if the system is simple enough with low dimensions. Moreover, for problems with continuous state and action spaces, one needs to discretize the spaces and apply the algorithms to the grids. Then one can use different types of interpolation and extrapolation to evaluate the values in the whole space.

Due to these complexities, in practice, approximate DP and RL are alternatives to DP. These methods use some form of function approximators for the optimal value function and optimal policy to avoid computational complexity. Then these methods give an approximation solution for the MDPs. Of course, a richer function approximator can provide a better approximation, but one has to care about the complexity simultaneously. The next chapter details RL methods to solve MDPs.



(a) Optimal policy $\boldsymbol{\pi}^\star$ resulted from DP.

(b) Optimal value function $V^\star$ resulted from DP.

Figure 2.1: Using Dynamic Programming to solve MDPs

3. Reinforcement Learning

head1.png

Reinforcement Learning (RL) is a powerful tool for tackling MDPs without prior knowledge of the process to be controlled [5]. Instead of finding the exact solution, RL seeks to optimize the parameters of a function approximation underlying a given policy in view of minimizing the expected cumulative cost. RL methods solve MDPs usually either directly based on an approximation of the optimal policy or indirectly based on an approximation of the action-value function. In this chapter we formulate Q-learning as a well-known indirect RL method, and policy gradient methods as popular direct methods.

3.1 Q-learning

The indirect RL method, generically labelled Q-learning, approximates the optimal action-value function $Q^\star$ via a parametrized function approximator $Q_{\boldsymbol{\theta}}$. The parameters $\boldsymbol{\theta}$ are then adjusted using data such that $Q_{\boldsymbol{\theta}^\star} \approx Q^\star$ for the optimal parameters $\boldsymbol{\theta}^\star$.

Q-learning solves the following Least Square (LS) problem in order to achieve the best parameters $\boldsymbol{\theta}^\star$, describing the optimal action-value function $Q^\star$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) - Q^\star(\mathbf{s}_k, \mathbf{a}_k)\right)^2\right]. \tag{3.1}$$

Temporal-Difference (TD) learning is a common way to tackle (3.1). More specifically, a basic TD-based learning step uses the following update rule for the parameters $\boldsymbol{\theta}$ at time instance k in the discounted setting (and the undiscounted setting when $\gamma = 1$):

$$\delta_k = L(\mathbf{s}_k, \mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{3.2a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \zeta \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{3.2b}$$

where the scalar $\zeta > 0$ is the learning step-size, δ_k is labelled the TD error. Note that there are more advanced methods to tackle (3.1) in the literature.

An approximation of the optimal policy $\boldsymbol{\pi}^\star$ can then be obtained using:

$$\hat{\boldsymbol{\pi}}^\star(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a}) \tag{3.3}$$

3.2 Policy Gradient

Deterministic policy gradient algorithms are widely used in RL with continuous action spaces [5]. These methods attempt to learn the optimal parameters of a parameterized policy $\boldsymbol{\pi_\theta}$ using only state transitions observed on the real system. These methods commonly use gradient descent methods to optimize a discounted sum of stage costs, called closed-loop performance $J(\boldsymbol{\theta})$. Depending on the policy type, these approaches are divided into the deterministic and the stochastic policy gradient methods. In the stochastic policy gradient methods, a parametrized distribution of action $\mathbf{a}$ conditioned on each state $\mathbf{s}$ taking the form of $\boldsymbol{\pi_\theta}(\mathbf{a}\,|\,\mathbf{s})$ is considered, while deterministic policy methods use $\mathbf{a} = \boldsymbol{\pi_\theta}(\mathbf{s})$ to specify a deterministic action for each state $\mathbf{s}$. Both methods adjust the parameter vector $\boldsymbol{\theta}$ in order to optimize J. In practice, stochastic policy gradient may need more data when the action space has many dimensions [6]. Hence, in this section, we focus on deterministic policies.

The value function $V^{\boldsymbol{\pi_\theta}}$ and action-value function $Q^{\boldsymbol{\pi_\theta}}(\mathbf{s}, \mathbf{a})$ are defined as follows:

$$Q^{\boldsymbol{\pi_\theta}}(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_{p(\cdot\,|\,\mathbf{s},\mathbf{a})}\left[V^{\boldsymbol{\pi_\theta}}(\mathbf{s}^+)\,|\,\mathbf{s}, \mathbf{a}\right], \tag{3.4a}$$

$$V^{\boldsymbol{\pi_\theta}}(\mathbf{s}) = Q^{\boldsymbol{\pi_\theta}}(\mathbf{s}, \boldsymbol{\pi_\theta}(\mathbf{s})), \tag{3.4b}$$

where $\gamma \in (0, 1]$ is a discount factor. The performance objective $J(\boldsymbol{\theta})$ is given as follows[1]:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}_0}\left[V^{\boldsymbol{\pi_\theta}}(\mathbf{s}_0)\right] = \mathbb{E}_{\mathbf{s}}\left[L(\mathbf{s}, \boldsymbol{\pi_\theta}(\mathbf{s}))\right]. \tag{3.5}$$

Note that we simplified the expectation notation $\mathbb{E}_{\mathbf{s}_0\sim p_1(\mathbf{s}_0)}[\cdot] = \mathbb{E}_{\mathbf{s}_0}[\cdot]$ and $\mathbb{E}_{\mathbf{s}}[\cdot]$ is taken over the expected sum of the discounted state distribution of the Markov chain in closed-loop with policy $\boldsymbol{\pi_\theta}$. The purpose is to solve the following optimization problem:

$$\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}). \tag{3.6}$$

In the following, we make an assumption in order to guarantee the existence of the policy gradient and we recall the deterministic policy gradient theorem.

Assumption 1. $p(\mathbf{s}'\,|\,\mathbf{s}, \mathbf{a})$, $\nabla_{\mathbf{a}} p(\mathbf{s}'\,|\,\mathbf{s}, \mathbf{a})$, $\boldsymbol{\pi_\theta}(\mathbf{s})$, $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}(\mathbf{s})$, $L(\mathbf{s}, \mathbf{a})$, $\nabla_{\mathbf{a}} L(\mathbf{s}, \mathbf{a})$, $p_1(\mathbf{s})$ *are continuous in all parameters and variables* $\mathbf{s}$, $\mathbf{a}$, $\mathbf{s}'$, $\boldsymbol{\theta}$. *Also there exist* b *and* $\bar{L}$ *such that:*

$$\begin{aligned} &\sup_{\mathbf{s}} p_1(\mathbf{s}) < b, && \sup_{\{\mathbf{a},\mathbf{s},\mathbf{s}'\}} p(\mathbf{s}'\,|\,\mathbf{s}, \mathbf{a}) < b, \\ &\sup_{\{\mathbf{a},\mathbf{s}\}} \|\nabla_{\mathbf{a}} L(\mathbf{s}, \mathbf{a})\| < \bar{L}, && \sup_{\{\mathbf{a},\mathbf{s},\mathbf{s}'\}} \|\nabla_{\mathbf{a}} p(\mathbf{s}'\,|\,\mathbf{s}, \mathbf{a})\| < \bar{L}. \end{aligned} \tag{3.7}$$

Moreover, there exists a policy $\boldsymbol{\pi_\theta}$ *such that* $J(\boldsymbol{\theta})$ *is finite.*

Assumption 1 is a standard assumption that is made in [6] in order to derive policy gradients. All derivatives are also bounded for a smooth enough p, such as the Gaussian distribution. Moreover, one can select the initial state distribution from a bounded probability function. The existence of a policy that makes the performance $J(\boldsymbol{\theta})$ finite can be interpreted as a controllability assumption in the control literature. Policy gradient methods usually solve (3.6) using gradient descent method, i.e., at each iteration k, we update $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}, \tag{3.8}$$

where α is a positive step size.

[1] Based on the notations introduced in chapter 1, the performance function associated with the policy $\boldsymbol{\pi_\theta}$ is $J(\boldsymbol{\pi_\theta})$, but in this chapter, we denote it by $J(\boldsymbol{\theta})$ for the simplicity.

Theorem 3.2.1 (Deterministic Policy Gradient) Suppose that the MDP satisfies Assumption 1; then $\nabla_{\mathbf{a}} Q^{\pi_{\theta}}$ exists and the deterministic policy gradient reads as:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}} \left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \nabla_{\mathbf{a}} Q^{\pi_{\theta}}(\mathbf{s}, \mathbf{a}) \big|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})} \right]. \tag{3.9}$$

Proof. See in [6]. ■

Note that in (3.9), one can replace $\nabla_{\mathbf{a}} Q^{\pi_{\theta}}$ by $\nabla_{\mathbf{a}} A^{\pi_{\theta}}$ where $A^{\pi_{\theta}}(\mathbf{s}, \mathbf{a}) = Q^{\pi_{\theta}}(\mathbf{s}, \mathbf{a}) - V^{\pi_{\theta}}(\mathbf{s})$ is the advantage function associated to $\boldsymbol{\pi}_{\boldsymbol{\theta}}$.

Under some conditions detailed in [6], the action-value function $Q^{\pi_{\theta}}$ in (3.9) can be replaced by an approximation $Q^{\mathbf{w}}$ without affecting the policy gradient. Such an approximation is labelled *compatible* and can, e.g., take the form:

$$Q^{\mathbf{w}}(\mathbf{s}, \mathbf{a}) = (\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}))^{\top} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^{\top} \mathbf{w} + V^{\mathbf{v}}(\mathbf{s}), \tag{3.10}$$

where $\mathbf{w}$ is a parameters vector estimating the action-value function and $V^{\mathbf{v}} \approx V^{\pi_{\theta}}$ is a baseline function approximating the value function, which can, e.g., take a linear form:

$$V^{\mathbf{v}}(\mathbf{s}) = \Phi(\mathbf{s})^{\top} \mathbf{v}, \tag{3.11}$$

where Φ is a state feature vector and $\mathbf{v}$ is the corresponding parameters vector. The parameters $\mathbf{w}$ and $\mathbf{v}$ of the action-value function approximation (3.10) ought to be the solution of the Least Squares problem:

$$\min_{\mathbf{w},\mathbf{v}} \mathbb{E} \left[\left(Q^{\pi_{\theta}}(\mathbf{s}, \mathbf{a}) - Q^{\mathbf{w}}(\mathbf{s}, \mathbf{a}) \right)^{2} \right]. \tag{3.12}$$

For instance, problem (3.12) can be tackled via Least Squares Temporal Difference (LSTD) [7].

■ **Example 3.1** **Continue Example 2.1:** We use deterministic policy gradient for example 2.1. Note that we use a parameterized MPC scheme in order to generate the parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. We will further detail this technique in chapter 4. Moreover, we use a linear value approximator with a quadratic state feature vector [8]. ■

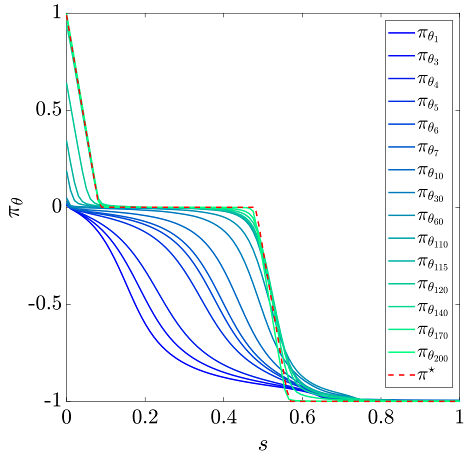


Figure 3.1: The policy improvement using deterministic policy gradient method.

Unfortunately, the convergence rate of classical gradient descent is limited, especially when the Hessian of closed-loop performance J is far from a scalar multiple of the Identity matrix [9]. In [10], the global convergence of policy gradient methods have been investigated for the Linear Quadratic Regulator (LQR) problems. Various studies propose to use the Hessian of the policy performance in Newton-type methods in order to deliver a faster learning [11].

Natural policy gradient methods have attracted much attention in the RL community recently due to their capability for better convergence [12]. The efficiency of the natural policy gradient in RL was shown in [13]. The natural policy gradient methods use the *Fisher information matrix* as an approximate Hessian [14]. In [15], a natural policy gradient method is developed for Constrained MDPs. A Quasi-Newton method is developed in [16] for Temporal Difference (TD) learning in order to get faster convergence. Natural Actor-critic has been investigated in [17]. Although the Fisher information matrix, as an approximation for the Hessian, is positive definite, it does not asymptotically converge to the exact Hessian necessarily, when the policy converges to the optimal policy [12]. As a result, the rate of convergence of the natural policy gradient method is linear, i.e., the same as the regular gradient descent [11]. Therefore, providing an approximation of the Hessian (without imposing heavy computation) that converges to the exact Hessian at the optimal policy can improve the convergence rate.

In the next section, we present a model-free approximation for the Hessian of the performance of deterministic policies to use in the context of RL based on Quasi-Newton steps in the policy parameters.

3.2.1 Quasi-Newton Iteration in Policy Gradient

In this section, we first derive a formulation for the exact Hessian of deterministic policy performance with respect to the parameters. Then we provide a model-free approximation for the Hessian of the performance function J. We show that the approximate Hessian converges to the exact Hessian at the optimal policy when the parameterized policy is rich. As a result, it gives a superlinear convergence using a Quasi-Newton optimization. This section is based on the developments of paper [18].

The next standard assumption will be made to ensure the existence of the Hessian of the policy with respect to the policy parameters $\boldsymbol{\theta}$ and the Hessian of action-value function with respect to the input $\mathbf{a}$.

Assumption 2. $\nabla^2_{\mathbf{a}} p(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})$, $\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$, $\nabla^2_{\mathbf{a}} L(\mathbf{s}, \mathbf{a})$, *are continuous in all parameters and variables* $\mathbf{s}$, $\mathbf{a}$, $\mathbf{s}'$, $\boldsymbol{\theta}$. *Moreover, there exists* M *such that:*

$$\sup_{\mathbf{a},\mathbf{s},\mathbf{s}'} \|\nabla^2_{\mathbf{a}} p(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})\| < M, \quad \sup_{\mathbf{a},\mathbf{s}} \|\nabla^2_{\mathbf{a}} L(\mathbf{s}, \mathbf{a})\| < M. \tag{3.13}$$

Similar to the assumption 1, assumption 2 is made to derive the Hessian of the performance. In practice, the assumption is satisfied for a smooth enough transition p, policy $\boldsymbol{\pi}$, and stage cost L. In the following, we provide the exact Hessian of the deterministic policy performance with respect to the policy parameters.

Definition 1. *In this paper, we use the operation* $\otimes : \mathbb{R}^{n_1 \times n_2 \times n_3} \times \mathbb{R}^{n_3} \to \mathbb{R}^{n_1 \times n_2}$ *for the product of a tensor* T *and a vector* $\mathbf{v}$*, such that:*

$$T \otimes \mathbf{v} \triangleq \sum_{i=1}^{n_3} v_i T_{(:,:,i)}, \tag{3.14}$$

where scalar v_i is the i^{th} element of vector $\mathbf{v}$ and matrix $[T_{(:,:,i)}]_{n_1\times n_2}$ is the i^{th} frontal slice of tensor T [19].

Theorem 3.2.2 (Deterministic Policy Hessian) Under Assumptions 1 and 2, $\nabla^2_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}$ and the deterministic policy Hessian exist. The latter is given by:

$$\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = H(\boldsymbol{\theta}) + \gamma\Lambda(\boldsymbol{\theta}), \tag{3.15}$$

where $H(\boldsymbol{\theta})$ and $\Lambda(\boldsymbol{\theta})$ are defined as follows:

$$\begin{aligned} H(\boldsymbol{\theta}) \triangleq \mathbb{E}_{\mathrm{s}}\Big[&\nabla^2_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a})\Big|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} + && \text{(3.16a)}\\ &\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\nabla^2_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a})\Big|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top\Big], \\ \Lambda(\boldsymbol{\theta}) \triangleq \mathbb{E}_{\mathrm{s}}\Big[&\int \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'\,|\,\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\nabla_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(\mathrm{s}')^\top \mathrm{ds}' + && \text{(3.16b)}\\ &\int \nabla_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(\mathrm{s}')\nabla_{\boldsymbol{\theta}} p(\mathrm{s}'\,|\,\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{ds}'\Big]. \end{aligned}$$

Proof. See Appendix. ■

The terms in (3.16a) only depend on the policy and the action-value function, but the terms in (3.16b) depend on the gradient of the transition probability $p(\mathrm{s}'\,|\,\mathrm{s},\mathbf{a})$, which is difficult to calculate directly from data. Hence, we use $H(\boldsymbol{\theta})$ as a model-free approximator of the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J$. Next section, we will show that the approximate Hessian $H(\boldsymbol{\theta})$ converges to the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J$ at the optimal policy.

Remark 1. *Note that one can approximate $p(\mathrm{s}'\,|\,\mathrm{s},\mathbf{a})$ from observed data in order to obtain a more accurate Hessian, e.g., using system identification techniques [20]. Such an estimation can require a heavy computation if the state-action space of the problem is not small. Hence, in order to provide a model-free approximator and for sake of brevity we ignore such evaluation in this paper.*

Quasi-Newton methods are alternatives to Newton's approach where the Hessian of the cost function is unavailable or too expensive to compute at every iteration. A Quasi-Newton update rule for the optimization problem (3.6) can be written as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha H^{-1}(\boldsymbol{\theta}_k)\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}, \tag{3.17}$$

where H is an approximation of Hessian of the performance function J. Note that using a Hessian in the policy optimization is advantageous when the different parameters would require very different step sizes in a first-order method, i.e., when $\nabla^2 J$ is far from being a multiple of the identity matrix. This is often the case in practice unless a pre-scaling is performed on the policy formulation. From the computational viewpoint, the Hessian of a policy is usually dense, and it can be troublesome to use in (3.17) for a policy parametrization using a very large number of parameters. Hence the proposed second-order method is arguably best for policies using a few dozen, up to a few hundred of parameters. E.g., policy parametrizations based on model predictive control techniques fall in that range of parameters [21]. Next mild assumptions are made to allow one to use the Newton-type optimization in the policy gradient methods.

Assumption 3. *1. The parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is rich enough. I.e., there exists $\boldsymbol{\theta}^\star$ such that $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathrm{s}) = \boldsymbol{\pi}^\star(\mathrm{s})$.*

2. $J(\boldsymbol{\theta})$ has a Lipschitz continuous Hessian and $\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta})^{-1}$ exists in a neighbourhood of $\boldsymbol{\theta}^\star$.

The first statement of Assumption 3 is a standard assumption in the theoretical developments associated to the policy gradient method. For instance, for a Linear dynamic with Quadratic cost, a policy in the form of $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) = \Theta_1 \mathrm{s} + \Theta_2$ with proper matrix dimension Θ_1 and Θ_2 satisfies Assumption 3.1, where $\boldsymbol{\theta} = \{\Theta_1, \Theta_2\}$. In practice, for a general problem such assumption is satisfied approximately by choosing a generic function approximator for the deterministic policy, e.g., Deep Neural Networks [22] and Fuzzy Neural Networks [23]. Then a richer policy satisfies the assumption asymptotically. A key consequence of this assumption is that the optimal policy $\boldsymbol{\pi}^\star$ is independent of the distribution of the initial state $p_1(\mathrm{s}_0)$. The second statement guarantees the continuity of the Hessian and allows one to use a Quasi-Newton approach.

Lemma 1. *Assume that $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a bounded, continuous function of $\mathbf{x} \in \mathbb{R}^n$ and for any probability density $g(\mathbf{x})$, we have $\mathbb{E}_{\mathbf{x}\sim g}[\mathbf{f}(\mathbf{x})] = \mathbf{0}$. Then $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ holds almost everywhere in Lebesgue measure.*

Proof. If $\mathbf{f}(\mathbf{x}) \neq \mathbf{0}$ holds on a measurable set, then there exists a probability density $\tilde{g}$ on that set such that $\mathbb{E}_{\mathbf{x}\sim\tilde{g}}[\mathbf{f}(\mathbf{x})] \neq 0$ ■

Theorem 3.2.3 Under Assumptions 1-3, the approximate Hessian $H(\boldsymbol{\theta})$ converges to the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ at the optimal policy, i.e.,

$$\lim_{\boldsymbol{\theta}\to\boldsymbol{\theta}^\star} \Lambda(\boldsymbol{\theta}) = 0. \tag{3.18}$$

Proof. The initial distribution $p_1(\mathrm{s}_0)$ is independent of the policy parameters $\boldsymbol{\theta}$. From the optimality condition of (3.5), we have:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathrm{s}_0}[V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}_0)] = \mathbb{E}_{\mathrm{s}_0}[\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}_0)] = 0$$

at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$ for any initial distribution $p_1(\mathrm{s}_0)$ (Assumption 3.1). Using Lemma 1, it implies

$$\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) \equiv 0 \tag{3.19}$$

at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. Under Assumptions 3 and for any bounded $\nabla_{\boldsymbol{\theta}} p$, it reads:

$$\begin{aligned} &\int \nabla_{\boldsymbol{\theta}} p(\mathrm{s}' \,|\, \mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')^\top \mathrm{d}\mathrm{s}' = \\ &\int \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla_{\boldsymbol{\theta}} p(\mathrm{s}' \,|\, \mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{d}\mathrm{s}' = 0 \end{aligned} \tag{3.20}$$

at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. Then, from the continuity of the Hessian (Assumption 3.2) and (3.16b), it implies (3.18). Note that Assumption 1 guarantees the boundedness of $\nabla_{\boldsymbol{\theta}} p$. ■

Next theorem provides necessary and sufficient conditions for the superlinear[2] convergence of the Quasi-Newton method.

[2] The sequence x_k is said to converge superlinearly to x if $\lim_{k\to\infty} \frac{|x_{k+1}-x|}{|x_k-x|} = 0$.

Theorem 3.2.4 (superlinear convergence of Quasi-Newton methods) Suppose that $f:\mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Consider the iteration $x_{k+1} = x_k - B_k^{-1}\nabla f_k$. Let us assume that $\{x_k\}$ converges to a point such that $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite. Then $\{x_k\}$ converges superlinearly to $x^\star$ if and only if:

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^\star))B_k^{-1}\nabla f_k\|}{\|B_k^{-1}\nabla f_k\|} = 0. \tag{3.21}$$

Proof. See Theorem 3.7 in [9]. ■

The next corollary concludes that the proposed Hessian implies a superlinear convergence.

Corollary 3.2.5 (From theorem 3.2.3 and 3.2.4): Under Assumption 3 and the assumptions in the theorem 3.2.4, the policy parameters $\boldsymbol{\theta}_k$ converge to the optimal policy parameters $\boldsymbol{\theta}^\star$ superlinearly when $H(\boldsymbol{\theta})$ defined in (3.16a) is an approximator of the exact Hessian (3.15) with $J(\boldsymbol{\theta})$ defined in (3.5) and the Quasi-Newton update rule (3.17) is used.

Natural policy gradient utilizes the Fisher information matrix as its approximate Hessian in the policy gradient method. The Fisher matrix for deterministic policies can be written as follows [24]:

$$F(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}}\left[\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^\top\right]. \tag{3.22}$$

The following corollary connects our proposed Hessian with the Fisher Information matrix.

Corollary 1. *Fisher Information matrix, defined in (3.22), is positive definite and by comparison with (3.16a) and this matrix can be written equal to (3.16a) under the following conditions:*

1. $\nabla^2_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} = I$,
2. $\nabla^2_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \otimes \nabla_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} = 0$.

Then clearly $F(\boldsymbol{\theta})$ does not converge to the exact Hessian at the optimal policy necessarily. I.e., the parameters will not converge superlinearly to the optimal parameters if the Fisher information matrix is used as a Hessian approximation (see Theorem 3.2.4).

Remark 2. *Under assumptions 1-3, $H(\boldsymbol{\theta})$ is positive definite in a neighborhood of $\boldsymbol{\theta}^\star$. Nevertheless $H(\boldsymbol{\theta})$ is not necessarily positive definite for a parameter $\boldsymbol{\theta}$ that is far from the optimal parameter $\boldsymbol{\theta}^\star$ because of the term $\nabla^2_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \otimes \nabla_{\mathbf{a}} Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}}$, while the Fisher information matrix $F(\boldsymbol{\theta})$ is (semi) positive definite by construction. A regularization of H may be needed in practice, and one can use the Fisher information matrix F to regularize the approximate Hessian H, when H is not positive definite. This regularization can be applied using a Hessian in the form of $H + \beta F$ at every step, where $\beta \geq 0$ is a constant that must be ideally selected at every step. However, other methods e.g., trust-region methods can effectively take advantage of indefinite Hessian approximations.*

Remark 3. *Many RL methods deliver a sequence of parameters $\boldsymbol{\theta}_k$ that is stochastic by nature because they are based on measurements taken from a stochastic system. From the theoretical viewpoint, all of the results in this paper are valid for large data sets, where sample averages converge to the true expectations. However, in practice, one can use the method to improve the stochastic convergence rate and derive an extension of the current theorems.*

■ **Example 3.2 An Analytical Example:** We consider a simple Linear Quadratic Regulator (LQR) problem in order to verify the method analytically. Consider the following scalar linear dynamics:

$$s^+ = s + a + w, \tag{3.23}$$

where $w \sim \mathcal{N}(0, \sigma^2)$, i.i.d., $\mathbb{E}_w[wa] = 0$ and $\mathbb{E}_w[ws] = 0$. Transition probability of the MDP (3.23) reads as follows:

$$p(s'|s, a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{(s' - s - a)^2}{2\sigma^2}\Big). \tag{3.24}$$

Initial state distribution is $p_1(s_0) \sim \mathcal{N}(0, \sigma_0^2)$, deterministic policy reads as $\pi_\theta = -\theta s$ and stage cost is $L(s, a) = 0.5(s^2 + a^2)$. We assume value function in the from of $V^{\pi_\theta}(s) = p_\theta s^2 + q_\theta$ and we show it will satisfy the fundamental Bellman equations (3.4), then we have:

$$\begin{aligned} V^{\pi_\theta}(s) &= L(s, \pi_\theta(s)) + \gamma \mathbb{E}_w[V^{\pi_\theta}(s - \theta s + w)] \\ &= 0.5s^2(1 + \theta^2) + \gamma(1 - \theta)^2 p_\theta s^2 + \gamma p_\theta \sigma^2 + \gamma q_\theta. \end{aligned} \tag{3.25}$$

It implies:

$$p_\theta = \frac{0.5(1 + \theta^2)}{1 - \gamma(1 - \theta)^2}, \qquad q_\theta = \frac{\gamma\sigma^2}{1 - \gamma} p_\theta. \tag{3.26}$$

Using the Bellman equations (3.4), the action-value function $Q^{\pi_\theta}(s, a)$ can be evaluated as follows:

$$\begin{aligned} Q^{\pi_\theta}(s, a) &= L(s, a) + \gamma \mathbb{E}\left[V^{\pi_\theta}(s^+|s, a)\right] = 0.5(s^2 + a^2) + \gamma \mathbb{E}[p_\theta(s + a + w)^2 + q_\theta] \\ &= (0.5 + \gamma p_\theta)s^2 + 2\gamma p_\theta sa + (0.5 + \gamma p_\theta)a^2 + q_\theta. \end{aligned} \tag{3.27}$$

One can check the identity $V^{\pi_\theta}(s) = Q^{\pi_\theta}(s, \pi(s))$. Then:

$$\nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta} = \frac{\gamma\theta^2 + \theta - \gamma}{1 - \gamma(1 - \theta)^2} s^2 \tag{3.28a}$$

$$\nabla_\theta \pi_\theta \nabla_a^2 Q^{\pi_\theta}(s, a)|_{a=\pi_\theta} \nabla_\theta \pi_\theta = s^2(1 + 2\gamma p_\theta). \tag{3.28b}$$

Note that $\nabla_\theta^2 \pi_\theta = 0$. The closed-loop performance J reads:

$$J(\theta) = \mathbb{E}_{s_0}[V^{\pi_\theta}(s_0)] = \frac{0.5(1 + \theta^2)}{1 - \gamma(1 - \theta)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1 - \gamma}). \tag{3.29}$$

Then, by taking derivation of J with respect to the parameters $\boldsymbol{\theta}$:

$$J'(\theta) = \frac{\gamma\theta^2 + \theta - \gamma}{(1 - \gamma(1 - \theta)^2)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1 - \gamma}). \tag{3.30}$$

From policy gradient (3.9) and (3.28a), we can write:

$$J'(\theta) = \mathbb{E}_s[\nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta}] = \mathbb{E}_s[\frac{\gamma\theta^2 + \theta - \gamma}{1 - \gamma(1 - \theta)^2} s^2]. \tag{3.31}$$

Then (3.30) and (3.31) imply:

$$\mathbb{E}_s[s^2] = \frac{(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma})}{1-\gamma(1-\theta)^2}. \tag{3.32}$$

From (3.29), the exact Hessian of the performance J reads:

$$J''(\theta) = p''_\theta(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}) = \frac{-2\gamma^2\theta^3 - 3\gamma\theta^2 + 6\gamma^2\theta - 4\gamma^2 + \gamma - 1}{(1-\gamma(1-\theta)^2)^3}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{3.33}$$

From (3.16a) and (3.28b), the approximate Hessian $H(\theta)$ reads:

$$H(\theta) = \mathbb{E}_s[s^2(1+2\gamma p_\theta)] = \frac{1+2\gamma\theta}{(1-\gamma(1-\theta)^2)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{3.34}$$

From (3.16b) and (3.24), we can write:

$$\begin{aligned}\Lambda(\theta) =& 2\int_{\mathcal{S}} \nabla_\theta V^{\pi_\theta}(s')\nabla_\theta p(s'|s,\pi_\theta)\mathrm{d}s' \qquad (3.35)\\ =& 2\int_{-\infty}^{\infty} -p'_\theta((s')^2 + \frac{\gamma\sigma^2}{1-\gamma})\frac{s(s'-s+\theta s)}{\sqrt{2\pi}\sigma^3}\exp\Big(-\frac{(s'-s+\theta s)^2}{2\sigma^2}\Big)\mathrm{d}s'\\ =& -4p'_\theta s^2(1-\theta) = \frac{-4(\gamma\theta^2+\theta-\gamma)(1-\theta)}{(1-\gamma(1-\theta)^2)^3}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}).\end{aligned}$$

Therefore, one can easily verify (3.15) by substitution (3.33), (3.34) and (3.35) in (3.15). Note that we used the following integration in (3.35):

$$\int_{-\infty}^{\infty}(x^2+a)(x-b)\exp(-c(x-b)^2)\mathrm{d}x = \frac{\sqrt{\pi}b}{c^{\frac{3}{2}}}, \tag{3.36}$$

where a, b and $c > 0$ are constraints. Fig. 3.2 (right) compares the exact Hessian $\nabla_\theta^2 J(\theta)$, the proposed approximate Hessian $H(\theta)$ and the Fisher matrix $F(\theta)$ for this example with $\gamma = 0.9$ and $\sigma_0^2 = \sigma^2 = 0.1$. As can be seen, $\nabla_\theta^2 J$ meets $H(\theta)$ at the optimal parameter. Fig. 3.2 (left) shows the superlinear convergence of the policy parameters during the learning using the Quasi-Newton policy gradient method, while the (first order) policy gradient method and natural policy gradient method results in a linear convergence during the learning. ■

■ **Example 3.3 A Numerical Simulation:** Cart-Pendulum balancing is a well-known benchmark in the RL community. The dynamics of a cart-pendulum system, shown in fig. 3.3, reads as:

$$(M+m)\ddot{x} + \frac{1}{2}ml\ddot{\phi}\cos\phi = \frac{1}{2}ml\dot{\phi}^2\sin\phi + u, \tag{3.37a}$$

$$\frac{1}{3}ml^2\ddot{\phi} + \frac{1}{2}ml\ddot{x}\cos\phi = -\frac{1}{2}mgl\sin\phi, \tag{3.37b}$$

where M and m are the cart mass and pendulum mass, respectively, l is the pendulum length and ϕ is its angle from the vertical axis. Force u is the control input, x is the cart displacement and g is gravity. We used the Runge-Kutta 4^{th}-order method to discretize (3.37) with a sampling time $\mathrm{d}t = 0.1\mathrm{s}$ and cast it in the form of $\mathbf{s}^+ = \mathbf{f}(\mathbf{s},\mathbf{a}) + \boldsymbol{\xi}$, where $\mathbf{s} = [\dot{x}, x, \dot{\phi}, \phi]^\top$ is the state, $\mathbf{a} = u$ is the input, $\boldsymbol{\xi}$ is a Gaussian noise and $\mathbf{f}$ is a nonlinear function representing (3.37) in discrete time. A stabilizing quadratic stage cost is considered as $L(\mathbf{s},\mathbf{a}) = \mathbf{s}^\top\mathbf{s} + 0.01\mathbf{a}^\top\mathbf{a}$, and the deterministic policy is considered in the form of $\boldsymbol{\pi_\theta} = -\boldsymbol{\theta}\mathbf{s}$. Fig. 3.4 (right) shows the closed-loop performance J using the proposed Hessian $H(\boldsymbol{\theta})$ (green) and the natural policy gradient method (red). Moreover, the deterministic policy parameters $\boldsymbol{\theta}$ are shown in fig. 3.4 (left). ■

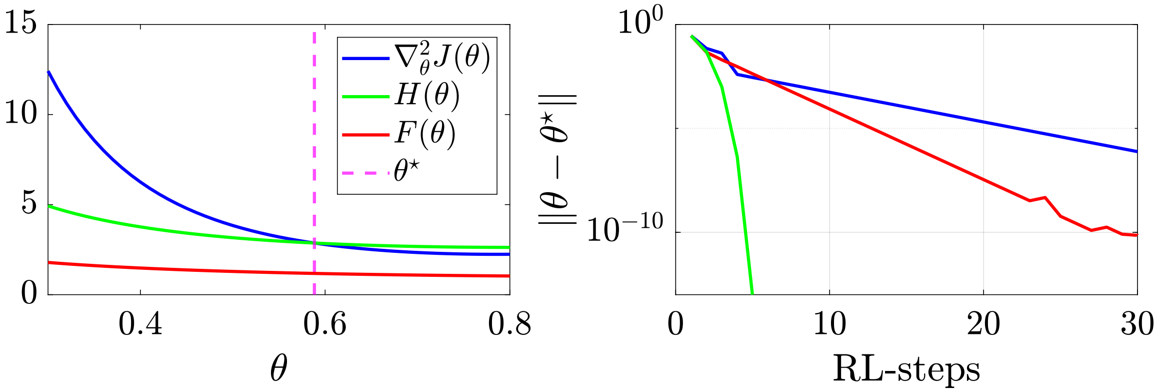


Figure 3.2: Right: Superlinear convergence of the proposed method. blue: policy gradient method, red: natural policy gradient method, green: proposed method. Left: Comparison of the exact Hessian $\nabla^2_\theta J(\theta)$, the proposed approximate Hessian $H(\theta)$, and the Fisher matrix $F(\theta)$.

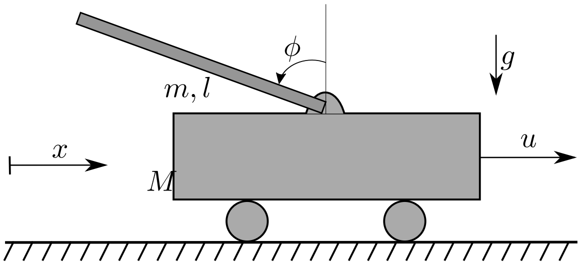


Figure 3.3: The cart-pendulum system. We use $M = 0.5\text{kg}$, $m = 0.2\text{kg}$, $l = 0.3\text{m}$ and $g = 9.8\text{m/s}^2$ for the simulation.

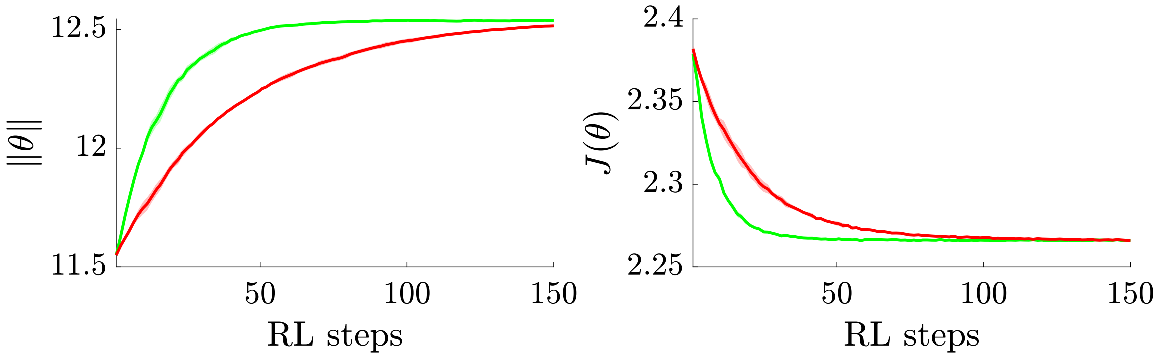


Figure 3.4: Right: Closed-loop performance $J(\boldsymbol{\theta})$; Left: Convergence of the policy parameters $\boldsymbol{\theta}$ using the proposed Hessian (green) and natural policy gradient method (red).

3.3 Summary

The chapter detailed Q-learning and policy gradient methods as popular methods in the context of RL. Moreover, we provided a Hessian approximation for the performance of deterministic policies. We use the model-independent terms of the exact Hessian as an approximate Hessian, and we showed that the resulting approximate Hessian converges to the exact Hessian at the optimal policy. Therefore, the approximate Hessian can be used in the Quasi-Newton optimization to provide a superlinear convergence. We analytically verified our formulation in a simple example, and we compare our method with the natural policy gradient in a cart-pendulum system.

3.4 Appendix: Proof of Theorem 3.2.2

Proof. We first calculate the Hessian of $V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s})$ as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) =& \nabla^2_{\boldsymbol{\theta}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} = \nabla^2_{\boldsymbol{\theta}} \left(L(\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' \right) \\ =& \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} L(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \nabla^2_{\mathbf{a}} L(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top \\ &+ \nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\mathbf{a}) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' \end{aligned} \tag{3.38}$$

The third term can be calculated as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\mathbf{a}) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' =& \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla^2_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\mathrm{ds}' \\ + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')^\top \mathrm{ds}' &+ \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{ds}' \\ + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' & \end{aligned} \tag{3.39}$$

The first term can be extended as follows:

$$\begin{aligned}\int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla^2_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\mathrm{ds}' =& \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} p(\mathrm{s}'|\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})}\mathrm{ds}' + \\ + \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \nabla^2_{\mathbf{a}} p(\mathrm{s}'|\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top \mathrm{ds}' & \end{aligned} \tag{3.40}$$

By rearranging (3.38), we can write:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) =& \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} \left(L(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}'\right) \\ &+ \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \nabla^2_{\mathbf{a}} \left(L(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}'\right) \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top \\ &+ \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')^\top \mathrm{ds}' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{ds}' \\ &+ \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' = \mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}) + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' \end{aligned} \tag{3.41}$$

where $\mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s})$ is defined as follows:

$$\begin{aligned}\mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}) \triangleq& \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \nabla^2_{\mathbf{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top + \\ & \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')^\top \mathrm{ds}' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{ds}' \end{aligned} \tag{3.42}$$

where we used:

$$Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s},\mathbf{a}) = L(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}' \tag{3.43}$$

Now, we can go one step further for the last term of (3.41):

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) =& \mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}) + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}')\mathrm{ds}' + \\ & \int_{\mathcal{S}}\int_{\mathcal{S}} \gamma^2 p(\mathrm{s}'|\mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))p(\mathrm{s}''|\mathrm{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}'))\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}'')\mathrm{ds}'\mathrm{ds}''\end{aligned} \tag{3.44}$$

where we have used the following equality:

$$\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}') = \mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}') + \int_{\mathcal{S}} \gamma p(\mathrm{s}''|\mathrm{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}'))\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}'')\mathrm{ds}'' \tag{3.45}$$

We can define:

$$p(\mathrm{s} \to \mathrm{s}'', 2, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = \int_{\mathcal{S}} p(\mathrm{s}'|\mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))p(\mathrm{s}''|\mathrm{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}'))\mathrm{ds}'$$

and interpret it probability of transition from s to s'' in 2 steps by policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. Then in last term we can alter integral notation $\mathrm{s}'' \to \mathrm{s}'$ and rewrite (3.44) as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) = \mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}) + \int_{\mathcal{S}} \gamma p(\mathrm{s}'|\mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}')\mathrm{ds}' + \\ \int_{\mathcal{S}} \gamma^2 p(\mathrm{s} \to \mathrm{s}', 2, \boldsymbol{\pi}_{\boldsymbol{\theta}})\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}')\mathrm{ds}'\end{aligned} \tag{3.46}$$

By continuing this procedure, we have:

$$\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}) = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p(\mathrm{s} \to \mathrm{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}})\mathcal{F}_{\boldsymbol{\theta}}(\mathrm{s}')\mathrm{ds}' \tag{3.47}$$

where

$$p(\mathrm{s} \to \mathrm{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = \int_{\mathcal{S}} p(\mathrm{s} \to \hat{\mathrm{s}}, t-1, \boldsymbol{\pi}_{\boldsymbol{\theta}})p(\mathrm{s}'|\hat{\mathrm{s}}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\hat{\mathrm{s}}))\mathrm{d}\hat{\mathrm{s}}$$

starting from $p(\mathrm{s} \to \mathrm{s}', 1, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = p(\mathrm{s}'|\mathrm{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))$. Then, taking the expectation over p_1 for Hessian of policy we have:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} p_1(\mathrm{s})V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s})\mathrm{ds} = \int_{\mathcal{S}} p_1(\mathrm{s})\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s})\mathrm{ds} = \int_{\mathcal{S}}\int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p_1(\mathrm{s}) \\ p(\mathrm{s} \to \mathrm{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}}) \Bigg[\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}') \otimes \nabla_{\mathbf{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}', \mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}')} + \\ \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}')\nabla^2_{\mathbf{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}', \mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}')}\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}')^\top + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\mathrm{s}''|\mathrm{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}')) \\ \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}'')^\top \mathrm{ds}'' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathrm{s}'')\nabla_{\boldsymbol{\theta}} p(\mathrm{s}''|\mathrm{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}'))^\top \mathrm{ds}'' \Bigg] \mathrm{ds}'\mathrm{ds}\end{aligned}$$

Or equivalently:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}}\Big[&\nabla^2_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}) \otimes \nabla_{\mathbf{a}} Q^{\boldsymbol{\pi}_\theta}(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} + \nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})\nabla^2_{\mathbf{a}} Q^{\boldsymbol{\pi}_\theta}(\mathrm{s},\mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}}\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s})^\top \\ &+ \int \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_\theta}(\mathrm{s}')\nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))^\top \mathrm{ds}' + \int \gamma \nabla_{\boldsymbol{\theta}} p(\mathrm{s}'|\mathrm{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathrm{s}))\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_\theta}(\mathrm{s}')^\top \mathrm{ds}'\Big]\end{aligned} \tag{3.48}$$

where $\mathbb{E}_{\mathbf{s}}[\cdot]$ is taken over discounted state distribution of the Markov chain in closed-loop with policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. ■

4. MPC based RL

head1.png

In general, we may not have full knowledge of the probability transition of the real MDP (1.1). One then typically considers an imperfect model of the real MDP (1.1), having the state transition:

$$\mathbf{s}_+ \sim \hat{p}\left(\cdot \,|\, \mathbf{s}, \mathbf{a}\right). \tag{4.1}$$

In this chapter, we will investigate the idea of modifying a stage cost and terminal cost of an undiscounted finite-horizon value function in order to capture the optimal policy and optimal value function of the real MDP even if the wrong model, described by (4.1) is used.

4.1 Equivalency statements

We are interested in capturing the optimal policy and value functions of a given (possibly discounted) MDP using undiscounted settings because it has been shown that proving closed-loop stability of the Markov Chains with the optimal policy resulting from an undiscounted Optimal Control Problem (OCP) is more straightforward than a discounted setting [25, 26]. This observation is also well-known in the context of Model Predictive Control (MPC) [27] where the closed-loop stability of an optimal policy resulting from discounted MPC is challenging. For a discounted finite-horizon problem, it is shown in [28] that even if the provided stage cost, terminal cost, and terminal set satisfy the stability requirements, the closed-loop might be unstable for some discount factors. Indeed, the discount factor has a critical role in the stability of the closed-loop system under the optimal policy of the discounted cost. The conditions for the asymptotic stability for discounted optimal control problems have been recently developed in [29] for deterministic systems with the exact model. Therefore, an undiscounted MPC scheme is more desirable, where the closed-loop stability analysis is straightforward and well-developed [30].

4.1.1 Finite-horizon OCP

Consider the following undiscounted finite-horizon OCP associated with model (4.1):

$$\hat{V}_N^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \hat{V}_N^{\boldsymbol{\pi}}(\mathbf{s}) :=\mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\hat{T}(\mathbf{s}_N) + \sum_{k=0}^{N-1} \hat{L}(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right))\right], \quad \mathbf{s}_{k+1} \sim \hat{p}\left(\cdot \,|\, \mathbf{s}_k, \mathbf{a}_k\right) , \quad \mathbf{s}_0 = \mathbf{s} , \tag{4.2}$$

where N is the horizon length, $\hat{T}$, $\hat{L}$, $\hat{V}_N^\star$ and $\hat{V}_N^{\boldsymbol{\pi}}$ are the terminal cost, the stage cost, the optimal value function, and the value function of the policy $\boldsymbol{\pi}$ associated to model (4.1), respectively. The expectation $\mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}$ in (4.2) is taken over undiscounted closed-loop Markov Chain (4.1) with policy $\boldsymbol{\pi}$. We denote $\hat{\boldsymbol{\pi}}_N^\star$ the optimal policy resulting from (4.2). Moreover, the action-value function $\hat{Q}_N^\star$ associated to (4.2) is defined as follows:

$$\hat{Q}_N^\star(\mathbf{s}, \mathbf{a}) := \hat{L}(\mathbf{s}, \mathbf{a}) + \mathbb{E}_{\hat{p}}\left[\hat{V}_{N-1}^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right] , \tag{4.3a}$$

$$\hat{V}_0^\star(\mathbf{s}) := \hat{T}(\mathbf{s}) . \tag{4.3b}$$

The next assumption expresses a requirement on the optimal model trajectories $\mathbf{s}_{0,1,\ldots}^\star$ with the optimal policy $\boldsymbol{\pi}^\star$ which allows us to develop the theoretical results of this section.

Assumption 4. *The set*

$$\mathcal{S} =: \left\{\mathbf{s} \in S \;\middle|\; |\mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}^\star}}\left[V^\star(\mathbf{s}_k^\star)\right]| < \infty,\, \forall\, k \le \bar{N}\right\} \tag{4.4}$$

is non-empty for a given $\bar{N} \in \mathbb{N}$.

Assumption 4 requires that there exists a non-empty set $\mathcal{S}$ such that for all trajectories starting in it, the expected value of $V^\star$ is bounded at all future times under the state distribution given by the model. This assumption plays a vital role in the derivation of our main result.

The next theorem provides theoretical support to the idea that one can recover the optimal policy and value functions by means of an MPC scheme which is based on an imperfect model and has an undiscounted formulation over a finite prediction horizon [31].

Theorem 4.1.1 Suppose that Assumption 4 holds for $\bar{N} \ge N$. Then, there exist a terminal cost $\hat{T}$ and a stage cost $\hat{L}$ such that the following identities hold, $\forall \gamma, N \in \mathbb{N}, \mathbf{s} \in \mathcal{S}$:

(i) $\hat{V}_N^\star(\mathbf{s}) = V^\star(\mathbf{s})$,

(ii) $\hat{\boldsymbol{\pi}}_N^\star(\mathbf{s}) = \boldsymbol{\pi}^\star(\mathbf{s})$,

(iii) $\hat{Q}_N^\star(\mathbf{s}, \mathbf{a}) = Q^\star(\mathbf{s}, \mathbf{a})$, for the inputs $\mathbf{a} \in A$ such that $|\mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right]| < \infty$

Proof. We select the terminal cost $\hat{T}$ and the stage cost $\hat{L}$ as follows:

$$\hat{T}(\mathbf{s}) = V^\star(\mathbf{s}) \tag{4.5a}$$

$$\hat{L}(\mathbf{s}, \mathbf{a}) = \begin{cases} Q^\star(\mathbf{s}, \mathbf{a}) - \mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right] & \text{If } |\mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right]| < \infty \\ \infty & \text{otherwise} \end{cases} \tag{4.5b}$$

Under Assumption 4, the terminal and stage costs in (4.2) have a finite expected value for all $\mathbf{s}_0 \in \mathcal{S}$. By substitution of (4.5) in (4.2) and using telescopic sum, we have:

$$
\begin{aligned}
\hat{V}_N^{\boldsymbol{\pi}}(\mathbf{s}) &= \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\hat{T}(\mathbf{s}_N) + \sum_{k=0}^{N-1} \hat{L}(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right))\right] \\
&\overset{(4.5)}{=} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[V^\star(\mathbf{s}_N) + \sum_{k=0}^{N-1} \Big(Q^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)) - V^\star(\mathbf{s}_{k+1})\Big)\right] \\
&= Q^\star(\mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} \left(Q^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)) - V^\star(\mathbf{s}_k)\right)\right] \\
&= Q^\star(\mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} A^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right))\right],
\end{aligned}
\tag{4.6}
$$

where $\mathbf{s}_0 = \mathbf{s}$. From (1.4) and (1.6), we know that:

$$
\boldsymbol{\pi}^\star(\cdot) = \arg\min_{\boldsymbol{\pi}} A^\star\left(\cdot, \boldsymbol{\pi}\left(\cdot\right)\right) = \arg\min_{\boldsymbol{\pi}} Q^\star\left(\cdot, \boldsymbol{\pi}\left(\cdot\right)\right) . \tag{4.7}
$$

Then from (4.6):

$$
\boldsymbol{\pi}^\star(\mathbf{s}) = \arg\min_{\boldsymbol{\pi}} \hat{V}_N^{\boldsymbol{\pi}}(\mathbf{s}) = \arg\min_{\boldsymbol{\pi}} Q^\star(\mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} A^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right))\right]
$$

Note that $\boldsymbol{\pi}^\star$ minimizes all terms in the cost above, i.e., $A^\star$ and $Q^\star$, such that is must also minimize $\hat{V}_N^{\boldsymbol{\pi}}$. This proves (ii), i.e.,

$$
\boldsymbol{\pi}^\star(\mathbf{s}) = \hat{\boldsymbol{\pi}}_N^\star(\mathbf{s}). \tag{4.8}
$$

In turn, this further proves (i), since

$$
\hat{V}_N^\star(\mathbf{s}) = \hat{V}_N^{\boldsymbol{\pi}^\star}(\mathbf{s}) = Q^\star(\mathbf{s}, \boldsymbol{\pi}^\star(\mathbf{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}^\star}}\left[\sum_{k=1}^{N} \underbrace{A^\star(\mathbf{s}_k, \boldsymbol{\pi}^\star\left(\mathbf{s}_k\right))}_{\overset{(1.6)}{=}0}\right] = Q^\star(\mathbf{s}, \boldsymbol{\pi}^\star(\mathbf{s})) \overset{(1.4)}{=} V^\star(\mathbf{s})\,.
$$

Moreover, from (4.3) and (4.5b), for any inputs $\mathbf{a} \in A$ such that $|\mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right]| < \infty$, we have:

$$
\begin{aligned}
\hat{Q}_N^\star(\mathbf{s}, \mathbf{a}) &= \hat{L}(\mathbf{s}, \mathbf{a}) + \mathbb{E}_{\hat{p}}\left[\hat{V}_{N-1}^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right] \\
&\overset{(4.5b)}{=} Q^\star(\mathbf{s}, \mathbf{a}) + \mathbb{E}_{\hat{p}}\left[\hat{V}_{N-1}^\star(\mathbf{s}^+) - V^\star(\mathbf{s}^+)|\mathbf{s}, \mathbf{a}\right] = Q^\star(\mathbf{s}, \mathbf{a}),
\end{aligned}
\tag{4.9}
$$

where the last inequality is obtained by noting that (i) also implies $V_{N-1}^\star(\mathbf{s}) = V^\star(\mathbf{s})$. This directly yields (iii). ■

Theorem 4.1.1 states that independent of the discount factor γ is it possible to find a finite-horizon OCP that provides the optimal policy and optimal value functions of a discounted MDP if an inexact model is used. We observe that the setup of this paper has been analyzed in [26], under the assumption of a perfect model, i.e., $\hat{p}(\cdot|\mathbf{s}, \mathbf{a}) = p(\cdot|\mathbf{s}, \mathbf{a})$. In that case (4.5b) reads:

$$
\hat{L}(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + (\gamma - 1)\mathbb{E}_p[V^\star(\mathbf{s}^+) \mid \mathbf{s}, \mathbf{a}], \tag{4.10}
$$

which corresponds to the cost modification discussed in [26].

4.1.2 Infinite-horizon OCP

In this section, we investigate the case $N \rightarrow \infty$ for which, under some conditions, the terminal cost can be dismissed. In this case, we first make the next additional assumption.

Assumption 5. *We assume that the optimal value function converges to a constant and finite value with model* (4.1) *under the optimal policy* $\boldsymbol{\pi}^\star$*. I.e:*

$$-\infty < \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}_N^\star)\right] = \hat{v}_\infty < \infty \tag{4.11}$$

Assumption 5 can be interpreted as some forms of the stability condition on the model dynamics under the optimal policy $\boldsymbol{\pi}^\star$.

In this section, we consider the following undiscounted value function without terminal cost:

$$\hat{V}^\star_\infty(\mathbf{s}) := \min_{\boldsymbol{\pi}} \hat{V}^{\boldsymbol{\pi}}_\infty(\mathbf{s}) := \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{N-1} \hat{L}(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right))\right] \tag{4.12}$$

with initial state $\mathbf{s}_0 = \mathbf{s}$. We denote the optimal policy solution of (4.12) as $\hat{\boldsymbol{\pi}}^\star_\infty(\mathbf{s})$. We then define the optimal action-value function $\hat{Q}^\star_\infty$ associated to (4.12) as follows:

$$\hat{Q}^\star_\infty(\mathbf{s},\mathbf{a}) = \hat{L}(\mathbf{s},\mathbf{a}) + \mathbb{E}_{\hat{p}}\left[\hat{V}^\star_\infty(\mathbf{s}^+)\,|\,\mathbf{s},\mathbf{a}\right] \,, \tag{4.13}$$

We are now ready to state the equivalent of Theorem 4.1.1 in case of an infinite horizon without a terminal cost.

Theorem 4.1.2 Suppose that Assumptions 4 and 5 hold, then the following hold $\forall \mathbf{s} \in \mathcal{S}, \forall \gamma$:
(i) $\hat{\boldsymbol{\pi}}^\star_\infty(\mathbf{s}) = \boldsymbol{\pi}^\star(\mathbf{s})$
(ii) $\hat{V}^\star_\infty(\mathbf{s}) = V^\star(\mathbf{s}) - \hat{v}_\infty$
(iii) $\hat{Q}^\star_\infty(\mathbf{s},\mathbf{a}) = Q^\star(\mathbf{s},\mathbf{a}) - \hat{v}_\infty$, for the inputs $\mathbf{a} \in A$ such that $|\mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}^+)|\mathbf{s},\mathbf{a}\right]| < \infty$,
if the stage cost $\hat{L}$ is selected according Equation (4.5b).

Proof. Using stage cost $\hat{L}$ in (4.5b), we have:

$$\begin{aligned}
\hat{V}^{\boldsymbol{\pi}}_\infty(\mathbf{s}) &= \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{N-1} Q^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)) - \mathbb{E}_{\hat{p}}\left[V^\star(\mathbf{s}_{k+1})|\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right]\right] \\
&= \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{N-1} Q^\star(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)) - V^\star(\mathbf{s}_{k+1})\right] \\
&= Q^\star(\mathbf{s},\boldsymbol{\pi}(\mathbf{s})) + \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[-V^\star(\mathbf{s}_N) + \sum_{k=1}^{N-1} Q^\star(\mathbf{s}_k,\boldsymbol{\pi}(\mathbf{s}_k)) - V^\star(\mathbf{s}_k)\right] \\
&= Q^\star(\mathbf{s},\boldsymbol{\pi}(\mathbf{s})) + \lim_{N\rightarrow\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[-V^\star(\mathbf{s}_N) + \sum_{k=1}^{N-1} A^\star(\mathbf{s}_k,\boldsymbol{\pi}(\mathbf{s}_k))\right],
\end{aligned} \tag{4.14}$$

where $\mathbf{s}_0 = \mathbf{s}$. By (1.4) and (1.6) we know that the policy $\boldsymbol{\pi}(\mathbf{s}) = \boldsymbol{\pi}^\star(\mathbf{s})$ minimizes all terms $A^\star(\cdot,\boldsymbol{\pi}(\cdot))$ and $Q^\star(\cdot,\boldsymbol{\pi}(\cdot))$, such that it also minimizes $\hat{V}^{\boldsymbol{\pi}}_\infty(\mathbf{s})$ and:

$$\hat{V}^{\boldsymbol{\pi}^\star}_\infty(\mathbf{s}) = V^\star(\mathbf{s}) - \lim_{N\rightarrow\infty} \mathbb{E}\left[V^\star(\mathbf{s}_N^\star)\right]. \tag{4.15}$$

Using (4.11) we have:

$$\hat{V}_{\infty}^{\star}(\mathbf{s}) = \hat{V}_{\infty}^{\boldsymbol{\pi}^{\star}}(\mathbf{s}) = V^{\star}(\mathbf{s}) - \hat{v}_{\infty}. \tag{4.16}$$

Moreover, for the inputs $\mathbf{a} \in A$ such that $|\mathbb{E}_{\hat{p}}\left[V^{\star}(\mathbf{s}^{+}) \,|\, \mathbf{s}, \mathbf{a}\right]| < \infty$:

$$\begin{aligned}\hat{Q}_{\infty}^{\star}(\mathbf{s}, \mathbf{a}) = \hat{L}(\mathbf{s}, \mathbf{a}) + \mathbb{E}_{\hat{p}}\left[\hat{V}_{\infty}^{\star}(\mathbf{s}^{+}) \,|\, \mathbf{s}, \mathbf{a}\right] = Q^{\star}(\mathbf{s}, \mathbf{a}) - \mathbb{E}_{\hat{p}}\left[V^{\star}(\mathbf{s}^{+}) \,|\, \mathbf{s}, \mathbf{a}\right] \qquad (4.17)\\ + \mathbb{E}_{\hat{p}}\left[\hat{V}_{\infty}^{\star}(\mathbf{s}^{+}) \,|\, \mathbf{s}, \mathbf{a}\right] = Q^{\star}(\mathbf{s}, \mathbf{a}) - \mathbb{E}_{\hat{p}}\left[V_{\infty}^{\star}(\mathbf{s}^{+}) - \hat{V}^{\star}(\mathbf{s}^{+}) \,|\, \mathbf{s}, \mathbf{a}\right] = Q^{\star}(\mathbf{s}, \mathbf{a}) - \hat{v}_{\infty},\end{aligned}$$

which completes the proof. ■

Theorem 4.1.2 extends Theorem 4.1.1 to the case of an infinite horizon with zero terminal cost and states that under the additional Assumption 5. This assumption is necessary in order to be able to remove the terminal cost.

In the next section we will detail the use of the theorems in practice and reformulate OCP (4.2) as a Model Predictive Control (MPC)-scheme.

4.2 MPC as a function approximator for RL

As it was shown in the previous section, the optimal policy and value functions of any MDP with either discounted or undiscounted criteria can be captured using a finite-horizon undiscounted OCP (4.2) even if the model is not accurate. Since the equivalence only holds at the initial state, if one is interested in recovering the optimal MDP policy, the finite-horizon OCP needs to be solved from scratch for each initial state. In practice, this amounts to deploying the finite-horizon OCP in an MPC framework, i.e., in closed-loop.

As discussed above, the equivalence is only obtained if a properly modified stage and terminal costs are introduced for the finite-horizon undiscounted MPC scheme. However finding such costs requires knowledge about the optimal value functions of the real MDP. In this section, we detail how the theorems we provided in the previous sections can be used in practice to exploit MPC as a structured function approximator of the optimal policy and value functions of the real MDP. One of the main advantages of MPC is that it allows us to straightforwardly introduce state and input constraints in the policy.

We parameterize the MPC scheme with parameter vector $\boldsymbol{\theta}$ such that RL methods can be deployed to tune $\boldsymbol{\theta}$ in order to achieve the equivalence yielding the optimal policy and value functions of the real system and, consequently, the best possible closed-loop performance. Using MPC as a function approximator of a given MDP has been first proposed and justified in [21].

As the MPC model is not required to capture the real system dynamics exactly, for the sake of reducing the computational burden, and due to the (relative) simplicity of the resulting MPC scheme, a popular choice of model $\hat{p}\,(\mathbf{s}^{+} \,|\, \mathbf{s}, \mathbf{a})$ is a deterministic model, i.e.:

$$\hat{p}\left(\mathbf{s}^{+} \,|\, \mathbf{s}, \mathbf{a}\right) = \delta\left(\mathbf{s}^{+} - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})\right) \tag{4.18}$$

where $\delta(\cdot)$ is the Dirac measure and $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$ is a parameterized deterministic (possibly nonlinear) model.

We approximate the modified costs $\hat{L}$ and $\hat{T}$ by parametric functions $L_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$, respectively. Due to the mismatch between the model and the real system, hard constraints in the MPC scheme could become infeasible. This is a well-known issue in the MPC community and one simple solution consists in formulating the state constraints as soft constraints [32]. We therefore formulate the MPC finite-horizon OCP as:

$$\begin{aligned}\hat{V}_N^{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{u},\mathbf{x},\boldsymbol{\sigma}} \quad & -\lambda_{\boldsymbol{\theta}}(\mathbf{x}_0) + T_{\boldsymbol{\theta}}(\mathbf{s}_N) + \boldsymbol{\mu}_{\mathrm{f}}^\top \boldsymbol{\sigma}_N \\ & + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\mu}^\top \boldsymbol{\sigma}_k \qquad (4.19a)\\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k), \;\; \mathbf{s}_0 = \mathbf{s}, \qquad (4.19b)\\ & \mathbf{u}_k \in A, \;\; 0 \le \boldsymbol{\sigma}_k, \;\; 0 \le \boldsymbol{\sigma}_N, \qquad (4.19c)\\ & \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k) \le \boldsymbol{\sigma}_k^\star, \;\; \mathbf{h}_{\boldsymbol{\theta}}^{\mathrm{f}}(\mathbf{x}_N) \le \boldsymbol{\sigma}_N^\star, \qquad (4.19d)\end{aligned}$$

where $\hat{V}_N^{\boldsymbol{\theta}}$ is the MPC-based parameterized value function, $\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$ is a mixed input-state constraint, $\mathbf{h}_{\boldsymbol{\theta}}^{\mathrm{f}}(\mathbf{x})$ is the terminal constraint, $\boldsymbol{\sigma}_k$ and $\boldsymbol{\sigma}_N$ are slack variables guaranteeing the feasibility of the MPC scheme and $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{f}}$ are constant vectors that ought to be selected sufficiently large [32]. Note that these constants allow the MPC scheme to find a feasible solution, but penalize constraint violations enough to guarantee that a feasible solution is found whenever possible. While alternative feasibility-enforcing strategies, e.g., robust MPC, do exist, an exhaustive discussion on the topic is beyond the scope of this paper. Function $\lambda_{\boldsymbol{\theta}}$ parameterizes the so-called storage function, which has been added to the cost in order to enable the MPC scheme to tackle the case of so-called economic problems. Such situations arise when the MDP stage cost is not positive definite, while the MPC stage cost is forced to be positive definite in order to obtain a stabilizing feedback policy. Note that since the term $-\lambda_{\boldsymbol{\theta}}(\mathbf{x}_0)$ only depends on the current state, it does not modify the optimal policy. For more details, we refer the interested readers to [21, 33].

While Theorem 4.1.1 states that one can find suitable stage and terminal costs for any given model, adjusting the model parameters is not essential from the theoretical perspective. However, in practice, the stage and the terminal cost parameterization may not capture $\hat{L}$ and $\hat{T}$ exactly. Since $\hat{L}$ and $\hat{T}$ are (implicitly) functions of the model, using a parameterized model $\mathbf{f}_{\boldsymbol{\theta}}$ introduces extra degrees of freedom to bring $\hat{L}$ and $\hat{T}$ closer to the functions that can be represented by $L_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$. In turn, this can yield a better approximation of the optimal policy and value function.

The MPC parameterized policy can be obtained from (4.19) as follows:

$$\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{u}_0^\star(\boldsymbol{\theta}, \mathbf{s}), \qquad (4.20)$$

where $\mathbf{u}_0^\star$ is the solution of (4.19), corresponding to the first input $\mathbf{u}_0$. Moreover, the parameterized action-value function $Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$ based on MPC scheme (4.19) can be formulated as follows:

$$\begin{aligned}\hat{Q}_N^{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) := \min_{\mathbf{u},\mathbf{x},\boldsymbol{\sigma}} \quad & (4.19a) \qquad (4.21a)\\ \text{s.t.} \quad & (4.19b) - (4.19d) \qquad (4.21b)\\ & \mathbf{u}_0 = \mathbf{a}. \qquad (4.21c)\end{aligned}$$

Then one obtains the following identities:

$$\hat{V}_N^{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{a}} \hat{Q}_N^{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \quad \hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}(\mathbf{s}) \in \arg\min_{\mathbf{a}} \hat{Q}_N^{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}). \qquad (4.22)$$

We can use RL techniques, such as Q-learning and policy gradient method to tune the parameters $\boldsymbol{\theta}$ of parameterized MPC scheme (4.19) and approach the *optimal* parameter $\boldsymbol{\theta}^\star$. The use of RL for the tuning the MPC scheme can be found e.g., in [21, 34, 35].

The next section provides an analytical case study to illustrate the theoretical developments of this paper.

4.3 Analytical Case Study

We consider a Linear Quadratic Regulator (LQR) example in order to obtain the corresponding optimal value functions analytically and verify Theorem 4.1.2. The real system state transition and stage cost are given as follows:

$$\begin{aligned}\mathbf{s}^+ =& A\mathbf{s} + B\mathbf{a} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(0, \Sigma), && (4.23a)\\ L(\mathbf{s}, \mathbf{a}) =& \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}^\top \begin{bmatrix} T & N \\ N^\top & R\end{bmatrix} \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}, && (4.23b)\end{aligned}$$

with discount factor γ. One can verify the following optimal value functions:

$$V^\star(\mathbf{s}) = \mathbf{s}^\top S \mathbf{s} + \hat{v}_\infty, \;\; Q^\star(\mathbf{s}, \mathbf{a}) = \hat{v}_\infty + \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}^\top \begin{bmatrix} T + \gamma A^\top S A & N + \gamma A^\top S B \\ N^\top + \gamma B^\top S A & R + \gamma B^\top S B\end{bmatrix} \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}, \quad (4.24)$$

where $\hat{v}_\infty = \frac{\gamma}{1-\gamma}\mathrm{Tr}(S\Sigma)$ and S is obtained form the following Riccati equations:

$$\begin{aligned} T + \gamma A^\top S A =& S + (N + \gamma A^\top S B)\left(K^\star_\gamma\right)^\top, && (4.25a)\\ (R + \gamma B^\top S B) K^\star_\gamma =& N^\top + \gamma B^\top S A. && (4.25b)\end{aligned}$$

Then $\boldsymbol{\pi}^\star(\mathbf{s}) = -K^\star_\gamma \mathbf{s}$ and $\bar{\boldsymbol{\pi}}^\star(\mathbf{s}) = \tilde{\boldsymbol{\pi}}^\star(\mathbf{s}) = -K^\star_1 \mathbf{s}$, where $K^\star_1 = \lim_{\gamma \to 1} K^\star_\gamma$. We then consider a linear deterministic model:

$$\mathbf{s}^+ = \hat{A}\mathbf{s} + \hat{B}\mathbf{a}, \quad (4.26)$$

and an undiscounted OCP with the following stage cost, defined accordingly to Equation (4.5b) as:

$$\begin{aligned}\hat{L}(\mathbf{s}, \mathbf{a}) =& Q^\star(\mathbf{s}, \mathbf{a}) - V^\star(\hat{\mathbf{s}}^+) \overset{(4.24)}{=} \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}^\top \begin{bmatrix} T + \gamma A^\top S A & N + \gamma A^\top S B \\ N^\top + \gamma B^\top S A & R + \gamma B^\top S B\end{bmatrix} \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}\\ & - (\hat{A}\mathbf{s} + \hat{B}\mathbf{a})^\top S (\hat{A}\mathbf{s} + \hat{B}\mathbf{a}) := \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}^\top \begin{bmatrix} \hat{T} & \hat{N} \\ \hat{N}^\top & \hat{R}\end{bmatrix} \begin{bmatrix}\mathbf{s}\\ \mathbf{a}\end{bmatrix}. && (4.27)\end{aligned}$$

The Riccati equations for the undiscounted problem with the model (4.26) read as:

$$\begin{aligned} \hat{T} + \hat{A}^\top \hat{S} \hat{A} =& \hat{S} + (\hat{N} + \hat{A}^\top \hat{S} \hat{B})\left(\hat{K}^\star\right)^\top, && (4.28a)\\ (\hat{R} + \hat{B}^\top \hat{S} \hat{B}) \hat{K}^\star =& \hat{N}^\top + \hat{B}^\top \hat{S} \hat{A}. && (4.28b)\end{aligned}$$

with the optimal policy $\hat{\boldsymbol{\pi}}^\star_\infty(\mathbf{s}) = -\hat{K}^\star \mathbf{s}$ and the optimal value function $\hat{V}^\star_\infty(\mathbf{s}) = \mathbf{s}^\top \hat{S} \mathbf{s}$. From (4.27), we have:

$$\begin{aligned} T + \gamma A^\top S A - \hat{A}^\top S \hat{A} =& \hat{T}, && (4.29a)\\ N + \gamma A^\top S B - \hat{A}^\top S \hat{B} =& \hat{N}, && (4.29b)\\ R + \gamma B^\top S B - \hat{B}^\top S \hat{B} =& \hat{R}. && (4.29c)\end{aligned}$$

Equivalently, this entails that $\hat{T}$, $\hat{N}$ and $\hat{R}$ must satisfy

$$\hat{T} + \hat{A}^\top S \hat{A} = T + \gamma A^\top S A, \tag{4.30a}$$
$$\hat{N} + \hat{A}^\top S \hat{B} = N + \gamma A^\top S B, \tag{4.30b}$$
$$\hat{R} + \hat{B}^\top S \hat{B} = R + \gamma B^\top S B. \tag{4.30c}$$

Then:

$$\begin{aligned}\hat{T} + \hat{A}^\top S \hat{A} &\overset{(4.30a)}{=} T + \gamma A^\top S A \overset{(4.25a)}{=} S + \\ S(N + \gamma A^\top S B)\left(K_\gamma^\star\right)^\top &\overset{(4.30b)}{=} S + (\hat{N} + \hat{A}^\top S \hat{B})\left(K_\gamma^\star\right)^\top ,\end{aligned} \tag{4.31}$$

and

$$\begin{aligned}(\hat{R} + \hat{B}^\top S \hat{B}) K_\gamma^\star &\overset{(4.30c)}{=} (R + \gamma B^\top S B) K_\gamma^\star \\ &\overset{(4.25b)}{=} N^\top + \gamma B^\top S A \overset{(4.30b)}{=} \hat{N} + \hat{A}^\top S \hat{B}.\end{aligned} \tag{4.32}$$

Equations (4.31) and (4.32) show that $\hat{S} = S$ and $\hat{K}^\star = K_\gamma^\star$ satisfy the undiscounted Riccati equations (4.28). Then it reads that $\boldsymbol{\pi}^\star(\mathbf{s}) = \hat{\boldsymbol{\pi}}_\infty^\star(\mathbf{s})$ and $V^\star(\mathbf{s}) = \hat{V}_\infty^\star(\mathbf{s}) + \hat{v}_\infty$.

In the next section, we will investigate some numerical examples in order to illustrate the efficiency of the proposed method.

4.4 Numerical Examples

4.4.1 Non-quadratic stage cost

In this example, we provide a benchmark optimal investment problem with a non-quadratic stage cost. Consider the following dynamics and stage cost [36]:

$$s_{k+1} = a_k \,, \qquad L(s,a) = -\ln(As^\alpha - a), \tag{4.33}$$

where A and $0 < \alpha < 1$ are given constants. It is known that for the discount factor γ, the optimal value and policy functions are $V^\star(s) = B + C\ln(s)$ and $\pi^\star(s) = \gamma\alpha A s^\alpha$, where [37]:

$$B = \frac{\ln((1-\alpha\gamma)A) + \frac{\gamma\alpha}{1-\gamma\alpha}\ln(\alpha\gamma A)}{\gamma - 1}, \qquad C = \frac{\alpha}{\alpha\gamma - 1}. \tag{4.34}$$

We then consider a model of the dynamics with $\hat{s}_{k+1} = \mu\hat{a}_k$ and, based on this model, we construct a finite-horizon undiscounted MPC with the costs according to Equation (4.5) in Theorem 4.1.1 and $N = 10$. In this example we have considered $A = 5$, $\alpha = 0.34$, $\mu = 0.8$ and $\gamma = 0.9$. Figures 4.1 and 4.2 compare the optimal value and policy functions from the discounted real system (4.33) and from the MPC scheme with a wrong model. As predicted by Theorem 4.1.1, one can see that they match perfectly. Note that the results are valid for every discount factor $0 < \gamma < 1$, every horizon length, and for other values of the constants A, α, and μ.

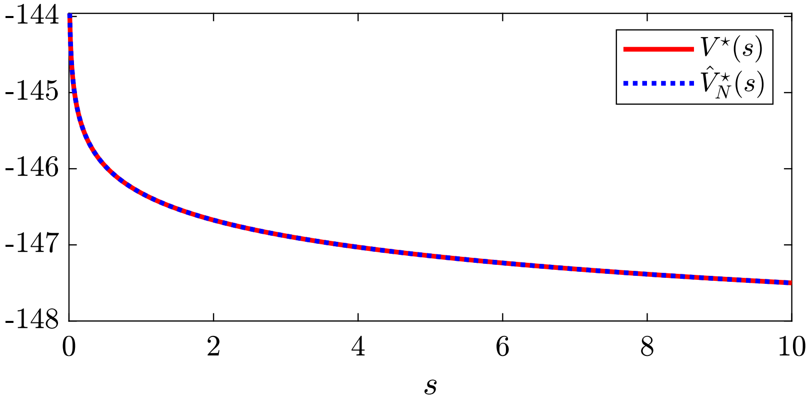


Figure 4.1: Optimal value functions resulting from the discounted real system and undiscounted MPC scheme with the wrong model.

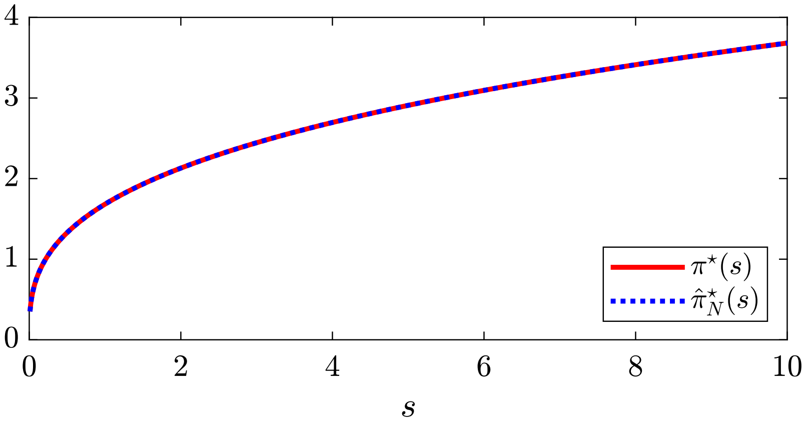


Figure 4.2: Optimal policy functions resulting from the discounted real system and undiscounted MPC scheme with the wrong model.

4.4.2 Inverted pendulum with process noise

We consider the following discrete-time stochastic dynamics, representing an inverted pendulum with a random support excitation:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \begin{bmatrix} s_k(2) \\ (\frac{g}{l} + \xi)\sin(s_k(1)) \end{bmatrix} \delta t + \begin{bmatrix} 0 \\ \frac{\delta t}{ml^2} \end{bmatrix} \mathbf{a}_k \tag{4.35}$$

where $g = 9.81$, $l = 0.3$, $m = 0.5$ and $\delta t = 0.1$ are constants representing the gravity, mass, length, and sampling time of the discrete dynamics. Disturbance $\xi \sim \mathcal{U}[-0.5, 0.5]$ has a uniform distribution and $\mathbf{s}_k := [s_k(1),\, s_k(2)]^\top$ is the system state and $\mathbf{a}_k$ is the system input. We consider $L(\mathbf{s}, \mathbf{a}) = \mathbf{s}^\top \mathbf{s} + \mathbf{a}^2$ as a stage cost with the discount factor $\gamma = 0.95$. We first aim to find an approximate solution for the optimal policy and the optimal value functions using Dynamic Programming (DP).

We consider the state constraints $-1 \leq s_k(1) \leq 1$, $-1 \leq s_k(2) \leq 1$ and the input constraint $-0.8 \leq a_k \leq 0.8$. Figure 4.3 and Figure 4.4 show the optimal value function and the optimal policy function resulting from DP for the discounted infinite-horizon MDP.

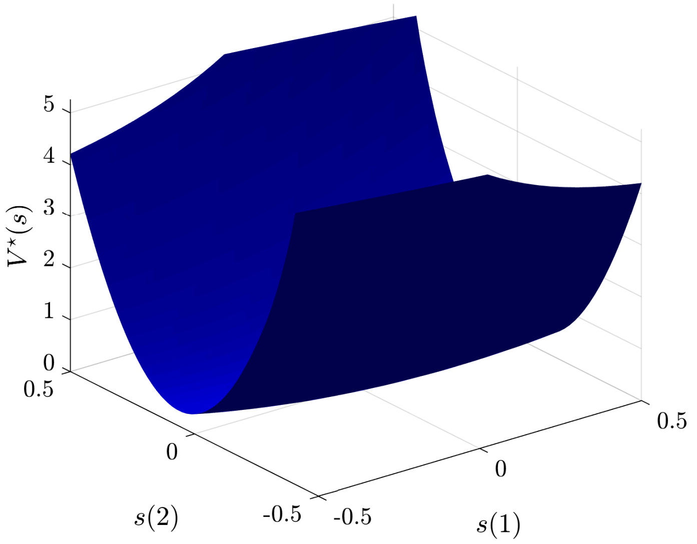


Figure 4.3: Optimal Value function resulting from ADP.

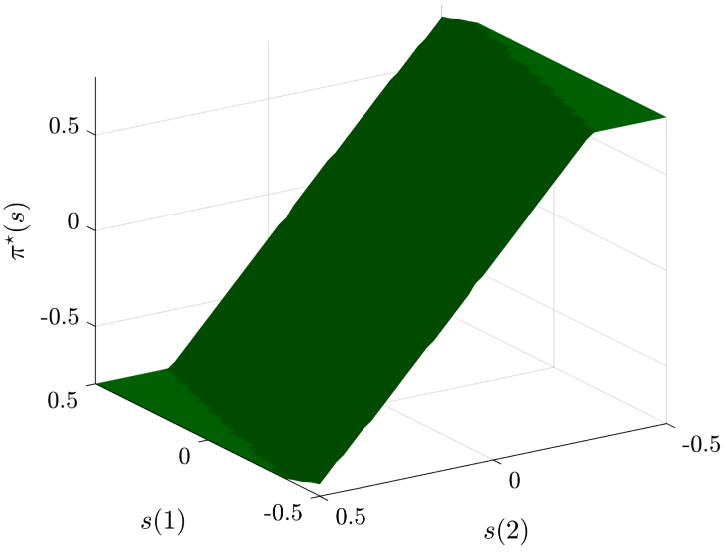


Figure 4.4: Optimal Policy function resulting from DP.

We build an undiscounted finite-horizon OCP with a wrong model in order to capture the optimal value and the optimal policy functions of the discounted infinite-horizon MDP. To do this, we consider an MPC scheme with a deterministic linearized form of the dynamics as a model of the real system as follows:

$$\hat{\mathbf{s}}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k) = \hat{\mathbf{s}}_k + \begin{bmatrix} \hat{s}_k(2) \\ \frac{g}{\theta_l}\hat{s}_k(1) \end{bmatrix} \delta t + \begin{bmatrix} 0 \\ \frac{\delta t}{m\theta_l^2} \end{bmatrix} \hat{\mathbf{a}}_k \tag{4.36}$$

where $\hat{\mathbf{s}}_k := [\hat{s}_k(1),\, \hat{s}_k(2)]^\top$ and $\hat{\mathbf{a}}_k$ are the model state and input. Moreover, we consider an uncertain l with an adjustable parameter θ_l, with an initial value 0.25. We consider the parameterized MPC scheme with the horizon length $N = 10$ and the following parameterized quadratic stage and terminal cost:

$$T_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{s}^\top G\mathbf{s}, \qquad L_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \begin{bmatrix} \mathbf{s} \\ a \end{bmatrix}^\top H \begin{bmatrix} \mathbf{s} \\ a \end{bmatrix} \tag{4.37}$$

where G and H are parametric positive definite matrices. Then the parameters vector $\boldsymbol{\theta}$ gathers all of the adjustable parameters as:

$$\boldsymbol{\theta} = \{\theta_l,\, G,\, H\}. \tag{4.38}$$

Figure 4.5 shows the difference between the MPC value function $\hat{V}_N^{\boldsymbol{\theta}}$ and the optimal value function $V^\star$ computed by DP. The blue and red curves represent this difference at the beginning of the learning and after 500 learning steps, respectively. Figure 4.6 shows the difference between the MPC policy $\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}$ and the optimal policy $\boldsymbol{\pi}^\star$ computed by DP. As it can be seen, the results are getting closer to zero as the learning proceeds. Note that the stage and terminal costs yielding a perfect match of $V^\star$ and $\boldsymbol{\pi}^\star$, as per Theorem 4.1.1, do not have a quadratic form, hence the selected MPC formulation cannot capture them exactly. The green curves in Figures 4.5 and 4.6 have been obtained by computing these stage and terminal costs numerically and shows the corresponding $\hat{V}_N^\star - V^\star$ and $\hat{\boldsymbol{\pi}}_N^\star - \boldsymbol{\pi}^\star$. As expected the difference is zero, modulo tiny numerical inaccuracies.

Finally, Figure 4.7 illustrates the closed-loop performance of the system under the MPC policy $\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}$. As the closed loop cost decreases, this demonstrates that RL can be effective in tuning the MPC parameters so as to achieve the best closed-loop performance.

4.4.3 Learning based MPC

In this example, we consider example 3.3. We suppose that we have a state constraint in the form of $x \geq 0$, and we set discount factor $\gamma = 0.95$ and the following MDP stage cost:

$$L(\mathbf{s}, \mathbf{a}) = \begin{bmatrix} \mathbf{s} \\ \mathbf{a} \end{bmatrix}^\top \begin{bmatrix} I_4 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{a} \end{bmatrix} + \lambda \max(-x, 0), \tag{4.39}$$

where λ is a large constant value introduced to model the state constraint as a soft constraint.

In the MPC scheme, we use the linear model $\mathbf{s}^+ = \hat{A}\mathbf{s} + \hat{B}\mathbf{a}$ obtained by linearizing $\mathbf{f}$ at the origin. We provide a parametrized quadratic stage and terminal cost and select prediction horizon $N = 20$.

We use the deterministic policy gradient method to minimize the performance function $J(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{s}_0}[\hat{V}_N^{\boldsymbol{\theta}}(\mathbf{s}_0)]$, and we run a simulation for 1000 learning steps. Figure 4.8 shows the value function over the learning steps for a fixed initial state. This illustrates that RL successfully manages to reduce J throughout the iterates, therefore tuning MPC as desired.

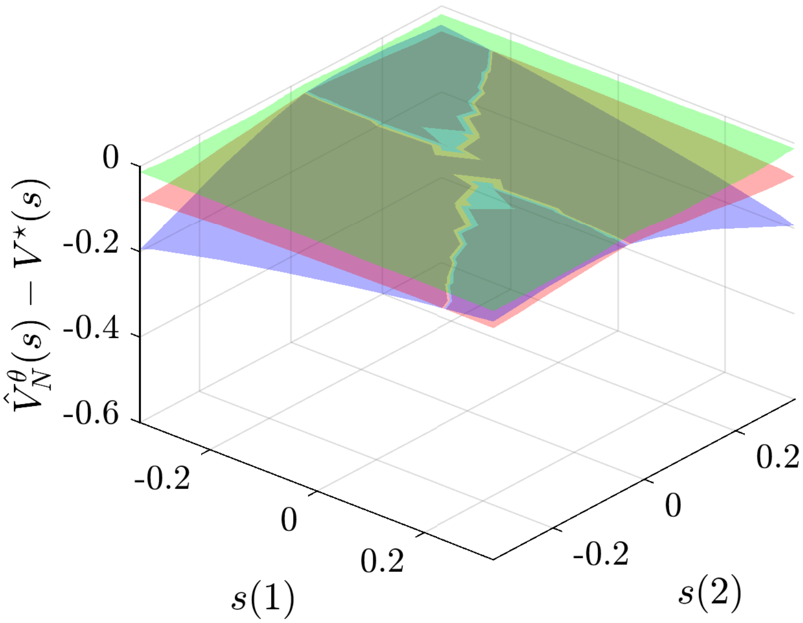


Figure 4.5: The difference between the MPC based parameterized value function and optimal value function for the beginning of the learning (blue) and after 500 learning steps (red) and $\hat{V}_N^\star - V^\star$ (green).

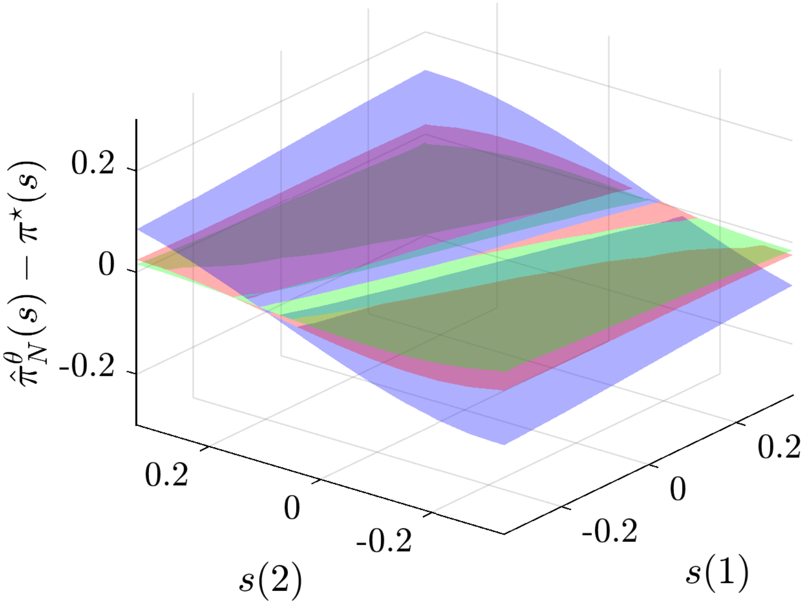


Figure 4.6: The difference between the MPC based parameterized policy function and optimal policy function for the beginning of the learning (blue) and after 500 learning steps (red) and $\hat{\pi}_N^\star - \pi^\star$ (green).

Figure 4.9 shows the states and input trajectories of the real system corresponding the 1000^{th} learning step. The MPC scheme with the positive definite stage cost and other stability conditions in

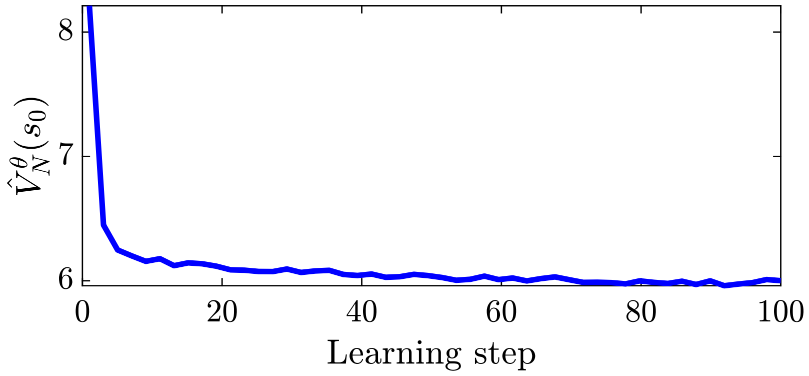


Figure 4.7: The MPC-based value function $\hat{V}_N^{\boldsymbol{\theta}}(\mathrm{s}_0)$ during the learning.

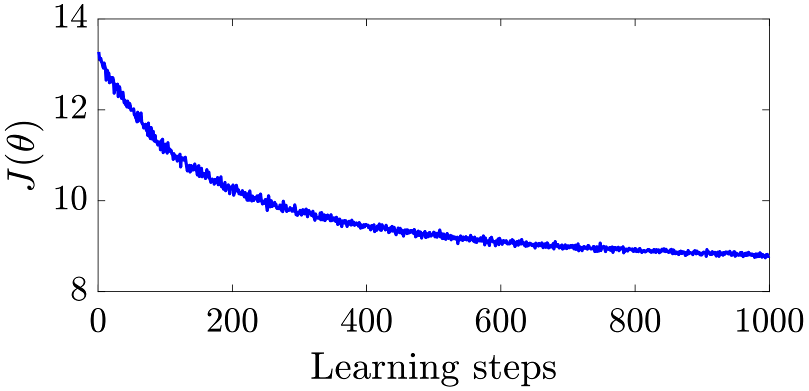


Figure 4.8: The closed-loop performance of the MPC scheme over RL-steps.

the terminal cost and terminal constraint is able to deliver the stabilizing policy for the closed-loop system for the small enough model error [30]. Note that the terminal cost and constraint conditions can be relaxed for the large enough MPC horizon [38].

Figure 4.10 compares the state constraint violation for $x \geq 0$ in the first and last (1000^{th}) learning step. As one can see, RL reduces the state constraint violation.

4.5 Summary

In this chapter, we showed that a finite-horizon OCP can capture the optimal policy and value functions of any MDPs with either discounted or undiscounted cost even if we use an inexact model in the OCP. We showed that an MPC scheme can be interpreted as a particular case of the OCP where we use a deterministic model to avoid computational complexity. In practice, we proposed the used of a parameterized MPC scheme to provide a structured function approximator for the RL techniques. RL algorithms then can be used in order to tune the MPC parameters to achieve the best closed-loop performance. We verified the theorems in an LQR case and investigated some nonlinear examples to illustrate the efficiency of the method numerically.

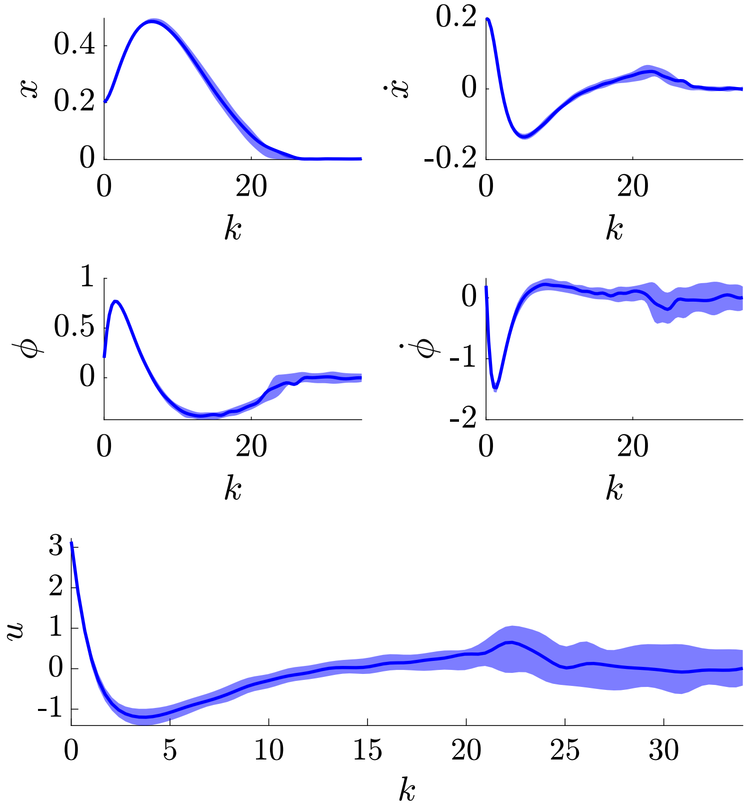


Figure 4.9: States and input trajectories of the real system for the last learning step.

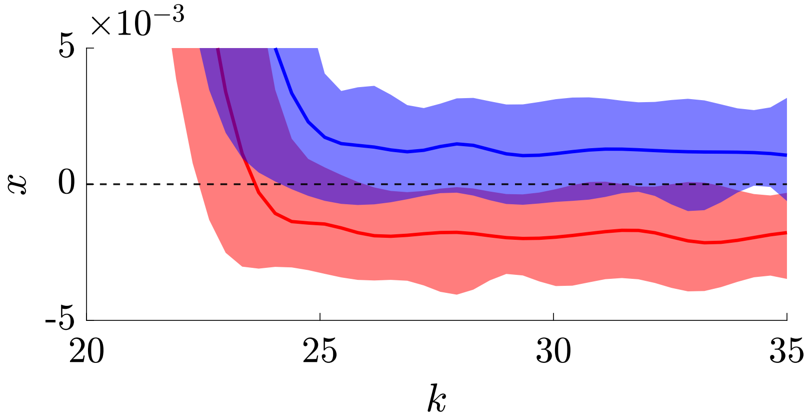


Figure 4.10: Violation of the state constraint $x \geq 0$ in the first step (red) and the last step (blue).

References

head1.png

[1] Richard Bellman. “A Markovian decision process”. In: *Journal of mathematics and mechanics* (1957), pp. 679–684.

[2] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[3] Sridhar Mahadevan. “Average reward reinforcement learning: Foundations, algorithms, and empirical results”. In: *Machine learning* 22.1 (1996), pp. 159–195.

[4] Arne Groß, Antonia Lenders, Tobias Zech, Christof Wittwer, and Moritz Diehl. “Using Probabilistic Forecasts in Stochastic Optimization”. In: *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. 2020, pp. 1–6.

[5] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[6] David Silver et al. “Deterministic policy gradient algorithms”. In: *International conference on machine learning*. PMLR. 2014, pp. 387–395.

[7] Michail G Lagoudakis and Ronald Parr. “Least-squares policy iteration”. In: *The Journal of Machine Learning Research* 4 (2003), pp. 1107–1149.

[8] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. “MPC-based reinforcement learning for economic problems with application to battery storage”. In: *2021 European Control Conference (ECC)* (2021), pp. 2573–2578. DOI: 10.23919/ECC54610.2021.9654852.

[9] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[10] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. “Global convergence of policy gradient methods for the linear quadratic regulator”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1467–1476.

[11] Thomas Furmston, Guy Lever, and David Barber. “Approximate newton methods for policy search in markov decision processes”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 8055–8105.

[12] Kay Hansel, Janosch Moos, and Cedric Derstroff. “Benchmarking the Natural Gradient in Policy Gradient Methods and Evolution Strategies”. In: *Reinforcement Learning Algorithms: Analysis and Applications* (2021), pp. 69–84.

[13] Shun-ichi Amari. “Natural Gradient Works Efficiently in Learning”. In: *Neural Computation* 10.2 (1998), pp. 251–276.

[14] Sham M Kakade. “A natural policy gradient”. In: *Advances in neural information processing systems*. 2002, pp. 1531–1538.

[15] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. “Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[16] Arash Givchi and Maziar Palhang. “Quasi Newton temporal difference learning”. In: *Asian Conference on Machine Learning*. PMLR. 2015, pp. 159–172.

[17] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. “Natural actor-critic”. In: *European Conference on Machine Learning*. Springer. 2005, pp. 280–291.

[18] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Wenqi Cai, and Sebastien Gros. “Quasi-Newton Iteration in Deterministic Policy Gradient”. In: *2022 American Control Conference (ACC)* (2022), pp. 2124–2129. doi: 10.23919/ACC53348.2022.9867217.

[19] Karen Braman. “Third-order tensors as linear operators on a space of matrices”. In: *Linear Algebra and its Applications* 433.7 (2010), pp. 1241–1253.

[20] Andreas B Martinsen, Anastasios M Lekkas, and Sebastien Gros. “Combining system identification with reinforcement learning-based MPC”. In: *arXiv preprint arXiv:2004.03265* (2020).

[21] Sébastien Gros and Mario Zanon. “Data-driven economic NMPC using reinforcement learning”. In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[22] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. “An introduction to deep reinforcement learning”. In: *arXiv preprint arXiv:1811.12560* (2018).

[23] Arash Bahari Kordabad and Mehrdad Boroushaki. “Emotional learning based intelligent controller for mimo peripheral milling process”. In: *Journal of Applied and Computational Mechanics* 6.3 (2020), pp. 480–492.

[24] J. Andrew (Drew) Bagnell and Jeff Schneider. “Covariant Policy Search”. In: *Proceedings of the International Joint Conference on Artifical Intelligence*. 2003, pp. 1019–1024.

[25] Sébastien Gros and Mario Zanon. “A Dissipativity Theory for Undiscounted Markov Decision Processes”. In: *arXiv preprint arXiv:2104.10997* (2021).

[26] Mario Zanon, Sébastien Gros, and Michele Palladino. “Stability-Constrained Markov Decision Processes Using MPC”. In: *Automatica* 143 (2022), p. 110399.

[27] Romain Postoyan, Lucian Buşoniu, Dragan Nešić, and Jamal Daafouz. “Stability analysis of discrete-time infinite-horizon optimal control with discounted cost”. In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2736–2749.

[28] Mathieu Granzotto, Romain Postoyan, Lucian Buşoniu, Dragan Nešić, and Jamal Daafouz. "Finite-horizon discounted optimal control: stability and performance". In: *IEEE Transactions on Automatic Control* 66.2 (2020), pp. 550–565.

[29] Mario Zanon and Sébastien Gros. "A new dissipativity condition for asymptotic stability of discounted economic MPC". In: *Automatica* 141 (2022), p. 110287.

[30] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[31] Arash Bahari Kordabad, Mario Zanon, and Sebastien Gros. "Equivalency of Optimality Criteria of Markov Decision Process and Model Predictive Control". In: *arXiv preprint, Submitted* (2022). DOI: 10.48550/arXiv.2210.04302.

[32] Eric C Kerrigan and Jan M Maciejowski. "Soft constraints and exact penalty functions in model predictive control". In: *Control 2000 Conference, Cambridge*. Citeseer. 2000, pp. 2319–2327.

[33] Arash Bahari Kordabad and Sebastien Gros. "Verification of Dissipativity and Evaluation of Storage Function in Economic Nonlinear MPC using Q-Learning". In: *IFAC-PapersOnLine* 54.6 (2021). 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021, pp. 308–313.

[34] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Anastasios M Lekkas, and Sébastien Gros. "Reinforcement learning based on scenario-tree MPC for ASVs". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 1985–1990.

[35] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. "Multi-agent battery storage management using MPC-based reinforcement learning". In: *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2021, pp. 57–62.

[36] Manuel S Santos and Jesus Vigo-Aguiar. "Analysis of a numerical dynamic programming algorithm applied to economic models". In: *Econometrica* (1998), pp. 409–426.

[37] Lars Grüne, Christopher M Kellett, and Steven R Weller. "On a discounted notion of strict dissipativity". In: *IFAC-PapersOnLine* 49.18 (2016), pp. 247–252.

[38] Ali Jadbabaie and John Hauser. "On the stability of receding horizon control with a general terminal cost". In: *IEEE Transactions on Automatic Control* 50.5 (2005), pp. 674–678.