# Contributions to Batch Mode Reinforcement Learning

PhD dissertation by
**Raphael Fonteneau**

Department of Electrical Engineering and Computer Science,
University of Liège, BELGIUM
2011

# Foreword

When I was studying electrical engineering and computer science at the French Grande Ecole *SUPELEC* (*Ecole Supérieure d'Electricité*), I had the opportunity to work with Dr. Damien Ernst, who was professor there, on a research project dealing with the use of system theory for better understanding the dynamics of the HIV infection. This first experience was the trigger to pursue this research adventure at the *University of Liège* (Belgium) under the supervision of Dr. Damien Ernst and Prof. Louis Wehenkel on the practical problem of extracting decision rules from clinical data in order to better treat patients suffering from chronic-like diseases. This problem is often formalized as a batch mode reinforcement learning problem.

The work done during my PhD thesis enriches this body of work in batch mode reinforcement learning so as to try to bring it to a level of maturity closer to the one required for finding decision rules from clinical data. Most of the research exposed in this dissertation has been done in collaboration with Prof. Susan A. Murphy from the *University of Michigan* who has pioneered the use of reinforcement learning techniques for inferring dynamic treatment regimes, and who has had the kindness to invite me in her lab in November 2008. This dissertation is a collection a several research publications that have emerged from this work.

# Acknowledgements

First and foremost, I would like to express my deepest gratitude and appreciation to Dr. Damien Ernst, for offering me the opportunity to discover the world of research. All along these four years, he proved to be a great collaborator on many scientific and human aspects. The present work is mostly due to his support, patience, enthusiasm, creativity and, of course, personal friendship.

I would like to extend my deepest thanks to Prof. Louis Wehenkel for his up to the point suggestions and advice regarding every research contribution reported in this dissertation. His remarkable talent and research experience have been very valuable at every stage of this research.

My deepest gratitude also goes to Prof. Susan A. Murphy for being such an inspiration in research, for her suggestions and for her enthusiasm. I would also like to thank her very much for welcoming me in her lab at the University of Michigan.

I also address my warmest thanks to all the SYSTMOD research unit, the Department of Electrical Engineering and Computer Science, the GIGA and the University of Liège, where I found a friendly and stimulating research environment. Many thanks to the academic staff, especially to Dr. Pierre Geurts, Prof. Quentin Louveaux and Prof. Rodolphe Sepulchre. A special acknowledgement to my office neighbors, Bertrand Cornélusse and Renaud Detry. Many additional thanks to Julien Becker, Vincent Botta, Anne Collard, Boris Defourny, Guillaume Drion, Fabien Heuze, Samuel Hiard, Vân-Anh Huynh-Thu, Michel Journée, Thibaut Libert, David Lupien St-Pierre, Francis Maes, Alexandre Mauroy, Gilles Meyer, Laurent Poirrier, Pierre Sacré, Firas Safadi, Alain Sarlette, François Schnitzler, Olivier Stern, Laura Trotta, Wang Da and many other colleagues and friends from Montefiore and the GIGA that I forgot to mention here. I also would like to thank the administrative staff of the University of Liège, and, in particular, Marie-Berthe Lecomte, Charline Ledent-De Baets and Diane Zander for their help.

I also would like to thank all the scientists, non-affiliated with the University of Liège, with whom I have had interesting scientific discussions, among others: Lucian

Busoniu, Bibhas Chakraborty, Jing Dai, Matthieu Geist, Wassim Jouini, Eric Laber, Daniel Lizotte, Marie-José Mhawej, Mahdi Milani Fard, Claude Moog, Min Qian, Emmanuel Rachelson, Guy-Bart Stan and Peng Zhang.

Many thanks to the members of the jury for carefully reading this dissertation and for their advice to improve its quality.

I am very grateful to the FRIA (Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture) from the Belgium fund for scientific research FRS-FNRS and to the PAI (Poles d'Attraction Interuniversitaires) BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks) and DYSCO (Dynamical Systems, Control and Optimization).

Finally, I would like to express my deepest personal gratitude to my parents for teaching me the value of knowledge and work, and of course, for their love. Many thanks to my sisters Adeline and Anne-Elise, my brother Emmanuel, my whole family and family-in-law, and longtime friends for their unconditional support before and during these last four years.

Thank you to my little Gabrielle for her encouraging smiles.

To you Florence, my beloved wife, no words can express how grateful I feel. Thank you for everything.

Raphael Fonteneau
Liège,
January 2011.

# Abstract

This dissertation presents various research contributions published during these four years of PhD in the field of batch mode reinforcement learning, which studies optimal control problems for which the only information available on the system dynamics and the reward function is gathered in a set of trajectories.

We first focus on deterministic problems in continuous spaces. In such a context, and under some assumptions related to the smoothness of the environment, we propose a new approach for inferring bounds on the performance of control policies. We also derive from these bounds a new inference algorithm for generalizing the information contained in the batch collection of trajectories in a cautious manner. This inference algorithm as itself lead us to propose a $\min\max$ generalization framework.

When working on batch mode reinforcement learning problems, one has also often to consider the problem of generating informative trajectories. This dissertation proposes two different approaches for addressing this problem. The first approach uses the bounds mentioned above to generate data tightening these bounds. The second approach proposes to generate data that are predicted to generate a change in the inferred optimal control policy.

While the above mentioned contributions consider a deterministic framework, we also report on two research contributions which consider a stochastic setting. The first one addresses the problem of evaluating the expected return of control policies in the presence of disturbances. The second one proposes a technique for selecting relevant variables in a batch mode reinforcement learning context, in order to compute simplified control policies that are based on smaller sets of state variables.

# Résumé

Ce manuscrit rassemble différentes publications scientifiques réalisées au cours de ces quatre années de thèse dans le domaine de l'apprentissage par renforcement en mode "batch", dans lequel on souhaite contrôler de manière optimale un système pour lequel on ne connait qu'un ensemble fini de trajectoires données a priori.

Dans un premier temps, cette problématique a été développée dans un contexte déterministe, en considérant des espaces continus. En travaillant sous certaines hypothèses de régularité de l'environnement, une nouvelle approche de calcul de bornes sur les performances des lois de contrôle a été developpée. Cette approche a ensuite permis le dévelopement d'un algorithme d'inférence de loi de contrôle abordant le problème de généralisation de manière précautionneuse. De manière plus formelle, une réflexion sur la possibilité de généraliser suivant le paradigme $\min\max$ a également été proposée.

Lorsque l'on travaille en mode batch, on doit également souvent faire face au problème relatif à la génération de bases de données aussi informatives que possible. Ce problème est abordé de deux manières différentes dans ce manuscrit. La première consiste à faire appel aux bornes décrites ci-dessus dans le but de générer des données menant à une augmentation de la précision de ces bornes. La deuxième propose de générer des données en des endroits pour lesquels il est prédit (en utilisant un modèle de prédiction) qu'une modification de la loi de contrôle courante sera induite.

La majorité des contributions rassemblées dans ce manuscrit considèrent un environnement déterministe, mais on y présente également deux contributions se plaçant dans un environnement stochastique. La première traite de l'évaluation de l'espérance du retour des lois de contrôle sous incertitudes. La deuxième propose une technique de sélection de variables qui permet de construire des lois de contrôles simplifées basées sur des petits sous-ensembles de variables.

# Contents

# Chapter 1

# Overview

*In this first chapter, we introduce the general batch mode reinforcement learning setting, and we give a short summary of the different contributions exposed in the following chapters.*

## 1.1 Introduction

Optimal control problems arise in many real-life applications, such as for instance engineering [21], medicine [4, 16, 17] or artificial intelligence [15]. Over the last decade, techniques developed by the reinforcement learning community have become more and more popular for addressing those types of problems.

Initially, reinforcement learning was focusing on how to design intelligent agents able to interact with their environment so as to maximize a numerical criterion [1, 22, 23]. Since the end of the nineties, many researchers have focused on the resolution of a subproblem of reinforcement learning: computing a high-performance policy when the only information available on the environment is contained in a batch collection of trajectories of the agent [2, 3, 15, 19, 21]. This subfield of reinforcement learning is known as "batch mode reinforcement learning", a term that was first coined in the work of Ernst et al. 2005 [3].

Among the different applications of batch mode reinforcement learning, a very promising but challenging one is the inference of dynamic treatment regimes from clinical data representing the evolution of patients [18, 20]. Dynamic treatment regimes are sets of sequential decision rules defining what actions should be taken at a specific instant to treat a patient based on the information observed up to that instant. Ideally, dynamic treatment regimes should lead to treatments which result in the most favorable clinical outcome possible. The information available to compute dynamic treatment regimes is usually provided by clinical protocols where several patients are monitored through different (randomized) treatments.

While batch mode reinforcement learning appears to be a promising paradigm for learning dynamic treatment regimes, many challenges still need to be addressed for these methods to keep their promises:

1. Medical applications expect high guarantees on the performance of treatments, while these are usually not provided by batch mode reinforcement learning algorithms. Additionally, the problem of computing tight estimates of the performance of control policies is still challenging in some specific frameworks;

2. The experimental protocols for generating the clinical data should be designed so as to get highly informative data. Therefore, it would be desirable to have techniques for generating highly informative batch collections of trajectories;

3. The design of dynamic treatment regimes has to take into consideration the fact that treatments should be based on a limited number of clinical indicators to be easier to apply in real-life. Batch mode reinforcement learning algorithms do not address this problem of inferring "simplified" control policies;

4. Clinical data gathered from experimental protocols may be highly noisy or incomplete;

5. Confounding issues and partial observability occur frequently when dealing with specific types of chronic-like diseases, such as for instance psychotic diseases.

These challenges, especially challenges 1., 2. and 3., have served as inspiration for the research in batch mode reinforcement learning reported in this dissertation. The different research contributions that have emerged from these challenges are briefly described in this introduction.

### 1.1.1 Batch mode reinforcement learning

All along this dissertation, we will consider a (possibly stochastic) discrete-time system, governed by a system dynamics $f$, which has to be controlled so as to collect high cumulated rewards induced by a reward function $\rho$. The optimization horizon is denoted by $T$, and this optimization horizon is assumed to be finite, i.e. $T \in \mathbb{N}_0$. For every time $t \in \{0, \ldots, T-1\}$, the system is represented by a state $x_t$ that belongs to a continuous, normed state space $\mathcal{X}$, and the system can be controlled through an action $u_t$, that belongs to an action space $\mathcal{U}$.

For each optimal control problem, a batch collection of data is available. This collection of data is given in the form of a finite set of one-step system transitions, i.e., a set of four tuples

$$\mathcal{F}_n = \left\{ \left(x^l, u^l, r^l, y^l\right) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X} \right\}_{l=1}^{n} , \qquad (1.1)$$

where, for all $l \in \{1, \ldots, n\}$, $(x^l, u^l)$ is a state-action point, and $r^l$ and $y^l$ are the (eventually stochastic) values induced by the reward function $\rho$ and the system dynamics $f$ in the state-action point $(x^l, u^l)$.

Within this general context, several frameworks (deterministic or stochastic, continuous action space or finite action space) and different objectives are considered in this dissertation. For each contribution, we will clearly specify in which setting we work, and which objectives are addressed.

### 1.1.2 Main contributions presented in this dissertation

The main contributions exposed in this dissertation are the following:

- In a deterministic framework, we propose a new approach for computing, from a batch collection of system transitions $\mathcal{F}_n$, bounds on the performance of control policies when the system dynamics $f$, the reward function $\rho$ and the control policies are Lipschitz continuous. This contribution is briefly described hereafter in Section 1.2, and fully detailed in Chapter 2;

- We propose, in a deterministic framework, a $\min \max$ approach to address the generalization problem in a batch mode reinforcement learning context. We also introduce a new batch mode reinforcement learning algorithm having cautious generalization properties; those contributions are briefly presented in Section 1.3 and fully reported in Chapter 3;
- We propose, still in a deterministic framework, new sampling strategies to select areas of the state-action space where to sample additional system transitions to enrich the current batch sample $\mathcal{F}_n$; those contributions are summarized in Sections 1.4 and 1.5, and detailed in Chapters 4 and 5;
- We propose, in a stochastic framework, a new approach for building an estimator of the performances of control policies in a model-free setting; this contribution is summarized in Section 1.6 and fully reported in Chapter 6;
- We propose, in a stochastic framework, a variable ranking technique for batch mode reinforcement learning problems. The objective of this technique is to compute control policies that are based on smaller subsets of variables. This approach is briefly presented in Section 1.7 and fully developed in Chapter 7.

Each of the following chapters (from 2 to 7) of this dissertation is a research publication that has been slightly edited. Each of these chapters can be read independently. Chapter 8 will conclude and discuss research directions suggested by this work. In the following sections of this introduction, we give a short technical summary of the different contributions of the present dissertation.

## 1.2 Chapter 2: Inferring bounds on the performance of a control policy from a sample of trajectories

In Chapter 2, we consider a deterministic discrete-time system whose dynamics over $T$ stages is described by the time-invariant equation:

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \dots, T-1, \tag{1.2}$$

where for all $t$, the state $x_t$ is an element of the continuous normed state space $(\mathcal{X}, \|.\|_{\mathcal{X}})$ and the action $u_t$ is an element of the continuous normed action space $(\mathcal{U}, \|.\|_{\mathcal{U}})$. The transition from $t$ to $t+1$ is associated with an instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R}. \tag{1.3}$$

We consider in this chapter deterministic time-varying $T$-stage policies

$$h : \{0, 1, \ldots, T-1\} \times \mathcal{X} \to \mathcal{U} \tag{1.4}$$

which select at time $t$ the action $u_t$ based on the current time and the current state ($u_t = h(t, x_t)$). The return over $T$ stages of a policy $h$ from a state $x_0$ is denoted by

$$J^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t)). \tag{1.5}$$

We also assume that the unknown dynamics $f$, the unknown reward function $\rho$ and the policy $h$ are Lipschitz continuous, and that three constants $L_f$, $L_\rho$, $L_h$ satisfying the Lipschitz inequalities are known. Under these assumptions, we show how to compute a lower bound $L^h_{\mathcal{F}_n}(x_0)$ on the return over $T$ stages of any given policy $h$ when starting from a given initial state $x_0$:

$$L^h_{\mathcal{F}_n}(x_0) \leq J^h(x_0)\ . \tag{1.6}$$

This lower bound is computed from a specific sequence of system transitions

$$\tau = \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} \tag{1.7}$$

as follows

$$L^h_{\mathcal{F}_n}(x_0) = \sum_{t=0}^{T-1} (r^{l_t} - L_{Q_{T-t}} \delta_t) \leq J^h(x_0), \tag{1.8}$$

where

$$\forall t \in \{0, 1, \ldots, T-1\}, \delta_t = \left\|x^{l_t} - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u^{l_t} - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} \tag{1.9}$$

with $y^{l_{-1}} = x_0$, and

$$L_{Q_{T-t}} = L_\rho \left( \sum_{t=0}^{T-t-1} [L_f(1 + L_h)]^t \right). \tag{1.10}$$

Moreover, we show that the lower bound $L^h_{\mathcal{F}_n}(x_0)$ converges towards the actual return $J^h(x_0)$ when the sparsity $\alpha^*_{\mathcal{F}_n}$ of the set of system transitions converges towards zero:

$$\exists\, C \in \mathbb{R}^+ : J^h(x_0) - L^h_{\mathcal{F}_n}(x_0) \leq C\alpha^*_{\mathcal{F}_n} . \tag{1.11}$$

The material presented in this Chapter 2 as been published in the Proceedings of the *IEEE Symposium Series on Computational Intelligence - Adaptive Dynamic Programming and Reinforcement Learning* (IEEE ADPRL 2009) [5].

## 1.3 Chapter 3: Towards $\min\max$ generalization in reinforcement learning

In Chapter 3, we still consider a deterministic setting and a continuous normed state space $(\mathcal{X}, \|.\|_{\mathcal{X}})$, but the action space $\mathcal{U}$ is assumed to be finite.

The approach developed in [5] (introduced above in Section 1.2) can also be derived in the context of discrete action spaces. In such a context, given an initial state $x_0 \in \mathcal{X}$ and sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, we show how to compute, from a sample of system transitions $\mathcal{F}_n$, a lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ on the $T-$stage return of the sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$:

$$L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq J^{u_0,\ldots,u_{T-1}}(x_0) . \tag{1.12}$$

This lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ is also computed from a specific sequence of system transitions

$$\tau = \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} \tag{1.13}$$

under the condition that it is compatible with the sequence of actions $(u_0, \ldots, u_{T-1})$ as follows:

$$u^{l_t} = u_t,\ \forall t \in \{0, \ldots, T-1\} . \tag{1.14}$$

The lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ then writes

$$L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \doteq \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}}\right] , \tag{1.15}$$

$$y^{l_{-1}} = x_0 , \tag{1.16}$$

$$L_{Q_{T-t}} = L_\rho \sum_{i=0}^{T-t-1} (L_f)^i \tag{1.17}$$

where $L_f$ and $L_\rho$ are upper bounds on the Lipschitz constants of the functions $f$ and $\rho$.

Furthermore, the resulting lower bound $L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$ can be used in order to compute, from a sample of trajectories, a control policy $(\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ leading to the maximization of the previously mentioned lower bound:

$$(\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \underset{(u_0,\ldots,u_{T-1})\in\mathcal{U}^T}{\arg\max} \left\{ L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \right\} . \tag{1.18}$$

Such a control policy is given by a sequence of actions extracted from a sequence of system transitions leading to the maximization of the previous lower bound. Due to the tightness properties of the lower bound, the sequence of actions is proved to converge towards an optimal control policy when the sparsity of the sample of transitions converges towards zero.

The resulting batch mode reinforcement learning algorithm, called CGRL for “Cautious approach to Generalization in Reinforcement Learning”, was shown to have cautious generalization properties that turned out to be crucial in “dangerous” environments for which standard batch mode reinforcement learning algorithms would fail. This work was published in the Proceedings of the *International Conference on Agents and Artificial Intelligence* (ICAART 2010) [7], where it received a “Best Student Paper Award”.

The cautious generalization properties of the CGRL algorithm were shown to be quite conservative, and we decided to better investigate how they could be “optimized” so as to result into a $\min\max$ approach to generalization. The main results of this preliminary investigation have been published in an extended and revised version of [7] as a book chapter [6].

Both the CGRL algorithm [7] and the work about the $\min\max$ approach towards generalization [6] are reported in Chapter 3, which can be seen as an extended version of the book chapter [6].

## 1.4 Chapter 4: Generating informative trajectories by using bounds on the return of control policies

Even if, in a batch mode reinforcement learning context, we assume that all the available information about the optimal control problem is contained in a set of system transitions, there are many interesting engineering problems for which one has to decide where to sample additional system transitions.

In Chapter 4, we address this issue in a deterministic setting, and where the state space is continuous and the action space is finite. The system dynamics and the reward function are assumed to be Lipschitz continuous. While the ideas developed in the previous sections [7, 6] rely on the computation of lower bounds on the return of control policies, one can, in a similar way, compute tight upper bounds from a sample of system transitions $\mathcal{F}_n$. The lower bound $L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$ and the upper bound $U_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$,

$$L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \leq J^{u_0,\ldots,u_{T-1}}(x_0) \leq U_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)\,. \tag{1.19}$$

can be simultaneously exploited to select where to sample new system transitions in order to generate a significant decrease of current bounds width:

$$\Delta_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) = U_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) - L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)\,. \tag{1.20}$$

The preliminary results of this research have been published as a 2-page highlight paper in the Proceedings of the *Workshop on Active Learning and Experimental Design* 2010 [8] (In conjunction with the *International Conference on Artificial Intelligence an Statistics* (AISTATS 2010)).

## 1.5 Chapter 5: Active exploration by searching for experiments that falsify the computed control policy

In this chapter, we still consider a deterministic setting, a continuous state space and a finite action space. The objective is similar to the one of the previous section: determining where to sample additional system transitions.

While the preliminary approach [8] mentioned above in Section 1.4 suffers from its computational complexity, we present in this chapter a different strategy for selecting where to sample new information. This second sampling strategy does not require Lipschitz continuity assumptions on the system dynamics and the reward function. It is based on the intuition that interesting areas of the state-action space where to sample new information are those that are likely to lead to the falsification of the current inferred control policy.

This sampling strategy uses a predictive model $PM$ of the environment to predict the system transitions that are likely to be sampled in any state-action point. Given a state-action point and using predicted data in this point (computing from $PM$) together with already sampled system transitions, a predicted inferred optimal control policy can be computed using a batch mode reinforcement learning algorithm $BMRL$. If this predicted control policy differs from the current control policy (inferred by $BMRL$

from the actual data), then we consider that we have found an interesting point to sample information.

The procedure followed by this iterative sampling strategy to select a state-action point where to sample an additional system transition is summarized below:

- Using the sample $\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^{n}$ of already collected transitions, we first compute a sequence of actions

$$\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = \left(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)\right) = BMRL(\mathcal{F}_n, x_0)\ . \tag{1.21}$$

- Next, we draw a state-action point $(x, u) \in \mathcal{X} \times \mathcal{U}$ according to a uniform probability distribution $p_{\mathcal{X}\times\mathcal{U}}(\cdot)$ over the state-action space $\mathcal{X} \times \mathcal{U}$:

$$(x, u) \sim p_{\mathcal{X}\times\mathcal{U}}(\cdot) \tag{1.22}$$

- Using the sample $\mathcal{F}_n$ and the predictive model $PM$, we then compute a "predicted" system transition by:

$$(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u)) = PM(\mathcal{F}_n, x, u)\ . \tag{1.23}$$

- Using $(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u))$, we build the "predicted" augmented sample by:

$$\hat{\mathcal{F}}_{n+1}(x, u) = \mathcal{F}_n \cup \{(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u))\}\ , \tag{1.24}$$

  and use it to predict the revised policy by:

$$\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) = BMRL(\hat{\mathcal{F}}_{n+1}(x, u), x_0)\ . \tag{1.25}$$

  - If $\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) \neq \tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$, we consider $(x, u)$ as informative, because it is potentially falsifying our current hypothesis about the optimal control policy. We hence use it to make an experiment on the real-system so as to collect a new transition

$$\left(x^{n+1}, u^{n+1}, r^{n+1}, y^{n+1}\right) \tag{1.26}$$

    with

$$\begin{cases} x^{n+1} = x, \\ u^{n+1} = u, \\ r^{n+1} = \rho(x, u), \\ y^{n+1} = f(x, u)\ . \end{cases} \tag{1.27}$$

and we augment the sample with it:

$$\mathcal{F}_{n+1} = \mathcal{F}_n \cup \left\{ \left( x^{n+1}, u^{n+1}, r^{n+1}, y^{n+1} \right) \right\} \ . \tag{1.28}$$

- If $\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) = \tilde{\mathbf{u}}^*_{\mathcal{F}_{\mathbf{n}}}(x_0)$ , we draw another state-action point $(x', u')$ according to $p_{\mathcal{X}\times\mathcal{U}}(\cdot)$:

  $$(x', u') \sim p_{\mathcal{X}\times\mathcal{U}}(\cdot) \tag{1.29}$$

  and repeat the process of prediction followed by policy revision.
- If $L_n \in \mathbb{N}_0$ state-action points have been tried without yielding a potential falsifier of the current policy, we give up and merely draw a state-action point $\left( x^{n+1}, u^{n+1} \right)$ "at random" according to $p_{\mathcal{X}\times\mathcal{U}}(\cdot)$:

  $$\left( x^{n+1}, u^{n+1} \right) \sim p_{\mathcal{X}\times\mathcal{U}}(\cdot) \ , \tag{1.30}$$

  and augment $\mathcal{F}_n$ with the transition

  $$\left( x^{n+1}, u^{n+1}, \rho\left( x^{n+1}, u^{n+1} \right), f\left( x^{n+1}, u^{n+1} \right) \right) \ . \tag{1.31}$$

The paper describing this sampling strategy has been accepted for publication in the Proceedings of the *IEEE Symposium Series on Computational Intelligence - Adaptive Dynamic Programming and Reinforcement Learning* (IEEE ADPRL 2011) [11].

## 1.6 Chapter 6: Model-free Monte Carlo–like policy evaluation

The work mentioned in the previous sections was considering deterministic settings, for which the uncertainties come from the incomplete knowledge of the optimal control problem (system dynamics and reward function) in continuous spaces. In the work presented in Chapter 6, we consider a stochastic setting. We introduce a new way for computing an estimator of the performance of control policies, in a context where both the state and the action space are continuous and normed, and where the system dynamics $f$, the reward function $\rho$ and the probability distribution of the disturbances are unknown (and hence inaccessible to simulation).

In this setting, we chose to evaluate the performance of a given deterministic control policy $h$ through its expected return, defined as follow:

$$J^h(x_0) = \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(\cdot)}{\mathbb{E}} \left[ R^h(x_0) \right] \ , \tag{1.32}$$

where

$$R^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t) \; , \tag{1.33}$$

$$x_{t+1} = f(x_t, h(t, x_t), w_t) \; , \tag{1.34}$$

and where the stochasticity of the control problem is induced by the unobservable random process $w_t \in \mathcal{W}$, which we suppose to be drawn i.i.d. according to a probability distribution $p_{\mathcal{W}}(.)$, $\forall t = 0, \ldots, T-1$.

In such a context, we propose an algorithm that computes from the sample $\mathcal{F}_n$ an estimator of the expected return $J^h(x_0)$ of the given policy $h$ for a given initial state $x_0$ [9]. The estimator - called MFMC for Model-free Monte Carlo - works by selecting $p \in \mathbb{N}$ sequences of transitions of length $T$ from this sample that we call "broken trajectories". These broken trajectories will then serve as proxies for $p$ "actual" trajectories that could be obtained by simulating the policy $h$ on the given control problem. Our estimator averages the cumulated returns over these broken trajectories to compute its estimate of $J^h(x_0)$.

To build a sample of $p$ substitute broken trajectories of length $T$ starting from $x_0$ and similar to trajectories that would be induced by a policy $h$, our algorithm uses each one-step transition in $\mathcal{F}_n$ at most once; we thus assume that $pT \leq n$. The $p$ broken trajectories of $T$ one-step transitions are created sequentially. Every broken trajectory is grown in length by selecting, among the sample of not yet used one-step transitions, a transition whose first two elements minimize the distance $-$ using a distance metric $\Delta$ in $\mathcal{X} \times \mathcal{U}$ $-$ with the couple formed by the last element of the previously selected transition and the action induced by $h$ at the end of this previous transition.

Under some Lipschitz continuity assumptions, the MFMC estimator is shown to behave similarly to a Monte Carlo estimator when the sparsity of the sample of trajectories decreases towards zero. More precisely, one can show that the expected value $E^h_{p,\mathcal{P}_n}(x_0)$ of the MFMC estimator and the variance $V^h_{p,\mathcal{P}_n}(x_0)$ of the MFMC estimator satisfy the following relationships:

$$\left| J^h(x_0) - E^h_{p,\mathcal{P}_n}(x_0) \right| \leq C \alpha_{pT}\left(\mathcal{P}_n\right) \tag{1.35}$$

$$V^h_{p,\mathcal{P}_n}(x_0) \leq \left( \frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C\alpha_{pT}\left(\mathcal{P}_n\right) \right)^2 \tag{1.36}$$

with

$$C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left[ L_f (1 + L_h) \right]^i \; . \tag{1.37}$$

where $L_f$, $L_\rho$ and $L_h$ are upper bounds on the Lipschitz constants of the function $f$, $\rho$ and $h$, respectively, $\sigma^2_{R^h}(x_0)$ is the (supposed finite) variance of $R^h(x_0)$

$$\sigma^2_{R^h}(x_0) = \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}{Var}\left[R^h(x_0)\right] < \infty, \tag{1.38}$$

$p$ is the number of sequences of transitions used to compute the MFMC estimator, $\alpha_{pT}(\mathcal{P}_n)$ is a term that describes the sparsity of the sample of data $\mathcal{F}_n$ which is directly computed from the "projection" $\mathcal{P}_n$ of $\mathcal{F}_n$ on the state-action space.

This work was published in the Proceedings of the *International Conference on Artificial Intelligence and Statistics* (AISTATS 2010). It also received the "Best Student Paper Award" from the French *Conférence Francophone sur l'Apprentissage Artificiel* (CAp2010) [10] where it was presented too.

## 1.7 Chapter 7: Variable selection for dynamic treatment regimes: a reinforcement learning approach

In this chapter, we consider a stochastic framework, and we propose an approach for ranking the relevance of state variables in a batch mode reinforcement learning problem, in order to compute "simplified" control policies that are based on the best ranked variables. This research was initially motivated by the design of dynamic treatment regimes, which may require variable selection techniques in order to simplify the decision rules [14] and lead to more convenient ways to specify treatment for patients.

The approach we have developed for ranking the $n_{\mathcal{X}} \in \mathbb{N}_0$ state variables of the optimal control problem [12] exploits a variance reduction–type criterion which can be extracted from the solution of the batch mode reinforcement learning problem using the Fitted $Q$ Iteration algorithm [3] with ensembles of regression trees [13] (the fitted $Q$ iteration algorithm is fully specified in Appendix A). When running the fitted $Q$ iteration algorithm, a sequence of approximated value functions $\tilde{Q}_1, \ldots, \tilde{Q}_T$ is built from $T$ ensembles of trees, and, using these ensembles of trees, the ranking approach evaluates the relevance of each state variable $x(i) \quad i = 1 \ldots n_{\mathcal{X}}$ by the score function:

$$S(x(i)) = \frac{\sum_{N=1}^{T} \sum_{\tau\in\tilde{Q}_N} \sum_{\nu\in\tau} \delta\left(\nu, x(i)\right) \Delta_{var}(\nu)|\nu|}{\sum_{N=1}^{T} \sum_{\tau\in\tilde{Q}_N} \sum_{\nu\in\tau} \Delta_{var}(\nu)|\nu|} \tag{1.39}$$

where $\nu$ is a nonterminal node in a tree $\tau$, $\delta(\nu, x(i)) = 1$ if $x(i)$ is used to split at node $\nu$ or equal to zero otherwise, $|\nu|$ is the number of samples at node $\nu$, $\Delta_{var}(\nu)$ is the variance reduction when splitting node $\nu$:

$$\Delta_{var}(\nu) = v(\nu) - \frac{|\nu_L|}{|\nu|}v(\nu_L) - \frac{|\nu_R|}{|\nu|}v(\nu_R) \tag{1.40}$$

where $\nu_L$ (resp. $\nu_R$) is the left-son node (resp. the right-son node) of node $\nu$, and $v(\nu)$ (resp. $v(\nu_L)$ and $v(\nu_R)$) is the variance of the sample at node $\nu$ (resp. $\nu_L$ and $\nu_R$).

The approach then sorts the state variables $x(i)$ by decreasing values of their score so as to identify the $m_{\mathcal{X}} \in \mathbb{N}_0$ most relevant ones. A simplified control policy defined on this subset of variables is then computed by running the fitted $Q$ iteration algorithm again on a modified sample of system transitions, where the state variables of $x^l$ and $y^l$ that are not among these $m_{\mathcal{X}}$ most relevant ones are discarded.

The algorithm for computing a simplified control policy defined on a small subset of state variables is thus as follows:

1. Compute the $\tilde{Q}_N$-functions ($N = 1, \dots, T$) using the fitted $Q$ iteration algorithm on $\mathcal{F}_n$;

2. Compute the score function for each state variable, and determine the $m_{\mathcal{X}}$ best ones;

3. Run the fitted $Q$ iteration algorithm on

$$\widetilde{\mathcal{F}}_n = \left\{ \left( \widetilde{x}^l, u^l, r^l, \widetilde{y}^l \right) \right\}_{l=1}^{n} \tag{1.41}$$

where

$$\widetilde{x} = \widetilde{M}x, \tag{1.42}$$

and $\widetilde{M}$ is a $m_{\mathcal{X}} \times n_{\mathcal{X}}$ boolean matrix where $\widetilde{m}_{i,j} = 1$ if the state variable $x(j)$ is the $i$-th most relevant one and 0 otherwise.

This work [12] was presented as a short paper at the *European Workshop on Reinforcement Learning* (EWRL 2008).

## 1.8 List of publications

As mentioned above, the present dissertation is a collection of research publications. These research publications are:

- R. Fonteneau, L. Wehenkel, and D. Ernst. Variable selection for dynamic treatment regimes: a reinforcement learning approach. In *European Workshop on Reinforcement Learning (EWRL 2008)*, Villeneuve d'Ascq, France, 2008.
- R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. A cautious approach to generalization in reinforcement learning. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010)*, Valencia, Spain, 2010.
- R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards $\min\max$ generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers*. Series: Communications in Computer and Information Science (CCIS), volume 129, pages 61-77. Springer, Heidelberg, 2011.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, JMLR: W & CP 9, pages 217-224, Chia Laguna, Sardinia, Italy, 2010.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Generating informative trajectories by using bounds on the return of control policies. In *Proceedings of the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*, Chia Laguna, Sardinia, Italy, 2010.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Actes de la conférence francophone sur l'apprentissage automatique (CAP 2010)*, Clermont-Ferrand, France, 2010.
- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Estimation Monte Carlo sans modèle de politiques de décision. To be published in *Revue d'Intelligence Artificielle*.

- R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Active exploration by searching for experiments falsifying an already induced policy. To be published in *Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2011)*, Paris, France, 2011.

In addition to the publications listed above, I have coauthored the following papers, not directly related to batch mode reinforcement learning, during my PhD thesis:

- G.B. Stan, F. Belmudes, R. Fonteneau, F. Zeggwagh, M.A. Lefebvre, C. Michelet and D. Ernst. Modelling the influence of activation-induced apoptosis of CD4+ and CD8+ T-cells on the immune system response of a HIV infected patient. In *IET Systems Biology 2008*, March 2008 - Volume 2, Issue 2, p. 94-102.

- M.J. Mhawej, C.B. Brunet-François, R. Fonteneau, D. Ernst, V. Ferré, G.B. Stan, F. Raffi and C.H. Moog. Apoptosis characterizes immunological failure of HIV infected patients. In *Control Engineering Practice 17 (2009)*, p. 798-804.

- P.S. Rivadeneira, M.-J. Mhawej, C.H. Moog, F. Biafore, D.A. Ouattara, C. Brunet-Francois, V. Ferre, D. Ernst, R. Fonteneau, G.-B. Stan, F. Bugnon, F. Raffi, X. Xia. Mathematical modeling of HIV dynamics after antiretroviral therapy initiation. Submitted.

# Bibliography

[1] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[2] S.J. Bradtke and A.G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.

[3] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[4] D. Ernst, G.B. Stan, J. Goncalves, and L. Wehenkel. Clinical data based optimal STI strategies for HIV: a reinforcement learning approach. In *Machine Learning Conference of Belgium and The Netherlands.*, pages page 65–72, 2006.

[5] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.

[6] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series: Communications in Computer and Information Science (CCIS)*, volume 129, pages 61–77. Springer, Heidelberg, 2011.

[7] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. A cautious approach to generalization in reinforcement learning. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010)*, Valencia, Spain, 2010.

[8] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Generating informative trajectories by using bounds on the return of control policies. In *Proceedings of*

*the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*, 2010.

[9] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, JMLR: W&CP 9*, pages 217–224, Chia Laguna, Sardinia, Italy, 2010.

[10] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Actes de la conférence francophone sur l'apprentissage automatique (CAP 2010), Clermont-Ferrand (France)*, 2010.

[11] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Active exploration by searching for experiments falsifying an already induced policy. *To be published in the Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2011), Paris, France*, 2011.

[12] R. Fonteneau, L. Wehenkel, and D. Ernst. Variable selection for dynamic treatment regimes: a reinforcement learning approach. In *European Workshop on Reinforcement Learning (EWRL 2008)*, Villeneuve d'Ascq, France, 2008.

[13] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning.*, 36(Number 1):3–42, 2006.

[14] L. Gunter, J. Zhu, and S.A. Murphy. *Artificial Intelligence in Medicine.*, volume 4594/2007, chapter Variable Selection for Optimal Decision Making, pages 149–154. Springer Berlin / Heidelberg, 2007.

[15] M.G. Lagoudakis and R. Parr. Least-squares policy iteration. *Jounal of Machine Learning Research*, 4:1107–1149, 2003.

[16] S.A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, 65(2):331–366, 2003.

[17] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[18] S.A. Murphy and D. Almirall. Dynamic Treatment Regimes. *Encyclopedia of Medical Decision Making*, 2008.

[19] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[20] M. Qian, I. Nahum-Shani, and S.A. Murphy. Dynamic treatment regimes. *To appear as a book chapter in Modern Clinical Trial Analysis , edited by X. Tu and W. Tang, Springer Science*, 2009.

[21] M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML 2005)*, pages 317–328, Porto, Portugal, 2005.

[22] R.S. Sutton. Learning to predict by the methods of temporal difference. *Machine Learning*, 3:9–44, 1988.

[23] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, 1998.

# Chapter 2

# Inferring bounds on the performance of a control policy from a sample of trajectories

*We propose an approach for inferring bounds on the finite-horizon return of a control policy from an off-policy sample of trajectories collecting state transitions, rewards, and control actions. In this chapter, the dynamics, control policy, and reward function are supposed to be deterministic and Lipschitz continuous. Under these assumptions, a polynomial algorithm, in terms of the sample size and length of the optimization horizon, is derived to compute these bounds, and their tightness is characterized in terms of the sample density.*

The work presented in this chapter has been published in the *Proceedings of the IEEE Symposium Series in Computational Intelligence - Adaptive Dynamic Programming and Reinforcement Learning* (IEEE ADPRL 2009) [6].

In this chapter, we consider:

- a deterministic framework,
- a continuous state-action space.

## 2.1 Introduction

In financial [7], medical [10] and engineering sciences [1], as well as in artificial intelligence [14], variants (or generalizations) of the following discrete-time optimal control problem arise quite frequently: a system, characterized by its state-transition function

$$x_{t+1} = f(x_t, u_t) \tag{2.1}$$

should be controlled by using a policy $u_t = h(t, x_t)$ so as to maximize a cumulated reward

$$\sum_{t=0}^{T-1} \rho(x_t, u_t) \tag{2.2}$$

over a finite optimization horizon $T$.

Among the solution approaches that have been proposed for this class of problems we have, on the one hand, dynamic programming [1] and model predictive control [3] which compute optimal solutions from an analytical or computational model of the real system, and, on the other hand, reinforcement learning approaches [14, 9, 5, 12] which compute approximations of optimal control policies based only on data gathered from the real system. In between, we have approximate dynamic programming approaches which use datasets generated by using a model (e.g. by Monte Carlo simulation) so as to derive approximate solutions while complying with computational requirements [2].

Whatever the approach (model-based, data-based, Monte Carlo-based, (or even finger-based)) used to derive a control policy for a given problem, one major question that remains open today is to ascertain the *actual* performance of the derived control policy [8, 13] when applied to the *real* system behind the model or the dataset (or the finger). Indeed, for many applications, even if it is perhaps not paramount to have a policy $h$ which is very close to the optimal one, it is however crucial to be able to guarantee that the considered policy $h$ leads for some initial states $x_0$ to high-enough cumulated rewards on the real system that is considered.

In this chapter, we thus focus on the evaluation of control policies on the sole basis of the actual behavior of the concerned real system. We use to this end a sample of trajectories

$$(x_0, u_0, r_0, x_1, \ldots, r_{T-1}, x_T) \tag{2.3}$$

gathered from interactions with the real system, where states $x_t \in \mathcal{X}$, actions $u_t \in \mathcal{U}$ and instantaneous rewards

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{2.4}$$

at successive discrete instants $t = 0, 1, \ldots, T-1$ will be exploited so as to evaluate bounds on the performance of a given control policy

$$h : \{0, 1, \ldots, T-1\} \times \mathcal{X} \to \mathcal{U} \tag{2.5}$$

when applied to a given initial state $x_0$ of the real system.

Actually, our proposed approach does not require full-length trajectories since it relies only on a set of $n \in \mathbb{N}_0$ one-step system transitions

$$\mathcal{F}_n = \left\{ \left( x^l, u^l, r^l, y^l \right) \right\}_{l=1}^{n}, \tag{2.6}$$

each one providing the knowledge of a sample of information $(x, u, r, y)$, named four-tuple, where $y$ is the state reached after taking action $u$ in state $x$ and $r$ the instantaneous reward associated with the transition. We however assume that the state and action spaces are normed and that the system dynamics ($y = f(x, u)$) and the reward function ($r = \rho(x, u)$) and control policy ($u = h(t, x)$) are deterministic and Lipschitz continuous.

In a few words, the approach works by identifying in $\mathcal{F}_n$ a sequence of $T$ four-tuples

$$\left[ \left( x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0} \right), \left( x^{l_1}, u^{l_1}, r^{l_1}, y^{l_1} \right), \ldots, \left( x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}} \right) \right] \tag{2.7}$$

(with $l_t \in \{1, \ldots, n\}$), which maximizes a specific numerical criterion. This criterion is made of the sum of the $T$ rewards corresponding to these four-tuples

$$\sum_{t=0}^{T-1} r^{l_t} \tag{2.8}$$

and $T$ negative terms. The negative term corresponding to the four-tuple

$$(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}) \quad t \in \{0, \ldots, T-1\} \tag{2.9}$$

of the sequence represents an upper bound variation of the cumulated rewards over the remaining time steps that can occur by simulating the system from a state $x^{l_t}$ rather than $y^{l_{t-1}}$ (with $y^{l_{-1}} = x_0$) and by using at time $t$ the action $u^{l_t}$ rather than $h(t, y^{l_{t-1}})$. We provide a polynomial algorithm to compute this optimal sequence of tuples and derive a tightness characterization of the corresponding performance bound in terms of the density of the sample $\mathcal{F}_n$.

The rest of this chapter is organized as follows. In Section 2.2, we formalize the problem considered in this chapter. In Section 2.3, we show that the state-action value function of a policy over the $N$ last steps of an episode is Lipschitz continuous. Section 2.4 uses this result to compute from a sequence of four-tuples a lower bound on the cumulated reward obtained by a policy $h$ when starting from a given $x_0 \in \mathcal{X}$, while Section 2.5 proposes a polynomial algorithm for identifying the sequence of four-tuples which leads to the best bound. Section 2.6 studies the tightness of this bound and shows that it can be characterized by $C\alpha^*_{\mathcal{F}_n}$ where $C$ is a positive constant and $\alpha^*_{\mathcal{F}_n}$ is the maximum distance between any element of the state-action space $\mathcal{X} \times \mathcal{U}$ and its closest state-action pair $(x^l, u^l) \in \mathcal{F}_n$. Finally, Section 2.7 concludes and outlines directions for future research.

## 2.2 Formulation of the problem

We consider a discrete-time system whose dynamics over $T$ stages is described by a time-invariant equation:

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \ldots, T-1, \tag{2.10}$$

where for all $t$, the state $x_t$ is an element of the state space $\mathcal{X}$ and the action $u_t$ is an element of the action space $\mathcal{U}$ (both $\mathcal{X}$ and $\mathcal{U}$ are assumed to be normed vector spaces). $T \in \mathbb{N}_0$ is referred to as the *optimization horizon*. The transition from $t$ to $t+1$ is associated with an instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R}. \tag{2.11}$$

For every initial state $x_0$ and for every sequence of actions $(u_0, u_1, \ldots, u_{T-1}) \in \mathcal{U}^T$, the cumulated reward over $T$ stages (also named return over $T$ stages) is defined as follows:

**Definition 2.2.1 ($T-$stage return of the sequence $(u_0, \ldots, u_{T-1})$)**
$\forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$J^{u_0,\ldots,u_{T-1}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t). \tag{2.12}$$

We consider in this chapter deterministic time-varying $T$-stage policies

$$h : \{0, 1, \ldots, T-1\} \times X \to U \tag{2.13}$$

which select at time $t$ the action $u_t$ based on the current time and the current state ($u_t = h(t, x_t)$). The return over $T$ stages of a policy $h$ from a state $x_0$ is defined as follows:

**Definition 2.2.2** (**$T-$stage return of the policy $h$**)
$\forall x_0 \in \mathcal{X}$,

$$J^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t)) \tag{2.14}$$

*where*

$$\forall t \in \{0, \dots, T-1\}, x_{t+1} = f(x_t, h(t, x_t)) \ . \tag{2.15}$$

We also assume that the dynamics $f$, the reward function $\rho$ and the policy $h$ are Lipschitz continuous:

**Assumption 2.2.3 (Lipschitz continuity of $f$, $\rho$ and $h$)**
*There exist finite constants $L_f, L_\rho, L_h \in \mathbb{R}$ such that:*

$\forall (x, x') \in \mathcal{X}^2, \forall (u, u') \in \mathcal{U}^2, \forall t \in \{0, \dots, T-1\}$,

$$\begin{aligned} \|f(x,u) - f(x',u')\|_{\mathcal{X}} &\leq L_f\big(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}\big), & (2.16)\\ |\rho(x,u) - \rho(x',u')| &\leq L_\rho\big(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}\big), & (2.17)\\ \|h(t,x) - h(t,x')\|_{\mathcal{U}} &\leq L_h\|x - x'\|_{\mathcal{X}}, & (2.18) \end{aligned}$$

*where $\|.\|_{\mathcal{X}}$ (resp. $\|.\|_{\mathcal{U}}$) denotes the chosen norm over the space $\mathcal{X}$ (resp. $\mathcal{U}$). The smallest constants satisfying those inequalities are named the Lipschitz constants.*

We further suppose that:

**Assumption 2.2.4**

1. *The system dynamics $f$ and the reward function $\rho$ are unknown,*
2. *An arbitrary set of $n$ one-step system transitions (also named four-tuples)*

$$\mathcal{F}_n = \big\{(x^l, u^l, r^l, y^l)\big\}_{l=1}^{n} \tag{2.19}$$

*is known. Each four-tuple is such that*

$$\begin{cases} y^l = f(x^l, u^l), \\ r^l = \rho(x^l, u^l). \end{cases} \tag{2.20}$$

3. *Three constants $L_f$, $L_\rho$, $L_h$ satisfying the above-written Lipschitz inequalities are known. These constants do not necessarily have to be the smallest ones satisfying these inequalities (i.e., the Lipschitz constants), even if, the smaller they are, the tighter the bound will be.*

Under these assumptions, we want to find for an arbitrary initial state $x_0$ of the system a *lower bound* on the return over $T$ stages of any given policy $h$.

## 2.3 Lipschitz continuity of the state-action value function

For $N = 1, \ldots, T$, let us define the family of functions $Q_N^h : X \times U \to \mathbb{R}$ as follows:

**Definition 2.3.1 (State-action value functions)**
$\forall N \in \{1, \ldots, T\}\ , \forall (x, u) \in \mathcal{X} \times \mathcal{U}$,

$$Q_N^h(x,u) = \rho(x,u) + \sum_{t=T-N+1}^{T-1} \rho\left(x_t, h(t, x_t)\right), \tag{2.21}$$

*where*

$$x_{T-N+1} = f(x,u). \tag{2.22}$$

*and*

$$\forall t \in \{T-N+1, \ldots, T-1\}\, , x_{t+1} = f(x_t, h(t, x_t))\, . \tag{2.23}$$

$Q_N^h(x,u)$ gives the sum of rewards from instant $t = T - N$ to instant $T - 1$ when

- The system is in state $x$ at instant $T - N$,
- The action chosen at instant $T - N$ is $u$,
- The actions are selected at subsequent instants according to the policy $h$:

$$\forall t \in \{T-N+1, \ldots, T-1\}\, , u_t = h(t, x_t). \tag{2.24}$$

We have the following trivial propositions:

**Proposition 2.3.2**
*The function $J^h$ can be deduced from $Q_N^h$ as follows:*

$$\forall x_0 \in \mathcal{X},\ J^h(x_0) = Q_T^h(x_0, h(0, x_0)). \tag{2.25}$$

**Proposition 2.3.3**
$\forall N \in \{1, \ldots, T-1\}, \forall (x, u) \in \mathcal{X} \times \mathcal{U}$,

$$Q^h_{N+1}(x, u) = \rho(x, u) + Q^h_N(f(x, u), h(T-N, f(x, u)). \tag{2.26}$$

We prove hereafter the Lipschitz continuity of $Q^h_N, \forall N \in \{1, \ldots, T\}$.

**Lemma 2.3.4 (Lipschitz continuity of $Q^h_N$)**
$\forall N \in \{1, \ldots, T\}$, $\exists L_{Q_N} \in \mathbb{R}^+$,
$\forall (x, x') \in \mathcal{X}^2, \forall (u, u') \in \mathcal{U}^2$,

$$\left| Q^h_N(x, u) - Q^h_N(x', u') \right| \leq L_{Q_N} \big( \|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}} \big). \tag{2.27}$$

**Proof.** We consider the statement $\mathcal{H}(N)$:
$\exists L_{Q_N} \in \mathbb{R}^+ : \forall (x, x') \in \mathcal{X}^2, \forall (u, u') \in \mathcal{U}^2$,

$$\left| Q^h_N(x, u) - Q^h_N(x', u') \right| \leq L_{Q_N} \big( \|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}} \big). \tag{2.28}$$

We prove by induction that $\mathcal{H}(N)$ is true $\forall N \in \{1, \ldots, T\}$. For the sake of clarity, we use the notation:

$$\Delta_N = \left| Q^h_N(x, u) - Q^h_N(x', u') \right|. \tag{2.29}$$

- Basis: $N = 1$

We have

$$\Delta_1 = |\rho(x, u) - \rho(x', u')|, \tag{2.30}$$

and the Lipschitz continuity of $\rho$ allows to write

$$\Delta_1 \leq L_{Q_1} \big( \|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}} \big), \tag{2.31}$$

with $L_{Q_1} \doteq L_\rho$. This proves $\mathcal{H}(1)$.

- Induction step: we suppose that $\mathcal{H}(N)$ is true, $1 \leq N \leq T - 1$.

Using Proposition (2.3.3), we can write

$$\begin{aligned} \Delta_{N+1} &= \left| Q^h_{N+1}(x, u) - Q^h_{N+1}(x', u') \right| && (2.32) \\ &= \big| \rho(x, u) - \rho(x', u') + Q^h_N(f(x, u), h(T-N, f(x, u))) \\ &- Q^h_N(f(x', u'), h(T-N, f(x', u'))) \big| && (2.33) \end{aligned}$$

and, from there,

$$\begin{aligned} \Delta_{N+1} &\leq \big|\rho(x,u) - \rho(x',u')\big| \\ &+ \big|Q_N^h(f(x,u), h(T-N, f(x,u))) \\ &- Q_N^h(f(x',u'), h(T-N, f(x',u')))\big|. \end{aligned} \tag{2.34}$$

$\mathcal{H}(N)$ and the Lipschitz continuity of $\rho$ give

$$\begin{aligned} \Delta_{N+1} &\leq L_\rho\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big) \\ &+ L_{Q_N}\Bigg(\|f(x,u) - f(x',u')\|_{\mathcal{X}} \\ &+ \|h(T-N, f(x,u)) - h(T-N, f(x',u'))\|_{\mathcal{U}}\Bigg). \end{aligned} \tag{2.35}$$

Using the Lipschitz continuity of $f$ and $h$, we have

$$\begin{aligned} \Delta_{N+1} &\leq L_\rho\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big) \\ &+ L_{Q_N}\Bigg(L_f\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big) + L_h L_f\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big)\Bigg), \end{aligned} \tag{2.36}$$

and, from there,

$$\Delta_{N+1} \leq L_{Q_{N+1}}\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big), \tag{2.37}$$

with

$$L_{Q_{N+1}} \doteq L_\rho + L_{Q_N} L_f (1 + L_h)\ . \tag{2.38}$$

This proves that $\mathcal{H}\ (N+1)$ is true, and ends the proof. ■

Let $L^*_{Q_N}$ be the Lipschitz constant of the function $Q_N^h$, that is the smallest value of $L_{Q_N}$ that satisfies inequality (2.27). We have the following result:

**Lemma 2.3.5 (Upper bound on $L^*_{Q_N}$)**
$\forall N \in \{1, \ldots, T\}$,

$$L^*_{Q_N} \leq L_\rho\Bigg(\sum_{t=0}^{N-1} \left[L_f(1+L_h)\right]^t\Bigg) \tag{2.39}$$

**Proof.** A sequence of positive constants $L_{Q_1}, \ldots, L_{Q_N}$ is defined in the proof of Lemma 2.3.4. Each constant $L_{Q_N}$ of this sequence is an upper-bound on the Lipschitz constant related to the function $Q^h_N$. These $L_{Q_N}$ constants satisfy the relationship

$$L_{Q_{N+1}} = L_\rho + L_{Q_N} L_f (1 + L_h) \tag{2.40}$$

(with $L_{Q_1} = L_\rho$) from which the lemma can be proved in a straightforward way. ■

The value of the constant $L_{Q_N}$ will influence the lower bound on the return of the policy $h$ that will be established later in this chapter. The larger this constant, the looser the bounds. When using these bounds, $L_{Q_N}$ should therefore preferably be chosen as small as possible while still ensuring that inequality (2.27) is satisfied. Later in this chapter, we will use the upper bound (2.39) to select a value for $L_{Q_N}$. More specifically, we will choose

$$L_{Q_N} = L_\rho \bigg( \sum_{t=0}^{N-1} [L_f(1 + L_h)]^t \bigg). \tag{2.41}$$

## 2.4 Computing a lower bound on $J^h(x_0)$ from a sequence of four-tuples

The algorithm described in Table 1 provides a way of computing from any $T$-length sequence of four-tuples

$$\tau = \left[ \left( x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t} \right) \right]_{t=0}^{T-1} \tag{2.42}$$

a lower bound on $J^h(x_0)$, provided that the initial state $x_0$, the policy $h$ and three constants $L_f$, $L_\rho$ and $L_h$ satisfying inequalities (2.16 - 2.18) are given. The algorithm is a direct consequence of Theorem 2.4.1 below.

The lower bound on $J^h(x_0)$ derived in Theorem 2.4.1 can be interpreted as follows. The sum of the rewards of the "broken" trajectory formed by the sequence of four-tuples $\tau$ can never be greater than $J^h(x_0)$, provided that every reward $r^{l_t}$ is penalized by a factor

$$L_{Q_{T-t}} \left( \left\| x^{l_t} - y^{l_{t-1}} \right\|_{\mathcal{X}} + \left\| u^{l_t} - h(t, y^{l_{t-1}}) \right\|_{\mathcal{U}} \right). \tag{2.43}$$

This factor is in fact an upper bound on the variation of the function $Q^h_{T-t}$ that can occur when "jumping" from $\left( y^{l_t}, h(t, y^{l_t}) \right)$ to $\left( x^{l_{t+1}}, u^{l_{t+1}} \right)$. An illustration of this interpretation is given in Figure 2.1.

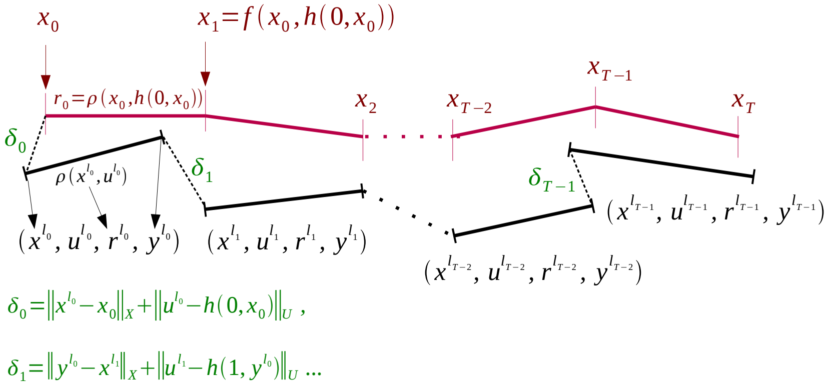
$x_0$
$x_1=f(x_0,h(0,x_0))$
$r_0=\rho(x_0,h(0,x_0))$
$x_2$
$x_{T-2}$
$x_{T-1}$
$x_T$
$\delta_0$
$\rho(x^{l_0},u^{l_0})$
$\delta_1$
$\delta_{T-1}$
$(x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0})$
$(x^{l_1}, u^{l_1}, r^{l_1}, y^{l_1})$
$(x^{l_{T-2}}, u^{l_{T-2}}, r^{l_{T-2}}, y^{l_{T-2}})$
$(x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}})$
$\delta_0=\|x^{l_0}-x_0\|_X+\|u^{l_0}-h(0,x_0)\|_U$ ,
$\delta_1=\|y^{l_0}-x^{l_1}\|_X+\|u^{l_1}-h(1,y^{l_0})\|_U$ ...

Figure 2.1: A graphical interpretation of the different terms composing the bound on $J^h(x_0)$ inferred from a sequence of four-tuples (see Equation (2.45)). The bound is equal to the sum of all the rewards corresponding to this sequence of four-tuples (the terms $r^{l_t}$ $t = 0, 1, \ldots, T-1$ on the figure) minus the sum of all the terms $L_{Q_{T-t}}\delta_t$.

**Algorithm 1** An algorithm for computing from a sequence of four-tuples $\tau$ a lower bound on $J^h(x_0)$.

```
Inputs:
An initial state $x_0$,
A policy $h$,
A sequence of four-tuples $\tau = \left[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})\right]_{t=0}^{T-1}$,
Three constants $L_f$, $L_\rho$, $L_h$ which satisfy inequalities (2.16 - 2.18) ;
Output: A lower bound on $J^h(x_0)$;
Algorithm:
$lb \leftarrow 0$ ;
$y^{l_{-1}} \leftarrow x_0$ ;
for $t = 0$ to $T-1$ do
    $L_{Q_{T-t}} \leftarrow L_\rho \Bigg( \sum_{k=0}^{T-t-1} [L_f(1+L_h)]^k \Bigg)$ ;
    $lb \leftarrow lb + r^{l_t} - L_{Q_{T-t}} \big( \|x^{l_t} - y^{l_{t-1}}\|_{\mathcal{X}} + \|u^{l_t} - h(t, y^{l_{t-1}})\|_{\mathcal{U}} \big)$ ;
end for
Return: $lb$.
```

**Theorem 2.4.1 (Lower bound on $J^h(x_0)$)**
*Let $x_0$ be an initial state of the system, $h$ a policy, and $\tau$ a sequence of tuples:*

$$\tau = \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} . \tag{2.44}$$

*Then we have the following lower bound on $J^h(x_0)$:*

$$\sum_{t=0}^{T-1} (r^{l_t} - L_{Q_{T-t}} \delta_t) \leq J^h(x_0), \tag{2.45}$$

*where*

$$\forall t \in \{0, 1, \ldots, T-1\}, \delta_t = \left\|x^{l_t} - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u^{l_t} - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} \tag{2.46}$$

*with $y^{l_{-1}} = x_0$.*

**Proof.** Using Proposition (2.3.2) and the Lipschitz continuity of $Q^h_T$, we can write

$$\left|Q^h_T(x_0, u_0) - Q^h_T\left(x^{l_0}, u^{l_0}\right)\right| \leq L_{Q_T} \left(\left\|x_0 - x^{l_0}\right\|_{\mathcal{X}} + \left\|u_0 - u^{l_0}\right\|_{\mathcal{U}}\right), \tag{2.47}$$

and, with $u_0 = h(0, x_0)$,

$$\begin{aligned} \left|J^h(x_0) - Q_T^h\left(x^{l_0}, u^{l_0}\right)\right| &= \left|Q_T^h\left(x_0, h(0,x_0)\right) - Q_T^h\left(x^{l_0}, u^{l_0}\right)\right| && (2.48) \\ &\leq L_{Q_T}\left(\left\|x_0 - x^{l_0}\right\|_{\mathcal{X}} + \left\|h(0,x_0) - u^{l_0}\right\|_{\mathcal{U}}\right). && (2.49) \end{aligned}$$

It follows that

$$Q_T^h\left(x^{l_0}, u^{l_0}\right) - L_{Q_T}\delta_0 \leq J^h(x_0). \tag{2.50}$$

By definition of the state-action evaluation function $Q_T^h$, we have

$$Q_T^h\left(x^{l_0}, u^{l_0}\right) = \rho\left(x^{l_0}, u^{l_0}\right) + Q_{T-1}^h\left(f\left(x^{l_0}, u^{l_0}\right), h\left(1, f\left(x^{l_0}, u^{l_0}\right)\right)\right) \tag{2.51}$$

and from there

$$Q_T^h\left(x^{l_0}, u^{l_0}\right) = r^{l_0} + Q_{T-1}^h\left(y^{l_0}, h(1, y^{l_0})\right). \tag{2.52}$$

Thus,

$$Q_{T-1}^h\left(y^{l_0}, h(1, y^{l_0})\right) + r^{l_0} - L_{Q_T}\delta_0 \leq J^h(x_0). \tag{2.53}$$

By using the Lipschitz continuity of the function $Q_{T-1}^h$, we can write

$$\begin{aligned} &\left|Q_{T-1}^h(y^{l_0}, h(1, y^{l_0})) - Q_{T-1}^h(x^{l_1}, u^{l_1})\right| \\ &\leq L_{Q_{T-1}}\left(\left\|y^{l_0} - x^{l_1}\right\|_{\mathcal{X}} + \left\|h(1, y^{l_0}) - u^{l_1}\right\|_{\mathcal{U}}\right) && (2.54) \\ &= L_{Q_{T-1}}\delta_1, && (2.55) \end{aligned}$$

which implies that

$$Q_{T-1}^h\left(x^{l_1}, u^{l_1}\right) - L_{Q_{T-1}}\delta_1 \leq Q_{T-1}^h\left(y^{l_0}, h(1, y^{l_0})\right). \tag{2.56}$$

We have therefore

$$Q_{T-1}^h\left(x^{l_1}, u^{l_1}\right) + r^{l_0} - L_{Q_T}\delta_0 - L_{Q_{T-1}}\delta_1 \leq J^h(x_0). \tag{2.57}$$

By iterating this derivation, we obtain inequality (2.45). ■

## 2.5 Finding the highest lower bound

Let

$$B^h(\tau, x_0) = \sum_{t=0}^{T-1} \left[ r^{l_t} - L_{Q_{T-t}} \delta_t \right] , \tag{2.58}$$

with

$$\delta_t = \left\| x^{l_t} - y^{l_{t-1}} \right\|_{\mathcal{X}} + \left\| u^{l_t} - h(t, y^{l_{t-1}}) \right\|_{\mathcal{U}} , \tag{2.59}$$

be the function that maps a $T$-length sequence of four-tuples $\tau$ and the initial state of the system $x_0$ into the lower bound on $J^h(x_0)$ proved by Theorem 2.4.1.

Let ${\mathcal{F}_n}^T$ denote the set of all possible $T$-length sequences of four-tuples built from the elements of $\mathcal{F}_n$, and let $L^h_{\mathcal{F}_n}(x_0)$ be defined as follows:

$$L^h_{\mathcal{F}_n}(x_0) = \max_{\tau \in {\mathcal{F}_n}^T} B^h(\tau, x_0) . \tag{2.60}$$

In this section, we provide an algorithm for computing in an efficient way the value of $L^h_{\mathcal{F}_n}(x_0)$. A naive approach for computing this value would consist in doing an exhaustive search over all the elements of ${\mathcal{F}_n}^T$. However, as soon as the optimization horizon $T$ grows, this approach becomes computationally impractical even if $\mathcal{F}_n$ has only a handful of elements.

Our algorithm for computing $L^h_{\mathcal{F}_n}(x_0)$ is summarized in Table 2. It is in essence identical to the Viterbi algorithm [15], and we observe that its complexity is linear with respect to the optimization horizon $T$ and quadratic with respect to the size $n$ of the sample of four-tuples.

The rationale behind this algorithm is the following. Let us first introduce some notations. Let $\tau(i)$ denote the index of the $i$th four-tuple of the sequence $\tau$ ($\tau(i) = l_i$), let

$$B^h(\tau, x_0)(j) = \sum_{t=0}^{j} (r^{l_t} - L_{Q_{T-t}} \delta_t) \tag{2.61}$$

and let $\tau^*$ be a sequence of tuples such that

$$\tau^* \in \arg\max_{\tau \in {\mathcal{F}_n}^T} B^h(\tau, x_0). \tag{2.62}$$

We have that

$$L^h_{\mathcal{F}_n}(x_0) = B^h(\tau^*, x_0)(T-2) + V_1(\tau^*(T-1)) \tag{2.63}$$

where $V_1$ is a $n$-dimensional vector whose $i-$th component is:

$$\max_{i'}\big(r^{i'} - L_{Q_1}\big(\|x^{i'} - y^i\|_{\mathcal{X}} + \|u^{i'} - h(T-1, y^i)\|_{\mathcal{U}}\big)\big). \tag{2.64}$$

Now let use observe that:

$$L^h_{\mathcal{F}_n}(x_0) = B^h(\tau^*, x_0)(T-3) + V_2(\tau^*(T-2)) \tag{2.65}$$

where $V_2$ is a $n$-dimensional vector whose $i$th component is:

$$\max_{i'}\big(r^{i'} - L_{Q_2}\big(\|x^{i'} - y^i\|_{\mathcal{X}} + \|u^{i'} - h(T-2, y^i)\|_{\mathcal{U}}\big) + V_1(i')\big). \tag{2.66}$$

By proceeding recursively, it is therefore possible to determine the value of

$$B^h(\tau^*, x_0) = L^h_{\mathcal{F}_n}(x_0) \tag{2.67}$$

without having to screen all the elements of ${\mathcal{F}_n}^T$.

Although this is rather evident, we want to stress the fact that $L^h_{\mathcal{F}_n}(x_0)$ can not decrease when new elements are added to $\mathcal{F}_n$. In other words, the quality of this lower bound is monotonically increasing when new samples are collected. To quantify this behavior, we characterize in the next section the tightness of this lower bound as a function of the density of the sample of four-tuples.

## 2.6 Tightness of the lower bound $L^h_{\mathcal{F}_n}(x_0)$

In this section we study the relation of the tightness of $L^h_{\mathcal{F}_n}(x_0)$ with respect to the distance between the elements $(x, u) \in \mathcal{X} \times \mathcal{U}$ and the pairs $(x^l, u^l)$ formed by the two first elements of the four-tuples composing $\mathcal{F}_n$. We prove in Theorem 2.6.1 that if $\mathcal{X} \times \mathcal{U}$ is bounded, then

$$J^h(x_0) - L^h_{\mathcal{F}_n}(x_0) \leq C\alpha^*_{\mathcal{F}_n}, \tag{2.68}$$

where $C$ is a constant depending only on the control problem and where $\alpha^*_{\mathcal{F}_n}$ is the maximum distance from any $(x, u) \in \mathcal{X} \times \mathcal{U}$ to its closest neighbor in $\big\{(x^l, u^l)\big\}_{l=1}^n$.

The main philosophy behind the proof is the following. First, a sequence of four-tuples whose state-action pairs $(x^{l_t}, u^{l_t})$ stand close to the different state-action pairs $(x_t, u_t)$ visited when the system is controlled by $h$ is built. Then, it is shown that the lower bound $B$ computed when considering this particular sequence is such that

$$J^h(x_0) - B \leq C\alpha^*_{\mathcal{F}_n}. \tag{2.69}$$

From there, the proof follows immediately.

**Algorithm 2** A Viterbi-like algorithm for computing the highest lower bound $L^h_{\mathcal{F}_n}(x_0)$ (see Eqn (2.58)) over all the sequences of four-tuples $\tau$ made from elements of $\mathcal{F}_n$.

```
**Inputs:**
An initial state $x_0$,
A policy $h$,
A set of four-tuples $\mathcal{F}_n = \{(x^l, u^l, r^l, y^l)\}_{l=1}^n$
Three constants $L_f$, $L_\rho$, $L_h$ which satisfy inequalities (2.16 - 2.18) ;
**Output:** A lower bound on $J^h(x_0)$ equal to $L^h_{\mathcal{F}_n}(x_0)$ ;
**Algorithm:**
Create two $n$-dimensional vectors $V_A$ and $V_B$ ;
$V_A(i) \leftarrow 0, \forall i = \{1, \dots, n\}$ ;
$V_B(i) \leftarrow 0, \forall i = \{1, \dots, n\}$ ;
for $t = T - 1$ to $1$ do
  for $i = 1$ to $n$ (update the value of $V_A$) do
    $L_{Q_{T-t}} \leftarrow L_\rho \Bigg( \sum_{k=0}^{T-t-1} [L_f(1 + L_h)]^k \Bigg)$ ;
    $u \leftarrow h(t, y^i)$ ;
    $V_A(i) \leftarrow \max_{i'} (r^{i'} - L_{Q_{T-t}} \big( \|x^{i'} - y^i\|_\mathcal{X} + \|u^{i'} - u\|_\mathcal{U} \big) + V_B(i'))$ ;
  end for
  $V_B \leftarrow V_A$;
end for
$u_0 \leftarrow h(0, x_0)$;
$lb^* \leftarrow \max_{i'} \Bigg( r^{i'} - L_{Q_T} \big( \|x^{i'} - x_0\|_\mathcal{X} + \|u^{i'} - u_0\|_\mathcal{U} \big) + V_B(i') \Bigg)$;
**Return:** $lb^*$.
```

**Theorem 2.6.1**

*Let $x_0$ be an initial state, $h$ a policy, and $\mathcal{F}_n = \big\{(x^l, u^l, r^l, y^l)\big\}_{l=1}^n$ a set of four-tuples. We suppose that*
$\exists\ \alpha \in \mathbb{R}^+$ :

$$\sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \left\{ \min_{l\in\{1,\dots,n\}} \left\{ \|x^l - x\|_\mathcal{X} + \|u^l - u\|_\mathcal{U} \right\} \right\} \le \alpha, \qquad (2.70)$$

*and we note $\alpha^*_{\mathcal{F}_n}$ the smallest constant which satisfies (2.70).*
*Then*

$$\exists\ C \in \mathbb{R}^+ : J^h(x_0) - L^h_{\mathcal{F}_n}(x_0) \le C\alpha^*_{\mathcal{F}_n}. \qquad (2.71)$$

**Proof.** Let

$$(x_0, u_0, r_0, x_1, u_1, \ldots, x_{T-1}, u_{T-1}, r_{T-1}, x_T) \tag{2.72}$$

be the trajectory of the system starting from $x_0$ when the actions are selected $\forall t \in \{0, 1, \ldots, T-1\}$ according to the policy $h$. Let $\tau = \left[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})\right]_{t=0}^{T-1}$ be a sequence of four-tuples that satisfies

$$\begin{aligned}&\forall t \in \{0, 1, \ldots, T-1\},\\ &\left\|x^{l_t} - x_t\right\|_{\mathcal{X}} + \left\|u^{l_t} - u_t\right\|_{\mathcal{U}} = \min_{l \in \{1,\ldots,n\}} \left\|x^l - x_t\right\|_{\mathcal{X}} + \left\|u^l - u_t\right\|_{\mathcal{U}}\end{aligned} \tag{2.73}$$

We have

$$B^h(\tau, x_0) = \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \delta_t\right] \tag{2.74}$$

where

$$\forall t \in \{0, 1, \ldots, T-1\}, \delta_t = \left\|x^{l_t} - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u^{l_t} - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}}. \tag{2.75}$$

Let us focus on $\delta_t$. We have that

$$\delta_t = \left\|x^{l_t} - x_t + x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u^{l_t} - u_t + u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}}, \tag{2.76}$$

and hence

$$\delta_t \leq \left\|x^{l_t} - x_t\right\|_{\mathcal{X}} + \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u^{l_t} - u_t\right\|_{\mathcal{U}} + \left\|u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}}. \tag{2.77}$$

Using inequality (2.70), we can write

$$\left\|x^{l_t} - x_t\right\|_{\mathcal{X}} + \left\|u^{l_t} - u_t\right\|_{\mathcal{U}} \leq \alpha^*_{\mathcal{F}_n}, \tag{2.78}$$

and so we have

$$\delta_t \leq \alpha^*_{\mathcal{F}_n} + \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}}. \tag{2.79}$$

- On the one hand, we have

  $$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} = \left\|f(x_{t-1}, u_{t-1}) - f(x^{l_{t-1}}, u^{l_{t-1}})\right\|_{\mathcal{X}} \tag{2.80}$$

  and the Lipschitz continuity of $f$ implies that

  $$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} \leq L_f\big(\left\|x_{t-1} - x^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u_{t-1} - u^{l_{t-1}}\right\|_{\mathcal{U}}\big). \tag{2.81}$$

So, as

$$\left\|x_{t-1} - x^{l_{t-1}}\right\|_{\mathcal{X}} + \left\|u_{t-1} - u^{l_{t-1}}\right\|_{\mathcal{U}} \leq \alpha^*_{\mathcal{F}_n}, \tag{2.82}$$

we have

$$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} \leq L_f \alpha^*_{\mathcal{F}_n}. \tag{2.83}$$

- On the other hand, we have

$$\left\|u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} = \left\|h(t, x_t) - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} \tag{2.84}$$

and the Lipschitz continuity of $h$ implies that

$$\left\|u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} \leq L_h \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}}. \tag{2.85}$$

Since, according to Equation (2.83), we have

$$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} \leq L_f \alpha^*_{\mathcal{F}_n}, \tag{2.86}$$

we then obtain

$$\left\|u_t - h(t, y^{l_{t-1}})\right\|_{\mathcal{U}} \leq L_h L_f \alpha^*_{\mathcal{F}_n}. \tag{2.87}$$

Furthermore, (2.79), (2.83) and (2.87) imply that

$$\delta_t \leq \alpha^*_{\mathcal{F}_n} + L_f \alpha^*_{\mathcal{F}_n} + L_h L_f \alpha^*_{\mathcal{F}_n} = \alpha^*_{\mathcal{F}_n}(1 + L_f(1 + L_h)) \tag{2.88}$$

and

$$B^h(\tau, x_0) \geq \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n}(1 + L_f(1 + L_h))\right] \doteq B. \tag{2.89}$$

We also have, by definition of $L^h_{\mathcal{F}_n}(x_0)$,

$$J^h(x_0) \geq L^h_{\mathcal{F}_n}(x_0) \geq B^h(\tau, x_0) \geq B. \tag{2.90}$$

Thus,

$$\left|J^h(x_0) - L^h_{\mathcal{F}_n}(x_0)\right| \leq \left|J^h(x_0) - B\right| = J^h(x_0) - B, \tag{2.91}$$

and we have

$$
\begin{aligned}
J^h(x_0) - B &= \left| \sum_{t=0}^{T-1} \left[ \left(r_t - r^{l_t}\right) + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f(1 + L_h)) \right] \right|, && (2.92) \\
&\leq \sum_{t=0}^{T-1} \left[ \left| r_t - r^{l_t} \right| + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f(1 + L_h)) \right]. && (2.93)
\end{aligned}
$$

The Lipschitz continuity of $\rho$ allows to write

$$
\begin{aligned}
\left| r_t - r^{l_t} \right| &= \left| \rho(x_t, u_t) - \rho\left(x^{l_t}, u^{l_t}\right) \right| && (2.94) \\
&\leq L_\rho \left( \left\| x_t - x^{l_t} \right\|_{\mathcal{X}} + \left\| u_t - u^{l_t} \right\|_{\mathcal{U}} \right), && (2.95)
\end{aligned}
$$

and using inequality (2.70), we have

$$
\left| r_t - r^{l_t} \right| \leq L_\rho \alpha^*_{\mathcal{F}_n}. \qquad (2.96)
$$

Finally, we obtain

$$
\begin{aligned}
J^h(x_0) - B &\leq \sum_{t=0}^{T-1} \left[ L_\rho \alpha^*_{\mathcal{F}_n} + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f(1 + L_h)) \right] && (2.97) \\
&\leq T L_\rho \alpha^*_{\mathcal{F}_n} + \sum_{t=0}^{T-1} L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f(1 + L_h)) && (2.98) \\
&\leq \alpha^*_{\mathcal{F}_n} \Bigg( T L_\rho + \sum_{t=0}^{T-1} L_{Q_{T-t}} \big(1 + L_f(1 + L_h)\big) \Bigg). && (2.99)
\end{aligned}
$$

Thus

$$
J^h(x_0) - L^h_{\mathcal{F}_n}(x_0) \leq \alpha^*_{\mathcal{F}_n} \Bigg( T L_\rho + \sum_{t=0}^{T-1} L_{Q_{T-t}} \big(1 + L_f(1 + L_h)\big) \Bigg), \qquad (2.100)
$$

which completes the proof. ■

## 2.7 Conclusions and future research

We have introduced in this chapter an approach for deriving from a sample of trajectories a *lower* bound on the finite-horizon return of any policy from any given initial state.

We also have proposed a dynamic programming (Viterbi-like) algorithm for computing this lower bound whose complexity is linear in the optimization horizon and quadratic in the total number of state transitions of the sample of trajectories. This approach and algorithm may directly be transposed in order to compute an *upper* bound, so as to bracket the performance of the given policy, when applied to a given initial state. We also have derived a characterization of these bounds, in terms of the density of the coverage of the state-action space by the sample of trajectories used to compute them. This analysis shows that the lower (and upper) bound converges at least linearly towards the true value of the return with the density of the sample (measured by the maximal distance of any state-action pair to this sample).

The Lipschitz continuity assumptions upon which the results have been built may seem restrictive, and they indeed are. Indeed, when facing a real-life problem, it may be difficult to establish whether its systems dynamics and reward function are indeed Lipschitz continuous. Secondly, even if one can guarantee that the Lipschitz assumptions are satisfied, it is still important to be able to establish some not too-conservative approximations of the Lipschitz constants. Indeed, the larger they are, the looser the bounds will be. In the same order of ideas, the choice of the norms on the state space and the action space might influence the value of the bounds and should thus also be chosen carefully.

While the approach has been designed for computing some lower bounds on the cumulated reward obtained by a given policy, it could also serve as the base for designing new reinforcement learning algorithms which would output policies that lead to the maximization of these lower bounds.

The proposed approach could also be used in combination with batch mode reinforcement learning algorithms for identifying the pieces of trajectories that influence the most the lower bounds of the RL policy and, from there, for selecting a concise set of four-tuples from which it is possible to extract a good policy. This problem is particularly important when batch mode RL algorithms are used to design autonomous intelligent agents. Indeed, after a certain time of interaction with their environment, the sample of information these agents collect may become so numerous that batch mode RL techniques may become computationally impractical [4].

Since there exist in this context many non-deterministic problems for which it would be interesting to be able to have a lower bound on the performances of a policy (e.g., those related to the inference from clinical data of decision rules for treating chronic-like diseases [11]), extending our approach to stochastic systems would certainly be relevant. Future research on this topic could follow several paths: the study of lower bounds on the expected cumulated rewards, the design of worst-case lower bounds, a study of the case where the disturbances are part of the trajectories, etc.

# Bibliography

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

[2] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] E.F. Camacho and C. Bordons. *Model Predictive Control.* Springer, 2004.

[4] D. Ernst. Selecting concise sets of samples for a reinforcement learning agent. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*, Singapore, 2005.

[5] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[6] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.

[7] J.E. Ingersoll. *Theory of Financial Decision Making*. Rowman and Littlefield Publishers, Inc., 1987.

[8] M. Kearns and S. Singh. Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems*, pages 996–1002. MIT Press, 1999.

[9] M.G. Lagoudakis and R. Parr. Least-squares policy iteration. *Jounal of Machine Learning Research*, 4:1107–1149, 2003.

[10] S.A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, 65(2):331–366, 2003.

[11] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[12] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[13] R.E. Schapire. On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning*, 22(1/2/3), 1996.

[14] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[15] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260– 269, 1967.

# Chapter 3

# Towards $\min\max$ generalization in reinforcement learning

*In this chapter, we introduce a* $\min\max$ *approach for addressing the generalization problem in Reinforcement Learning. The* $\min\max$ *approach works by determining a sequence of actions that maximizes the worst return that could possibly be obtained considering any dynamics and reward function compatible with the sample of trajectories and some prior knowledge on the environment. We consider the particular case of deterministic Lipschitz continuous environments over continuous state spaces, finite action spaces, and a finite optimization horizon. We discuss the non-triviality of computing an exact solution of the* $\min\max$ *problem even after reformulating it so as to avoid search in function spaces. For addressing this problem, we propose to replace, inside this* $\min\max$ *problem, the search for the worst environment given a sequence of actions by an expression that lower bounds the worst return that can be obtained for a given sequence of actions. This lower bound has a tightness that depends on the sample sparsity. From there, we propose an algorithm of polynomial complexity that returns a sequence of actions leading to the maximization of this lower bound. We give a condition on the sample sparsity ensuring that, for a given initial state, the proposed algorithm produces an optimal sequence of actions in open-loop. Our experiments show that this algorithm can lead to more cautious policies than algorithms combining dynamic programming with function approximators.*

Parts of this work have been published in the *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART 2010) [13], where it received a

“Best Sudent Paper Award”. An extended version will be published as a book chapter by Springer [12]. This chapter is an extended version of [12].

In this chapter, we consider:

- a deterministic setting,
- a continuous state space and a finite action space.

## 3.1 Introduction

Since the late sixties, the field of Reinforcement Learning (RL) [28] has studied the problem of inferring from the sole knowledge of observed system trajectories, near-optimal solutions to optimal control problems. The original motivation was to design computational agents able to learn by themselves how to interact in a rational way with their environment. The techniques developed in this field have appealed researchers trying to solve sequential decision making problems in many fields such as Finance [16], Medicine [20, 21] or Engineering [24].

RL algorithms are challenged when dealing with large or continuous state spaces. Indeed, in such cases they have to generalize the information contained in a generally sparse sample of trajectories. The dominating approach for generalizing this information is to combine RL algorithms with function approximators [2, 17, 9]. Usually, these approximators generalize the information contained in the sample to areas poorly covered by the sample by implicitly assuming that the properties of the system in those areas are similar to the properties of the system in the nearby areas well covered by the sample. This in turn often leads to low performance guarantees on the inferred policy when large state space areas are poorly covered by the sample. This can be explained by the fact that when computing the performance guarantees of these policies, one needs to take into account that they may actually drive the system into the poorly visited areas to which the generalization strategy associates a favorable environment behavior, while the environment may actually be particularly adversarial in those areas. This is corroborated by theoretical results which show that the performance guarantees of the policies inferred by these algorithms degrade with the sample sparsity where, loosely speaking, the sparsity can be seen as the radius of the largest non-visited state space area. [1]

As in our previous work [13] from which this chapter is an extended version, we assume a deterministic Lipschitz continuous environment over continuous state spaces, finite action spaces, and a finite time-horizon. In this context, we introduce a $\min\max$ approach to address the generalization problem. The $\min\max$ approach works by determining a sequence of actions that maximizes the worst return that could possibly be obtained considering any dynamics and reward functions compatible with the sample of trajectories, and a weak prior knowledge given in the form of upper bounds on the Lipschitz constants of the environment. However, we show that finding an exact solu-

[1] Usually, these theoretical results do not give lower bounds per se but a distance between the actual return of the inferred policy and the optimal return. However, by adapting in a straightforward way the proofs behind these results, it is often possible to get a bound on the distance between the estimate of the return of the inferred policy computed by the RL algorithm and its actual return and, from there, a lower bound on the return of the inferred policy.

tion of the $\min\max$ problem is far from trivial, even after reformulating the problem so as to avoid the search in the space of all compatible functions. To circumvent these difficulties, we propose to replace, inside this $\min\max$ problem, the search for the worst environment given a sequence of actions by an expression that lower bounds the worst return that can be obtained for a given sequence of actions. This lower bound is derived from [11] (also reported in Chapter 2) and has a tightness that depends on the sample sparsity. From there, we propose a Viterbi–like algorithm [29] for computing an open-loop sequence of actions to be used from a given initial state to maximize that lower bound. This algorithm is of polynomial computational complexity in the size of the dataset and the optimization horizon. It is named CGRL for Cautious Generalization (oriented) Reinforcement Learning since it essentially shows a cautious behavior in the sense that it computes decisions that avoid driving the system into areas of the state space that are not well enough covered by the available dataset, according to the prior information about the dynamics and reward function. Besides, the CGRL algorithm does not rely on function approximators and it computes, as a byproduct, a lower bound on the return of its open-loop sequence of decisions. We also provide a condition on the sample sparsity ensuring that, for a given initial state, the CGRL algorithm produces an optimal sequence of actions in open-loop, and we suggest directions for leveraging our approach to a larger class of problems in RL.

The rest of this chapter is organized as follows. Section 3.2 briefly discusses related work. In Section 3.3, we formalize the $\min\max$ approach to generalization, and we discuss its non trivial nature in Section 3.4. In Section 3.5, we exploit the results of [11] (also reported in Chapter 2) for lower bounding the worst return that can be obtained for a given sequence of actions. Section 3.6 proposes a polynomial algorithm for inferring a sequence of actions maximizing this lower bound and states a condition on the sample sparsity for its optimality. Section 3.7 illustrates the features of the proposed algorithm and Section 3.8 discusses its interest, while Section 3.9 concludes.

## 3.2 Related work

The $\min\max$ approach to generalization followed by the CGRL algorithm results in the output of policies that are likely to drive the agent only towards areas well enough covered by the sample. Heuristic strategies have already been proposed in the RL literature to infer policies that exhibit such a conservative behavior. As a way of example, some of these strategies associate high negative rewards to trajectories falling outside of the well covered areas. Other works in RL have already developed $\min\max$ strategies when the environment behavior is partially unknown [18, 4, 25]. However, these strategies usually consider problems with finite state spaces where the uncertainities come from the lack of knowledge of the transition probabilities [7, 5]. In model predictive control (MPC) where the environment is supposed to be fully known [10], $\min\max$ approaches have been used to determine the optimal sequence of actions with respect to the “worst case” disturbance sequence occuring [1]. The CGRL algorithm relies on a methodology for computing a lower bound on the worst possible return (considering any compatible environment) in a deterministic setting with a mostly unknown actual environment. In this, it is related to works in the field of RL which try to get from a sample of trajectories lower bounds on the returns of inferred policies [19, 23].

## 3.3 Problem Statement

We consider a discrete-time system whose dynamics over $T$ stages is described by a time-invariant equation

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \ldots, T-1, \tag{3.1}$$

where for all $t$, the state $x_t$ is an element of the compact state space $\mathcal{X} \subset \mathbb{R}^{d_\mathcal{X}}$ where $\mathbb{R}^{d_\mathcal{X}}$ denotes the $d_\mathcal{X}-$dimensional Euclidean space and $u_t$ is an element of the finite (discrete) action space $\mathcal{U}$. $T \in \mathbb{N}_0$ is referred to as the optimization horizon. An instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{3.2}$$

is associated with the action $u_t$ taken while being in state $x_t$. For every initial state $x_0 \in \mathcal{X}$ and for every sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, the cumulated reward over $T$ stages (also named $T-$stage return) is defined as

**Definition 3.3.1 ($T-$stage return of the sequence $(u_0, \ldots, u_{T-1})$)**
$\forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$J^{u_0,\ldots,u_{T-1}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t)\ , \tag{3.3}$$

*where*

$$x_{t+1} = f(x_t, u_t)\ , \forall t \in \{0, \ldots, T-1\}. \tag{3.4}$$

We assume that the system dynamics $f$ and the reward function $\rho$ are Lipschitz continuous:

**Assumption 3.3.2 (Lipschitz continuity of $f$ and $\rho$)**
*There exist finite constants $L_f, L_\rho \in \mathbb{R}$ such that:*
$\forall (x, x') \in \mathcal{X}^2, \forall u \in \mathcal{U}$,

$$\begin{aligned} \|f(x,u) - f(x',u)\|_{\mathcal{X}} &\leq L_f \|x - x'\|_{\mathcal{X}} \ , \qquad (3.5) \\ |\rho(x,u) - \rho(x',u)| &\leq L_\rho \|x - x'\|_{\mathcal{X}} \ , \qquad (3.6) \end{aligned}$$

*where $\|.\|_{\mathcal{X}}$ denotes the Euclidean norm over the space $\mathcal{X}$.*

We further suppose that:

**Assumption 3.3.3**

1. *The system dynamics $f$ and the reward function $\rho$ are unknown,*

2. *A set of one-step transitions*

$$\mathcal{F}_n = \left\{ (x^l, u^l, r^l, y^l) \right\}_{l=1}^{n} \qquad (3.7)$$

*is known where each one-step transition is such that*

$$\begin{cases} y^l = f(x^l, u^l), \\ r^l = \rho(x^l, u^l). \end{cases} \qquad (3.8)$$

3. *Each action $a \in \mathcal{U}$ appears at least once in $\mathcal{F}_n$:*

$$\forall a \in \mathcal{U}, \ \exists (x, u, r, y) \in \mathcal{F}_n \ : u = a \qquad (3.9)$$

4. *Two constants $L_f$ and $L_\rho$ satisfying the above-written inequalities are known. These constants do not necessarily have to be the smallest ones satisfying these inequalities (i.e., the Lipschitz constants).*

We define the set of functions $\mathcal{L}^f_{\mathcal{F}_n}$ (resp. $\mathcal{L}^\rho_{\mathcal{F}_n}$) from $\mathcal{X} \times \mathcal{U}$ into $\mathcal{X}$ (resp. into $\mathbb{R}$) as follows :

**Definition 3.3.4 (Compatible environments)**

$$\mathcal{L}^f_{\mathcal{F}_n} = \left\{ f' : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X} \middle| \left\{ \begin{array}{l} \forall x, x' \in \mathcal{X}, \forall u \in \mathcal{U} \ , \\ \|f'(x,u) - f'(x',u)\|_{\mathcal{X}} \leq L_f \|x - x'\|_{\mathcal{X}} \ , \\ \forall l \in \{1, \ldots, n\}, f'(x^l, u^l) = f(x^l, u^l) = y^l \end{array} \right. \right\} , \qquad (3.10)$$

$$\mathcal{L}^{\rho}_{\mathcal{F}_n} = \left\{ \rho' : \mathcal{X} \times \mathcal{U} \to \mathbb{R} \middle| \left\{ \begin{array}{l} \forall x, x' \in \mathcal{X}, \forall u \in \mathcal{U} \ , \\ |\rho'(x,u) - \rho'(x',u)| \leq L_{\rho} \|x - x'\|_{\mathcal{X}} \ , \\ \forall l \in \{1, \ldots, n\}, \rho'(x^l, u^l) = \rho(x^l, u^l) = r^l \end{array} \right. \right\} . \tag{3.11}$$

*In the following, we call a "compatible environment" any pair*

$$(f', \rho') \in \mathcal{L}^{f}_{\mathcal{F}_n} \times \mathcal{L}^{\rho}_{\mathcal{F}_n} \ . \tag{3.12}$$

Given a compatible environment $(f', \rho')$, a sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ and an initial state $x_0 \in \mathcal{X}$, we introduce the $(f', \rho')-$return over $T$ stages when starting from $x_0 \in \mathcal{X}$:

**Definition 3.3.5 (**$(f', \rho')-$**return over** $T$ **stages)**
$\forall (f', \rho') \in \mathcal{L}^{f}_{\mathcal{F}_n} \times \mathcal{L}^{\rho}_{\mathcal{F}_n}, \forall (u_0, \ldots, u_{T-1}), \forall x_0 \in \mathcal{X}$,

$$J^{u_0, \ldots, u_{T-1}}_{(f', \rho')}(x_0) = \sum_{t=0}^{T-1} \rho'(x'_t, u_t) \ , \tag{3.13}$$

*where*

$$x'_{t+1} = f'(x'_t, u_t), \ \forall t \in \{0, \ldots, T-1\} \ , \tag{3.14}$$

*and* $x'_0 = x_0$.

We introduce $I^{u_0, \ldots, u_{T-1}}_{\mathcal{F}_n}(x_0)$ such that

$$I^{u_0, \ldots, u_{T-1}}_{\mathcal{F}_n}(x_0) = \min_{(f', \rho') \in \mathcal{L}^{f}_{\mathcal{F}_n} \times \mathcal{L}^{\rho}_{\mathcal{F}_n}} \left\{ J^{u_0, \ldots, u_{T-1}}_{(f', \rho')}(x_0) \right\} \ . \tag{3.15}$$

The existence of $I^{u_0, \ldots, u_{T-1}}_{\mathcal{F}_n}(x_0)$ is ensured by the following arguments:

1. The space $\mathcal{X}$ is compact,

2. The set $\mathcal{L}^{f}_{\mathcal{F}_n} \times \mathcal{L}^{\rho}_{\mathcal{F}_n}$ is closed and bounded considering the $\|.\|_{\infty}$ norm

$$\|(f', \rho')\|_{\infty} = \sup_{(x,u) \in \mathcal{X} \times \mathcal{U}} \|(f'(x,u), \rho'(x,u))\|_{\mathbb{R}^{d_{\mathcal{X}}+1}} \tag{3.16}$$

   where $\|.\|_{\mathbb{R}^{d_{\mathcal{X}}+1}}$ is the Euclidean norm over $\mathbb{R}^{d_{\mathcal{X}}+1}$

3. One can show that the mapping

$$\mathcal{M}_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}} : \mathcal{L}_{\mathcal{F}_n}^{f} \times \mathcal{L}_{\mathcal{F}_n}^{\rho} \to \mathbb{R} \tag{3.17}$$

such that

$$\mathcal{M}_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}}(f',\rho') = J_{(f',\rho')}^{u_0,\ldots,u_{T-1}}(x_0) \tag{3.18}$$

is a continuous mapping. Furthermore, this also proves that $\forall (u_0,\ldots,u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$\begin{aligned} \exists (f_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}}, \rho_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}}) \in \mathcal{L}_{\mathcal{F}_n}^{f} \times \mathcal{L}_{\mathcal{F}_n}^{\rho} : \\ J_{(f_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}}, \rho_{\mathcal{F}_n,x_0}^{u_0,\ldots,u_{T-1}})}^{u_0,\ldots,u_{T-1}}(x_0) = I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0). \end{aligned} \tag{3.19}$$

Our goal is to compute, given an initial state $x_0 \in \mathcal{X}$, an open-loop sequence of actions $(\dot{u}_0(x_0),\ldots,\dot{u}_{T-1}(x_0)) \in \mathcal{U}^T$ that gives the highest return in the least favorable compatible environment. This problem can be formalized as the $\min\max$ problem:

$$(\dot{u}_0(x_0),\ldots,\dot{u}_{T-1}(x_0)) \in \underset{(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T}{\arg\max} \left\{ I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \right\}. \tag{3.20}$$

## 3.4 Reformulation of the $\min\max$ problem

Since $\mathcal{U}$ is finite, one could solve the $\min\max$ problem by computing for each sequence of actions $(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T$ the value of $I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$. As the latter computation is posed as an infinite-dimensional minimization problem over the function space $\mathcal{L}_{\mathcal{F}_n}^{f} \times \mathcal{L}_{\mathcal{F}_n}^{\rho}$, we first show that it can be reformulated as a finite-dimensional problem over $\mathcal{X}^{T-1} \times \mathbb{R}^T$. This is based on the observation that $I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$ is actually equal to the lowest sum of rewards that could be collected along a trajectory compatible with an environment from $\mathcal{L}_{\mathcal{F}_n}^{f} \times \mathcal{L}_{\mathcal{F}_n}^{\rho}$, and is precisely stated by the following Theorem.

**Theorem 3.4.1 (Equivalence)**
*Let $(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T$ and $x_0 \in \mathcal{X}$. Let $K_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$ be the solution of the following optimization problem:*

$$K_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) = \min_{\substack{\hat{r}_0 \;\ldots\; \hat{r}_{T-1} \in \mathbb{R} \\ \hat{x}_0 \;\ldots\; \hat{x}_{T-1} \in \mathcal{X}}} \left\{ \sum_{t=0}^{T-1} \hat{r}_t \right\}, \tag{3.21}$$

*where the variables $\hat{x}_0, \ldots, \hat{x}_{T-1}$ and $\hat{r}_0, \ldots, \hat{r}_{T-1}$ satisfy the constraints*

$$\left.\begin{array}{l} \left|\hat{r}_t - r^{l_t}\right| \leq L_\rho \left\|\hat{x}_t - x^{l_t}\right\|_{\mathcal{X}} \ , \\ \left\|\hat{x}_{t+1} - y^{l_t}\right\|_{\mathcal{X}} \leq L_f \left\|\hat{x}_t - x^{l_t}\right\|_{\mathcal{X}} \end{array}\right\} \forall l_t \in \{1, \ldots, n | u^{l_t} = u_t\} \ , \tag{3.22}$$

$$\left.\begin{array}{l} |\hat{r}_t - \hat{r}_{t'}| \leq L_\rho \left\|\hat{x}_t - \hat{x}_{t'}\right\|_{\mathcal{X}} \ , \\ \left\|\hat{x}_{t+1} - \hat{x}_{t'+1}\right\|_{\mathcal{X}} \leq L_f \left\|\hat{x}_t - \hat{x}_{t'}\right\|_{\mathcal{X}} \end{array}\right\} \forall t, t' \in \{0, \ldots, T-1 | u_t = u_{t'}\} \ , \tag{3.23}$$

$$\hat{x}_0 = x_0 \ . \tag{3.24}$$

*Then,*

$$K^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) = I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \ . \tag{3.25}$$

**Proof.**

- Let us first prove that

  $$I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq K^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \ . \tag{3.26}$$

  Let us assume that we know a set of variables $\hat{x}_0, \ldots, \hat{x}_{T-1}$ and $\hat{r}_0, \ldots, \hat{r}_{T-1}$ that are solution of the optimization problem. To each action $u \in \mathcal{U}$, we associate the sets

  $$A_u = \left\{x^l \in \{x^1, \ldots, x^n\} | u^l = u\right\} \tag{3.27}$$

  and

  $$B_u = \{\hat{x}_t \in \{\hat{x}_0, \ldots, \hat{x}_{T-1}\} | u_t = u\} \ . \tag{3.28}$$

  Let $S_u = A_u \cup B_u$. For simplicity in the proof, we assume that the points of $S_u$ are in *general position*, i.e., no $(d_{\mathcal{X}} + 1)$ points from $S_u$ lie in a $(d_{\mathcal{X}} - 1)-$dimensional plane (the points are affinely independent). This allows to compute a $d_{\mathcal{X}}-$dimensional triangulation $\{\Delta^1, \ldots, \Delta^p\}$ of the convex hull $H(S_u)$ defined by the set of points $S_u$ [6]. We introduce for every value of $u \in \mathcal{U}$ two Lipschitz continuous functions $\tilde{f}_u : \mathcal{X} \to \mathcal{X}$ and $\tilde{\rho}_u : \mathcal{X} \to \mathbb{R}$ defined as follows:

  - *Inside the convex hull* $H(S_u)$
    Let $g^f_u : S_u \to \mathcal{X}$ and $g^\rho_u : S_u \to \mathbb{R}$ be such that:

    $$\forall x^l \in A_u \ , \left\{\begin{array}{l} g^f_u(x^l) = f(x^l, u) \\ g^\rho_u(x^l) = \rho(x^l, u) \end{array}\right. \quad \text{and } \forall \hat{x}_t \in B_u \backslash A_u \ , \left\{\begin{array}{l} g^f_u(\hat{x}_t) = \hat{x}_{t+1} \\ g^\rho_u(\hat{x}_t) = \hat{r}_t \end{array}\right. \ . \tag{3.29}$$

Then, we define the functions $\tilde{f}_u$ and $\tilde{\rho}_u$ inside $H(S_u)$ as follows:
$\forall k \in \{1, \ldots, p\}, \forall x' \in \Delta^k$ ,

$$\tilde{f}_u(x') = \sum_{i=1}^{d_{\mathcal{X}}+1} \lambda_i^k(x') g_u^f(s_i^k) , \tag{3.30}$$

$$\tilde{\rho}_u(x') = \sum_{i=1}^{d_{\mathcal{X}}+1} \lambda_i^k(x') g_u^\rho(s_i^k) , \tag{3.31}$$

where $s_i^k \quad i = 1 \ldots (d_{\mathcal{X}} + 1)$ are the vertices of $\Delta^k$ and $\lambda_i^k(x)$ are such that

$$x' = \sum_{i=1}^{d_{\mathcal{X}}+1} \lambda_i^k(x') s_i^k \tag{3.32}$$

with

$$\sum_{i=1}^{d_{\mathcal{X}}+1} \lambda_i^k(x') = 1 \tag{3.33}$$

and

$$\lambda_i^k(x') \geq 0, \ \forall i . \tag{3.34}$$

– *Outside the convex hull* $H(S_u)$
According the Hilbert Projection Theorem [26], for every point $x'' \in \mathcal{X}$, there exists a unique point $y'' \in H(S_u)$ such that $\|x'' - y''\|_{\mathcal{X}}$ is minimized over $H(S_u)$. This defines a mapping

$$t_u : \mathcal{X} \to H(S_u) \tag{3.35}$$

which is $1-$Lipschitzian. Using the mapping $t_u$, we define the functions $\tilde{f}_u$ and $\tilde{\rho}_u$ outside $H(S_u)$ as follows:
$\forall x'' \in \mathcal{X} \backslash H(S_u)$,

$$\tilde{f}_u(x'') = \tilde{f}_u(t_u(x'')) \tag{3.36}$$

$$\tilde{\rho}_u(x'') = \tilde{\rho}_u(t_u(x'')) . \tag{3.37}$$

We finally introduce the functions $\tilde{f}$ and $\tilde{\rho}$ over the space $\mathcal{X} \times \mathcal{U}$ as follows:
$\forall (x, u) \in \mathcal{X} \times \mathcal{U}$,

$$\tilde{f}(x, u) = \tilde{f}_u(x) \tag{3.38}$$

$$\tilde{\rho}(x, u) = \tilde{\rho}_u(x) . \tag{3.39}$$

One can easily show that the pair $(\tilde{f}, \tilde{\rho})$ belongs to $\mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$ and satisfies

$$\begin{aligned} J^{u_0,\ldots,u_{T-1}}_{(\tilde{f},\tilde{\rho})}(x_0) &= \sum_{t=0}^{T-1} \tilde{\rho}(\hat{x}_t, u_t) && (3.40)\\ &= \sum_{t=0}^{T-1} \hat{r}_t && (3.41)\end{aligned}$$

with

$$\hat{x}_{t+1} = \tilde{f}(\hat{x}_t, u_t) \qquad (3.42)$$

and $\hat{x}_0 = x_0$. This proves that

$$I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq K^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)\ . \qquad (3.43)$$

(Note that one could still build two functions $(\tilde{f}, \tilde{\rho}) \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$ even if the sets of points $(S_u)_{u \in \mathcal{U}}$ are not in general position)

- Then, let us prove that

$$K^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)\ . \qquad (3.44)$$

We consider the environment $(f^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}, \rho^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0})$ introduced in Equation (3.19) at the end of Section 3.3. One has

$$\begin{aligned} I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) &= J^{u_0,\ldots,u_{T-1}}_{(f^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}, \rho^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0})}(x_0) && (3.45)\\ &= \sum_{t=0}^{T-1} \tilde{r}_t\ , && (3.46)\end{aligned}$$

with, $\forall t \in \{0, \ldots, T-1\}$ ,

$$\begin{aligned} \tilde{r}_t &= \rho^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}(\tilde{x}_t, u_t)\ , && (3.47)\\ \tilde{x}_{t+1} &= f^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}(\tilde{x}_t, u_t)\ , && (3.48)\\ \tilde{x}_0 &= x_0\ . && (3.49)\end{aligned}$$

The variables $\tilde{x}_0, \ldots, \tilde{x}_{T-1}$ and $\tilde{r}_0, \ldots, \tilde{r}_{T-1}$ satisfy the constraints introduced in Theorem (3.4.1). This proves that

$$K^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \qquad (3.50)$$

and completes the proof.

■

Unfortunately, this latter minimization problem turns out to be non-convex in its generic form and, hence "off the shelf" algorithms will only be able to provide upper bounds on its value. Furthermore, the overall complexity of an algorithm that would be based on the enumeration of $\mathcal{U}^T$, combined with a local optimizer for the inner loop, may be intractable as soon as the cardinality of the action space $\mathcal{U}$ and/or the optimization horizon $T$ become large.

We leave the exploration of the above formulation for future research. Instead, in the following subsections, we use the results from [11] (reported in Chapter 2) to define a maximal lower bound

$$L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \leq I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \tag{3.51}$$

for a given initial state $x_0 \in \mathcal{X}$ and a sequence $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$. Furthermore, we show that the maximization of this lower bound $L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$ with respect to the choice of a sequence of actions lends itself to a dynamic programming type of decomposition. In the end, this yields a polynomial algorithm for the computation of a sequence of actions $(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ maximizing a lower bound of the original $\min\max$ problem, i.e.

$$(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \underset{(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T}{\arg\max} \left\{ L_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0) \right\} . \tag{3.52}$$

## 3.5 Lower bound on the return of a given sequence of actions

In this section, we present a method for computing, from a given initial state $x_0 \in \mathcal{X}$, a sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, a dataset of transitions, and weak prior knowledge about the environment, a lower bound on $I_{\mathcal{F}_n}^{u_0,\ldots,u_{T-1}}(x_0)$. The method is adapted from [11] (reported in Chapter 2). In the following, we denote by $\mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})}$ the set of all sequences of $T$ one-step system transitions that may be built from elements of $\mathcal{F}_n$ and that are compatible with $(u_0, \ldots, u_{T-1})$:

**Definition 3.5.1 (Compatible sequences of transitions)**
$\forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$,

$$\begin{aligned}\mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})} = \Big\{ & \left[\left(x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0}\right), \ldots, \left(x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}}\right)\right] \\ \Big| \ & u^{l_t} = u_t, \ \forall t \in \{0, \ldots, T-1\} \Big\}\end{aligned} \tag{3.53}$$

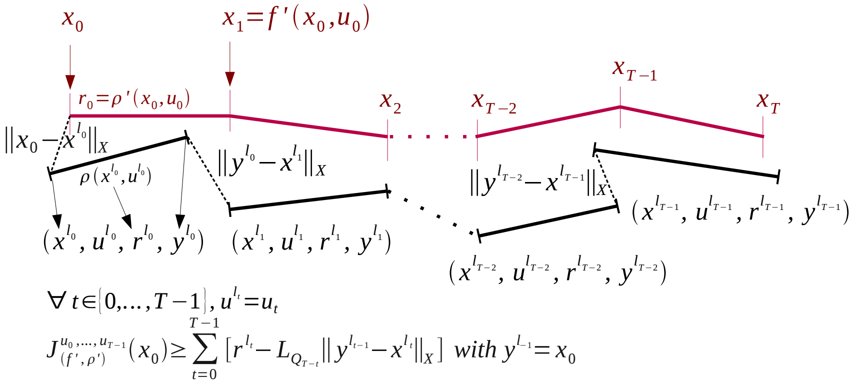
$x_0$
$x_1 = f'(x_0, u_0)$
$r_0 = \rho'(x_0, u_0)$
$x_2$
$x_{T-2}$
$x_{T-1}$
$x_T$
$\|x_0 - x^{l_0}\|_X$
$\rho(x^{l_0}, u^{l_0})$
$\|y^{l_0} - x^{l_1}\|_X$
$\|y^{l_{T-2}} - x^{l_{T-1}}\|_X$
$(x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0})$
$(x^{l_1}, u^{l_1}, r^{l_1}, y^{l_1})$
$(x^{l_{T-2}}, u^{l_{T-2}}, r^{l_{T-2}}, y^{l_{T-2}})$
$(x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}})$
$\forall\, t \in [0, \ldots, T-1], u^{l_t} = u_t$
$J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0) \geq \sum_{t=0}^{T-1} [r^{l_t} - L_{Q_{T-t}} \|y^{l_{t-1}} - x^{l_t}\|_X]$ with $y^{l_{-1}} = x_0$

Figure 3.1: A graphical interpretation of the different terms composing the bound on $J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0)$ computed from a sequence of one-step transitions.

First, we compute a lower bound on $I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ from any given element $\tau$ from $\mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})}$. This lower bound $B(\tau,x_0)$ is made of the sum of the $T$ rewards corresponding to $\tau$ ($\sum_{t=0}^{T-1} r^{l_t}$) and $T$ negative terms. Every negative term is associated with a one-step transition. More specifically, the negative term corresponding to the transition $(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})$ of $\tau$ represents an upper bound on the variation of the cumulated rewards over the remaining time steps that can occur by simulating the system from a state $x^{l_t}$ rather than $y^{l_{t-1}}$ (with $y^{l_{-1}} = x_0$) and considering any compatible environment $(f', \rho')$ from $\mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$. By maximizing $B(\tau,x_0)$ over $\mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})}$, we obtain a maximal lower bound on $I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$. Furthermore, we prove that the distance from the maximal lower bound to the actual return $J^{u_0,\ldots,u_{T-1}}(x_0)$ can be characterized in terms of the sample sparsity.

### 3.5.1 Computing a bound from a given sequence of one-step transitions

We have the following lemma.

**Lemma 3.5.2**
*Let $(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T$ be a sequence of actions and $x_0 \in \mathcal{X}$ an initial state. Let $\tau$ be a sequence of one-step transitions:*

$$\tau = \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} \in \mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})} \ . \tag{3.54}$$

*Then,*

$$B(\tau,x_0) \le I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \le J^{u_0,\ldots,u_{T-1}}(x_0) \ , \tag{3.55}$$

*with*

$$B(\tau,x) \doteq \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}}\right] \ , \tag{3.56}$$

$$y^{l_{-1}} = x_0 \ , \tag{3.57}$$

$$L_{Q_{T-t}} = L_\rho \sum_{i=0}^{T-t-1} (L_f)^i \ . \tag{3.58}$$

Before proving Lemma 3.5.2, we prove a preliminary result related to the Lipschitz continuity of state-action value functions.

For any compatible environment $(f', \rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$, and for $N = 1, \ldots, T$, let us define the family of $(f', \rho')-$state-action value functions

$$Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')} : \mathcal{X} \times \mathcal{U} \to \mathbb{R} \tag{3.59}$$

as follows:

**Definition 3.5.3 ($(f', \rho')-$state-action value functions)**
$\forall (f', \rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}, \forall N \in \{1, \ldots, T\}, \forall (x, u) \in \mathcal{X} \times \mathcal{U}$,

$$Q^{u_0,\ldots,u_{T-1}}_{N,(f'\rho')}(x, u) = \rho'(x, u) + \sum_{t=T-N+1}^{T-1} \rho'(x'_t, u_t), \tag{3.60}$$

*where*

$$x'_{t+1} = f'(x'_t, u_t), \ \forall t \in \{T - N + 1, \ldots, T - 1\}, \tag{3.61}$$

*and*

$$x'_{T-N+1} = f'(x, u). \tag{3.62}$$

$Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x, u)$ gives the sum of rewards from instant $t = T - N$ to instant $T - 1$ given the compatible environment $(f', \rho')$ when

- The system is in state $x \in \mathcal{X}$ at instant $T - N$,
- The action chosen at instant $T - N$ is $u$,
- The actions chosen at instants $t > T - N$ are $u_t$.

We have the following trivial propositions:

**Proposition 3.5.4**
$\forall (f', \rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}, \forall x_0 \in \mathcal{X}$,

$$J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0) = Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}(x_0, u_0). \tag{3.63}$$

**Proposition 3.5.5**
$\forall (f', \rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}, \forall (x, u) \in \mathcal{X} \times \mathcal{U}, \forall N \in \{1, \ldots, T - 1\}$

$$Q^{u_0,\ldots,u_{T-1}}_{N+1,(f',\rho')}(x, u) = \rho'(x, u) + Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(f'(x, u), u_{T-N}) . \tag{3.64}$$

**Lemma 3.5.6 (Lipschitz continuity of $Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}$)**
$\forall (f',\rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}, \forall N \in \{1,\ldots,T\}$, $\forall (x,x') \in \mathcal{X}^2, \forall u \in \mathcal{U}$ ,

$$\left| Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x,u) - Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x',u) \right| \leq L_{Q_N} \left\| x - x' \right\|_{\mathcal{X}} \ , \tag{3.65}$$

*with*

$$L_{Q_N} = L_\rho \sum_{i=0}^{N-1} (L_f)^i \ . \tag{3.66}$$

**Proof.** Let $\forall (f',\rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$ be a compatible environment. We consider the statement $\mathcal{H}(N)$: $\forall (x,x') \in \mathcal{X}^2, \forall u \in \mathcal{U}$,

$$\left| Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x,u) - Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x',u) \right| \leq L_{Q_N} \|x - x'\|_{\mathcal{X}}. \tag{3.67}$$

We prove by induction that $\mathcal{H}(N)$ is true $\forall N \in \{1,\ldots,T\}$. For the sake of clarity, we use the notation

$$\left| Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x,u) - Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(x',u) \right| = \Delta_N \ . \tag{3.68}$$

- *Basis ($N = 1$) :* We have

  $$\Delta_1 = \left| \rho'(x,u) - \rho'(x',u) \right| , \tag{3.69}$$

  and since $\rho' \in \mathcal{L}^\rho_{\mathcal{F}_n}$, we can write

  $$\Delta_1 \leq L_\rho \|x - x'\|_{\mathcal{X}} \ . \tag{3.70}$$

  This proves $\mathcal{H}(1)$.

- *Induction step:* We suppose that $\mathcal{H}(N)$ is true, $1 \leq N \leq T-1$. Using Proposition (3.5.5), we can write

  $$\begin{aligned} \Delta_{N+1} &= \left| Q^{u_0,\ldots,u_{T-1}}_{N+1,(f',\rho')}(x,u) - Q^{u_0,\ldots,u_{T-1}}_{N+1,(f',\rho')}(x',u) \right| & (3.71) \\ &= \big| \rho'(x,u) - \rho'(x',u) + Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(f'(x,u),u_{T-N}) \\ &- Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(f'(x',u),u_{T-N}) \big| & (3.72) \end{aligned}$$

  and, from there,

  $$\begin{aligned} \Delta_{N+1} \leq\ & \left| \rho'(x,u) - \rho'(x',u) \right| \\ +\ & \left| Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(f'(x,u),u_{T-N}) - Q^{u_0,\ldots,u_{T-1}}_{N,(f',\rho')}(f'(x',u),u_{T-N}) \right| \ . \end{aligned} \tag{3.73}$$

$\mathcal{H}(N)$ and the Lipschitz continuity of $\rho'$ give

$$\Delta_{N+1} \leq L_\rho \|x - x'\|_\mathcal{X} + L_{Q_N} \|f'(x,u) - f'(x',u)\|_\mathcal{X}. \tag{3.74}$$

Since $f' \in \mathcal{L}^f_{\mathcal{F}_n}$, the Lipschitz continuity of $f'$ gives

$$\Delta_{N+1} \leq L_\rho \|x - x'\|_\mathcal{X} + L_{Q_N} L_f \|x - x'\|_\mathcal{X} \ , \tag{3.75}$$

and then

$$\Delta_{N+1} \leq L_{Q_{N+1}} \|x - x'\|_\mathcal{X} \tag{3.76}$$

since

$$L_{Q_{N+1}} = L_\rho + L_{Q_N} L_f. \tag{3.77}$$

This proves $\mathcal{H}(N+1)$ and ends the proof.

■

**Proof of Lemma 3.5.2.**

- The inequality

$$I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq J^{u_0,\ldots,u_{T-1}}(x_0) \tag{3.78}$$

is trivial since $(f, \rho)$ belongs to $\mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$.

- Let $(f', \rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$ be a compatible environment. By assumption we have $u^{l_0} = u_0$, then we use Proposition (3.5.4) and the Lipschitz continuity of $Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}$ to write

$$\left| J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0) - Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}\left(x^{l_0}, u_0\right) \right| \leq L_{Q_T} \left\| x_0 - x^{l_0} \right\|_\mathcal{X} . \tag{3.79}$$

It follows that

$$Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}\left(x^{l_0}, u_0\right) - L_{Q_T} \left\| x_0 - x^{l_0} \right\|_\mathcal{X} \leq J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0). \tag{3.80}$$

According to Proposition (3.5.5), we have

$$Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}\left(x^{l_0}, u_0\right) = \rho'\left(x^{l_0}, u_0\right) + Q^{u_0,\ldots,u_{T-1}}_{T-1,(f'\rho')}\left(f'(x^{l_0}, u_0), u_1\right) \tag{3.81}$$

and from there

$$Q^{u_0,\ldots,u_{T-1}}_{T,(f',\rho')}\left(x^{l_0},u_0\right) = r^{l_0} + Q^{h}_{T-1,(f',\rho')}\left(y^{l_0},u_1\right). \tag{3.82}$$

Thus,

$$Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(y^{l_0},u_1\right) + r^{l_0} - L_{Q_T}\left\|x_0 - x^{l_0}\right\|_{\mathcal{X}} \leq J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0). \tag{3.83}$$

The Lipschitz continuity of $Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}$ with $u_1 = u^{l_1}$ gives

$$\left|Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(y^{l_0},u_1\right) - Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(x^{l_1},u^{l_1}\right)\right| \leq L_{Q_{T-1}}\left\|y^{l_0} - x^{l_1}\right\|_{\mathcal{X}} . \tag{3.84}$$

This implies that

$$Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(x^{l_1},u_1\right) - L_{Q_{T-1}}\left\|y^{l_0} - x^{l_1}\right\|_{\mathcal{X}} \leq Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(y^{l_0},u_1\right). \tag{3.85}$$

We have therefore

$$\begin{aligned} Q^{u_0,\ldots,u_{T-1}}_{T-1,(f',\rho')}\left(x^{l_1},u_1\right) + r^{l_0} &- L_{Q_T}\left\|x_0 - x^{l_0}\right\|_{\mathcal{X}} - L_{Q_{T-1}}\left\|y^{l_0} - x^{l_1}\right\|_{\mathcal{X}} \\ &\leq J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0). \end{aligned} \tag{3.86}$$

By developing this iteration, we obtain

$$J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0) \geq \sum_{t=0}^{T-1}\left[r^{l_t} - L_{Q_{T-t}}\|y^{l_{t-1}} - x^{l_t}\|_{\mathcal{X}}\right] . \tag{3.87}$$

The right side of Equation (3.87) does not depend on the choice of $(f',\rho') \in \mathcal{L}^{f}_{\mathcal{F}_n} \times \mathcal{L}^{\rho}_{\mathcal{F}_n}$; Equation (3.87) is thus true for a compatible environment $(f',\rho')$ such that

$$(f',\rho') = (f^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}, \rho^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0}) . \tag{3.88}$$

(cf. Equation (3.19) in Section 3.3). This finally gives

$$I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \geq \sum_{t=0}^{T-1}\left[r^{l_t} - L_{Q_{T-t}}\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}}\right] \tag{3.89}$$

since

$$I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) = J^{u_0,\ldots,u_{T-1}}_{(f^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0},\rho^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n,x_0})}(x_0) . \tag{3.90}$$

■

The lower bound on $I^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ derived in this lemma can be interpreted as follows. Given any compatible environment $(f',\rho') \in \mathcal{L}^f_{\mathcal{F}_n} \times \mathcal{L}^\rho_{\mathcal{F}_n}$, the sum of the rewards of the "broken" trajectory formed by the sequence of one-step system transitions $\tau$ can never be greater than $J^{u_0,\ldots,u_{T-1}}_{(f',\rho')}(x_0)$, provided that every reward $r^{l_t}$ is penalized by a factor $L_{Q_{T-t}} \left\| y^{l_{t-1}} - x^{l_t} \right\|_{\mathcal{X}}$. This factor is in fact an upper bound on the variation of the $(T-t)$-state-action value function given any compatible environment $(f',\rho')$ that can occur when "jumping" from $\left(y^{l_{t-1}}, u_t\right)$ to $\left(x^{l_t}, u_t\right)$. An illustration of this is given in Figure 3.1.

### 3.5.2 Tightness of highest lower bound over all compatible sequences of one-step transitions

We define the highest lower bound over all compatible sequences of one-step transitions:

**Definition 3.5.7 (Highest lower bound)**
$\forall x_0 \in \mathcal{X}$,

$$L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) = \max_{\tau \in \mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})}} B(\tau, x_0)\ . \tag{3.91}$$

We analyze in this subsection the distance from the lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ to the actual return $J^{u_0,\ldots,u_{T-1}}(x_0)$ as a function of the sample sparsity. The sample sparsity is defined as follows:

**Definition 3.5.8 (Sample sparsity)**
*Let $a \in \mathcal{U}$, and let $\mathcal{F}_{n,a}$ be defined as follows:*

$$\mathcal{F}_{n,a} = \left\{ \left(x^l, u^l, r^l, y^l\right) \in \mathcal{F}_n | u^l = a \right\} \tag{3.92}$$

*($\forall a,\ \mathcal{F}_{n,a} \neq \emptyset$ since each action $a$ appears at least once in $\mathcal{F}_n$). Since $\mathcal{X}$ is a compact subset of $\mathbb{R}^{d_{\mathcal{X}}}$, it is bounded and there exists $\alpha \in \mathbb{R}^+$ :*

$$\forall a \in \mathcal{U}\ , \sup_{x' \in X} \left\{ \min_{(x^l,u^l,r^l,y^l) \in \mathcal{F}_{n,a}} \left\{ \left\| x^l - x' \right\|_{\mathcal{X}} \right\} \right\} \leq \alpha\ . \tag{3.93}$$

*The smallest $\alpha$ which satisfies equation (3.93) is named the sample sparsity and is denoted by $\alpha^*_{\mathcal{F}_n}$.*

We have the following theorem.

**Theorem 3.5.9 (Tightness of highest lower bound)**
$\exists\, C > 0 : \forall x_0 \in \mathcal{X}, \forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T,$

$$J^{u_0,\ldots,u_{T-1}}(x_0) - L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq C\alpha^*_{\mathcal{F}_n}. \tag{3.94}$$

**Proof.** Let

$$(x_0, u_0, r_0, x_1, u_1, \ldots, x_{T-1}, u_{T-1}, r_{T-1}, x_T) \tag{3.95}$$

be the trajectory of an agent starting from $x_0 = x$ when following the open-loop policy $u_0, \ldots, u_{T-1}$ under the (actual) environment $(f, \rho)$. Using equation (3.93), we define the sequence of transitions $\tau$:

$$\tau = \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} \in \mathcal{F}^T_{n,(u_0,\ldots,u_{T-1})} \tag{3.96}$$

that satisfies $\forall t \in \{0, 1, \ldots, T-1\}$

$$\left\|x^{l_t} - x_t\right\|_{\mathcal{X}} = \min_{l \in \{1,\ldots,n\}} \left\|x^l - x_t\right\|_{\mathcal{X}} \leq \alpha^*_{\mathcal{F}_n}. \tag{3.97}$$

We have

$$B(\tau, x_0) = \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}}\right] \tag{3.98}$$

with $y^{l_{-1}} = x_0$. Let us focus on $\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}}$. We have

$$\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}} = \left\|x^{l_t} - x_t + x_t - y^{l_{t-1}}\right\|_{\mathcal{X}}, \tag{3.99}$$

and hence

$$\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}} \leq \left\|x^{l_t} - x_t\right\|_{\mathcal{X}} + \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}}. \tag{3.100}$$

Using inequality (3.97), we can write

$$\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}} \leq \alpha^*_{\mathcal{F}_n} + \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}}. \tag{3.101}$$

For $t = 0$, one has

$$\begin{aligned} \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} &= \|x_0 - x_0\|_{\mathcal{X}} && (3.102)\\ &= 0\,. && (3.103) \end{aligned}$$

For $t > 0$,

$$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} = \left\|f\left(x_{t-1}, u_{t-1}\right) - f\left(x^{l_{t-1}}, u_{t-1}\right)\right\|_{\mathcal{X}} \tag{3.104}$$

and the Lipschitz continuity of $f$ implies that

$$\left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} \leq L_f \left\|x_{t-1} - x^{l_{t-1}}\right\|_{\mathcal{X}} . \tag{3.105}$$

So, as

$$\|x_{t-1} - x^{l_{t-1}}\|_{\mathcal{X}} \leq \alpha^*_{\mathcal{F}_n} , \tag{3.106}$$

we have

$$\forall t > 0,\ \left\|x_t - y^{l_{t-1}}\right\|_{\mathcal{X}} \leq L_f \alpha^*_{\mathcal{F}_n} . \tag{3.107}$$

Equations (3.101) and (3.107) imply that for $t > 0$,

$$\left\|y^{l_{t-1}} - x^{l_t}\right\|_{\mathcal{X}} \leq\ \alpha^*_{\mathcal{F}_n}(1 + L_f) \tag{3.108}$$

and, for $t = 0$,

$$\left\|y^{l_{-1}} - x^{l_0}\right\|_{\mathcal{X}} \leq \alpha^*_{\mathcal{F}_n} \leq \alpha^*_{\mathcal{F}_n}(1 + L_f) . \tag{3.109}$$

This gives

$$B(\tau, x_0) \geq \sum_{t=0}^{T-1} \left[r^{l_t} - L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n}(1 + L_f)\right] \doteq B . \tag{3.110}$$

We also have, by definition of $L^{u_0,\ldots,u_{T-1}}(x_0)$ ,

$$J^{u_0,\ldots,u_{T-1}}(x_0) \geq L^{u_0,\ldots,u_{T-1}}(x_0) \geq B(\tau, x_0) \geq B . \tag{3.111}$$

Thus,

$$\begin{aligned}
&|J^{u_0,\ldots,u_{T-1}}(x_0) - L^{u_0,\ldots,u_{T-1}}(x_0)| \\
&\leq |J^{u_0,\ldots,u_{T-1}}(x_0) - B| && (3.112)\\
&= J^{u_0,\ldots,u_{T-1}}(x_0) - B && (3.113)\\
&= \left|\sum_{t=0}^{T-1} \left[\left(r_t - r^{l_t}\right) + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n}(1 + L_f)\right]\right| && (3.114)\\
&\leq \sum_{t=0}^{T-1} \left[\left|r_t - r^{l_t}\right| + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n}(1 + L_f)\right] . && (3.115)
\end{aligned}$$

The Lipschitz continuity of $\rho$ allows to write

$$\begin{aligned} \left| r_t - r^{l_t} \right| &= \left| \rho(x_t, u_t) - \rho\left(x^{l_t}, u_t\right) \right| && (3.116) \\ &\leq L_\rho \left\| x_t - x^{l_t} \right\|_{\mathcal{X}}, && (3.117) \end{aligned}$$

and using inequality (3.97), we have

$$\left| r'_t - r^{l_t} \right| \leq L_\rho \alpha^*_{\mathcal{F}_n}. \tag{3.118}$$

Finally, we obtain

$$\begin{aligned} J^{u_0,\ldots,u_{T-1}}(x_0) - B &\leq \sum_{t=0}^{T-1} \left[ L_\rho \alpha^*_{\mathcal{F}_n} + L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f) \right] && (3.119) \\ &\leq T L_\rho \alpha^*_{\mathcal{F}_n} + \sum_{t=0}^{T-1} L_{Q_{T-t}} \alpha^*_{\mathcal{F}_n} (1 + L_f) && (3.120) \\ &\leq \alpha^*_{\mathcal{F}_n} \left( T L_\rho + \sum_{t=0}^{T-1} L_{Q_{T-t}} (1 + L_f) \right). && (3.121) \end{aligned}$$

Thus

$$J^{u_0,\ldots,u_{T-1}}(x_0) - L^{u_0,\ldots,u_{T-1}}(x_0) \leq \Big( T L_\rho + (1 + L_f) \sum_{t=0}^{T-1} L_{Q_{T-t}} \Big) \alpha^*_{\mathcal{F}_n}, \tag{3.122}$$

which completes the proof. ■

The lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$ thus converges to the $T-$stage return of the sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ when the sample sparsity $\alpha^*_{\mathcal{F}_n}$ decreases to zero.

## 3.6 Computing a sequence of actions maximizing the highest lower bound

Let $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$ be the set of sequences of actions maximizing the highest lower bound:

**Definition 3.6.1 (Sequences of actions maximizing the highest lower bound)**

$$\forall x_0 \in \mathcal{X}, \mathfrak{L}^*_{\mathcal{F}_n}(x_0) = \underset{(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T}{\arg\max} \left\{ L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \right\}. \tag{3.123}$$

The CGRL algorithm computes for each initial state $x_0 \in \mathcal{X}$ a sequence of actions $(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ that belongs to $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$. From what precedes, it follows that the actual return $J^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0)$ of this sequence is lower-bounded as follows:

$$\max_{(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T} L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0) \leq J^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0) \,. \tag{3.124}$$

Due to the tightness of the lower bound $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$, the value of the return which is guaranteed will converge to the true return of the sequence of actions when $\alpha^*_{\mathcal{F}_n}$ decreases to zero. Additionally, we prove in Section 3.6.1 that when the sample sparsity $\alpha^*_{\mathcal{F}_n}$ decreases below a particular threshold, the sequence

$$(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \mathcal{U}^T \tag{3.125}$$

is optimal. To identify a sequence of actions that belongs to $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$ without computing for all sequences $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ the value $L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_n}(x_0)$, the CGRL algorithm exploits the fact that the problem of finding an element of $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$ can be reformulated as a shortest path problem.

### 3.6.1 Convergence of $(\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ towards an optimal sequence of actions

We prove hereafter that when $\alpha^*_{\mathcal{F}_n}$ gets lower than a particular threshold, the CGRL algorithm can only output optimal policies.

**Theorem 3.6.2 (Convergence of the CGRL algorithm)**
*Let $x_0 \in \mathcal{X}$. Let*

$$\mathfrak{J}^*(x_0) = \left\{(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T | J^{u_0,\ldots,u_{T-1}}(x_0) = J^*(x_0)\right\} \,, \tag{3.126}$$

*and let us suppose that*

$$\mathfrak{J}^*(x_0) \neq \mathcal{U}^T \tag{3.127}$$

*(if $\mathfrak{J}^*(x_0) = \mathcal{U}^T$, the search for an optimal sequence of actions is indeed trivial). We define*

$$\epsilon(x_0) = \min_{(u_0,\ldots,u_{T-1}) \in \mathcal{U}^T \setminus \mathfrak{J}^*(x_0)} \left\{J^*(x_0) - J^{u_0,\ldots,u_{T-1}}(x_0)\right\} . \tag{3.128}$$

*Then*

$$C\alpha^*_{\mathcal{F}_n} < \epsilon(x_0) \implies (\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \mathfrak{J}^*(x_0) \,. \tag{3.129}$$

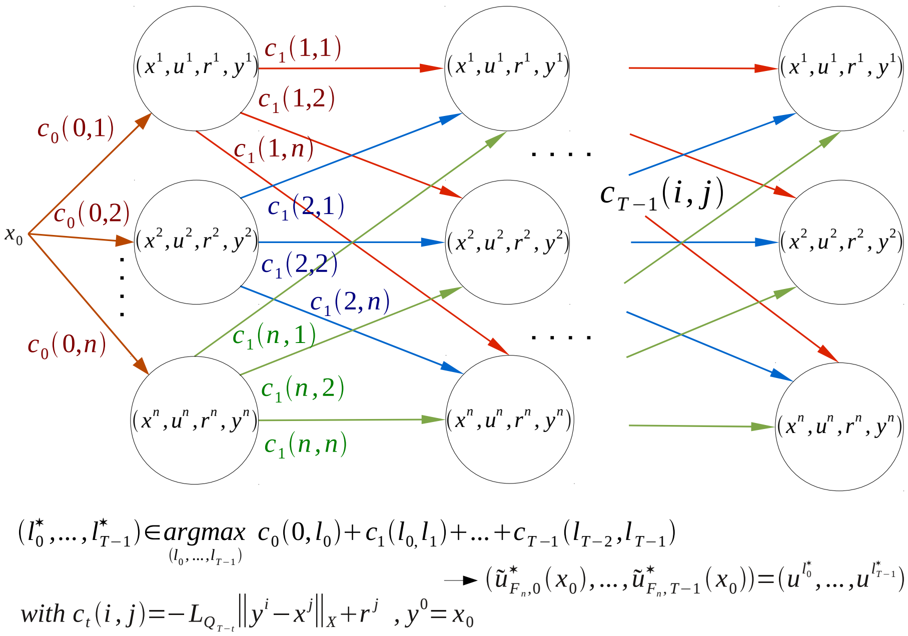

Figure 3.2: A graphical interpretation of the CGRL algorithm.

**Proof.** Let us prove that by *Reductio ad absurdum*. Let us suppose that the algorithm does not return an optimal sequence of actions, which means that

$$J^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0) \leq J^*(x_0) - \epsilon(x_0) \ . \tag{3.130}$$

Let us consider a sequence $u_0^*(x_0), \ldots, u_{T-1}^*(x_0)$ such that

$$(u_0^*(x_0), \ldots, u_{T-1}^*(x_0)) \in \mathfrak{J}^*(x_0) \ . \tag{3.131}$$

Then,

$$J^{u_0^*(x_0),\ldots,u_{T-1}^*(x_0)}(x_0) = J^*(x_0). \tag{3.132}$$

The lower bound $L^{u_0^*(x_0),\ldots,u_{T-1}^*(x_0)}(x_0)$ satisfies the relationship

$$J^*(x_0) - L^{u_0^*(x_0),\ldots,u_{T-1}^*(x_0)}(x_0) \leq C\alpha^*_{\mathcal{F}_n}. \tag{3.133}$$

Knowing that

$$C\alpha^*_{\mathcal{F}_n} < \epsilon(x_0), \tag{3.134}$$

we have

$$L^{u_0^*(x_0),\ldots,u_{T-1}^*(x_0)}(x_0) > J^*(x_0) - \epsilon(x_0). \tag{3.135}$$

By definition of $\epsilon(x_0)$,

$$J^*(x_0) - \epsilon(x_0) \geq J^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0), \tag{3.136}$$

and since

$$J^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0) \geq L^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0), \tag{3.137}$$

we have

$$L^{u_0^*(x_0),\ldots,u_{T-1}^*(x_0)}(x_0) > L^{\tilde{u}^*_{\mathcal{F}_n,0}(x_0),\ldots,\tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)}(x_0) \ , \tag{3.138}$$

which contradicts the fact that the algorithm returns the sequence that leads to the highest lower bound. ■

### 3.6.2 Cautious Generalization Reinforcement Learning algorithm

The CGRL algorithm computes an element of the set $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$ defined previously.

**Definition 3.6.3**
*Let*

$$\mathcal{D} : \mathcal{F}_n^T \to \mathcal{U}^T \tag{3.139}$$

*be the operator that maps a sequence of one-step system transitions*

$$\tau = \left[ (x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}) \right]_{t=0}^{T-1} \in \mathcal{F}_n^T \tag{3.140}$$

*into the sequence of actions* $(u^{l_0}, \ldots, u^{l_{T-1}})$*:*

$$\forall \tau = \left[ (x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}) \right]_{t=0}^{T-1}, \quad \mathcal{D}(\tau) = (u^{l_0}, \ldots, u^{l_{T-1}}) \, . \tag{3.141}$$

Using this operator, we can write

$$\forall x_0 \in \mathcal{X},$$
$$\mathfrak{L}^*_{\mathcal{F}_n}(x_0) = \left\{ (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T \left| \begin{array}{l} \exists \tau \in \underset{\tau \in \mathcal{F}_n^T}{\arg\max} \left\{ B(\tau, x_0) \right\} \, , \\ \mathcal{D}(\tau) = (u_0, \ldots, u_{T-1}) \end{array} \right. \right\} . \tag{3.142}$$

Or, equivalently
$\forall x_0 \in \mathcal{X}$,

$$\mathfrak{L}^*_{\mathcal{F}_n}(x_0) =$$
$$\left\{ (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T \left| \begin{array}{l} \exists \tau \in \underset{\tau \in \mathcal{F}_n^T}{\arg\max} \sum_{t=0}^{T-1} \left[ r^{l_t} - L_{Q_{T-t}} \| y^{l_{t-1}} - x^{l_t} \|_{\mathcal{X}} \right] , \\ \mathcal{D}(\tau) = (u_0, \ldots, u_{T-1}) \end{array} \right. \right\} . \tag{3.143}$$

From this expression, we can notice that a sequence of one-step transitions $\tau$ such that $\mathcal{D}(\tau)$ belongs to $\mathfrak{L}^*_{\mathcal{F}_n}(x_0)$ can be obtained by solving a shortest path problem on the graph given in Figure 3.2. The CGRL algorithm works by solving this problem using the Viterbi algorithm and by applying the operator $\mathcal{D}$ to the sequence of one-step transitions $\tau$ corresponding to its solution. Its complexity is quadratic with respect to the cardinality $n$ of the input sample $\mathcal{F}_n$ and linear with respect to the optimization horizon $T$.

## 3.7 Illustration

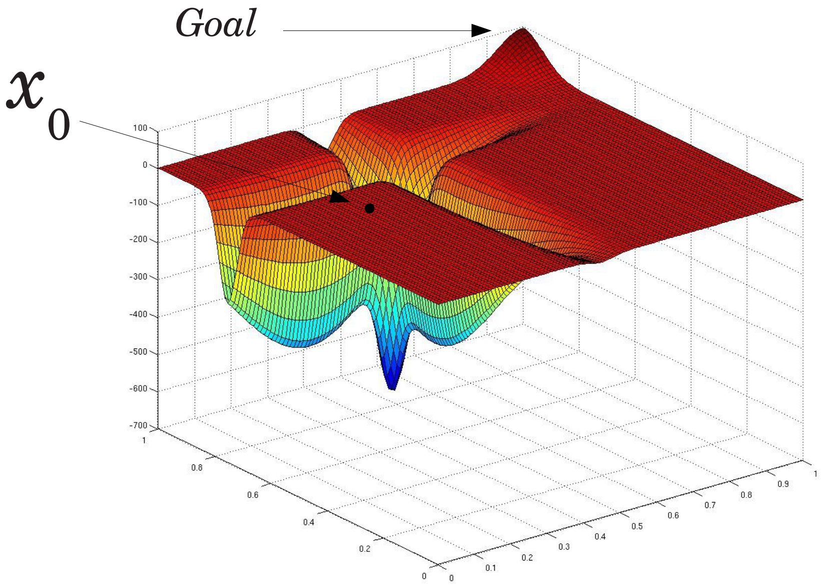

Figure 3.3: The puddle world benchmark.

In this section, we illustrate the CGRL algorithm on a variant of the puddle world benchmark introduced in [27]. In this benchmark, a robot whose goal is to collect high cumulated rewards navigates on a plane. A puddle stands in between the initial position of the robot and the high reward area. If the robot is in the puddle, it gets highly negative rewards. An optimal navigation strategy drives the robot around the puddle to reach the high reward area. Two datasets of one-step transitions have been used in our example. The first set $\mathcal{F}$ contains elements that uniformly cover the area of the state space that can be reached within $T$ steps. The set $\mathcal{F}'$ has been obtained by removing from $\mathcal{F}$ the elements corresponding to the highly negative rewards.[2]

[2] Although this problem might be treated by on-line learning methods, in some settings - for whatever reason - on-line learning may be impractical and all one will have is a batch of trajectories

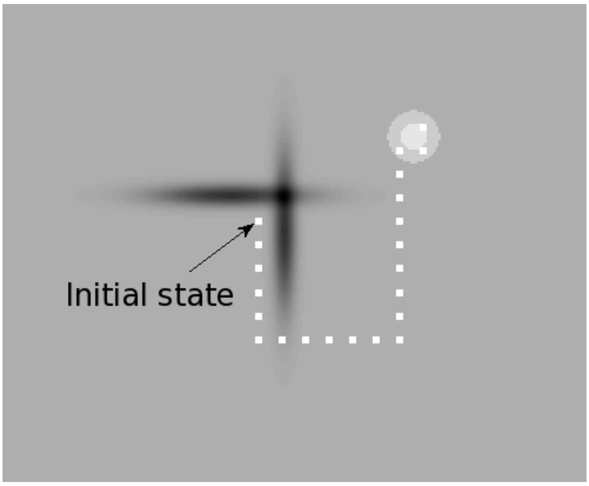

Figure 3.4: CGRL with $\mathcal{F}$.

The full specification of the puddle world benchmark and the exact procedure for generating $\mathcal{F}$ and $\mathcal{F}'$ is the following. The state space $\mathcal{X}$ is

$$\mathcal{X} = \mathbb{R}^2 \ . \tag{3.144}$$

The action space $\mathcal{U}$ is given by

$$\mathcal{U} = \{\begin{pmatrix}0.1 & 0\end{pmatrix}, \begin{pmatrix}-0.1 & 0\end{pmatrix}, \begin{pmatrix}0 & 0.1\end{pmatrix}, \begin{pmatrix}0 & -0.1\end{pmatrix}\}. \tag{3.145}$$

The system dynamics $f$ is defined as follows:

$$f(x, u) = x + u \ , \tag{3.146}$$

and the reward function $\rho$:

$$\rho(x, u) = k_1 \mathcal{N}_{\mu_1, \Sigma_1}(x) - k_2 \mathcal{N}_{\mu_2, \Sigma_2}(x) - k_3 \mathcal{N}_{\mu_3, \Sigma_3}(x) \ , \tag{3.147}$$

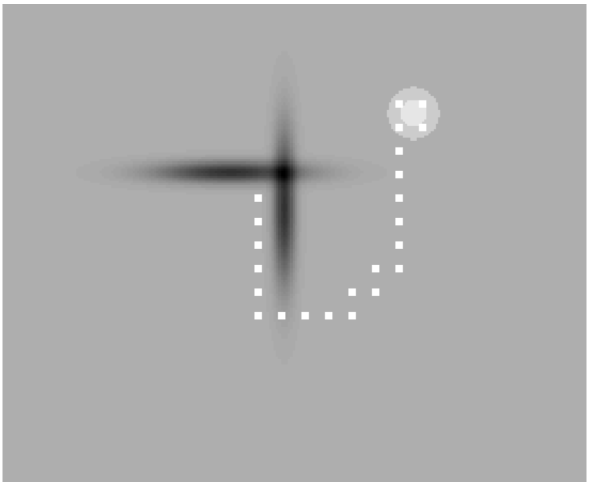

Figure 3.5: FQI with $\mathcal{F}$.

where

$$\mathcal{N}_{\mu,\Sigma}(x) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{\frac{-(x-\mu)\Sigma^{-1}(x-\mu)'}{2}}, \tag{3.148}$$

$$\mu_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}, \tag{3.149}$$

$$\mu_2 = \begin{pmatrix} 0.225 & 0.75 \end{pmatrix}, \tag{3.150}$$

$$\mu_3 = \begin{pmatrix} 0.45 & 0.6 \end{pmatrix}, \tag{3.151}$$

$$\Sigma_1 = \begin{pmatrix} 0.005 & 0 \\ 0 & 0.005 \end{pmatrix}, \tag{3.152}$$

$$\Sigma_2 = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.001 \end{pmatrix}, \tag{3.153}$$

$$\Sigma_3 = \begin{pmatrix} 0.001 & 0 \\ 0 & 0.05 \end{pmatrix}, \tag{3.154}$$

and

$$k_1 = 1, k_2 = k_3 = 20. \tag{3.155}$$

The Lipschitz constants $L_f$ and $L_\rho$ are

$$L_f = 1, L_\rho = 1.3742 * 10^6 \ . \tag{3.156}$$

The time horizon $T$ is set to $T = 25$, and the initial state of the system to

$$x_0 = (0.35, 0.65). \tag{3.157}$$

The sets of one-step system transitions are

$$\mathcal{F} = \left\{ (x, u, \rho(x,u), f(x,u)) \,\middle|\, \begin{pmatrix} x \in \left\{ \left(-2.15 + \frac{5i}{203}, -1.85 + \frac{5j}{203}\right) | i, j = 1 : 203 \right\} \\ u \in \mathcal{U} \end{pmatrix} \right\}, \tag{3.158}$$

$$\mathcal{F}' = \mathcal{F} \backslash \left\{ (x, u, r, y) \in \mathcal{F}_1 | x \in [0.4, 0.5] \times [0.25, 0.95] \cup [-0.1, 0.6] \times [0.7, 0.8] \right\}. \tag{3.159}$$

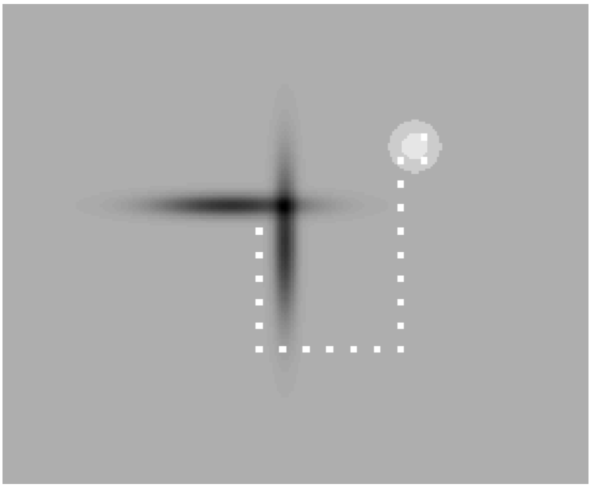

Figure 3.6: CGRL with $\mathcal{F}'$.

On Figure 3.4, we have drawn the trajectory of the robot when following the sequence of actions computed by the CGRL algorithm. Every state encountered is represented by a white square. The plane upon which the robot navigates has been colored such that the darker the area, the smaller the corresponding rewards are. In particular, the puddle area is colored in dark grey/black. We see that the CGRL policy drives the robot around the puddle to reach the high-reward area $-$ which is represented by the

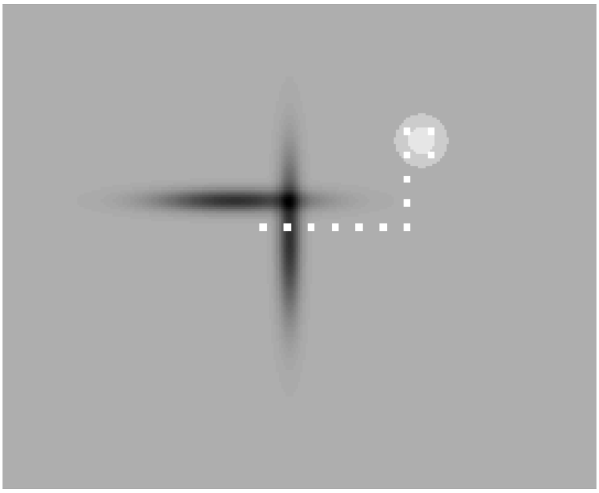

Figure 3.7: FQI with $\mathcal{F}'$.

light-grey circles. The CGRL algorithm also computes a lower bound on the cumulated rewards obtained by this action sequence. Here, we found out that this lower bound was rather conservative.

Figure 3.5 represents the policy inferred from $\mathcal{F}$ by using the (finite-time version of the) Fitted $Q$ Iteration algorithm (FQI) combined with extremely randomized trees as function approximators [9] (the FQI algorithm is also described in Appendix A). The FQI algorithm combined with extremely randomized trees is run using its default parameters given in [9]. The trajectories computed by the CGRL and FQI algorithms are very similar and so are the sums of rewards obtained by following these two trajectories. However, by using $\mathcal{F}'$ rather that $\mathcal{F}$, the CGRL and FQI algorithms do not lead to similar trajectories, as it is shown on Figures 3.6 and 3.7. Indeed, while the CGRL policy still drives the robot around the puddle to reach the high reward area, the FQI policy makes the robot cross the puddle. In terms of optimality, this latter navigation strategy is much worse. The difference between both navigation strategies can be explained as follows. The FQI algorithm behaves as if it were associating to areas of the state space that are not covered by the input sample, the properties of the elements of this sample that are located in the neighborhood of these areas. This in turn explains why it computes a policy that makes the robot cross the puddle. The same behavior could probably be observed by using other algorithms that combine dynamic programming strategies with kernel-based approximators or averagers [3, 15, 22]. The CGRL algorithm generalizes the information contained in the dataset, by assuming, given the initial state, the most adverse behavior for the environment according to its weak prior knowledge about the environment. This results in the fact that the CGRL algorithm penalizes sequences of decisions that could drive the robot in areas not well covered by the sample, and this explains why the CGRL algorithm drives the robot around the puddle when run with $\mathcal{F}'$.

## 3.8 Discussion

The CGRL algorithm outputs a sequence of actions as well as a lower bound on its return. When $L_f > 1$ (e.g. when the system is unstable), this lower bound will decrease exponentially with $T$. This may lead to very low performance guarantees when the optimization horizon $T$ is large. However, one can also observe that the terms $L_{Q_{T-t}}$ – which are responsible for the exponential decrease of the lower bound with the optimization horizon – are multiplied by the distance between the end state of a one-step transition and the beginning state of the next one-step transition of the sequence $\tau$ $(\|y^{l^*_{t-1}} - x^{l^*_t}\|_{\mathcal{X}})$ solution of the shortest path problem of Figure 3.2. Therefore, if these states $y^{l^*_{t-1}}$ and $x^{l^*_t}$ are close to each other, the CGRL algorithm can lead to good performance guarantees even for large values of $T$. It is also important to notice that this lower bound does not depend explicitly on the sample sparsity $\alpha^*_{\mathcal{F}_n}$, but depends rather on the initial state for which the sequence of actions is computed. Therefore, this may lead to cases where the CGRL algorithm provides good performance guarantees for some specific initial states, even if the sample does not cover every area of the state space well enough.

Other RL algorithms working in a similar setting as the CGRL algorithm, while not exploiting the weak prior knowledge about the environment, do not output a lower bound on the return of the policy $h$ they infer from the sample of trajectories $\mathcal{F}_n$. However, some lower bounds on the return of $h$ can still be computed. For instance, this can be done by exploiting the results of [11] (reported in Chapter 2) upon which the CGRL algorithm is based. However, one can show that following the strategy described in [11] would necessarily lead to a bound lower than the lower bound associated to the sequence of actions computed by the CGRL algorithm. Another strategy would be to design global lower bounds on their policy by adapting proofs used to establish the consistency of these algorithms. As a way of example, by proceeding like this, we can design a lower bound on the return of the policy given by the FQI algorithm when combined with some specific approximators which have, among others, Lipschitz continuity properties. These algorithms compute a sequence of state-action value functions

$$\tilde{Q}_1, \tilde{Q}_2, \ldots, \tilde{Q}_T \tag{3.160}$$

and compute the policy $h : \{0, 1, \ldots, T-1\} \times \mathcal{X}$ defined as follows :

$$\forall (x,t) \in \mathcal{X} \times \{0, \ldots, T-1\}, h(t,x) \in \arg\max_{u \in \mathcal{U}} \tilde{Q}_{T-t}(x,u). \tag{3.161}$$

For instance when using kernel-based approximators [22], we have as result that the return of $h$ when starting from a state $x_0$ is bounded as follows:

$$\forall x_0 \in \mathcal{X}, J^h(x_0) \geq \tilde{Q}_T(x_0, h(0, x_0)) - (C_1 T + C_2 T^2) \cdot b \tag{3.162}$$

where $C_1$ and $C_2$ depends on $L_f$, $L_\rho$, the Lipschitz constants of the class of approximation and an upper bound on $\rho$, and $b$ is the bandwidth parameter (the proof of this result can be found in [14], also reported in Appendix B). The dependence of this lower bound on $\alpha^*_{\mathcal{F}_n}$ (through the choice of the bandwidth parameter $b$) as well as the large values of $C_1$ and $C_2$ tend to lead to a very conservative lower bound, especially when $\mathcal{F}_n$ is sparse.

## 3.9 Conclusions

In this chapter, we have considered $\min\max$-based approaches for addressing the generalization problem in RL. In particular, we have proposed and studied an algorithm that outputs a policy that maximizes a lower bound on the worst return that may be obtained with an environment compatible with some observed system transitions. The proposed algorithm is of polynomial complexity and avoids regions of the state space where the sample density is too low according to the prior information. A simple example has illustrated that this strategy can lead to cautious policies where other batch-mode RL algorithms fail because they unsafely generalize the information contained in the dataset.

From the results given in [11], it is also possible to derive in a similar way tight upper bounds on the return of a policy. In this respect, it would also be possible to adopt a "$\max\max$" generalization strategy by inferring policies that maximize these tight upper bounds. We believe that exploiting together the policy based on a $\min\max$ generalization strategy and the one based on a $\max\max$ generalization strategy could offer interesting possibilities for addressing the exploitation-exploration trade-off faced when designing intelligent agents. For example, if the policies coincide, it could be an indication that further exploration is not needed.

When using batch mode reinforcement learning algorithms to design autonomous intelligent agents, a problem arises. After a long enough time of interaction with their environment, the sample the agents collect may become so large that batch mode RL-techniques may become computationally impractical, even with small degree polynomial algorithms. As suggested by [8], a solution for addressing this problem would be to retain only the most "informative samples". In the context of the proposed algorithm, the complexity for computing the optimal sequence of decisions is quadratic in the size of the dataset. We believe that it would be interesting to design lower complexity algorithms based on sub-sampling the dataset based on the initial state information.

The work reported in this chapter has been carried out in the particular context of deterministic Lipschitz continuous environments. We believe that extending this work to environments which satisfy other types of properties (for instance, Hölder continuity assumptions or properties that are not related with continuity) or which are possibly also stochastic is a natural direction for further research.

# Bibliography

[1] A. Bemporad and M. Morari. Robust model predictive control: A survey. *Robustness in Identification and Control*, 245:207–226, 1999.

[2] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] J.A. Boyan and A.W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems 7 (NIPS 1995)*, pages 369–376, Denver, CO, USA, 1995. MIT Press.

[4] Suman Chakratovorty and David Hyland. Minimax reinforcement learning. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, USA, 2003.

[5] Balázs Csanád Csáji and László Monostori. Value function based reinforcement learning in changing Markovian environments. *Journal of Machine Learning Research*, 9:1679–1709, 2008.

[6] Mark De Berg, Otfried Cheong, Marc Van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.

[7] E. Delage and S. Mannor. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 2006.

[8] D. Ernst. Selecting concise sets of samples for a reinforcement learning agent. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*, Singapore, 2005.

[9] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[10] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel. Reinforcement learning versus model predictive control: a comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39:517–529, 2009.

[11] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.

[12] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series: Communications in Computer and Information Science (CCIS)*, volume 129, pages 61–77. Springer, Heidelberg, 2011.

[13] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. A cautious approach to generalization in reinforcement learning. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010)*, Valencia, Spain, 2010.

[14] Raphael Fonteneau, Susan A. Murphy, Louis Wehenkel, and Damien Ernst. Computing bounds for kernel-based policy evaluation in reinforcement learning. Technical report, Arxiv, 2010.

[15] G.J. Gordon. *Approximate Solutions to Markov Decision Processes*. PhD thesis, Carnegie Mellon University, 1999.

[16] J.E. Ingersoll. *Theory of Financial Decision Making*. Rowman and Littlefield Publishers, Inc., 1987.

[17] M.G. Lagoudakis and R. Parr. Least-squares policy iteration. *Jounal of Machine Learning Research*, 4:1107–1149, 2003.

[18] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML 1994)*, New Brunswick, NJ, USA, 1994.

[19] S. Mannor, D. Simester, P. Sun, and J.N. Tsitsiklis. Bias and variance in value function estimation. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)*, Banff, Alberta, Canada, 2004.

[20] S.A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, 65(2):331–366, 2003.

[21] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[22] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[23] M. Qian and S.A. Murphy. Performance guarantees for individualized treatment rules. Technical Report 498, Department of Statistics, University of Michigan, 2009.

[24] M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML 2005)*, pages 317–328, Porto, Portugal, 2005.

[25] Maria Rovatous and Michail Lagoudakis. Minimax search and reinforcement learning for adversarial tetris. In *Proceedings of the 6th Hellenic Conference on Artificial Intelligence (SETN'10)*, Athens, Greece, 2010.

[26] Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.

[27] R.S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coding. In *Advances in Neural Information Processing Systems 8 (NIPS 1996)*, pages 1038–1044, Denver, CO, USA, 1996. MIT Press.

[28] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[29] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260– 269, 1967.

# Chapter 4

# Generating informative trajectories by using bounds on the return of control policies

*We propose new methods for guiding the generation of informative trajectories when solving discrete-time optimal control problems. These methods exploit recently published results that provide ways for computing bounds on the return of control policies from a set of trajectories.*

The work presented in this chapter as been published as a 2-page highlight paper in the *Proceedings of the Workshop on Active Learning and Experimental Design* [4] (In conjunction with AISTATS 2010).

In this chapter, we consider:

- a deterministic setting,
- a continuous state space and a finite action space.

## 4.1 Introduction

Discrete-time optimal control problems arise in many fields such as finance, medicine, engineering as well as artificial intelligence. Whatever the techniques used for solving such problems, their performance is related to the amount of information available on the system dynamics and the reward function of the optimal control problem.

In this chapter, we consider settings in which information on the system dynamics must be inferred from trajectories and, furthermore, due to cost and time constraints, only a limited number of trajectories can be generated. We assume that a regularity structure - given in the form of Lipschitz continuity assumptions - exists on the system dynamics and the reward function. Under such assumptions, we exploit recently published methods for computing bounds on the return of control policies from a set of trajectories ([1, 3, 2], reported in Chapters 2 and 3) in order to sample the state-action space so as to be able to discriminate between optimal and non-optimal policies.

## 4.2 Problem statement

We consider a discrete-time system whose dynamics over $T$ stages is described by a time-invariant equation

$$x_{t+1} = f(x_t, u_t) \quad t = 0, \dots, T-1, \tag{4.1}$$

where for all $t$, the state $x_t$ is an element of the compact normed state space $\mathcal{X}$ and $u_t$ is an element of the finite (discrete) action space $\mathcal{U}$. $T \in \mathbb{N}_0$ is referred to as the optimization horizon. An instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{4.2}$$

is associated with the action $u_t$ taken while being in state $x_t$. The initial state of the system is fixed to $x_0 \in \mathcal{X}$. For every policy $(u_0, \dots, u_{T-1}) \in \mathcal{U}^T$, the $T-$stage return of $(u_0, \dots, u_{T-1})$ is defined as follows:

**Definition 4.2.1 ($T-$stage return of the sequence $(u_0, \dots, u_{T-1})$)**
$\forall (u_0, \dots, u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$J^{u_0,\dots,u_{T-1}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t) \ , \tag{4.3}$$

*where*

$$x_{t+1} = f(x_t, u_t) \ , \forall t \in \{0, \dots, T-1\} \ . \tag{4.4}$$

**Definition 4.2.2 (Optimal policies)**
*For a given initial state $x_0 \in \mathcal{X}$, an optimal policy is a policy $\big(u_0^*(x_0), \ldots, u_{T-1}^*(x_0)\big)$ such that*

$$\big(u_0^*(x_0), \ldots, u_{T-1}^*(x_0)\big) \in \underset{(u_0,\ldots,u_{T-1})\in\mathcal{U}^T}{\arg\max} \left\{J^{u_0,\ldots,u_{T-1}}(x_0)\right\} \ . \tag{4.5}$$

Here, the functions $f$ and $\rho$ are assumed to be Lipschitz continuous:

**Assumption 4.2.3 (Lipschitz continuity of $f$ and $\rho$)**
$\exists L_f, L_\rho > 0$ :

$$\begin{aligned} \forall (x, x') \in \mathcal{X}^2, \forall u \in \mathcal{U}, \|f(x,u) - f(x',u)\|_{\mathcal{X}} &\leq L_f \|x - x'\|_{\mathcal{X}} \ , & (4.6) \\ |\rho(x,u) - \rho(x',u)| &\leq L_\rho \|x - x'\|_{\mathcal{X}} \ , & (4.7) \end{aligned}$$

*where $\|.\|_{\mathcal{X}}$ denotes the chosen norm over the state space $\mathcal{X}$. We also assume that we have access to two constants $L_f, L_\rho > 0$ satisfying the above inequalities.*

Initially, the values of $f$ and $\rho$ are only known for $n$ state-action pairs. These values are given in a set of one-step transitions

$$\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^n \tag{4.8}$$

where $\forall l \in \{1, \ldots, n\}$,

$$\begin{cases} y^l = f(x^l, u^l), \\ r^l = \rho(x^l, u^l). \end{cases} \tag{4.9}$$

We suppose that additional transitions can be sampled, and we detail hereafter a sampling strategy to select state-action pairs $(x, u)$ for generating $f(x, u)$ and $\rho(x, u)$ so as to be able to discriminate rapidly $-$ as new one-step transitions are generated $-$ between optimal and non-optimal policies.

# 4.3 Algorithm

The work presented in [3] and reported in Chapter 3 proposes a method for computing from any set of transitions $\mathcal{F}$ such that each action $u \in \mathcal{U}$ appears at least once in $\mathcal{F}$ and for any policy $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ a lower bound $L_{\mathcal{F}}^{u_0,\ldots,u_{T-1}}(x_0)$ and an upper bound $U_{\mathcal{F}}^{u_0,\ldots,u_{T-1}}(x_0)$ on $J^{u_0,\ldots,u_{T-1}}(x_0)$:

**Lemma 4.3.1 (Bounds on $J^{u_0,\ldots,u_{T-1}}(x_0)$)**
$\forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$L_{\mathcal{F}}^{u_0,\ldots,u_{T-1}}(x_0) \leq J^{u_0,\ldots,u_{T-1}}(x_0) \leq U_{\mathcal{F}}^{u_0,\ldots,u_{T-1}}(x_0) \, . \tag{4.10}$$

Furthermore, these bounds converge towards $J^{u_0,\ldots,u_{T-1}}(x_0)$ when the sparsity of $\mathcal{F}$ decreases towards zero.

Before describing our proposed sampling strategy, let us introduce a few definitions. First, note that a policy can only be optimal given a set of one-step transitions $\mathcal{F}$ if its upper bound is not lower than the lower bound of any element of $\mathcal{U}^T$. We qualify as “candidate optimal policies given $\mathcal{F}$” and we denote by $\Pi(\mathcal{F}, x_0)$ the set of policies which satisfy this property:

**Definition 4.3.2 (Candidate optimal policies given $\mathcal{F}$)**
$\forall x_0 \in \mathcal{X}$,

$$\begin{aligned}\Pi(\mathcal{F}, x_0) \quad = \quad & \Big\{ (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T | \\ & \forall (u'_0, \ldots, u'_{T-1}) \in \mathcal{U}^T, U_{\mathcal{F}}^{u_0,\ldots,u_{T-1}}(x_0) \geq L_{\mathcal{F}}^{u'_0,\ldots,u'_{T-1}}(x_0) \Big\} \, .\end{aligned} \tag{4.11}$$

We also define the set of “compatible transitions given $\mathcal{F}$” as follows:

**Definition 4.3.3 (Compatible transitions given $\mathcal{F}$)**
*A transition $(x, u, r, y) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X}$ is said compatible with the set of transitions $\mathcal{F}$ if:*

$\forall (x^l, u^l, r^l, y^l) \in \mathcal{F}$,

$$u^l = u \implies \begin{cases} \left| r - r^l \right| & \leq \quad L_\rho \| x - x^l \|_{\mathcal{X}} \, , \\ \left\| y - y^l \right\|_{\mathcal{X}} & \leq \quad L_f \| x - x^l \|_{\mathcal{X}} \, . \end{cases} \tag{4.12}$$

*We denote by $\mathcal{C}(\mathcal{F}) \subset \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{U}$ the set that gathers all transitions that are compatible with the set of transitions $\mathcal{F}$.*

Our sampling strategy generates new one-step transitions iteratively. Given an existing set $\mathcal{F}_m$ of $m$ one-step transitions, which is made of the elements of the initial set $\mathcal{F}_n$ and the $m$-$n$ one-step transitions generated during the first $m$-$n$ iterations of this algorithm, it selects as next sampling point $(x^{m+1}, u^{m+1}) \in \mathcal{X} \times \mathcal{U}$, the point that minimizes in the worst conditions the largest bound width among the candidate optimal policies at the next iteration:

$$(x^{m+1}, u^{m+1}) \in \underset{(x,u)\in\mathcal{X}\times\mathcal{U}}{\arg\min} \left\{ \max_{\substack{(r,y)\in\mathbb{R}\times\mathcal{X}\ s.t.\\ (x,u,r,y)\in\mathcal{C}(\mathcal{F}_m)}} \left\{ \max_{\substack{(u_0,\ldots,u_{T-1})\in\\ \Pi(\mathcal{F}_m\cup\{(x,u,r,y)\},x_0)}} \Delta^{u_0,\ldots,u_{T-1}}_{\mathcal{F}_m\cup\{(x,u,r,y)\}}(x_0) \right\} \right\} \tag{4.13}$$

where

$$\Delta^{u_0,\ldots,u_{T-1}}_{\mathcal{F}}(x_0) = U^{u_0,\ldots,u_{T-1}}_{\mathcal{F}}(x_0) - L^{u_0,\ldots,u_{T-1}}_{\mathcal{F}}(x_0)\ . \tag{4.14}$$

Based on the convergence properties of the bounds, we conjecture that the sequence $\left(\Pi\left(\mathcal{F}_m, x_0\right)\right)_{m\in\mathbb{N}}$ converges towards the set of all optimal policies in a finite number of iterations:

**Conjecture 4.3.4 (Finite convergence of** $\left(\Pi\left(\mathcal{F}_m, x_0\right)\right)_{m\in\mathbb{N}}$**)**
*$\forall x_0 \in \mathcal{X}$,*

$$\begin{aligned}&\exists m_0 \in \mathbb{N}_0 :\\ &\forall m\in\mathbb{N}, m \geq m_0 \implies \Pi\left(\mathcal{F}_m, x_0\right) = \underset{(u_0,\ldots,u_{T-1})\in\mathcal{U}^T}{\arg\max} \left\{J^{u_0,\ldots,u_{T-1}}(x_0)\right\}\ .\end{aligned} \tag{4.15}$$

The analysis of the theoretical properties of the sampling strategy and its empirical validation are left for future work.

# Bibliography

[1] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.

[2] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series: Communications in Computer and Information Science (CCIS)*, volume 129, pages 61–77. Springer, Heidelberg, 2011.

[3] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. A cautious approach to generalization in reinforcement learning. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010)*, Valencia, Spain, 2010.

[4] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Generating informative trajectories by using bounds on the return of control policies. In *Proceedings of the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*, 2010.

# Chapter 5

# Active exploration by searching for experiments that falsify the computed control policy

*We propose a strategy for experiment selection - in the context of reinforcement learning-based on the idea that the most interesting experiments to carry out at some stage are those that are the most liable to falsify the current hypothesis about the optimal control policy. We cast this idea in a context where a policy learning algorithm and a model identification method are given a priori. Experiments are selected if, using the learned environment model, they are predicted to yield a revision of the learned control policy. Algorithms and simulation results are provided for a deterministic system with discrete action space. They show that the proposed approach is promising.*

The work presented in this chapter has been accepted for publication in the*Proceedings of the IEEE Symposium Series in Computational Intelligence - Adaptive Dynamic Programming and Reinforcement Learning* (IEEE ADPRL 2011) [12].

In this chapter, we consider:

- a deterministic setting,
- a continuous state space and a finite action space.

## 5.1 Introduction

Many relevant decision problems in the field of engineering [20], finance [13], medicine ([16, 17]) or artificial intelligence [21] can be formalized as optimal control problems, which are problems where one seeks to compute a control policy so as to maximize a numerical performance criterion.

Often, for solving these problems, one has to deal with an incomplete knowledge of the two key elements of the optimal control problem, which are the system dynamics and the reward function. A vast literature has already proposed ways for computing approximate optimal solutions to these problems when the only information available on these elements is in the form of a set of system transitions, where every system transition is made of a state, the action taken while being in this state, and the values of the reward function and system dynamics observed in this state-action point. In particular, researchers in the field of reinforcement learning (RL) - where the goal was initially to design intelligent agents able to interact with an environment so as to maximize a numerical reward signal - have developed efficient algorithms to address this particular problem, commonly known as batch mode reinforcement learning (BMRL) algorithms.

In this chapter, we consider the problem of choosing additional data gathering experiments on the real system in order to complete an already available sample of system trajectories, so as to improve the policy learned by a given BMRL algorithm as much as possible, i.e. by using a minimum number of additional data gathering experiments. Our strategy is based on using a predictive model (PM) of the system performance inferred from the already collected datasets. The PM allows us to predict the outcome of new putative experiments with the real system in terms of putative trajectories, and hence to predict the effect of including these putative trajectories into the sample used by the BMRL algorithm in terms of their impact on the policy inferred by this algorithm. In order to choose the next experiment, we suggest that a good strategy is to select an experiment which (putatively) would lead to a revision of the policy learned from the augmented dataset. In essence, this strategy consists in always trying to find experiments which are likely to falsify the current hypothesis about the optimal control policy.

This approach relies on two intuitions backed by many works/numerical experiments in the field of optimal control. The first intuition is that if when adding a new system transition to the set of existing ones, the BMRL algorithm run on this new set outputs a policy that falsifies the previously computed policy, then this new system transition may be particularly informative. The second intuition is related to the fact that for many problems, one may easily use the already collected information on the dynamics and reward function to build a PM of the system. Based on these two observations, our approach (i) iteratively screens a set of potential sampling locations, i.e.

a set of state-action points candidate for sampling, (ii) computes for each one of these points a predicted system transition, and (iii) analyzes the influence that each such predicted transition would have on the policy computed by the BMRL algorithm when combined with the "true" system transitions previously collected. The output of this analysis is then used to (iv) select a sampling location which is "predicted" to generate a new system transition that falsifies the policy computed by the BMRL algorithm.

After detailing this approach and the context in which it is proposed in sections 5.2, 5.3 and 5.4, we report in section 5.5 simulation results with the car-on-the-hill problem. Section 5.6 discusses related work and Section 5.7 concludes.

## 5.2 Problem statement

We consider a deterministic time-invariant system whose discrete-time dynamics over $T$ stages is described by

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \dots, T-1, \tag{5.1}$$

where for all $t \in \{0, \dots, T-1\}$, the state $x_t$ is an element of the normed state space $(\mathcal{X}, \|.\|_{\mathcal{X}})$ and $u_t$ is an element of a finite action space $\mathcal{U} = \left\{a^1, \dots, a^m\right\}$ with $m \in \mathbb{N}_0$. $T \in \mathbb{N}_0$ denotes the finite optimization horizon. An instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{5.2}$$

is associated with the action $u_t \in \mathcal{U}$ taken while being in state $x_t \in \mathcal{X}$. We assume that the initial state of the system $x_0 \in \mathcal{X}$ is known. For a given sequence of actions $\mathbf{u} = (u_0, \dots, u_{T-1}) \in \mathcal{U}^T$, we denote by $J^{\mathbf{u}}(x_0)$ the $T-$stage return of the sequence of actions $\mathbf{u}$ when starting from $x_0$, defined as follows:

**Definition 5.2.1** (**$T-$stage return of the sequence of actions $\mathbf{u}$**)
$\forall x_0 \in \mathcal{X}, \forall \mathbf{u} \in \mathcal{U}^T$,

$$J^{\mathbf{u}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t) \tag{5.3}$$

*with*

$$x_{t+1} = f(x_t, u_t), \forall t \in \{0, \dots, T-1\}\ . \tag{5.4}$$

We denote by $J^*(x_0)$ the maximal value of $J^{\mathbf{u}}(x_0)$ over $\mathcal{U}^T$:

**Definition 5.2.2 (Maximal $T-$stage return and optimal sequences of actions)** $\forall x_0 \in \mathcal{X}$,

$$J^*(x_0) = \max_{\mathbf{u} \in \mathcal{U}^T} J^{\mathbf{u}}(x_0) \; . \tag{5.5}$$

*An optimal sequence of actions* $\mathbf{u}^*(x_0)$ *is a sequence for which*

$$J^{\mathbf{u}^*(x_0)}(x_0) = J^*(x_0) \; . \tag{5.6}$$

In the following, we call "system transition" a $4-$tuple

$$(x, u, \rho(x, u), f(x, u)) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X} \tag{5.7}$$

that gathers information on the functions $f$ and $\rho$ in a point $(x, u)$ of the state-action space $\mathcal{X} \times \mathcal{U}$. Batch mode RL algorithms ([8, 18, 20]) have been introduced to infer near optimal control policies from the sole knowledge of a sample of system transitions

$$\mathcal{F}_n = \left\{ \left(x^l, u^l, r^l, y^l\right) \right\}_{l=1}^{n} \tag{5.8}$$

where

$$\begin{cases} r^l = \rho(x^l, u^l), \\ y^l = f(x^l, u^l). \end{cases} \tag{5.9}$$

In the rest of this chapter, we denote by $BMRL$ a generic batch mode RL algorithm and by $BMRL(\mathcal{F}_n, x_0)$ the policy it computes.

The problem we address is to find a sampling strategy which allows to collect a set of system transitions $\mathcal{F}_n$ from which a high quality sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) \in \mathcal{U}^T$ can be inferred by $BMRL$, i.e. a sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) \in \mathcal{U}^T$ such that $J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)}(x_0)$ is as close as possible to $J^*(x_0)$. The sampling process is limited to $N_{\max} \in \mathbb{N}$ transitions, i.e. one can afford to collect at most $N_{\max}$ system transitions.

## 5.3 Iterative sampling strategy to collect informative system transitions

In this section we describe one way to implement the general falsification strategy presented in Section 5.1 for addressing the problem stated in Section 5.2.

Assuming that we are given a batch mode RL algorithm, $BMRL$, a predictive model $PM$, and a sequence of numbers $L_n \in \mathbb{N}_0$, we proceed iteratively, by carrying out the following computations at any iteration $n < N_{\max}$:

- Using the sample $\mathcal{F}_n = \left\{ \left(x^l, u^l, r^l, y^l\right) \right\}_{l=1}^{n}$ of already collected transitions, we first compute a sequence of actions

$$\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = BMRL(\mathcal{F}_n, x_0) \ . \tag{5.10}$$

- Next, we draw a state-action point $(x, u) \in \mathcal{X} \times \mathcal{U}$ according to a uniform probability distribution $p_{\mathcal{X} \times \mathcal{U}}(\cdot)$ over the state-action space $\mathcal{X} \times \mathcal{U}$:

$$(x, u) \sim p_{\mathcal{X} \times \mathcal{U}}(\cdot) \tag{5.11}$$

- Using the sample $\mathcal{F}_n$ and the predictive model $PM$, we then compute a "predicted" system transition by:

$$(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u)) = PM(\mathcal{F}_n, x, u) \ . \tag{5.12}$$

- Using $(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u))$, we build the "predicted" augmented sample by:

$$\hat{\mathcal{F}}_{n+1}(x, u) = \mathcal{F}_n \cup \{(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u))\} \ , \tag{5.13}$$

and use it to predict the revised policy by:

$$\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) = BMRL(\hat{\mathcal{F}}_{n+1}(x, u), x_0) \ . \tag{5.14}$$

  - If $\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) \neq \tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$, we consider $(x, u)$ as informative, because it is potentially falsifying our current hypothesis about the optimal control policy. We hence use it to make an experiment on the real-system so as to collect a new transition

$$\left(x^{n+1}, u^{n+1}, r^{n+1}, y^{n+1}\right) \tag{5.15}$$

with

$$\begin{cases} x_{n+1} = x, \\ u_{n+1} = u, \\ r^{n+1} = \rho(x, u), \\ y^{n+1} = f(x, u) \ . \end{cases} \tag{5.16}$$

and we augment the sample with it:

$$\mathcal{F}_{n+1} = \mathcal{F}_n \cup \left\{ \left(x^{n+1}, u^{n+1}, r^{n+1}, y^{n+1}\right) \right\} \ . \tag{5.17}$$

- If $\hat{\mathbf{u}}^*_{\hat{\mathcal{F}}_{\mathbf{n+1}}(\mathbf{x},\mathbf{u})}(x_0) = \tilde{\mathbf{u}}^*_{\mathcal{F}_{\mathbf{n}}}(x_0)$ , we draw another state-action point $(x', u')$ according to $p_{\mathcal{X}\times\mathcal{U}}(\cdot)$:

  $$(x', u') \sim p_{\mathcal{X}\times\mathcal{U}}(\cdot) \tag{5.18}$$

  and repeat the process of prediction followed by policy revision.

- If $L_n$ state-action points have been tried without yielding a potential falsifier of the current policy, we give up and merely draw a state-action point $\left(x^{n+1}, u^{n+1}\right)$ "at random" according to $p_{\mathcal{X}\times\mathcal{U}}(\cdot)$:

  $$\left(x^{n+1}, u^{n+1}\right) \sim p_{\mathcal{X}\times\mathcal{U}}(\cdot) \ , \tag{5.19}$$

  and augment $\mathcal{F}_n$ with the transition

  $$\left(x^{n+1}, u^{n+1}, \rho\left(x^{n+1}, u^{n+1}\right), f\left(x^{n+1}, u^{n+1}\right)\right) \ . \tag{5.20}$$

### 5.3.1 Influence of the $BMRL$ algorithm and the predictive model $PM$

For this iterative sampling strategy to behave well, the inference capabilities of the $BMRL$ algorithm it uses should obviously be as good as possible. Usually, BMRL algorithms rely on the training of function approximators [4] that either represent the system dynamics and the reward function of the underlying control problem, a (state-action) value function or a policy. Given the fact that here, at any iteration of the algorithm, the only knowledge on the problem is given in the form of a sample of system transitions, we advocate using BMRL algorithms with non-parametric function approximators such as, for example, nearest neighbor or tree-based methods.

The best predictive model $PM$ would be an algorithm that would, given a state action pair $(x, u)$, output a predicted transition equal to $(x, u, \rho(x, u), f(x, u))$. Since predicting with great accuracy $\rho(x, u)$ and $f(x, u)$ may be difficult, one could also imagine an algorithm that computes a set of predictions rather than a single "best guess". Indeed, with such a choice, it would be more likely that at least one of these predicted transitions would also lead to a predicted policy falsification if the exact one leads to a true policy falsification. However, working with a large predicted set may also increase the likelihood that a sampling location would be predicted as a policy falsifier while it is actually not the case. Notice also that if some prior knowledge on the problem is available, it may be possible to exploit it to define for a given sampling location, a set of transitions which is "compatible" with the previous samples collected (see, e.g., [10] where a compatible set is defined when assuming that the problem is Lipschitz continuous with known Lipschitz constants). This could be used to increase the performance of a prediction algorithm by avoiding incompatible predictions.

### 5.3.2 Influence of the $L_n$ sequence of parameters

$L_n$ sets the maximal number of trials for searching a new experiment when $n$ transitions have already been collected. Its value should be chosen large enough so as to ensure that, if there exist transitions that indeed lead to a policy falsification, one of those would be identified with high probability. It may however happen that, at some iteration $n$, there doesn't exist any (predicted) transition that would lead to a (predicted) policy falsification. In this case, our algorithm will conduct $L_n$ trials, which may be problematic from the computational point of view if $L_n$ is very large. Thus the choice of $L_n$ is a trade-off between the desirability to have at any iteration a high probability to find a sample that leads to a policy falsification, and the need to avoid excessive computations when such a sampling location does not exist.

## 5.4 $BMRL/PM$ implementation based on nearest-neighbor approximations

In this section, we present the batch mode RL algorithm $BMRL$ and the predictive model $PM$ to which our iterative sampling strategy will be applied in the context of simulations reported in Section 5.5.

As $BMRL$ algorithm, we have chosen a model learning–type RL algorithm. It first approximates the functions $f$ and $\rho$ from the available sample of system transitions, and then solves "exactly" the optimal control problem defined by these approximations. This algorithm is fully detailed in Section 5.4.1. In Section 5.4.2, we present the $PM$ used in our experiments. It computes its predictions based on the same approximations as those used by the $BMRL$ algorithm.

### 5.4.1 Choice of the inference algorithm $BMRL$

**Model learning–type RL**

Model learning–type RL aims at solving optimal control problems by approximating the unknown functions $f$ and $\rho$ and solving the so approximated optimal control problem instead of the unknown actual optimal control problem. The values $y^l$ (resp. $r^l$) of the function $f$ (resp. $\rho$) in the state-action points $(x^l, u^l) \quad l = 1 \dots n$ are used to learn a function $\tilde{f}_{\mathcal{F}_n}$ (resp. $\tilde{\rho}_{\mathcal{F}_n}$) over the whole space $\mathcal{X} \times \mathcal{U}$. The approximated optimal control problem defined by the functions $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$ is solved and its solution is kept as an approximation of the solution of the optimal control problem defined by the actual functions $f$ and $\rho$.

Given a sequence of actions $\mathbf{u} \in \mathcal{U}^T$ and a model learning–type RL algorithm, we denote by $\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0)$ the approximated $T-$stage return of the sequence of actions $\mathbf{u}$, i.e. the $T-$stage return when considering the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$:

**Definition 5.4.1 (Approximated $T-$stage return)**
$\forall \mathbf{u} \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) = \sum_{t=0}^{T-1} \tilde{\rho}_{\mathcal{F}_n}(\tilde{x}_t, u_t) \tag{5.21}$$

*with*

$$\tilde{x}_{t+1} = \tilde{f}_{\mathcal{F}_n}(\tilde{x}_t, u_t), \forall t \in \{0, \ldots, T-1\} \tag{5.22}$$

*and* $\tilde{x}_0 = x_0$.

We denote by $\tilde{J}^{*}_{\mathcal{F}_n}(x_0)$ the maximal approximated $T-$stage return when starting from the initial state $x_0 \in \mathcal{X}$ according to the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$:

**Definition 5.4.2 (Maximal approximated $T-$stage return)**
$\forall x_0 \in \mathcal{X}$,

$$\tilde{J}^{*}_{\mathcal{F}_n}(x_0) = \max_{\mathbf{u} \in \mathcal{U}^T} \tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) . \tag{5.23}$$

Using these notations, model learning–type RL algorithms aim at computing a sequence of actions $\tilde{\mathbf{u}}^{*}_{\mathcal{F}_\mathbf{n}}(x_0) \in \mathcal{U}^T$ such that $\tilde{J}^{\tilde{\mathbf{u}}^{*}_{\mathcal{F}_\mathbf{n}}(x_0)}_{\mathcal{F}_n}(x_0)$ is as close as possible (and ideally equal to) to $\tilde{J}^{*}_{\mathcal{F}_n}(x_0)$. These techniques implicitly assume that an optimal policy for the learned model leads also to high returns on the real problem.

### Voronoi tessellation-based RL algorithm

We describe here the model-learning type of RL algorithm that will be used later in our simulations. This algorithm approximates the reward function $\rho$ and the system dynamics $f$ using piecewise constant approximations on a Voronoi–like [2] partition of the state-action space (which is equivalent to a nearest-neighbour approximation) and will be referred to by the VRL algorithm. Given an initial state $x_0 \in \mathcal{X}$, the VRL algorithm computes an open-loop sequence of actions which corresponds to an "optimal navigation" among the Voronoi cells.

Before fully describing this algorithm, we first assume that all the state-action pairs $\left\{(x^l, u^l)\right\}_{l=1}^{n}$ given by the sample of transitions $\mathcal{F}_n$ are unique:

**Assumption 5.4.3**

$$\forall l, l' \in \{1, \ldots, n\}, (x^l, u^l) = (x^{l'}, u^{l'}) \implies l = l' \ . \tag{5.24}$$

We also assume that each action of the action space $\mathcal{U}$ has been tried at least once:

**Assumption 5.4.4**

$$\forall u \in \mathcal{U}, \exists l \in \{1, \ldots, n\}, u^l = u \ . \tag{5.25}$$

The model is based on the creation of $n$ Voronoi cells $\left\{V^l\right\}_{l=1}^n$ which define a partition of size $n$ of the state-action space. The Voronoi cell $V^l$ associated to the element $(x^l, u^l)$ of $\mathcal{F}_n$ is defined as the set of state-action pairs $(x, u) \in \mathcal{X} \times \mathcal{U}$ that satisfy:

$$(i) \quad u = u^l \ , \tag{5.26}$$

$$(ii) \quad l \in \underset{l' : u^{l'} = u}{\arg\min} \left\{ \|x - x^{l'}\|_{\mathcal{X}} \right\} \ , \tag{5.27}$$

$$(iii) \quad l = \min_{l'} \left\{ l' \in \underset{l' : u^{l'} = u}{\arg\min} \left\{ \|x - x^{l'}\|_{\mathcal{X}} \right\} \right\} . \tag{5.28}$$

One can verify that $\left\{V^l\right\}_{l=1}^n$ is indeed a partition of the state-action space $\mathcal{X} \times \mathcal{U}$ since every state-action $(x, u) \in \mathcal{X} \times \mathcal{U}$ belongs to one and only one Voronoi cell.

The function $f$ (resp. $\rho$) is approximated by a piecewise constant function $\tilde{f}_{\mathcal{F}_n}$ (resp. $\tilde{\rho}_{\mathcal{F}_n}$) defined as follows:

**Definition 5.4.5 (Approximations of $f$ and $\rho$)**

$$\forall l \in \{1, \ldots, n\}, \forall (x, u) \in V^l, \quad \tilde{f}_{\mathcal{F}_n}(x, u) = y^l, \tag{5.29}$$

$$\tilde{\rho}_{\mathcal{F}_n}(x, u) = r^l \ . \tag{5.30}$$

Using the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$, we define a sequence of approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$ as follows :

**Definition 5.4.6 (Approximated optimal state-action value functions)**
$\forall t \in \{0, \ldots, T-1\}, \forall (x, u) \in \mathcal{X} \times \mathcal{U}$ ,

$$\begin{aligned} \tilde{Q}^*_{T-t}(x, u) &= \tilde{\rho}_{\mathcal{F}_n}(x, u) \\ &+ \underset{u' \in \mathcal{U}}{\arg\max}\, \tilde{Q}^*_{T-t-1}\left(\tilde{f}_{\mathcal{F}_n}(x, u), u'\right) \ , \end{aligned} \tag{5.31}$$

*with*

$$Q^*_1(x, u) = \tilde{\rho}_{\mathcal{F}_n}(x, u), \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U}. \tag{5.32}$$

Using the sequence of approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$, one can infer an open-loop sequence of actions

$$\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = (\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \mathcal{U}^T \tag{5.33}$$

which is an exact solution of the approximated optimal control problem, i.e. which is such that

$$\tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) = \tilde{J}^*_{\mathcal{F}_n}(x_0) \tag{5.34}$$

as follows:

$$\tilde{u}^*_{\mathcal{F}_n,0}(x_0) \quad \in \quad \arg\max_{u' \in \mathcal{U}} \tilde{Q}^*_T(\tilde{x}^*_0, u')\ , \tag{5.35}$$

and, $\forall t \in \{0, \ldots, T-2\}$ ,

$$\tilde{u}^*_{\mathcal{F}_n,t+1}(x_0) \quad \in \quad \arg\max_{u' \in \mathcal{U}} \tilde{Q}^*_{T-(t+1)}\left(\tilde{f}_{\mathcal{F}_n}\left(\tilde{x}^*_t, \tilde{u}^*_{\mathcal{F}_n,t}(x_0)\right), u'\right) \tag{5.36}$$

where

$$\tilde{x}^*_{t+1} = \tilde{f}_{\mathcal{F}_n}(\tilde{x}^*_t, \tilde{u}^*_{\mathcal{F}_n,t}(x_0)), \forall t \in \{0, \ldots, T-1\}. \tag{5.37}$$

and $\tilde{x}^*_0 = x_0$.

All the approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$ are piecewise constant over each Voronoi cell, a property that can be exploited for computing them easily as it is shown in Figure 3. The VRL algorithm has linear complexity with respect to the cardinality $n$ of the sample of system transitions $\mathcal{F}_n$, the optimization horizon $T$ and the cardinality $m$ of the action space $\mathcal{U}$. Furthermore, the VRL algorithm has consistency properties in Lipschitz continuous environments, for which the open-loop sequence of actions computed by the VRL algorithm converges towards an optimal sequence of actions when the sparsity of the sample of system transitions converges towards zero [9].

### 5.4.2 Choice of the predictive model $PM$

Model learning–type RL uses a predictive model of the environment. Our predictive model $PM$ is thus given by the approximated system dynamics $\tilde{f}_{\mathcal{F}_n}$ and reward function $\tilde{\rho}_{\mathcal{F}_n}$ computed by the VRL algorithm. Given a sample of transitions $\mathcal{F}_n$ and a

**Algorithm 3** The Voronoi Reinforcement Learning (VRL) algorithm. $Q_{T-t,l}$ is the value taken by the function $\tilde{Q}^*_{T-t}$ in the Voronoi cell $V^l$.

```
Inputs: an initial state $x_0 \in \mathcal{X}$, a sample of transitions $\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^{n}$ ;
Output: a sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$ and $\tilde{J}^*_{\mathcal{F}_n}(x_0)$ ;
Initialization:
Create a $n \times m$ matrix $V$ such that $V(i,j)$ contains the index of the Voronoi cell (VC) where $\left(\tilde{f}_{\mathcal{F}_n}(x^i, u^i), a^j\right)$ lies ;
for $i = 1$ to $n$ do
  $Q_{1,i} \leftarrow r^i$ ;
end for
Algorithm:
for $t = T-2$ to $0$ do
  for $i = 1$ to $n$ do
    $l \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max} \left\{Q_{T-t-1,V(i,l')}\right\}$ ;
    $Q_{T-t,i} \leftarrow r^i + Q_{T-t-1,V(i,l)}$ ;
  end for
end for
$l \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max}\ Q_{T,i'}$ where $i'$ denotes the index of the VC where $(x_0, a^{l'})$ lies ;
$l_0^* \leftarrow$ index of the VC where $(x_0, a^l)$ lies ;
$\tilde{J}^*_{\mathcal{F}_n}(x_0) \leftarrow Q_{T,l_0^*}$ ;
$i \leftarrow l_0^*$ ;
$\tilde{u}^*_{\mathcal{F}_n,0}(x_0) \leftarrow u^{l_0^*}$ ;
for $t = 0$ to $T-2$ do
  $l_{t+1}^* \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max} \left\{Q_{T-t-1,V(i,l')}\right\}$ ;
  $\tilde{u}^*_{\mathcal{F}_n,t+1}(x_0) \leftarrow a^{l_{t+1}^*}$ ;
  $i \leftarrow V(i, l_{t+1}^*)$ ;
end for
Return: $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = (\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ and $\tilde{J}^*_{\mathcal{F}_n}(x_0)$.
```

state-action point $(x, u) \in \mathcal{X} \times \mathcal{U}$, the $PM$ algorithm computes a predicted system transition

$$(x, u, \hat{r}_{\mathcal{F}_n}(x, u), \hat{y}_{\mathcal{F}_n}(x, u)) = PM(\mathcal{F}_n, x, u) \tag{5.38}$$

such that:

$$\begin{aligned} \forall (x, u) \in \mathcal{X} \times \mathcal{U} : \qquad & \hat{r}_{\mathcal{F}_n}(x, u) = \tilde{\rho}_{\mathcal{F}_n}(x, u) \ , && (5.39) \\ & \hat{y}_{\mathcal{F}_n}(x, u) = \tilde{f}_{\mathcal{F}_n}(x, u) \ . && (5.40) \end{aligned}$$

## 5.5 Experimental simulation results with the car-on-the-hill problem

We propose in this section to illustrate the sampling strategy proposed in the previous sections on the car-on-the-hill problem [7] which has been vastly used as benchmark for validating RL algorithms. First we describe the benchmark. Afterwards we detail the experimental protocol and finally, we present and discuss our simulation results.

### 5.5.1 The car-on-the-hill benchmark

In the car-on-the-hill benchmark, a point mass - which represents a car - has to be driven past the top of a hill by applying a horizontal force. For some initial states, the maximum available force is not sufficient to drive the car directly up the right hill. Instead, the car has to first be driven up the opposite (left) slope in order to gather energy prior to accelerating towards the goal. An illustration of the car-on-the-hill benchmark is given below in Figure 5.1.

The continuous-time dynamics of the car is given by

$$\ddot{z} = \frac{1}{1 + \left(\frac{dH(z)}{dz}\right)^2} \left( \frac{u}{m_c} - g\frac{dH(z)}{dz} - \dot{z}^2 \frac{dH(z)}{dz} \frac{d^2H(z)}{dz^2} \right) \tag{5.41}$$

where $z \in [-1, 1]$ is the horizontal position of the car (expressed in $m$), $\dot{z} \in [-3, 3]$ is the velocity of the car (given in $m/s$), $u \in \{-4, 4\}$ is the horizontal force applied to the car (expressed in $N$), $g = 9.81 m/s^2$ is the gravitational acceleration and $H$ denotes the slope of the hill:

$$H(z) = \begin{cases} z^2 + z & \text{if} \quad z < 0 \ , \\ \frac{z}{\sqrt{1+5z^2}} & \text{if} \quad z \geq 0 \ . \end{cases} \tag{5.42}$$

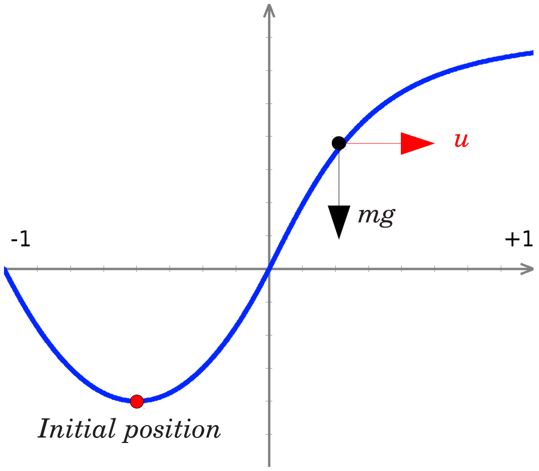

Figure 5.1: Illustration of the car-on-the-hill benchmark.

We assume that the car has a mass $m_c = 1kg$. The discrete time step is set to $T_s = 0.1s$ and the discrete time dynamics $f$ is obtained by integrating the continuous-time dynamics between subsequent time steps. The action space $\mathcal{U}$ is made of the two elements: $-4$ and $4$. Whenever the position $z$ or velocity $\dot{z}$ exceeds the bounds, the car reaches an absorbing state in which it stays whatever the control actions taken. If $z_{t+1} < -1$ or if $|\dot{z}_{t+1}| > 3$, then the car reaches a "loosing" absorbing state $s_{-1}$ and gets a $-1$ reward at each time-step till the end of the trial. If $z_{t+1} \geq 1$ and $|\dot{z}_{t+1}| \leq 3$, then the car reaches a "winning" absorbing state $s_1$, and gets a $+1$ reward at each time-step till the end of the trial. We have assumed in our simulation that we know that $s_{-1}$ and $s_{+1}$ are two absorbing states. The state space of the system is equal to

$$\mathcal{X} = [-1, 1] \times [-3, 3] \cup \{s_1, s_{-1}\} \ . \tag{5.43}$$

The goal is to find a sequence of actions that leads to the highest sum of rewards over an optimization horizon $T = 20$ when the car starts in $x_0 = [-0.5, 0]$. Such a sequence of actions also drives the car in a minimum amount of time to the top of the hill.

The VRL algorithm was described in Section 5.4.1 for problems with no absorbing states. It can easily be amended to handle the absorbing states of the car-on-the-hill problem. This can be done for example by modifying the set of system transitions used as input of the algorithm by adding $m \times n_{abs}$ "fake system transitions", where $n_{abs}$ is the number of absorbing states of the problem. With respect to the car-on-the-hill problem, this results in the addition of the following four fake system transitions into any sample of transitions

$$\big\{(s_1, 4, 1, s_1), (s_1, -4, 1, s_1), (s_{-1}, 4, -1, s_{-1}), (s_{-1}, -4, -1, s_{-1})\big\} \tag{5.44}$$

The definition of the Voronoi cells remains the same as in Equations (5.26), (5.27) and (5.28) if $x^l$ is not an absorbing state. Otherwise, the norm $\|.\|_{\mathcal{X}}$ can be (abusively) "extended" to absorbing states as follows:

$$\|x - x^l\|_{\mathcal{X}} = \begin{cases} 0 & \text{if } x = x^l \ , \\ +\infty & \text{if } x \neq x^l \ . \end{cases} \tag{5.45}$$

### 5.5.2 Experimental protocol

We propose to compare the performance of our sampling strategy described in Section 5.3 with the performance of a uniform sampling strategy. To this end, we run $q = 50$ times our sampling strategy, where each run $k = 1 \dots q$ is initialized with a sample $\mathcal{F}_m^k$ that contains $m = 2$ system transitions (one transition for each action of the action space) as follows:

$$\begin{aligned} \forall k \in \{1, \dots, q\}, \mathcal{F}_m^k = \{&(x_0, -4, \rho(x_0, -4), f(x_0, -4)) \ , \\ &(x_0, +4, \rho(x_0, +4), f(x_0, +4))\} \ . \end{aligned} \tag{5.46}$$

We sequentially run our sampling strategy on each sample of transitions $\mathcal{F}_m^k \quad k = 1 \dots q$ until it gathers $N_{\max} = 1000$ system transitions. These runs lead to $q$ sequences of $(N_{\max} - m + 1)$ samples of system transitions:

$$\begin{array}{ccc} \mathcal{F}_m^1, \mathcal{F}_{m+1}^1, & \dots & , \mathcal{F}_{N_{\max}}^1 \\ & \dots & \\ \mathcal{F}_m^q, \mathcal{F}_{m+1}^q, & \dots & , \mathcal{F}_{N_{\max}}^q . \end{array} \tag{5.47}$$

We also generate $q$ sequences of $(N_{\max} - m + 1)$ samples of system transitions

$$\begin{array}{ccc} \mathcal{G}_m^1, \mathcal{G}_{m+1}^1, & \dots & , \mathcal{G}_{N_{\max}}^1 \\ & \dots & \\ \mathcal{G}_m^q, \mathcal{G}_{m+1}^q, & \dots & , \mathcal{G}_{N_{\max}}^q \end{array} \tag{5.48}$$

where, for all $k = 1 \dots q$, for all $n = m \dots N_{\max} - 1$, each sample $\mathcal{G}_{n+1}^k$ is obtained by adding one system transition $(x, u, \rho(x, u), f(x, u))$ to $\mathcal{G}_n^k$ for which $(x, u)$ is drawn according to $p_{\mathcal{X} \times \mathcal{U}}(\cdot)$. The sequence of parameters $L_n$ used for these experiments is defined as follows:

$$\forall n \in \{m, \dots, N_{\max}\}, L_n = mn \ . \tag{5.49}$$

The probability distribution $p_{\mathcal{X} \times \mathcal{U}}(\cdot)$ is such that the probability of drawing a state-action point $(x, u)$ with $x = s_1$ or $x = s_{-1}$ is zero, and uniform elsewhere.

### 5.5.3 Results and discussions

**Performances of the control policies inferred from the samples of $N_{\max}$ transitions**

We first compute the returns of the $2q$ control policies respectively inferred by the VRL algorithm from the samples of $N_{\max}$ system transitions $\mathcal{F}_{N_{\max}}^k$ and with the randomly generated samples $\mathcal{G}_{N_{\max}}^k \quad k = 1 \dots q$. The obtained results are reported on Figure 5.2 in terms of the distribution of returns of the inferred control policy over the 50 runs.

We observe that the VRL algorithm manages to compute for $28\%$ of the runs a control policy for which the return is equal to 2, whereas not even a single control policy with a return greater than 0 was inferred from the $q$ samples $\mathcal{G}_{N_{\max}}^k \quad k = 1 \dots q$ generated using the uniform sampling strategy.

Notice that in order to obtain results of similar quality to those of our iterative sampling strategy, we found that one would need to use about $10,000$ randomly generated system transitions.

**Average performance and distribution of the returns of the inferred control policies**

For a given cardinality $n$ ($m \leq n \leq N_{\max}$), we compute the average actual performance $\mathcal{M}(n)$ of the $q$ sequences of actions $\tilde{\mathbf{u}}_{\mathcal{F}_\mathbf{n}^\mathbf{k}}^*(x_0) \quad k = 1 \dots q$ computed by the VRL algorithm from the sample of system transitions $\mathcal{F}_n^k \quad k = 1 \dots q$:

$$\mathcal{M}(n) = \frac{1}{q} \sum_{k=1}^{q} J^{\tilde{\mathbf{u}}_{\mathcal{F}_\mathbf{n}^\mathbf{k}}^*(\mathbf{x_0})}(x_0) \ . \tag{5.50}$$

The average performance $\mathcal{M}(n) \quad n = m \dots N_{\max}$ is compared with the average performance $\mathcal{M}_{unif}(n)$ of the $q$ sequences of actions $\tilde{\mathbf{u}}_{\mathcal{G}_\mathbf{n}^\mathbf{k}}^*(x_0) \quad k = 1 \dots q$ inferred

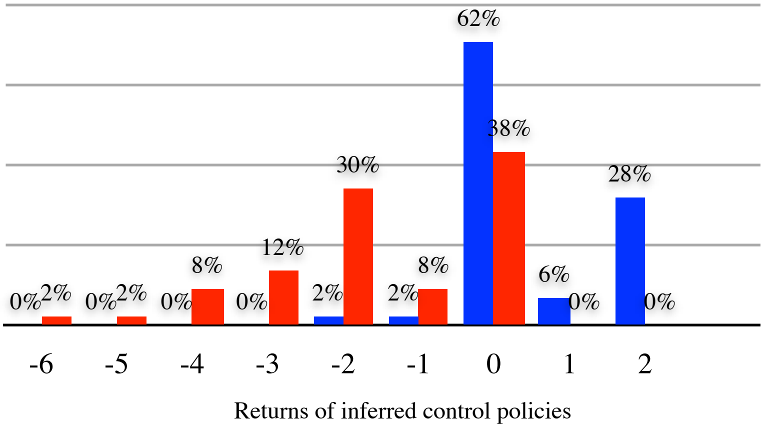

Figure 5.2: Distribution of the returns of the control policies inferred from $\mathcal{F}^k_{N_{\max}} \quad k = 1 \dots q$ (blue histogram, on the left) and $\mathcal{G}^k_{N_{\max}} \quad k = 1 \dots q$ (red histogram, on the right).

by the VRL algorithm from samples of system transitions $\mathcal{G}^k_n \quad k = 1 \dots q$ gathered according to a uniform sampling strategy:

$$\mathcal{M}_{unif}(n) = \frac{1}{q} \sum_{k=1}^{q} J^{\tilde{\mathbf{u}}^*_{\mathcal{G}^\mathbf{k}_\mathbf{n}}(\mathbf{x_0})}(x_0) \ . \tag{5.51}$$

The values of $\mathcal{M}(n)$ and $\mathcal{M}_{unif}(n)$ for $n = m \dots N_{\max}$ are reported on Figure 5.3. We also report the distribution of the return of the policies $\tilde{\mathbf{u}}^*_{\mathcal{F}^\mathbf{k}_\mathbf{n}}(x_0) \quad k = 1 \dots q, n = m \dots N_{\max}$ (resp. $\tilde{\mathbf{u}}^*_{\mathcal{G}^\mathbf{k}_\mathbf{n}}(x_0) \quad k = 1 \dots q, n = m \dots N_{\max}$) on Figure 5.4 (resp. Figure 5.5). We observe that, with our sampling strategy, control policies leading to a return of 2 can be inferred from samples of less than 200 system transitions. We also notice that no policy leading to a return of 2 could be inferred from any of the uniformly sampled system transitions $\mathcal{G}^k_n \quad k = 1 \dots q$.

### Representation of $\mathcal{F}^1_{N_{\max}}$ and $\mathcal{G}^1_{N_{\max}}$

We finally plot the system transitions gathered in the sample $\mathcal{F}^1_{N_{\max}}$ (resp. $\mathcal{G}^1_{N_{\max}}$) on Figure 5.6 (resp. on Figure 5.7). Each system transition $(x^l, u^l, r^l, y^l)$ is represented

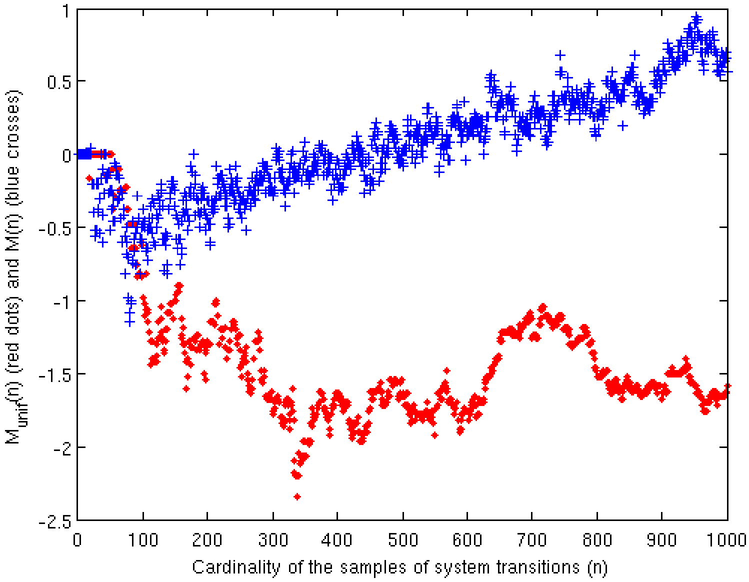

Figure 5.3: Evolution of the average performance of our sampling strategy $\mathcal{M}(n)$ (blue crosses) compared with the average performance of the uniform sampling strategy $M_{unif}(n)$ (red dots).

by a colored symbol located at $x^l = [z, \dot{z}]$. A '+' sign indicates that $u^l = +4$, whereas a '•' sign indicates that $u^l = -4$. The symbol is colored in blue if $r^l = 0$. Larger symbols colored in black (green) are used if $r^l = -1$ ($r^l = 1$). The red curve represents the trajectory of the car when driven according to the inferred policy $\tilde{\mathbf{u}}^*_{\mathcal{F}^1_\mathbf{n}}(x_0)$ (resp. $\tilde{\mathbf{u}}^*_{\mathcal{G}^1_\mathbf{n}}(x_0)$). One can observe on Figure 5.6 that our sampling strategy tends to sample state-action points that are located in the neighborhood of high-performance trajectories.

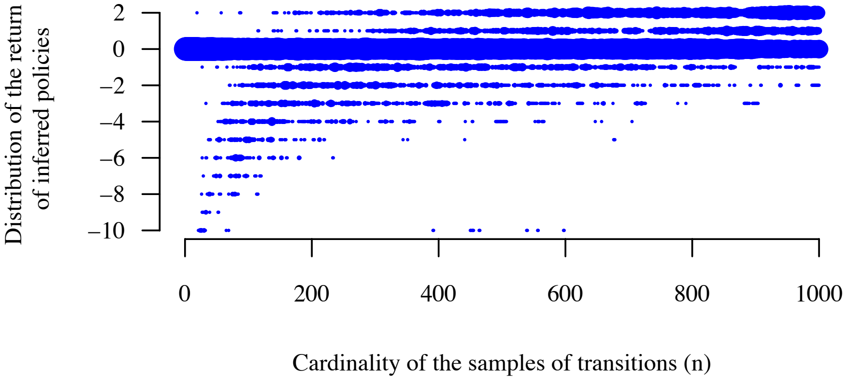

Figure 5.4: Distribution of the return of the control policies $\tilde{\mathbf{u}}^*_{\mathcal{F}^{\mathbf{k}}_{\mathbf{n}}}(x_0)$ k=1 ...q, $n = m \dots N_{\max}$. For each value of $n$, the area covered by a bullet to which corresponds a return $r = -10 \dots 2$ is proportional to the number of control policies from $\left\{\tilde{\mathbf{u}}^*_{\mathcal{F}^{\mathbf{k}}_{\mathbf{n}}}(x_0)\right\}_{k=1}^{q}$ whose return is equal to $r$.

## 5.6 Related work

The problem of sampling parsimoniously the state-action space of an optimal control problem for identifying good policies has already been addressed by several authors. The approach detailed in [6] is probably the closest to ours. In this chapter, the authors propose a sequential sampling strategy which also favours sampling locations that are predicted to have a high-influence on the policy that will be inferred. While we focus in this chapter on deterministic problems with continuous state spaces, their approach is particularized to stationary stochastic problems with finite state spaces.

In [11] (reported in Chapter 4), another sequential sampling strategy is proposed. It works by computing bounds on the return of control policies and selects as sampling area the one which is expected to lead to the highest increase of the bounds' tightness. The approach requires the system dynamics and the reward function to be Lipschitz continuous and, relies at its heart on the resolution of a complex optimization problem.

Most of the works in the field of RL related to the generation of informative samples have focused on the problem of controlling a system so as to generate samples that can be used to increase the performance of the control policy while at the same

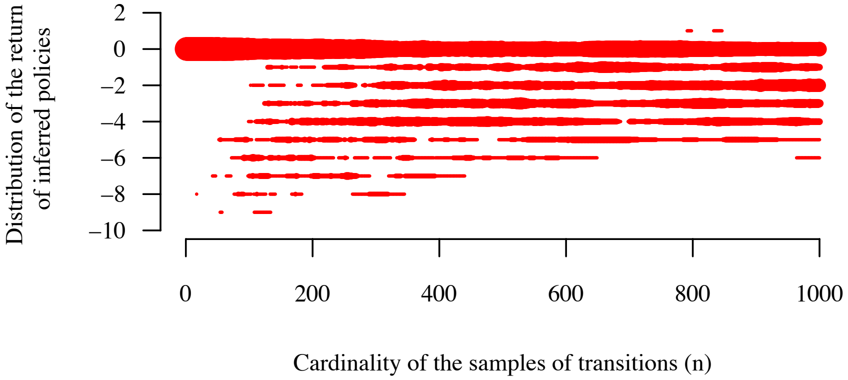

Figure 5.5: Distribution of the return of the control policies $\tilde{\mathbf{u}}^*_{\mathcal{G}^{\mathrm{k}}_{\mathrm{n}}}(x_0)$ k=1 . . . q, $n = m \dots N_{\max}$. For each value of $n$, the area covered by a bullet to which corresponds a return $r = -10 \dots 2$ is proportional to the number of control policies from $\left\{\tilde{\mathbf{u}}^*_{\mathcal{G}^{\mathrm{k}}_{\mathrm{n}}}(x_0)\right\}_{k=1}^{q}$ whose return is equal to $r$.

time generating high-rewards. One common approach for addressing this “exploration-exploitation” dilemma ([1, 5]) is to use a so-called $\epsilon$-Greedy policy which is a policy that deviates with a certain probability from the estimate of the optimal one ([22, 14, 21]). The problem has been recently well-studied for stochastic Markov Decision Processes having one single state ([3]).

There is a considerable body of work in the field of adaptive discretization techniques in dynamic programming which is also related to our approach. In these works, the state-action space is iteratively sampled so as to lead rapidly to an optimal policy (see e.g., [15]). If at the inner loop of our approach, exact samples rather than predicted samples were used, it could certainly be assimilated to this body of work. The amount of computation required by our approach to identify at every iteration a new sample would however not make it necessarily a good adaptive discretization technique. Indeed, the efficiency of an adaptive discretization technique does not depend solely only on the number of samples it uses to identify a good policy, but well on its overall computational complexity.

Finally, it is worth mentioning that the problem of identifying a concise set of samples from which a good policy can be inferred has also been addressed in other

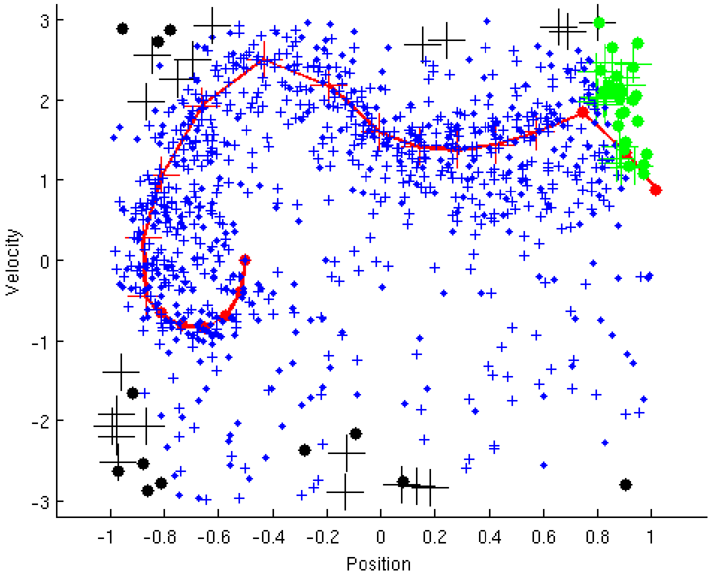

Figure 5.6: Representation of the sample of system transitions $\mathcal{F}^1_{N_{\max}}$ (obtained through inferred policy variations-based sampling strategy).

contexts than the one considered in this chapter. For example, [7] proposes a strategy for extracting from a given sample of system transitions, a much smaller subset that can still lead to a good policy. The strategy relies on the computation of errors in a Bellman equation and showed good results on problems having a smooth environment. In [19], the authors focus on the identification of a small sample of transitions that can lead to a good policy when combined with a BMRL algorithm without assuming any constraints on the number of samples that can be generated. The simulation results given in this chapter show that for the car-on-the-hill benchmark, less than twenty well chosen samples can lead to an optimal policy. However, for identifying these samples, the state-action space had to be sampled a very large number of times (about hundreds of thousands of times).

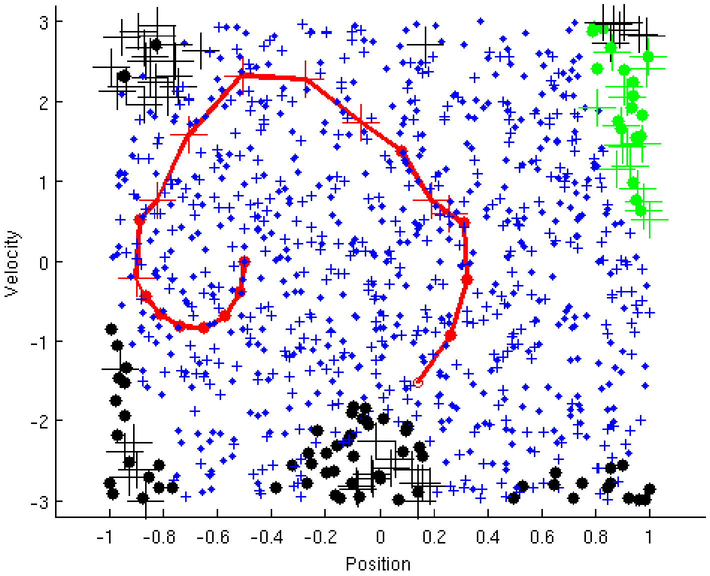

Figure 5.7: Representation of the sample of system transitions $\mathcal{G}^{1}_{N_{\max}}$ (obtained through uniform sampling strategy).

## 5.7 Conclusions

We have proposed a sequential strategy for sampling informative collections of system transitions for solving deterministic optimal control problems in continuous state spaces. This sampling strategy uses the ability of predicting system transitions, in order to identify experiments whose outcome would be likely to falsify the current hypothesis about the solution of the optimal control problem. Algorithms have been fully specified for the case of finite horizon deterministic optimal control problems with finite action spaces, by using nearest-neighbor approximations of the optimal control problem both in the RL algorithm and for predicting the outcome of experiments in terms of hypothetical system transitions.

The simulations were carried out on the car-on-the-hill problem and the results were promising. In particular, our sampling strategy was found to be much more efficient than a uniform sampling one. These results motivate further study of the algorithms proposed in this chapter. In particular, it would be interesting to establish under which conditions policy falsification caused by new samples also corresponds to actual policy improvements and what may be the influence of the prediction errors done when generating the “predicted system transitions” on the “predicted policy changes”. This should be very helpful for analytically investigating the convergence speed of the proposed sampling strategy towards a sample of system transitions from which optimal or near-optimal policies could be inferred.

Finally, while an instance of this policy falsification concept for generating new experiments has been fully specified and validated for deterministic problems with discrete action spaces, we believe that it would also be interesting to investigate ways to exploit it successfully in other settings.

# Bibliography

[1] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Reserch*, 3:397 – 422, 2003.

[2] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[3] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. Online optimization in X-armed bandits. In *Advances in Neural Information Processing Systems 21*, pages 201–208. MIT Press, 2009.

[4] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming using Function Approximators*. Taylor & Francis CRC Press, 2010.

[5] J.D. Cohen, S.M. McClure, and A.J. Yu. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B 29*, 362(1481):933–942, 2007.

[6] A. Ephsteyn, A. Vogel, and G. DeJong. Active reinforcement learning. In *Proceedings of the 25th international conference on Machine learning (ICML 2008)*, volume 307, 2008.

[7] D. Ernst. Selecting concise sets of samples for a reinforcement learning agent. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*, Singapore, 2005.

[8] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[9] R. Fonteneau and D. Ernst. Voronoi model learning for batch mode reinforcement learning. Technical report, University of Liège, 2010.

[10] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series: Communications in Computer and Information Science (CCIS)*, volume 129, pages 61–77. Springer, Heidelberg, 2011.

[11] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Generating informative trajectories by using bounds on the return of control policies. In *Proceedings of the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*, 2010.

[12] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Active exploration by searching for experiments falsifying an already induced policy. *To be published in the Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2011), Paris, France*, 2011.

[13] J.E. Ingersoll. *Theory of Financial Decision Making*. Rowman and Littlefield Publishers, Inc., 1987.

[14] L.P. Kaelbling. *Learning in Embedded Systems*. MIT Press, 1993.

[15] R. Munos and A. Moore. Variable resolution discretization in optimal control. *Machine Learning*, 49:291–323, 2002.

[16] S.A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, 65(2):331–366, 2003.

[17] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[18] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[19] E. Rachelson, F. Schnitzler, L. Wehenkel, and D. Ernst. Optimal sample selection for batch-mode reinforcement learning. In *3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, Roma, Italy, 2011.

[20] M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML 2005)*, pages 317–328, Porto, Portugal, 2005.

[21] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[22] S. Thrun. The role of exploration in learning control. In D. White and D. Sofge, editors, *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold, 1992.

# Chapter 6

# Model-free Monte Carlo–like policy evaluation

*We propose an algorithm for estimating the finite-horizon expected return of a closed loop control policy from an a priori given (off-policy) sample of one-step transitions. It averages cumulated rewards along a set of "broken trajectories" made of one-step transitions selected from the sample on the basis of the control policy. Under some Lipschitz continuity assumptions on the system dynamics, reward function and control policy, we provide bounds on the bias and variance of the estimator that depend only on the Lipschitz constants, on the number of broken trajectories used in the estimator, and on the sparsity of the sample of one-step transitions.*

The work presented in this chapter has been published in the *Proceedings of the International Conference on Artificial Intelligence and Statistics* (AISTATS 20010) [7]. This work has also been presented at the *Conférence francophone sur l'Apprentissage Automatique* (CAp 2010) [8] where it received the "Best Student Paper Award".

In this chapter, we consider:

- a stochastic framework,
- a continuous state-action space.

## 6.1 Introduction

Discrete-time stochastic optimal control problems arise in many fields such as finance, medicine, engineering as well as artificial intelligence. Many techniques for solving such problems use an oracle that evaluates the performance of any given policy in order to navigate rapidly in the space of candidate optimal policies to a (near-)optimal one.

When the considered system is accessible to experimentation at low cost, such an oracle can be based on a Monte Carlo (MC) approach. With such an approach, several "on-policy" trajectories are generated by collecting information from the system when controlled by the given policy, and the cumulated rewards observed along these trajectories are averaged to get an unbiased estimate of the performance of that policy. However if obtaining trajectories under a given policy is very costly, time consuming or otherwise difficult, e.g. in medicine or in safety critical problems, the above approach is not feasible.

In this chapter, we propose a policy evaluation oracle in a *model-free* setting. In our setting, the only information available on the optimal control problem is contained in a sample of one-step transitions of the system, that have been gathered by some arbitrary experimental protocol, i.e. independently of the policy that has to be evaluated.

Our estimator is inspired by the MC approach. Similarly to the MC estimator, it evaluates the performance of a policy by averaging the sums of rewards collected along several trajectories. However, rather than "real" on-policy trajectories of the system generated by fresh experiments, it uses a set of "broken trajectories" that are rebuilt from the given sample and from the policy that is being evaluated. Under some Lipschitz continuity assumptions on the system dynamics, reward function and policy, we provide bounds on the bias and variance of our model-free policy evaluator, and show that it behaves like the standard MC estimator when the sample sparsity decreases towards zero.

The core of the chapter is organized as follows. Section 6.2 discusses related work, Section 6.3 formalizes the problem, and Section 6.4 states our algorithm and its theoretical properties. Section 6.5 provides some simulation results. Proofs of our main theorems are sketched in the Appendix.

## 6.2 Related Work

Model-free policy evaluation has been well studied, in particular in reinforcement learning. This field has mostly focused on the estimation of the *value function* that maps initial states into returns of the policy from these states. Temporal Difference methods ([13, 16, 12, 2]) are techniques for estimating value functions from the sole knowledge of one-step transitions of the system, and their underlying theory has been well investigated, e.g., ([4, 15]). In large state-spaces, these approaches have to be combined with function approximators to compactly represent the value function ([14]). More recently, batch mode approximate value iteration algorithms have been successful in using function approximators to estimate value functions in a model-free setting ([10, 6, 11]), and several papers have analyzed some of their theoretical properties ([1, 9]).

The Achilles' heel of all these techniques is their strong dependence on the choice of a suitable function approximator, which is not straightforward ([3]). Contrary to these techniques, the estimator proposed in this chapter does not use function approximators. As mentioned above, it is an extension of the standard MC estimator to a model-free setting, and in this, it is related to current work seeking to build computationally efficient model-based Monte Carlo estimators, e.g., ([5]).

## 6.3 Problem statement

We consider a discrete-time system whose behavior over $T$ stages is characterized by a time-invariant dynamics

$$x_{t+1} = f(x_t, u_t, w_t) \quad t = 0, 1, \dots, T-1, \tag{6.1}$$

where $x_t$ belongs to a normed vector space $\mathcal{X}$ of states, and $u_t$ belongs to a normed vector space $\mathcal{U}$ of control actions. An instantaneous reward

$$r_t = \rho(x_t, u_t, w_t) \in \mathbb{R} \tag{6.2}$$

is associated with the transition from $t$ to $t+1$. The stochasticity of the control problem is induced by the unobservable random process $w_t \in \mathcal{W}$, which we suppose to be drawn i.i.d. according to a probability distribution $p_{\mathcal{W}}(.)$, $\forall t = 0, \dots, T-1$. In the following, we signal this by $w_t \sim p_{\mathcal{W}}(.)$ and, as induced by the notation, we assume that $p_{\mathcal{W}}(.)$ depends neither on $(x_t, u_t)$ nor on $t \in \{0, \dots, T-1\}$ . $T \in \mathbb{N}_0$ is referred to as the optimization horizon of the control problem. Let

$$h : \{0, \dots, T-1\} \times \mathcal{X} \to \mathcal{U} \tag{6.3}$$

be a deterministic closed-loop time-varying control policy that maps the time $t$ and the current state $x_t$ into the action $u_t = h(t, x_t)$, and let $J^h(x_0)$ denote the expected return of this policy $h$, defined as follows :

**Definition 6.3.1 (Expected $T-$stage return of the policy $h$)**
$\forall x_0 \in \mathcal{X}$,

$$J^h(x_0) = \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[R^h(x_0)\right] \ , \tag{6.4}$$

*where*

$$R^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t) \tag{6.5}$$

*and*

$$x_{t+1} = f(x_t, h(t, x_t), w_t). \tag{6.6}$$

A realization of the random variable $R^h(x_0)$ corresponds to the cumulated reward of $h$ when used to control the system from the initial condition $x_0$ over $T$ stages while disturbed by the random process $w_t \sim p_{\mathcal{W}}(.)$. We suppose that $R^h(x_0)$ has a finite variance:

**Assumption 6.3.2 (Finite variance of $R^h(x_0)$)**
$\forall x_0 \in \mathcal{X}$,

$$\sigma^2_{R^h}(x_0) = \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}{Var} \left[R^h(x_0)\right] < \infty. \tag{6.7}$$

In our setting, $f$, $\rho$ and $p_{\mathcal{W}}(.)$ are fixed but *unknown* (and hence inaccessible to simulation). The only information available on the control problem is gathered in a given sample of $n$ one-step transitions

$$\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^{n} , \tag{6.8}$$

where:

- The first two elements ($x^l$ and $u^l$) of every one-step transition are chosen in an arbitrary way,
- The pairs $(r^l, y^l)$ are consistently determined by $(\rho(x^l, u^l, .), f(x^l, u^l, .))$, drawn according to $p_{\mathcal{W}}(.)$.

We want to estimate from such a sample $\mathcal{F}_n$, the expected return $J^h(x_0)$ of the given policy $h$ for a given initial state $x_0$.

## 6.4 A model-free Monte Carlo–like estimator of $J^h(x_0)$

We first remind the classical model-based MC estimator and its bias and variance in Section 6.4.1. In Section 6.4.2 we explain our estimator which mimics the MC estimator in a model-free setting, and in Section 6.4.3 we provide a theoretical analysis of the bias and variance of this estimator.

### 6.4.1 Model-based MC estimator

The MC estimator works in a model-based setting (i.e., in a setting where $f$, $\rho$ and $p_{\mathcal{W}}(.)$ are known). It estimates $J^h(x_0)$ by averaging the returns of several (say $p \in \mathbb{N}_0$) trajectories of the system which have been generated by simulating the system from $x_0$ using the policy $h$. More formally, the MC estimator of the expected return of the policy $h$ when starting from the initial state $x_0$ writes as follows:

**Definition 6.4.1 (Model-based Monte Carlo estimator)**
$\forall p \in \mathbb{N}_0, \forall x_0 \in \mathcal{X}$,

$$\mathbb{M}_p^h(x_0) = \frac{1}{p} \sum_{i=1}^{p} \sum_{t=0}^{T-1} \rho\left(x_t^i, h\left(t, x_t^i\right), w_t^i\right) \tag{6.9}$$

*with*

$$\begin{aligned} \forall t \in \{0, \dots, T-1\}, \forall i \in \{1, \dots, p\} : w_t^i &\sim p_{\mathcal{W}}(.) \,, & (6.10)\\ x_0^i &= x_0 \,, & (6.11)\\ x_{t+1}^i &= f\left(x_t^i, h\left(t, x_t^i\right), w_t^i\right) . & (6.12) \end{aligned}$$

It is well known that the bias and variance of the MC estimator are:

**Proposition 6.4.2 (Bias of the MC estimator)**
$\forall p \in \mathbb{N}_0, \forall x_0 \in \mathcal{X}$,

$$\underset{w_t^i \sim p_{\mathcal{W}}(.), i=1\dots p, t=0\dots T-1}{\mathbb{E}} \left[\mathbb{M}_p^h(x_0) - J^h(x_0)\right] = 0 \,. \tag{6.13}$$

**Proposition 6.4.3 (Variance of the MC estimator)**
$\forall p \in \mathbb{N}_0, \forall x_0 \in \mathcal{X}$,

$$\underset{w_t^i \sim p_{\mathcal{W}}(.), i=1\dots p, t=0\dots T-1}{Var} \left[\mathbb{M}_p^h(x_0)\right] = \frac{\sigma_{R^h}^2(x_0)}{p} \,. \tag{6.14}$$

### 6.4.2 Model-free MC estimator

From a sample $\mathcal{F}_n$, our model-free MC (MFMC) estimator works by selecting $p$ sequences of transitions of length $T$ from this sample that we call "broken trajectories". These broken trajectories will then serve as proxies of $p$ "actual" trajectories that could be obtained by simulating the policy $h$ on the given control problem. Our estimator averages the cumulated returns over these broken trajectories to compute its estimate of $J^h(x_0)$. The main idea behind our method consists of selecting the broken trajectories so as to minimize the discrepancy of these trajectories with a classical MC sample that could be obtained by simulating the system with policy $h$.

To build a sample of $p$ substitute broken trajectories of length $T$ starting from $x_0$ and similar to trajectories that would be induced by a policy $h$, our algorithm uses each one-step transition in $\mathcal{F}_n$ at most once; we thus assume that $pT \leq n$. The $p$ broken trajectories of $T$ one-step transitions are created sequentially. Every broken trajectory is grown in length by selecting, among the sample of not yet used one-step transitions, a transition whose first two elements minimize the distance $-$ using a distance metric $\Delta$ in $\mathcal{X} \times \mathcal{U}$ $-$ with the couple formed by the last element of the previously selected transition and the action induced by $h$ at the end of this previous transition.

**Algorithm 4** MFMC algorithm to generate a set of size $p$ of $T-$length broken trajectories from a sample of $n$ one-step transitions.

```
**Input:** $\mathcal{F}_n, h(.,.), x_0, \Delta(.,.), T, p$
Let $\mathcal{G}$ denote the current set of not yet used one-step transitions in $\mathcal{F}_n$; Initially,
$\mathcal{G} \leftarrow \mathcal{F}_n$;
**for** $i = 1$ to $p$ (extract a broken trajectory) **do**
  $t \leftarrow 0$;
  $x_t^i \leftarrow x_0$;
  **while** $t < T$ do **do**
    $u_t^i \leftarrow h\left(t, x_t^i\right)$;
    $\mathcal{H} \leftarrow \underset{(x,u,r,y) \in \mathcal{G}}{\arg\min} \left(\Delta\left((x,u), \left(x_t^i, u_t^i\right)\right)\right)$;
    $l_t^i \leftarrow$ lowest index in $\mathcal{F}_n$ of the transitions that belong to $\mathcal{H}$;
    $t \leftarrow t + 1$;
    $x_t^i \leftarrow y^{l_t^i}$;
    $\mathcal{G} \leftarrow \mathcal{G} \setminus \left\{\left(x^{l_t^i}, u^{l_t^i}, r^{l_t^i}, y^{l_t^i}\right)\right\}$;
  **end while**
**end for**
**Return** the set of indices $\left\{l_t^i\right\}_{i=1,t=0}^{i=p,t=T-1}$.
```

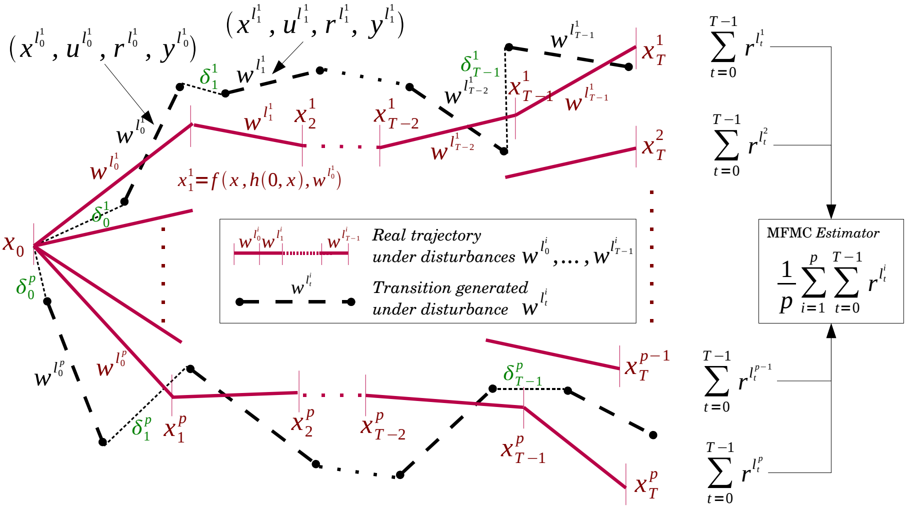
$(x^{l_0^1}, u^{l_0^1}, r^{l_0^1}, y^{l_0^1})$
$(x^{l_1^1}, u^{l_1^1}, r^{l_1^1}, y^{l_1^1})$
$x_0$
$x_1^1=f(x,h(0,x),w^{l_0^1})$
$x_2^1$
$x_{T-2}^1$
$x_{T-1}^1$
$x_T^1$
$x_T^2$
$x_T^{p-1}$
$x_1^p$
$x_2^p$
$x_{T-2}^p$
$x_{T-1}^p$
$x_T^p$
$\delta_0^1$
$\delta_1^1$
$\delta_{T-1}^1$
$\delta_0^p$
$\delta_1^p$
$\delta_{T-1}^p$
Real trajectory under disturbances $w^{l_0^i},\dots,w^{l_{T-1}^i}$
Transition generated under disturbance $w^{l_t^i}$
MFMC Estimator
$\frac{1}{p}\sum_{i=1}^{p}\sum_{t=0}^{T-1} r^{l_t^i}$
$\sum_{t=0}^{T-1} r^{l_t^1}$
$\sum_{t=0}^{T-1} r^{l_t^2}$
$\sum_{t=0}^{T-1} r^{l_t^{p-1}}$
$\sum_{t=0}^{T-1} r^{l_t^p}$

Figure 6.1: The MFMC estimator builds $p$ broken trajectories made of one-step transitions.

A tabular version of the algorithm for building the broken trajectories is given on Table 4. It returns a set of indices of one-step transitions $\left\{l_t^i\right\}_{i=1,t=0}^{i=p,t=T-1}$ from $\mathcal{F}_n$ based on $h$, $x_0$, the distance metric $\Delta$ and the parameter $p$. Based on this set of indices, we define our MFMC estimate of the expected return of the policy $h$ when starting from the initial state $x_0$ by:

**Definition 6.4.4 (Model-free Monte Carlo estimator)**

$$\forall x_0 \in \mathcal{X}, \mathfrak{M}_p^h\left(\mathcal{F}_n, x_0\right) = \frac{1}{p}\sum_{i=1}^{p}\sum_{t=0}^{T-1} r^{l_t^i}. \qquad (6.15)$$

Figure 6.1 illustrates the MFMC estimator. Note that the computation of the MFMC estimator $\mathfrak{M}_p^h\left(\mathcal{F}_n, x_0\right)$ has a linear complexity with respect to the cardinality $n$ of $\mathcal{F}_n$ and the length $T$ of the broken trajectories.

### 6.4.3 Analysis of the MFMC estimator

In this section we characterize some main properties of our estimator. To this end, we proceed as follows:

1. we first abstract away from the given sample $\mathcal{F}_n$ by instead considering an ensemble of samples of pairs which are "compatible" with $\mathcal{F}_n$ in the following sense: from the sample
$$\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^{n}, \tag{6.16}$$
we keep only the sample
$$\mathcal{P}_n = \left\{\left(x^l, u^l\right)\right\}_{l=1}^{n} \in (\mathcal{X} \times \mathcal{U})^n \tag{6.17}$$
of state-action pairs, and we then consider the ensemble of samples of one-step transitions of size $n$ that could be generated by completing each pair $(x^l, u^l)$ of $\mathcal{P}_n$ by drawing for each $l$ a disturbance signal $w^l$ at random from $p_{\mathcal{W}}(.)$, and by recording the resulting values of $f(x^l, u^l, w^l)$ and $\rho(x^l, u^l, w^l)$. We denote by $\tilde{\mathcal{F}}_n$ one such "random" set of one-step transitions defined by a random draw of $n$ disturbance signals $w^l \quad l = 1 \dots n$. The sample of one-step transitions $\mathcal{F}_n$ is thus a realization of the random set $\tilde{\mathcal{F}}_n$;

2. we then study the distribution of our estimator $\mathfrak{M}_p^h(\tilde{\mathcal{F}}_n, x_0)$, seen as a function of the random set $\tilde{\mathcal{F}}_n$ ; in order to characterize this distribution, we express its bias and its variance as a function of a measure of the density of the sample $\mathcal{P}_n$, defined by its "$k-$sparsity"; this is the smallest radius such that all $\Delta$-balls in $\mathcal{X} \times \mathcal{U}$ of this radius contain at least $k$ elements from $\mathcal{P}_n$. The use of this notion implies that the space $\mathcal{X} \times \mathcal{U}$ is bounded (when measured using the distance metric $\Delta$).

The bias and variance characterization will be done under some additional assumptions detailed below. After that, we state the main theorems formulating these characterizations. Proofs are given in the Appendix.

**Assumption 6.4.5 (Lipschitz continuity of the functions $f$, $\rho$ and $h$)**
*We assume that the dynamics $f$, the reward function $\rho$ and the policy $h$ are Lipschitz continuous, i.e.,*
$\exists L_f, L_\rho, L_h \in \mathbb{R}^+ : \forall\, (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W}, \forall t \in \{0, \dots, T-1\},$

$$\|f(x,u,w) - f(x',u',w)\|_{\mathcal{X}} \leq L_f(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}), \tag{6.18}$$
$$|\rho(x,u,w) - \rho(x',u',w)| \leq L_\rho(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}), \tag{6.19}$$
$$\|h(t,x) - h(t,x')\|_{\mathcal{U}} \leq L_h\|x - x'\|_{\mathcal{X}}\ , \tag{6.20}$$

*where $\|.\|_{\mathcal{X}}$ and $\|.\|_{\mathcal{U}}$ denote the chosen norms over the spaces $\mathcal{X}$ and $\mathcal{U}$, respectively.*

**Definition 6.4.6 (Distance metric $\Delta$)**
$\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2$,

$$\Delta((x,u),(x',u')) = (\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}) . \tag{6.21}$$

**Definition 6.4.7 ($k-$sparsity of a sample $\mathcal{P}_n$)**
*We suppose that $\mathcal{X} \times \mathcal{U}$ is bounded when measured using the distance metric $\Delta$, and, given $k \in \mathbb{N}_0$ with $k \leq n$, we define the $k-$sparsity, $\alpha_k(\mathcal{P}_n)$ of the sample $\mathcal{P}_n$ by*

$$\alpha_k(\mathcal{P}_n) = \sup_{(x,u) \in \mathcal{X} \times \mathcal{U}} \left\{ \Delta_k^{\mathcal{P}_n}(x,u) \right\} , \tag{6.22}$$

*where $\Delta_k^{\mathcal{P}_n}(x,u)$ denotes the distance of $(x,u)$ to its $k-$th nearest neighbor (using the distance metric $\Delta$) in the $\mathcal{P}_n$ sample.*

We propose to compute an upper bound of the bias and variance of the MFMC estimator. To this end, we denote by $E^h_{p,\mathcal{P}_n}(x_0)$ the expected value:

**Definition 6.4.8 (Expected value of the MFMC estimator)**
$\forall x_0 \in \mathcal{X}$,

$$E^h_{p,\mathcal{P}_n}(x_0) = \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \mathfrak{M}^h_p(\tilde{\mathcal{F}}_n, x_0) \right] . \tag{6.23}$$

**Bias of the MFMC estimator.**

We have the following theorem:

**Theorem 6.4.9 (Bias of the MFMC estimator)**

$$\forall x_0 \in \mathcal{X}, \qquad \left| J^h(x_0) - E^h_{p,\mathcal{P}_n}(x_0) \right| \leq C \alpha_{pT}(\mathcal{P}_n) \tag{6.24}$$

$$\textit{with } C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left[ L_f (1 + L_h) \right]^i . \tag{6.25}$$

**Proof.** Before giving the proof of Theorem 6.4.9, we first give three preliminary lemmas. Given a disturbance vector

$$\Omega = [\Omega(0), \ldots, \Omega(T-1)] \in \mathcal{W}^T , \tag{6.26}$$

we define the $\Omega$-disturbed state-action value function $Q^{h,\Omega}_{T-t}(x,u)$ for $t \in \{0, \ldots, T-1\}$ as follows:

**Definition 6.4.10 ( $\Omega$-disturbed state-action value function)**
$\forall t \in \{0, \ldots, T-1\}, \forall (x,u) \in \mathcal{X} \times \mathcal{U}, \forall \Omega \in \mathcal{W}^T$,

$$Q^{h,\Omega}_{T-t}(x,u) = \rho(x,u,\Omega(t)) + \sum_{t'=t+1}^{T-1} \rho(x_{t'}, h(t', x_{t'}), \Omega(t')) \tag{6.27}$$

*with*

$$x_{t+1} = f(x,u,\Omega(t)) \tag{6.28}$$

*and*

$$\forall t' \in \{t+1, \ldots, T-1\}\,, x_{t'+1} = f(x_{t'}, h(t', x_{t'}), \Omega(t')). \tag{6.29}$$

Then, we define the expected return given $\Omega$ the quantity

**Definition 6.4.11 (Expected return given $\Omega$)**
$\forall x_0 \in \mathcal{X}, \forall \Omega \in \mathcal{W}^T$,

$$\mathbb{E}[R^h(x_0)|\Omega] = \underset{w_0,\ldots,w_{T-1} \sim p_{\mathcal{W}}(.)}{\mathbb{E}}[R^h(x_0)|w_0 = \Omega(0), \ldots, w_{T-1} = \Omega(T-1)]. \tag{6.30}$$

From there, we have the two following trivial results:

**Proposition 6.4.12**
$\forall x_0 \in \mathcal{X}, \forall \Omega \in \mathcal{W}^T$,

$$\mathbb{E}[R^h(x_0)|\Omega] = Q^{h,\Omega}_T(x_0, h(0, x_0))\,. \tag{6.31}$$

**Proposition 6.4.13**
$\forall (x,u) \in \mathcal{X} \times \mathcal{U}$, $\forall \Omega \in \mathcal{W}^T$,

$$\begin{aligned} Q^{h,\Omega}_{T-t+1}(x,u) &= \rho(x,u,\Omega(t-1)) \\ &+ Q^{h,\Omega}_{T-t}\big(f\big(x,u,\Omega(t-1)\big), h\big(t, f(x,u,\Omega(t-1))\big)\big)\,. \end{aligned} \tag{6.32}$$

Then, we have the following lemma.

**Lemma 6.4.14 (Lipschitz Continuity of $Q^{h,\Omega}_{T-t}$)**
$\forall t \in \{0, \ldots, T-1\}, \forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2$,

$$\left| Q^{h,\Omega}_{T-t}(x,u) - Q^{h,\Omega}_{T-t}(x',u') \right| \leq L_{Q_{T-t}} \Delta((x,u),(x',u')) \tag{6.33}$$

*with*

$$L_{Q_{T-t}} = L_\rho \sum_{i=0}^{T-t-1} \left[L_f(1+L_h)\right]^i. \tag{6.34}$$

**Proof.** We denote by $\mathcal{H}(T-t)$ the proposition:
$\mathcal{H}(T-t) : \forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2$,

$$\left|Q^{h,\Omega}_{T-t}(x,u) - Q^{h,\Omega}_{T-t}(x',u')\right| \leq L_{Q_{T-t}} \Delta((x,u),(x',u'))\ . \tag{6.35}$$

We prove by induction that $\mathcal{H}(T-t)$ is true $\forall t \in \{0, \ldots, T-1\}$. For the sake of conciseness, we denote use the notation

$$\Delta^Q_{T-t} = \left|Q^{h,\Omega}_{T-t}(x,u) - Q^{h,\Omega}_{T-t}(x',u')\right|\ . \tag{6.36}$$

- **Basis:** $t = T-1$

We have

$$\Delta^Q_1 = |\rho(x,u,\Omega(T-1)) - \rho(x',u',\Omega(T-1)|, \tag{6.37}$$

and the Lipschitz continuity of $\rho$ allows to write

$$\Delta^Q_1 \leq L_\rho\big(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}\big)\ = L_\rho \Delta((x,u),(x',u'))\ . \tag{6.38}$$

This proves $\mathcal{H}(1)$.

- **Induction step:** We suppose that $\mathcal{H}(T-t)$ is true, $1 \leq t \leq T-1$.

Using Equation 6.4.13, one has

$$\begin{aligned}
\Delta^Q_{T-t+1} \quad &= \quad \left|Q^{h,\Omega}_{T-t+1}(x,u) - Q^{h,\Omega}_{T-t+1}(x',u')\right| && (6.39)\\
&= \quad \Big|\rho(x,u,\Omega(t-1)) - \rho(x',u',\Omega(t-1)) \\
&+ \quad Q^{h,\Omega}_{T-t}(f(x,u,\Omega(t-1)), h(t, f(x,u,\Omega(t-1)))) \\
&- \quad Q^{h,\Omega}_{T-t}(f(x',u',\Omega(t-1)), h(t, f(x',u',\Omega(t-1))))\Big| && (6.40)
\end{aligned}$$

and, from there,

$$\begin{aligned}
\Delta^Q_{T-t+1} \quad &\leq \quad \big|\rho(x,u,\Omega(t-1)) - \rho(x',u',\Omega(t-1))\big| \\
&+ \quad \big|Q^{h,\Omega}_{T-t}(f(x,u,\Omega(t-1)), h(t, f(x,u,\Omega(t-1)))) \\
&- \quad Q^{h,\Omega}_{T-t}(f(x',u',\Omega(t-1)), h(t, f(x',u',\Omega(t-1))))\big|. && (6.41)
\end{aligned}$$

$\mathcal{H}(T-t)$ and the Lipschitz continuity of $\rho$ give

$$\begin{aligned}
\Delta^{Q}_{T-t+1} &\leq L_{\rho}\Delta((x,u),(x',u')) \\
&+ L_{Q_{T-t}}\Delta((f(x,u,\Omega(t-1)),h(t,f(x,u,\Omega(t-1)))), \\
&\quad (f(x',u',\Omega(t-1)),h(t,f(x',u',\Omega(t-1))))) \; .
\end{aligned} \tag{6.42}$$

Using the Lipschitz continuity of $f$ and $h$, we have

$$\begin{aligned}
\Delta^{Q}_{T-t+1} &\leq L_{\rho}\Delta((x,u),(x',u')) \\
&+ L_{Q_{T-t}}\big(L_f\Delta((x,u),(x',u')) + L_h L_f \Delta((x,u),(x',u'))\big),
\end{aligned} \tag{6.43}$$

and, from there,

$$\Delta^{Q}_{T-t+1} \leq L_{Q_{T-t+1}}\Delta((x,u),(x',u')) \tag{6.44}$$

since

$$L_{Q_{T-t+1}} \doteq L_{\rho} + L_{Q_{T-t}} L_f (1 + L_h). \tag{6.45}$$

This proves $\mathcal{H}\ (T-t+1)$ and ends the proof. ■

**Definition 6.4.15 (Disturbance vector associated with a broken trajectory)**
*Given a broken trajectory*

$$\tau^i = \left[\left(x^{l^i_t}, u^{l^i_t}, r^{l^i_t}, y^{l^i_t}\right)\right]_{t=0}^{T-1} \tag{6.46}$$

*we denote by* $\Omega^{\tau^i}$ *its associated disturbance vector*

$$\Omega^{\tau^i} = [w^{l^i_0}, \ldots, w^{l^i_{T-1}}] \; , \tag{6.47}$$

*i.e. the vector made of the* $T$ *unknown disturbances that affected the generation of the one-step transitions* $\left(x^{l^i_t}, u^{l^i_t}, r^{l^i_t}, y^{l^i_t}\right)$ *(cf. first item of Section 6.4.3).*

We give the following lemma.

**Lemma 6.4.16 (Bounds on the expected return given $\Omega$)**
$\forall x_0 \in \mathcal{X}, \forall i \in \{1,\ldots,p\}$,

$$b^h(\tau^i, x_0) \leq \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \leq a^h(\tau^i, x_0) \; , \tag{6.48}$$

*with*

$$b^h(\tau^i, x_0) = \sum_{t=0}^{T-1} \left[ r^{l^i_t} - L_{Q_{T-t}} \delta^i_t \right] \ , \tag{6.49}$$

$$a^h(\tau^i, x_0) = \sum_{t=0}^{T-1} \left[ r^{l^i_t} + L_{Q_{T-t}} \delta^i_t \right] \ , \tag{6.50}$$

$$\delta^i_t = \Delta\left( \left( x^{l^i_t}, u^{l^i_t} \right), \left( y^{l^i_{t-1}}, h\left( t, y^{l^i_{t-1}} \right) \right) \right) \ , \forall t \in \{0, \dots, T-1\} \ , \tag{6.51}$$

$$y^{l^i_{-1}} = x_0, \forall i \in \{1, \dots, p\}. \tag{6.52}$$

**Proof.** Let us first prove the lower bound. With $u_0 = h(0, x_0)$, the Lipschitz continuity of $Q_T^{h,\Omega^{\tau^i}}$ gives

$$\left| Q_T^{h,\Omega^{\tau^i}}(x_0, u_0) - Q_T^{h,\Omega^{\tau^i}}(x^{l^i_0}, u^{l^i_0}) \right| \leq L_{Q_T} \Delta\left( (x_0, u_0), \left( x^{l^i_0}, u^{l^i_0} \right) \right) \ . \tag{6.53}$$

According to Proposition (6.4.12),

$$Q_T^{h,\Omega^{\tau^i}}(x_0, u_0) = \mathbb{E}\left[ R^h(x_0) | \Omega^{\tau^i} \right] . \tag{6.54}$$

Thus,

$$\begin{aligned} & \left| \mathbb{E}\left[ R^h(x_0) | \Omega^{\tau^i} \right] - Q_T^{h,\Omega^{\tau^i}} \left( x^{l^i_0}, u^{l^i_0} \right) \right| \\ & = \left| Q_T^{h,\Omega^{\tau^i}}(x_0, h(0, x_0)) - Q_T^{h,\Omega^{\tau^i}} \left( x^{l^i_0}, u^{l^i_0} \right) \right| \qquad (6.55) \\ & \leq L_{Q_T} \Delta\left( (x_0, h(0, x_0)), \left( x^{l^i_0}, u^{l^i_0} \right) \right) \ . \qquad (6.56) \end{aligned}$$

It follows that

$$Q_T^{h,\Omega^{\tau^i}} \left( x^{l^i_0}, u^{l^i_0} \right) - L_{Q_T} \delta^i_0 \leq \mathbb{E}\left[ R^h(x_0) | \Omega^{\tau^i} \right] \ . \tag{6.57}$$

Using Equation (6.4.13) we have

$$\begin{aligned} Q_T^{h,\Omega^{\tau^i}} \left( x^{l^i_0}, u^{l^i_0} \right) \ & = \ \rho\left( x^{l^i_0}, u^{l^i_0}, w^{l^i_0} \right) \\ & + \ Q_{T-1}^{h,\Omega^{\tau^i}} \left( f\left( x^{l^i_0}, u^{l^i_0}, w^{l^i_0} \right), h\left( 1, f\left( x^{l^i_0}, u^{l^i_0}, w^{l^i_0} \right) \right) \right) \ . \end{aligned} \tag{6.58}$$

By definition of $\Omega^{\tau^i}$, we have

$$\rho\left(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}\right) = r^{l_0^i} \tag{6.59}$$

and

$$f\left(x^{l_0^i}, u^{l_0^i}, w^{l_0^i}\right) = y^{l_0^i} \ . \tag{6.60}$$

From there

$$Q_T^{h,\Omega^{\tau^i}}\left(x^{l_0^i}, u^{l_0^i}\right) = r^{l_0^i} + Q_{T-1}^{h,\Omega^{\tau^i}}\left(y^{l_0^i}, h\left(1, y^{l_0^i}\right)\right) \ , \tag{6.61}$$

and

$$Q_{T-1}^{h,\Omega^{\tau^i}}\left(y^{l_0^i}, h\left(1, y^{l_0^i}\right)\right) + r^{l_0^i} - L_{Q_T}\delta_0^i \leq \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \ . \tag{6.62}$$

The Lipschitz continuity of $Q_{T-1}^{h,\Omega^{\tau^i}}$ gives

$$\begin{aligned} &\left|Q_{T-1}^{h,\Omega^{\tau^i}}\left(y^{l_0^i}, h\left(1, y^{l_0^i}\right)\right) - Q_{T-1}^{h,\Omega^{\tau^i}}\left(x^{l_1^i}, u^{l_1^i}\right)\right| \\ &\leq L_{Q_{T-1}}\Delta\left(\left(y^{l_0^i}, h\left(1, y^{l_0^i}\right)\right), \left(x^{l_1^i}, u^{l_1^i}\right)\right) &(6.63) \\ &= L_{Q_{T-1}}\delta_1^i, &(6.64) \end{aligned}$$

which implies that

$$Q_{T-1}^{h,\Omega^{\tau^i}}\left(x^{l_1^i}, u^{l_1^i}\right) - L_{Q_{T-1}}\delta_1^i \leq Q_{T-1}^{h,\Omega^{\tau^i}}\left(y^{l_0^i}, h\left(1, y^{l_0^i}\right)\right) \ . \tag{6.65}$$

We therefore have

$$Q_{T-1}^{h,\Omega^{\tau^i}}\left(x^{l_1^i}, u^{l_1^i}\right) + r^{l_0^i} - L_{Q_T}\delta_0^i - L_{Q_{T-1}}\delta_1^i \leq \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right]. \tag{6.66}$$

The proof is completed by iterating this derivation. The upper bound is proved similarly. ■

We give a third lemma.

**Lemma 6.4.17**
$\forall x_0 \in \mathcal{X}, \forall i \in \{1, \ldots, p\}$,

$$a^h\left(\tau^i, x_0\right) - b^h\left(\tau^i, x_0\right) \leq 2C\alpha_{pT}\left(\mathcal{P}_n\right) \tag{6.67}$$

*with*

$$C = \sum_{t=0}^{T-1} L_{Q_{T-t}} \ . \tag{6.68}$$

**Proof.** By construction of the bounds, one has

$$a^h\left(\tau^i, x_0\right) - b^h\left(\tau^i, x_0\right) = \sum_{t=0}^{T-1} 2L_{Q_{T-t}} \delta_t^i \ . \tag{6.69}$$

The MFMC algorithm chooses $p \times T$ different one-step transitions to build the MFMC estimator by minimizing the distance $\Delta((y^{l_{t-1}^i}, h(t, y^{l_{t-1}^i})), (x^{l_t^i}, u^{l_t^i}))$, so by definition of the $k$-sparsity of the sample $\mathcal{P}_n$ with $k = pT$, one has

$$\begin{aligned}
\delta_t^i &= \Delta\left(\left(y^{l_{t-1}^i}, h\left(t, y^{l_{t-1}^i}\right)\right), \left(x^{l_t^i}, u^{l_t^i}\right)\right) && (6.70)\\
&\leq \Delta_{pT}^{\mathcal{P}_n}\left(y^{l_{t-1}^i}, h\left(t, y^{l_{t-1}^i}\right)\right) && (6.71)\\
&\leq \alpha_{pT}\left(\mathcal{P}_n\right) \ , && (6.72)
\end{aligned}$$

which ends the proof. ■

Using those three lemmas, one can now compute an upper bound on the bias of the MFMC estimator.

**Proof of Theorem 6.4.9** By definition of $a^h(\tau^i, x_0)$ and $b^h(\tau^i, x_0)$, we have

$$\forall i \in \{1, \ldots, p\}, \frac{b^h\left(\tau^i, x_0\right) + a^h\left(\tau^i, x_0\right)}{2} = \sum_{t=0}^{T-1} r^{l_t^i} \ . \tag{6.73}$$

Then, according to Lemmas 6.4.16 and 6.4.17, we have $\forall i \in \{1, \ldots, p\}$ ,

$$\begin{aligned}
&\left| \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \mathbb{E}\left[R^h(x_0) | \Omega^{\tau^i}\right] - \sum_{t=0}^{T-1} r^{l_t^i} \right] \right| \\
&\leq \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \left| \mathbb{E}\left[R^h(x_0) | \Omega^{\tau^i}\right] - \sum_{t=0}^{T-1} r^{l_t^i} \right| \right] && (6.74)\\
&\leq C \alpha_{pT}(\mathcal{P}_n) \ . && (6.75)
\end{aligned}$$

Thus,

$$
\begin{aligned}
&\left|\frac{1}{p}\sum_{i=1}^{p} \underset{w^1,\ldots,w^n\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[\mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] - \sum_{t=0}^{T-1} r^{l_t^i}\right]\right| \\
&\leq \frac{1}{p}\sum_{i=1}^{p} \left| \underset{w^1,\ldots,w^n\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[\mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] - \sum_{t=0}^{T-1} r^{l_t^i}\right]\right| && (6.76)\\
&\leq C\alpha_{pT}\left(\mathcal{P}_n\right) \;, && (6.77)
\end{aligned}
$$

which can be reformulated

$$
\left| \underset{w^1,\ldots,w^n\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[\frac{1}{p}\sum_{i=1}^{p} \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right]\right] - E^h_{p,\mathcal{P}_n}(x_0)\right| \leq C\alpha_{pT}\left(\mathcal{P}_n\right) \;, \qquad (6.78)
$$

since

$$
\frac{1}{p}\sum_{i=1}^{p}\sum_{t=0}^{T-1} r^{l_t^i} = \mathfrak{M}_p^h(\tilde{\mathcal{F}}_n, x_0) \;. \qquad (6.79)
$$

Since the MFMC algorithm chooses $p \times T$ different one-step transitions, all the disturbances $\left\{w^{l_t^i}\right\}_{i=1,t=0}^{i=p,t=T-1}$ are i.i.d. according to $p_{\mathcal{W}}(.)$. For all $i \in \{1,\ldots,p\}$, The law of total expectation gives

$$
\begin{aligned}
&\underset{w^{l_0^i},\ldots,w^{l_{T-1}^i}\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \underset{w^{l_0^i},\ldots,w^{l_{T-1}^i}\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[R^h(x_0)|\Omega^{\tau^i}\right]\right] \\
&= \underset{w_0,\ldots,w_{T-1}\sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[R^h(x_0)\right] && (6.80)\\
&= J^h(x_0) \;. && (6.81)
\end{aligned}
$$

This ends the proof. ■

This formula shows that the bias is bounded closer to the target estimate if the sample sparsity is small. Note that the sample sparsity itself actually only depends on the sample $\mathcal{P}_n$ and on the value of $p$ (it will increase with the number of trajectories used by our algorithm).

### Variance of the MFMC estimator

We denote by $V^h_{p,\mathcal{P}_n}(x_0)$ the variance of the MFMC estimator defined as follows.

**Definition 6.4.18 (Variance of the MFMC estimator)**
$\forall x_0 \in \mathcal{X}$,

$$V^h_{p,\mathcal{P}_n}(x_0) = \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[\mathfrak{M}^h_p(\tilde{\mathcal{F}}_n, x_0)\right] \tag{6.82}$$

$$= \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[\left(\mathfrak{M}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) - E^h_{p,\mathcal{P}_n}(x_0)\right)^2\right] . \tag{6.83}$$

We give the following theorem.

**Theorem 6.4.19 (Variance of the MFMC estimator)**
*$\forall x_0 \in \mathcal{X}$,*

$$V^h_{p,\mathcal{P}_n}(x_0) \leq \left(\frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C\alpha_{pT}\left(\mathcal{P}_n\right)\right)^2 \tag{6.84}$$

*with*

$$C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left[L_f(1 + L_h)\right]^i . \tag{6.85}$$

**Proof.** We first have the following lemma.

**Lemma 6.4.20 (Variance of a sum of random variables)**
*Let $X_0, \ldots, X_{T-1}$ be $T$ random variables with finite variances $\sigma^2_0, \ldots, \sigma^2_{T-1}$ respectively. Then,*

$$Var\left[\sum_{t=0}^{T-1} X_t\right] \leq \left(\sum_{t=0}^{T-1} \sigma_t\right)^2 . \tag{6.86}$$

**Proof.** The proof is obtained by induction on the number of random variables using the formula

$$Cov(X_i, X_j) \leq \sigma_i \sigma_j \ , \forall i, j \in \{0, \ldots, T-1\} \tag{6.87}$$

which is a straightforward consequence of the Cauchy-Schwarz inequality.
**Proof of Theorem 6.4.19.**

**Definition 6.4.21**
*Let $x_0 \in \mathcal{X}$. We denote by $\mathfrak{N}^h_p(\tilde{\mathcal{F}}_n, x_0)$ the random variable*

$$\mathfrak{N}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) = \mathfrak{M}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) - \frac{1}{p}\sum_{i=1}^{p} \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] . \tag{6.88}$$

According to Lemma 6.4.20, we can write

$$\begin{aligned} V^h_{p,\mathcal{P}_n}(x_0) \quad & \leq \quad \left( \sqrt{\underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \frac{1}{p} \sum_{i=1}^{p} \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right]} \right. \\ & + \quad \left. \sqrt{\underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \mathfrak{N}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) \right]} \right)^2 \end{aligned} \tag{6.89}$$

Since all the $\left\{ w^{l^i_t} \right\}_{i=1,t=0}^{i=p,t=T-1}$ are i.i.d. according to $p_{\mathcal{W}}(.)$ (cf proof of Theorem 6.4.9), the law of total expectation gives

$$\underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \frac{1}{p} \sum_{i=1}^{p} \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right] = \frac{\sigma^2_{R^h}(x_0)}{p} \ . \tag{6.90}$$

Now, let us focus on $\underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \mathfrak{N}^h_p(\tilde{\mathcal{F}}_n, x_0) \right]$. By definition, we have

$$\mathfrak{N}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) = \frac{1}{p} \sum_{i=1}^{p} \left[ \sum_{t=0}^{T-1} r^{l^i_t} - \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right] \ . \tag{6.91}$$

Then, according to Lemma 6.4.20, we have

$$\begin{aligned} & \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \mathfrak{N}^h_p\left(\tilde{\mathcal{F}}_n, x_0\right) \right] \\ & \leq \frac{1}{p^2} \left( \sum_{i=1}^{p} \sqrt{\underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \sum_{t=0}^{T-1} r^{l^i_t} - \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right]} \ \right)^2 \end{aligned} \tag{6.92}$$

Then, we can write

$$\begin{aligned} & \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{Var} \left[ \sum_{t=0}^{T-1} r^{l^i_t} - \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right] \\ & \leq \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \left( \sum_{t=0}^{T-1} r^{l^i_t} - \mathbb{E}\left[R^h(x_0)|\Omega^{\tau^i}\right] \right)^2 \right] && (6.93) \\ & \leq \underset{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[ \left( a^h\left(\tau^i, x_0\right) - b^h\left(\tau^i, x_0\right) \right)^2 \right] = \left( a^h\left(\tau^i, x_0\right) - b^h\left(\tau^i, x_0\right) \right)^2 && (6.94) \\ & \leq 4C^2 (\alpha_{pT}\left(\mathcal{P}_n\right))^2 \ , && (6.95) \end{aligned}$$

since $\sum_{t=0}^{T-1} r^{l_t^i}$ and $\mathbb{E}[R^h(x_0)|\Omega^{\tau^i}]$ both belong to the interval $[b^h(\tau^i, x_0), a^h(\tau^i, x_0)]$ whose width is bounded by $2C\alpha_{pT}(\mathcal{P}_n)$ according to Lemma 6.4.17.

Using Equations (6.89), (6.90), (6.92) and (6.95), we have

$$V^h_{p,\mathcal{P}_n}(x_0) \leq \left( \frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C\alpha_{pT}(\mathcal{P}_n) \right)^2 \tag{6.96}$$

which ends the proof. ■

We see that the variance of our MFMC estimator is guaranteed to be close to that of the classical MC estimator if the sample sparsity is small enough. Note, however, that our bounds are quite conservative given the very weak assumptions that we exploit about the considered optimal control problem.

## 6.5 Illustration

In this section, we illustrate the MFMC estimator on an academic problem.

### 6.5.1 Problem statement

The system dynamics and the reward function are given by

$$x_{t+1} = \sin\left( \frac{\pi}{2}(x_t + u_t + w_t) \right) \tag{6.97}$$

and

$$\rho(x_t, u_t, w_t) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_t^2 + u_t^2)} + w_t \tag{6.98}$$

with the state space $\mathcal{X}$ being equal to $[-1, 1]$ and the action space $\mathcal{U}$ to $[-\frac{1}{2}, \frac{1}{2}]$ . The disturbance $w_t$ is an element of the interval $\mathcal{W} = [-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ with $\epsilon = 0.1$ and $p_{\mathcal{W}}$ is a uniform probability distribution over the interval $\mathcal{W}$. The optimization horizon $T$ is equal to 15. The policy $h$ whose performances have to be evaluated writes

$$h(t, x) = -\frac{x}{2}, \forall x \in \mathcal{X}, \forall t \in \{0, \dots, T-1\} \,. \tag{6.99}$$

The initial state of the system is set $x_0 = -0.5$ . The samples of one-step transitions $\mathcal{F}_n$ that are used as substitute for $f$, $\rho$ and $p_{\mathcal{W}}(.)$ in our experiments have been generated according to the mechanism described in Section 6.4.3.

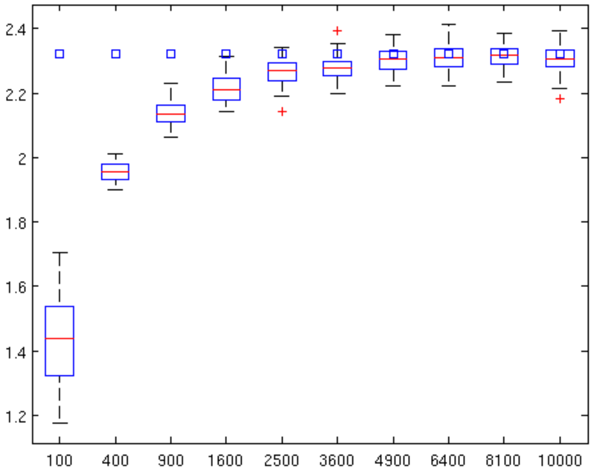

Figure 6.2: Computations of the MFMC estimator for different cardinalities of the sample of one-step transitions with $p = 10$. Squares represent $J^h(x_0)$.

### 6.5.2 Results

For our first set of experiments, we choose to work with a value of $p = 10$ i.e., the MFMC estimator rebuilds $10$ broken trajectories to estimate $J^h(-0.5)$. In these experiments, for different cardinalities

$$n_j = (10j)^2 \quad j = 1 \ldots 10, \tag{6.100}$$

we generate $50$ sets

$$\mathcal{F}^1_{n_j}, \ldots, \mathcal{F}^{50}_{n_j} \tag{6.101}$$

and run our MFMC estimator on each of these sets. For a given cardinality $n_j = m_j^2$, all the different samples $\mathcal{F}^1_{n_j}, \ldots, \mathcal{F}^{50}_{n_j}$ are generated considering the same couples $(x^l, u^l) \quad l = 1 \ldots n_j$ that uniformly cover the space according to the relationships

$$x^l = -1 + \frac{2j_1}{m_j} \tag{6.102}$$

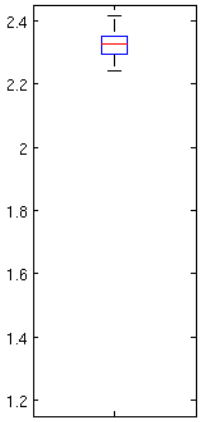

Figure 6.3: Computations of the MC estimator with $p = 10$.

and

$$u^l = -1 + \frac{2j_2}{m_j} \tag{6.103}$$

with

$$j_1, j_2 \in \{0, \ldots, m_j - 1\}. \tag{6.104}$$

The results of this first set of experiments are gathered in Figure 6.2. For every value of $n_j$ considered in our experiments, the $50$ values outputted by the MFMC estimator are concisely represented by a box plot. The box has lines at the lower quartile, median, and upper quartile values. Whiskers extend from each end of the box to the adjacent values in the data within $1.5$ times the interquartile range from the ends of the box. Outliers are data with values beyond the ends of the whiskers and are displayed with a red $+$ sign. The squares represent an accurate estimate of $J^h(-0.5)$ computed by running thousands of Monte Carlo simulations. As we observe, when the samples increase in size (which corresponds to a decrease of the $pT-$sparsity $\alpha_{pT}(\mathcal{P}_n)$) the MFMC estimator is more likely to output accurate estimations of $J^h(-0.5)$. As explained throughout this chapter, there exist many similarities between the model-free

MFMC estimator and the model-based MC estimator. These can be empirically illustrated by putting Figure 6.2 in perspective with Figure 6.3. This figure reports the results obtained by 50 independent runs of the MC estimator, every of these runs using also $p = 10$ trajectories. As expected, one can see that the MFMC estimator tends to behave similarly to the MC estimator when the cardinality of the sample increases.

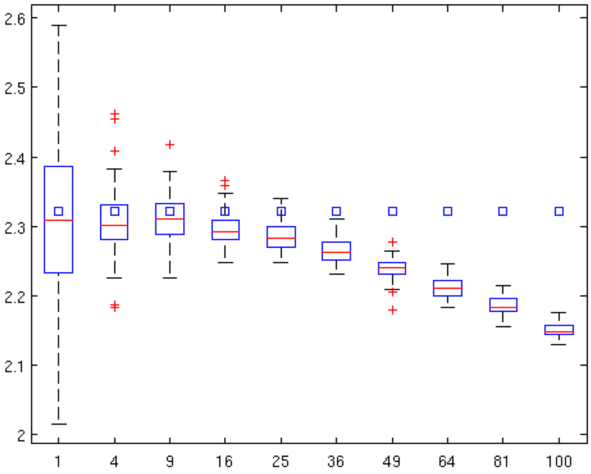

Figure 6.4: Computations of the MFMC estimator for different values of the number of broken trajectories $p$. Squares represent $J^h(x_0)$.

In our second set of experiments, we choose to study the influence of the number of broken trajectories $p$ upon which the MFMC estimator bases its prediction. In these experiments, for each value

$$p_j = j^2 \quad j = 1 \dots 10 \tag{6.105}$$

we generate 50 samples

$$\mathcal{F}^1_{10,000}, \dots, \mathcal{F}^{50}_{10,000} \tag{6.106}$$

of one-step transitions of cardinality $10,000$ and use these samples to compute the MFMC estimator. The results are plotted in Figure 6.4. This figure shows that the bias of the MFMC estimator seems to be relatively small for small values of $p$ and to

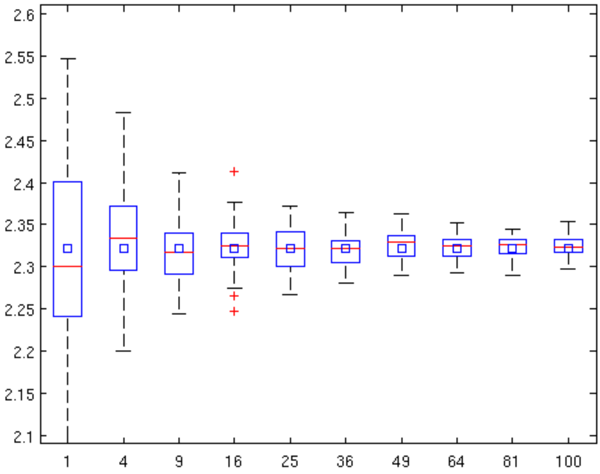

Figure 6.5: Computations of the MC estimator for different values of the number of trajectories $p$. Squares represent $J^h(x_0)$.

increase with $p$. This is in accordance with Theorem 6.4.9 which bounds the bias with an expression that is increasing with $p$.

In Figure 6.5, we have plotted the evolution of the values outputted by the model-based MC estimator when the number of trajectories it considers in its prediction increases. While, for small number of trajectories, it behaves similarly to the MFMC estimator, the quality of its predictions steadily increases with $p$, while it is not the case for the MFMC estimator whose performances degrade once $p$ crosses a threshold value. Notice that this threshold value could be made larger by increasing the size of the samples of one-step system transitions used as input of the MFMC algorithm.

## 6.6 Conclusions

We have proposed in this chapter an estimator of the expected return of a policy in a model-free setting. The estimator named MFMC works by rebuilding from a sample of one-step transitions a set of broken trajectories and by averaging the sum of rewards gathered along these latter trajectories. In this respect, it can be seen as an extension to a model-free setting of the standard model-based Monte Carlo policy evaluation technique. We have provided bounds on the bias and variance of the MFMC estimator ; these were depending among others on the sparsity of the sample of one-step transitions and the Lipschitz constants associated with the system dynamics, reward function and policy. These bounds show that when the sample sparsity becomes small, the bias of the estimator decreases to zero and its variance converges to the variance of the Monte Carlo estimator.

The work presented in this chapter could be extended along several lines. For example, it would be interesting to consider disturbances whose probability distributions are conditioned on the states and the actions and to study how the bounds given in this chapter should be modified to remain valid in such a setting. Another interesting research direction would be to investigate how the bounds proposed in this chapter could be useful for choosing automatically the parameters of the MFMC estimator which are the number $p$ of broken trajectories it rebuilds and the distance metric $\Delta$ it uses to select its set of broken trajectories.

However, the bound on the variance of the MFMC estimator depends explicitly on the "natural" variance of the sum of rewards along trajectories of the system when starting from the same initial state. Using this bound for determining automatically $p$ (and/or $\Delta$) suggests therefore to investigate how an upper bound on this natural variance could be inferred from the sample of one-step transitions. Finally, this MFMC estimator adds to the arsenal of techniques that have been proposed in the literature for computing an estimate of the expected return of a policy in a model-free setting. However, it is not yet clear how it would compete with such techniques. All these techniques have pros and cons and establishing which one to exploit for a specific problem certainly deserves further research.

# Bibliography

[1] A. Antos, R. Munos, and C. Szepesvári. Fitted Q-iteration in continuous action space MDPs. In *Advances in Neural Information Processing Systems 20, NIPS 2007*, 2007.

[2] S.J. Bradtke and A.G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.

[3] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming using Function Approximators*. Taylor & Francis CRC Press, 2010.

[4] P. Dayan. The convergence of TD($\lambda$) for general $\lambda$. *Machine Learning*, 8:341–162, 1992.

[5] C Dimitrakakis and M. G. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning*, 72:157–171, 2008.

[6] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[7] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, JMLR: W&CP 9*, pages 217–224, Chia Laguna, Sardinia, Italy, 2010.

[8] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Actes de la conférence francophone sur l'apprentissage automatique (CAP 2010), Clermont-Ferrand (France)*, 2010.

[9] R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, pages 815–857, 2008.

[10] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[11] M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML 2005)*, pages 317–328, Porto, Portugal, 2005.

[12] G.A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report 166, Cambridge University Engineering Department, 1994.

[13] R.S. Sutton. Learning to predict by the methods of temporal difference. *Machine Learning*, 3:9–44, 1988.

[14] R.S. Sutton, H. Reza Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[15] J.N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16:185–202, 1994.

[16] C.J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):179–192, 1992.

# Chapter 7

# Variable selection for dynamic treatment regimes: a reinforcement learning approach

*Dynamic treatment regimes (DTRs) can be inferred from data collected through some randomized clinical trials by using reinforcement learning algorithms. During these clinical trials, a large set of clinical indicators are usually monitored. However, it is often more convenient for clinicians to have DTRs which are only defined on a small set of indicators rather than on the original full set. To address this problem, we analyze the approximation architecture of the state-action value functions computed by the fitted Q iteration algorithm - a RL algorithm - using tree-based regressors in order to identify a small subset of relevant ones. The RL algorithm is then rerun by considering only as state variables these most relevant indicators to have DTRs defined on a small set of indicators. The approach is validated on benchmark problems inspired from the classical 'car on the hill' problem and the results obtained are positive.*

The work presented in this chapter was accepted for presentation at the *European Workshop on Reinforcement Learning* (EWRL 2008) [3].

In this chapter, we consider:

- a stochastic framework,
- a continuous state space and a finite action space.

## 7.1 Introduction

Nowadays, many diseases as for example HIV/AIDS, cancer, inflammatory or neurological diseases are seen by the medical community as being chronic-like diseases, resulting in medical treatments that can last over very long periods. For treating such diseases, physicians often adopt explicit, operationalized series of decision rules specifying how drug types and treatment levels should be administered over time, which are referred to in the medical community as Dynamic Treatment Regimes (DTRs). Designing an appropriate DTR for a given disease is a challenging issue. Among the difficulties encountered, we can mention the complex dynamics of the human body interacting with treatments and other environmental factors, as well as the often poor compliance to treatments due to the side effects of the drugs. While typically DTRs are based on clinical judgment and medical insight, since a few years the biostatistics community is investigating a new research field addressing specifically the problem of inferring in a well principled way DTRs directly from clinical data gathered from patients under treatment. Among the results already published in this area, we mention [6] which uses statistical tools for designing DTRs for psychotic patients.

One possible approach to infer DTR from the data collected through clinical trials is to formalize this problem as an optimal control problem for which most of the information available on the 'system dynamics' (the system is here the patient and the input of the system is the treatment) is 'hidden' in the clinical data. This problem has been vastly studied in Reinforcement Learning (RL), a subfield of machine learning (see e.g., [2]). Its application to the DTR problem would consist of processing the clinical data so as to compute a closed-loop treatment strategy which takes as inputs all the various clinical indicators which have been collected from the patients. Using policies computed in this way may however be inconvenient for the physicians who may prefer DTRs based on an as small as possible subset of *relevant* indicators rather than on the possibly very large set of variables monitored through the clinical trial. In this research, we therefore address the problem of determining a small subset of indicators among a larger set of candidate ones, in order to infer by RL convenient decision strategies. Our approach is closely inspired by work on 'variable selection' for supervised learning.

The rest of this chapter is organized as follows. In Section II we formalize the problem of inferring DTRs from clinical data as an optimal control problem for which the sole information available on the system dynamics is the one contained in the clinical data. We also briefly present the fitted $Q$ iteration algorithm which will be used to compute from these data a good approximate of the optimal policy. In Section III, we present our algorithm for selecting the most relevant clinical indicators and computing (near-) optimal policies defined only on these indicators. Section IV reports our simulation results and, finally, Section V concludes.

## 7.2 Learning from a sample

We assume that the information available for designing DTRs is a sample of discrete-time trajectories of treated patients, i.e. successive tuples $(x_t, u_t, x_{t+1})$, where $x_t$ represents the state of a patient at some time-step $t$ and lies in an $n$-dimensional space $\mathcal{X}$ of clinical indicators, $u_t$ is an element of the finite action space $\mathcal{U}$ (representing treatments taken by the patient in the time interval $[t, t+1]$), and $x_{t+1}$ is the state at the subsequent time-step.

We further suppose that the responses of patients suffering from a specific type of chronic disease all obey the same discrete-time dynamics:

$$x_{t+1} = f(x_t, u_t, w_t) \quad t = 0, 1, \dots \tag{7.1}$$

where disturbances $w_t \in \mathcal{W}$ are generated according to a probability distribution $p_{\mathcal{W}}(\cdot|x, u)$. Finally, we assume that one can associate to the state of the patient at time $t$ and to the action at time $t$, a reward signal

$$r_t = \rho(x_t, u_t, w_t) \in \mathbb{R} \tag{7.2}$$

which represents the 'well being' of the patient over the time interval $[t, t+1]$. Once the choice of the function $\rho$ has been realized (a problem often known as preference elicitation, see e.g., [4]), the problem of finding a 'good' DTR may be stated as an optimal control problem for which one seeks to find a policy which leads to a sequence of actions $(u_0, u_1, \dots, u_{T-1}) \in \mathcal{U}^T$, which maximizes, over the time horizon $T \in \mathbb{N}$, and for any initial state the $T-$stage return:

**Definition 7.2.1 ($T-$stage return)**
$\forall (u_0, \dots, u_{T-1}) \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$J^{u_0,\dots,u_{T-1}}(x_0) = \underset{\substack{w_t \\ t=0,1,\dots,T-1}}{\mathbb{E}} \left[ \sum_{t=0}^{T-1} \rho(x_t, u_t, w_t) \right] \tag{7.3}$$

One can show (see e.g., [2]) that there exists a policy

$$\pi_T^* : \{0, \dots, T-1\} \times \mathcal{X} \to \mathcal{U} \tag{7.4}$$

which produces such a sequence of actions for any initial state $x_0$. To characterize these optimal $T$-stage policies, let us define iteratively the sequence of *state-action value functions* $Q_N : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, $N = 1, \dots, T$ as follows:

**Definition 7.2.2 (State-action value functions)**
$\forall N \in \{1, \ldots, T\}, \forall (x, u) \in \mathcal{X} \times \mathcal{U}$,

$$Q_N(x, u) = \mathop{\mathbb{E}}_{w} \left[ \rho(x, u, w) + \max_{u' \in U} Q_{N-1}(f(x, u, w), u') \right] \tag{7.5}$$

*with*

$$Q_0(x, u) = 0 \,, \forall (x, u) \in \mathcal{X} \times \mathcal{U} \,. \tag{7.6}$$

By using results from the dynamic programming theory, one can write that, for all $t \in \{1, \ldots, T-1\}$ and $x \in \mathcal{X}$, the policy

$$\pi_T^\star(t, x) = \arg\max_{u \in U} Q_{T-t}(x, u) \tag{7.7}$$

is a $T$-step optimal policy.

Exploiting directly (7.5) for computing the $Q_N$-functions is not possible in our context since $f$ is unknown and replaced here by a sample of $n \in \mathbb{N}_0$ one-step trajectories:

$$\mathcal{F}_n = \left\{ \left( x^l, u^l, r^l, y^l \right) \right\}_{l=1}^{n} \tag{7.8}$$

where $\forall l \in \{1, \ldots, n\}$ ,

$$r^l = \rho(x^l, u^l, w^l) \tag{7.9}$$

$$y^l = f(x^l, u^l, w^l) \tag{7.10}$$

and

$$w^l \sim p(\cdot | x^l, u^l) \,. \tag{7.11}$$

To address this problem, we exploit the fitted $Q$ iteration algorithm which offers a way for computing the $Q_N$-functions from the sole knowledge of $\mathcal{F}_n$ [2] (the fittted $Q$ iteration algorithm is also detailed in Appendix A). In a few words, this RL algorithm computes these functions by solving a $T$-length sequence of standard supervised learning problems. A $\tilde{Q}_N$-function - approximation of the $Q_N$-function as defined by Eqn (7.5) - is computed by solving the $N$th supervised learning problem of the sequence. The training set for this problem is computed from $\mathcal{F}_n$ and the $\tilde{Q}_{N-1}$-function. We exploit the particular structure of these tree-based approximators in order to identify the most relevant clinical indicators among the $n_{\mathcal{X}}$ candidate ones.

## 7.3 Selection of clinical indicators

As mentioned in Section 7.1, we propose to find a small subset of state variables (clinical indicators), the $m_{\mathcal{X}}$ ($m_{\mathcal{X}} \ll n_{\mathcal{X}}$) most relevant ones with respect to a certain criterion, so as to create an $m_{\mathcal{X}}$-dimensional subspace of $\mathcal{X}$ on which DTRs will be computed. The approach we propose for this exploits the tree structure of the $\tilde{Q}_N$-functions computed by the fitted $Q$ iteration algorithm. This approach will score each attribute by estimating the variance reduction it can be associated with by propagating the training sample over the different tree structures (this criterion was originally proposed in the context of supervised learning for identifying relevant attributes in the context of regression tree induction [7]). In our context, it evaluates the relevance of each state variable $x(i) \quad i = 1 \ldots n_{\mathcal{X}}$, by the score function defined as follows:

**Definition 7.3.1 (Score function)**
$\forall i \in \{1, \ldots, n_{\mathcal{X}}\}$,

$$S(x(i)) = \frac{\sum_{N=1}^{T} \sum_{\tau \in \tilde{Q}_N} \sum_{\nu \in \tau} \delta\left(\nu, x(i)\right) \Delta_{var}(\nu) |\nu|}{\sum_{N=1}^{T} \sum_{\tau \in \tilde{Q}_N} \sum_{\nu \in \tau} \Delta_{var}(\nu) |\nu|} \tag{7.12}$$

*where $\nu$ is a nonterminal node in a tree $\tau$ (one of those used to build the ensemble model representing one of the $\tilde{Q}_N$-functions), $\delta(\nu, x(i)) = 1$ if $x(i)$ is used to split at node $\nu$ or equal to zero otherwise, $|\nu|$ is the number of samples at node $\nu$, $\Delta_{var}(\nu)$ is the variance reduction when splitting node $\nu$:*

$$\Delta_{var}(\nu) = v(\nu) - \frac{|\nu_L|}{|\nu|} v(\nu_L) - \frac{|\nu_R|}{|\nu|} v(\nu_R) \tag{7.13}$$

*where $\nu_L$ (resp. $\nu_R$) is the left-son node (resp. the right-son node) of node $\nu$, and $v(\nu)$ (resp. $v(\nu_L)$ and $v(\nu_R)$) is the variance of the sample at node $\nu$ (resp. $\nu_L$ and $\nu_R$).*

The approach then sorts the state variables $x(i)$ by decreasing values of their score so as to identify the $m_{\mathcal{X}}$ most relevant ones. A DTR defined on this subset of variables is then computed by running the fitted $Q$ iteration algorithm again on a 'modified $\mathcal{F}_n$', where the state variables of $x^l$ and $y^l$ that are not among these $m_{\mathcal{X}}$ most relevant ones are discarded.

The algorithm for computing a DTR defined on a small subset of state variables is thus as follows:

1. Compute the $\tilde{Q}_N$-functions ($N = 1, \ldots, T$) using the fitted $Q$ iteration algorithm on $\mathcal{F}_n$;

2. Compute the score function for each state variable, and determine the $m_{\mathcal{X}}$ best ones;
3. Run the fitted $Q$ iteration algorithm on

$$\widetilde{\mathcal{F}}_n = \left\{ \left( \widetilde{x}^l, u^l, r^l, \widetilde{y}^l \right) \right\}_{l=1}^{n} \tag{7.14}$$

where

$$\widetilde{x} = \widetilde{M} x, \tag{7.15}$$

and $\widetilde{M}$ is a $m_{\mathcal{X}} \times n_{\mathcal{X}}$ boolean matrix where $\widetilde{m}_{i,j} = 1$ if the state variable $x(j)$ is the $i$-th most relevant one and 0 otherwise.

## 7.4 Preliminary validation

Table 7.1: Variance reduction scores of the different state variables for various experimental settings. The first column gives the cardinality of the sets $\mathcal{F}_n$ considered (the elements of these sets have been generated by drawing $(x^l, u^l)$ at random in $\mathcal{X} \times \mathcal{U}$ and computing $y^l$ from the system dynamics (7.1)). The second column gives the number of Non-Relevant Variables (NRV) added to the original state vector. The remaining columns report the different scores $S(\cdot)$ computed for the different (relevant and non-relevant) variables considered in each scenario.

| $\#\mathcal{F}_n = n$ | NB. OF NRV | $z$ | $\dot{z}$ | NRV 1 | NRV 2 | NRV 3 |
|---|---|---|---|---|---|---|
| 5000 | 0 | 0.24 | 0.35 | - | - | - |
| 5000 | 1 | 0.27 | 0.30 | 0.08 | - | - |
| 5000 | 2 | 0.16 | 0.26 | 0.12 | 0.06 | - |
| 5000 | 3 | 0.15 | 0.18 | 0.07 | 0.07 | 0.09 |
| 10000 | 1 | 0.16 | 0.34 | 0.09 | - | - |
| 10000 | 2 | 0.20 | 0.19 | 0.08 | 0.12 | - |
| 10000 | 3 | 0.15 | 0.31 | 0.05 | 0.05 | 0.06 |
| 20000 | 1 | 0.18 | 0.27 | 0.10 | - | - |
| 20000 | 2 | 0.15 | 0.24 | 0.08 | 0.10 | - |
| 20000 | 3 | 0.15 | 0.21 | 0.08 | 0.08 | 0.07 |

We report in this section simulation results that have been obtained by testing the proposed approach on a modified version of the classical car-on-the-hill benchmark problem ([2], also reported in Section 5.5.1 of Chapter 5).[1] The original car-on-the-hill problem has two state variables, the position $z$ and the speed $\dot{z}$ of the car, and one action variable $u$ which represents the acceleration of the car. The action can only take two discrete values (full acceleration or full deceleration).

For illustrating our approach, we have slightly modified the car-on-the-hill problem by adding new "dummy state variables" to the problem. These variables take at each time $t$ a value which is drawn independently from all other variable-values according to a uniform probability distribution over the interval $[0, 1]$ and do not affect the actual dynamics of the problem.

In such a context, our approach is expected to associate the highest scores $S(\cdot)$ to the variables $z$ and $\dot{z}$ since these are the only ones that actually contain relevant information about the optimal policy of the system. Results obtained are presented in Table 1. As one can see, the approach consistently gives the two highest scores to $z$ and $\dot{z}$.

## 7.5 Conclusions

We have proposed in this chapter an approach for computing from clinical data DTR strategies defined on a small subset of clinical indicators. The approach is based on a formalization of the problem as an optimal control problem for which the system dynamics is unknown and replaced to some extent by the information contained in the clinical data. Once this formalization is done, the tree-based approximators computed by the fitted $Q$ iteration algorithm used for inferring policies from the data are analyzed to identify the 'most relevant variables'. This identification is carried out by exploiting variance reduction concepts which are determinant in our approach. Preliminary simulation results carried out on some academic examples have shown that the proposed approach for selecting the most relevant indicators is promising.

Techniques based on variance reduction for selecting the most relevant indicators have already been successfully used in supervised learning (SL) (see, e.g., [7]) and have inspired the work reported in this chapter. But many other techniques for selecting relevant variables have also been proposed in the literature on supervised learning, such as for example those based on Bayesian approaches [1, 5]. In this respect, it will be interesting to investigate to which extent these other approaches could be usefully exploited in our reinforcement learning context.

[1] The optimality criterion of the car on the hill problem is usually chosen as being the sum of the discounted rewards observed over an infinite time horizon. We have chosen here to shorten this infinite time horizon to 50 steps and not use discount factors in order to have an optimality criterion in accordance with (7.3).

A next step in our research is to test our variable selection approach for getting policies defined on a small subset of indicators on real-life clinical data. However, in such a context, one difficulty we will face is the inability to determine whether the indicators selected by our approach are indeed the right ones since no accurate model of the system will be available. This issue is closely related to the problem of estimating the quality of a policy in model-free RL. We believe it is made particularly relevant in the context of DTRs since it would probably be unacceptable to adopt some dynamic treatment regimes which would trade the use of a smaller number of decision variables at the expense of a significant deterioration of the health of patients.

# Bibliography

[1] W. Cui. *Variable Selection: Empirical Bayes vs. Fully Bayes.* PhD thesis, The University of Texas at Austin, 2002.

[2] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[3] R. Fonteneau, L. Wehenkel, and D. Ernst. Variable selection for dynamic treatment regimes: a reinforcement learning approach. In *European Workshop on Reinforcement Learning (EWRL 2008)*, Villeneuve d'Ascq, France, 2008.

[4] D.G. Froberg and R.L. Kane. Methodology for measuring health-state preferences–ii: Scaling methods. *Journal of Clinical Epidemiology*, 42:459–471, 1989.

[5] E.I. George and R.E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7, 2:339–373, 1997.

[6] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[7] L. Wehenkel. *Automatic Learning Techniques in Power Systems*. Kluwer Academic, Boston, 1998.

# Chapter 8

# Conclusions and future works

The present dissertation gathers research contributions in the field of batch mode reinforcement learning. This research work was motivated by real life applications, and especially challenges raised by the design of dynamic treatment regimes. More specifically, we have addressed the problems of:

- Computing bounds on the performance of control policies in a deterministic framework [3, 6, 4],
- Computing tight estimates of the performance of control policies in a stochastic framework [8],
- Determining where to sample additional information in order to enrich the current batch collection of data [7, 9],
- Selecting subsets of relevant variables for building more convenient control policies [10].

For each contribution, restricted assumptions have been done and finding ways to relax these assumptions would certainly be useful to extend our results. Section 8.1 elaborates on such extensions of our work. Section 8.2 presents more general research directions for enriching this body of work in batch mode reinforcement learning.

## 8.1 Choices and assumptions

### 8.1.1 Finite optimization horizon

All along this dissertation, we have considered a finite optimization horizon. In the reinforcement learning literature, many works consider infinite horizon and discounted sum of rewards, and focus on the computation of high-performance stationary control policies. The decision to consider optimal control problems with finite optimization horizons was suggested by the fact that, for many real life applications such as for instance the building of dynamic treatment regimes, searching only for stationary control policies is not appropriate.

However, we believe that a majority of the finite horizon approaches exposed in this dissertation could be extended to infinite discounted frameworks. This has already been done by other authors for several of them (see for instance [20], where an infinite time-horizon inference algorithm is built upon the approach developed in Chapter 2 [3]).

### 8.1.2 Observability

In the present work, we have chosen to consider optimal control problems for which systems are fully observable. However, for many real-life applications, the state vector may not be fully observable. Investigating how the research contributions presented in this dissertation could be extended to a partially observable setting [13] would certainly be interesting.

### 8.1.3 Lipschitz continuity assumptions

Theoretical results exposed in this dissertation have been obtained under Lipschitz continuity assumptions on the system dynamics, reward function, and sometimes, control policies. Lipschitz continuity assumptions are quite popular in batch mode reinforcement learning probably because they can easily be used to prove the convergence towards optimal solutions when the sparsity of the batch collection of data decreases towards zero.

However, Lipschitz continuity assumptions are often too restrictive, and, for instance, they are violated when the system dynamics and/or the reward function are not continuous. It would be interesting to investigate how our results could be adapted to fit other (less restrictive) assumptions such as for instance Hölder continuity assumptions, or even assumptions which are not related to continuity.

### 8.1.4 Extensions to stochastic framework

A large part of the research presented in this dissertation has been developed in deterministic frameworks. Some of these contributions have already been, in a sense, extended to a stochastic framework. Indeed, the introduction of the Model-free Monte Carlo estimator [8], presented in Chapter 6, can be seen as an extension of the approaches for computing bounds proposed in Chapter 2.

The sampling strategies for determining where to sample informative additional data, presented in Chapters 4 and 5, could also be extended to stochastic frameworks. Indeed, the first sampling strategy [7], exposed in Chapter 4, could probably be extended in a similar way to what is proposed in Chapter 6 since it is based on the computation of bounds that are similar to those presented in Chapter 3. With respect to the second sampling strategy exposed in Chapter 5, the policy falsification principle upon which it is built could in our opinion still be exploited in a stochastic setting.

The $\min\max$ approach towards generalization in batch mode reinforcement learning was proposed in a deterministic framework, mainly for clarity reasons. In a stochastic framework, a $\min\max$ approach towards generalization would certainly have to exploit risk-sensitive formulations. This is developed below in Section 8.2.2.

## 8.2 Promising research directions

Beyond the technical extensions detailed above in Section 8.1, we believe that more general promising research directions can be suggested by the research material reported in previous chapters.

### 8.2.1 A Model-free Monte Carlo-based inference algorithm

An immediate extension would be to exploit the MFMC estimator for developing a new batch mode inference algorithm. For instance, the MFMC estimator could be integrated into a direct policy search algorithm, or one could also develop a policy iteration algorithm based on the MFMC criterion.

### 8.2.2 Towards risk-sensitive formulations

All along this dissertation, the performances of control policies in stochastic environments have been evaluated through their expected return. The model-free Monte Carlo estimator [8] which is detailed in Chapter 6 estimates the expected return of control policies using artificial trajectories, also called broken trajectories. We believe that the estimation technique based on artificial trajectories could be extended to estimate the return distribution of control policies. From there, one could derive an inference algorithm for computing risk-sensitive control policies [1, 14].

### 8.2.3 Analytically investigating the policy falsification-based sampling strategy

The sampling strategy based on the policy falsification principle [9] reported in Chapter 5 is still empirical. We plan to analyze the theoretical properties of algorithms built upon the policy falsification principle by using regularity assumptions on the problems such as for instance Lipschitz continuity assumptions.

### 8.2.4 Developing a unified formalization around the notion of artificial trajectories

One common characteristic of the majority of the contributions reported in this dissertation is the use of sequences of one-step system transitions, also called “artificial trajectories" or "broken trajectories". Existing batch mode reinforcement learning algorithms using regression trees, nearest-neighbors methods [2] or kernel-based approximators [18] output solutions that can also be characterized using sets of artificial trajectories. We believe this concept of artificial trajectories could lead to a general paradigm for designing and analyzing reinforcement learning algorithms.

### 8.2.5 Testing algorithms on actual clinical data

The inference algorithm CGRL [6] exposed in Chapter 3 as well as the variable selection technique [10] detailed in Chapter 7 have already been run on simulated data which were generated using a mathematical model of the HIV infection [11, 5].

It would indeed be interesting to see how the different algorithms developed in this dissertation behave when run on real clinical data [12, 15, 16, 17, 19]. It is however worth stressing that getting access to actual clinical data is difficult.

# Bibliography

[1] B. Defourny, D. Ernst, and L. Wehenkel. Risk-aware decision making and dynamic programming. *Selected for oral presentation at the NIPS-08 Workshop on Model Uncertainty and Risk in Reinforcement Learning, Whistler, Canada*, 2008.

[2] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[3] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2009)*, Nashville, TN, USA, 2009.

[4] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst. Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence: International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series: Communications in Computer and Information Science (CCIS)*, volume 129, pages 61–77. Springer, Heidelberg, 2011.

[5] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Dynamic treatment regimes using reinforcement learning: a cautious generalization approach. In *Benelux Bioinformatics Conference (BBC) 2009 (Poster), Liège, Belgium, December 14-15*, 2009.

[6] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. A cautious approach to generalization in reinforcement learning. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010)*, Valencia, Spain, 2010.

[7] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Generating informative trajectories by using bounds on the return of control policies. In *Proceedings of*

*the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*, 2010.

[8] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Model-free Monte Carlo–like policy evaluation. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, JMLR: W&CP 9*, pages 217–224, Chia Laguna, Sardinia, Italy, 2010.

[9] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Active exploration by searching for experiments falsifying an already induced policy. *To be published in the Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2011), Paris, France*, 2011.

[10] R. Fonteneau, L. Wehenkel, and D. Ernst. Variable selection for dynamic treatment regimes: a reinforcement learning approach. In *European Workshop on Reinforcement Learning (EWRL 2008)*, Villeneuve d'Ascq, France, 2008.

[11] R. Fonteneau, L. Wehenkel, and D. Ernst. Variable selection for dynamic treatment regimes: a reinforcement learning approach. In *Computer Intelligence and Learning (CIL) doctoral school (Poster), in parallel to the ECML/PKDD conference in Antwerpen*, 2008.

[12] A. Guez, R. Vincent, M. Avoli, and J. Pineau. Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *Innovative Applications of Artificial Intelligence (IAAI)*, 2008.

[13] Michael L. Littman. A tutorial on partially observable markov decision processes. *Journal of Mathematical Psychology*, 53(3):119 – 125, 2009. Special Issue: Dynamic Decision Making.

[14] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka. Nonparametric return density estimation for reinforcement learning. In *Proceedings of 27th International Conference on Machine Learning (ICML2010), Haifa, Israel, Jun. 21-25*, 2010.

[15] S.A. Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24:1455–1481, 2005.

[16] S.A. Murphy and D. Almirall. Dynamic Treatment Regimes. *Encyclopedia of Medical Decision Making*, 2008.

[17] S.A. Murphy, D. Oslin, A.J. Rush, and J. for MCATS 2006 Zhu. Methodological challenges in constructing effective treatment sequences for chronic disorders. In *Neuropsychopharmacology*, volume 32(2), pages 257–62, 2006.

[18] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

[19] M. Qian, I. Nahum-Shani, and S.A. Murphy. Dynamic treatment regimes. *To appear as a book chapter in Modern Clinical Trial Analysis , edited by X. Tu and W. Tang, Springer Science*, 2009.

[20] E. Rachelson and M. G. Lagoudakis. On the Locality of Action Domination in Sequential Decision Making. In *Tenth Intl. Symposium on AI and Math*, 2010.

# Appendix A

# Fitted $Q$ iteration

*We detail in this appendix the Fitted Q Iteration (FQI) algorithm combined with extremely randomized trees (Extra Trees). This algorithm was first published by Ernst et al. [5] in 2005.*

In this appendix, we consider:

- a stochastic framework,
- a continuous state space and a finite action space.

## A.1 Introduction

This appendix details the Fitted $Q$ Iteration (FQI) algorithm [5]. This algorithm was one of the first batch mode reinforcement learning algorithms to be published. Nowadays, it is probably the most popular batch mode reinforcement learning algorithm, probably because of its excellent inference performances.

Many successful applications of the fitted $Q$ iteration algorithm have been reported, for instance in the field of robotics [9, 10] power systems [6], image processing [7], water reservoir optimization [3] and dynamic treatment regimes [4].

In this dissertation, the FQI algorithm is used in Chapter 3 where it is compared with the CGRL algorithm on the puddle world benchmark, and in Chapter 7 where we build upon it a variable selection strategy.

## A.2 Problem statement

We consider a system having a discrete-time dynamics described by

$$x_{t+1} = f(x_t, u_t, w_t), \quad t = 0, 1, \ldots \tag{A.1}$$

where for all $t \in \mathbb{N}$, the state $x_t$ is an element of the state space $\mathcal{X}$, the action $u_t$ is an element of the finite action space $\mathcal{U}$ and the random disturbance $w_t$ an element of the disturbance space $\mathcal{W}$. The disturbance $w_t$ is generated by the time-invariant conditional probability distribution $p_{\mathcal{W}}(.|x_t, u_t)$. To the transition from $t$ to $t+1$ is associated an instantaneous reward signal

$$r_t = \rho(x_t, u_t, w_t) \tag{A.2}$$

where $\rho$ is the reward function supposed here to be bounded by some constant $B_\rho$. Let

$$h : \mathcal{X} \to \mathcal{U} \tag{A.3}$$

denote a stationary control policy and $J_\infty^\mu(x_0)$ denote the expected return obtained over an infinite time horizon when the system is controlled using the policy $h$ (i.e., when $u_t = h(x_t)$ , $\forall t$) when starting from the initial state $x_0 \in \mathcal{X}$. For a given initial state $x_0$, $J_\infty^h(x_0)$ is defined as follows:

**Definition A.2.1 (Infinite time horizon expected return)**
$\forall x_0 \in \mathcal{X}$,

$$J_\infty^h(x_0) = \lim_{N \to \infty} \underset{w_t \quad t=0,1,\ldots}{\mathbb{E}} \left[ \sum_{t=0}^{N} \gamma^t \rho(x_t, u_t, w_t) \right] \tag{A.4}$$

*where $\gamma$ is a discount factor ($0 \leq \gamma < 1$) that weights short-term rewards more than long-term ones, and where the conditional expectation is taken over all trajectories starting with the initial state $x_0$.*

The goal is to find an optimal stationary policy $h^*$, i.e. a stationary policy that maximizes $J_\infty^h(x_0)$:

$$h^* \in \arg\max_h J_\infty^h(x_0) \; . \tag{A.5}$$

## A.3 The fitted $Q$ iteration algorithm

**Algorithm 5** The Fitted $Q$ Iteration algorithm.

```
Input:
    a set of one-step system transitions $\mathcal{F}_n = \{(x^l, u^l, r^l, y^l)\}$ ;
    a regression algorithm $\mathcal{RA}$ ;
Output: a near-optimal state-action value function from which a near-optimal control policy can be derived ;
Initialization:
Set $N$ to 0 ;
Let $\tilde{Q}_N$ be equal to zero all over the state-action space $\mathcal{X} \times \mathcal{U}$ ;
Algorithm:
while Stopping conditions are not reached do
  $N \leftarrow N + 1$ ;
  Build the dataset $D = \{(i^l, o^l)\}_{l=1}^n$ based on the function $\tilde{Q}_{N-1}$ and on the full set of one step system transitions $\mathcal{F}_n$ :

      $i^l = (x^l, u^l)$                                   (A.6)
      $o^l = r^l + \gamma \max_{u \in \mathcal{U}} \hat{Q}_{N-1}(y^l, u)$   (A.7)

  Use the regression algorithm $\mathcal{RA}$ to infer from $D$ the function $\tilde{Q}_N$:

      $\tilde{Q}_N = \mathcal{RA}(D)$ .                    (A.8)

end while
```

The Fitted $Q$ Iteration algorithm computes a near-optimal stationary policy $\tilde{h}^*$ from a sample of system transitions

$$\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\} \tag{A.9}$$

where $\forall l \in \{1, \ldots, n\}$ ,

$$r^l = \rho(x^l, u^l, w^l) \tag{A.10}$$

$$y^l = f(x^l, u^l, w^l) \tag{A.11}$$

and

$$w^l \sim p_{\mathcal{W}}(\cdot|x^l, u^l)\ . \tag{A.12}$$

**Definition A.3.1 (State-action value functions)**
*Let $(Q_N)_N$ be a sequence of state-action value functions defined over the state-action space $\mathcal{X} \times \mathcal{U}$ as follows:*

$$\begin{aligned} &\forall (x, u) \in \mathcal{X} \times \mathcal{U}\ , \\ &Q_0(x, u) = 0, && \text{(A.13)} \\ &\forall N \in \mathbb{N}, Q_{N+1}(x, u) = \underset{w \sim p_{\mathcal{W}}(\cdot|x,u)}{\mathbb{E}} \left[\rho(x, u, w) + \gamma \underset{u' \in \mathcal{U}}{\max} Q_N(f(x, u, w), u')\right]\ . \end{aligned}$$

(A.14)

Results from the Dynamic Programming theory [1, 2] ensure that the sequence of functions $(Q_N)_N$ converges towards a function $Q^*$ from which an optimal stationary control policy $h^*$ can be derived as follows:

$$\forall x \in \mathcal{X}, h^*(x) = \underset{u \in \mathcal{U}}{\arg\max} \quad Q^*(x, u) \tag{A.15}$$

The fitted $Q$ algorithm algorithm computes, from the set of system transitions $\mathcal{F}_n$, an approximation $\tilde{Q}^*$ of the optimal state-action value function $Q^*$, from which a near-optimal stationary control policy $\tilde{h}^*$ can be derived:

$$\forall x \in \mathcal{X}, \tilde{h}^*(x) = \underset{u \in \mathcal{U}}{\arg\max} \quad \tilde{Q}^*(x, u) \tag{A.16}$$

A tabular version of the fitted Q iteration algorithm is given in Figure 5. At each step this algorithm may use the full set of system transitions $\mathcal{F}_n$ together with the function computed at the previous step to determine a new training set which is used by a regression algorithm $\mathcal{RA}$ to compute the next function of the sequence. It produces a sequence of $\tilde{Q}_N$ functions, approximations of the $Q_N$ functions defined in Definition A.3.1.

## A.4 Finite-horizon version of FQI

The Fitted $Q$ iteration algorithm can also be applied to finite optimization horizon optimal control problems. Given a finite optimization horizon $T \in \mathbb{N}_0$, one can adapt the fitted $Q$ iteration algorithm by storing the $T$ approximated value functions $\tilde{Q}_1^*, \ldots, \tilde{Q}_T^*$ introduced in the previous section. A finite-time near-optimal (non-stationary) control policy $\tilde{h}^*$ is then obtained as follows:

$$\forall t \in \{0, \ldots, T-1\}, \forall x \in \mathcal{X}, \tilde{h}^*(t,x) = \arg\max_{u \in \mathcal{U}} \tilde{Q}_{T-t}^*(x,u) \ . \tag{A.17}$$

## A.5 Extremely randomized trees

The implementation of the fitted $Q$ iteration algorithm used in this dissertation uses extremely randomized trees [8] as regression algorithm $\mathcal{RA}$. This algorithm works by building several ($M \in \mathbb{N}_0$) regression trees and by averaging their predictions. Each tree is built from the complete original training set. To determine a test at a node, this algorithm selects $K \in \mathbb{N}_0$ cut-directions at random and for each cut-direction, a cut-point at random. It then computes a score for each of the $K$ tests and chooses among these $K$ tests the one that maximizes the score. The algorithm stops splitting a node when the number of elements in this node is less than a parameter $n_{\min} \in \mathbb{N}_0$.

Three parameters are thus associated to this algorithm: the number $M$ of trees to build, the number $K$ of candidate tests at each node and the minimal leaf size $n_{\min}$. We give in Algorithm 6 the full procedure for building an extremely randomized tree.

**Algorithm 6** Extremely Randomized Trees.

```
Function: Build a tree ;
Input: a training set $\mathcal{TS} = \left\{\left(i^l, o^l\right)\right\}_{l=1}^{n_{\mathcal{TS}}}$ ;
Output: a Tree ;
if
    (i) the cardinality of the training set $n_{\mathcal{TS}}$ satisfies $n_{\mathcal{TS}} < n_{\min}$ or
    (ii) all input variables are constant in $\mathcal{TS}$ or
    (iii) the output variables is constant in $\mathcal{TS}$,
then return a leaf labeled by the average $\frac{1}{n_{\mathcal{TS}}} \sum_{l=1}^{n_{\mathcal{TS}}} o^l$
otherwise
    Let $[i_j < t_j]$ = Find a test$(\mathcal{TS})$;
    Split $\mathcal{TS}$ into two subsets $\mathcal{TS}_l$ and $\mathcal{TS}_r$ according to the test $[i_j < t]$;
    Build $T_l$ = Buil a tree$(\mathcal{TS}_l)$ and $T_r$ = Buil a tree$(\mathcal{TS}_r)$ from these two subsets;
    Create a node with the test $[i_j < t_j]$, attach $T_l$ and $T_r$ as left and right subtrees of
this node and return the resulting tree.

Function: Find a test ;
Input: a training set $\mathcal{TS}$ ;
Output: a test $[i_j < t_j]$ ;
Select $K$ inputs $\{i_1, \ldots, i_K\}$, at random, without replacement, among all non constant input variables ;
for $k = 1$ to $K$ do
    Compute the maximal and minimal value of $i_k$ in $\mathcal{TX}$, denoted respectively $i_{k,\min}^{\mathcal{TS}}$
    and $i_{k,\max}^{\mathcal{TS}}$;
    Draw a discretization threshold $t_k$ uniformly in $\left]i_{k,\min}^{\mathcal{TS}}, i_{k,\max}^{\mathcal{TS}}\right[$
    Compute the score $S_k = Score\left([i_k < t_k], \mathcal{TS}\right)$ ;
end for
Return a test $[i_j < t_j]$ such that $S_j = \max_{k=1,\ldots,K} S_k$.

Function: Score ;
Input: a test $[i_j < t_j]$, a training set $\mathcal{TS}$;
Let $\mathcal{TS}_l$ (resp. $\mathcal{TS}_r$) the subset of cases from $\mathcal{TS}$ such that $[i_j < t_j]$ (resp. $[i_j \geq t_j]$);
Return $Score([i_j, t_j], \mathcal{TS}) = \frac{var(o|\mathcal{TS}) - \frac{n_{\mathcal{TS}_l}}{n_{\mathcal{TS}}} var(o|\mathcal{TS}_l) - \frac{n_{\mathcal{TS}_r}}{n_{\mathcal{TS}}} var(o|\mathcal{TS}_r)}{var(o|\mathcal{TS})}$ where
$var(o|\mathcal{TS})$ (resp. $var(o|\mathcal{TS}_l)$ and $var(o|\mathcal{TS}_r)$ ) is the empirical variance of the
output $o$ in the training set $\mathcal{TS}$ (resp. $\mathcal{TS}_l$ and $\mathcal{TS}_r$), and $n_{\mathcal{TS}}$ (resp. $n_{\mathcal{TS}_l}$ and
$n_{\mathcal{TS}_r}$) denotes the cardinality of the training set $\mathcal{TS}$ (resp. $\mathcal{TS}_l$ and $\mathcal{TS}_r$).
```

# Bibliography

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[2] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] A. Castelletti, S. Galelli, M. Restelli, and R. Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46, W09507,doi:10.1029/2009WR008898, 2010.

[4] D. Ernst. Selecting concise sets of samples for a reinforcement learning agent. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*, Singapore, 2005.

[5] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[6] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel. Reinforcement learning versus model predictive control: a comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39:517–529, 2009.

[7] D. Ernst, R. Marée, and L. Wehenkel. Reinforcement learning with raw image pixels as state input. In *International Workshop on Intelligent Computing in Pattern Analysis/Synthesis (IWICPAS). Proceedings series: Lecture Notes in Computer Science*, volume 4153, pages 446–454, 2006.

[8] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning.*, 36(Number 1):3–42, 2006.

[9] S. Lange and M. Riedmiller. Deep learning of visual control policies. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2010), Brugge, Belgium*, 2010.

[10] M. Riedmiller, T. Gabel, Hafner R., and S. Lange. Reinforcement learning fo robot soccer. *Autonomous Robots*, 27(1):55–74, 2009.

# Appendix B

# Computing bounds for kernel–based policy evaluation in reinforcement learning

*This appendix proposes an approach for computing bounds on the finite-time return of a policy using kernel-based approximators from a sample of trajectories in a continuous state space and deterministic framework.*

This appendix details some technical results cited in Section 3.8 of Chapter 3.

In this appendix, we consider:

- a deterministic framework,
- a continuous state space,
- a finite action space in the first part, and a continuous action space in the second part.

## B.1 Introduction

This appendix proposes an approach for computing bounds on the finite-time return of a policy using kernel-based approximators from a sample of trajectories in a continuous state space and deterministic framework. The computation of the bounds is detailed in two different settings. The first setting (Section B.3) focuses on the case of a finite action space where policies are open-loop sequences of actions. The second setting (Section B.4) considers a normed continuous action space with closed-loop Lipschitz continuous policies.

## B.2 Problem statement

We consider a deterministic discrete-time system whose dynamics over $T$ stages is described by a time-invariant equation:

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \dots, T-1, \tag{B.1}$$

where for all $t$, the state $x_t$ is an element of the continuous normed state space $(\mathcal{X}, \|.\|_{\mathcal{X}})$ and the action $u_t$ is an element of the finite action space $\mathcal{U}$. $T \in \mathbb{N}_0$ is referred to as the optimization horizon. The transition from $t$ to $t+1$ is associated with an instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{B.2}$$

where $\rho : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the reward function. We assume in this appendix that the reward function is bounded by a constant $A_\rho > 0$:

**Assumption B.2.1**

$$\exists A_\rho > 0 : \forall (x, u) \in \mathcal{X} \times \mathcal{U}, |\rho(x, u))| \leq A_\rho \,. \tag{B.3}$$

The system dynamics $f$ and the reward function $\rho$ are unknown. An arbitrary set of one-step system transitions

$$\mathcal{F} = \{(x^l, u^l, r^l, y^l)\}_{l=1}^{n} \tag{B.4}$$

is known, where each transition is such that

$$y^l = f(x^l, u^l) \tag{B.5}$$

and

$$r^l = \rho(x^l, u^l) \tag{B.6}$$

Given an initial state $x_0 \in \mathcal{X}$ and a sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, the $T-$stage return $J^{u_0,\ldots,u_{T-1}}(x_0)$ of the sequence $(u_0, \ldots, u_{T-1})$ is defined as follows.

**Definition B.2.2 ($T-$stage return of the sequence $(u_0, \ldots, u_{T-1})$)**
$\forall x_0 \in \mathcal{X}, \forall (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$,

$$J^{u_0,\ldots,u_{T-1}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t) \ .$$

In this appendix, the goal is to compute bounds on $J^{u_0,\ldots,u_{T-1}}(x_0)$ using kernel-based approximators. We first consider a finite action space with open-loop sequences of actions in Section B.3. In Section B.4, we consider a continuous normed action space where the sequences of actions are chosen according to a closed-loop control policy.

## B.3 Finite action space and open-loop control policy

In this section, we assume a finite action space $\mathcal{U}$. We consider open-loop sequences of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, $u_t$ being the action taken at time $t \in \{0, \ldots, T-1\}$ . We assume that the dynamics $f$ and the reward function $\rho$ are Lipschitz continuous:

**Assumption B.3.1 (Lipschitz continuity of $f$ and $\rho$)**
$\exists L_f, L_\rho \in \mathbb{R} : \forall (x, x') \in \mathcal{X}^2, \forall u \in \mathcal{U}, \forall t \in \{0, \ldots, T-1\}$,

$$\begin{aligned} \|f(x,u) - f(x',u)\|_{\mathcal{X}} &\leq L_f \|x - x'\|_{\mathcal{X}} \ , && \text{(B.7)} \\ |\rho(x,u) - \rho(x',u)| &\leq L_\rho \|x - x'\|_{\mathcal{X}} \ , && \text{(B.8)} \end{aligned}$$

*We further assume that two constants $L_f$ and $L_\rho$ satisfying the above-written inequalities are known.*

Under these assumptions, we want to compute for an arbitrary initial state $x_0 \in \mathcal{X}$ of the system some bounds on the $T-$stage return of any sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$.

### B.3.1 Kernel-based policy evaluation

Given a state $x \in \mathcal{X}$, we introduce the $(T-t)-$stage return of a sequence of actions $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ as follows:

**Definition B.3.2** (**$(T-t)-$stage return of a sequence of actions** $(u_0, \ldots, u_{T-1})$)
*Let $x \in \mathcal{X}$. For $t' \in \{T-t, \ldots, T-1\}$, we denote by $x_{t'+1}$ the state*

$$x_{t'+1} = f(x_{t'}, u_{t'}) \tag{B.9}$$

*with $x_{T-t} = x$. The $(T-t)-$stage return of the sequence $(u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$ when starting from $x \in \mathcal{X}$ is defined as*

$$J_{T-t}^{u_0,\ldots,u_{T-1}}(x) = \sum_{t'=T-t}^{T-1} \rho(x_{t'}, u_{t'}) \; . \tag{B.10}$$

The $T-$stage return of the sequence $(u_0, \ldots, u_{T-1})$ is thus given by

$$J^{u_0,\ldots,u_{T-1}}(x) = J_T^{u_0,\ldots,u_{T-1}}(x) \; . \tag{B.11}$$

We propose to approximate the sequence of mappings $\left(J_{T-t}^{u_0,\ldots,u_{T-1}}(.)\right)_{t=0}^{T-1}$ using kernels (see [1]) by a sequence $\left(\tilde{J}_{T-t}^{u_0,\ldots,u_{T-1}}(.)\right)_{t=0}^{T-1}$ computed as follows:

$$\forall x \in \mathcal{X}, \tilde{J}_0^{u_0,\ldots,u_{T-1}}(x) = J_0^{u_0,\ldots,u_{T-1}}(x) = 0 \; , \tag{B.12}$$

and, $\forall x \in \mathcal{X},\ \forall t \in \{0, \ldots, T-1\}$

$$\tilde{J}_{T-t}^{u_0,\ldots,u_{T-1}}(x) = \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) \Big( r^l + \hat{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(y^l) \Big) \; , \tag{B.13}$$

with

$$k_l(x) = \frac{\Phi\left(\frac{\|x - x^l\|_{\mathcal{X}}}{b}\right)}{\sum_{i=1}^{n} \mathbb{I}_{\{u^i = u_t\}} \Phi\left(\frac{\|x - x^i\|_{\mathcal{X}}}{b}\right)} \; , \tag{B.14}$$

where $\Phi : \mathbb{R}^+ \to \mathbb{R}^+$ is a univariate non-negative "mother kernel" function, and $b > 0$ is the bandwidth parameter. We also assume that

$$\forall x > 1, \Phi(x) = 0 \; . \tag{B.15}$$

We suppose that the functions $\{k_l\}_{l=1}^n$ are Lipschitz continuous:

**Assumption B.3.3 (Lipschitz continuity of $\{k_l\}_{l=1}^n$)**
$\forall l \in \{1, \ldots, n\}, \exists L_{k_l} > 0 :$

$$\forall (x', x'') \in \mathcal{X}^2, \big| k_l(x') - k_l(x'')) \big| \leq L_{k_l} \|x' - x''\|_{\mathcal{X}} \; . \tag{B.16}$$

Then, we define $L_k$ such that $L_k = \max_{l \in \{1,\ldots,n\}} L_{k_l}$. The kernel-based estimator (KBE), denoted by $\mathfrak{K}^{u_0,\ldots,u_{T-1}}(x)$, is defined as follows:

**Definition B.3.4 (Kernel-based estimator)**
$\forall x_0 \in \mathcal{X}$,

$$\mathfrak{K}^{u_0,\ldots,u_{T-1}}(x_0) = \tilde{J}_T^{u_0,\ldots,u_{T-1}}(x_0) \ . \tag{B.17}$$

We introduce the family of kernel operators $\left(K_{T-t}^{u_0,\ldots,u_{T-1}}\right)_{t=0}^{T-1}$ such that

**Definition B.3.5 (Finite action space kernel operators)**
*Let* $g : \mathcal{X} \to \mathbb{R}$. $\forall t \in \{0,\ldots,T-1\}, \forall x \in \mathcal{X}$,

$$\left(K_{T-t}^{u_0,\ldots,u_{T-1}} \circ g\right)(x) = \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) \left(r^l + g(y^l)\right) \ . \tag{B.18}$$

One has

$$\tilde{J}_{T-t}^{u_0,\ldots,u_{T-1}}(x) \quad = \quad \left(K_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}\right)(x) \ . \tag{B.19}$$

We also introduce the family of finite-horizon Bellman operators $\left(B_{T-t}^{u_0,\ldots,u_{T-1}}\right)_{t=0}^{T-1}$ as follows:

**Definition B.3.6 (Bellman operators)**
*Let* $g : \mathcal{X} \to \mathbb{R}$. $\forall t \in \{1,\ldots,T\}, \forall x \in \mathcal{X}$,

$$\left(B_{T-t}^{u_0,\ldots,u_{T-1}} \circ g\right)(x) = \rho(x, u_t) + g(f(x, u_t)) \ . \tag{B.20}$$

One has

$$J_{T-t}^{u_0,\ldots,u_{T-1}}(x) \quad = \quad \left(B_{T-t}^{u_0,\ldots,u_{T-1}} \circ J_{T-t-1}^{u_0,\ldots,u_{T-1}}\right)(x) \ . \tag{B.21}$$

We propose a first lemma that bounds the difference between the two operators $K_{T-t}^{u_0,\ldots,u_{T-1}}$ and $B_{T-t}^{u_0,\ldots,u_{T-1}}$ when applied to the approximated $(T-t-1)-$ return $\tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}$.

**Lemma B.3.7**
$\forall t \in \{0,\ldots,T-1\}, \forall x \in \mathcal{X}$,

$$\left|\left(K_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}\right)(x) - \left(B_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}\right)(x)\right| \leq C_{T-t} b \tag{B.22}$$

*with*

$$C_{T-t} = L_\rho + L_k L_f A_\rho (T-t-1) \ . \tag{B.23}$$

**Proof** Let $x \in \mathcal{X}$ .

- Let $t \in \{0, \dots, T-2\}$. Since

$$\sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) = 1, \tag{B.24}$$

one can write

$$\begin{aligned}
&\left| \left( K_{T-t}^{u_0,\dots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}} \right)(x) - \left( B_{T-t}^{u_0,\dots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}} \right)(x) \right| \\
&= \left| \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) \Big[ r^l - \rho(x, u_t) \right. \\
&\left. + \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(f(x, u_t)) \Big] \right| \qquad \text{(B.25)} \\
&\leq L_\rho \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) \|x^l - x\|_{\mathcal{X}} \\
&+ \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} \left| k_l(x) \left( \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(f(x, u_t)) \right) \right|
\end{aligned}$$

(B.26)

On the one hand, since

$$\forall z > 1, \Phi(z) = 0, \tag{B.27}$$

one has

$$\|x^l - x\|_{\mathcal{X}} \geq b \implies k_l(x) = 0. \tag{B.28}$$

Thus,

$$L_\rho \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_t\}} k_l(x) \|x^l - x\|_{\mathcal{X}} \leq L_\rho b\,. \tag{B.29}$$

On the other hand, one has

$$\begin{aligned}
&\tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\dots,u_{T-1}}(f(x, u_t)) \\
&= \sum_{j=1}^{n} \mathbb{I}_{\{u^j = u_{t+1}\}} \Big[ k_j(y^l) - k_j(f(x, u_t)) \Big] (r^j + \tilde{J}_{T-t-2}^{u_0,\dots,u_{T-1}}(y^j))
\end{aligned}$$

(B.30)

Since the reward function $\rho$ is bounded by $A_\rho$, one can write

$$\left|(r^j + \tilde{J}_{T-t-2}^{u_0,\ldots,u_{T-1}}(y^j))\right| \leq (T-t-1)A_\rho \; . \tag{B.31}$$

and according to the Lipschitz continuity of $k_j$ and $f$, one has

$$\begin{aligned}\left|k_j(y^l) - k_j(f(x,u_t))\right| &\leq L_{k_j}\|y^l - f(x,u_t)\|_{\mathcal{X}} && \text{(B.32)}\\ &\leq L_k\|y^l - f(x,u_t)\|_{\mathcal{X}} && \text{(B.33)}\\ &\leq L_kL_f\|x^l - x\|_{\mathcal{X}} \; . && \text{(B.34)}\end{aligned}$$

Equations (B.30), (B.31) and (B.34) allow to write

$$\left|\left(\tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(f(x,u_t))\right)\right| \\ \leq L_kL_f(T-t-1)A_\rho\|x^l - x\|_{\mathcal{X}} \; . \tag{B.35}$$

Equations (B.28) and (B.35) give

$$\left|\left(\tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(f(x,u_t))\right)\right| \leq L_kL_f(T-t-1)A_\rho b \tag{B.36}$$

and since

$$\sum_{l=1}^{n} \mathbb{I}_{u^l=u_t} k_l(x) = 1 \; , \tag{B.37}$$

one has

$$\sum_{l=1}^{n} \mathbb{I}_{u^l=u_t} \left\|k_l(x)(\tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(y^l) - \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(f(x,u_t)))\right\| \\ \leq L_kL_f b(T-t-1)A_\rho \tag{B.38}$$

Using Equations (B.26), (B.29) and (B.38), we can finally write
$\forall (x,t) \in \mathcal{X} \times \{0,\ldots,T-2\}$,

$$\left|K_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(x) - B_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(x)\right| \\ \leq (L_\rho + L_kL_f(T-t-1)A_\rho)b \; , \tag{B.39}$$

which proves the lemma for $t \in \{0,\ldots,T-2\}$.

- Let $t = T-1$. One has

$$\left|\left(K_1^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_0^{u_0,\ldots,u_{T-1}}\right)(x) - \left(B_1^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_0^{u_0,\ldots,u_{T-1}}\right)(x)\right|$$

$$\leq \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_{T-1}\}} k_l(x) \left|r^l - \rho(x, u_t)\right| \tag{B.40}$$

$$\leq \sum_{l=1}^{n} \mathbb{I}_{\{u^l = u_{T-1}\}} k_l(x) L_\rho \|x - x^l\| \leq L_\rho b \, , \tag{B.41}$$

since

$$\|x - x^l\| \geq b \implies k_l(x) = 0 \tag{B.42}$$

and

$$\sum_{l=1}^{n} \mathbb{I}_{u^l = u_t} k_l(x) = 1. \tag{B.43}$$

This shows that Equation (B.39) is also valid for $t = T-1$, and ends the proof.

Then, we have the following theorem.

**Theorem B.3.8 (Bounds on the actual return of a sequence $(u_0, \ldots, u_{T-1})$)**
*Let $x_0 \in \mathcal{X}$ be a given initial state. Then,*

$$|\mathfrak{K}^{u_0,\ldots,u_{T-1}}(x_0) - J^{u_0,\ldots,u_{T-1}}(x_0)| \leq \beta b \, , \tag{B.44}$$

*with*

$$\beta = \sum_{t=0}^{T-1} C_{T-t} \, . \tag{B.45}$$

**Proof** We use the notation $x_{t+1} = f(x_t, u_t),\ \forall t \in \{0, \ldots, T-1\}$. One has

$$J_T^{u_0,\ldots,u_{T-1}}(x_0) - \tilde{J}_T^{u_0,\ldots,u_{T-1}}(x_0)$$

$$= B_T^{u_0,\ldots,u_{T-1}} \circ J_{T-1}^{u_0,\ldots,u_{T-1}}(x_0) - K_T^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_0) \tag{B.46}$$

$$= B_T^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_0) - K_T^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_0)$$

$$+ B_T^{u_0,\ldots,u_{T-1}} J_{T-t-1}^{u_0,\ldots,u_{T-1}}(x_0) - B_T^{u_0,\ldots u_{T-1}} \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(x_0) \tag{B.47}$$

$$= B_T^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_0) - K_T^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_0)$$

$$+ J_{T-1}^{u_0,\ldots,u_{T-1}}(x_1) - \tilde{J}_{T-1}^{u_0,\ldots,u_{T-1}}(x_1) \, . \tag{B.48}$$

Using the recursive form of Equation (B.48), one has

$$J^{u_0,\ldots,u_{T-1}}(x) - \mathfrak{K}^{u_0,\ldots,u_{T-1}}(x) = J_T^{u_0,\ldots,u_{T-1}}(x) - \tilde{J}_T^{u_0,\ldots,u_{T-1}}(x) \tag{B.49}$$

$$= \sum_{t=0}^{T-1} B_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(x_t) - K_{T-t}^{u_0,\ldots,u_{T-1}} \circ \tilde{J}_{T-t-1}^{u_0,\ldots,u_{T-1}}(x_t) \tag{B.50}$$

Equation (B.50) and Lemma B.3.7 allow to write

$$\left| J_T^{u_0,\ldots,u_{T-1}}(x_0) - \mathfrak{K}^{u_0,\ldots,u_{T-1}}(x_0) \right| \leq \sum_{t=0}^{T-1} C_{T-t} b \,, \tag{B.51}$$

which ends the proof.

## B.4 Continuous action space and closed-loop control policy

In this section, the action space $(\mathcal{U}, \|.\|_{\mathcal{U}})$ is assumed to be continuous and normed. We consider a deterministic time-varying control policy

$$h : \{0, 1, \ldots, T-1\} \times X \to U \tag{B.52}$$

that selects at time $t$ the action $u_t$ based on the current time and the current state ($u_t = h(t, x_t)$). The $T-$stage return of the policy $h$ when starting from $x_0$ is defined as follows.

**Definition B.4.1 ($T-$stage return of the policy $h$)**
$\forall x_0 \in \mathcal{X}$,

$$J^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t)). \tag{B.53}$$

*where*

$$x_{t+1} = f(x_t, h(t, x_t)) \quad \forall t \in \{0, \ldots, T-1\} \,. \tag{B.54}$$

We assume that the dynamics $f$, the reward function $\rho$ and the policy $h$ are Lipschitz continuous:

**Assumption B.4.2 (Lipschitz continuity of $f$, $\rho$ and $h$)**
$\exists L_f, L_\rho, L_h \in \mathbb{R} : \forall (x, x') \in X^2, \forall (u, u') \in U^2, \forall t \in \{0, \dots, T-1\}$,

$$\begin{aligned} \|f(x,u) - f(x',u')\|_{\mathcal{X}} &\leq L_f\big(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}\big) \ , && \text{(B.55)} \\ |\rho(x,u) - \rho(x',u')| &\leq L_\rho\big(\|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}\big) \ , && \text{(B.56)} \\ \|h(t,x) - h(t,x')\|_{\mathcal{U}} &\leq L_h\|x - x'\|_{\mathcal{X}} \ . && \text{(B.57)} \end{aligned}$$

The dynamics and the reward function are unknown, but we assume that three constants $L_f$, $L_\rho$, $L_h$ satisfying the above-written inequalities are known. Under those assumptions, we want to compute bounds on the $T-$stage return of a given policy $h$.

### B.4.1 Kernel-based policy evaluation

Given a state $x \in \mathcal{X}$, we also introduce the $(T-t)-$stage return of a policy $h$ when starting from $x \in \mathcal{X}$ as follows:

**Definition B.4.3 (**$(T-t)-$**stage return of a policy** $h$**)**
*Let $x \in \mathcal{X}$. For $t' \in \{t, \dots, T-1\}$, we denote by $x_{t'+1}$ the state*

$$x_{t'+1} = f(x_{t'}, u_{t'}) \tag{B.58}$$

*with*

$$u_{t'} = h(t', x_{t'}) \tag{B.59}$$

*and $x_t = x$. The $(T-t)-$stage return of the policy $h$ when starting from $x$ is defined as follows:*

$$J^h_{T-t}(x) = \sum_{t'=t}^{T-1} \rho(x_{t'}, u_{t'}) \ .$$

The stage return of the policy $h$ is thus given by

$$J^h(x_0) = J^h_T(x_0). \tag{B.60}$$

The sequence of functions $\big(J^h_{T-t}(.)\big)_{t=0}^{T-1}$ is approximated using kernels ([1]) by a sequence $\big(\tilde{J}^h_{T-t}(.)\big)_{t=0}^{T-1}$ computed as follows

$$\forall x \in \mathcal{X}, \tilde{J}^h_0(x) = J^h_0(x) = 0 \ , \tag{B.61}$$

and, $\forall x \in \mathcal{X}, \forall t \in \{0, \dots, T-1\}$,

$$\tilde{J}^h_{T-t}(x) = \sum_{l=1}^{n} k_l(x, h(t,x)) \Big( r^l + \tilde{J}^h_{T-t-1}(y^l) \Big) \ , \tag{B.62}$$

where $k_l : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is defined as follows:

$$k_l(x,u) = \frac{\Phi\left(\frac{\|x-x^l\|_{\mathcal{X}} + \|u-u^l\|_{\mathcal{U}}}{b}\right)}{\sum_{i=1}^{n} \Phi\left(\frac{\|x-x^i\|_{\mathcal{X}} + \|u-u^i\|_{\mathcal{U}})}{b}\right)} \ , \tag{B.63}$$

where $b > 0$ is the bandwidth parameter and $\Phi : \mathbb{R}^+ \to \mathbb{R}^+$ is a univariate non-negative “mother kernel” function. We also assume that

$$\forall x > 1, \Phi(x) = 0 \ , \tag{B.64}$$

and we suppose that each function $k_l$ is Lipschitz continuous.

**Assumption B.4.4 (Lipschitz continuity of $\{k_l\}_{l=1}^{n}$)**
$\forall l \in \{1, \dots, n\}, \exists L_{k_l} > 0 :$

$$\begin{aligned} &\forall (x', x'', u', u'') \in \mathcal{X}^2 \times \mathcal{U}^2, \\ &|k_l(x', u') - k_l(x'', u'')| \leq L_{k_l} \left( \|x' - x''\|_{\mathcal{X}} + \|u' - u''\|_{\mathcal{U}} \right) \ . \end{aligned} \tag{B.65}$$

We define $L_k$ such that

$$L_k = \max_{l \in \{1, \dots, n\}} L_{k_l} \ . \tag{B.66}$$

The kernel-based estimator KBE, denoted by $\mathfrak{K}^h(x_0)$, is defined as follows:

**Definition B.4.5 (Kernel-based estimator)**
$\forall x_0 \in \mathcal{X}$,

$$\mathfrak{K}^h(x_0) = \tilde{J}^h_T(x_0) \ . \tag{B.67}$$

We introduce the family of kernel operators $\left(K^h_{T-t}\right)_{t=0}^{T-1}$ such that

**Definition B.4.6 (Continuous action space kernel operators)**
*Let $g : \mathcal{X} \to \mathbb{R}$. $\forall t \in \{0, \dots, T-1\}, \forall x \in \mathcal{X}$,*

$$\left(K^h_{T-t} \circ g\right)(x) = \sum_{l=1}^{n} k_l(x, h(t,x)) \left( r^l + g(y^l) \right) \ . \tag{B.68}$$

One has

$$\tilde{J}^h_{T-t}(x) \quad = \quad \left(K^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x)\,. \tag{B.69}$$

We also introduce the family of finite-horizon Bellman operators $\left(B^h_{T-t}\right)_{t=0}^{T-1}$ as follows:

**Definition B.4.7 (Continuous Bellman operator)**
*Let* $g : \mathcal{X} \to \mathbb{R}$. $\forall t \in \{1, \ldots, T\}, \forall x \in \mathcal{X}$,

$$\left(B^h_{T-t} \circ g\right)(x) = \rho(x, h(t,x)) + g(f(x, h(t,x)))\,. \tag{B.70}$$

One has

$$J^h_{T-t}(x) \quad = \quad \left(B^h_{T-t} \circ J^h_{T-t-1}\right)(x)\,. \tag{B.71}$$

We propose a second lemma that bounds the distance between the two operators $K^h_{T-t}$ and $B^h_{T-t}$ when applied to the approximated $(T-t-1)-$ return $\tilde{J}^h_{T-t-1}$.

**Lemma B.4.8**
$\forall t \in \{1, \ldots, T-1\}, \forall x \in \mathcal{X}$,

$$\left|\left(K^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x) - \left(B^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x)\right| \le C_{T-t} b \tag{B.72}$$

*with*

$$C_{T-t} = L_\rho + L_k L_f A_\rho (1 + L_h)(T - t - 1)\,. \tag{B.73}$$

**Proof** Let $x \in \mathcal{X}$ .

- Let $t \in \{0, \ldots, T-2\}$. Since

$$\sum_{l=1}^{n} \mathbb{I}_{\{u^l = h(t,x)\}} k_l(x) = 1, \tag{B.74}$$

one can write

$$\left|\left(K^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x) - \left(B^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x)\right|$$

$$= \Bigg| \sum_{l=1}^{n} k_l(x, h(t,x)) \Big[ r^l - \rho(x, h(t,x))$$

$$+ \tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, h(t,x))) \Big] \Bigg| \tag{B.75}$$

$$\leq L_\rho \sum_{l=1}^{n} k_l(x, h(t,x)) \left( \|x^l - x\|_{\mathcal{X}} + \|u^l - h(t,x)\|_{\mathcal{U}} \right)$$

$$+ \sum_{l=1}^{n} \left| k_l(x, h(t,x)) \left( \tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, h(t,x))) \right) \right| \tag{B.76}$$

Since

$$\forall z > 1, \Phi(z) = 0, \tag{B.77}$$

one has

$$\left( \|x^l - x\|_{\mathcal{X}} + \|u^l - h(t,x)\|_{\mathcal{U}} \right) \geq b \implies k_l(x, h(t,x)) = 0 \; . \tag{B.78}$$

This gives

$$L_\rho \sum_{l=1}^{n} k_l(x, h(t,x)) \left( \|x^l - x\|_{\mathcal{X}} + \|u^l - h(t,x)\|_{\mathcal{U}} \right) \leq L_\rho b \; . \tag{B.79}$$

On the other hand, one has

$$\tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, h(t,x))) = \sum_{j=1}^{n} \Big[ k_j(y^l, h(t+1, y^l))$$

$$- k_j(f(x, h(t,x)), h(t+1, f(x, h(t,x)))) \Big] (r^j + \tilde{J}^h_{T-t-2}(y^j)) \tag{B.80}$$

Since the reward function $\rho$ is bounded by $A_\rho$, one can write

$$\left| (r^j + \tilde{J}^h_{T-t-2}(y^j)) \right| \leq (T - t - 1) A_\rho \; . \tag{B.81}$$

and according to the Lipschitz continuity of $k_j$,$f$ and $h$, one has

$$\begin{aligned}
&\left|k_j(y^l, h(t+1, y^l)) - k_j(f(x, u_t), h(t+1, f(x, h(t, x))))\right| \\
&\leq L_{k_j}\left(\|y^l - f(x, h(t, x))\|_{\mathcal{X}} + \|h(t+1, y^l) - h(t+1, f(x, h(t, x)))\|_{\mathcal{U}}\right) \qquad \text{(B.82)} \\
&\leq L_k\left(\|y^l - f(x, h(t, x))\|_{\mathcal{X}} + \|h(t+1, y^l) - h(t+1, f(x, h(t, x)))\|_{\mathcal{U}}\right) \qquad \text{(B.83)} \\
&\leq L_k L_f (1 + L_h)\left(\|x^l - x\|_{\mathcal{X}} + \|u^l - h(t, x)\|_{\mathcal{U}}\right) \ . \qquad \text{(B.84)}
\end{aligned}$$

Equations (B.80), (B.81) and (B.84) allow to write

$$\begin{aligned}
&\left|\left(\tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, u_t))\right)\right| \\
&\leq L_k L_f (1 + L_h)(T - t - 1) A_\rho \left(\|x^l - x\|_{\mathcal{X}} + \|u^l - h(t, x)\|_{\mathcal{U}}\right)
\end{aligned} \qquad \text{(B.85)}$$

Equations (B.78) and (B.85) give

$$\begin{aligned}
&\left|\left(\tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, h(t, x)))\right)\right| \\
&\qquad \leq L_k L_f (1 + L_h)(T - t - 1) A_\rho b
\end{aligned} \qquad \text{(B.86)}$$

and since

$$\sum_{l=1}^{n} k_l(x, h(t, x)) = 1 \ , \qquad \text{(B.87)}$$

$$\begin{aligned}
&\sum_{l=1}^{n}\left|k_l(x, h(t, x))(\tilde{J}^h_{T-t-1}(y^l) - \tilde{J}^h_{T-t-1}(f(x, h(t, x))))\right| \\
&\leq L_k L_f (1 + L_h) b (T - t - 1) A_\rho
\end{aligned} \qquad \text{(B.88)}$$

Using Equations (B.76), (B.79) and (B.88), we can finally write
$\forall (x, t) \in \mathcal{X} \times \{0, \ldots, T-2\}$,

$$\begin{aligned}
&\left|\left(K^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x) - \left(B^h_{T-t} \circ \tilde{J}^h_{T-t-1}\right)(x)\right| \\
&\leq (L_\rho + L_k L_f (1 + L_h)(T - t - 1) A_\rho) b
\end{aligned} \qquad \text{(B.89)}$$

This proves the lemma for $t \in \{0, \ldots, T-2\}$.

- Let $t = T - 1$. One has

$$\begin{aligned}
&\left|\left(K_1^h \circ \tilde{J}_0^h\right)(x) - \left(B_1^h \circ \tilde{J}_0^h\right)(x)\right| \\
&\leq \sum_{l=1}^{n} k_l(x, h(T-1, x)) \left|r^l - \rho(x, h(T-1, x))\right| && \text{(B.90)} \\
&\leq \sum_{l=1}^{n} k_l(x, h(T-1, x)) L_\rho \left(\|x - x^l\| + \|h(T-1, x) - u^l\|\right) \\
& && \text{(B.91)} \\
&\leq L_\rho b\,, && \text{(B.92)}
\end{aligned}$$

since

$$\left(\|x - x^l\| + \|h(T-1, x) - u^l\|_{\mathcal{U}}\right) \geq b \implies k_l(x, h(T-1, x)) = 0 \quad \text{(B.93)}$$

and

$$\sum_{l=1}^{n} k_l(x, h(T-1, x)) = 1. \quad \text{(B.94)}$$

This shows that Equation (B.89) is also valid for $t = T - 1$, and ends the proof.

According to the previous lemma, we have the following theorem.

**Theorem B.4.9 (Bounds on the actual return of $h$)**
*Let $x_0 \in \mathcal{X}$ be a given initial state. Then,*

$$\left|\mathfrak{K}^h(x_0) - J^h(x_0)\right| \leq \beta b\,, \quad \text{(B.95)}$$

*with*

$$\beta = \sum_{t=1}^{T} C_{T-t}\,. \quad \text{(B.96)}$$

**Proof** We use the notation $x_{t+1} = f(x_t, u_t)$ with $u_t = h(t, x_t)$. One has

$$\begin{aligned}
J_T^h(x_0) - \tilde{J}_T^h(x_0) &= B_{T-1}^h \circ J_{T-1}^h(x_0) - K_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) && \text{(B.97)} \\
&= B_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) - K_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) && \text{(B.98)} \\
&+ B_{T-1}^h \circ J_{T-1}^h(x_0) - B_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) \\
&= B_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) - K_{T-1}^h \circ \tilde{J}_{T-1}^h(x_0) \\
&+ J_{T-1}^h(x_1) - \tilde{J}_{T-1}^h(x_1) && \text{(B.99)}
\end{aligned}$$

Using the recursive form of Equation (B.99), one has

$$J^h(x_0) - \mathfrak{K}^h(x_0) = J^h_T(x_0) - \tilde{J}^h_T(x_0) \tag{B.100}$$

$$= \sum_{t=0}^{T-1} B^h_{T-t} \circ \tilde{J}^h_{T-t-1}(x_t) - K^h_{T-t} \circ \tilde{J}^h_{T-t-1}(x_t) \tag{B.101}$$

Then, according to Lemma 1, we can write

$$\left| J^h_T(x_0) - \mathfrak{K}^h(x_0) \right| \leq \sum_{t=0}^{T-1} C_{T-t} b \, , \tag{B.102}$$

which ends the proof.

# Bibliography

[1] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3):161–178, 2002.

# Appendix C

# Voronoi model learning for batch mode reinforcement learning

*We consider deterministic optimal control problems with continuous state spaces where the information on the system dynamics and the reward function is constrained to a set of system transitions. Each system transition gathers a state, the action taken while being in this state, the immediate reward observed and the next state reached. In such a context, we propose a new model learning–type reinforcement learning (RL) algorithm in batch mode, finite-time and deterministic setting. The algorithm, named Voronoi reinforcement learning (VRL), approximates from a sample of system transitions the system dynamics and the reward function of the optimal control problem using piecewise constant functions on a Voronoi–like partition of the state-action space.*

This appendix reports on a theoretical analysis of the Voronoi RL algorithm first introduced in [2] and reported in Chapter 5.

In this appendix, we consider:

- a deterministic framework,
- a continuous state space and a finite action space.

## C.1 Problem statement

We consider a discrete-time system whose dynamics over $T$ stages is described by a time-invariant equation

$$x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \ldots, T-1, \tag{C.1}$$

where for all $t \in \{0, \ldots, T-1\}$, the state $x_t$ is an element of the bounded normed state space $\mathcal{X} \subset \mathbb{R}^{d_{\mathcal{X}}}$ and $u_t$ is an element of a finite action space $\mathcal{U} = \left\{a^1, \ldots, a^m\right\}$ with $m \in \mathbb{N}_0$. $x_0 \in \mathcal{X}$ is the initial state of the system. $T \in \mathbb{N}_0$ denotes the finite optimization horizon. An instantaneous reward

$$r_t = \rho(x_t, u_t) \in \mathbb{R} \tag{C.2}$$

is associated with the action $u_t \in \mathcal{U}$ taken while being in state $x_t \in \mathcal{X}$. We assume that the initial state of the system $x_0 \in \mathcal{X}$ is fixed. For a given open-loop sequence of actions $\mathbf{u} = (u_0, \ldots, u_{T-1}) \in \mathcal{U}^T$, we denote by $J^{\mathbf{u}}(x_0)$ the $T-$stage return of the sequence of actions $\mathbf{u}$ when starting from $x_0$, defined as follows:

**Definition C.1.1 ($T-$stage return)**
$\forall \mathbf{u} \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$J^{\mathbf{u}}(x_0) = \sum_{t=0}^{T-1} \rho(x_t, u_t) \tag{C.3}$$

*with*

$$x_{t+1} = f(x_t, u_t), \forall t \in \{0, \ldots, T-1\} . \tag{C.4}$$

We denote by $J^*(x_0)$ the maximal value:

**Definition C.1.2 (Maximal return)**
$\forall x_0 \in \mathcal{X}$,

$$J^*(x_0) = \max_{\mathbf{u} \in \mathcal{U}^T} J^{\mathbf{u}}(x_0) . \tag{C.5}$$

Considering the fixed initial state $x_0$, an optimal sequence of actions $\mathbf{u}^*(x_0)$ is a sequence for which

$$J^{\mathbf{u}^*(x_0)}(x_0) = J^*(x_0) . \tag{C.6}$$

In this appendix, we assume that the functions $f$ and $\rho$ are unknown. Instead, we know a sample of $n$ system transitions

$$\mathcal{F}_n = \left\{ \left( x^l, u^l, r^l, y^l \right) \right\}_{l=1}^{n} \tag{C.7}$$

where for all $l \in \{1, \ldots, n\}$

$$r^l = \rho(x^l, u^l) \tag{C.8}$$

and

$$y^l = f(x^l, u^l) \; . \tag{C.9}$$

The problem addressed in this appendix is to compute from the sample $\mathcal{F}_n$, an open-loop sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$ such that $\tilde{J}_{\mathcal{F}_n}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0)$ is as close as possible to $\tilde{J}^*_{\mathcal{F}_n}(x_0)$.

## C.2 Model learning–type RL

Model learning–type reinforcement learning aims at solving optimal control problems by approximating the unknown functions $f$ and $\rho$ and solving the so approximated optimal control problem instead of the unknown actual optimal control problem. The values $y^l$ (resp. $r^l$) of the function $f$ (resp. $\rho$) in the state-action points $(x^l, u^l)\quad l = 1 \ldots n$ are used to learn a function $\tilde{f}_{\mathcal{F}_n}$ (resp. $\tilde{\rho}_{\mathcal{F}_n}$) over the whole space $\mathcal{X} \times \mathcal{U}$. The approximated optimal control problem defined by the functions $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$ is solved and its solution is kept as an approximation of the solution of the optimal control problem defined by the actual functions $f$ and $\rho$.

Given a sequence of actions $\mathbf{u} \in \mathcal{U}^T$ and a model learning–type reinforcement learning algorithm, we denote by $\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0)$ the approximated $T-$stage return of the sequence of actions $\mathbf{u}$, i.e. the $T-$stage return when considering the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$:

**Definition C.2.1 (Approximated $T-$stage return)**
$\forall \mathbf{u} \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$

$$\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) = \sum_{t=0}^{T-1} \tilde{\rho}_{\mathcal{F}_n}\left(\tilde{x}_t, u_t\right) \tag{C.10}$$

*with*

$$\tilde{x}_{t+1} = \tilde{f}_{\mathcal{F}_n}\left(\tilde{x}_t, u_t\right), \; \forall t \in \{0, \ldots, T-1\} \tag{C.11}$$

*and* $\tilde{x}_0 = x_0$.

We denote by $\tilde{J}^{*}_{\mathcal{F}_n}(x_0)$ the maximal approximated $T-$stage return when starting from the initial state $x_0 \in \mathcal{X}$ according to the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$:

**Definition C.2.2 (Maximal approximated $T-$stage return)**
$\forall x_0 \in \mathcal{X}$,

$$\tilde{J}^{*}_{\mathcal{F}_n}(x_0) = \max_{\mathbf{u} \in \mathcal{U}^T} \tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) \; . \tag{C.12}$$

Using these notations, model learning–type RL algorithms aim at computing a sequence of actions $\tilde{\mathbf{u}}^{*}_{\mathcal{F}_\mathbf{n}}(x_0) \in \mathcal{U}^T$ such that $\tilde{J}^{\tilde{\mathbf{u}}^{*}_{\mathcal{F}_\mathbf{n}}(x_0)}_{\mathcal{F}_n}(x_0)$ is as close as possible (and ideally equal to) to $\tilde{J}^{*}_{\mathcal{F}_n}(x_0)$. These techniques implicitly assume that an optimal policy for the learned model also leads to high returns on the real problem.

# C.3 The Voronoi Reinforcement Learning algorithm

This algorithm approximates the reward function $\rho$ and the system dynamics $f$ using piecewise constant approximations on a Voronoi–like [1] partition of the state-action space (which is equivalent to a nearest-neighbour approximation) and will be referred to by the VRL algorithm. Given an initial state $x_0 \in \mathcal{X}$, the VRL algorithm computes an open-loop sequence of actions which corresponds to an “optimal navigation” among the Voronoi cells.

Before fully describing this algorithm, we first assume that all the state-action pairs $\left\{(x^l, u^l)\right\}_{l=1}^{n}$ given by the sample of transitions $\mathcal{F}_n$ are unique, i.e.

$$\forall l, l' \in \{1, \ldots, n\}, (x^l, u^l) = (x^{l'}, u^{l'}) \implies l = l' \; . \tag{C.13}$$

We also assume that each action of the action space $\mathcal{U}$ has been tried at least once, i.e.,

$$\forall u \in \mathcal{U}, \exists l \in \{1, \ldots, n\}, u^l = u \; . \tag{C.14}$$

The model is based on the creation of $n$ Voronoi cells $\left\{V^l\right\}_{l=1}^{n}$ which define a partition of size $n$ of the state-action space. The Voronoi cell $V^l$ associated to the element $(x^l, u^l)$ of $\mathcal{F}_n$ is defined as the set of state-action pairs $(x, u) \in \mathcal{X} \times \mathcal{U}$ that satisfy:

$$(i) \quad u = u^l \;\; , \tag{C.15}$$

$$(ii) \quad l \in \arg\min_{l' : u^{l'} = u} \left\{ \|x - x^{l'}\|_{\mathcal{X}} \right\} \; , \tag{C.16}$$

$$(iii) \quad l = \min_{l'} \left\{ l' \in \arg\min_{l' : u^{l'} = u} \left\{ \|x - x^{l'}\|_{\mathcal{X}} \right\} \right\} . \tag{C.17}$$

One can verify that $\left\{V^l\right\}_{l=1}^n$ is indeed a partition of the state-action space $\mathcal{X} \times \mathcal{U}$ since every state-action $(x,u) \in \mathcal{X} \times \mathcal{U}$ belongs to one and only one Voronoi cell.

The function $f$ (resp. $\rho$) is approximated by a piecewise constant function $\tilde{f}_{\mathcal{F}_n}$ (resp. $\tilde{\rho}_{\mathcal{F}_n}$) defined as follows:

$$\begin{aligned} \forall l \in \{1,\ldots,n\}, \forall (x,u) \in V^l, \quad \tilde{f}_{\mathcal{F}_n}(x,u) &= y^l, & \text{(C.18)} \\ \tilde{\rho}_{\mathcal{F}_n}(x,u) &= r^l\,. & \text{(C.19)} \end{aligned}$$

### C.3.1 Open-loop formulation

Using the approximations $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$, we define a sequence of approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$ as follows :

**Definition C.3.1 (Approximated optimal state-action value functions)**
$\forall t \in \{0,\ldots,T-1\}, \forall (x,u) \in \mathcal{X} \times \mathcal{U}$ ,

$$\begin{aligned} \tilde{Q}^*_{T-t}(x,u) &= \tilde{\rho}_{\mathcal{F}_n}(x,u) \\ &+ \arg\max_{u' \in \mathcal{U}} \tilde{Q}^*_{T-t-1}\left(\tilde{f}_{\mathcal{F}_n}(x,u), u'\right)\,, \end{aligned} \tag{C.20}$$

*with*

$$Q^*_1(x,u) = \tilde{\rho}_{\mathcal{F}_n}(x,u), \quad \forall (x,u) \in \mathcal{X} \times \mathcal{U}. \tag{C.21}$$

Using the sequence of approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$, one can infer an open-loop sequence of actions

$$\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = (\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0)) \in \mathcal{U}^T \tag{C.22}$$

which is an exact solution of the approximated optimal control problem, i.e. which is such that

$$\tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) = \tilde{J}^*_{\mathcal{F}_n}(x_0) \tag{C.23}$$

as follows:

$$\tilde{u}^*_{\mathcal{F}_n,0}(x_0) \quad \in \quad \arg\max_{u' \in \mathcal{U}} \tilde{Q}^*_T(\tilde{x}^*_0, u')\,, \tag{C.24}$$

and, $\forall t \in \{0,\ldots,T-2\}$ ,

$$\tilde{u}^*_{\mathcal{F}_n,t+1}(x_0) \quad \in \quad \arg\max_{u' \in \mathcal{U}} \tilde{Q}^*_{T-(t+1)}\left(\tilde{f}_{\mathcal{F}_n}\left(\tilde{x}^*_t, \tilde{u}^*_{\mathcal{F}_n,t}(x_0)\right), u'\right) \tag{C.25}$$

where

$$\tilde{x}^*_{t+1} = \tilde{f}_{\mathcal{F}_n}(\tilde{x}^*_t, \tilde{u}^*_{\mathcal{F}_n,t}(x_0)), \forall t \in \{0, \ldots, T-1\}. \tag{C.26}$$

and $\tilde{x}^*_0 = x_0$.

All the approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$ are piecewise constant over each Voronoi cell, a property that can be exploited for computing them easily as it is shown in Figure 7. The VRL algorithm has linear complexity with respect to the cardinality $n$ of the sample of system transitions $\mathcal{F}_n$, the optimization horizon $T$ and the cardinality $m$ of the action space $\mathcal{U}$.

### C.3.2 Closed-loop formulation

Using the sequence of approximated optimal state-action value functions $\left(\tilde{Q}^*_{T-t}\right)_{t=0}^{T-1}$, one can infer a closed-loop sequence of actions

$$\tilde{\mathbf{v}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = (\tilde{v}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{v}^*_{\mathcal{F}_n,T-1}(x_0)) \in \mathcal{U}^T \tag{C.27}$$

by replacing the approximated system dynamics $\tilde{f}_{\mathcal{F}_n}$ with the true system dynamics in Equations (C.24), (C.25) and (C.26) as follows:

$$\tilde{v}^*_{\mathcal{F}_n,0}(x_0) \quad = \quad \arg\max_{v' \in \mathcal{U}} \tilde{Q}^*_T(\tilde{x}^*_0, v') \ ,$$

and, $\forall t \in \{0, \ldots, T-2\}$ ,

$$\tilde{v}^*_{\mathcal{F}_n,t+1}(x_0) \quad = \quad \arg\max_{v' \in \mathcal{U}} \tilde{Q}^*_{T-(t+1)} \left(f\left(\tilde{x}^*_t, \tilde{v}^*_{\mathcal{F}_n,t}(x_0)\right), v'\right)$$

where

$$\tilde{x}^*_{t+1} = f(\tilde{x}^*_t, \tilde{v}^*_{t,\mathcal{F}_n}(x_0)), \forall t \in \{0, \ldots, T-1\}. \tag{C.28}$$

and $\tilde{x}^*_0 = x_0$.

## C.4 Theoretical analysis of the VRL algorithm

We propose to analyze the convergence of the Voronoi RL algorithm when the functions $f$ and $\rho$ are Lipschitz continuous and the sparsity of the sample of transitions decreases towards zero. We first assume the Lipschitz continuity of the functions $f$ and $\rho$ :

**Algorithm 7** The Voronoi Reinforcement Learning (VRL) algorithm. $Q_{T-t,l}$ is the value taken by the function $\tilde{Q}^*_{T-t}$ in the Voronoi cell $V^l$.

```
**Inputs:** an initial state $x_0 \in \mathcal{X}$, a sample of transitions $\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^{n}$ ;
**Output:** a sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$ and $\tilde{J}^*_{\mathcal{F}_n}(x_0)$ ;
**Initialization:**
Create a $n \times m$ matrix $V$ such that $V(i,j)$ contains the index of the Voronoi cell (VC) where $\left(\tilde{f}_{\mathcal{F}_n}(x^i, u^i), a^j\right)$ lies ;
**for** $i = 1$ **to** $n$ **do**
  $Q_{1,i} \leftarrow r^i$ ;
**end for**
**Algorithm:**
**for** $t = T-2$ **to** $0$ **do**
  **for** $i = 1$ **to** $n$ **do**
    $l \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max} \left\{Q_{T-t-1,V(i,l')}\right\}$ ;
    $Q_{T-t,i} \leftarrow r^i + Q_{T-t-1,V(i,l)}$ ;
  **end for**
**end for**
$l \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max} Q_{T,i'}$ where $i'$ denotes the index of the VC where $(x_0, a^{l'})$ lies ;
$l_0^* \leftarrow$ index of the VC where $(x_0, a^l)$ lies ;
$\tilde{J}^*_{\mathcal{F}_n}(x_0) \leftarrow Q_{T,l_0^*}$ ;
$i \leftarrow l_0^*$ ;
$\tilde{u}^*_{\mathcal{F}_n,0}(x_0) \leftarrow u^{l_0^*}$ ;
**for** $t = 0$ **to** $T-2$ **do**
  $l^*_{t+1} \leftarrow \underset{l' \in \{1,\ldots,m\}}{\arg\max} \left\{Q_{T-t-1,V(i,l')}\right\}$ ;
  $\tilde{u}^*_{\mathcal{F}_n,t+1}(x_0) \leftarrow a^{l^*_{t+1}}$ ;
  $i \leftarrow V(i, l^*_{t+1})$ ;
**end for**
**Return:** $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0) = (\tilde{u}^*_{\mathcal{F}_n,0}(x_0), \ldots, \tilde{u}^*_{\mathcal{F}_n,T-1}(x_0))$ and $\tilde{J}^*_{\mathcal{F}_n}(x_0)$.
```

**Assumption C.4.1 (Lipschitz continuity of $f$ and $\rho$)**

$$\begin{aligned} \exists L_f, L_\rho > 0 : \forall u \in \mathcal{U}, \forall x, x' \in \mathcal{X}, & \\ \|f(x,u) - f(x',u)\|_{\mathcal{X}} &\leq L_f \|x - x'\|_{\mathcal{X}} \,, \quad (C.29) \\ |\rho(x,u) - \rho(x',u)| &\leq L_\rho \|x - x'\|_{\mathcal{X}} \,. \quad (C.30) \end{aligned}$$

For each action $u \in \mathcal{U}$, we denote by $f_u$ (resp. $\rho_u$) the restrictions of the function $f$ (resp. $\rho$) to the action $u$:

$$\begin{aligned} \forall u \in \mathcal{U}, \forall x \in \mathcal{X}, f_u(x) &= f(x,u) \,, \quad (C.31) \\ \rho_u(x) &= \rho(x,u) \,. \quad (C.32) \end{aligned}$$

All the functions $\{f_u\}_{u \in \mathcal{U}}$ and $\{\rho_u\}_{u \in \mathcal{U}}$ are thus also Lipschitz continuous. Given a sample of system transitions $\mathcal{F}_n$, and given an action $u \in \mathcal{U}$, we also introduce the restrictions of the function $\tilde{f}_{\mathcal{F}_n,u}$ and $\tilde{\rho}_{\mathcal{F}_n,u}$ as follows:

$$\begin{aligned} \forall u \in \mathcal{U}, \forall x \in \mathcal{X}, \tilde{f}_{\mathcal{F}_n,u}(x) &= \tilde{f}_{\mathcal{F}_n}(x,u) \,, \quad (C.33) \\ \tilde{\rho}_{\mathcal{F}_n,u}(x) &= \tilde{\rho}_{\mathcal{F}_n}(x,u) \,. \quad (C.34) \end{aligned}$$

Given a Voronoi cell $V^l \quad l \in \{1, \ldots, n\}$, we denote by $\Delta^l_{\mathcal{F}_n}$ the radius of the Voronoi–like cell $V^l$ defined as follows :

**Definition C.4.2 (Radius of Voronoi cells)**
$\forall l \in \{1, \ldots, n\}$,

$$\Delta^l_{\mathcal{F}_n} = \sup_{(x,u^l) \in V^l} \left\| x - x^l \right\|_{\mathcal{X}} \,. \quad (C.35)$$

We then introduce the sparsity of the sample of transitions $\mathcal{F}_n$, denoted by $\alpha_{\mathcal{F}_n}$:

**Definition C.4.3 (Sparsity of $\mathcal{F}_n$)**

$$\alpha_{\mathcal{F}_n} = \max_{l \in \{1,\ldots,n\}} \Delta^l_{\mathcal{F}_n} \,. \quad (C.36)$$

The sparsity of the sample of system transitions $\mathcal{F}_n$ can be seen, in a sense, as the "maximal radius" of all Voronoi cells. We suppose that a sequence of sample of transitions $(\mathcal{F}_n)_{n=n_0}^{\infty}$ (with $n_0 \geq m$) is known, and we assume that the corresponding sequence of sparsities $(\alpha_{\mathcal{F}_n})_{n=n_0}^{\infty}$ converges towards zero.

### C.4.1 Consistency of the open-loop VRL algorithm

To each sample of transitions $\mathcal{F}_n$ are associated two piecewise constant approximated functions $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$, and a sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$ computed using the VRL algorithm which is a solution of the approximated optimal control problem defined by the functions $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$. We have the following theorem:

**Theorem C.4.4 (Consistency of the Voronoi RL algorithm)**
$\forall x_0 \in \mathcal{X}$,

$$\lim_{n\to\infty} J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)}(x_0) = J^*(x_0)\,. \tag{C.37}$$

Before giving the proof of Theorem C.4.4, let us first introduce a few lemmas.

**Lemma C.4.5 (Uniform convergence of $\tilde{f}_{\mathcal{F}_n,u}$ and $\tilde{\rho}_{\mathcal{F}_n,u}$ towards $f_u$ and $\rho_u$)**

$$\forall u \in \mathcal{U}, \qquad \lim_{n\to\infty} \sup_{x\in\mathcal{X}} \left\| f_u(x) - \tilde{f}_{\mathcal{F}_n,u}(x) \right\|_{\mathcal{X}} = 0\,, \tag{C.38}$$

$$\lim_{n\to\infty} \sup_{x\in\mathcal{X}} \left|\rho_u(x) - \tilde{\rho}_{\mathcal{F}_n,u}(x)\right| = 0\,. \tag{C.39}$$

**Proof.** Let $u \in \mathcal{U}$, let $x \in \mathcal{X}$, and let $V^l$ be the Voronoi cell where $(x, u)$ lies (then, $u = u^l$). One has

$$\tilde{f}_{\mathcal{F}_n,u}(x) = y^l\,, \tag{C.40}$$

$$\tilde{\rho}_{\mathcal{F}_n,u}(x) = r^l\,. \tag{C.41}$$

which implies that

$$\left\| \tilde{f}_{\mathcal{F}_n,u}(x) - f_u(x^l) \right\|_{\mathcal{X}} = 0\,, \tag{C.42}$$

$$\left| \tilde{\rho}_{\mathcal{F}_n,u}(x) - \rho_u(x^l) \right| = 0\,. \tag{C.43}$$

Then,

$$\begin{aligned}
\left\| f_u(x) - \tilde{f}_{\mathcal{F}_n,u}(x) \right\|_{\mathcal{X}} &\le \left\| f_u(x) - f_u(x^l) \right\|_{\mathcal{X}} \\
&+ \left\| f_u(x^l) - \tilde{f}_{\mathcal{F}_n,u}(x) \right\|_{\mathcal{X}} && \text{(C.44)}\\
&\le L_f \left\| x - x^l \right\|_{\mathcal{X}} + 0 && \text{(C.45)}\\
&\le L_f \Delta^l_{\mathcal{F}_n} && \text{(C.46)}\\
&\le L_f \alpha_{\mathcal{F}_n}\,, && \text{(C.47)}
\end{aligned}$$

and similarly for the functions $\rho_u$ and $\tilde{\rho}_{\mathcal{F}_n,u}$,

$$|\rho_u(x) - \tilde{\rho}_{\mathcal{F}_n,u}(x)| \leq L_\rho \alpha_{\mathcal{F}_n} \; . \tag{C.48}$$

This ends the proof since $\alpha_{\mathcal{F}_n} \to 0$. ■

**Lemma C.4.6 (Uniform convergence of the sum of functions)**
*Let $(h_n : \mathcal{X} \to \mathbb{R})_{n\in\mathbb{N}}$ (resp. $(h'_n : \mathcal{X} \to \mathbb{R})_{n\in\mathbb{N}}$) be a sequence of functions that uniformly converges towards $h : \mathcal{X} \to \mathbb{R}$ (resp. $h' : \mathcal{X} \to \mathbb{R}$). Then, the sequence of functions $((h_n + h'_n) : \mathcal{X} \to \mathbb{R})_{n\in\mathbb{N}}$ uniformly converges towards the function $(h+h')$.*

**Proof.** Let $\epsilon > 0$. Since $(h_n)_{n\in\mathbb{N}}$ uniformly converges towards $h$, there exists $n_h \in \mathbb{N}$ such that

$$\forall n \geq n_h, \forall x \in \mathcal{X}, |h_n(x) - h(x)| \leq \frac{\epsilon}{2} \; . \tag{C.49}$$

Since $(h'_n)_{n\in\mathbb{N}}$ uniformly converges towards $h'$, there exists $n_{h'} \in \mathbb{N}$ such that

$$\forall n \geq n_{h'}, \forall x \in \mathcal{X}, |h'_n(x) - h'(x)| \leq \frac{\epsilon}{2} \; . \tag{C.50}$$

We denote by $n_{\max} = \max(n_h, n_{h'})$. One has
$\forall n \geq n_{\max}, \forall x \in \mathcal{X}$,

$$\begin{aligned}
|(h_n(x) - h'_n(x)) - (h(x) + h'(x))| &\leq |h_n(x) - h(x)| + |h'_n(x) - h'(x)| && \text{(C.51)} \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} && \text{(C.52)} \\
&\leq \epsilon \, , && \text{(C.53)}
\end{aligned}$$

which ends the proof. ■

**Lemma C.4.7 (Uniform convergence of composed functions)**

- *Let $(g_n : \mathcal{X} \to \mathcal{X})_{n\in\mathbb{N}}$ be a sequence of functions that uniformly converges towards $g : \mathcal{X} \to \mathcal{X}$;*
- *Let $(g'_n : \mathcal{X} \to \mathcal{X})_{n\in\mathbb{N}}$ be a sequence of functions that uniformly converges towards $g' : \mathcal{X} \to \mathcal{X}$. Let us assume that $g'$ is $L_{g'}-$Lipschitzian;*

- *Let $(h_n : \mathcal{X} \to \mathbb{R})_{n\in\mathbb{N}}$ be a sequence of functions that uniformly converges towards $h : \mathcal{X} \to \mathbb{R}$. Let us assume that $h$ is $L_h-$Lipschitzian.*

*Then,*

- *The sequence of functions $(g'_n \circ g_n)_{n\in\mathbb{N}}$ uniformly converges towards the function $g' \circ g$.*
- *The sequence of functions $(h_n \circ g_n)_{n\in\mathbb{N}}$ uniformly converges towards the function $h \circ g$,*

*where the notation $h_n \circ g_n$ (resp. $g'_n \circ g$, $h \circ g$ and $g' \circ g$) denotes the mapping $x \to h_n\left(g_n(x)\right)$ (resp. $x \to g'_n(g_n(x))$, $x \to h(g(x))$ and $x \to g'(g(x))$ ).*

**Proof.** Let us prove the second bullet. Let $\epsilon > 0$. Since $(g_n)_{n\in\mathbb{N}}$ uniformly converges towards $g$, there exists $n_g \in \mathbb{N}$ such that

$$\forall n \geq n_g, \forall x \in \mathcal{X}, \|g_n(x) - g(x)\|_{\mathcal{X}} \leq \frac{\epsilon}{2L_h} \ . \tag{C.54}$$

Since $(h_n)_{n\in\mathbb{N}}$ uniformly converges towards $h$, there exists $n_h \in \mathbb{N}$ such that

$$\forall n \geq n_h, \forall x \in \mathcal{X}, |h_n(x) - h(x)| \leq \frac{\epsilon}{2} \ . \tag{C.55}$$

We denote by $n_{h\circ g} = \max(n_h, n_g)$. One has
$\forall n \geq n_{h\circ g}, \forall x \in \mathcal{X}$,

$$\begin{aligned}
|h_n(g_n(x)) - h(g(x))| &\leq |h_n(g_n(x)) - h(g_n(x))| + |h(g_n(x)) - h(g(x))| && \text{(C.56)}\\
&\leq \frac{\epsilon}{2} + L_h\|g_n(x) - g(x)\|_{\mathcal{X}} && \text{(C.57)}\\
&\leq \frac{\epsilon}{2} + L_h\frac{\epsilon}{2L_h} && \text{(C.58)}\\
&\leq \epsilon, && \text{(C.59)}
\end{aligned}$$

which proves that the sequence of functions $(h_n \circ g_n)_n$ uniformly converges towards $h \circ g$. ■

**Lemma C.4.8 (Convergence of $\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0)$ towards $J^{\mathbf{u}}(x_0)$ ,$\forall \mathbf{u} \in \mathcal{U}^T$ )**
$\forall \mathbf{u} \in \mathcal{U}^T, \forall x_0 \in \mathcal{X}$,

$$\lim_{n\to\infty} \left|\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) - J^{\mathbf{u}}(x_0)\right| = 0 \ . \tag{C.60}$$

**Proof.** Let $\mathbf{u} \in \mathcal{U}^T$ be a fixed sequence of actions. For all $n \in \mathbb{N}, n \geq n_0$ the function $\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n} : \mathcal{X} \to \mathbb{R}$ can be written as follows :

$$\begin{aligned}
\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n} &= \tilde{\rho}_{\mathcal{F}_n,u_0} + \tilde{\rho}_{\mathcal{F}_n,u_1} \circ \tilde{f}_{\mathcal{F}_n,u_0} \\
&+ \dots \\
&+ \tilde{\rho}_{\mathcal{F}_n,T-1} \circ \tilde{f}_{\mathcal{F}_n,u_{T-2}} \circ \ldots \circ \tilde{f}_{\mathcal{F}_n,u_0} \; . \qquad \text{(C.61)}
\end{aligned}$$

Since all the functions $\{\tilde{\rho}_{\mathcal{F}_n,u_t}\}_{0\leq t\leq T-1}$ and $\left\{\tilde{f}_{\mathcal{F}_n,u_t}\right\}_{0\leq t\leq T-1}$ uniformly converge towards the functions $\{f_{u_t}\}_{0\leq t\leq T-1}$ and $\{\rho_{u_t}\}_{0\leq t\leq T-1}$, respectively, and since all the functions $\{f_{u_t}\}_{0\leq t\leq T-1}$ and $\{\rho_{u_t}\}_{0\leq t\leq T-1}$ are Lipschitz continuous, Lemma C.4.6 and Lemma C.4.7 ensure that the function $x_0 \to \tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0)$ uniformly converges to the function $x_0 \to J^{\mathbf{u}}(x_0)$. This implies the convergence of the sequence $\left(\tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0)\right)_{n\in\mathbb{N}}$ towards $J^{\mathbf{u}}(x_0)$, for any sequence of actions $\mathbf{u} \in \mathcal{U}^T$, and for any initial state $x_0 \in \mathcal{X}$. ■

**Proof of Theorem C.4.4.** Let us proof Equation C.37. Let $\mathbf{u}^*(x_0)$ be an optimal sequence of actions, and $\left(\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)\right)_{n\in\mathbb{N}}$ be a sequence of sequence of actions computed by the Voronoi RL algorithm. Each sequence of actions $\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)$ is optimal with respect to the approximated model defined by the approximated functions $\tilde{f}_{\mathcal{F}_n}$ and $\tilde{\rho}_{\mathcal{F}_n}$. One then has

$$\forall n \geq m, \forall \mathbf{u} \in \mathcal{U}^T, \tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)}_{\mathcal{F}_n}(x_0) \geq \tilde{J}^{\mathbf{u}}_{\mathcal{F}_n}(x_0) \; . \qquad \text{(C.62)}$$

The previous inequality is also valid for the sequence of actions $\mathbf{u}^*(x_0)$:

$$\forall n \geq m, \tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(x_0)}_{\mathcal{F}_n}(x_0) \geq \tilde{J}^{\mathbf{u}^*(x_0)}_{\mathcal{F}_n}(x_0) \; . \qquad \text{(C.63)}$$

Then, $\forall n \geq m$,

$$\begin{aligned}
&\tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) - J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) + J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) \\
&\geq \tilde{J}^{\mathbf{u}^*(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) - J^{\mathbf{u}^*(\mathbf{x_0})}(x_0) + J^{\mathbf{u}^*(\mathbf{x_0})}(x_0) \; . \qquad \text{(C.64)}
\end{aligned}$$

According to Lemma C.4.8, one can write

$$\lim_{n\to\infty} \tilde{J}^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) - J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) = 0 \; , \qquad \text{(C.65)}$$

$$\lim_{n\to\infty} \tilde{J}^{\mathbf{u}^*(\mathbf{x_0})}_{\mathcal{F}_n}(x_0) - J^{\mathbf{u}^*(\mathbf{x_0})}(x_0) = 0 \; . \qquad \text{(C.66)}$$

which leads to

$$\lim_{n\to\infty} J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) \geq \lim_{n\to\infty} J^{\mathbf{u}^*(\mathbf{x_0})}(x_0) = J^*(x_0)\ . \tag{C.67}$$

On the other hand, since $\mathbf{u}^*(\mathbf{x_0})$ is an optimal sequence of actions, one has

$$\forall n \in \mathbb{N}_0, J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) \leq J^{\mathbf{u}^*(x_0)}(x_0) = J^*(x_0)\ , \tag{C.68}$$

which leads to

$$\lim_{n\to\infty} J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) \leq J^*(x_0)\ . \tag{C.69}$$

Equations C.67 and C.69 allow to conclude the proof:

$$\lim_{n\to\infty} J^{\tilde{\mathbf{u}}^*_{\mathcal{F}_\mathbf{n}}(\mathbf{x_0})}(x_0) = J^*(x_0)\ . \tag{C.70}$$

■

# Bibliography

[1] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[2] R. Fonteneau, S.A. Murphy, L. Wehenkel, and D. Ernst. Active exploration by searching for experiments falsifying an already induced policy. *To be published in the Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 2011), Paris, France*, 2011.