ITERATIVE PATH INTEGRAL STOCHASTIC OPTIMAL CONTROL: THEORY AND APPLICATIONS TO MOTOR CONTROL

by

Evangelos A. Theodorou

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

May 2011

Epigraph

VIRTUE, then, being of two kinds, intellectual and moral, intellectual virtue in the main owes both its birth and its growth to teaching (for which reason it requires experience and time) while the moral virtue comes about as a result of a habit, whence also its name *ηθικήή* is one that is formed by the slight variation from of the word *έθos* (habit). From this it is also plain that none of the moral virtues arises in us by nature; for nothing that exist by nature can form a habit contrary to its nature. For instance the stone which by nature moves downwards cannot be habituated to move upwards, not even if one tries to train it by throwing it up ten thousand times; nor can fire be habituated to move downwards, nor can anything else that by nature behaves in one way be trained to behave in another. Neither by nature nor contrary to nature do the virtues arise in us; rather we are adapted by nature to receive them, and are made perfect by habit.

The Nicomachean Ethics, Aristotle, 384-323 B.C[1]

[1]Text taken from the book 'Aristotle:The Nicomachean Ethics' translated by David Ross(Ross 2009)

Dedication

To my first teacher in physics my mother Anastasia.

To my brother Zacharias.

To my guard angel Choanna.

Acknowledgements

In this journey towards my Ph.D. there have been three very important people, Anastasia, Zacharias and Choanna, who deeply understood my intellectual goals and encourage me in every difficult moment. My mother Anastasia was my first teacher in physics, my first intellectual mentor who taught me fairness and morality. She has been giving me the love, the courage and the strength to make my visions real. My brother Zacharias was always there to remind me that I had to stand up and that life is like a marathon. My guard angel Choanna has been always on my side, by teaching me how to enjoy every moment of life, giving me love and positive energy and inspiring me intellectually and mentally. Without the support and the unconditional love of these people I would not have been able to create, fight for and reach my dream.

I would like to thank my colleagues in the Computational Learning and Motor Control lab and in the Brain Body Dynamics lab. Special thanks go to Mike Mistry who besides my colleague, he was my roommate for the first two years in LA and a very good friend. Special thanks go also to Heiko Hoffman. I thank him for his kindness, generosity and friendship all these years. During the last year of my Ph.D I met Daniel Braun as a roommate and a colleague. I am thankful to Daniel for all of our analytical conversations regarding philosophy, epistemology and life.

Inspiration for the work in this thesis comes from the work on path integrals and stochastic optimal control by Prof. Bert Kappen. I can not forget my enthusiasm when I first read his papers. I would like to thank him for his work and the interactions that we had. I am deeply grateful to Prof. Stefan Schaal, my main advisor, for trusting me and giving me the opportunity to study at USC. Stefan gave me the support and the freedom to work on a topic of my choice. I also thank Prof. Francisco J. Valero Cuevas for giving me the opportunity to work on applications of control theory to biomechanics. I am thankful to Prof. Emo Todorov for accepting my request to work in his lab as a visiting student for a summer and for his feedback. Finally I would like to thank Prof. Gaurav Sukhatme and Prof. Nicholas Schweighofer for being members of my committee and for their feedback.

Table of Contents

List Of Tables

List Of Figures

Abstract

Motivated by the limitations of current optimal control and reinforcement learning methods in terms of their efficiency and scalability, this thesis proposes an iterative stochastic optimal control approach based on the generalized path integral formalism. More precisely, we suggest the use of the framework of stochastic optimal control with path integrals to derive a novel approach to RL with parameterized policies. While solidly grounded in value function estimation and optimal control based on the stochastic Hamilton Jacobi Bellman (HJB) equation, policy improvements can be transformed into an approximation problem of a path integral which has no open algorithmic parameters other than the exploration noise. The resulting algorithm can be conceived of as model-based, semi-model-based, or even model free, depending on how the learning problem is structured. The new algorithm, **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$), demonstrates interesting similarities with previous RL research in the framework of probability matching and provides intuition why the slightly heuristically motivated probability matching approach can actually perform well. Applications to high dimensional robotic systems are presented for a variety of tasks that require optimal planning and gain scheduling.

In addition to the work on generalized path integral stochastic optimal control, in this thesis we extend model based iterative optimal control algorithms to the stochastic

setting. More precisely we derive the Differential Dynamic Programming algorithm for stochastic systems with state and control multiplicative noise. Finally, in the last part of this thesis, model based iterative optimal control methods are applied to bio-mechanical models of the index finger with the goal to find the underlying tendon forces applied for the movements of, tapping and flexing.

Chapter 1

Introduction

1.1 Motivation

Given the technological breakthroughs of the last three decades in the areas of computer science and engineering, the speedup in processing power has reached the point where computationally expensive algorithms are now days implemented and executed in an efficient and fast way. At the same time, advancements in memory technology offered the capability for fast and reliable storage of huge amount of information. All this progress in computer science has benefitted robotics due to the fact that computationally heavy control, estimation and machine learning algorithms can now be executed online and in real time. The breakthroughs of technology in terms of computational speed and increasing memory size created new visions in robotics. In particular, future robots will not only perform in industrial environments but they will also safely co-exist with humans in environments less structural and more dynamic and stochastic than the environment of a factory.

Despite all this evolution, learning for a robot how to autonomously perform human-like motor control tasks such as object manipulation, walking, running etc, remains an open problem. There is a combination of characteristics in humanoid robots which is unique and it does not often exist in other cases of dynamical systems. These systems are usually high dimensional. Depending on how many degrees of freedom are considered, their dimensionality can easily exceed 100 states. Moreover their dynamical model is usually unknown and hard estimate. In cases where a model is available, it is an approximation of the real dynamics, especially if one considers contact phenomena with the environment as well as the various sources of stochasticity such as sensor and actuation noise. Therefore, there is a level of uncertainty in humanoid robotic systems which is structural and parametric, because it results from the lack of accurate dynamical models, as well as stochastic due to noisy and imperfect sensors.

All these characteristics of humanoid robots open the question of how humans resolve these issues due to the fact that they also perform motor control tasks in stochastic environments and deal with contact phenomena and sensor noise. As for the characteristic of dimensionality, this is also present in an even more pronounced way in bio-mechanical systems. It suffices to realize that just for the control of the hand there are up to 30 actuated tendons.

Motivated by all these issues and difficulties, this thesis proposes a new stochastic optimal control formalism based on the framework of path integral control, which extends to optics of robot learning and reinforcement learning. Path integral control framework and its extensions to iterative optimal control are the central topic sof this thesis. Moreover, inspired by the mystery of bio-mechanical motor control of the index finger, this thesis

investigates the underlying control strategies and studies their sensitivity with respect to model changes.

Since reinforcement learning and stochastic optimal control are the main frameworks of this thesis, a complete presentation should incorporate views coming from different communities of science and engineering. For this reason, in the next two sections we discuss the optimal control and reinforcement learning frameworks from the control theoretic and machine learning point of view. In the last section of this introductory chapter we provide an outline of this work with a short description of the structure and the contents of each chapter.

1.2 Stochastic optimal control theory

Among the areas of control theory, optimal control is one of the most significant, with a plethora of applications from the very early development of aerospace engineering, to robotics, traffic control, biology and computational motor control. With respect to other control theoretic frameworks, optimal control was the first to introduce optimization as a method to find controls. In fact, optimal control can be thought as a constrained optimization problem that has the characteristic that the constraints are not static, in the sense of algebraic equations, but they correspond to dynamical systems and therefore they are represented by differential equations. The addition of differential equations as constraints in the optimization problem leads to the property that in optimal control theory the minimum is not represented by one point x^* in state space but by a trajectory $\boldsymbol{\tau}^* = (x_1^*, x_2^* ..., x_N^*)$, which is the optimal trajectory.

There are two fundamental principles that establish the theoretical basis of optimal control theory in its early developments. These principles are the Pontryagin Maximum principle and the Bellman Principle of Optimality or Dynamic Programming. The maximum principle was introduced by Lev Semenovich Pontryagin a Russian mathematician, in his work *The Mathematical Theory of Optimal Processes* which was first published in Russian in 1961 and then translated in english in 1962 (Pontryagin, Boltyanskii, Gamkrelidze & Mishchenko 1962). The Dynamic Programming framework was introduced in 1953, by Richard Ernest Bellman, an applied mathematician at the University of Southern California.

In the history of optimal control theory, there has been criticism due to the fact that most of the design and analysis of optimal control theory takes place in time domain. Therefore there was no definite answer regarding the stability margins of optimal controllers and their sensitivity with respect to unknown model parameters and uncertainty. Rudolff Kalman, in his paper *"When is a linear control system optimal?"* which was published in 1964 (Kalman 1964), studied the stability margins of optimal controllers for a special class of disturbances.

Almost one decade later, early research on robust control theory (Safonov & Athans 1976),(Doyle 1978), investigated the stability margins of stochastic optimal controllers and showed that stochastic optimal controllers have poor stability margins. Most of the analysis and design in robust control theory takes place in frequency domain. As a result, many of the applications of robust control theory dealt with the cases of infinite horizon optimal control problems and time invariant dynamical systems. In these cases the analysis in frequency domain is straight forward since the close loop system is time invariant

and the application of Fourrier or Laplace transform does not result in convolution. The risk sensitive optimal control framework and its connection to differential game theory and to H_∞ control provided a method to perform robust control for time varying systems and finite horizon optimal control problems, provided that the disturbances are bounded.

1.3 Reinforcement learning: The machine learning view of optimal control theory

In the theory of machine learning (Bishop 2006), there are 3 learning paradigms, supervised, unsupervised and reinforcement learning. Starting with the domain of supervised learning, the goal is to find high level mathematical representations of the kind $\mathbf{y}_i = f(\mathbf{x}_i)$ between data sets $\mathbf{x}_i, \mathbf{y}_i$. Thus, in most cases the data set $\mathbf{x}_i, \mathbf{y}_i$ is given and the question is weather or not the function $f(\mathbf{x})$ can be found. Classical applications of supervised learning are problems like function approximation, regression and classification. In unsupervised learning the goal is to discover structure in data sets. Applications of unsupervised learning techniques are found in the problems of image segmentation, compression and dimensionality reduction.

The most recent branch of machine learning is the area of reinforcement learning. In a typical reinforcement learning scenario an agent explores the environment such that it finds the set of optimal actions which will move the agent to a desired state. The desirability to reach a goal state is encoded by the reward function. The reward is state depended, and therefore, it has high values in the states close to the goal. An additional characteristic of reinforcement learning is that the reward is the only feedback provided

to the agent. From this feedback the agent has to find a policy $\mathbf{u}(\mathbf{x}, t)$ such that it maximizes its reward, i.e., the optimal policy. The policy $\mathbf{u}(\mathbf{x}, t)$ can be a function of state and/or time, depended on how the learning problem is formulated. Essentially the optimal policy provides the actions at given state and/or time that the agent needs to take in order to maximize its reward.

Reinforcement learning can be also thought as a generalization of Markov Decision Processes(MDP) (Russell & Norvig 2003), (Sutton & Barto 1998). The essential components of MDPs are an initial state $\mathbf{x}_0$, a transition model $T(\mathbf{x}_{i+1}, \mathbf{u}_i, \mathbf{x}_i)$ and the reward function $R(\mathbf{x})$. In MDPs the goal for the agent is to maximize the total expected reward. However, for the case of reinforcement learning the transition model and the reward function may be unknown and subject to be learned by the agent as it explores the environment.

When the agent is a dynamical system, reinforcement learning can be thought as an optimal control problem. In both cases, the task is to optimize a cost function or total expected reward subject to constraints imposed by the dynamics of the system under consideration.

1.4 Dissertation outline

There are 9 chapters in this thesis including the introductory chapter. Chapter 2 is a review of the theory of stochastic optimal control. More precisely we start chapter 2 with the definition of a stochastic optimal control problem and the Dynamic Programming framework. Our discussion continues with the Pontryagin maximum principle and its

connection to dynamic programming. Next the iterative optimal control methods are presented starting with stochastic differential dynamic programming and showing how it is related to Differential Dynamic Programming. Having in mind the criticism of optimal control theory related to robustness, we discuss risk sensitive optimal control framework and its connection to differential game theory. We close chapter 2 with the entropy formulation of stochastic optimal control.

Chapter 3 contains important mathematical background on forward and backward partial differential equations(PDEs), stochastic differential equations(SDEs) and path integrals. Essentially the goal in this chapter is to highlight the connection of these three mathematical structures which are commonly used in mathematical physics and control theory. More precisely we start with the history of path integrals and the way how it is introduced in quantum mechanics. We continue by investigating the connection between forward PDEs, SDEs and path integrals. The Feynman-Kac lemma is presented and its role in bridging the gap between backwards PDEs and SDEs is discussed. After presenting the connection between PDEs, SDEs and path integrals we focus our discussion on PDEs and we investigate the relation between backward and forward PDEs in 3 different levels.

Chapter 4 contains the main theory of path integral control formalism and its application to stochastic optimal control. In particular the stochastic optimal control problem is transformed to an approximation of a path integral through the application of Feynman-Kac lemma. The presentation continues with the derivation of the path integral optimal control for the case of stochastic dynamical systems with state dependent control transition matrices. Variations of the path integral formalism based on the Itô and Stratonovich stochastic calculus and the special classes of dynamical systems, are presented. In this

chapter we go one step further with the iterative version of path integral control and its risk sensitive version.

Chapter 5 is a review of model free reinforcement learning algorithms with an emphasis on policy gradient methods. Starting with the vanilla policy gradient method and the REINFORCE algorithm we show the main derivations of the estimated corresponding gradient for each one of these algorithms. Next, the concept of the natural gradient is introduced, and the Episodic Natural Actor Critic is discussed.

Chapter 6 is dedicated to applications of path integral stochastic optimal control to learning robotic control problems. More precisely, we start with an introduction to dynamic movement primitives (DMPs) and their mathematical representation as nonlinear dynamical systems with adjustable attractor landscapes. Next, we explain how DMPs are used for optimal planning and control of robotic systems. We continue with the application of iterative path integral control to DMPs and the presentation of the resulting algorithm, called **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$). In the remaining of chapter 6, applications of $\mathbf{PI}^2$ to robotic optimal control and planning problems are discussed. These applications include planing and variable stiffness control on simulated as well as real robots.

In chapters 7 and 8, the optimal control framework is applied to bio-mechanical models of the index finger with the goal to understand the underlying control strategies and to study their sensitivity. In particular, chapter 7 is a review of current methodologies in modeling bio-mechanical systems based on the characteristic skeletal mechanics, muscle redundancy and the tendon routing. Moreover the differences between tendons driven and torque driven systems are discussed and a review on previous work of optimal control

and its application to bio-mechanical and psychophysical systems is given. In chapter 8 the basic physiology of the index finger is presented. Furthermore ,the iterative optimal control algorithm is used on two bio-mechanical models of the index finger. The underlying control strategies are computed for a flexing and a tapping movement and their sensitivity with respect to model change is discussed.

In the last chapter 9, we conclude and discuss future research.

Chapter 2

Optimal Control Theory

In this chapter we review the theory of optimal control starting from the Pontryagin maximum principle and the Dynamic Programming framework. In particular in section 2.1 we discuss Dynamic Programming and we explain the concept of a value function or cost to go. Moreover we derive the Hamilton Jacobi Bellman (HJB) equation, a fundamental Partial Differential Equation (PDE) in control. Solving the HJB equation results in finding the value function and defining the optimal control policy. In section 2.2, we review Pontryagin maximum principle, and we derive the Euler Langrange equations. Furthermore we provide the connection between the Pontryagin Maximum Principle and Hamiltonian approach in mechanics.

The application of Dynamic Programming to infinite and/or finite horizon nonlinear optimal control problems in continuous state - action spaces yields a family of iterative algorithms for optimal control. We start our presentation of iterative algorithms in section 2.3 with the derivation of Stochastic Differential Dynamic Programming(SDDP) for state dependend, control dependend and additive noise and we illustrate how SDDP is a generalization of its deterministic version i.e., Differential Dynamic Programming(DDP).

In section 2.4, the connection between the risk sensitive optimal control and the differential game theory is discussed. In particular, our presentation includes the derivation of the HJB equation for the case of risk sensitive cost functions and the case of differential game theoretic optimal control. Finally in section 2.5, we present the entropy formulation of stochastic control theory and in the last section we conclude our discussion of optimal control theory

2.1 Dynamic programming and the Bellman principle of optimality: The continuous case

The goal in stochastic optimal control is to control a dynamical system while minimizing a performance criterion. Thus, the stochastic optimal control problem is a constrained optimization problem where constraints are not only algebraic equation or inequalities but differential equations which consist of the model of the dynamical system. In mathematical terms, the stochastic optimal control problem (Stengel 1994), (Basar & Berhard 1995), (Fleming & Soner 2006), (Bellman & Kalaba 1964) for a nonlinear dynamical systems is expressed as follows:

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{x}) = \min_{\mathbf{u}} \left\langle \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u})dt \right\rangle \tag{2.1}$$

with $\boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u}) = q(\mathbf{x}) + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$ subject to the constrains:

$$\mathbf{dx} = \Big(f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\Big)dt + \mathbf{B}(\mathbf{x})d\mathbf{w} \tag{2.2}$$

or in a more compact form:

$$\mathbf{dx} = F(\mathbf{x}, \mathbf{u})dt + \mathbf{B}(\mathbf{x})d\mathbf{w} \tag{2.3}$$

with $F(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$ and $\mathbf{x} \in \Re^n$ as the state and $\mathbf{u} \in \Re^m$ as the control vector. The immediate cost $\boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u})$ includes the state dependent cost $q(\mathbf{x})$ and the control dependent cost $\mathbf{u}^T\mathbf{R}\mathbf{u}$ while $\phi(\mathbf{x}_{t_N})$ is the terminal cost. The main idea in optimal control is to find the control or policy $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ for which the cost function $J(\mathbf{u}, \mathbf{x})$ is minimized. The minimum of the cost function, the so called *value function* or *cost to go*, $V(\mathbf{x})$ it is defined as $V(\mathbf{x}) = \min_{\mathbf{u}} J(\mathbf{x}, \mathbf{u})$. The value function is a function only of state since the optimal control - policy $\mathbf{u}^* = \mathbf{u}^*(\mathbf{x}, t)$ is a function of the state. Therefore we can write:

$$\min_{\mathbf{u}} J(\mathbf{x}, \mathbf{u}) = J(\mathbf{x}, \mathbf{u}^*) = J(\mathbf{x}, \mathbf{u}^*(\mathbf{x}, t)) = V(\mathbf{x}) \tag{2.4}$$

From the equations above it is clear that the value function depends only on the state. The concept of the value function is essential for the Bellman principle of optimality and the development of the Dynamic Programming framework. More precisely the Bellman principle (Dorato, Cerone & Abdallah 2000) states that:

Bellman Principle of Optimality: *If* $\mathbf{u}^*(\mathbf{x}, \tau)$ *is optimal over the interval* $[t, t_N]$, *starting at state* $\mathbf{x}(t)$ *then* $\mathbf{u}^*(\mathbf{x}, \tau)$ *is necessarily optimal over the subinterval* $[t, t + \Delta t]$ *for any* Δt *such that* $T - t \geq \Delta t \geq 0$.

Proof by contradiction: Let us assume that there exists a policy $\mathbf{u}^{**}(\mathbf{x}, t)$ that yields a smaller value for the cost

$$\left\langle \phi(\mathbf{x}_{t_N}) + \int_{t+\Delta t}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) d\tau \right\rangle \tag{2.5}$$

than $\mathbf{u}^*(\mathbf{x}, t)$ over the subinterval $[t + \Delta t, T]$. It make sense to create the new control law

$$\mathbf{u}(\tau) = \begin{cases} \mathbf{u}^*(\tau) & \text{for } t \leq \tau \leq t + \Delta t, \\ \mathbf{u}^{**}(\tau) & \text{for } t + \Delta t \leq \tau \leq t_N \end{cases} \tag{2.6}$$

Then over the interval $[t, t_N]$ we have

$$\begin{aligned} &\left\langle \int_t^{t+\Delta t} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*) d\tau + \int_{t+\Delta t}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}^{**}, \mathbf{u}^{**}) d\tau + \phi(\mathbf{x}_{t_N}) \right\rangle = \\ &\left\langle \int_t^{t+\Delta t} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*) d\tau \right\rangle + \left\langle \int_{t+\Delta t}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}^{**}, \mathbf{u}^{**}) d\tau + \phi(\mathbf{x}_{t_N}^{**}) \right\rangle \\ &< \left\langle \int_t^{t+\Delta t} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*) d\tau \right\rangle + \left\langle \int_{t+\Delta t}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*) d\tau + \phi(\mathbf{x}_{t_N}^*) \right\rangle \end{aligned} \tag{2.7}$$

Since $\mathbf{u}*$ is optimal, by assumption, over the interval $[t, t_N]$ and the inequality above implies that $\mathbf{u}$ results in a smaller value of the cost function than the optimal we reach a contradiction.

The principle of optimality for the continuous case is formulated as follows:

$$\begin{aligned} V(\mathbf{x}, t) &= \min_{\mathbf{u}[\mathbf{x},(t,t+\Delta t)]} \left\langle \int_t^{t+\Delta t} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, \tau) d\tau + V(\mathbf{x}, t + \Delta t) \right\rangle \\ &= \min_{\mathbf{u}[\mathbf{x},(t,t+\Delta t)]} \left\langle - \int_{t+\Delta t}^{t} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, \tau) d\tau + V(\mathbf{x}, t + \Delta t) \right\rangle \end{aligned} \tag{2.8}$$

$$= -\int_{t+\Delta t}^{t} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*, \tau) d\tau + V(\mathbf{x}, t + \Delta t)$$

In the last line, we are assuming for the analysis that the optimal trajectory and control $\mathbf{x}^*$ and $\mathbf{u}^*$ are known and thus the expectation drops. The total derivative of the value function $V(\mathbf{x}, t)$ with respect to time is expressed as follows:

$$\frac{dV(\mathbf{x}, t)}{dt} = -\boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*, \tau) \tag{2.9}$$

Since the value function V is a function of the state which is a random variable, we can apply the Itô differentiation rule and obtain:

$$dV = \left(\frac{\partial V}{\partial t} + (\nabla_{\mathbf{x}} V)^T \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \right) dt + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V) \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T dt \right) \tag{2.10}$$

By equating the two equation above we arrive at:

$$\frac{\partial V}{\partial t} + \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*, \tau) + (\nabla_{\mathbf{x}} V)^T \mathbf{F}(\mathbf{x}^*, \mathbf{u}^*, t) + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V) \mathbf{B}(\mathbf{x}^*) \mathbf{B}(\mathbf{x}^*)^T \right) = 0 \tag{2.11}$$

The equation above can be also written in the form:

$$\inf_{\mathbf{u}} \left(\frac{\partial V}{\partial t} + \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*, \tau) + (\nabla_{\mathbf{x}} V)^T \mathbf{F}(\mathbf{x}^*, \mathbf{u}^*, t) + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V) \mathbf{B}(\mathbf{x}^*) \mathbf{B}(\mathbf{x}^*)^T \right) \right) = 0 \tag{2.12}$$

Since the term $\frac{\partial V}{\partial t}$ does not depend on $\mathbf{u}$ the equations can be re-arranged as:

$$-\frac{\partial V}{\partial t} = \inf_{\mathbf{u}} \left(\boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, \tau) + (\nabla_{\mathbf{x}} V)^T \mathbf{F}(\mathbf{x}, \mathbf{u}, t) + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V)\, \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T \right) \right) \tag{2.13}$$

The equation above is the so called Hamilton - Jacobi - Bellman PDE derived for the case of stochastic dynamical systems. Since $F(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$ and $\boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) = q(\mathbf{x}) + \frac{1}{2}\mathbf{u}^T \mathbf{R} \mathbf{u}$ the right hand side of the equation above is convex with respect to the controls $\mathbf{u}$ and therefore its minimization will results in the following equation:

$$\mathbf{u}^*(\mathbf{x}, t) = -\mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T \nabla_{\mathbf{x}} V \tag{2.14}$$

The optimal control policy $\mathbf{u}^*(\mathbf{x}, t)$ will move the system towards the direction of the minimum value function since it is proportional to the negative direction of the gradient of the value function, projected on the state space $\mathbf{x}$ by the multiplication with $\mathbf{G}(\mathbf{x})$ and weighted with the inverse of the control cost matrix $\mathbf{R}$. Now substitution of the optimal controls in the HJB equation yields the following PDE:

$$\begin{aligned} -\frac{\partial V}{\partial t} = {} & q(\mathbf{x}, t) + (\nabla_{\mathbf{x}} V_t)^T \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} (\nabla_{\mathbf{x}} V_t)^T \mathbf{B}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{B}(\mathbf{x})^T (\nabla_{\mathbf{x}} V_t) \\ & + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V_t) \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T \right) \end{aligned} \tag{2.15}$$

with the boundary terminal condition $V(\mathbf{x}(t_N) = \phi(\mathbf{x}(t_N))$. The equation above is a backward PDE of second order and nonlinear. Its solution is required in order to find the value function $V(\mathbf{x}, t)$ and then the gradient of value function is computed to determine

the optimal control policy. There are few special cases of the PDE in (2.15) depending on the cost function. More precisely if the stochastic optimal control problem is infinite horizon with the cost function:

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{x}) = \min_{\mathbf{u}} \left\langle \int_{t_0}^{\infty} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) dt \right\rangle \tag{2.16}$$

then the value function $V(\mathbf{x})$ is not a function of time and thus the resulting PDE is expressed as:

$$0 = q(\mathbf{x}, t) + (\nabla_{\mathbf{x}} V_t)^T \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}(\nabla_{\mathbf{x}} V_t)^T \mathbf{B}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{B}(\mathbf{x})^T (\nabla_{\mathbf{x}} V_t) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} V_t) \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T \right)$$

For the case of the discounted cost function of the form:

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{x}) = \min_{\mathbf{u}} \left\langle \int_{t_0}^{\infty} e^{-\beta t} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) dt \right\rangle \tag{2.17}$$

the partial time derivative of the value function will be equal to $\frac{\partial V}{\partial t} = \beta V$, and thus the PDE in (2.15) is formulated as follows:

$$\begin{aligned} -\beta V = {} & q(\mathbf{x}, t) + (\nabla_{\mathbf{x}} V_t)^T \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}(\nabla_{\mathbf{x}} V_t)^T \mathbf{B}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{B}(\mathbf{x})^T (\nabla_{\mathbf{x}} V_t) \\ & + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} V_t) \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T \right) \end{aligned} \tag{2.18}$$

In all cases of cost functions above, the solution of the corresponding PDEs especially in high dimensional state spaces is challenging and this is what makes, in general the optimal control problem difficult when applied to high dimensional and nonlinear dynamical systems. For linear systems, the PDE above collapses to the so called Riccati equations (Stengel 1994),(Dorato et al. 2000), the solution of which provides a linear control policy in the state $\mathbf{x}$ of the form $\mathbf{u}(\mathbf{x},t) = -\mathbf{K}(t)\ \mathbf{x}$ where the matrix $\mathbf{K}(t) \in \Re^{n\times m}$ is the control gain.

2.2 Pontryagin maximum principle

In this section we discuss the Pontryagin's Maximum principle (Pontryagin et al. 1962), (Stengel 1994) one of most important principles in the history of the optimal control theory. In our presentation of Pontryagin mimimum principle, we are dealing with deterministic systems and therefore, the deterministic optimal control problem:

$$J(\mathbf{u},\mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u})dt \tag{2.19}$$

subject to dynamics $\frac{d\mathbf{x}}{dt} = F(\mathbf{x},\mathbf{u}) = f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$. The constraint is pushed into the cost function and with lagrange multiplier (Stengel 1994). More precisely the augmented cost function is expressed by the equation:

$$J_A(\mathbf{u},\mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u})dt + \int_{t_0}^{t_N} \boldsymbol{\lambda}^T \left(\frac{d\mathbf{x}}{dt} - F(\mathbf{x},\mathbf{u})\right)dt$$

or

$$J_A(\mathbf{u},\mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \left(\boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u}) - \boldsymbol{\lambda}^T \left(\frac{d\mathbf{x}}{dt} - F(\mathbf{x},\mathbf{u}) \right) \right) dt$$

By defining the Hamiltonian as $\boldsymbol{\mathcal{H}}(\mathbf{x},\mathbf{u}) = \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T F(\mathbf{x},\mathbf{u})$ the augmented cost function can be rewritten in the form:

$$J_A(\mathbf{u},\mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \left(\boldsymbol{\mathcal{H}}(\mathbf{x},\mathbf{u}) - \boldsymbol{\lambda}^T \frac{d\mathbf{x}}{dt} \right) dt$$

Integration by parts will result in:

$$J_A(\mathbf{u},\mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{H}}(\mathbf{x},\mathbf{u}) dt + \left(\boldsymbol{\lambda}(t_0)^T \mathbf{x}(t_0) - \boldsymbol{\lambda}(t_N)^T \mathbf{x}(t_N) \right) + \int_{t_0}^{t_N} \boldsymbol{\lambda}^T \mathbf{x} dt \quad (2.20)$$

Now we will find the variation in the augmented cost δJ_A which is expressed by the equation that follows:

$$\delta J_A = \nabla_{\mathbf{x}} J_A^T \delta\mathbf{x} + \nabla_{\mathbf{u}} J_A^T \delta\mathbf{u}$$

Thus we will have that:

$$\begin{aligned} \delta J_A = \nabla_{\mathbf{x}} \phi^T \ \delta\mathbf{x} \mid_{t=t_N} + \nabla_{\mathbf{x}} \left(\boldsymbol{\lambda}(t_0)^T \mathbf{x}(t_0) - \boldsymbol{\lambda}(t_N)^T \mathbf{x}(t_N) \right) \ \delta\mathbf{x} \\ + \int_{t_0}^{t_N} \left(\nabla_{\mathbf{x}} \boldsymbol{\mathcal{H}}^T \delta\mathbf{x} + \nabla_{\mathbf{x}} \left(\dot{\boldsymbol{\lambda}}^T \mathbf{x}(t) \right) \delta\mathbf{x} + \nabla_{\mathbf{u}} \boldsymbol{\mathcal{H}}^T \delta\mathbf{u} \right) dt \end{aligned}$$

By rearranging the terms the equation above is formulated as follows:

$$\delta J_A = \Big(\nabla_{\mathbf{x}}\phi^T - \boldsymbol{\lambda}(t_N)\Big)\delta\mathbf{x}\mid_{t=t_N} + \boldsymbol{\lambda}(t_0)^T\ \delta\mathbf{x} + \int_{t_0}^{t_N}\left(\Big(\nabla_{\mathbf{x}}\boldsymbol{\mathcal{H}}^T + \dot{\boldsymbol{\lambda}}^T\Big)\delta\mathbf{x} + \nabla_{\mathbf{u}}\boldsymbol{\mathcal{H}}^T\delta\mathbf{u}\right)dt$$

or

$$\delta J_A = \delta J_A(t_0) + \delta J_A(t_N) + \delta J_A(t_0 \to t_N)$$

For $\delta J_A = 0$ we require that $\delta J_A(t_N) = \delta J_A(t_0) = \delta J_A(t_0 \to t_N) = 0$ and thus we will have that:

$$\nabla_{\mathbf{u}}\boldsymbol{\mathcal{H}} = 0 \tag{2.21}$$

and $\boldsymbol{\lambda}(t) = \nabla_{\mathbf{x}}\boldsymbol{\mathcal{H}}$ which since $\boldsymbol{\mathcal{H}}(\mathbf{x},\mathbf{u}) = \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T F(\mathbf{x},\mathbf{u})$ is formulated as follows:

$$\dot{\boldsymbol{\lambda}}(t) = \nabla_{\mathbf{x}}\boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T\nabla_{\mathbf{x}}F(\mathbf{x},\mathbf{u}) \tag{2.22}$$

with the boundary terminal condition:

$$\boldsymbol{\lambda}(t_N) = \nabla_{\mathbf{x}}\phi(\mathbf{x})|_{t=t_N} \tag{2.23}$$

The equations (2.21),(2.22) and (2.23) are the so called Euler Lagrange equations. There are few important observations based on the structure of the Euler Lagrange equations. The Lagrange multiplier or adjoint vector otherwise $\boldsymbol{\lambda}$ represents the cost sensitivity

to dynamic effects. This sensitivity is specified at the final time t_N by providing a boundary condition for the solution of $\boldsymbol{\lambda}$. Another way of interpreting the role of the adjoint vector $\boldsymbol{\lambda}$ is that it quantifies the sensitivity of the cost as a function to state perturbations on the optimal trajectory beginning from the terminal state $\mathbf{x}(t_N)$ and backward propagating towards $\mathbf{x}(t_0)$. Thus the idea in these backward propagation scheme is that in order to decide which way to go it helps to know the effects of the future variation from the resulting path in state space. This knowledge is encoded in the adjoint vector $\boldsymbol{\lambda}$. Clearly the optimal strategy is resulted by tracing paths back from the destination and therefore looking in that way into the future outcomes of possible variations.

The necessary and sufficient conditions for optimality of $\mathbf{x}^*(t)$ and $\mathbf{u}^*(t)$ in the interval $[t_0, t_N]$ if the dynamic systems under consideration is normal and the optimal path contains no conjugate points afre expressed by the equations:

$$\nabla_{\mathbf{x}} \boldsymbol{\mathcal{H}} \Big(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, t \Big) = 0 \tag{2.24}$$

$$\nabla_{\mathbf{xx}} \boldsymbol{\mathcal{H}} \Big(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, t \Big) \geq 0 \tag{2.25}$$

The Pontryagin minimum principle of optimality states that if the variables $\mathbf{x}^*(t), \boldsymbol{\lambda}^*(t)$ are kept fixed then for any admissible neighboring non-optimal control history $\mathbf{u}(t)$ in $[t_0, t_N]$ we have that:

$$\boldsymbol{\mathcal{H}}^* = \boldsymbol{\mathcal{H}}^* \Big(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t \Big) \leq \boldsymbol{\mathcal{H}}^* \Big(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t \Big) \tag{2.26}$$

if $\boldsymbol{\mathcal{H}}$ is stationary and convex the minimum principle is satisfied. If it is the one but no the other then the minimum principle is not satisfied . A stronger condition for the minimum principle is formulated as follows:

$$J(\mathbf{u}^* + \delta\mathbf{u}) - J(\mathbf{u}^*) = \tag{2.27}$$

$$= \int_{t_0}^{t_N} \left[\boldsymbol{\mathcal{H}}^* \left(\mathbf{x}^*(t), \mathbf{u}^*(t) + \delta\mathbf{u}, \boldsymbol{\lambda}^*(t), t \right) - \boldsymbol{\mathcal{H}}^* \left(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t \right) \right] dt \geq 0 \tag{2.28}$$

2.3 Iterative optimal control algorithms

There is a variety of optimal control algorithms depending on 1) the order of the expansion of the dynamics, 2) the order of the expansion of the cost function and 3) the existence of noise.

More precisely, if the dynamics under consideration are linear in the state and the controls, deterministic, and the cost function is quadratic with respect to states and controls, we can use one of the most established tools in control theory: the Linear Quadratic Regulator (Stengel 1994). For such type of optimal control problems the dynamics are formulated as $f(\mathbf{x}, \mathbf{u}) = A\mathbf{x} + B\mathbf{u}$, $F(\mathbf{x}, \mathbf{u}) = 0$ and the immediate cost $l\left(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau, \mathbf{x}(\tau))\right) = \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}$. Under the presence of stochastic dynamics $F(\mathbf{x}, \mathbf{u}) \neq 0$, the resulting algorithm is called the Linear Gaussian Quadratic Regulator (LQG).

For nonlinear deterministic dynamical systems, expansion of the dynamics is performed and the optimal control algorithm is solved in iterative fashion. Under a first

	LQR	LQG	iLQR	iLQG	DDP	SDDP
Linear Dynamics	x	x	-	-	-	-
Quadratic Cost	x	x		-	-	-
FOE of Dynamics	-		x	x	-	-
SOE of Cost	-		x	x	x	x
SOE of Dynamics	-		-	-	x	x
Deterministic	x		x	-	x	-
Stochastic	-	x	-	x	-	x

Table 2.1: Optimal Control Algorithms according to First Order Expansion (FOE) or Second Order Expansion (SOE) of dynamics and cost function and the existence of Noise.

order expansions of the dynamics and a second order expansion of the immediate cost function $l(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau, \mathbf{x}(\tau)))$ the derived algorithm is called Iterative Linear Quadratic Regulator (iLQR) (Li & Todorov 2004). A better approximation of dynamics up to the second order results in one of the most well know optimal control algorithm especially in the area of Robotics, Differential Dynamic Programming (Jacobson & Mayne 1970). Both iLQR and DDP are iterative algorithms that start with an initial trajectory in states and controls $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ and result in an optimal trajectory $\mathbf{x}^*$, an optimal open loop control command $\mathbf{u}^*$, and a sequence of control gains $\mathbf{L}$ which are activated whenever deviations from the optimal trajectory $\mathbf{x}^*$ are observed. The difference between iLQR and DDP is that DDP provides a better approximation of the dynamics but with an additional computational cost necessary to find the second order derivatives.

In cases where noise is present in the dynamics either as multiplicative in the controls or state, or both, we have the stochastic version of iLQR and DDP, the Iterative Linear Quadratic Gaussian Regulator (iLQG) (Todorov 2005) and the Stochastic Differential Dynamic Programming (SDDP) (Theodorou 2010). Essentially SDDP contains as special cases all the previous algorithms iLQR, iLQG and DDP since it requires second order expansion of the cost and dynamics and it takes into account control and state

dependent noise. This is computationally costly because second order derivatives have to be calculated. An important aspect of stochastic optimal control theory is that, in cases of additive noise, the optimal control $\mathbf{u}^*$ and the optimal control gains $\mathbf{L}$ are both independent of the noise and, therefore, the same with the corresponding deterministic solution. In cases where the noise is control or state dependent, the resulting solutions iLQG and SDDP differ from the solutions of the deterministic versions iLQR and DDP. In the table 2.1 we provide the classification of the optimal control algorithms based on the expansion of dynamics and cost function as well as the existence of noise.

2.3.1 Stochastic differential dynamic programming

We consider the class of nonlinear stochastic optimal control problems with cost

$$v^{\pi}(\mathbf{x}, t) = \left\langle h(\mathbf{x}(T)) + \int_{t_0}^{T} \ell\left(\tau, \mathbf{x}(\tau), \pi(\tau, \mathbf{x}(\tau))\right) d\tau \right\rangle \tag{2.29}$$

subject to the stochastic dynamics of the form:

$$\mathbf{dx} = f(\mathbf{x}, \mathbf{u})dt + F(\mathbf{x}, \mathbf{u})d\mathbf{w} \tag{2.30}$$

where $\mathbf{x} \in \Re^{n\times 1}$ is the state, $\mathbf{u} \in \Re^{m\times 1}$ is the control and $d\mathbf{w} \in \Re^{p\times 1}$ is brownian noise. The term $h(\mathbf{x}(T))$ in the cost function (2.29), is the terminal cost while the $\ell\left(\tau, \mathbf{x}(\tau), \pi(\tau, \mathbf{x}(\tau))\right)$ is the instantaneous cost rate which is a function of the state $\mathbf{x}$ and control policy $\pi(\tau, \mathbf{x}(\tau))$. The cost-to - go $v^{\pi}(\mathbf{x}, t)$ is defined as the expected cost

accumulated over the time horizon $(t_0, ..., T)$ starting from the initial state $\mathbf{x}_t$ to the final state $\mathbf{x}(T)$.

To enhance the readability of our derivations we write the dynamics as a function $\Phi \in \Re^{n\times 1}$ of the state, control and instantiation of the noise:

$$\Phi(\mathbf{x}, \mathbf{u}, d\omega) \equiv f(\mathbf{x}, \mathbf{u})dt + F(\mathbf{x}, \mathbf{u})d\mathbf{w} \tag{2.31}$$

It will sometimes be convenient to write the matrix $F(\mathbf{x}, \mathbf{u}) \in \Re^{n\times p}$ in terms of its rows or columns:

$$F(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} F_r^1(\mathbf{x}, \mathbf{u}) \\ \vdots \\ F_r^n(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \begin{bmatrix} F_c^1(\mathbf{x}, \mathbf{u}), \ldots, F_c^p(\mathbf{x}, \mathbf{u}) \end{bmatrix}$$

Every element of the vector $\Phi(\mathbf{x}, \mathbf{u}, d\omega) \in \Re^{n\times 1}$ can now be expressed as:

$$\Phi^j(\mathbf{x}, \mathbf{u}, d\omega) = f^j(\mathbf{x}, \mathbf{u})\delta t + F_r^j(\mathbf{x}, \mathbf{u})d\mathbf{w}$$

Given a nominal trajectory of states and controls $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ we expand the dynamics around this trajectory to second order:

$$\Phi(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}, d\mathbf{w}) =$$

$$\Phi(\bar{\mathbf{x}}, \bar{\mathbf{u}}, d\mathbf{w}) + \nabla_x\Phi \cdot \delta\mathbf{x} + \nabla_{\mathbf{u}}\Phi \cdot \delta\mathbf{u} + \mathbf{O}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w})$$

where $\mathbf{O}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) \in \Re^{n\times 1}$ contains all the second order terms in the deviations in states, controls and noise[1]. Writing this term element-wise:

$$\mathbf{O}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) = \begin{pmatrix} O^{(1)}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) \\ \vdots \\ O^{(n)}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) \end{pmatrix},$$

we can express the elements $O^{(j)}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) \in \Re$ as:

$$O^{(j)}(\delta\mathbf{x}, \delta\mathbf{u}, d\mathbf{w}) = \frac{1}{2} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{xx}}\Phi^j & \nabla_{\mathbf{xu}}\Phi^j \\ \nabla_{\mathbf{ux}}\Phi^j & \nabla_{\mathbf{uu}}\Phi^j \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}. \tag{2.32}$$

We would now like to express the derivatives of Φ in terms of the given quantities. Beginning with the first-order terms, we find that:

$$\nabla_{\mathbf{x}}\Phi = \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{u})\delta t + \nabla_{\mathbf{x}} \left(\sum_{i=1}^{m} F_c^i \; d\mathbf{w}_t^{(i)} \right)$$
$$\nabla_{\mathbf{u}}\Phi = \nabla_{\mathbf{u}} f(\mathbf{x}, \mathbf{u})\delta t + \nabla_{\mathbf{u}} \left(\sum_{i=1}^{m} F_c^i \; d\mathbf{w}_t^{(i)} \right)$$

Next we find the second order derivatives and we have that:

$$\nabla_{\mathbf{xx}}\Phi^{(j)} = \nabla_{\mathbf{xx}} f^{(j)}(\mathbf{x}, \mathbf{u})\delta t + \nabla_{\mathbf{xx}} \left(F_r^{(j)}(\mathbf{x}, \mathbf{u}) d\mathbf{w}_t \right)$$
$$\nabla_{\mathbf{uu}}\Phi^{(j)} = \nabla_{\mathbf{uu}} f^{(j)}(\mathbf{x}, \mathbf{u})\delta t + \nabla_{\mathbf{uu}} \left(F_r^{(j)}(\mathbf{x}, \mathbf{u}) d\mathbf{w}_t \right)$$
$$\nabla_{\mathbf{ux}}\Phi^{(j)} = \nabla_{\mathbf{ux}} f^{(j)}(\mathbf{x}, \mathbf{u})\delta t + \nabla_{\mathbf{ux}} \left(F_r^{(j)}(\mathbf{x}, \mathbf{u}) d\mathbf{w}_t \right)$$

[1] Not to be confused with "big-O".

$$\nabla_{\mathbf{xu}}\Phi^{(j)} = \left(\nabla_{\mathbf{ux}}\Phi^{(j)}\right)^{\mathsf{T}}$$

After expanding the dynamics up to the second order we can transition from continuous to discrete time. More precisely the discrete-time dynamics are formulated as:

$$\begin{aligned}\delta\mathbf{x}_{t+\delta t} = &\left(I_{n\times n} + \nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{u})\delta t + \nabla_{\mathbf{x}}\left(\sum_{i=1}^{m} F_c^{(i)}\ \xi_t^{(i)}\sqrt{\delta t}\right)\right)\delta\mathbf{x}_t \\ &+ \left(\nabla_{\mathbf{u}} f(\mathbf{x},\mathbf{u})\delta t + \nabla_{\mathbf{u}}\left(\sum_{i=1}^{m} F_c^{(i)}\ \boldsymbol{\xi}_t^{(i)}\sqrt{\delta t}\right)\right)\delta\mathbf{u}_t \\ &+ F(\mathbf{x},\mathbf{u})\sqrt{\delta t}\boldsymbol{\xi}_t + \mathbf{O}_d(\delta\mathbf{x},\delta\mathbf{u},\boldsymbol{\xi},\delta t)\end{aligned}$$

with $\delta t = t_{k+1} - t_k$ corresponding to a small discretization interval. Note that the term $\mathbf{O}_d$ is the equivalent of $\mathbf{O}$ but in discrete time and therefore it is now a function of δt. In fact, since $\mathbf{O}_d$ contains all the second order expansion terms of the dynamics it contains second order derivatives WRT state and control expressed as follows:

$$\begin{aligned}\nabla_{\mathbf{xx}}\Phi^{(j)} &= \nabla_{\mathbf{xx}} f^{(j)}(\mathbf{x},\mathbf{u})\delta t + \nabla_{\mathbf{xx}}\left(F_r^{(j)}(\mathbf{x},\mathbf{u})\boldsymbol{\xi}_t\right)\sqrt{\delta t} \\ \nabla_{\mathbf{uu}}\Phi^{(j)} &= \nabla_{\mathbf{uu}} f^{(j)}(\mathbf{x},\mathbf{u})\delta t + \nabla_{\mathbf{uu}}\left(F_r^{(j)}(\mathbf{x},\mathbf{u})\boldsymbol{\xi}_t\right)\sqrt{\delta t} \\ \nabla_{\mathbf{ux}}\Phi^{(j)} &= \nabla_{\mathbf{ux}} f^{(j)}(\mathbf{x},\mathbf{u})\delta t + \nabla_{\mathbf{ux}}\left(F_r^{(j)}(\mathbf{x},\mathbf{u})\boldsymbol{\xi}_t\right)\sqrt{\delta t} \\ \nabla_{\mathbf{xu}}\Phi^{(j)} &= \left(\nabla_{\mathbf{ux}}\Phi^{(j)}\right)^{\mathsf{T}}\end{aligned}$$

The random variable $\boldsymbol{\xi} \in \Re^{p\times 1}$ is zero mean and Gaussian distributed with covariance $\Sigma = \sigma^2 I_{m\times m}$ The discretized dynamics can be written in a more compact form by grouping the state, control and noise dependent terms, and leaving the second order term separate:

$$\delta\mathbf{x}_{t+\delta t} = A_t\delta\mathbf{x}_t + B_t\delta\mathbf{u}_t + \Gamma_t\boldsymbol{\xi}_t + \mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}, \delta t) \tag{2.33}$$

where the matrices $A_t \in \Re^{n\times n}, B_t \in \Re^{n\times m}$ and $\Gamma_t \in \Re^{n\times p}$ are defined as

$$\begin{aligned} A_t &= I_{n\times n} + \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{u})\delta t \\ B_t &= \nabla_{\mathbf{u}} f(\mathbf{x}, \mathbf{u})\delta t \\ \Gamma_t &= \begin{bmatrix} \Gamma^{(1)} & \Gamma^{(2)} & \dots & \Gamma^{(m)} \end{bmatrix} \end{aligned}$$

with $\Gamma^{(i)} \in \Re^{n\times 1}$ defined $\Gamma^{(i)} = \nabla_{\mathbf{u}} F_c^{(i)}\delta\mathbf{u}_t + \nabla_{\mathbf{x}} F_c^{(i)}\delta\mathbf{x}_t + F_c^{(i)}$. For the derivation of the optimal control it is useful to expresses Γ_t as the summation of terms that depend on variations in state and controls and terms that are independent of such variations. More precisely we will have that:

$$\Gamma_t = \Delta_t(\delta\mathbf{x}, \delta\mathbf{u}) + F(\mathbf{x}, \mathbf{u}) \tag{2.34}$$

where each column vector of Δ_t is defined as $\Delta_t^{(i)}(\delta\mathbf{x}, \delta\mathbf{u}) = \nabla_{\mathbf{u}} F_c^{(i)}\delta\mathbf{u}_t + \nabla_{\mathbf{x}} F_c^{(i)}\delta\mathbf{x}_t$.

2.3.1.1 Value function second order approximation

As in classical DDP, the derivation of stochastic DDP requires the second order expansion of the cost-to-go function around a nominal trajectory $\bar{\mathbf{x}}$:

$$V(\bar{\mathbf{x}} + \delta\mathbf{x}) = V(\bar{\mathbf{x}}) + V_{\mathbf{x}}^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T V_{\mathbf{xx}} \delta\mathbf{x} \tag{2.35}$$

Substitution of the discretized dynamics (2.33) in the second order Value function expansion (8.7) results in:

$$\begin{aligned} V(\bar{\mathbf{x}}_{t+\delta t} + \delta\mathbf{x}_{t+\delta t}) &= V(\bar{\mathbf{x}}_{t+\delta t}) + V_{\mathbf{x}}^T \left(A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \Gamma_t \boldsymbol{\xi} + \mathbf{O}_d\right) \\ &+ \left(A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \Gamma_t \boldsymbol{\xi} + \mathbf{O}_d\right)^T V_{\mathbf{xx}} \left(A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \Gamma_t \boldsymbol{\xi} + \mathbf{O}_d\right) \end{aligned} \tag{2.36}$$

Next we will compute $E\left(V(\bar{\mathbf{x}}_{t+\delta t} + \delta\mathbf{x}_{t+\delta t})\right)$ which requires the calculation of the expectation of the all the terms that appear in the equation above. This is what the rest of the analysis is dedicated to. More precisely in the next two sections we will calculate the expectation of the terms:

$$\left\langle V_{\mathbf{x}}^T \delta\mathbf{x}_{t+\delta t} \right\rangle \quad \text{and} \quad \left\langle \delta\mathbf{x}_{t+\delta t}^T V_{\mathbf{xx}} \delta\mathbf{x}_{t+\delta t} \right\rangle \tag{2.37}$$

where the state deviation $\delta\mathbf{x}_{t+\delta t}$ at time instant $t + \delta t$ is given by the linearized dynamics:

$$\delta\mathbf{x}_{t+\delta t} = A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \Gamma_t \boldsymbol{\xi} + \mathbf{O}_d \tag{2.38}$$

The analysis that follows in section 2.3.1.1 consist of the computation of the expectation of the four terms which result from the substitution of the linearized dynamics (2.38) into $E\left\langle V_{\mathbf{x}}^T \delta\mathbf{x}_{t+\delta t} \right\rangle$. In section 2.3.1.1 we compute the expectation of the 16 terms that result from the substitution of (2.38) into $\left\langle \delta\mathbf{x}_{t+\delta t}^T V_{\mathbf{xx}} \delta\mathbf{x}_{t+\delta t} \right\rangle$.

Expectation of the 1st order term of the value function.

The expectation of the first order term results in:

$$\left\langle V_{\mathbf{x}}^T \left(A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \Gamma_t \boldsymbol{\xi}_t + \mathbf{O}_d \right) \right\rangle = V_{\mathbf{x}}^T \left(A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \left\langle \mathbf{O}_d \right\rangle \right) \tag{2.39}$$

In order to find the expectation of $\mathbf{O}_d \in \Re^{n\times 1}$ we need to find the expectation of each one of the elements of this column vector. Thus we will have that:

$$\begin{aligned} \left\langle O^{(j)}(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t) \right\rangle &= \left\langle \frac{1}{2} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}^T \begin{pmatrix} \nabla_{\mathbf{xx}}\Phi^{(j)} & \nabla_{\mathbf{xu}}\Phi^{(j)} \\ \nabla_{\mathbf{ux}}\Phi^{(j)} & \nabla_{\mathbf{uu}}\Phi^{(j)} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} \right\rangle = \\ &= \frac{\delta t}{2} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}^T \begin{pmatrix} \nabla_{\mathbf{xx}} f^{(j)} & \nabla_{\mathbf{xu}} f^{(j)} \\ \nabla_{\mathbf{ux}} f^{(j)} & \nabla_{\mathbf{uu}} f^{(j)} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} = \tilde{O}^j \end{aligned} \tag{2.40}$$

Therefore we will have that:

$$\left\langle V_{\mathbf{x}}^T \delta\mathbf{x}_{t+\delta t} \right\rangle = V_{\mathbf{x}}^T \left\langle A_t \delta\mathbf{x}_t + B_t \delta\mathbf{u}_t + \tilde{\mathbf{O}}_d \right\rangle \tag{2.41}$$

Where the term $\tilde{\mathbf{O}}_d$ is defined as:

$$\tilde{\mathbf{O}}_d(\delta\mathbf{x}, \delta\mathbf{u}, \delta t) = \begin{pmatrix} \tilde{O}^{(1)}(\delta\mathbf{x}, \delta\mathbf{u}, \delta t) \\ \\ ... \\ \\ ... \\ \\ \tilde{O}^{(n)}(\delta\mathbf{x}, \delta\mathbf{u}, \delta t) \end{pmatrix} \tag{2.42}$$

The term $\nabla_{\mathbf{x}} V^T \tilde{\mathbf{O}}_d$ is quadratic in variations in the states and controls $\delta\mathbf{x}, \delta\mathbf{u}$ an thus there are the symmetric matrices $\boldsymbol{\mathcal{F}} \in \Re^{n\times n}$, $\boldsymbol{\mathcal{Z}} \in \Re^{m\times m}$ and $\boldsymbol{\mathcal{L}} \in \Re^{m\times n}$ such that:

$$V_{\mathbf{x}}^T \tilde{\mathbf{O}}_d = \frac{1}{2}\delta\mathbf{x}^T \boldsymbol{\mathcal{F}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{u}^T \boldsymbol{\mathcal{Z}} \delta\mathbf{u} + \delta\mathbf{u}^T \boldsymbol{\mathcal{L}} \delta\mathbf{x} \tag{2.43}$$

with

$$\boldsymbol{\mathcal{F}} = \left(\sum_{j=1}^{n} \nabla_{\mathbf{xx}} f^{(j)} V_{x_j} \right) \tag{2.44}$$

$$\boldsymbol{\mathcal{Z}} = \left(\sum_{j=1}^{n} \nabla_{\mathbf{uu}} f^{(j)} V_{x_j} \right) \tag{2.45}$$

$$\boldsymbol{\mathcal{L}} = \left(\sum_{j=1}^{n} \nabla_{\mathbf{ux}} f^{(j)} V_{x_j} \right) \tag{2.46}$$

From the analysis above we can see that the expectation $\nabla_x V^T \delta\mathbf{x}_{t+\delta t}$ is a quadratic function with respect to variations in states and controls $\delta\mathbf{x}, \delta\mathbf{u}$. As we will prove in

the next section the expectation of $\delta\mathbf{x}_{t+\delta t}^T \nabla_{\mathbf{xx}} V^T \delta\mathbf{x}_{t+\delta t}$ is also a quadratic function of variations in states and controls $\delta\mathbf{x}, \delta\mathbf{u}$.

Expectation of the 2nd order term of the value function.

In this section we compute all the terms that appear due to the expectation of the second approximation of the value function $\left\langle \delta\mathbf{x}_{t+\delta t}^T \nabla_{xx} V \delta\mathbf{x}_{t+\delta t} \right\rangle$. The term $\delta\mathbf{x}_{t+\delta t}$ is given by the stochastic dynamics in (2.38). Substitution of (2.38) results in 16 terms. To make our analysis clear we classify these 16 terms terms above into five classes. More precisely we will have that:

$$\left\langle \delta\mathbf{x}_{t+\delta t}^T V_{\mathbf{xx}}^T \delta\mathbf{x}_{t+\delta t} \right\rangle = \boldsymbol{\mathcal{E}}_1 + \boldsymbol{\mathcal{E}}_2 + \boldsymbol{\mathcal{E}}_3 + \boldsymbol{\mathcal{E}}_4 + \boldsymbol{\mathcal{E}}_5 \tag{2.47}$$

where the terms $\boldsymbol{\mathcal{E}}_1, \boldsymbol{\mathcal{E}}_2, \boldsymbol{\mathcal{E}}_3, \boldsymbol{\mathcal{E}}_4$ and $\boldsymbol{\mathcal{E}}_5$ are defined as follows:

$$\begin{aligned}
\boldsymbol{\mathcal{E}}_1 &= \left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle + \left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle + \left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle \\
&\quad + \left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle \\
\boldsymbol{\mathcal{E}}_2 &= \left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x} \right\rangle + \left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u} \right\rangle + \left\langle \delta\mathbf{x}^T A_t^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle \\
&\quad + E\left\langle \delta\mathbf{u}^T B_t^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle + \left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle \\
\boldsymbol{\mathcal{E}}_3 &= \left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle + \left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle
\end{aligned}$$

$$\boldsymbol{\mathcal{E}}_4 = \left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle + \left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle + \left\langle \mathbf{O}_d^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle + \left\langle \mathbf{O}_d^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle$$

$$\boldsymbol{\mathcal{E}}_5 = \left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle$$

In the first category we have all these terms that depend neither on $\boldsymbol{\xi}_t$ and nor on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$. These are the terms that define $\boldsymbol{\mathcal{E}}_1$. The second category $\boldsymbol{\mathcal{E}}_2$ includes terms that depend on $\boldsymbol{\xi}_t$ but not on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$. In the third class $\boldsymbol{\mathcal{E}}_3$, there are terms that depends both on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$ and $\boldsymbol{\xi}_t$. In the fourth class $\boldsymbol{\mathcal{E}}_4$, we have terms that depend on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$. Finally in the fifth class $\boldsymbol{\mathcal{E}}_5$, we have all these terms that depend on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$ quadratically. The expectation operator will cancel all the terms that include noise up the first order. Moreover, the mean operator for terms that depend on the noise quadratically will result in covariance.

We compute the expectations of all the terms in the $\boldsymbol{\mathcal{E}}_1$ class. More precisely we will have that:

$$\begin{aligned}
\left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle &= \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \\
\left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle &= \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \\
\left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle &= \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \\
\left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle &= \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t
\end{aligned} \tag{2.48}$$

We continue our analysis by calculating all the terms in the class $\boldsymbol{\mathcal{E}}_2$. More presicely we will have:

$$\begin{aligned}
\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x} \right\rangle &= 0 \\
\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u} \right\rangle &= 0 \\
\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x} \right\rangle^T &= 0 \\
\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u} \right\rangle^T &= 0
\end{aligned} \tag{2.49}$$

The terms above are equal to zero since the brownian noise is zero mean. The expectation of the term that does not depend on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$ and it is quadratic with respect to the noise is given as follows:

$$\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = tr\left(\Gamma_t^T V_{\mathbf{xx}} \Gamma_t \Sigma_\omega\right) \tag{2.50}$$

Since matrix Γ depends on variations in states and controls $\delta\mathbf{x}, \delta\mathbf{u}$ we can further massage the expressions above so that it can be expressed as quadratic functions in $\delta\mathbf{x}, \delta\mathbf{u}$.

$$tr\left(\Gamma_t^T V_{\mathbf{xx}} \Gamma_t \Sigma_\omega\right) = \sigma_{d\omega}^2 \delta t \mathbf{tr} \left(\begin{pmatrix} \Gamma^{(1)T} \\ \cdots \\ \cdots \\ \Gamma^{(m)T} \end{pmatrix} V_{\mathbf{xx}} \begin{pmatrix} \Gamma^{(1)} & \cdots & \cdots & \Gamma^{(m)} \end{pmatrix} \right) \tag{2.51}$$

$$= \sigma_{d\omega}^2 \delta t \sum_{i=1}^{m} \Gamma^{(i)T} V_{\mathbf{xx}} \Gamma^{(i)} \tag{2.52}$$

The last equation is written in the form:

$$tr\left(\Gamma_t^T V_{\mathbf{xx}} \Gamma_t \Sigma_\omega\right) = \delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{F}}} \delta\mathbf{x} + 2\delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{L}}} \delta\mathbf{u} + \delta\mathbf{u}^T \tilde{\boldsymbol{\mathcal{Z}}} \delta\mathbf{u} + 2\delta\mathbf{u}^T \tilde{\boldsymbol{\mathcal{U}}} + 2\delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{S}}} + \gamma \tag{2.53}$$

Where the terms $\tilde{\boldsymbol{\mathcal{F}}} \in \Re^{n\times m}, \tilde{\boldsymbol{\mathcal{L}}} \in \Re^{n\times m}, \tilde{\boldsymbol{\mathcal{Z}}} \in \Re^{m\times m}, \tilde{\boldsymbol{\mathcal{U}}} \in \Re^{m\times 1}, \tilde{\boldsymbol{\mathcal{S}}} \in \Re^{n\times 1}$ and $\gamma \in \Re$ are defined as follows:

$$\tilde{\boldsymbol{\mathcal{F}}} = \sigma^2 \delta t \sum_{i=1}^{m} \nabla_{\mathbf{x}} F_c^{(i)T} V_{\mathbf{xx}} \nabla_{\mathbf{x}} F_c^{(i)} \tag{2.54}$$

$$\tilde{\boldsymbol{\mathcal{L}}} = \sigma^2 \delta t \sum_{i=1}^{m} \nabla_{\mathbf{x}} F_c^{(i)T} V_{\mathbf{xx}} \nabla_{\mathbf{u}} F_c^{(i)} \tag{2.55}$$

$$\tilde{\boldsymbol{\mathcal{Z}}} = \sigma^2 \delta t \sum_{i=1}^{m} \nabla_{\mathbf{u}} F_c^{(i)T} V_{\mathbf{xx}} \nabla_{\mathbf{u}} F_c^{(i)} \tag{2.56}$$

$$\tilde{\boldsymbol{\mathcal{U}}} = \sigma^2 \delta t \sum_{i=1}^{m} \nabla_{\mathbf{u}} F_c^{(i)T} V_{\mathbf{xx}} F_c^{(i)} \tag{2.57}$$

$$\tilde{\boldsymbol{\mathcal{S}}} = \sigma^2 \delta t \sum_{i=1}^{m} \nabla_{\mathbf{x}} F_c^{(i)T} V_{\mathbf{xx}} F_c^{(i)} \tag{2.58}$$

$$\gamma = \sigma^2 \delta t \sum_{i=1}^{m} F_c^{(i)T} V_{\mathbf{xx}} F_c^{(i)} \tag{2.59}$$

For those terms that depend both on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$ and on the noise class $\boldsymbol{\mathcal{E}}_3$ we will have:

$$\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = \left\langle tr\left(V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \mathbf{O}_d^T\right) \right\rangle = tr\left(V_{\mathbf{xx}} \Gamma_t E\left(\boldsymbol{\xi}_t \mathbf{O}_d^T\right)\right) \tag{2.60}$$

By writing the term $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \boldsymbol{\xi}_t, \delta t)$ in a matrix form and putting the noise vector insight the this matrix we have:

$$\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = tr\left(V_{\mathbf{xx}} \Gamma_t E \left[\begin{array}{ccc} \boldsymbol{\xi}_t O^{(1)} & ... & \boldsymbol{\xi}_t O^{(n)} \end{array} \right] \right) \tag{2.61}$$

Calculation of the expectation above requires to find the terms $\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t O^{(j)} \right\rangle$ more precisely we will have:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t O^{(j)} \right\rangle = \frac{1}{2}\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \Phi^{(i)}_{\mathbf{xx}} \delta\mathbf{x} \right\rangle + \frac{1}{2}\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{u}^T \Phi^{(i)}_{\mathbf{uu}} \delta\mathbf{u} \right\rangle + \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{u}^T \Phi^{(i)}_{\mathbf{ux}} \delta\mathbf{x} \right\rangle \tag{2.62}$$

We first calculate the term:

$$\begin{aligned} \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \Phi^{(i)} \delta\mathbf{x} \right\rangle &= \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\nabla_{\mathbf{xx}} f^{(i)} \delta t + \nabla_{\mathbf{xx}} F_r^{(i)} \boldsymbol{\xi}_t \sqrt{\delta t} \right) \delta\mathbf{x} \right\rangle \\ &= \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\nabla_{\mathbf{xx}} f^{(i)} \delta t \right) \delta\mathbf{x} \right\rangle + \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\nabla_{\mathbf{xx}} F_r^{(i)} \boldsymbol{\xi}_t \sqrt{\delta t} \right) \delta\mathbf{x} \right\rangle \end{aligned} \tag{2.63}$$

The term $\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\nabla_{\mathbf{xx}} f^{(i)} \delta t \right) \delta\mathbf{x} \right\rangle = 0$ since it depends linearly on the noise and $\left\langle \boldsymbol{\xi}_t \right\rangle = 0$. The second term $\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\nabla_{\mathbf{xx}} F_r^{(i)} \boldsymbol{\xi}_t \sqrt{\delta t} \right) \delta\mathbf{x} \right\rangle$ depends quadratically in the noise and thus the expectation operator will result in the variance on the noise. We follow the analysis:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \left(F_r^{(i)} \boldsymbol{\xi}_t \sqrt{\delta t} \right) \delta\mathbf{x} \right\rangle \tag{2.64}$$

Since the $\boldsymbol{\xi}_t = \left(\xi^{(1)}, ..., \xi^{(m)}\right)^T$ and $F_r^{(i)} = \left(F^{(i1)}, ..., F^{(im)}\right)$ we will have that:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \left\langle \delta t \boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \left(\sum_{j=1}^{m} F^{(ij)} \xi^{(j)} \right) \delta\mathbf{x} \right\rangle \tag{2.65}$$

$$= \left\langle \delta t \boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\sum_{j=1}^{m} \nabla_{\mathbf{xx}} \left(F^{(ij)} \xi^{(j)} \right) \right) \delta\mathbf{x} \right\rangle$$

$$= \left\langle \delta t \boldsymbol{\xi}_t \delta\mathbf{x}^T \left(\sum_{j=1}^{m} \xi^{(j)} \nabla_{\mathbf{xx}} \left(F^{(ij)} \right) \right) \delta\mathbf{x} \right\rangle$$

By writing $\boldsymbol{\xi}_t$ in vector form we have that:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \left\langle \delta t \begin{bmatrix} \xi^{(1)} \\ \cdots \\ \cdots \\ \xi^{(m)} \end{bmatrix} \delta\mathbf{x}^T \left(\sum_{j=1}^{m} \xi^{(j)} \nabla_{\mathbf{xx}} \left(F^{(ij)} \right) \right) \delta\mathbf{x} \right\rangle$$

The term $\delta\mathbf{x}^T \left(\sum_{j=1}^{m} \xi^{(j)} \nabla_{\mathbf{xx}} \left(F^{(ij)} \right) \right) \delta\mathbf{x}$ is scalar and it can multiply each one of the elements of the noise vector.

$$\begin{bmatrix} \delta t \left\langle \xi^{(1)} \delta\mathbf{x}^T \left(\sum_{j=1}^{m} \xi^{(j)} \nabla_{\mathbf{xx}} \left(F^{(ij)} \right) \right) \delta\mathbf{x} \right\rangle \\ \cdots \\ \cdots \\ \delta t \left\langle \xi^{(m)} \delta\mathbf{x}^T \left(\sum_{j=1}^{m} \xi^{(j)} \nabla_{\mathbf{xx}} \left(F^{(ij)} \right) \right) \delta\mathbf{x} \right\rangle \end{bmatrix} \tag{2.66}$$

Since $\left\langle \xi^{(i)}\xi^{(i)} \right\rangle = \sigma^2$ and $\left\langle \xi^{(i)}\xi^{(j)} \right\rangle = 0$ we can show that:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{xx}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \sigma^2 \delta t \begin{bmatrix} \delta\mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(i1)} \delta\mathbf{x} \\ \cdots \\ \cdots \\ \delta\mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(im)} \delta\mathbf{x} \end{bmatrix} \tag{2.67}$$

In a similar way we can show that:

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{x}^T \nabla_{\mathbf{uu}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \sigma^2 \delta t \begin{bmatrix} \delta\mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(i1)} \delta\mathbf{u} \\ \cdots \\ \cdots \\ \delta\mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(im)} \delta\mathbf{u} \end{bmatrix} \tag{2.68}$$

and

$$\left\langle \sqrt{\delta t}\boldsymbol{\xi}_t \delta\mathbf{u}^T \nabla_{\mathbf{xu}} \Phi^{(i)} \delta\mathbf{x} \right\rangle = \sigma^2 \delta t \begin{bmatrix} \delta\mathbf{u}^T \nabla_{\mathbf{ux}} F_r^{(i1)} \delta\mathbf{x} \\ \cdots \\ \cdots \\ \delta\mathbf{u}^T \nabla_{\mathbf{ux}} F_r^{(im)} \delta\mathbf{x} \end{bmatrix} \tag{2.69}$$

Since we have calculated all the terms of expression (2.62) we can proceed with the computation of (2.60). According to the analysis above the term $\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle$ can be written as follows:

$$\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = tr\left(V_{\mathbf{xx}} \Gamma_t \left(\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{G}} \right) \right) \tag{2.70}$$

Where the matrices $\boldsymbol{\mathcal{M}} \in \Re^{m\times n}$, $\boldsymbol{\mathcal{N}} \in \Re^{m\times n}$ and $\boldsymbol{\mathcal{G}} \in \Re^{m\times n}$ are defined as follows:

$$\boldsymbol{\mathcal{M}} = \sigma^2 \delta t \begin{bmatrix} \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(11)} \delta \mathbf{x} & ... & \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(1n)} \delta \mathbf{x} \\ ... & ... & ... \\ \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(m1)} \delta \mathbf{x} & ... & \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F_r^{(mn)} \delta \mathbf{x} \end{bmatrix} \tag{2.71}$$

Similarly

$$\boldsymbol{\mathcal{N}} = \sigma^2 \delta t \begin{bmatrix} \delta \mathbf{x}^T \nabla_{\mathbf{xu}} F_r^{(1,1)} \delta \mathbf{u} & ... & \delta \mathbf{x}^T \nabla_{\mathbf{xu}} F_r^{(1,n)} \delta \mathbf{u} \\ ... & ... & ... \\ \delta \mathbf{x}^T \nabla_{\mathbf{xu}} F_r^{(m,1)} \delta \mathbf{u} & ... & \delta \mathbf{x}^T \nabla_{\mathbf{xu}} F_r^{(m,n)} \delta \mathbf{u} \end{bmatrix} \tag{2.72}$$

and

$$\boldsymbol{\mathcal{G}} = \sigma^2 \delta t \begin{bmatrix} \delta \mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(1,1)} \delta \mathbf{u} & ... & \delta \mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(1,n)} \delta \mathbf{u} \\ ... & ... & ... \\ \delta \mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(m,1)} \delta \mathbf{u} & ... & \delta \mathbf{u}^T \nabla_{\mathbf{uu}} F_r^{(m,n)} \delta \mathbf{u} \end{bmatrix} \tag{2.73}$$

Based on (2.34) the term Γ_t depends on Δ which is a function of the variations in states and control up to the 1th order. In addition the matrices $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{G}}$ are also functions of the deviations in state and controls up to the 2th order. The product of Δ with each one of the matrices $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{G}}$ will result into 3th order terms that can be neglected. By neglecting these terms we can show that:

$$\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = tr\left(V_{\mathbf{xx}} (\Delta + F) \left(\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{G}} \right) \right) \tag{2.74}$$

$$= tr\left(V_{\mathbf{xx}} F\left(\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{G}}\right)\right)$$

Each element (i,j) of the product $\boldsymbol{\mathcal{C}} = V_{\mathbf{xx}} F$ can be expressed as $\boldsymbol{\mathcal{C}}^{(i,j)} = \sum_{r=1}^{n} V_{\mathbf{xx}}^{(i,r)} F^{(r,j)}$ where $\boldsymbol{\mathcal{C}} \in \Re^{n \times p}$. Furthermore the element (μ,ν) of the product $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}$ is formulated $\boldsymbol{\mathcal{H}}^{(\mu,\nu)} = \sum_{k=1}^{n} \boldsymbol{\mathcal{C}}^{(\mu,k)} \boldsymbol{\mathcal{M}}^{(k,\nu)}$ with $\boldsymbol{\mathcal{H}} \in \Re^{n \times n}$. Thus, the term $tr\left(V_{\mathbf{xx}} F \boldsymbol{\mathcal{M}}\right)$ can be now expressed as:

$$\begin{aligned} tr\left(V_{\mathbf{xx}} F \boldsymbol{\mathcal{M}}\right) &= \sum_{\ell=1}^{n} \boldsymbol{\mathcal{H}}^{(\ell,\ell)} \\ &= \sum_{\ell=1}^{n} \sum_{k=1}^{m} \boldsymbol{\mathcal{C}}^{(\ell,k)} \boldsymbol{\mathcal{M}}^{(k,\ell)} \\ &= \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \boldsymbol{\mathcal{M}}^{(k,\ell)} \end{aligned} \tag{2.75}$$

Since $\boldsymbol{\mathcal{M}}^{(k,\ell)} = \delta t \sigma_{d\omega_1}^2 \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F^{(k,\ell)} \delta \mathbf{x}$ the vectors $\delta t \sigma_{d\omega_1}^2 \delta \mathbf{x}^T$ and $\delta \mathbf{x}$ do not depend on k, ℓ, r and they can be taken outside the sum. Thus we can show that:

$$\begin{aligned} tr\left(V_{\mathbf{xx}} F \boldsymbol{\mathcal{M}}\right) &= \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \sigma^2 \delta t \delta \mathbf{x}^T \nabla_{\mathbf{xx}} F^{(k,\ell)} \delta \mathbf{x} \right) \\ &= \delta \mathbf{x}^T \sigma^2 \delta t \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \nabla_{\mathbf{xx}} F^{(k,\ell)} \right) \delta \mathbf{x} \\ &= \delta \mathbf{x}^T \tilde{\mathbf{M}} \delta \mathbf{x} \end{aligned} \tag{2.76}$$

where $\tilde{\mathbf{M}}$ is a matrix of dimensionality $\tilde{\mathbf{M}} \in \Re^{n \times n}$ and it is defined as:

$$\tilde{\mathbf{M}} = \sigma^2 \delta t \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \nabla_{\mathbf{xx}} F^{(k,\ell)} \right) \tag{2.77}$$

By following the same algebraic steps it can been shown that:

$$tr\left(V_{\mathbf{xx}} F \boldsymbol{\mathcal{N}}\right) = \delta \mathbf{x}^T \tilde{\mathbf{N}} \delta \mathbf{u} \tag{2.78}$$

with $\tilde{\mathbf{N}}$ matrix of dimensionality $\tilde{\mathbf{N}} \in \Re^{n \times m}$ defined as:

$$\tilde{\mathbf{N}} = \sigma^2 \delta t \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \nabla_{\mathbf{xu}} F^{(k,\ell)} \right) \tag{2.79}$$

and

$$tr\left(V_{\mathbf{xx}} F \boldsymbol{\mathcal{G}}\right) = \delta \mathbf{u}^T \tilde{\mathbf{G}} \delta \mathbf{u} \tag{2.80}$$

with $\tilde{\mathbf{G}}$ matrix of dimensionality $\tilde{\mathbf{N}} \in \Re^{m \times m}$ defined as:

$$\tilde{\mathbf{G}} = \sigma^2 \delta t \sum_{\ell=1}^{n} \sum_{k=1}^{m} \left(\left(\sum_{r=1}^{n} V_{\mathbf{xx}}^{(k,r)} F^{(r,\ell)} \right) \nabla_{\mathbf{uu}} F^{(k,\ell)} \right) \tag{2.81}$$

Thus the term $\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle$ is formulated as:

$$\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \Gamma_t \boldsymbol{\xi}_t \right\rangle = \frac{1}{2}\delta\mathbf{x}^T \tilde{\mathbf{M}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{u}^T \tilde{\mathbf{G}} \delta\mathbf{u} + \delta\mathbf{x}^T \tilde{\mathbf{N}} \delta\mathbf{u} \tag{2.82}$$

Similarly we can show that:

$$\left\langle \boldsymbol{\xi}_t^T \Gamma_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle = \frac{1}{2}\delta\mathbf{x}^T \tilde{\mathbf{M}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{u}^T \tilde{\mathbf{G}} \delta\mathbf{u} + \delta\mathbf{x}^T \tilde{\mathbf{N}} \delta\mathbf{u} \tag{2.83}$$

Next we will find the expectation for all terms that depend on $\mathbf{O}_d(\delta\mathbf{x}, \delta\mathbf{u}, \mathbf{d}\omega, \delta t)$ and not on the noise. Consequently, we will have that:

$$\begin{aligned}
\left\langle \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle &= \delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} \tilde{\mathbf{O}}_d = 0 \\
\left\langle \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle &= \delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} \tilde{\mathbf{O}}_d = 0 \\
\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t \right\rangle &= \tilde{\mathbf{O}}_d{}^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t = 0 \\
\left\langle \mathbf{O}_d^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \right\rangle &= \tilde{\mathbf{O}}_d{}^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t = 0
\end{aligned} \tag{2.84}$$

where the quantity $\tilde{\mathbf{O}}_d$ has been defined in (2.42). All the 4 terms above are equal to zero since they have variations in state and control of the order higher than 2 and therefore they can be neglected.

Finally we compute the terms of the 5th class and therefore we have the expression

$$\boldsymbol{\mathcal{E}}_5 = \left\langle \mathbf{O}_d^T V_{\mathbf{xx}} \mathbf{O}_d \right\rangle = \left\langle tr\left(V_{\mathbf{xx}} \mathbf{O}_d \mathbf{O}_d^T\right) \right\rangle = tr\left(V_{\mathbf{xx}} \left\langle \mathbf{O}_d \mathbf{O}_d^T \right\rangle\right) \tag{2.85}$$

$$= \left(V_{\mathbf{xx}} \left\langle \begin{bmatrix} O^{(1)} \\ \dots \\ O^{(n)} \end{bmatrix} \begin{bmatrix} O^{(1)} \\ \dots \\ O^{(n)} \end{bmatrix}^T \right\rangle \right)$$

The product $O^{(i)}O^{(j)}$ is a function of variation in state and control of order 4 since each term $O^{(i)}$ is a function of variation in states and control of order 2. Consequently, the term $\boldsymbol{\mathcal{E}}_5 = E\left(\mathbf{O}_d^T V_{\mathbf{xx}} \mathbf{O}_d\right)$ is equal to zero.

With the computation of the expectation of term that is quadratic WRT $\mathbf{O}_d$ we have calculated all the terms of the second order expansion of the cost to go function. In the next section we derive the optimal controls and we present the SDDP algorithm. Furthermore we show how SDDP recover the deterministic solution as well as the cases of only control multiplicative, only state multiplicative and only additive noise.

2.3.1.2 Optimal controls

In this section we provide the form of the optimal controls and we show how previous results are special cases of our generalized stochastic DDP formulation. Furthermore after we computed all the terms of expansion of the cost to go function $V(\mathbf{x}_t)$ at state$\mathbf{x}_t$

we show that its form remains quadratic WRT variations in state $\delta\mathbf{x}_t$ under the constraint of the nonlinear stochastic dynamics in (2.30). More precisely we have that:

$$\begin{aligned}
&V(\bar{\mathbf{x}}_{t+\delta t} + \delta\mathbf{x}_{t+\delta t}) = V(\bar{\mathbf{x}}_{t+\delta t}) + \nabla_x V^T A_t \delta\mathbf{x}_t + \nabla_x V^T B_t \delta\mathbf{u}_t \\
&+ \frac{1}{2}\delta\mathbf{x}^T \boldsymbol{\mathcal{F}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{u}^T \boldsymbol{\mathcal{Z}} \delta\mathbf{u} + \delta\mathbf{u}^T \boldsymbol{\mathcal{L}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t + \frac{1}{2}\delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t \\
&+ \frac{1}{2}\delta\mathbf{x}_t^T A_t^T V_{\mathbf{xx}} B_t \delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{u}_t^T B_t^T V_{\mathbf{xx}} A_t \delta\mathbf{x}_t + \frac{1}{2}\delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{F}}} \delta\mathbf{x} + \delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{L}}} \delta\mathbf{u} + \frac{1}{2}\delta\mathbf{u}^T \tilde{\boldsymbol{\mathcal{Z}}} \delta\mathbf{u} \\
&+ \delta\mathbf{u}^T \tilde{\boldsymbol{\mathcal{U}}} + \delta\mathbf{x}^T \tilde{\boldsymbol{\mathcal{S}}} + \frac{1}{2}\gamma + \frac{1}{2}\delta\mathbf{x}^T \tilde{\mathbf{M}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{u}^T \tilde{\mathbf{G}} \delta\mathbf{u} + \delta\mathbf{x}^T \tilde{\mathbf{N}} \delta\mathbf{u}
\end{aligned} \tag{2.86}$$

The unmaximized state, action value function is defined as follows:

$$Q(\mathbf{x}_k, \mathbf{u}_k) = \ell(\mathbf{x}_k, \mathbf{u}_k) + V(\mathbf{x}_{k+1}) \tag{2.87}$$

Given a trajectory in states and controls $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ we can approximate the state action value function as follows:

$$Q(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}) = Q_0 + \delta\mathbf{u}^T Q_{\mathbf{u}} + \delta\mathbf{x}^T Q_{\mathbf{x}} + \frac{1}{2}\begin{bmatrix} \delta\mathbf{x}^T & \delta\mathbf{u}^T \end{bmatrix} \begin{bmatrix} Q_{\mathbf{xx}} & Q_{\mathbf{xu}} \\ Q_{\mathbf{ux}} & Q_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \tag{2.88}$$

By equating the coefficients with similar powers between the state action value function $Q(\mathbf{x}_k, \mathbf{u}_k)$ and the immediate reward and cost to go $\ell(\mathbf{x}_k, \mathbf{u}_k)$ and $V(\mathbf{x}_{k+1})$ respectively we can show that:

$$\begin{aligned} Q_{\mathbf{x}} &= \ell_{\mathbf{x}} + A_t V_{\mathbf{x}} + \tilde{\boldsymbol{\mathcal{S}}} \\ Q_{\mathbf{u}} &= \ell_{\mathbf{u}} + A_t V_{\mathbf{x}} + \tilde{\boldsymbol{\mathcal{U}}} \\ Q_{\mathbf{xx}} &= \ell_{\mathbf{xx}} + A_t^T V_{\mathbf{xx}} A_t + \boldsymbol{\mathcal{F}} + \tilde{\boldsymbol{\mathcal{F}}} + \tilde{\mathbf{M}} \\ Q_{\mathbf{xu}} &= \ell_{\mathbf{xu}} + A_t^T V_{\mathbf{xu}} B_t + \boldsymbol{\mathcal{L}} + \tilde{\boldsymbol{\mathcal{L}}} + \tilde{\mathbf{N}} \\ Q_{\mathbf{uu}} &= \ell_{\mathbf{uu}} + B_t^T V_{\mathbf{uu}} B_t + \boldsymbol{\mathcal{Z}} + \tilde{\boldsymbol{\mathcal{Z}}} + \tilde{\mathbf{G}} \end{aligned} \tag{2.89}$$

where we have assumed a local quadratic approximation of the immediate reward $\ell(\mathbf{x}_k, \mathbf{u}_k)$ according to the equation:

$$\ell(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}) = \ell_0 + \delta\mathbf{u}^T \ell_{\mathbf{u}} + \delta\mathbf{x}^T \ell_{\mathbf{x}} + \frac{1}{2} \begin{bmatrix} \delta\mathbf{x}^T & \delta\mathbf{u}^T \end{bmatrix} \begin{bmatrix} \ell_{\mathbf{xx}} & \ell_{\mathbf{xu}} \\ \ell_{\mathbf{ux}} & \ell_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \tag{2.90}$$

with $\ell_{\mathbf{x}} = \frac{\partial \ell}{\partial \mathbf{x}}$, $\ell_{\mathbf{u}} = \frac{\partial \ell}{\partial \mathbf{u}}$, $\ell_{\mathbf{xx}} = \frac{\partial^2 \ell}{\partial \mathbf{x}^2}$, $\ell_{\mathbf{uu}} = \frac{\partial^2 \ell}{\partial \mathbf{u}^2}$ and $\ell_{\mathbf{ux}} = \frac{\partial^2 \ell}{\partial \mathbf{u} \partial \mathbf{x}}$. The local variations in control $\delta\mathbf{u}^*$ that maximize the state action value function are expressed by the equation that follows:

$$\delta\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmax}}\, Q(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}) = -Q_{\mathbf{uu}}^{-1} \left(Q_{\mathbf{u}} + Q_{\mathbf{ux}} \delta\mathbf{x} \right) \tag{2.91}$$

The optimal control variations have the form $\delta\mathbf{u}^* = \mathbf{l} + \mathbf{L}\delta\mathbf{x}$ where $\mathbf{l} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}}$ is the open loop control or feedforward command and $\mathbf{L} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}}$ is the closed loop - feedback gain. All the terms in (2.92) are functions of the gradient of the value function

$V_{\mathbf{x}}$ and the Hessian $V_{\mathbf{x},\mathbf{x}}$. This quantities are backward propagated from the terminal boundary conditions as follows:

If the noise is only control depended then $\tilde{\mathbf{M}}, \tilde{\mathbf{N}}, \tilde{\boldsymbol{\mathcal{L}}}, \tilde{\boldsymbol{\mathcal{F}}}, \tilde{\boldsymbol{\mathcal{S}}}$ will be zero since $\nabla_{\mathbf{xx}} F(\mathbf{u}) = 0, \nabla_{\mathbf{xu}} F(\mathbf{u}) = 0$ and $\nabla_{\mathbf{x}} F_c^{(i)}(\mathbf{x}) = 0$ while if it is state dependent then $\tilde{\mathbf{N}}, \tilde{\mathbf{G}}, \tilde{\boldsymbol{\mathcal{Z}}}, \tilde{\boldsymbol{\mathcal{L}}}, \tilde{\boldsymbol{\mathcal{U}}}$ will be zero since $\nabla_{\mathbf{xu}} F(\mathbf{x}) = 0, \nabla_{\mathbf{uu}} F(\mathbf{x}) = 0$ and $\nabla_{\mathbf{u}} F_c^{(i)}(\mathbf{x}) = 0$.

In the next two sub-sections we show that differential dynamic programming (DDP) and iterative linear quadratic regulators are special cases of the stochastic differential dynamic programming.

2.3.2 Differential dynamic programming

There are two cases in which we can recover the DDP equations. In particular, for the special case where the stochastic dynamics have only additive noise $F(\mathbf{u}, \mathbf{x}) = F$ then the terms $\tilde{\mathbf{M}}, \tilde{\mathbf{N}}, \tilde{\mathbf{G}}, \tilde{\boldsymbol{\mathcal{F}}}, \tilde{\boldsymbol{\mathcal{L}}}, \tilde{\boldsymbol{\mathcal{Z}}}, \tilde{\boldsymbol{\mathcal{U}}}, \tilde{\boldsymbol{\mathcal{S}}}$ will be zero since they are functions of $\nabla_{\mathbf{xx}} F$ and $\nabla_{\mathbf{xu}} F$ and $\nabla_{\mathbf{uu}} F$ and it holds that $\nabla_{\mathbf{xx}} F = 0$, $\nabla_{\mathbf{xu}} F = 0$ and $\nabla_{\mathbf{uu}} F = 0$. In systems with additive noire the control does not depend on the statistical characteristics of the noise.

In addition, for the case of deterministic systems the terms $\tilde{\mathbf{M}}, \tilde{\mathbf{N}}, \tilde{\mathbf{G}}, \tilde{\boldsymbol{\mathcal{F}}}, \tilde{\boldsymbol{\mathcal{L}}}, \tilde{\boldsymbol{\mathcal{Z}}}, \tilde{\boldsymbol{\mathcal{U}}}, \tilde{\boldsymbol{\mathcal{S}}}$ will be zero because these terms depend on the variance of the noise $\sigma_{\mathbf{d}\omega_i} = 0$, $\forall i = 1, ..., m$. Clearly in both of the cases above the resulting algorithm corresponds to DDP in which the equation are formulated as follows:

$$
\begin{aligned}
Q_{\mathbf{x}} &= \ell_{\mathbf{x}} + A_t V_{\mathbf{x}} \\
Q_{\mathbf{u}} &= \ell_{\mathbf{u}} + A_t V_{\mathbf{x}} \\
Q_{\mathbf{xx}} &= \ell_{\mathbf{xx}} + A_t^T V_{\mathbf{xx}} A_t \\
Q_{\mathbf{xu}} &= \ell_{\mathbf{xu}} + A_t^T V_{\mathbf{xu}} B_t \\
Q_{\mathbf{uu}} &= \ell_{\mathbf{uu}} + B_t^T V_{\mathbf{uu}} B_t
\end{aligned}
\tag{2.92}
$$

2.4 Risk sensitivity and differential game theory

The relation of risk sensitivity and differential game theory was first studied for the case on linear dynamics and quadratic cost function in (Jacobson 1973) When the case of imperfect state measurement is considered, the relation between the two frameworks was investigated in (Whittle 1991),(Whittle 1990),(Basar 1991)and (Runolfsson 1994). Another research direction on risk sensitivity and differential game theory considers the case of nonlinear stochastic dynamics and markov processes (James, Baras & Elliot 1994),(Fleming & Soner 2006). In addition to the theoretical developments, applications of risk sensitivity and differential game theoretic approaches to reinforcement learning showed the robustness of the resulting control policies against disturbances and uncertainty in highly nonlinear systems (Morimoto & Atkeson 2002),(Morimoto & Doya 2005). One of the main issues with these risk sensitive RL approaches is their poor scalability to high dimensional dynamical systems.

2.4.1 Stochastic differential games

The ultimate goal in stochastic optimal control framework (Stengel 1994), (Basar & Berhard 1995), (Fleming & Soner 2006) is to control a dynamical system while minimizing a performance criterion. For the case on nonlinear stochastic systems the stochastic optimal control problem is expressed as the minimization of a cost function. However when disturbances are present then the stochastic optimal control problem can be formulated as a differential game with two opponents that is formulated as:

$$\min_{\mathbf{u}} \max_{\mathbf{v}} J(\mathbf{x}, \mathbf{u}) = \min_{\mathbf{u}} \max_{\mathbf{v}} \left\langle \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, \mathbf{v}) dt \right\rangle \tag{2.93}$$

with $\boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = q(\mathbf{x}) + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u} - \gamma\mathbf{v}^T\mathbf{v}$ and under the stochastic nonlinear dynamical constrains:

$$\mathbf{dx} = (f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}) + \sqrt{\frac{\epsilon}{\gamma}}\mathbf{C}(\mathbf{x})\mathbf{L}\left(\mathbf{v}dt + d\mathbf{w}\right) \tag{2.94}$$

or

$$\mathbf{dx} = F(\mathbf{x}, \mathbf{u}, \mathbf{v})dt + \mathbf{C}(\mathbf{x})\mathbf{L}d\mathbf{w} \tag{2.95}$$

with $F(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} + \sqrt{\frac{\epsilon}{\gamma}}\mathbf{C}(\mathbf{x})\mathbf{L}\mathbf{v}$ and $\mathbf{L}$ is a state independent matrix defined as $\mathbf{L}\mathbf{L}^T = \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$. Essentially there are two controllers, $\mathbf{u} \in \Re^{m\times 1}$ the stabilizing controller and $\mathbf{v} \in \Re^{p\times 1}$ the destabilizing one while $\mathbf{x} \in \Re^{n\times 1}$ is the state and $d\mathbf{w}$ is brownian noise. The parameters ϵ, γ are positive. The stabilizing controller minimizes the cost function while the stabilizing one intents to maximize it. The value function is

defined as the optimal of the cost function $J(\mathbf{x}, \mathbf{u}, \mathbf{v})$ and therefore is a function only of the state:

$$V(\mathbf{x}) = \min_{\mathbf{u}} \max_{\mathbf{v}} J(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \min_{\mathbf{u}} \max_{\mathbf{v}} J(\mathbf{x}, \mathbf{u}^*, \mathbf{v}^*) \tag{2.96}$$

The stochastic Isaacs HJB equation associated with this stochastic optimal control problem is expressed as follows:

$$-\partial_t V = \min_{\mathbf{u}} \max_{\mathbf{v}} \left(\boldsymbol{\mathcal{L}} + (\nabla_{\mathbf{x}} V)^T \mathbf{F} + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V) \mathbf{C} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \mathbf{C}^T \right) \right) \tag{2.97}$$

Since the the left hand side of the HJB is convex with respect to control $\mathbf{u}$ and concave with respect to $\mathbf{v}$ the min and max are exact lead to the optimal controls:

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T (\nabla_{\mathbf{x}} V) \tag{2.98}$$

and optimal destabilizing controller:

$$\mathbf{v}^*(\mathbf{x}) = \frac{1}{\gamma} \mathbf{C}(\mathbf{x})^T (\nabla_{\mathbf{x}} V) \tag{2.99}$$

Substitution of the optimal stabilizing and destabilizing control to the HJB results into the nonlinear second order PDE:

$$\begin{aligned} -\partial_t V = q + (\nabla_{\mathbf{x}} V)^T \mathbf{f}_t - \frac{1}{2} (\nabla_{\mathbf{x}} V)^T \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T (\nabla_{\mathbf{x}} V) - \frac{1}{2\gamma} (\nabla_{\mathbf{x}} V)^T \mathbf{B} \mathbf{B}^T (\nabla_{\mathbf{x}} V) \\ + \frac{\epsilon}{2\gamma} tr \left((\nabla_{\mathbf{xx}} V) \mathbf{C} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \mathbf{C}^T \right) \end{aligned} \tag{2.100}$$

The PDE above can be written in the form:

$$-\partial_t V = q + (\nabla_{\mathbf{x}} V)^T \mathbf{f} - \frac{1}{2}(\nabla_{\mathbf{x}} V)^T \boldsymbol{\mathcal{H}}(\mathbf{x})(\nabla_{\mathbf{x}} V) + \frac{\epsilon}{2\gamma} tr\left((\nabla_{\mathbf{xx}} V)\mathbf{C}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}^T\right) \tag{2.101}$$

where the introduced term $\boldsymbol{\mathcal{H}}(\mathbf{x})$ is defined as

$$\boldsymbol{\mathcal{H}}(\mathbf{x}) = \mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T - \frac{1}{\gamma}\mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T \tag{2.102}$$

From (2.101) and (2.102) we see that for $\gamma \to \infty$ the Isaacs HJB is reduced to the HJB. In the next section we show under which conditions the nonlinear and second order PDE can be transformed to a linear PDE. Linear PDEs are easier to be solve via the application of the Feynman- Kac lemmas which provides a probabilistic representations of the solution of these PDEs. In the next session after transforming the PDE into a linear, we provide the Feynman Kac lemma.

2.4.2 Risk sensitive optimal control

We consider the optimal control problem (Basar & Berhard 1995) where the state dynamics are described by the Ito stochastic differential differential equation:

$$\mathbf{dx} = (\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u})dt + \sqrt{\frac{\epsilon}{\gamma}}\tilde{\mathbf{C}}(\mathbf{x})\mathbf{L}d\mathbf{w} \tag{2.103}$$

where $\mathbf{w}(t)$, $t > 0$ is an n-dimensional Wiener Precess, γ is a positive parameter and $\mathbf{u}_t \in U$ is the control and $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{L}\mathbf{L}^T$. The objective is to find the control law that minimizes the performance criterion:

$$J(\mathbf{x}, \mathbf{u}) = \epsilon \log \left\langle \exp \frac{1}{\epsilon} \left(\phi(t_N) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) dt \right) \right\rangle \tag{2.104}$$

For our analysis we need the following conditions:

i) Functions $\mathbf{f}(\mathbf{x}), \mathbf{G}(\mathbf{x})$ and $\boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u})$ are continuously differentiable in $(t, \mathbf{x}, \mathbf{u}) \in [0, t_f] \times \Re^n \times U$, ϕ is twice differentiable in $\mathbf{x} \in \Re^n$ and ϕ and $\boldsymbol{\mathcal{L}}$ are nonnegative.

ii) $\mathbf{C}(\mathbf{x})$ is continuously differentiable in $(t, \mathbf{x}) \in [0, t_f] \times \Re^n \times U$ and $\mathbf{C}(\mathbf{x})\mathbf{C}(\mathbf{x})^T > 0$.

iii) $\mathbf{F}(\mathbf{x}, \mathbf{u}), \nabla_{\mathbf{x}}\mathbf{F}, \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}), \boldsymbol{\mathcal{L}}_{\mathbf{x}}, \phi(\mathbf{x}(t_f)), \nabla_{\mathbf{x}}\phi$ are bounded on $[0, t_f] \times \Re^n \times U$.

iv) U is closed and bounded subset of $\Re^m$

Let us assume that:

$$V(\mathbf{x}, t) = \inf_{\mathbf{u}} J(\mathbf{x}, \mathbf{u}) = \epsilon \log \Phi(\mathbf{x}, t) \tag{2.105}$$

where $\Phi(t, \mathbf{x})$ is the value function that corresponds to the cost function:

$$\Phi(t, \mathbf{x}) = \inf_{\mathbf{u}} \left\langle \exp \frac{1}{\epsilon} \left(\phi(t_N) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) dt \right) \right\rangle \tag{2.106}$$

or

$$\Phi(t, \mathbf{x}) = \left\langle \exp \frac{1}{\epsilon} \left(\phi(t_N) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*) dt \right) \right\rangle \tag{2.107}$$

where $\mathbf{x}^*, \mathbf{u}^*$ is the optimal state and control trajectory. The total derivative $\forall t = t_0$ is given by:

$$\frac{d\Phi}{dt} = -\frac{1}{\epsilon}\boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*)\Phi(t, \mathbf{x}) \tag{2.108}$$

and thus we have that:

$$d\Phi = -\frac{1}{\epsilon}\boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*)\Phi(t, \mathbf{x})dt \tag{2.109}$$

By using the Ito differentiation rule we will have that:

$$d\Phi = \left(\partial_t\Phi + (\nabla_{\mathbf{x}}\Phi)^T\mathbf{F}\right)dt + \frac{\epsilon}{2\gamma}tr\left(\nabla\Phi_{\mathbf{xx}}\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) \tag{2.110}$$

By equating the two last equation above we will the resulting PDE expressed as follows:

$$-\partial_t\Phi = (\nabla_{\mathbf{x}}\Phi)^T\mathbf{F} + \frac{\epsilon}{2\gamma}tr\left(\nabla_{\mathbf{xx}}\Phi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) + \frac{1}{\epsilon}\boldsymbol{\mathcal{L}}\Phi \tag{2.111}$$

In this PDE $\mathbf{F} = \mathbf{F}(\mathbf{x}^*, \mathbf{u}^*)$ and $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}(\mathbf{x}^*, \mathbf{u}^*)$. The PDE above can be also written as follows:

$$0 = \inf_{\mathbf{u}\in U}\left(\partial_t\Phi + (\nabla_{\mathbf{x}}\Phi)^T\mathbf{F} + \frac{\epsilon}{2\gamma}tr\left(\nabla_{\mathbf{xx}}\Phi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) + \frac{1}{\epsilon}\boldsymbol{\mathcal{L}}\Psi\right)$$

or in form:

$$\partial_t\Phi_t = \inf_{\mathbf{u}\in U}\left((\nabla_{\mathbf{x}}\Phi)^T\mathbf{F} + \frac{\epsilon}{2\gamma}tr\left(\nabla_{\mathbf{xx}}\Phi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) + \frac{1}{\epsilon}\boldsymbol{\mathcal{L}}\Phi\right)$$

with the boundary condition $\Phi(\mathbf{x},t) = \exp\left(\frac{1}{\epsilon}\phi(\mathbf{x}(t_N))\right)$ and $\mathbf{F} = \mathbf{F}(\mathbf{x}^*,\mathbf{u})$ and $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}(\mathbf{x}^*,\mathbf{u})$. This is the Hamilton Jacobi Bellman equation for the case of the risk sensitive stochastic optimal control problem. Since $\mathbf{C}\mathbf{C}^T > 0$ the PDE above is a uniformly parabolic PDE. Moreover under the conditions $1,2,3,4$ the second order PDE has unique bounded positive solution. The value function $V(\mathbf{x},t)$ is related to $\Phi(\mathbf{x},t)$ through (2.105) and therefore $V(\mathbf{x},t)$ is smooth and satisfies the uniformly parabolic PDE:

$$\partial_t V_t = \inf_{\mathbf{u}\in U}\left((\nabla_{\mathbf{x}}V)^T\mathbf{F} + \boldsymbol{\mathcal{L}} + \frac{\epsilon}{2\gamma}(\nabla_{\mathbf{x}}V)^T\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T(\nabla_{\mathbf{x}}V) + \frac{\epsilon}{2\gamma}tr\left(\nabla_{\mathbf{xx}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right)\right) \tag{2.112}$$

with the boundary condition $V(\mathbf{x}(t_N) = \phi(\mathbf{x}(t_N))$. To obtain the equation above we make use of the equalities:

$$\frac{1}{\epsilon}\Phi\ (\partial_t V) = \partial_t\Phi \tag{2.113}$$

$$\frac{1}{\epsilon}\Phi\ (\nabla_{\mathbf{x}}V) = \nabla_{\mathbf{x}}\Phi \tag{2.114}$$

$$\nabla_{\mathbf{xx}}\Phi = \frac{1}{\epsilon}(\nabla_{\mathbf{x}}V)\ (\nabla_{\mathbf{x}}V)^T + \frac{1}{\epsilon}(\nabla_{\mathbf{xx}}V)\Phi \tag{2.115}$$

The optimal control law can be found explicitly and thus is given by:

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})(\nabla_{\mathbf{x}}V) \tag{2.116}$$

Substitution of the optimal control back to the parabolic PDE results in Hamilton Jacobi Bellman equation.

$$-\partial_t V_t = q + (\nabla_{\mathbf{x}} V)^T \mathbf{f} - \frac{1}{2}(\nabla_{\mathbf{x}} V)^T \mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T(\nabla_{\mathbf{x}} V) + \frac{1}{2\gamma}(\nabla_{\mathbf{x}} V)^T \tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T(\nabla_{\mathbf{x}} V) + \frac{\epsilon}{2\gamma} tr\left(\nabla_{\mathbf{xx}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right)$$

We write the equation above in a more compact form:

$$-\partial_t V_t = q + (\nabla_{\mathbf{x}} V)^T \mathbf{f} - \frac{1}{2}(\nabla_{\mathbf{x}} V)^T \boldsymbol{\mathcal{M}}(\mathbf{x})(\nabla_{\mathbf{x}} V) + \frac{\epsilon}{2\gamma} tr\left(\nabla_{\mathbf{xx}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) \tag{2.117}$$

where the term $\boldsymbol{\mathcal{M}}(\mathbf{x})$ is defined as:

$$\boldsymbol{\mathcal{M}}(\mathbf{x}) = \mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T - \frac{1}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T \tag{2.118}$$

The PDE above is equivalent the stochastic Isaacs HJB in (2.101) if $\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T$. Thus the following theorem as stated in (Basar & Berhard 1995) with some slight modifications[2]establishes the equivalence between stochastic differential games and Risk sensitivity:

Theorem: *The stochastic differential game expressed by* (2.120) *and* (2.94) *is equivalent under the conditions 1,2,3, and 4 with the risk sensitive stochastic optimal control problem defined by* (2.103) *and* (2.104) *in the sense that the former admits a game value function with continuously differentiable in t and twice differentiable in* $\mathbf{x}$ *if*

[2]In (Basar & Berhard 1995) pp 183 the corresponding theorem is stated for the case where $\mathbf{C}(\mathbf{x}) = \tilde{\mathbf{C}}(\mathbf{x})$ while in the present form there is the assumption $\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T$.

and only of the later admits an optimum value function with same features. Furthermore the optimal control and value functions are identical and they are specified by:

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}\mathbf{G}^T\nabla_{\mathbf{x}}V \tag{2.119}$$

where

$$\begin{aligned}-\partial_t V_t = q + (\nabla_{\mathbf{x}}V)^T\mathbf{f} - \frac{1}{2}(\nabla_{\mathbf{x}}V)^T\mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T(\nabla_{\mathbf{x}}V) \\ + \frac{1}{2\gamma}(\nabla_{\mathbf{x}}V)\,\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T(\nabla_{\mathbf{x}}V) + \frac{\epsilon}{2\gamma}tr\left(\nabla_{\mathbf{x}\mathbf{x}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right)\end{aligned}$$

with boundary condition $V_{t_N} = \phi_{t_N}$*, iff the following conditions holds* $\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T$*. The parameters* $\gamma, \lambda > 0$ *and* $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ *defined as* $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{L}\mathbf{L}^T$*.*

The use of the two parameters ϵ and γ in the analysis above may seem a bit confusing. As a first observation, ϵ and γ are tuning parameters in the cost function and therefore it does not make sense to multiply the process noise in the stochastic dynamics since in most control applications the stochastic dynamics are given and their uncertainty is not a matter of manipulation and user tuning.

To resolve the confusion we consider the special case where $\epsilon = \gamma$ which is the most studied in the control literature. In this case the parameters ϵ, γ drop from the stochastic dynamics and they appear only in the cost functions. When $\epsilon \neq \gamma$ this is a generalization since we can now ask an additional question: *Given the cost functions in risk sensitive and differential game optimal control problems and the difference between the risk parameter* ϵ *and the distrurbance weight* γ *what is the form of the stochastic dynamics for which*

these two problems are equivalent. Clearly for the dynamics (2.94) and (2.103) the two stochastic optimal control problems are equivalent. Due to this generalization we keep ϵ, γ for path integral stochastic differential games and path integral risk sensitivity.

In the next section we derive the risk sensitive path integral control and we show under which conditions it is equivalent with the path integral stochastic differential games.

2.5 Information theoretic interpretations of optimal control

One of the first information theoretic interpretations of stochastic optimal control is the work by (Saridis 1996). In this work, an alternative formulation of stochastic optimal control is proposed which relates the minimization of the performance index in optimal control with the concept of Shannon Differential Entropy. Moreover, the entropy formulation is not only applied to provide interpretations of the optimal control problem but it is also used to generated alternative views for the frameworks of stochastic estimation and adaptive control in a unified way. In this section, we are going to restrict our analysis to the case of optimal control problem and its entropy interpretation.

More precisely, we start our analysis with the "traditional" formulation of the optimal control problem which consists of a cost function under minimization of the form:

$$J(\mathbf{u}, \mathbf{x}) = \phi(\mathbf{x}_{t_N}) + \int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}) dt \tag{2.120}$$

subject to the stochastic dynamics: $\mathbf{dx} = (\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u})dt + \mathbf{B}(\mathbf{x})\mathbf{d}\omega$. We define the differential entropy:

$$H\Big(\mathbf{u}(\mathbf{x},t), p(\mathbf{u}), \mathbf{x}(t_0)\Big) = -\int_{\Omega_{\mathbf{x}_0}} \int_{\Omega_{\mathbf{x}}} p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big)\ \log p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big) d\mathbf{x}\ d\mathbf{x}_0 \qquad (2.121)$$

where $p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big)$ is the probability of selecting $\mathbf{u}$ while $\mathbf{x}_0$ is the initial state and $\Omega_{\mathbf{x}}, \Omega_{\mathbf{x}_0}$ the spaces of the states and the initial conditions. Next, we are looking for the probability distribution which best represents the random variable $\mathbf{u}$. The answer to this request is given by Jayne's maximum entropy principle which states that the best distribution is the one that maximizes the entropy formulation above. This maximization procedure is subjected to the constrains that $E\Big(J(\mathbf{u}, \mathbf{x})\Big) = K$ and also $\int p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big) d\mathbf{x}_0 = 1$. As stated in (Saridis 1996), this problem is more general than the optimal control since the parameter K is fixed and unknown and it depends on the selection of the controls $\mathbf{u}(\mathbf{x}, t)$. The unconstrained maximization problem is now formulated as follows:

$$\begin{aligned}
\Upsilon &= \beta\ H\Big(\mathbf{u}(\mathbf{x},t), p(\mathbf{u}), \mathbf{x}(t_0)\Big) - \gamma\Big(E\Big(J(\mathbf{u}, \mathbf{x})\Big) - K\Big) - \alpha\ \Big(\int p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big) d\mathbf{x}_0 - 1\Big) \\
&\propto -\int \Big(\beta\ p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big)\ \log p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big) + \gamma\ p\Big(\mathbf{u}, \mathbf{x}(t_0)\Big)\ J(\mathbf{u}, \mathbf{x})\Big) d\mathbf{x} \\
&\quad - \alpha\ \Big(\int p(\mathbf{u}, \mathbf{x}(t_0)) d\mathbf{x}_0 - 1\Big) \qquad (2.122)
\end{aligned}$$

The objective function above is concave with respect to the probability distribution since the second derivative $\frac{\partial \Upsilon}{\partial p} = -\beta \frac{1}{p} < 0$. Thus to find the maximum we take the first

derivative of the objective function with respect to the distribution $p(\mathbf{u})$ and equal to zero. More precisely we have:

$$-\beta \log p(\mathbf{u}) - \beta - \gamma J(\mathbf{u}, \mathbf{x}) - \alpha = 0 \tag{2.123}$$

The worst case distribution and therefore the one which maximizes the differential entropy $H\Big(\mathbf{u}(\mathbf{x}, t), p(\mathbf{u}), \mathbf{x}(t_0)\Big)$ is expressed as follows:

$$p(\mathbf{u}) = \frac{\exp\Big(-\frac{\gamma}{\beta} J(\mathbf{u}, \mathbf{x})\Big)}{\exp\Big(-\frac{\beta}{\beta+\alpha}\Big)} \tag{2.124}$$

by assuming that $\frac{1}{\lambda} = \frac{\gamma}{\beta}$ and $\exp\Big(-\frac{\beta}{\beta+\alpha}\Big) = \int \exp\Big(-\frac{1}{\lambda} J(\mathbf{u}, \mathbf{x})\Big) d\mathbf{x}$ we will have the final result:

$$p(\mathbf{u}) = \frac{\exp\Big(-\frac{1}{\lambda} J(\mathbf{u}(\mathbf{x}, t), \mathbf{x})\Big)}{\int \exp\Big(-\frac{1}{\lambda} J(\mathbf{u}(\mathbf{x}, t), \mathbf{x})\Big) d\mathbf{x}} \tag{2.125}$$

Substitution of the worst distribution results in the maximum differential entropy expressed by the equation:

$$H\Big(\mathbf{u}(\mathbf{x}, t), p(\mathbf{u}), \mathbf{x}(t_0)\Big) = \zeta + \frac{1}{\lambda} E\Big(J(\mathbf{u}(\mathbf{x}, t), \mathbf{x})\Big) \tag{2.126}$$

where $\zeta = \frac{\beta+\alpha}{\beta}$. Given the form of the probability $p(\mathbf{u})$ total time derivative expressed as follows:

$$\frac{dp(\mathbf{u})}{dt} = \frac{d}{dt}\left(\frac{\exp\left(-\frac{1}{\lambda}J(\mathbf{u},\mathbf{x})\right)}{\exp\left(-\frac{\beta}{\beta+\alpha}\right)}\right) = -\frac{1}{\lambda}\,\boldsymbol{\mathcal{L}}(\mathbf{u},\mathbf{x})\;p(\mathbf{u}) \tag{2.127}$$

At the same time we know that $\frac{dp(\mathbf{u})}{dt} = \frac{\partial p(\mathbf{u})}{\partial t} + \frac{\partial p}{\partial \mathbf{x}}^T \dot{\mathbf{x}}$. By equating the two equations we will have:

$$\frac{\partial p(\mathbf{u})}{\partial t} + \nabla_{\mathbf{x}} p^T \dot{\mathbf{x}} + \frac{1}{\lambda}\,\boldsymbol{\mathcal{L}}(\mathbf{u},\mathbf{x})\;p(\mathbf{u}) = 0 \tag{2.128}$$

We now consider the following properties:

$$\nabla_{\mathbf{x}} p = \frac{1}{\lambda}\nabla_{\mathbf{x}} J(\mathbf{x},\mathbf{u})\;p(\mathbf{u}), \quad \text{and} \quad \frac{\partial p}{\partial t} = \frac{1}{\lambda}\frac{\partial J}{\partial t}\;p(\mathbf{u}) \tag{2.129}$$

Substitution of the equation above results in the following PDE:

$$\left(\frac{\partial J(\mathbf{u})}{\partial t} + \nabla_{\mathbf{x}} J(\mathbf{x},\mathbf{u})^T \mathbf{f}(\mathbf{x},\mathbf{u},t) + \boldsymbol{\mathcal{L}}(\mathbf{u},\mathbf{x})\right)\;\frac{1}{\lambda}\;p(\mathbf{u}) = 0 \tag{2.130}$$

which is the generalized HJB equation. By assuming that $\forall p(\mathbf{u}) > 0$ the equation above yields:

$$\frac{\partial J(\mathbf{u})}{\partial t} + \nabla_{\mathbf{x}} J(\mathbf{x},\mathbf{u})^T \mathbf{f}(\mathbf{x},\mathbf{u},t) + \boldsymbol{\mathcal{L}}(\mathbf{u},\mathbf{x}) = 0 \tag{2.131}$$

The minimization of the equation yields the optimal control that minimizes the differential entropy $H\Big(\mathbf{u}(\mathbf{x},t), p(\mathbf{u}), \mathbf{x}(t_0)\Big)$. More precisely we will have that

$$-\frac{\partial J(\mathbf{u})}{\partial t} = \min_{\mathbf{u}}\left(\nabla_{\mathbf{x}} J(\mathbf{x},\mathbf{u})^T \mathbf{f}(\mathbf{x},\mathbf{u},t) + \boldsymbol{\mathcal{L}}(\mathbf{u},\mathbf{x})\right) \tag{2.132}$$

2.6 Discussion

In this chapter we have presented basic concepts and principles in the theory of optimal control starting from the Bellman principle of optimality and the Pontryagin maximum principle. We discussed a class of model based iterative optimal control approaches by deriving the Stochastic Differential Dynamic programming (SDDP) algorithm for non-linear systems with state and control multiplicative noise. The connection between risk sensitive and differential game theoretic optimal control problems was illustrated. In the previous section we presented information theoretic interpretations of optimal control problem.

The next chapter introduces fundamental mathematical concepts in physics and control theory. These mathematical concepts include PDEs, and SDEs and the path integral. Besides the presentation of each one of these concepts, emphasis is also given their connection.

Chapter 3

Path Integrals, Feynman Kac Lemmas and their connection to PDEs

The goal in this chapter is to introduce important, for the mathematical developments of this work, concepts in the area of PDEs and their connection to path integral formalisms and SDEs. Essentially we can think about the 3 mathematical formalisms of PDEs, Path integrals and SDEs as different mathematical representations of the same underlying physical processes. But, why are there more that one mathematical representation of the same phenomenon? The reason is because these mathematical structures offer representations on a macroscopic or microscopic level. In fact, PDEs provide a macroscopic view while SDEs and path integrals formalisms offer a more microscopic view of an underlying physical process.

Among other sciences and engineering fields, the aforementioned mathematical tools are also used in the physics and control theoretic communities. These communities are dealing with different problems. As an example, while in physics it is important to predict the position of a particle under a magnetic field, in control theory the question is how to construct a magnetic filed such that the particle has a desired behavior. Clearly in both

cases one could use PDEs, on the one hand, to predict the outcome of the force field, on the other hand, to find the control policy which when applied meets the desired behavior. So, both communities are using PDEs but for different purposes. This fact results also in different terminology. What is called a "force field", for physics, it can be renamed as "control policy" in control theory.

The observations above are not necessarily objective, but they are very much related to our experiences as we were trying to understand and bring together concepts from physics and control theory. With this background in our mind, in this chapter our goal is to bring together concepts from physics and control theory with emphasis on the connection between PDEs, SDEs and Path Integrals. More precisely, section 3.1 is a short journey in the world of quantum mechanics and the work on Path Integrals by one of the most brilliant intellectuals in the history of sciences, Dr. Richard Feynman. By no means, this section is not a review his work. This section just aims to show that the core concepts of this thesis, which is the Path Integral, has its historical origins in the work by Richard Feynman.

In sections 3.2 and 3.4 we highlight the convection between the forward Fokker Planck PDEs and the underlying SDE for both the Itô and the Stratonovich calculus. With the goal to establish the connection between the path integral formalism and SDEs, in section 3.3, we derive the path integral for the general stochastic integration scheme for 1-dimensional SDE and then we specialize for the cases of Itô and the Stratonovich calculus. In section 3.4 the derivation of the Itô path integral for multi-dimensional SDEs is presented. The last two sections are aiming to show how forwards PDEs are connected to Path Integrals and SDEs.

Forward PDEs such as the Fokker Planck equation or the Chapman- Kolmogorov PDE in its forward form, are typically used in estimation problems. In particular in the case where stochasticity is considered only for the state space dynamics, the Fokker Planck PDE is the appropriate mathematical description, while in cases in which there is also measurement uncertainty (partial observability), the forward Chapman-Kolmogorov equation is the corresponding PDE. However, in an optimal control setting, the PDEs are usually backward and since they are related to the concept of the value function and the Bellman principle of optimality. Is there any connection between these backward PDEs, SDEs and the corresponding Path Integrals formalisms? The answer is that this connection exists and it is established via the Feynman- Kac lemma in section (3.5). The Feynman Kac lemma is of great importance because is provides a way to probabilistically represent solution of PDEs. In section (3.5) we provide the full proof of the Feynman-Kac lemma, which in its complete form, is rarely found in the literature. In section 3.6 we discuss special cases of the Feynman-Kac lemma.

Besides the connection between the forward and backward PDEs, SDE and Path integrals we also discuss how the forward and backward Chapman-Kolmogorov equations are connected in 3 different levels which are: i) through the mathematical concept of *fundamental solutions* of PDEs, ii) via a slightly modified version the proof of the Feynman -Kac lemma and iii) through the Generalized Duality between the optimal estimation and control problems. All these issues are addressed in sections 3.7, 3.8 and 3.9. In the last section we conclude and prepare the discussion for the next chapter.

3.1 Path integrals and quantum mechanics

Since the mathematical construction of the path integral plays a central role in this work, it would have been a severe gap if this work did not include an introduction to path integrals and their use for the mathematical representation of quantum phenomena in physics. Therefore, in the next two subsections, we discuss the concept of least action in classical mechanics and its generalization to quantum mechanics via the use of the path integral. Moreover, we provide the connection between the path integral and the Schrődinger equation, one of the most important equations in quantum physics.

The Schrődinger equation was discovered in 1925 by the physicist and theoretical biologist, Erwin Rudolf Josef Alexander Schrődinger (Nobel-Lectures 1965). The initial idea of the path integral goes back Paul Adrien Morice Dirac, a theoretical physicist who together with Schrődinger was awarded the Nobel Prize in Physics in 1933 for their work *on discovery of the new productive forms of atomic theory.* Richard Phillips Feynman (Nobel-Lectures 1972), also a theoretical physicist and Nobel price winner in 1965 for his work on quantum electrodynamics, completed the theory of path integral in 1948.

3.1.1 The principle of least action in classical mechanics and the quantum mechanical amplitude.

Let us assume the case where a dynamical system moves from an initial state $\mathbf{x}_A$ to a terminal state $\mathbf{x}_B$. The principle of least action (Feynman & Hibbs 2005) states that the system will follow the trajectory $\mathbf{x}_A^*, \mathbf{x}_1^*, ..., \mathbf{x}_{N-1}^*, \mathbf{x}_B^*$ that is the extremum of the cost function:

$$S = \int_{t_A}^{t_B} \boldsymbol{\mathcal{L}}(\mathbf{x}, \dot{\mathbf{x}}, t)dt \tag{3.1}$$

where $\boldsymbol{\mathcal{L}}(\mathbf{x}, \dot{\mathbf{x}}, t)$ is the Langragian of the system defined as $\boldsymbol{\mathcal{L}} = E_{kin} - U$ with E_{kin} being the total kinetic energy and U being the potential energy of the system and S is the so called action. For a particle of mass m moving in a potential $V(x)$, the Lagrangian is $\boldsymbol{\mathcal{L}}(x, \dot{x}) = \frac{1}{2}m\dot{x}^2 - V(x)$. By using the Calculus of variations, the optimal path can be determined. We start by taking the Taylor series expansion of $S(\mathbf{x} + \delta\mathbf{x})$ and we have:

$$\begin{aligned} S(\mathbf{x} + \delta\mathbf{x}) &= \int_{t_A}^{t_B} \boldsymbol{\mathcal{L}}(\mathbf{x} + \delta\mathbf{x}, \dot{\mathbf{x}} + \delta\dot{\mathbf{x}}, t)dt \\ &= \int_{t_A}^{t_B} \Big(\boldsymbol{\mathcal{L}}(\mathbf{x}, \dot{\mathbf{x}}, t) + \delta\mathbf{x}^T \nabla_{\mathbf{x}} \boldsymbol{\mathcal{L}} + \delta\dot{\mathbf{x}}^T \nabla_{\dot{\mathbf{x}}} \boldsymbol{\mathcal{L}} \Big) dt \\ &= S(\mathbf{x}) + \int_{t_A}^{t^B} \Big(\delta\mathbf{x}^T \nabla_{\mathbf{x}} \boldsymbol{\mathcal{L}} + \delta\dot{\mathbf{x}}^T \nabla_{\dot{\mathbf{x}}} \boldsymbol{\mathcal{L}} \Big) dt \end{aligned}$$

After integrating by parts we will have

$$\Delta S = \Big[\delta\mathbf{x}^T \nabla_{\dot{\mathbf{x}}} \boldsymbol{\mathcal{L}} \Big]_{t_0}^{T} - \int_{t_0}^{T} \delta\mathbf{x}^T \Big[\frac{d}{dt} \Big(\nabla_{\dot{\mathbf{x}}} \boldsymbol{\mathcal{L}} \Big) - \nabla_{\mathbf{x}} \boldsymbol{\mathcal{L}} \Big] dt$$

Given the condition $\delta\mathbf{x}(t_0) = 0$ and $\delta\mathbf{x}_T = 0$ we will have:

$$\Delta S = - \int_{t_A}^{t_B} \delta\mathbf{x}^T \Big[\frac{d}{dt} \Big(\nabla_{\dot{\mathbf{x}}} \boldsymbol{\mathcal{L}} \Big) - \nabla_{\mathbf{x}} \boldsymbol{\mathcal{L}} \Big] dt$$

To find the optimal trajectory we set $\Delta S = 0$. Therefore the condition that the optimal trajectory satisfies is expressed as:

$$\frac{d}{dt}\left(\nabla_{\dot{\mathbf{x}}}\boldsymbol{\mathcal{L}}\right) - \nabla_{\mathbf{x}}\boldsymbol{\mathcal{L}} = 0$$

The equation above is the so called Euler- Lagrange equation. In quantum mechanics, for a motion of a particle from $\mathbf{x}_0$ to $\mathbf{x}_T$ there is the concept of amplitude $K(\mathbf{x}_0, \mathbf{x}_T)$ that is associated with it. This amplitude is defined as the sum of contributions $\phi(\mathbf{x})$ of all the trajectories that start from $\mathbf{x}_0$ and end in $\mathbf{x}_T$. The contributions $\phi(\mathbf{x})$ are defined as:

$$\phi(\mathbf{x}_A \to \mathbf{x}_B) = const \times \exp\left(\frac{j}{h} S(\mathbf{x}_A \to \mathbf{x}_B)\right)$$

where $S(\mathbf{x})$ is the action, and *const* is a normalization factor. Based on the definition of contributions of individual paths (Feynman & Hibbs 2005), the amplitude is defined as:

$$K(\mathbf{x}_A, \mathbf{x}_B) = K(\mathbf{x}_A \to \mathbf{x}_B) = \sum \phi(\mathbf{x}_A \to \mathbf{x}_B) = \sum const \times \exp\left(\frac{j}{h} S(\mathbf{x}_A \to \mathbf{x}_B)\right) \tag{3.2}$$

The probability of going from $\mathbf{x}_A$ to $\mathbf{x}_B$ is defined as the square of the amplitude $K(\mathbf{x}_A, \mathbf{x}_B)$ and thus it is expressed as $p(\mathbf{x}_A \to \mathbf{x}_B) = |K(\mathbf{x}_A, \mathbf{x}_B)|^2$ (Feynman & Hibbs 2005) .

Clearly, the mathematical term of the contribution of each individual path $\phi(\mathbf{x}_A \to \mathbf{x}_B)$ is represented by a complex number. This is because light can be thought not only as moving particles with tiny mass but also as waves traveling via different paths towards the same destination. Moreover, although the concept of amplitude $K(\mathbf{x}_A \to \mathbf{x}_B)$ is

associated with the probability $p(\mathbf{x}_A \rightarrow \mathbf{x}_B)$, it remains somehow an abstract concept. One can further understand it, by looking into how laws of classical mechanics arise from the quantum mechanical law and how the path integral formulation provides a mathematical representation for this relationship.

To investigate the relation between classical and quantum mechanical laws it is important to realize that the term $\bar{h} = \frac{h}{2\pi} = 1.055 \times 10^{-27} erg \cdot sec$ where h is Planck's constant, is a very small number. In addition, in classical mechanics the action S is much larger than $\bar{h}$ due to the scale of masses and time horizons of bodies and motions. Thus the fact that $\frac{1}{\bar{h}}$ is a very large number increases the sensitivity of the phase variable $\theta = \frac{S(\mathbf{x}+\delta\mathbf{x})}{\bar{h}}$ of a path with respect to the changes of the action $S(\mathbf{x})$ of the corresponding path. Small deviation of the action $S(\mathbf{x} + \delta\mathbf{x})$ create enormous changes in the phase variable θ of the path. As a consequence, neighbored paths of the classical extremum, will have very high phases with opposite signs which will cancel out their corresponding contributions. Only paths in the vicinity of the extremum path will be in-phase and they will contribute and create the extremum path which satisfies the Euler-Langrange equation. Thus, clearly in the classical mechanics there is only one path from $\mathbf{x}_0$ to $\mathbf{x}_T$.

In the quantum world, the scale of masses and time horizons of bodies and motions are such that the action $S(\mathbf{x})$ is comparable to the term $\frac{1}{\bar{h}}$. In this case, deviations of the action $S(\mathbf{x} + \delta\mathbf{x})$ do not create enormous changes and thus all paths will interfere by contributing to the total amplitude and the total probability of the motion of the corresponding particle from $\mathbf{x}_0$ to $\mathbf{x}_T$. We realize that the path integral in 3.2 provides a simple and intuitive way to understand how classical mechanical and quantum mechanical

phenomena are related by just changing the scales of body masses and time horizons. Essentially the path integral provides a generalization of the concept of action from classical mechanics to the quantum mechanical word.

Before we close this subsection, we present some alternative mathematical representations of the path integral in equation 3.2 . More precisely in a compact form, the path integral is written as:

$$\boxed{K(\mathbf{x}_A, \mathbf{x}_B) = \int_{\mathbf{x}_A}^{\mathbf{x}_B} \exp\left(\frac{i}{h} S(\mathbf{x})\right) D(\mathbf{x}(t))} \tag{3.3}$$

The path from the state $\mathbf{x}_A$ to $\mathbf{x}_B$ can be split into two pieces by incorporating a new state $\mathbf{x}_C$. Therefore, the equation above can be written as:

$$\begin{aligned}
&K(\mathbf{x}_A, \mathbf{x}_B) = \\
&= \int_{\mathbf{x}_A}^{\mathbf{x}_B} \exp\left(\frac{i}{h} S(\mathbf{x}_A \to \mathbf{x}_C) + \frac{i}{h} S(\mathbf{x}_C \to \mathbf{x}_B)\right) D(\mathbf{x}(t)) \\
&= \int \int_{\mathbf{x}_A}^{\mathbf{x}_B} \exp\left(\frac{i}{h} S(\mathbf{x}_A \to \mathbf{x}_C) + \frac{i}{h} S(\mathbf{x}_C \to \mathbf{x}_B)\right) D(\mathbf{x}_A \downarrow \mathbf{x}_C) D(\mathbf{x}_C \downarrow \mathbf{x}_B) d\mathbf{x}_C \\
&= \int \int_{\mathbf{x}_A}^{\mathbf{x}_C} \exp\left(\frac{i}{h} S(\mathbf{x}_A \to \mathbf{x}_C)\right) D(\mathbf{x}_A \downarrow \mathbf{x}_C) \int_{\mathbf{x}_C}^{\mathbf{x}_B} \exp\left(\frac{i}{h} S(\mathbf{x}_C \to \mathbf{x}_B)\right) D(\mathbf{x}_C \downarrow \mathbf{x}_B) d\mathbf{x}_C
\end{aligned}$$

Given that the path $\mathbf{x}_A \to \mathbf{x}_B$ is defined as $\mathbf{x}_A \to \mathbf{x}_B = \{\mathbf{x}_A, \mathbf{x}_1, ..., \mathbf{x}_{C-1}, \mathbf{x}_C, \mathbf{x}_{C+1}, ..., \mathbf{x}_B\}$ the term $D(\mathbf{x}_A \downarrow \mathbf{x}_C) = d\mathbf{x}_1 \times ... \times d\mathbf{x}_{C-1}$ and $D(\mathbf{x}_C \downarrow \mathbf{x}_B) = d\mathbf{x}_{C+1} \times ... \times d\mathbf{x}_B$. The equation above can be written as:

$$K(\mathbf{x}_A, \mathbf{x}_B) = \int_{-\infty}^{\infty} K(\mathbf{x}_A, \mathbf{x}_C) K(\mathbf{x}_B, \mathbf{x}_C) d\mathbf{x}_C$$

By continuing this process of splitting the paths from $\mathbf{x}_A$ to $\mathbf{x}_B$ into subpaths, the path integral takes the form:

$$\boxed{K(\mathbf{x}_A, \mathbf{x}_B) = \lim_{dt \to 0} \int ... \int \int \prod_{i=1}^{N} K\Big(\mathbf{x}_{i+1}, \mathbf{x}_i\Big) d\mathbf{x}_1 d\mathbf{x}_2 ... d\mathbf{x}_N} \tag{3.4}$$

where the kernel $K(\mathbf{x}_{i+1}, \mathbf{x}_i)$ is now defined as:

$$\boxed{K(\mathbf{x}_{i+1}, \mathbf{x}_i) = \frac{1}{A} \exp\left[\frac{i}{\bar{h}}\ \delta t\ \boldsymbol{\mathcal{L}}\left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\delta t}, \frac{\mathbf{x}_{i+1} + \mathbf{x}_i}{2}, \frac{t_{i+1} + t_i}{2}\right)\right]} \tag{3.5}$$

and $A = \left(\frac{2\pi i \bar{h} dt}{m}\right)^{1/2}$. The equations (3.4) and (3.5) above realize the path integral formulation in discrete time. The path integral formulation is an alternative view of Quantum Mechanics in the next section we discuss the Schrődinger equation and its connection to path integral.

3.1.2 The Schrődinger equation

In this section, we show how one of the most central equations in quantum mechanics, the Schrődinger equation, is derived from the mathematical concept of path integrals. The connection between the two descriptions is of critical importance since it provides a more complete view of quantum mechanics (Feynman & Hibbs 2005), but it is also an example of mathematical connection between path integrals and PDEs.

The derivation starts with the waive function $\psi(x, t)$ which can be though as an amplitude with the slight difference that the associate probability $P(\mathbf{x}, t) = |\psi(\mathbf{x}, t)|^2$ is the probability of being at state $\mathbf{x}$ at time t without looking into the past. Since the wave function is an amplitude function it satisfies the integral equation:

$$\psi(\mathbf{x}_N, t_N) = \int_{-\infty}^{\infty} K(\mathbf{x}_N, t_N; \mathbf{x}_{N-1}, t_{N-1}) \psi(\mathbf{x}_{N-1}, t_{N-1}) d\mathbf{x}_{N-1} \tag{3.6}$$

Substitution of the kernel $K(\mathbf{x}_N, t_N; \mathbf{x}_{N-1}, t_{N-1})$ yields:

$$\psi\bigg(\mathbf{x}(t+\delta t), t+\delta t\bigg) =$$
$$= \int_{-\infty}^{\infty} \exp\left[\frac{i}{\bar{h}} \delta t \boldsymbol{\mathcal{L}}\bigg(\frac{\mathbf{x}(t+\delta t) - \mathbf{x}(t)}{\delta t}, \frac{\mathbf{x}(t+\delta t) + \mathbf{x}(t)}{2}\bigg)\right] \psi(\mathbf{x}, t) d\mathbf{x}(t)$$

For simplifying the notation we make the substitutions, $\mathbf{x}(t+\delta t) = \mathbf{x}$ and $\mathbf{x}(t) = \mathbf{y}$.

$$\psi\bigg(\mathbf{x}, t+\delta t\bigg) = \int_{-\infty}^{\infty} \exp\left[\frac{i}{\bar{h}} \delta t \boldsymbol{\mathcal{L}}\bigg(\frac{\mathbf{x}-\mathbf{y}}{\delta t}, \frac{\mathbf{x}+\mathbf{y}}{2}\bigg)\right] \psi(\mathbf{y}, t) d\mathbf{y}(t)$$

Substitution of the Langragian results in:

$$\psi\bigg(\mathbf{x}, t+\delta t\bigg) =$$
$$= \frac{1}{A} \int_{-\infty}^{\infty} \exp\left[\frac{im}{2\bar{h}} \frac{\bigg(\mathbf{x}-\mathbf{y}\bigg)^T \bigg(\mathbf{x}-\mathbf{y}\bigg)}{\delta t}\right] \exp\left[-\frac{i}{\bar{h}}\ \delta t\ V\bigg(\frac{\mathbf{x}+\mathbf{y}}{2}\bigg)\right] \psi(\mathbf{y}, t) d\mathbf{y}(t)$$
$$= \frac{1}{A} \int_{-\infty}^{\infty} \exp\left[\frac{im\mathbf{v}^T\mathbf{v}}{2\bar{h}\delta t}\right] \exp\left[-\frac{i}{\bar{h}}\ \delta t\ V\bigg(\mathbf{x} + \frac{1}{2}\mathbf{v}\bigg)\right] \psi(\mathbf{y}, t) d\mathbf{y}(t)$$

where $\mathbf{v} = \mathbf{x} - \mathbf{y}$. Next the second exponential function is expanded, while $\psi(\mathbf{y}, t)$ is expended around $\mathbf{x}$. More precisely we will have:

$$\psi\Big(\mathbf{x}, t + \delta t\Big) =$$
$$= \frac{1}{A}\int_{-\infty}^{\infty} \exp\left[\frac{im\mathbf{v}^T\mathbf{v}}{2\bar{h}\delta t}\right]\left[1 - \frac{i}{\bar{h}}\delta t V(\mathbf{x}, t)\right]\left[\psi(\mathbf{x}, t) - \nabla\psi^T\mathbf{v} + \frac{1}{2}\mathbf{v}^T\nabla_{\mathbf{xx}}\psi\mathbf{v}\right]d\mathbf{y}(t)$$

By using the following equalities $\frac{1}{A}\int_{-\infty}^{+\infty}\mathbf{v}\exp\left(\frac{im}{2\bar{h}\delta t}\mathbf{v}^T\mathbf{v}\right)d\mathbf{v} = 0$ as well as the equation $\frac{1}{A}\int_{-\infty}^{+\infty}\mathbf{v}\mathbf{v}^T\exp\left(\frac{jm}{2\bar{h}\delta t}\mathbf{v}^T\mathbf{v}\right)d\mathbf{v} = \frac{i\bar{h}\delta t}{m}I_{n\times n}$ the wave function is formulated as:

$$\psi\Big(\mathbf{x}, t + \delta t\Big) = \psi(\mathbf{x}, t) - \frac{i}{\bar{h}}\delta t V(\mathbf{x}, t)\psi(\mathbf{x}, t) + \frac{i\bar{h}\delta t}{2m}tr\Big(\nabla_{\mathbf{xx}}\psi\Big)$$

The last step is to take the Taylor series expansion of $\psi\Big(\mathbf{x}, t + \delta t\Big) = \psi(\mathbf{x}, t) + \delta t\partial_t\psi$:

$$\psi(\mathbf{x}, t) + \delta t\partial_t\psi = \psi(\mathbf{x}, t) - \frac{i\delta t}{\bar{h}}V(\mathbf{x}, t)\psi(\mathbf{x}, t) + \frac{i\bar{h}\delta t}{2m}tr\Big(\nabla_{\mathbf{xx}}\psi\Big)$$

The final version of the Schrődinger equation takes the form:

$$\boxed{\partial_t\psi = -\frac{i}{\bar{h}}\Big[-\frac{\bar{h}^2}{2m}tr\Big(\nabla_{xx}\psi\Big) + V(\mathbf{x}, t)\psi\Big]} \tag{3.7}$$

By introducing the operator $H = -\frac{\bar{h}^2}{2m}tr\Big(\nabla_{\mathbf{xx}}\Big) + V(\mathbf{x}, t)$ the Schrődinger equation is formulated as:

$$\boxed{\partial_t\psi = -\frac{i}{\bar{h}}H\psi} \tag{3.8}$$

With the derivation of the Schrődinger equation, we close our introduction to path integrals in quantum mechanics.

3.2 Fokker Planck equation and SDEs

The Fokker planck PDE is of great importance in statistical mechanics as it has been used to describe the evolution of the probability of particles as a function of space and time. It can be thought as the equivalent of the Schrődinger equation in quantum mechanics. In the next two sections, we will derive the Fokker Planck PDE starting from the underlying Itô and Stratonovich stochastic differential equation (Chirikjian 2009).

3.2.1 Fokker Planck equation in Itô calculus

We start with the following stochastic differential equation:

$$\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)d\mathbf{w} \tag{3.9}$$

in which $\mathbf{x} \in \Re^{n\times 1}$ is the state, and $d\mathbf{w} = \mathbf{w}(t) - \mathbf{w}(t + dt)$ with $\mathbf{w}(t) \in \Re^{p\times 1}$ a *Wiener process* (or Brownian motion process). The equation above is an Itô stochastic differential equation if its solution:

$$\mathbf{x}(t) - \mathbf{x}(0) = \int_0^t \mathbf{f}(\mathbf{x}, t)d\tau + \int_0^t \mathbf{B}(\mathbf{x}, t)d\mathbf{w}(\tau) \tag{3.10}$$

can be interpreted in the sense:

$$\lim_{\mu\to\infty} E\Bigg(\left[\int_0^t \mathbf{B}(\mathbf{x}(\tau),\tau)d\tau - \sum_{k=1}^{\mu} \mathbf{B}(\mathbf{x}(t_{k-1}),t_{k-1}) \left[\mathbf{w}(t_k) - \mathbf{w}(t_{k-1})\right] \right]^2 \Bigg) = 0 \qquad (3.11)$$

where $t_0 = 0 < t_1 < t_2 < ... < t_N = t$. The drift part of the SDE in 3.9 is also interpreted as:

$$\lim_{\mu\to\infty} E\Bigg(\left[\int_0^t \mathbf{f}(\mathbf{x}(\tau),\tau)d\tau - \frac{1}{\mu}\sum_{k=1}^{\mu} \mathbf{f}(\mathbf{x}(t_{k-1}),t_{k-1}) \right]^2 \Bigg) = 0 \qquad (3.12)$$

for the cases where the function $\mathbf{f}(\mathbf{x},t)$ is not pathological, then the limit can be pushed inside the expectation. Consequently, the equation above is true due to the fact that

$$\lim_{\mu\to\infty} \int_0^t \mathbf{f}(\mathbf{x}(\tau),\tau)d\tau = \frac{1}{\mu}\sum_{k=1}^{\mu} \mathbf{f}(\mathbf{x}(t_{k-1}),t_{k-1}) \qquad (3.13)$$

For the derivation of the corresponding Fokker Planck equation we will make use of the expectation of terms of the form $E\left(\boldsymbol{\mathcal{Z}}\mathbf{dx}\right)$ and $E\left(\mathbf{dx}\ \boldsymbol{\mathcal{M}}\ \mathbf{dx}^T\right)$ where $\boldsymbol{\mathcal{Z}} \in \Re^{n\times 1}$ and $\boldsymbol{\mathcal{M}} \in \Re^{n\times n}$. Thus we will have:

$$E\left(\boldsymbol{\mathcal{Z}}\mathbf{dx}\right) = E\Big(\boldsymbol{\mathcal{Z}}\ \mathbf{f}(\mathbf{x},t)dt\Big) = \boldsymbol{\mathcal{Z}}\ \mathbf{f}(\mathbf{x},t)dt \qquad (3.14)$$

$$E\left(\mathbf{dx}\boldsymbol{\mathcal{M}}\mathbf{dx}^T\right) = E\Big(d\mathbf{w}^T\mathbf{B}(\mathbf{x},t)^T\boldsymbol{\mathcal{M}}\mathbf{B}(\mathbf{x},t)d\mathbf{w}\Big) = tr\Big(\mathbf{B}(\mathbf{x},t)^T\boldsymbol{\mathcal{M}}\mathbf{B}(\mathbf{x},t)dt\Big) \qquad (3.15)$$

where we have used the properties of the *Wiener process* $E\left(d\mathbf{w}\right)=0$, $E\left(d\mathbf{w}\ d\mathbf{w}^T\right)=dtI_{m\times m}$. Now we are ready to derive the Fokker Planck PDE and we start with the partial derivative of the probability function $p(\mathbf{x}(t)|\mathbf{y},t)$ where $\mathbf{y}=\mathbf{x}(0)$. To avoid any confusion, it is important to understand the notation $p(\mathbf{x}(t)|\mathbf{y},\delta t)$. In particular $p(\mathbf{x}(t)|\mathbf{y},\delta t)$ is interpreted as the probability of being at state $\mathbf{x}$ at time t_1 given that the state at time $t_2<t_1$ is $\mathbf{y}(t_2)$ and $t_1-t_2=\delta t$. Consequently, in case where $t_1=t$ and $t_2=0$, the transition probability $p(\mathbf{x}|\mathbf{y},t)$ is absolutely meaningful. The partial derivative of $p(\mathbf{x}|\mathbf{y},t)$ with respect to time is expressed as follows:

$$\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}=\lim_{\delta t\to 0}\frac{p(\mathbf{x}|\mathbf{y},t+\delta t)-p(\mathbf{x}|\mathbf{y},t)}{\delta t} \tag{3.16}$$

the probability $p(\mathbf{x}|\mathbf{y},t+\delta t)$ can be also written via the *Chapman Kolmogorov equation* as $p(\mathbf{x}|\mathbf{y},t+\delta t)=\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)d\mathbf{z}$. Therefore we will have that:

$$\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}=\lim_{\delta t\to 0}\frac{\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)d\mathbf{z}-p(\mathbf{x}|\mathbf{y},t)}{\delta t} \tag{3.17}$$

Lets define the function $\psi(\mathbf{x},t)\in\Re$ that is compactly supported and it is $\boldsymbol{\mathcal{C}}^2$. We project $\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}$ on $\psi(\mathbf{x},t)$ in Hilbert space and we have that:

$$\int\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}\psi(\mathbf{x})d\mathbf{x}=\lim_{\delta t\to 0}\frac{1}{\delta t}\int\left(\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)d\mathbf{z}-p(\mathbf{x}|\mathbf{y},t)\right)\psi(\mathbf{x},t)d\mathbf{x} \tag{3.18}$$

by exchanging the order of integration we will have that:

$$\int \frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}\psi(\mathbf{x},t)d\mathbf{x} = \lim_{\delta t\to 0}\frac{1}{\delta t}\Bigg(\int\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)\psi(\mathbf{x})d\mathbf{x}d\mathbf{z} - \int p(\mathbf{x}|\mathbf{y},t)\psi(\mathbf{x})d\mathbf{x}\Bigg) \tag{3.19}$$

The Taylor series expansion of $\psi(\mathbf{x}) = \psi(\mathbf{z}+\mathbf{dx})$ where $d\mathbf{x} = \mathbf{x}-\mathbf{z}$ is expressed as follows:

$$\psi(\mathbf{x}) = \psi(\mathbf{z}) + \nabla_{\mathbf{z}}\psi(\mathbf{z})\ (\mathbf{x}-\mathbf{z}) + \frac{1}{2}(\mathbf{x}-\mathbf{z})^T\ \nabla_{\mathbf{zz}}\psi(\mathbf{z})\ (\mathbf{x}-\mathbf{z}) \tag{3.20}$$

We substitute the expanded term $\psi(\mathbf{x})$ in the first term of the left side of (3.19) and we have:

$$\begin{aligned}&\int\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)\Bigg(\psi(\mathbf{z}) + \nabla_{\mathbf{z}}\psi(\mathbf{z})\ \mathbf{dx} + \frac{1}{2}\mathbf{dx}^T\ \nabla_{\mathbf{zz}}\psi(\mathbf{z})\ \mathbf{dx}\Bigg)d\mathbf{x}d\mathbf{z} = \\ &\int p(\mathbf{z}|\mathbf{y},t)\psi(\mathbf{z})d\mathbf{z} + \int\int p(\mathbf{x}|\mathbf{z},t)p(\mathbf{z}|\mathbf{y},\delta t)\Bigg(\nabla_{\mathbf{z}}\psi(\mathbf{z})\ \mathbf{dx} + \frac{1}{2}\mathbf{dx}^T\ \nabla_{\mathbf{zz}}\psi(\mathbf{z})\ \mathbf{dx}\Bigg)d\mathbf{x}d\mathbf{z}\end{aligned} \tag{3.21}$$

The terms $\int p(\mathbf{x}|\mathbf{z},\delta t)\nabla_{\mathbf{z}}\psi(\mathbf{z})^T\mathbf{dx}\ d\mathbf{y}$ and $\int p(\mathbf{x}|\mathbf{z},\delta t)\ \mathbf{dx}^T\ \nabla_{\mathbf{zz}}\psi(\mathbf{z})\mathbf{dx}\ d\mathbf{y}$ can be written in the form $E\left(\nabla_{\mathbf{z}}\psi(\mathbf{z})^T\mathbf{dx}\right)$ and $E\left(\mathbf{dx}^T\ \nabla_{\mathbf{zz}}\psi(\mathbf{z})\ \mathbf{dx}\right)$ which, according to (3.14) and (3.15) are equal to $\nabla_{\mathbf{z}}\psi(\mathbf{z})^T\mathbf{f}(\mathbf{z},t)dt$ and $tr\Bigg(\mathbf{B}(\mathbf{z},t)^T\nabla_{\mathbf{zz}}\psi(\mathbf{z})\ \mathbf{B}(\mathbf{z},t)dt\Bigg)$. By substituting (3.21) in to (3.19) it is easy to show that:

$$\int \frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}\psi(\mathbf{x})d\mathbf{x} = \int p(\mathbf{z}|\mathbf{y},t)\left(\nabla_{\mathbf{z}}\psi(\mathbf{z})^T\mathbf{f}(\mathbf{z},t) + \frac{1}{2}tr\left(\nabla_{\mathbf{zz}}\psi(\mathbf{z})\ \mathbf{B}(\mathbf{z},t)\mathbf{B}(\mathbf{z},t)^T\right)\right)d\mathbf{z} \tag{3.22}$$

where the terms $\int p(\mathbf{z}|\mathbf{y},t)\psi(\mathbf{z})d\mathbf{z}$ in (3.21) and $-\int p(\mathbf{x}|\mathbf{y},t)\psi(\mathbf{x})d\mathbf{x}$ in (3.19) are equal and therefore they have been cancelled out. In the final step we integrate by part the ride side of the equation above and therefore we will have that:

$$\begin{aligned}&\int \frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t}\psi(\mathbf{x})d\mathbf{x} \\ &= \int -\nabla_{\mathbf{z}}\cdot(\mathbf{f}(\mathbf{z},t)p(\mathbf{z}|\mathbf{y},t)) + \frac{1}{2}tr\left(\nabla_{\mathbf{zz}}\left(\mathbf{B}(\mathbf{z},t)\mathbf{B}(\mathbf{z},t)^T p(\mathbf{z}|\mathbf{y},t)\right)\right)\psi(\mathbf{z})d\mathbf{z}\end{aligned} \tag{3.23}$$

Since $\mathbf{x},\mathbf{z} \in \Re^n$ and the integrals are calculated in the entire $\Re^n$ the equation above is written in the following form:

$$\int \left(\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t} + \nabla_{\mathbf{x}}\cdot(\mathbf{f}(\mathbf{x},t)p(\mathbf{x}|\mathbf{y},t)) - \frac{1}{2}tr\left(\nabla_{\mathbf{xx}}\left(\mathbf{B}(\mathbf{x},t)\mathbf{B}(\mathbf{x},t)^T p(\mathbf{x}|\mathbf{y},t)\right)\right)\right)\psi(\mathbf{x})d\mathbf{x} \tag{3.24}$$

We now apply the fundamental theorem of calculus of variations (Leitmann 1981) according to which: *let* $f(x) \in \boldsymbol{\mathcal{C}}^k$ *, if* $\int_a^b f(x)h(x)dx = 0, \quad \forall\ h(x) \in \boldsymbol{\mathcal{C}}^k$ *and* $h(a) = h(b) = 0$ *then* $f(x) = 0$. Consequently we will have that:

$$\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{f}(\mathbf{x},t)p(\mathbf{x}|\mathbf{y},t)) - \frac{1}{2}tr\left(\nabla_{\mathbf{xx}}\left(\mathbf{B}(\mathbf{x},t)\mathbf{B}(\mathbf{x},t)^T p(\mathbf{x}|\mathbf{y},t)\right)\right) = 0 \quad (3.25)$$

or in the form:

$$\boxed{\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t} = -\sum_{i=0}^{n} \frac{\partial}{\partial x_i}\left(\mathbf{f}(\mathbf{x},t)p(\mathbf{x}|\mathbf{y},t)\right) + \frac{1}{2}\sum_{k=1}^{m}\sum_{i,j=1}^{n}\left(\mathbf{B}_{i,k}(\mathbf{x},t)\mathbf{B}_{k,j}(\mathbf{x},t)^T p(\mathbf{x}|\mathbf{y},t)\right)} \quad (3.26)$$

The PDE above is the so called Fokker Planck Equation which is a forward, second order and linear PDE. From the derivation it is clear that the Fokker Planck equation describes the evolution of the transition probability of a stochastic dynamical system of the form (3.9) over time. In fact, lets consider a number of trajectories as realizations of the same stochastic dynamics then the 2nd term, which corresponds to drift, in (3.37) controls the direction of the trajectories while the 3rd term, that corresponds to diffusion, quantifies how much these trajectories spread due to noise in the dynamics (3.9). As we will show in the next section the FKP PDE differs from the forward Kolmogorov PDE only in one term but we will leave this discussion for the future section.

3.2.2 Fokker Planck equation in Stratonovich calculus

In this section we derive the Fokker Planck PDE in case where the underlying stochastic differential equation is integrated in the Stratonovich sense. We start our analysis with the stochastic differential equation in (3.9) if its solution is interpreted as the integral:

$$\mathbf{x}(t) - \mathbf{x}(0) = \int_0^t \mathbf{f}^{(\mathbf{S})}(\mathbf{x}, \tau) d\tau + \int_0^t \mathbf{B}^{(\mathbf{S})}(\mathbf{x}, \tau) \oplus d\mathbf{w}(\tau) \tag{3.27}$$

where the superscript $\mathbf{S}$ is used to distinguish that the function $\mathbf{f}(\mathbf{x}, t)$ and $\mathbf{B}(\mathbf{x}, t)$ are evaluated in the Stratonovich convention and therefore they are different from the corresponding functions in the Itô calculus. More precisely the Stratonovich integration[1] is defined as:

$$\int_t^{t_0} f(\tau) \oplus \mathbf{w}(\tau) = \lim_{t \to \infty} \sum_{i=1}^{n} f\left(\frac{t_i + t_{i-1}}{2}\right) (\mathbf{w}(t_i) - \mathbf{w}(t_{i-1})) \tag{3.28}$$

where the equal sign above is understood in the mean square sense. Clearly, the drift part of solution (3.27) of the stochastic differential equation (3.9) can be interpreted as:

$$\int_0^t \mathbf{f}^{(\mathbf{S})}(\mathbf{x}, \tau) d\tau = \frac{1}{n} \lim_{n \to \infty} \sum_{i=1}^{n} \mathbf{f}\left(\frac{\mathbf{x}(t_i) + \mathbf{x}(t_{i-1})}{2}, \frac{t_i - t_{i-1}}{2}\right) \tag{3.29}$$

while the diffusion part is interpreted as:

$$\int_0^t \mathbf{B}^{(\mathbf{S})}(\mathbf{x}, \tau) \oplus d\mathbf{w}(\tau) = \frac{1}{n} \lim_{n \to \infty} \sum_{i=1}^{n} \mathbf{B}\left(\frac{\mathbf{x}(t_i) + \mathbf{x}(t_{i-1})}{2}, \frac{t_i - t_{i-1}}{2}\right) (\mathbf{w}(t_i) - \mathbf{w}(t_{i-1})) \tag{3.30}$$

[1] The symbol $\oplus$ is used to represent the Stratonovich integral.

The equalities (3.29) and (3.30) are understood in the mean square sense. In order to find the Fokker Planck equation for the case of the Stratonovich integration of (3.9) we will first find the connection between Itô and Stratonovich integrals. Through this connection we will be able to find the Stratonovich Fokker Planck PDE without explicitly deriving it. More precisely, we rewrite (3.9) in scalar form expressed by the equation:

$$dx_i = \mathbf{f}_i^{\mathbf{S}}(\mathbf{x},t)dt + \sum_{j=1}^{m} \mathbf{B}_{i,j}^{\mathbf{S}}(\mathbf{x},t) \oplus dw_j \tag{3.31}$$

where the terms $\mathbf{f}_i^{\mathbf{S}}(\mathbf{x},t)$ and $\mathbf{B}_{i,j}^{\mathbf{S}}(\mathbf{x},t)$ are given below:

$$\mathbf{f}_i^{\mathbf{S}}(\mathbf{x},t)dt = \mathbf{f}_i\left(\mathbf{x}\left(\frac{t_k+t_{k-1}}{2}\right),t\right)dt, \qquad \mathbf{B}_{i,j}^{\mathbf{S}}(\mathbf{x},t) = \mathbf{B}_{i,j}\left(\mathbf{x}\left(\frac{t_k+t_{k-1}}{2}\right),t\right)$$

We will take the Taylor series expansion of the term above since $\mathbf{x}\left(\frac{t_k+t_{k-1}}{2}\right) = \mathbf{x}(t_{k-1}) + \frac{1}{2}\mathbf{dx}$. More precisely we have that:

$$\begin{aligned}
&\mathbf{f}_i\left(\mathbf{x}(t_{k-1}) + \frac{1}{2}\mathbf{dx}\right)dt \qquad (3.32)\\
&= \mathbf{f}_i\left(\mathbf{x}(t_{k-1})\right)dt + \frac{1}{2}\left(\nabla_{\mathbf{x}(t_{k-1})}\mathbf{f}_i\left(\mathbf{x}(t)\right)\right)^T \mathbf{dx}\ dt\\
&= \mathbf{f}_i\left(\mathbf{x}(t_{k-1})\right)dt + \frac{1}{2}\left(\nabla_{\mathbf{x}(t_{k-1})}\mathbf{f}_i\left(\mathbf{x}(t)\right)\right)^T \left(\mathbf{f}(\mathbf{x},t)dt + \mathbf{B}(\mathbf{x},t)d\mathbf{w}\right)\ dt\\
&= \mathbf{f}_i\left(\mathbf{x}(t_{k-1})\right)dt + \frac{1}{2}\left(\nabla_{\mathbf{x}(t_{k-1})}\mathbf{f}_i\left(\mathbf{x}(t)\right)\right)^T \mathbf{f}(\mathbf{x},t)dt^2 + \frac{1}{2}\left(\nabla_{\mathbf{x}(t_{k-1})}\mathbf{f}_i\left(\mathbf{x}(t)\right)\right)^T \mathbf{B}(\mathbf{x},t)d\mathbf{w}dt
\end{aligned}$$

Since, $dt^2 \to 0$ and $d\mathbf{w}dt \to 0$, the 2nd and 3rd term in the equation above drop and this we have the result:

$$\mathbf{f}_i\bigg(\mathbf{x}(t_{k-1}) + \frac{1}{2}\mathbf{dx}\bigg)dt = \mathbf{f}_i\bigg(\mathbf{x}(t_{k-1})\bigg)dt \tag{3.33}$$

We continue with the term $\mathbf{B}_{i,j}\bigg(\mathbf{x}\left(\frac{t_k+t_{k-1}}{2}\right), t\bigg)$ and we will have that:

$$\begin{aligned}
&\mathbf{B}_{i,j}\bigg(\mathbf{x}(t_{k-1}) + \frac{1}{2}\mathbf{dx}\bigg)dw_j \qquad (3.34)\\
&= \mathbf{B}_{i,j}\bigg(\mathbf{x}(t_{k-1})\bigg)dw_j + \frac{1}{2}\nabla_{\mathbf{x}(t_{k-1})}\mathbf{B}_{i,j}\left(\mathbf{x}(t)\right)^T\ \mathbf{dx}\ dw_j\\
&= \mathbf{B}_{i,j}\bigg(\mathbf{x}(t_{k-1})\bigg)d\omega_j + \frac{1}{2}\nabla_{\mathbf{x}(t_{k-1})}\mathbf{B}_{i,j}\left(\mathbf{x}(t)\right)^T\ \bigg(\mathbf{f}(\mathbf{x}_{t_{k-1}})dt + \mathbf{B}(\mathbf{x}_{t_{k-1}})d\mathbf{w}\bigg)\ dw_j\\
&= \mathbf{B}_{i,j}\bigg(\mathbf{x}(t_{k-1})\bigg)dw_j + \frac{1}{2}\nabla_{\mathbf{x}(t_{k-1})}\mathbf{B}_{i,j}\left(\mathbf{x}(t)\right)^T\mathbf{B}(\mathbf{x}_{t_{k-1}})\ d\mathbf{w}\ dw_j\\
&= \mathbf{B}_{i,j}\bigg(\mathbf{x}(t_{k-1})\bigg)dw_j + \frac{1}{2}\sum_{l=1}^{n}\frac{\partial\mathbf{B}_{i,j}(\mathbf{x}_{t_{k-1}})}{\partial x_l}\ \mathbf{B}_{i,l}(\mathbf{x}_{t_{k-1}})\ dt
\end{aligned}$$

where we have used the fact that $d\mathbf{w}\ dw_j\ = dt$ and $d\mathbf{w}dt \to 0$. By substituting back into (3.31) we will have:

$$dx_i = \mathbf{f}_i(\mathbf{x},t)dt + \sum_{j=1}^{m}\mathbf{B}_{i,j}\left(\mathbf{x}\right)dw_j + \sum_{j=1}^{m}\sum_{l=1}^{n}\frac{\partial\mathbf{B}_{i,j}\left(\mathbf{x}\right)}{\partial x_l}\mathbf{B}_{i,l}\left(\mathbf{x}\right)\ dt \tag{3.35}$$

The stochastic differential equation above is expressed in Itô calculus and it is equivalent to its Stratonovich version equation (3.31). In other words the Stratonovich interpretation of the solution of (3.31) is equivalent to the Itô interpretation of the solution of the equation (3.35). Now that we found the equivalent of the Stratonovich stochastic

differential equation in Itô calculus, the Stratonovich Fokker Planck equation is nothing else than the Itô Fokker Planck equation of the stochastic differential equation:

$$\mathbf{dx} = \Big(\mathbf{f}(\mathbf{x},t) - \boldsymbol{\mathcal{C}}(\mathbf{x})\Big)dt + \mathbf{B}\left(\mathbf{x}\right)d\mathbf{w}, \quad \frac{1}{2}\boldsymbol{\mathcal{C}}_i(\mathbf{x}) = \sum_{j=1}^{m}\sum_{l=1}^{n}\frac{\partial \mathbf{B}_{i,j}\left(\mathbf{x}\right)}{\partial x_l}\mathbf{B}_{i,l}\left(\mathbf{x}\right) \tag{3.36}$$

Thus, the Stratonovich Fokker Planck equation has the form:

$$\boxed{\frac{\partial p(\mathbf{x}|\mathbf{y},t)}{\partial t} = -\nabla_{\mathbf{y}}\cdot\left(\Big(\mathbf{f}(\mathbf{y}) - \boldsymbol{\mathcal{C}}(\mathbf{y})\Big)p(\mathbf{x}|\mathbf{y},t)\right) + \frac{1}{2}tr\left(\nabla_{\mathbf{yy}}\left(\mathbf{B}(\mathbf{y},t)\mathbf{B}(\mathbf{y},t)^T p(\mathbf{x}|\mathbf{y},t)\right)\right)} \tag{3.37}$$

The difference between the Stratonovich and Itô Fokker Planck PDEs is in the extra term $\boldsymbol{\mathcal{C}}(\mathbf{x})$. In the question, which calculus to use, the answer depends on the application and the goal of the underlying derivation. It is generally accepted (Chirikjian 2009),(Øksendal 2003) that the Itô calculus is used for the cases where expectation operations have to be evaluated while the Stratonovich calculus has similar properties with the usual calculus. In this section we have derived the connection between the two calculi and therefore, one could take advantage of both by transforming the Stratonovich interpreted solution of a stochastic differential equation in to its Itô version and then apply Itô calculus. Besides these conceptual differences between the two calculi, there are additional characteristics of Itô integration which we do not find in the Stratonovich calculus and vice versa. More detailed discussion on the properties of the Itô integration can been found in (Øksendal 2003), (Karatzas & Shreve 1991), (Chirikjian 2009) and (Gardiner 2004).

Both Itô and Stratonovich stochastic integrations are special cases of a more general stochastic integration rule in which functions (3.32) are evaluated $\forall \alpha \in [0,1]$ as follows:

$$\mathbf{f}_i^{\alpha}(\mathbf{x},t) = \mathbf{f}_i\Big(\mathbf{x}\left(\alpha t_k + (1-\alpha)t_{k-1}\right), t\Big), \quad \mathbf{B}_{i,j}^{\alpha}(\mathbf{x},t) = \mathbf{B}_{i,j}\Big(\mathbf{x}\left(\alpha t_k + (1-\alpha)t_{k-1}\right), t\Big)$$

Similarly as before since $\mathbf{x}\left(\alpha t_k + (1-\alpha)t_{k-1}\right) = \mathbf{x}\left(t_k\right) + \alpha \mathbf{dx}$ we take the Taylor series expansions of the terms above and therefore we will have that:

$$\mathbf{f}_i\Big(\mathbf{x}\left(t_k\right) + \alpha \mathbf{dx}\Big) = \mathbf{f}_i\Big(\mathbf{x}\left(\alpha t_k + (1-\alpha)t_{k-1}\right), t\Big)dt = \mathbf{f}_i\Big(\mathbf{x}\left(t_k\right), t\Big)dt \tag{3.38}$$

and

$$\mathbf{B}_{i,j}\Big(\mathbf{x}(t_{k-1}) + \alpha \mathbf{dx}\Big)d\omega_j \quad = \mathbf{B}_{i,j}\Big(\mathbf{x}(t_{k-1})\Big)d\omega_j + \alpha \sum_{l=1}^{n} \frac{\partial \mathbf{B}_{i,j}(\mathbf{x}_{t_{k-1}})}{\partial x_l}\ \mathbf{B}_{i,l}(\mathbf{x}_{t_{k-1}})\ dt \tag{3.39}$$

Consequently, the term $\boldsymbol{\mathcal{C}}(\mathbf{x}) \in \Re^{n\times 1}$ in (3.36) is now defined as follows:

$$\boxed{\boldsymbol{\mathcal{C}}_i(\mathbf{x}) = \alpha \sum_{j=1}^{m}\sum_{l=1}^{n} \frac{\partial \mathbf{B}_{i,j}\left(\mathbf{x}\right)}{\partial x_l}\mathbf{B}_{i,l}\left(\mathbf{x}\right)} \tag{3.40}$$

Clearly, for $\alpha = 0$ we have the Itô calculus while for $\alpha = \frac{1}{2}$ we have the Stratonovich.

3.3 Path integrals and SDEs

The path integral formalism can been thought as an alternative, to Fokker Planck and Langevin equations, mathematical description of nonlinear stochastic dynamics (Lau & Lubensky 2007). In this section we will start with the derivation of the path integral formalism for the one dimensional stochastic differential equation:

$$dx = f(x,t)dt + B(x,t)dw \tag{3.41}$$

We start with the 1 dimensional cases because it is easier to understand the derivation and the rational behind the path integral concept. After the 1 dimensional case extensions to multidimensional cases are strait-forward for the case of Itô integration while the Stratonovich involves additional analysis. We will discretize the 1-dimensional stochastic differential equation as follows:

$$\begin{aligned} x(t+\delta t) - x(t) = &\int_t^{t+\delta t} f\Big(\beta x(t) + (1-\beta)x(t+\delta t)\Big)d\tau \\ &+ \int_t^{t+\delta t} B\Big(\alpha x(t) + (1-\alpha)x(t+\delta t)\Big)n(\tau)d\tau \end{aligned}$$

where the constants $\alpha, \beta \in [0,1]$. Since the drift and the diffusion term are evaluated at $\beta x(t) + (1-\beta)x(t+\delta t)$ and $\alpha x(t) + (1-\alpha)x(t+\delta t)$ they can be taken outside the integral and thus the equations above is expressed as:

$$x(t+\delta t)-x(t) = f\Big(\beta x(t)+(1-\beta)x(t+\delta t)\Big)\delta t+B\Big(\alpha x(t)+(1-\alpha)x(t+\delta t)\Big)\int_t^{t+\delta t} n(\tau)d\tau \tag{3.42}$$

The path integral derivation is based on the statistics of the state path $x(t_0 \to t_N)$. We discretize the state path to the segments $x(t_i)$ with $t_0 < t_1 < t_2 < ... < t_N$ and we define $\delta t = t_i - t_{i-1}$. The probability of the path now is defined as follows:

$$P\Big(x_N, t_N; x_{N-1}, t_{N-1}; ...; x_1, t_1|x_0, t_0\Big) = \Big\langle \delta[x_N - \phi(t_N; x_0, t_0]...\delta[x_1 - \phi(t_1; x_0, t_0]\Big\rangle$$

The function $\phi(t_i; x_{i-1}, t_{i-1})$ is the solution of the stochastic differential equation (3.41) for $x(t_i)$ given that $x(t_{i-1}) = x_{i-1}$. Due to the fact that the noise is delta correlated, in different time intervals the noise is uncorrelated and therefore we will have that:

$$P\Big(x_N, t_N; x_{N-1}, t_{N-1}; ...; x_1, t_1|x_0, t_0\Big) = \prod_{i=1}^{N}\Big\langle \delta[x_i - \phi(t_i; x_{i-1}, t_{i-1}]\Big\rangle$$

where the function $\Big\langle \delta[x_i - \phi(t_i; x_{i-1}, t_{i-1}]\Big\rangle = P\Big(x_i, t_i|x_{i-1}, t_{i-1}\Big)$ and thus it corresponds to conditional probability that the random variable $x(t)$ is state x_i at time t_i given that at t_{i-1} we have that $x(t_{i-1}) = x_{i-1}$. We can use the transition probabilities to calculate the probability $P(x_N, t_N|x_0, t_0)$ that is the probability of being at state $x(t_N) = x_N$ given that the initial state is $x(t_0) = x_0$ at time t_0. More precisely we have that:

$$P\Big(x_N, t_N|x_0, t_0\Big) = \int dx_{N-1} ... \int dx_1 \prod_{i=1}^{N} \Big\langle \delta[x_i - \phi(t_i; x_{i-1}, t_{i-1}] \Big\rangle \tag{3.43}$$

To find the path integral we need to evaluated the function $\delta[x_i - \phi(t_i; x_{i-1}, t_{i-1}]$ and then substitute to the equation above. The analysis for the evaluation of the function $\delta[x_i - \phi(t_i; x_{i-1}, t_{i-1})]$ requires the discretized version of the stochastic differential equation (3.41) expressed in (3.42). We can rewrite discrete version in the form:

$$x_i = x_{i-1} + \delta t \ f_i + B_i \int_{t_{i-1}}^{t_i} n(\tau) d\tau$$

where $f_i = f\left(\beta x_i + (1-\beta) x_{i-1}\right)$ and $B_i = B\left(\alpha x_i + (1-\alpha) x_{i-1}\right)$ and we introduce the function $h(x_i, x_{i-1})$ defined as follows:

$$h(x_i, x_{i-1}) = \frac{x_i - x_{i-1} - f_i \delta t}{B_i} - \int_{t_{i-1}}^{t_i} n(\tau) d\tau$$

for which the condition $h[\phi(t_i; x_{i-1}, t_{i-1}), x_{i-1}] = 0$. By using the property of the delta function $\delta(g(x)) = \frac{\delta(x - x_0)}{|g'(x_0|}$ we will have that:

$$\delta[h(x_i, x_{i-1})] = \left| \frac{\partial h(x_i, x_{i-1})}{\partial x_i} \right|^{-1}_{x_i = \phi(t_i)} \delta[x_i - \phi(t_i; x_{i-1}, t_{i-1})]$$

The transition probability $P(x_i, t_i|x_{i-1}, t_{i-1})$ is now written as:

$$P(x_i, t_i|x_{i-1}, t_{i-1}) = \Big\langle \delta[x_i - \phi(t_i; x_{i-1}, t_{i-1})] \Big\rangle = \left| \frac{\partial h(x_i, x_{i-1})}{\partial x_i} \right|_{x_i=\phi(t_i)} \Big\langle \delta[h(x_i, x_{i-1})] \Big\rangle$$

The term $\frac{\partial h(x_i, x_{i-1})}{\partial x_i}$ is expressed by the equation:

$$\begin{aligned} \frac{\partial h(x_i, x_{i-1})}{\partial x_i} &= \frac{\Big(1 - \beta\ \delta t\ (\partial_x f_i)\Big) B_i - \Big(x_i - x_{i-1} - f_i\ \delta t\Big)(\partial_x B_i)}{B_i^2} \\ &= \frac{1}{B_i}\left[1 - \beta\ \delta t\ (\partial_x f_i) - \alpha \frac{(\partial_x B_i)}{B_i}\Big(x_i - x_{i-1} - f_i\ \delta t\Big)\right] \end{aligned}$$

From the property of the inverse Fourier transform of the delta function $\delta(t) = \int_{-\infty}^{+\infty} d\omega \exp(j\omega t) \frac{1}{2\pi}$ with $j^2 = -1$ we will have that:

$$\begin{aligned} \delta[h(x_i, x_{i-1})] &= \int \frac{d\omega}{2\pi} \exp\Big(j\omega h(x_i, x_{i-1})\Big) \\ &= \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \exp\left(j\omega\left(\frac{x_i - x_{i-1} - f_i \delta t}{B_i} - \int_{t_{i-1}}^{t_i} n(\tau) d\tau\right)\right) \end{aligned}$$

The expectation operator over the noise results in:

$$\Big\langle \delta[h(x_i, x_{i-1})] \Big\rangle = \int \frac{d\omega}{2\pi} \exp\left(j\omega \frac{x_i - x_{i-1} - f_i \delta t}{B_i}\right) \Big\langle \exp\left(j\omega \int_{t_{i-1}}^{t_i} n(\tau) d\tau\right) \Big\rangle$$

By considering the Taylor series expansion of the exponential function and applying the expectation, it can be shown that:

$$\begin{aligned}
&\left\langle \exp\left(j\omega \int_{t_{i-1}}^{t_i} n(\tau)d\tau\right)\right\rangle = \\
&= \left\langle 1 + \frac{j\omega \int_{t_{i-1}}^{t_i} n(\tau)d\tau}{1!} - \frac{\omega^2(\int_{t_{i-1}}^{t_i} n(\tau)d\tau)^2}{2!} - \frac{j\omega^3(\int_{t_{i-1}}^{t_i} n(\tau)d\tau)^3}{3!}....\right\rangle \\
&= 1 + \frac{\omega^2\delta t}{2} = \exp -(\frac{1}{2}\omega^2\delta t)
\end{aligned}$$

Therefore $\left\langle \delta[h(x_i, x_{i-1})]\right\rangle = \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i} - \frac{1}{2}\omega^2\delta t)\right)$. By putting everything together the transition probability $P(x_i, t_i|x_{i-1}, t_{i-1})$ will take the form:

$$\begin{aligned}
P(x_i, t_i|x_{i-1}, t_{i-1}) = \int \frac{d\omega}{2\pi B_i} \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i} + \frac{1}{2}\omega^2\delta t\right) \\
\times \left[1 - \beta\ \delta t\ (\partial_x f_i) - \alpha \frac{(\partial_x B_i)}{B_i}\left(x_i - x_{i-1} - f_i\ \delta t\right)\right]
\end{aligned}$$

From the expression above we work with the term:

$$\begin{aligned}
&- \frac{\alpha(\partial_x B_i)}{2\pi B_i} \int d\omega \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i} + \frac{1}{2}\omega^2\delta t\right)\left(\frac{x_i - x_{i-1} - f_i\ \delta t}{B_i}\right) \\
&= -\frac{\alpha(\partial_x B_i)}{2\pi B_i} \int d\omega\ j\ \exp\left(-\frac{1}{2}\omega^2\delta t\right)\ \partial_\omega\left(\exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i}\right)\right) \\
&= -\frac{\alpha(\partial_x B_i)}{2\pi B_i} \int d\omega\ \left(j\omega\delta t\right)\ \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i} - \frac{1}{2}\omega^2\delta t\right)
\end{aligned}$$

The transition probability is expressed as follows:

$$P(x_i, t_i|x_{i-1}, t_{i-1}) = \int \frac{d\omega}{2\pi B_i} \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t}{B_i} - \frac{1}{2}\omega^2\delta t\right) \times \left[1 - \beta\ \delta t\ (\partial_x f_i) - j\omega\alpha(\partial_x B_i)\delta t\right]$$

The term $\left[1 - \beta\ \delta t\ (\partial_x f_i) - j\omega\alpha(\partial_x B_i)\delta t\right] \simeq \exp\left[1 - \beta\ \delta t\ (\partial_x f_i) - j\omega\alpha(\partial_x B_i)\delta t\right]$ as $\delta t \to 0$. Thus we will have:

$$\begin{aligned}
&P(x_i, t_i|x_{i-1}, t_{i-1}) \\
&= \int \frac{d\omega}{2\pi B_i} \exp\left(j\omega \frac{x_i - x_{i-1} - f_i\delta t + \alpha(\partial_x B_i)\delta t B_i}{B_i} - \frac{1}{2}\omega^2\delta t - \beta\ \delta t\ (\partial_x f_i)\right) \\
&= \int \frac{d\omega}{2\pi B_i} \exp\left(\left[j\frac{\omega}{B_i}\left(\frac{\delta x}{\delta t} - f_i + \alpha(\partial_x B_i)B_i\right) - \frac{1}{2}\omega^2\delta t - \beta\ (\partial_x f_i)\right]\delta t\right) \\
&= \frac{\exp\left[-\beta(\partial_x f_i)\delta t\right]}{\sqrt{2\pi\delta t}B_i} \int \frac{d\omega}{\sqrt{2\pi\delta t}B_i} \exp\left(\left[j\frac{\omega}{B_i}\left(\frac{\delta x}{\delta t} - f_i + \alpha(\partial_x B_i)B_i\right) - \frac{1}{2}\omega^2\right]\delta t\right)
\end{aligned}$$

we define the quantity $\eta = \frac{j}{B_i}\left(\frac{\delta x}{\delta t} - f_i + \alpha(\partial_x B_i)B_i\right)$. We can now write the transition probability as follows:

$$\begin{aligned}
&P(x_i, t_i|x_{i-1}, t_{i-1}) \\
&= \frac{\exp\left[-\beta(\partial_x f_i)\delta t\right]}{\sqrt{2\pi\delta t}B_i} \int \frac{d\omega}{\sqrt{2\pi\delta t}B_i} \exp\left(\left[\omega\eta - \frac{1}{2}\omega^2\right]\delta t\right) \\
&= \frac{\exp\left[-\beta(\partial_x f_i)\delta t\right]}{\sqrt{2\pi\delta t}B_i} \int \frac{d\omega}{\sqrt{2\pi\delta t}B_i} \exp\left(\left[\omega\eta - \frac{1}{2}\omega^2 - \frac{1}{2}\eta^2\right]\delta t\right) \exp\left(\frac{1}{2}\eta^2\delta t\right)
\end{aligned}$$

$$= \frac{\exp\left[-\beta(\partial_x f_i)\delta t\right]}{\sqrt{2\pi\delta t}B_i}\exp\left(\frac{1}{2}\eta^2\delta t\right)\int\frac{d\omega}{\sqrt{2\pi\delta t}B_i}\exp\left(\left[\omega\eta-\frac{1}{2}\omega^2-\frac{1}{2}\eta^2\right]\delta t\right)$$
$$= \frac{\exp\left[-\beta(\partial_x f_i)\delta t\right]}{\sqrt{2\pi\delta t}B_i}\exp\left(\frac{1}{2}\eta^2\delta t\right)$$
$$= \frac{1}{\sqrt{2\pi\delta t}B_i}\exp\left(-\left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t}-f_i-\alpha(\partial_x B_i)B_i\right)^2+\beta\ (\partial_x f_i)\right]\delta t\right)$$

The last line is valid up to the first order in δt. Clearly the path integral is given by the product of all the transition probabilities along the path $x_0, x_1, ..., x_N$. More precisely we will have that:

$$P\left(x_N, t_N|x_0, t_0\right) = \int\frac{dx_1}{\sqrt{2\pi\delta t}B_i}...\int\frac{dx_{N-1}}{\sqrt{2\pi\delta t}B_{N-1}}\int\frac{dx_N}{\sqrt{2\pi\delta t}B_N}$$
$$\times\exp\left(-\sum_{i=1}^{N}\left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t}-f_i+\alpha(\partial_x B_i)B_i\right)^2+\beta\ (\partial_x f_i)\right]\delta t\right)$$
$$= \int_{x_0}^{x_N}\mathcal{D}(x)\ e^{-\mathcal{S}(x)}$$

where $\mathcal{D}(x) = \prod_{i=1}^{N}\frac{dx_i}{\sqrt{2\pi\delta t}B_i}$ and $\mathcal{S}(x) = \sum_{i=1}^{N}\left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t}-f_i+\alpha(\partial_x B_i)B_i\right)^2+\beta\ (\partial_x f_i)\right]\delta t$ is the action defined on the path $x_1, ..., x_N$.

3.3.1 Path integral in Stratonovich calculus

The path integral defined above is general since for different values of the parameters α and β it can recover the Stratonovich and the Itô path integral mathematical forms. In particular, for the values $\alpha = \beta = \frac{1}{2}$ the path integral above corresponds to the

Stratonovich calculus while for the cases of $\alpha = \beta = 0$ the resulting path integral corresponds to the Itô calculus. Thus the Stratonovich path integral for the 1 dimensional case is expressed as

$$P\Big(x_N, t_N|x_0, t_0\Big) = \int \frac{dx_1}{\sqrt{2\pi\delta t}B_1} ... \int \frac{dx_{N-2}}{\sqrt{2\pi\delta t}B_{N-2}} \int \frac{dx_{N-1}}{\sqrt{2\pi\delta t}B_{N-1}} \times \exp\Bigg(-\sum_{i=1}^{N} \Bigg[\frac{1}{2B_i^2} \Bigg(\frac{\delta x}{\delta t} - f_i + \frac{1}{2}(\partial_x B_i)B_i \Bigg)^2 - \frac{1}{2}\,(\partial_x f_i) \Bigg] \delta t \Bigg)$$

or in a more compact form:

$$\boxed{P\Big(x_N, t_N|x_0, t_0\Big) = \int \prod_{i=1}^{N-1} \frac{dx_i \exp\Bigg(-\sum_{i=1}^{N} \Bigg[\Big(\frac{\frac{\delta x}{\delta t} - f_i + \frac{1}{2}(\partial_x B_i)B_i}{\sqrt{2}B_i} \Big)^2 - \frac{1}{2}\,(\partial_x f_i) \Bigg] \delta t \Bigg)}{\sqrt{2\pi\delta t}B_i}}$$

where $f_i = f\left(0.5x_i + 0.5x_{i-1}\right)$ and $B_i = B\left(0.5x_i + 0.5x_{i-1}\right)$.

3.3.2 Path integral in Itô calculus

Similarly the Itô path integral for the scalar case is expressed as:

$$P\Big(x_N, t_N|x_0, t_0\Big) = \int \frac{dx_1}{\sqrt{2\pi\delta t}B_1} ... \int \frac{dx_{N-2}}{\sqrt{2\pi\delta t}B_{N-2}} \int \frac{dx_{N-1}}{\sqrt{2\pi\delta t}B_{N-1}} \times \exp\Bigg(\sum_{i=1}^{N} \Bigg[\frac{1}{2B_i^2} \Bigg(\frac{\delta x}{\delta t} - f_i \Bigg)^2 \Bigg] \delta t \Bigg)$$

or in a more compact form:

$$\boxed{P\Big(x_N, t_N | x_0, t_0\Big) = \int \prod_{i=1}^{N-1} \frac{dx_i}{\sqrt{2\pi\delta t B_i}} \times \exp\Big(-\sum_{i=1}^{N} \Big[\frac{1}{2B_i^2}\Big(\frac{\delta x}{\delta t} - f_i\Big)^2\Big]\delta t\Big)}$$

where $f_i = f\left(x_{i-1}\right)$ and $B_i = B\left(x_{i-1}\right)$.

3.4 Path integrals and multi-dimensional SDEs

In this section we derive the path integral for the multidimensional SDE (Schulz 2006). More precisely we consider the multidimensional SDE:

$$\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)d\mathbf{w} \tag{3.44}$$

in which $\mathbf{x} \in \Re^{n\times 1},\ \mathbf{f}(\mathbf{x}) : \Re^{n\times 1} \to \Re^{n\times 1}$ and $\mathbf{B}(\mathbf{x}) : \Re^{n\times 1} \to \Re^{n\times p}$. We will consider the Itô representation of the SDE:

$$\mathbf{x}(t_i) = \mathbf{x}(t_{i-1}) + \mathbf{f}(\mathbf{x}, t)d\tau + \int_{t_{i-1}}^{t_i} \mathbf{B}(\mathbf{x}, t)d\mathbf{w}(\tau) \tag{3.45}$$

Similarly to the 1D case we define $\boldsymbol{\phi}(\mathbf{x}_{t_{i-1}}, t_{i-1})$ as the solution of the SDE above, as follows:

$$\boldsymbol{\phi}(\mathbf{x}_{t_{i-1}}, t_{i-1}) = \mathbf{x}(t_{i-1}) + \mathbf{f}(\mathbf{x}, t)d\tau + \int_{t_{i-1}}^{t_i} \mathbf{B}(\mathbf{x}, t)d\mathbf{w}(\tau) \tag{3.46}$$

Moreover we define the term $\mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})$ as:

$$\mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}}) = \mathbf{x}(t_i) - \mathbf{x}(t_{i-1}) - \mathbf{f}(\mathbf{x}, t)d\tau - \int_{t_{i-1}}^{t_i} \mathbf{B}(\mathbf{x}, t)d\mathbf{w}(\tau) \tag{3.47}$$

The probability of hitting state $\mathbf{x}_N$ at t_N starting from $\mathbf{x}_0$ at t_0 is formulated as follows:

$$P\Big(\mathbf{x}_N, t_N|\mathbf{x}_0, t_0\Big) = \int d\mathbf{x}_{N-1} ... \int d\mathbf{x}_1 \prod_{i=1}^{N} \Big\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}]\Big\rangle \tag{3.48}$$

To calculate the probability above the delta function $\Big\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}]\Big\rangle$ has to be found. The Fourier representation of delta function yields:

$$\begin{aligned}
\delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}] &= det\Big(\mathbf{J}_{\mathbf{x}_{t_i}} \mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\Big)\delta[\mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})] \\
&= det\Big(\mathbf{J}_{\mathbf{x}_{t_i}} \mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\Big) \int \frac{d\mathbf{w}}{(2\pi)^n} \exp\Big(j\mathbf{w}^T \mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\Big)
\end{aligned}$$

where $\mathbf{J}_{\mathbf{x}_{t_i}}$ is the Jacobian. For the Itô SDE above the jacobian $\mathbf{J}_{\mathbf{x}_{t_i}} = I_{n\times n}$ and therefore $det\Big(\mathbf{J}_{\mathbf{x}_{t_i}} \mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\Big) = 1$. Thus we will have that:

$$\begin{aligned}
\delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}] &= \delta[\mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})] \\
\Big\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}]\Big\rangle &= \Big\langle \int \frac{d\mathbf{w}}{(2\pi)^n} \exp\Big(j\mathbf{w}^T \mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\Big)\Big\rangle
\end{aligned}$$

Substitution of the term $\mathbf{h}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})$ yields the equation:

$$\left\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}] \right\rangle = \int \frac{d\mathbf{w}}{(2\pi)^n} \exp\left(j\mathbf{w}^T \left(\mathbf{x}(t_i) - \mathbf{x}(t_{i-1}) - \mathbf{f}(\mathbf{x}, t) d\tau \right) \right) \times \left\langle \exp\left(j\mathbf{w}^T \mathbf{B}(\mathbf{x}_{t_{i-1}}, t_{i-1}) d\mathbf{w}(t_{i-1}) \right) \right\rangle$$

We will further analyze the term:

$$\left\langle \exp\left(j\mathbf{w}^T \mathbf{B}(\mathbf{x}_{t_{i-1}}, t_{i-1}) d\mathbf{w}(t_{i-1}) \right) \right\rangle = \left\langle I + \frac{1}{1!} j\mathbf{w}^T \mathbf{B}(\mathbf{x}, t) d\mathbf{w}(t) \right\rangle - \left\langle \frac{1}{2!} \mathbf{w}^T \mathbf{B}(\mathbf{x}, t) d\mathbf{w}(t) d\mathbf{w}(t)^T \mathbf{B}(\mathbf{x}, t)^T \mathbf{w} \right\rangle + \left\langle \boldsymbol{\mathcal{O}}(dw_i(t) dw_j(t) dw_k(t)) \right\rangle$$

Since $d\mathbf{w}(t)$ is Wiener noise we have that $\left\langle d\mathbf{w}(t) d\mathbf{w}(t)^T \right\rangle = I_{n \times n} dt$. In addition the term $\boldsymbol{\mathcal{O}}(dw_i(t) dw_j(t) dw_k(t))$ has terms of order higher than quadratic in dw_i. The expectation of this term will result zero. More precisely, for these terms that are of order $\nu > 2$, where ν is an even number the expectation result in terms of order $\mu > 1$ in dt and therefore all these terms are zero. For the remaining terms, of order order $\nu > 2$, where ν is an odd number, the expectation will result in zero since $\left\langle d\mathbf{w}(t) \right\rangle = 0$. Thus, since $\lim_{dt \to 0} \left\langle \boldsymbol{\mathcal{O}}(dw_i(t) dw_j(t) dw_k(t)) \right\rangle = 0$ we will have:

$$\left\langle \exp\left(j\mathbf{w}^T \mathbf{B}(\mathbf{x}_{t_{i-1}}, t_{i-1}) d\mathbf{w}(t_{i-1}) \right) \right\rangle = I - \frac{1}{2!} \mathbf{w}^T \mathbf{B}(\mathbf{x}, t) \mathbf{B}(\mathbf{x}, t)^T \mathbf{w} dt = \exp\left(-\frac{1}{2} \mathbf{w}^T \mathbf{B}(\mathbf{x}, t) \mathbf{B}(\mathbf{x}, t)^T \mathbf{w} dt \right)$$

By substituting back we will have:

$$\begin{aligned}\left\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i;\mathbf{x}_{i-1},t_{i-1}]\right\rangle &= \int \frac{d\mathbf{w}}{(2\pi)^n}\exp\left(j\mathbf{w}^T\left(\mathbf{x}(t_i)-\mathbf{x}(t_{i-1})-\mathbf{f}(\mathbf{x},t)d\tau\right)\right)\\ &\quad\times\exp\left(-\frac{1}{2}\mathbf{w}^T\mathbf{B}(\mathbf{x},t)\mathbf{B}(\mathbf{x},t)^T\mathbf{w}dt\right)\\ &= \int \frac{d\mathbf{w}}{(2\pi)^n}\exp\left(j\mathbf{w}^T\boldsymbol{\mathcal{A}}\right)\exp\left(-\frac{1}{2}\mathbf{w}^T\boldsymbol{\mathcal{B}}\mathbf{w}dt\right)\end{aligned}$$

where $\boldsymbol{\mathcal{A}} = \mathbf{x}(t_i) - \mathbf{x}(t_{i-1}) - \mathbf{f}(\mathbf{x},t)d\tau$ and $\boldsymbol{\mathcal{B}} = \mathbf{B}(\mathbf{x},t)\mathbf{B}(\mathbf{x},t)^T$. The transition probability therefore is formulated as follows:

$$\boxed{\left\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i;\mathbf{x}_{i-1},t_{i-1}]\right\rangle = \int \frac{d\mathbf{w}}{(2\pi)^n}\exp\left(j\mathbf{w}^T\boldsymbol{\mathcal{A}} - \frac{1}{2}\mathbf{w}^T\boldsymbol{\mathcal{B}}\mathbf{w}dt\right)} \tag{3.49}$$

This form of the transition probability is very common in the physics community. In engineering fields, the transition probability is derived according to the distribution of the state space noise that is considered to be Gaussian distributed. Therefore it seems that the transition above is different than the one that would have been derived if we were considering the Gaussian distribution of the state space noise. However as we will show in the rest of our analysis, (3.49) takes the form of Gaussian distribution. More precisely we will have that:

$$\begin{aligned}&\left\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i;\mathbf{x}_{i-1},t_{i-1}]\right\rangle\\ &= \int \frac{d\mathbf{w}}{(2\pi)^n}\exp\left(j\mathbf{w}^T\boldsymbol{\mathcal{B}}dt(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}} - \frac{1}{2}\mathbf{w}^T\boldsymbol{\mathcal{B}}\mathbf{w}dt\right)\exp\left(-\frac{j^2}{2}\boldsymbol{\mathcal{A}}^T(\boldsymbol{\mathcal{B}}dt)^{-T}\boldsymbol{\mathcal{B}}dt(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\right)\end{aligned}$$

$$\times \exp\left(\frac{j^2}{2}\boldsymbol{\mathcal{A}}^T(\boldsymbol{\mathcal{B}}dt)^{-T}\boldsymbol{\mathcal{B}}dt(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\right)$$

$$= \int \frac{d\mathbf{w}}{(2\pi)^n}\exp\Big[-\frac{\Big(j\mathbf{w}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)^T(\boldsymbol{\mathcal{B}}dt)\Big(j\boldsymbol{\omega}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)}{2}\Big]$$

$$\times \exp\Big(-\frac{1}{2}\boldsymbol{\mathcal{A}}^T(\boldsymbol{\mathcal{B}}dt)^{-T}\boldsymbol{\mathcal{B}}dt(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)$$

$$= \frac{\sqrt{det\,(\boldsymbol{\mathcal{B}}dt)}}{\sqrt{(2\pi)^n}} \times \int d\mathbf{w}\exp\Big[\Big(j\mathbf{w}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)^T(\boldsymbol{\mathcal{B}}dt)\Big(j\boldsymbol{\omega}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)\Big]$$

$$\times \frac{1}{\sqrt{(2\pi)^n det\,(\boldsymbol{\mathcal{B}}dt)}}\exp\Big(-\boldsymbol{\mathcal{A}}^T(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)$$

since for term:

$$\frac{\sqrt{det\,(\boldsymbol{\mathcal{B}}dt)}}{\sqrt{(2\pi)^n}}\int d\mathbf{w}\exp\Big[\Big(j\mathbf{w}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)^T(\boldsymbol{\mathcal{B}}dt)\Big(j\boldsymbol{\omega}+(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)\Big] = 1 \qquad (3.50)$$

Finally we will have that:

$$\Big\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i;\mathbf{x}_{i-1},t_{i-1}]\Big\rangle = \frac{1}{\sqrt{(2\pi)^n det\,(\boldsymbol{\mathcal{B}}dt)}}\exp\Big(-\frac{1}{2}\boldsymbol{\mathcal{A}}^T(\boldsymbol{\mathcal{B}}dt)^{-1}\boldsymbol{\mathcal{A}}\Big)$$

Clearly the transition probability above has the form of a Gaussian distribution. Substitution of the transition probabilities in (3.48) yields the final result:

$$\boxed{P\Big(\mathbf{x}_N,t_N|\mathbf{x}_0,t_0\Big) = \int\prod_{i=1}^{N}\frac{\mathbf{dx}_i}{(2\pi\delta t)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}}\times\exp\Big(-\frac{1}{2}\sum_{i=1}^{N}\Big\|\frac{\delta\mathbf{x}}{\delta t}-\mathbf{f}(\mathbf{x})\Big\|^2_{\mathbf{B}\mathbf{B}^T}\delta t\Big)}$$

where the term $\left\| \frac{\delta \mathbf{x}}{\delta t} - \mathbf{f}(\mathbf{x}) \right\|^2_{\mathbf{B}\mathbf{B}^T} = \left(\frac{\delta \mathbf{x}}{\delta t} - \mathbf{f}(\mathbf{x}) \right)^T \mathbf{B}\mathbf{B}^T \left(\frac{\delta \mathbf{x}}{\delta t} - \mathbf{f}(\mathbf{x}) \right)$. With this section we have derived the path integral from the stochastic differential equation and therefore we have completed the presentation of the connection between the three different ways of mathematically expressing nonlinear stochastic dynamics. These 3 different mathematical representations are the stochastic differential equations, the corresponding Fokker Planck PDE and the path integral formalism. In the next section we focus on forward and backward PDEs, the so called forward and backward Chapman Kolmogorov PDEs, and we discuss the probabilistic representation of the their solutions.

3.5 Cauchy problem and the generalized Feynman Kac representation

The Feynman- Kac lemma provides a connection between stochastic differential equations and PDEs and therefore its use is twofold. On one side it can be used to find probabilistic solutions of PDEs based on forward sampling of diffusions while on the other side it can be used to find solution of SDEs based on deterministic methods that numerically solve PDEs. There are many cases in stochastic optimal control and estimation in which PDEs appear. In fact as we have seen in chapter 2, on the control side there is the so called Hamilton-Jacobi-Bellman PDEs which describes how the value function $V(\mathbf{x}, t)$ of a stochastic optimal control problem varies as a function of time t and state $\mathbf{x}$.

In this work, we compute the solution of the linear version of the HJB above with the use of the Feynman - Kac lemma (Øksendal 2003),and thus, in this section we provide the generalized version of the Feynman-Kac Lemma based on the theoretical development in

(Karatzas & Shreve 1991) and (Friedman 1975). This lemma is of a great significance for our analysis and application of path integral control and therefore we believe that it is essential to provide the proof of the lemma.

Let us assume an arbitrary but fixed time $T > 0$ and the constant $L > 0$ and $\lambda \geq 0$. Furthermore we consider the functions $\Psi(\mathbf{x}, t) : [0, T] \times \Re^{n\times 1} \to \Re$, $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) : [0, T]\times \in \Re^{n\times 1} \to \Re$ and $q(\mathbf{x}, t) : [0, T] \times \Re^{n\times 1} \to [0, \infty]$ to be continuous and satisfying the conditions:

$$(i)\ \ |\Psi(\mathbf{x}, t)| \leq L\left(1 + ||\mathbf{x}||^{2\lambda}\right) \quad or \quad (ii)\ \ \Psi(\mathbf{x}, t) \geq 0; \quad \forall \mathbf{x} \in \Re^{n\times 1} \tag{3.51}$$

$$(iii)\ \ |\boldsymbol{\mathcal{F}}(\mathbf{x}, t)| \leq L\left(1 + ||\mathbf{x}||^{2\lambda}\right) \quad or \quad (iv)\ \ \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \geq 0; \quad 0 \leq t \leq T,\ \mathbf{x} \in \Re^{n\times 1} \tag{3.52}$$

Feynman - Kac Theorem: *Suppose that the coefficients* $\mathbf{f}_i(\mathbf{x})$ *and* $\mathbf{B}_{i,j}(\mathbf{x})$ *satisfy the linear growth condition* $||\mathbf{f}_i(\mathbf{x})||^2 + ||\mathbf{B}_{i,j}(\mathbf{x})||^2 \leq K^2(1 + ||\mathbf{x}||^2)$ *where* K *is a positive constant. Let* $\Psi(\mathbf{x}, t) : [0, T] \times \Re^{n\times 1} \to \Re$ *is continuous and* $\Psi(\mathbf{x}, t) \in C^{1,2}$ *and it satisfies the Cauchy problem:*

$$\boxed{-\partial_t \Psi_t = -\frac{1}{\lambda} q_t \Psi_t + \mathbf{f}_t^T (\nabla_{\mathbf{x}} \Psi_t) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t)\mathbf{B}\mathbf{B}^T\right) + \boldsymbol{\mathcal{F}}(\mathbf{x}, t)} \tag{3.53}$$

in $[0, T) \times \Re^{n\times 1}$ *with the boundary condition:*

$$\boxed{\Psi(\mathbf{x}, T) = \xi(\mathbf{x});\ \mathbf{x} \in \Re(n \times 1)} \tag{3.54}$$

as well as the polynomial growth condition:

$$\max_{0\leq t\leq T}|\Psi(\mathbf{x},t)|\leq M\left(1+||\mathbf{x}||^{2\mu}\right);\mathbf{x}\in\Re^{n\times 1} \tag{3.55}$$

For some $M>0,\mu\geq 1$ *then* $\Psi(\mathbf{x},t)$ *admits the stochastic representation*

$$\boxed{\Psi(\mathbf{x},t)=\left\langle \xi(\mathbf{x}_T)\exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}_s,s)ds\right)+\int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta,\theta)\exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x},s)ds\right)d\theta\right\rangle} \tag{3.56}$$

on $[0,T]\times\Re^{n\times 1}$*; in particular, such a solution is unique.*

Proof: Let us consider $\boldsymbol{\mathcal{G}}(\mathbf{x},t_{0)},t)=\Psi(\mathbf{x},t)\ \boldsymbol{\mathcal{Z}}(t_0,t)$ where the term $\boldsymbol{\mathcal{Z}}(t_0,t)$ is defined as follows:

$$\boldsymbol{\mathcal{Z}}(t_0,t)=\exp\left(-\frac{1}{\lambda}\int_{t_0}^t Q(\mathbf{x})d\tau\right) \tag{3.57}$$

We apply the multidimensional version of the Itô lemma:

$$d\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t)=d\Psi(\mathbf{x},t)\ \boldsymbol{\mathcal{Z}}(t_0,t)+\Psi(\mathbf{x},t)\ d\boldsymbol{\mathcal{Z}}(t_0,t)+d\Psi(\mathbf{x},t)\ d\boldsymbol{\mathcal{Z}}(t_0,t) \tag{3.58}$$

Since $d\Psi(\mathbf{x},t)\ d\boldsymbol{\mathcal{Z}}(t_0,t)=0$ we will have that: $d\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t)=d\Psi(\mathbf{x},t)\ \boldsymbol{\mathcal{Z}}(t,t_N)+\Psi(\mathbf{x},t)\ d\boldsymbol{\mathcal{Z}}(t_0,t)$. We calculate the differentials $d\Psi(\mathbf{x},t),d\boldsymbol{\mathcal{Z}}(t,t_N)$ according to the Itô differentiation rule. More precisely for the term $d\boldsymbol{\mathcal{Z}}(t_0,t_N)$ we will have that:

$$d\boldsymbol{\mathcal{Z}}(t_0,t) = -\frac{1}{\lambda}Q(\mathbf{x})\boldsymbol{\mathcal{Z}}(t_0,t)\ dt \tag{3.59}$$

while the term $d\Psi(\mathbf{x},t)$ is equal to:

$$\begin{aligned} d\Psi(\mathbf{x},t) &= \partial_t\Psi\ dt + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{dx} + \frac{1}{2}\mathbf{dx}^T\ (\nabla_{\mathbf{xx}}\Psi)\ \mathbf{dx} \\ &= \partial_t\Psi\ dt + (\nabla_{\mathbf{x}}\Psi)^T\left(\mathbf{f}(\mathbf{x},t)dt + \mathbf{B}(\mathbf{x})d\mathbf{w}\right) \\ &+ \frac{1}{2}\left(\mathbf{f}(\mathbf{x},t)dt + \mathbf{B}(\mathbf{x})d\mathbf{w}\right)^T(\nabla_{\mathbf{xx}}\Psi)\left(\mathbf{f}(\mathbf{x},t)dt + \mathbf{B}(\mathbf{x})d\mathbf{w}\right) \end{aligned} \tag{3.60}$$

Since the following properties (Øksendal 2003) hold, $d\mathbf{w}^T d\mathbf{w} \to 0$, $d\mathbf{w}dt \to 0$, the equation above is further simplified into:

$$\begin{aligned} d\Psi(\mathbf{x},t) = \partial_t\Psi\ dt + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{f}(\mathbf{x},t)\ dt + \frac{1}{2}\left(\mathbf{B}(\mathbf{x})d\mathbf{w}\right)^T(\nabla_{\mathbf{xx}}\Psi)\left(\mathbf{B}(\mathbf{x})\mathbf{L}d\mathbf{w}\right) \\ + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{B}(\mathbf{x})d\mathbf{w} \end{aligned}$$

By considering $d\mathbf{w}d\mathbf{w}^T \to Idt$ we will have the equation that follows:

$$d\Psi(\mathbf{x},t) = \partial_t\Psi\ dt + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{f}(\mathbf{x},t)dt + \frac{1}{2}tr\Big(\,(\nabla_{\mathbf{xx}}\Psi)\,\mathbf{B}\mathbf{B}^T\Big)dt + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{B}(\mathbf{x})d\mathbf{w} \tag{3.61}$$

Since we have found the total differential in $\Psi(\mathbf{x}, t)$ we can substitute back to (3.59) and we get the following expression:

$$d\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t) = \boldsymbol{\mathcal{Z}}(t, t_N)\ dt\Bigg(-\frac{1}{\lambda}Q(\mathbf{x})\Psi + \partial_t\Psi + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{f}(\mathbf{x}, t) + \frac{1}{2}tr\Big((\nabla_{\mathbf{xx}}\Psi)\,\mathbf{B}\mathbf{B}^T\Big)\Bigg)$$
$$+\ \boldsymbol{\mathcal{Z}}(t_0, t)\,(\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{B}(\mathbf{x})\mathbf{L}d\mathbf{w}$$

According to the backward Kolmogorov PDE (3.53) the term inside the parenthesis equals $\boldsymbol{\mathcal{F}}(\mathbf{x}, t)$ and therefore the equation above is formulated as:

$$d\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t) = \boldsymbol{\mathcal{Z}}(t_0, t)\left(-\boldsymbol{\mathcal{F}}(\mathbf{x}, t)dt + (\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{B}(\mathbf{x})\mathbf{L}d\mathbf{w}\right)$$

With the definition of τ_p as $\tau_p \triangleq \{s \leq t; ||\mathbf{x}|| > p\}$ we integrate the equation above in the time interval $t \in [t_0, t_N \wedge \tau_n]$ and we will have then following expression:

$$\int_{t_0}^{t_N\wedge\tau_p} d\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t) = -\int_{t_0}^{t_N\wedge\tau_p} \boldsymbol{\mathcal{F}}(\mathbf{x}, t)\boldsymbol{\mathcal{Z}}(t_0, t)dt + \int_{t_0}^{t_N\wedge\tau_p} \boldsymbol{\mathcal{Z}}(t_0, t)\,(\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{C}(\mathbf{x})\mathbf{L}d\mathbf{w} \tag{3.62}$$

Expectation of the equation above is taken over the sampled paths $\boldsymbol{\tau} = (\mathbf{x}_0, ..., \mathbf{x}_{t_N})$ starting from the state $\mathbf{x}_0$. The resulting equation is expressed as follows:

$$\left\langle \int_{t_0}^{t_N\wedge\tau_p} d\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t) \right\rangle =$$
$$= \left\langle -\int_{t_0}^{t_N\wedge\tau_p} \boldsymbol{\mathcal{F}}(\mathbf{x}, t)\boldsymbol{\mathcal{Z}}(t_0, t)dt + \int_{t_0}^{t_N\wedge\tau_p} \boldsymbol{\mathcal{Z}}(t_0, t)\,(\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{C}(\mathbf{x})\mathbf{L}d\mathbf{w} \right\rangle$$

We change the order to the time integration and expectation and due to the fact that $E(d\mathbf{w}) = 0$ the last term of the right side of the equation above drops. Consequently we will have:

$$\left\langle \int_{t_0}^{t_N \wedge \tau_p} d\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t) \right\rangle = -\left\langle \int_{t_0}^{t_N \wedge \tau_p} \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \boldsymbol{\mathcal{Z}}(t, t_N) dt \right\rangle \tag{3.63}$$

The left hand side of the equation above is further written as:

$$\left\langle \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_N) \mathbf{1}_{\tau_p > t_N} + \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, \tau_p) \mathbf{1}_{\tau_p < t_N} - \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_0) \right\rangle = -\left\langle \int_{t_0}^{t_N \wedge \tau_p} \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \boldsymbol{\mathcal{Z}}(t_0, t) dt \right\rangle \tag{3.64}$$

or

$$\left\langle \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_0) \right\rangle = \left\langle \int_{t_0}^{t_N \wedge \tau_p} \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \boldsymbol{\mathcal{Z}}(t, t_N) dt + \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_N) \mathbf{1}_{\tau_p > t_N} + \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, \tau_p) \mathbf{1}_{\tau_p < t_N} \right\rangle \tag{3.65}$$

Since $\boldsymbol{\mathcal{G}}(\mathbf{x}, t, t_0) = \Psi(\mathbf{x}, t) \boldsymbol{\mathcal{Z}}(t_0, t)$ and $\boldsymbol{\mathcal{Z}}(t_0, t) = \exp\left(-\frac{1}{\lambda} \int_{t_0}^{t} Q(\mathbf{x}) d\tau\right)$ all the terms $\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_N), \boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_0)$ and $\boldsymbol{\mathcal{G}}(\mathbf{x}, \tau_p, t_N)$ are further specified by the equations that follow:

$$\begin{aligned}
\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_0) &= \Psi(\mathbf{x}, t_0) \\
\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, t_N) &= \Psi(\mathbf{x}, t) \exp\left(-\frac{1}{\lambda} \int_{t_0}^{t_N} Q(\mathbf{x}) d\tau\right) \\
\boldsymbol{\mathcal{G}}(\mathbf{x}, t_0, \tau_n) &= \Psi(\mathbf{x}, \tau_p) \exp\left(-\frac{1}{\lambda} \int_{t_0}^{\tau_p} Q(\mathbf{x}) d\tau\right)
\end{aligned}$$

Substituting the equations above to (3.65) results in:

$$\begin{aligned}\Psi(\mathbf{x},t_0)=\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t_0) &= \left\langle \int_{t_0}^{t_N\wedge\tau_p}\boldsymbol{\mathcal{F}}(\mathbf{x},t)\exp\left(-\frac{1}{\lambda}\int_{t_0}^{t_N}Q(\mathbf{x})d\tau\right)\right\rangle \\ &+ \left\langle \Psi(\mathbf{x},\tau_p)\exp\left(-\frac{1}{\lambda}\int_{t_0}^{\tau_p}Q(\mathbf{x})d\tau\right)\mathbf{1}_{\tau_p\le t_N}\right\rangle \\ &+ \left\langle \Psi(\mathbf{x},t_N)\exp\left(-\frac{1}{\lambda}\int_{t_0}^{t_N}Q(\mathbf{x})d\tau\right)\mathbf{1}_{\tau_p> t_N}\right\rangle \qquad (3.66)\end{aligned}$$

The next step in the derivation is to find the limit of the right hand side of the equation above as $p\to\infty$. More precisely either by using (iii) in (3.51)and the dominated convergence theorem or by considering the monotone convergence theorem (see section 3.11) and (iv) in (3.51) the limit of the first term in (3.66) equals:

$$\lim_{p\to\infty}\left\langle \int_{t_0}^{t_N\wedge\tau_p}\boldsymbol{\mathcal{F}}(\mathbf{x},t)\exp\left(-\frac{1}{\lambda}\int_{t_0}^{t_N}Q(\mathbf{x})d\tau\right)\right\rangle = \left\langle \int_{t_0}^{t_N}\boldsymbol{\mathcal{F}}(\mathbf{x},t)\exp\left(-\frac{1}{\lambda}\int_{t_0}^{t_N}Q(\mathbf{x})d\tau\right)\right\rangle$$

The second term in (3.66) is bounded as: $\left\langle |\Psi(\mathbf{x},t)|\mathbf{1}_{\tau_p\le T}\right\rangle \le M\left(1+p^{2\mu}\right)P\left(\tau_p\le T\right)$ where the probability $P\left(\tau_p\le T\right)$ is expressed as follows:

$$\begin{aligned}P\left(\tau_p\le T\right)=P\left(\max_{t\le\theta\le T}||\mathbf{x}_\theta||\ge p\right) &\le p^{2m}\left\langle \max_{t\le\theta\le T}||\mathbf{x}_\theta||^{2m}\right\rangle \\ &\le Cp^{-2m}\left(1+||\mathbf{x}||^{2m}\right)\end{aligned}$$

where the first inequality results from the chebyshev inequality and the second inequality comes from the property $\left\langle \max_{t\le\theta\le s}||\mathbf{x}_\theta||^{2m}\right\rangle \le C\left(1+||\mathbf{x}||^{2m}\right)e^{C(T-s)}$ where

$t \leq s \leq T$. Clearly as $p \to \infty$ we have $\left\langle |\Psi(\mathbf{x},t)| \mathbf{1}_{\tau_p \leq T} \right\rangle \leq M\left(1+p^{2\mu}\right) P\left(\tau_n \leq T\right) \to 0$. Thus:

$$\lim_{p\to\infty} \left\langle \Psi(\mathbf{x},\tau_p) \exp\left(-\frac{1}{\lambda}\int_{t_0}^{\tau_p} Q(\mathbf{x}) d\tau\right) \mathbf{1}_{\tau_p \leq t_N} \right\rangle = 0$$

Finally the third term converges to

$$\left\langle \Psi(\mathbf{x},t_N) \exp\left(-\frac{1}{\lambda}\int_{t_0}^{t_N} Q(\mathbf{x}) d\tau\right) \right\rangle$$

The final result of the Feynman Kac lemma is given by the equation that follows:

$$\boxed{\Psi(\mathbf{x},t) = \left\langle \xi(\mathbf{x}_T) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}_s,s) ds\right) + \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta,\theta) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x},s) ds\right) d\theta \right\rangle}$$

with $\Psi(\mathbf{x},t_N) = \xi(\mathbf{x}_T)$. This is the end of the proof of the Feynman-Kac lemma.

Since the Feynman-Kac lemma requires the condition of the linear growth of the elements of the drift term $\mathbf{f}(\mathbf{x},t)$ and the diffusion matrix $\mathbf{B}(\mathbf{x},t)$ in (3.9) one could ask what kind of dynamical systems fulfill these conditions. But before we discuss the generality of the applicability of the Feynman-Kac lemma to a variety of dynamical systems in control and planning application it is critical to identify the conditions under which a solution to the Cauchy problems exist.

The conditions that guarantee the existence of the solutions as they are reported in (Karatzas & Shreve 1991) and proven in (Friedman 1975) are given bellow:

i) *Uniform Ellipticity:* There exist as positive constant δ such that:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{i,j}(\mathbf{x},t)\xi_i\xi_j \geq \delta||\xi||^2 \tag{3.67}$$

holds for every $\xi \in \Re^{n\times 1}$ and $(t,\mathbf{x}) \in [0,\infty) \times \Re^{n\times 1}$.

ii) *Boundness:* The functions $\mathbf{f}(\mathbf{x},t), q(\mathbf{x},t), \alpha(\mathbf{x},t)$ are bounded in $[0,T] \times \Re^{n\times 1}$.

iii) *Holder Continuity:* The functions $\mathbf{f}(\mathbf{x},t), q(\mathbf{x},t), \alpha(\mathbf{x},t)$ and $\boldsymbol{\mathcal{F}}(\mathbf{x},t)$ are uniformly Holder continuous in $[0,T] \times \Re^{n\times 1}$.

iv) *Polynomial Growth:* the functions $\Psi(\mathbf{x}(t_N)) = \xi(\mathbf{x}(t_N))$ and $\boldsymbol{\mathcal{F}}(\mathbf{x},t)$ satisfy the (i) and (iii) in (3.51)

Conditions (i),(ii) and (iii) can be relaxed by assuming that they are locally true.

Essentially, the Feynman- Kac lemma provides solution of the PDE (3.53) in a probabilistic manner, if that solution exists, and it also tells us that this solution is unique. The conditions above are sufficient conditions for the existence of the solution of (3.53).

With the goal to apply the Feynman- Kac lemma to learning approximate solution for optimal planning and control problems, it is important to understand how the conditions of this lemma are related to properties and characteristics of the dynamical systems under consideration.

- The condition of linear growth, for the case of control, is translated as the requirement to deal with dynamical systems in which their vector field as a function of state is bounded either by a linear function or by number. Therefore the Feynman Kac lemma can be applied either to linear dynamical systems of the form

$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ or nonlinear dynamical systems of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u}$ in which $\mathbf{f}_i(\mathbf{x}) < M||\mathbf{x}||$. Examples of nonlinear functions that satisfy the linear growth condition are functions such as $\cos(\mathbf{x}), \sin(\mathbf{x})$. The dynamical systems under consideration can be stable or unstable, for as long as their vector field satisfies the linear growth condition then they "qualify" for the application of the Feynman-Kac lemma.

- But what happens for the case of dynamical systems in which the vector field $\mathbf{f}(\mathbf{x})$ cannot be bounded $\forall \mathbf{x} \in \Re^n$ such as for example the function $\mathbf{f}(\mathbf{x}) = \mathbf{x}^2$? The answer to this question is related to the locality condition. In particular if we know that the dynamical system under consideration operates in a pre-specified region of the state space then an upper bound for the vector field can almost always be found. Consequently the conditions of boundedness in combination with the relaxed condition of locality are important for the application of Feynman-Kac lemma to a rather general class of systems.

- Finally, our view in applying the Feynman Kac lemma and the path integral control formalism is for systems in which an initial set of state space trajectories is given or generated after an initial control policy has been applied. Thus these systems are initially controlled and clearly their vector field cannot be unbounded as a function of the state.

We will continue this discussion of the application of Feynman - Lemma for optimal control and planning for the chapter of path integral control formalism. In the next section we will try to identify the most important special case of the Feynman Kac lemma.

3.6 Special cases of the Feynman Kac lemma.

There are many special cases of the Feynman Kac lemma in the literature (Øksendal 2003),(Friedman 1975),(Karatzas & Shreve 1991),(Fleming & Soner 2006) which,at a first glance, might look confusing and different. Nevertheless, under the generalized version of the Feynman Kac lemma it is easy to recover and recognize all these special cases. We start with the case where there is no discount cost which is equivalent to $q(\mathbf{x}) = 0$. The backward Kolmogorov PDE, then is formulated as:

$$-\partial_t \Psi_t = \mathbf{f}_t^T(\nabla_{\mathbf{x}} \Psi_t) + \frac{1}{2} tr\Big((\nabla_{\mathbf{xx}} \Psi_t)\mathbf{B}\mathbf{B}^T\Big) + \boldsymbol{\mathcal{F}}(\mathbf{x}, t); \quad in \quad [0, T) \times \Re^{n\times 1} \tag{3.68}$$

with the Feynman - Kac representation:

$$\Psi(\mathbf{x}, t) = \left\langle \xi(\mathbf{x}_T) + \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) d\theta \right\rangle \tag{3.69}$$

and $\Psi(\mathbf{x}, t_N) = \xi(\mathbf{x}_T)$. If the forcing term $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) = 0 \ \forall \mathbf{x} \in \Re^{n\times 1}, \ t \in [0, T)$ then it drops from (3.68) while the Feynman- Kac representation of the resulting PDE is given by $\Psi(\mathbf{x}, t) = E\left(\xi(\mathbf{x}_T)\right)$. Moreover if the drift term $\mathbf{f}(\mathbf{x}, t) = 0 \ \forall \mathbf{x} \in \Re^{n\times 1}, \ t \in [0, T)$ the backward Kolmogorov PDE (3.68) collapses to:

$$-\partial_t \Psi_t = \frac{1}{2} tr\Big((\nabla_{\mathbf{xx}} \Psi_t)\mathbf{B}\mathbf{B}^T\Big); \quad in \quad [0, T) \times \Re^{n\times 1} \tag{3.70}$$

For the case where $\mathbf{B}(\mathbf{x}, t) = \mathbf{B}$ the difference between the backward Kolmogorov and forward Kolmogorov PDEs, for this special case of (3.70), is only the sign of the partial

derivative of Ψ with respect to time. To see that we just need to apply the transformation $\Psi(\mathbf{x},t) = \Phi(\mathbf{x},T-t) = \Phi(\mathbf{x},\tau)$ and thus we will have that $\partial_t \Psi_t = -\partial_\tau \Phi_\tau$. The backward kolmogorov PDE is now transformed to a the forward PDE:

$$\partial_\tau \Phi_\tau = \frac{1}{2} tr\Big((\nabla_{\mathbf{xx}} \Psi_t)\mathbf{B}\mathbf{B}^T\Big); \quad in \quad [0,T) \times \Re^{n\times 1} \tag{3.71}$$

The PDE above is the forward Kolmogorov PDE which corresponds to SDEs without the drift term and only diffusion term. In the most general cases, the transformation $\Psi(\mathbf{x},t) = \Phi(\mathbf{x},T-t) = \Phi(\mathbf{x},\tau)$ of the backward Kolmogorov PDE results in a forward PDE which does nor always correspond to the forward Kolmogorov PDE. In fact, this is true only in the case $\boldsymbol{\mathcal{F}}(\mathbf{x},t) = 0, q(\mathbf{x}) = 0$ and $\mathbf{f}(\mathbf{x},t) = 0\ \forall \mathbf{x} \in \Re^{n\times 1},\ t \in [0,T)$ and constant diffusion matrix $\mathbf{B}$. For the most general case the transformation $\Psi(\mathbf{x},t) = \Phi(\mathbf{x},T-t) = \Phi(\mathbf{x},\tau)$ results in the PDEs given by the equation that follows:

$$\partial_\tau \Phi_\tau^{(i)} = -\frac{1}{\lambda} q(\mathbf{x},T-\tau)\Phi_\tau^{(i)} + \mathbf{f}_\tau^{(i)T}(\nabla_{\mathbf{x}} \Phi_\tau^{(i)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Phi_\tau^{(i)})\mathbf{B}\mathbf{B}^T\right) + \boldsymbol{\mathcal{F}}(\mathbf{x},T-\tau) \tag{3.72}$$

with the initial condition $\Phi(\mathbf{x},0) = \exp(-\frac{1}{\lambda}\phi(t_N))$. By substituting $\tilde{q}(\mathbf{x},\tau) = q(\mathbf{x},T-\tau)$ and $\tilde{\boldsymbol{\mathcal{F}}}(\mathbf{x},\tau) = \boldsymbol{\mathcal{F}}(\mathbf{x},T-\tau)$ the Feynman Kac representation takes the form:

$$\Phi(\mathbf{x},\tau) \quad = \quad \left\langle \xi(\mathbf{x}_0) \exp\left(-\frac{1}{\lambda}\int_0^\tau \tilde{q}(\mathbf{x},s)ds\right) + \int_0^\tau \tilde{\boldsymbol{\mathcal{F}}}(\mathbf{x},\theta) \exp\left(-\frac{1}{\lambda}\int_t^\tau \tilde{q}(\mathbf{x},s)ds\right) d\theta \right\rangle$$

The forward PDE in (3.72) and its probabilistic solution above is another form of the Feynman- Kac lemma. In the next section we show how the backward and forward PDEs are related for the most general case.

3.7 Backward and forward Kolmogorov PDE and their fundamental solutions

After discussing the solution to the Cauchy problem and presenting special cases of the Feynman-Kac lemma, in this section we investigate the connection between the forward and backward Kolmogorov PDEs. The backward Kolmogorov PDE, as we will show in the next chapter, appears under certain conditions in a general optimal control problem while the forward Kolmogorov PDE is of great importance in nonlinear stochastic estimation. It is therefore of great importance to understand their connection in a mathematical as well as an intuitive level.

Towards this goal, we will start our analysis with the definition of the *fundamental solution* (Karatzas & Shreve 1991) of a second order PDE.

Definition: *Let consider the nonnegative function* $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ *with* $0 < t < \tau$, $\mathbf{x},\mathbf{y} \in \Re^n$ *and* $\xi \in \mathbf{C}$ *and* $\tau \in [0,T]$. *The function* $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ *is a fundamental solution of the PDE:*

$$-\partial_t \Psi_t = -\frac{1}{\lambda} q_t \Psi_t + \mathbf{f}(\mathbf{x}_t)^T (\nabla_{\mathbf{x}} \Psi_t) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t) \mathbf{B}(\mathbf{x},t) \mathbf{B}(\mathbf{x},t)^T \right) \tag{3.73}$$

if the function $\Psi(\mathbf{y},t)=\int_{\Re^n}\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)\xi(\mathbf{x})d\mathbf{x}$*, satisfies the PDE above and* $\lim_{t\to\tau^-}\Psi(\mathbf{y},t)=\xi(\mathbf{y},\tau)$.

Before we proceed with the theorem which establishes the connection between the forward and backward Kolmogorov PDE through the concept of *fundamental solution*, lets understand the "physical" meaning of the function $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ for $0<t<\tau,\ \mathbf{x},\mathbf{y}\in\Re^n$. Let us assume for the moment that $q(\mathbf{x})=0$ then through the Feynamn Kac lemma of the solution of the PDE is represented as $\Psi(\mathbf{x},t)=E\left(\xi(\mathbf{x}_T)\right)$. Inspection of the last equation and $\Psi(\mathbf{y},t)=\int_{\Re^n}\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)\xi(\mathbf{x})d\mathbf{x}$ tell us that any fundamental solution of the backward Kolmogotov PDE can be thought as an transition probability of the stochastic process $\mathbf{x}$ which evolves according to the stochastic differential equation (3.9). Consequently, we can write that $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)=p(\mathbf{x},\tau|\mathbf{y},t)$. Another property of the function of *fundamental solution* of a second order PDE comes from the fact that:

$$\lim_{t\to\tau^-}\Psi(\mathbf{y},t)=\lim_{t\to\tau^-}\int\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)\xi(\mathbf{x})d\mathbf{x}=\xi(\mathbf{y},\tau)\tag{3.74}$$

From the equation above, it is easy to see that the *fundamental solution* has the property that $\lim_{t\to\tau^-}\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)=\delta(\mathbf{y}-\mathbf{x})$ where $\delta(\mathbf{x})$ is the *Dirac* function. Clearly, since there is a probabilistic interpretation of $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ the transition probability $p(\mathbf{x},\tau|\mathbf{y},t)$ inherits the same property and therefore $p(\mathbf{x},\tau|\mathbf{y},t)=\delta(\mathbf{y}-\mathbf{x})$ for $t=\tau$.

Now, we present the theorem (Karatzas & Shreve 1991),(Friedman 1975) that establishes the connection between the forward and backward Kolmogorov PDE, trough the concept of *fundamental solution*.

Theorem: *Under the conditions of Uniform Ellipticity of* $\alpha(\mathbf{x},t)$*, Holder continuity and boundeness of* $\mathbf{f}(\mathbf{x},t), \boldsymbol{\mathcal{F}}(\mathbf{x},t), \alpha(\mathbf{x},t)$ *a fundamental solution of 3.73 exist. Furthermore for any fixed* $\tau, \mathbf{x}$ *the function* $\psi(\mathbf{y},t) = \boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ *satisfies the backward Kolmogorov PDE. In addition if the function* $(\partial/\partial x_i)\mathbf{f}(\mathbf{x},t)$*,* $(\partial/\partial x_i)\alpha(\mathbf{x}_{ik})$ *and* $(\partial^2/\partial x_i^2)\alpha(\mathbf{x}_{ik})$ *are bounded and Holder continuous the for fixed* $t, \mathbf{x}$ *the function* $\psi(\mathbf{y},\tau) = \boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ *satisfies the forward Kolmogorov equation:*

$$\partial_\tau \psi(\mathbf{y},\tau) = -\sum_{i=1}^{n} \frac{\partial}{\partial y_i}\Big(\mathbf{f}_i(\mathbf{y},\tau)\psi(\mathbf{y},\tau)\Big) + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial}{\partial y_i \partial y_j}\Big(\alpha_{i,j}(\mathbf{y},\tau)\psi(\mathbf{y},\tau)\Big) - q(\mathbf{y},t)\psi(\mathbf{y},\tau) \tag{3.75}$$

The proof of the theorem can be found in (Friedman 1975). Clearly, the *fundamental solution* $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ establishes a connection between the forward and backward Kolmogorov PDE. Essentially, if $\boldsymbol{\mathcal{D}}(\mathbf{y},t;\mathbf{x},\tau)$ is considered as a function of $\mathbf{x},t$ then it satisfies the former, while when it is thought as a function of $\mathbf{y},\tau$ then it satisfies the later.

To better understand this connection, lets study the example of the diffusion $dx = \mu dt + \sigma d\omega$. From the analysis above we know that the transition probability of this diffusion is a fundamental solution. Lets verify if this statement true. More precisely, the aforementioned diffusion can been written in the form $x(t+dt) - x(t) = \mu dt + \sigma d\omega$. By substituting with $x(t+dt) = x(\tau)$ for $\boldsymbol{\tau} = t + dt$ and $x(t) = y(t)$, we will have $x(\tau) - y(t) = \mu(\tau - t) + \sigma d\omega$. With this new form, the transition probability is expressed by the equation that follows:

$$p(x,\tau|y,t) = \boldsymbol{\mathcal{D}}(y,t;x,\tau) = \frac{1}{\sqrt{2\pi\sigma^2(t-\tau)}} \exp\left(- \frac{(x-y-\mu\ (\tau-t))^2}{2\sigma^2(t-\boldsymbol{\tau})}\right) \tag{3.76}$$

The backward and forward Kolmogorov PDEs for the stochastic diffusions $dx = \mu dt + \sigma d\omega$ are formulated as follows:

$$-\frac{\partial}{\partial t}p(x,\tau|y,t) = \mu\ \frac{\partial}{\partial y}p(x,\tau|y,t) + \frac{1}{2}\sigma^2\frac{\partial^2}{\partial y^2}p(x,\tau|y,t)$$

$$\frac{\partial}{\partial \tau}p(x,\tau|y,t) = -\mu\ \frac{\partial}{\partial x}p(x,\tau|y,t) + \frac{1}{2}\sigma^2\frac{\partial^2}{\partial x^2}p(x,\tau|y,t)$$

The verify the theorem of the *fundamental solution* we compute the following terms:

$$\begin{aligned}
\frac{\partial}{\partial y}p(x,\tau|y,t) &= \left(\frac{x-y-\mu\ (\tau-t)}{\sigma^2(\tau-t)}\right)p(x,\tau|y,t)\\
\frac{\partial^2}{\partial y^2}p(x,\tau|y,t) &= \frac{-1}{\sigma^2(\tau-t)}\left(1+\frac{x-y-\mu\ (\tau-t)}{\sigma^2(\tau-t)}\right)p(x,\tau|y,t)\\
\frac{\partial}{\partial x}p(x,\tau|y,t) &= -\frac{\partial}{\partial y}p(x,\tau|y,t)\\
\frac{\partial^2}{\partial y^2}p(x,\tau|y,t) &= \frac{1}{\sigma^2(\tau-t)}\left(-1+\frac{x-y-\mu\ (\tau-t)}{\sigma^2(\tau-t)}\right)p(x,\tau|y,t)
\end{aligned}$$

In addition to the partial derivative with respect to the state x and y, the time derivatives of the transition probability are formulated as follows:

$$\frac{\partial}{\partial \tau}p(x,\tau|y,t) = \left(-\frac{1}{\tau-t} + \frac{x-y-\mu(\tau-t)}{\sigma^2(\tau-t)} + \frac{(x-y-\mu(\tau-t))^2}{\sigma^2(\tau-t)^2}\right)p(x,\tau|y,t)$$

$$\frac{\partial}{\partial t}p(x,\tau|y,t)=\left(\frac{1}{\tau-t}-\frac{x-y-\mu(\tau-t)}{\sigma^2(\tau-t)}-\frac{(x-y-\mu(\tau-t))^2}{\sigma^2(\tau-t)^2}\right)p(x,\tau|y,t)$$

By computing the terms in the left sides of the PDEs and the time derivatives $\frac{\partial}{\partial t}p(x,\tau|y,t)$ and $\frac{\partial}{\partial \tau}p(x,\tau|y,t)$ it is easy to show that, indeed, $p(x,\tau|y,t)$ satisfies the backward Kolmogorov in $\mathbf{y},\tau$ and the forward Kolmogorov in $\mathbf{x},t$.

3.8 Connection of backward and forward Kolmogorov PDE via the Feynman Kac lemma

The connection between the backward and the forward Kolmogorov PDEs can be also seen in the derivation of the Feynman Kac lemma. Towards an understanding in depth of the connection between the two PDEs, our goal in this section is to show that in the derivation of the Feynman Kac lemma both PDEs are involved. In particular, we are assuming that the backward Kolmogorov PDE holds, and while we are trying to find its solution, the forward PDE appears from our mathematical manipulations. More precisely, we will start our derivation from equation (3.61) in the Feynman Kac lemma but we will assume for simplicity that $q(\mathbf{x},t)=0$. More precisely we will have that:

$$d\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t)=dt\left(\partial_t\Psi+(\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{f}(\mathbf{x},t)+\frac{1}{2}tr\left(\nabla_{\mathbf{xx}}\Psi\mathbf{B}\mathbf{B}^T\right)\right)+(\nabla_{\mathbf{x}}\Psi)^T\,\mathbf{B}(\mathbf{x})\mathbf{L}d\mathbf{w}$$

By integrating and taking the expectation of the equation above and since $\left\langle d\mathbf{w}\right\rangle=0$:

$$\left\langle d\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t)\right\rangle = \left\langle \int \left(\partial_t \Psi + (\nabla_{\mathbf{x}}\Psi)^T \mathbf{f}(\mathbf{x},t) + \frac{1}{2} tr\left(\nabla_{\mathbf{xx}}\Psi \mathbf{B}\mathbf{B}^T\right) \right) dt \right\rangle$$

The expectation above is taken with respect to the transition probability $p(\mathbf{x},t|\mathbf{x}_0,t_0)$ defined based on the Itô diffusion (3.9). Consequently we will have:

$$\left\langle d\boldsymbol{\mathcal{G}}(\mathbf{x},t_0,t)\right\rangle = \int\int p(\mathbf{x},t|\mathbf{x},t_0) \left(\partial_t \Psi + (\nabla_{\mathbf{x}}\Psi)^T \mathbf{f}(\mathbf{x},t) + \frac{1}{2} tr\left(\nabla_{\mathbf{xx}}\Psi \mathbf{B}\mathbf{B}^T\right) \right) dt\mathbf{dx}$$

We are skipping few of the steps that we followed in the Feynman- Kac lemma, however it is easy to show that the equation above can be written in the form:

$$\begin{aligned}
&\left\langle \Psi(\mathbf{x}(t_N)\right\rangle - \Psi(\mathbf{x}(t_0)) = \\
&\qquad = \int\int p(\mathbf{x},t|\mathbf{x}_0,t_0) \left(\partial_t \Psi + (\nabla_{\mathbf{x}}\Psi)^T \mathbf{f}(\mathbf{x},t) + \frac{1}{2} tr\left(\nabla_{\mathbf{xx}}\Psi \mathbf{B}\mathbf{B}^T\right) \right) dt\mathbf{dx}
\end{aligned}$$

we integrate by parts and therefore:

$$\begin{aligned}
&\int_{\Re^n} p(\mathbf{x},t_N|\mathbf{x}_0,t_0)\Psi(\mathbf{x}(t_N))\mathbf{dx} - \Psi(\mathbf{x}(t_0)) = \\
&\int_{\Re^n}\int \Psi \left(-\partial_t p(\mathbf{x},t|\mathbf{x}_0,t_0) + \nabla_{\mathbf{x}}\left(\mathbf{f}(\mathbf{x},t)p(\mathbf{x},t|\mathbf{x}_0,t_0)\right) \right) \\
&+ \int_{\Re^n}\int \Psi \left(\frac{1}{2} tr\left(\nabla_{\mathbf{xx}} p(\mathbf{x},t|\mathbf{x}_0,t_0)\mathbf{B}\mathbf{B}^T\right) \right) dt\mathbf{dx} \\
&+ \int_{\Re^n} p(\mathbf{x},t|\mathbf{x}_0,t_0)\Psi(\mathbf{x},t)\mathbf{dx}|_{t=t_0}^{t=t_N}
\end{aligned}$$

The last term $\int_{\Re^n} p(\mathbf{x},t|\mathbf{x}_0,t_0)\Psi(\mathbf{x},t)\mathbf{dx}|_{t=t_0}^{t=t_N}$ is further written as:

$$\int_{\Re^n} p(\mathbf{x},t_N|\mathbf{x}_0,t_0)\Psi(\mathbf{x},t)\mathbf{dx} - \int_{\Re^n} p(\mathbf{x},t_0|\mathbf{x}_0,t_0)\Psi(\mathbf{x},t)\mathbf{dx}$$

From the equation above we conclude the following:

$$\Psi(\mathbf{x}(t_0)) = \int_{\Re^n} p(\mathbf{x},t_0|\mathbf{x}_0,t_0)\Psi(\mathbf{x},t_0)\mathbf{dx} \tag{3.77}$$

and also

$$-\partial_t p(\mathbf{x},t|\mathbf{x}_0,t_0) + \nabla_{\mathbf{x}}\left(\mathbf{f}(\mathbf{x},t)p(\mathbf{x},t|\mathbf{x}_0,t_0)\right) + \frac{1}{2}tr\left(\nabla_{\mathbf{xx}}p(\mathbf{x},t|\mathbf{x}_0,t_0)\mathbf{B}\mathbf{B}^T\right) = 0 \tag{3.78}$$

The first equation tells us that the transition probability $p(\mathbf{x},t|\mathbf{x}_0,t_0)$ acts as a *Dirac* function since $\lim_{t\to t_0^+} p(\mathbf{x},t|\mathbf{x}_0,t_0) = \delta(\mathbf{x}-\mathbf{x}_0)$. We have arrived at the same conclusion, regarding the transition probability, with the result in the previous section where we showed that, in fact this is a general property of the *fundamental solutions* of PDEs and therefore since the transition $\lim_{t\to t_0^+} p(\mathbf{x},t|\mathbf{x}_0,t_0)$ is a *fundamental solutions*, it inherits the same property. Clearly in this section we do not use the theory of *fundamental solutions* but we find the same result by slightly changing the derivation of the Feynman-Kac lemma. The second equation is nothing else than the forward Kolmogorov PDE and it tells us that that the transition probability $p(\mathbf{x},t|\mathbf{x}_0,t_0)$ satisfies the forward Kolmogorov PDE in $\mathbf{x}$ and t . Essentially the derivation of the Feynman- Kac lemma can be used to 1) find the probabilistic interpretation of the solution of the backward kolmogorov equation and thus provide a solution to the Cauchy problem. 2) Shown that the transition

probability acts as a *dirac* function thus it shares the same property with the fundamental solution of the PDEs and 3) Prove that the forward kolmogorov can be thought as an outcome of the Feynman-Kac lemma and thus to offer another perceptual view of the connection between the forward and backward Kolmogorov PDEs.

The discussion so far seems a bit abstract. So one could ask why all these? What do really this PDEs represent? Where do we find them in engineering?

3.9 Forward and backward Kolmogorov PDEs in estimation and control

We will close this chapter on the connection between the forward and backward Kolmogorov PDEs with the discussion on how these PDEs appear in nonlinear control and estimation problems. We start our analysis with the Zakai equation which is found in nonlinear estimation theory. More precisely we consider the nonlinear filtering problem in which the stochastic dynamics are expressed by the equation:

$$\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)d\mathbf{w}$$

while the observations are given by the diffusion:

$$\mathbf{dy} = \mathbf{h}(\mathbf{x}, t)dt + d\mathbf{v}$$

The goal is to estimate the state of the stochastic dynamics which is equivalent of finding the probability density $p(\mathbf{x}, t)$ of the state $\mathbf{x}$ at time t. This probability density satisfies the Zakai equation:

$$\partial p = -\sum_{i=0}^{n} \frac{\partial}{\partial x_i}\Big(\mathbf{f}(\mathbf{x},t)p\Big)dt + \frac{1}{2}\sum_{k=1}^{m}\sum_{i,j=1}^{n}\Big(\mathbf{B}_{i,k}(\mathbf{x},t)\mathbf{B}_{k,j}(\mathbf{x},t)^T p\Big)dt + p\mathbf{h}(\mathbf{x},t)^T\mathbf{dy}$$

The PDE above is linear, second order and stochastic. The stochasticity is incorporated due the the last term which is a function of the observations $d\mathbf{y}$. Substitution of the observation model to the PDE above, results in the following linear stochastic PDE:

$$\partial p = -\sum_{i=0}^{n} \frac{\partial}{\partial x_i}\Big(\mathbf{f}(\mathbf{x},t)p\Big)dt + \frac{1}{2}\sum_{k=1}^{m}\sum_{i,j=1}^{n}\Big(\mathbf{B}_{i,k}(\mathbf{x},t)\mathbf{B}_{k,j}(\mathbf{x},t)^T p\Big)dt + p\mathbf{h}(\mathbf{x},t)^T\mathbf{h}(\mathbf{x},t)dt$$
$$+ p\mathbf{h}(\mathbf{x},t)^T d\mathbf{v}$$

From the equation above we can see that for $\mathbf{h}(\mathbf{x}, t) = 0$ the forward Zakai equation collapses to a forward Chapman- Kolmogorov PDE. As it will be shown in the next chapter, the backward chapman Kolmogorov PDE appears in optimal control and it has the form:

$$-\partial_t \Psi_t = -\frac{1}{\lambda} q_t \Psi_t + \mathbf{f}_t^T(\nabla_{\mathbf{x}}\Psi_t) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}}\Psi_t)\mathbf{B}\mathbf{B}^T\right)$$

With respect to the forward Zakai PDE, the backward Chapman Kolmogorov is also linear but deterministic. This last difference is one of the main reasons why the duality between optimal linear filtering and linear control was not generalized for the nonlinear

case. Recently (Todorov 2008) , the generalized duality was exploited when the backward Zakai equation in nonlinear smoothing is considered. Essentially the backward Zakai equation can be turned into a deterministic PDE and then a direct mapping between the two PDEs can be made in the same way how it is made between the backward and forward Riccati equations in linear control and filtering problems.

3.10 Conclusions

In this chapter we investigated the connection between SDEs, linear PDEs and Path Integrals. My goal was to give an introduction to these mathematical concepts and their connections by keeping a balance between a pedagogical and intuitive presentation and a presentation that is characterized by rigor and mathematical precision.

In the next chapter, the path integral formalism is applied to stochastic optimal control and reinforcement learning and the generalized path integral control is derived. More precisely, the backward Chapman Kolmogorov is formulated and the Feynman-Kac lemma is applied. Finally the path integral control is derived. Extensions of path integral control to iterative and risk sensitive control are presented.

3.11 Appendix

We assume the stochastic differential equation $\mathbf{dx} = \mathbf{f}(\mathbf{x},t)dt + \mathbf{B}(\mathbf{x},t)d\mathbf{w}$. *If the drift* $\mathbf{f}(\mathbf{x},t)$ *and diffusion term* $\mathbf{B}(\mathbf{x},t)$ *satisfy the condition:*$||\mathbf{f}(\mathbf{y},t)||^2 + ||\mathbf{B}(\mathbf{y},t)||^2 < K\left(1+max||\mathbf{y}(s)||^2\right)$ then $\left\langle \max_{0<s<t} ||\mathbf{x}_s||^{2m} \right\rangle \leq C\left(1+\left\langle ||\mathbf{x}_o||^{2m} \right\rangle\right)e^{Ct}, \quad \forall 0 \leq t \leq T.$

Hölder Continuity Definition: *A function* $f(\mathbf{x}) : \Re^n \to \Re$ *is Hölder continuous is there is* $\alpha > 0$ *such that* $|f(\mathbf{x}) - f(\mathbf{x})| \leq |\mathbf{x} - \mathbf{y}|^\alpha$.

Monotone Convergence Theorem: *if* f_n *is a sequence of measurable function with* $0 \leq f_n \leq f_{n+1}, \quad \forall n$ *then* $\lim_{n\to\infty} \int f_n d\mu = \int \lim_{n\to\infty} f_n d\mu$.

Dominated Convergence Theorem: *Let* f_n *the sequence of real value and measurable functions. If the sequence convergence pointwise to the function* f *and it is dominated by some integrable function* g *then* $\lim_{n\to\infty} \int f_n d\mu = \int f d\mu$. *A function is dominated by* g *if* $|f_n(x)| < g(x)$.

Chapter 4

Path Integral Stochastic Optimal Control

After discussing the connection between PDEs, SDEs and the Path Integrals in this chapter we present the application of path integral formalism to stochastic optimal control and reinforcement learning problems. While reinforcement learning (RL) is among the most general frameworks of learning control to create truly autonomous learning systems, its scalability to high-dimensional continuous state-action system, e.g., humanoid robots, remains problematic. Classical value-function based methods with function approximation offer one possible approach, but function approximation under the non-stationary iterative learning process of the value-function remains difficult when one exceeds about 5-10 dimensions. Alternatively, direct policy learning from trajectory roll-outs has recently made significant progress (Peters 2007), but can still become numerically brittle and full of open tuning parameters in complex learning problems. In new developments, RL researchers have started to combine the well-developed methods from statistical learning and empirical inference with classical RL approaches in order to minimize tuning parameters and numerical problems, such that ultimately more efficient algorithms can be developed that scale to significantly more complex learning system (Dayan

& Hinton 1997, Kober & Peters 2009, Peters & Schaal 2008*a*, Toussaint & Storkey 2006) and (Ghavamzadeh & Yaakov 2007, Deisenroth, Rasmussen & Peters 2009, Vlassis, Toussaint, Kontes & S. 2009, Jetchev & Toussaint 2009).

In the spirit of these latter ideas, in this chapter we derive the necessary mathematical background for the development of a new method of probabilistic reinforcement learning based on the framework of stochastic optimal control and path integrals. We start our analysis motivated by the original work of (Kappen 2007, Broek, Wiegerinck & Kappen. 2008) and we extend the path integral control framework in new directions which include 1) stochastic dynamics with state dependent control and diffusion matrices, 2) the iterative version of the proposed framework and 3) different integration schemes of stochastic calculus which include but they are not limited to Itô and Stratonovich calculus.

The present chapter is organized as follows: in section 4.1, we go through the first steps of Path Integral control, starting with the presentation of a general stochastic optimal control problem and the corresponding HJB equation. We continue with the transformation of HJB to a linear and second order PDE, the so called the Backward Chapman Kolmogorov PDE. This transformation allows us to use the Feynman-Kac lemma, from chapter 3, and to represent the solution for the backward Chapman Kolmogorov PDE as the expectation of the exponentiated state depended part of the cost function over all possible trajectories.

In section 4.2 we derive the path integral formalism for stochastic dynamic systems in which the state is partitioned into directly actuated and not directly actuated parts. There is a plethora of dynamical systems that have this property such as Rigid body and

Multi-body dynamics as well as the Dynamic Movement Primitives. DMPs are nonlinear attractors with adjustable landscapes and they can be used to represent state space trajectories as well as control policies. We will continue the discussion on DMP and their application to robotic optimal control and planning in chapter 6. The derivation of path integral for such type of systems is based on the Itô calculus and it is presented step by step.

In section 4.3 the generalized path integral control for the case of systems with state dependent control transition and diffusion matrices is derived. The derivation is presented in details in the appendix of the present chapter and it consists of 2 lemmas and 1 theorem. All the analysis in sections 4.2 and 4.3 is according to Itô calculus. To complete the presentation on the generalized path integral control we present the derivation of the optimal controls in Stratonovich calculus. Furthermore, we discus the case in which the Stratonovich and the Itô calculus lead to the same results in the terms of the final formula that provides the path integral optimal controls.

With the goal to apply the Path Integrals control to high dimensional robotic control and planning problems, in section 4.6 we present the Iterative version of the path integral control and we have a discussion on the convergence analysis of the proposed algorithm ($\mathbf{PI}^2$). When the iterative path integral control approach is applied to DMPs then the resulting algorithm is the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$). This algorithm is presented in great detail in chapter 6.

Finally, in section 4.7 we discuss the risk sensitive version of path integral control. More precisely we derive the condition under which the path integral control formalism could be applied for the case of stochastic optimal control problems with risk sensitive

cost functions. In the last section we discuss the main points of this chapter and we conclude.

4.1 Path integral stochastic optimal control

The goal in stochastic optimal control is to control a stochastic dynamical system while minimizing a performance criterion. Therefore, in mathematical term a stochastic optimal control problem can be formulated as follows:

$$V(\mathbf{x}) = \min_{\mathbf{u}} J(\mathbf{x}, \mathbf{u}) = \min_{\mathbf{u}} \left\langle \phi(\mathbf{x}_{t_N}) + \int_{to}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u}, t) dt \right\rangle \tag{4.1}$$

subject to the stochastic dynamical constrains:

$$\mathbf{dx} = (\mathbf{f}(\mathbf{x}, t) + \mathbf{G}(\mathbf{x}, t)\mathbf{u})\, dt + \mathbf{B}(\mathbf{x}, t)\mathbf{L} d\mathbf{w} \tag{4.2}$$

with $\mathbf{x}_t \in \Re^{n\times 1}$ denoting the state of the system, $\mathbf{G}_t = \mathbf{G}(\mathbf{x}, t) \in \Re^{n\times p}$ the control matrix, $\mathbf{B}_t = \mathbf{B}(\mathbf{x}, t) \in \Re^{n\times p}$ is the diffusions matrix $\mathbf{f}_t = \mathbf{f}(\mathbf{x}, t) \in \Re^{n\times 1}$ the passive dynamics, $\mathbf{u}_t \in \Re^{p\times 1}$ the control vector and $d\mathbf{w} \in \Re^{p\times 1}$ brownian noise. $\mathbf{L} \in \Re^{p\times p}$ is a state independent matrix with $\boldsymbol{\Sigma}_{\mathbf{w}} = \mathbf{L}\mathbf{L}^T$. As immediate reward we consider

$$\boldsymbol{\mathcal{L}}_t = \boldsymbol{\mathcal{L}}(\mathbf{x}_t, \mathbf{u}_t, t) = q_t + \frac{1}{2}\mathbf{u}_t^T \mathbf{R} \mathbf{u}_t \tag{4.3}$$

where $q_t = q(\mathbf{x}_t, t)$ is an arbitrary state-dependent cost function, and $\mathbf{R}$ is the positive definite weight matrix of the quadratic control cost. The stochastic HJB equation (Stengel

1994, Fleming & Soner 2006) associated with this stochastic optimal control problem is expressed as follows:

$$-\partial_t V_t = \min_{\mathbf{u}} \left(\boldsymbol{\mathcal{L}}_t + (\nabla_{\mathbf{x}} V_t)^T \mathbf{F}_t + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V_t) \mathbf{B}_t \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}_t^T \right) \right) \tag{4.4}$$

To find the minimum, the reward function (4.3) is inserted into (4.4) and the gradient of the expression inside the parenthesis is taken with respect to controls $\mathbf{u}$ and set to zero. The corresponding optimal control is given by the equation:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_t = -\mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T (\nabla_{\mathbf{x}} V(\mathbf{x}, t)) \tag{4.5}$$

Substitution of the optimal control into the stochastic HJB (4.4) results in the following nonlinear and second order PDE:

$$-\partial_t V_t = q_t + (\nabla_{\mathbf{x}} V_t)^T \mathbf{f}_t - \frac{1}{2} (\nabla_{\mathbf{x}} V_t)^T \mathbf{G}_t \mathbf{R}^{-1} \mathbf{G}_t^T (\nabla_{\mathbf{x}} V_t) + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} V_t) \mathbf{B}_t \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}_t^T \right) \tag{4.6}$$

To transform the PDE above into a linear one, we use a exponential transformation of the value function $V_t = -\lambda \log \Psi_t$. Given this logarithmic transformation, the partial derivatives of the value function with respect to time and state are expressed as follows:

$$\partial_t V_t = -\lambda \frac{1}{\Psi_t} \partial_t \Psi_t \tag{4.7}$$

$$\nabla_{\mathbf{x}} V_t = -\lambda \frac{1}{\Psi_t} \nabla_{\mathbf{x}} \Psi_t \tag{4.8}$$

$$\nabla_{\mathbf{xx}} V_t = \lambda \frac{1}{\Psi_t^2} \nabla_{\mathbf{x}} \Psi_t \ \nabla_{\mathbf{x}} \Psi_t^T - \lambda \frac{1}{\Psi_t} \nabla_{\mathbf{xx}} \Psi_t \tag{4.9}$$

Inserting the logarithmic transformation and the derivatives of the value function we obtain:

$$\frac{\lambda}{\Psi_t}\partial_t\Psi_t = q_t - \frac{\lambda}{\Psi_t}(\nabla_{\mathbf{x}}\Psi_t)^T\mathbf{f}_t - \underline{\frac{\lambda^2}{2\Psi_t^2}(\nabla_{\mathbf{x}}\Psi_t)^T\mathbf{G}_t\mathbf{R}^{-1}\mathbf{G}_t^T(\nabla_{\mathbf{x}}\Psi_t)} \tag{4.10}$$

$$+\frac{1}{2}tr\left(\Gamma\right) \tag{4.11}$$

where the term Γ is expressed as:

$$\Gamma = \left(\lambda\frac{1}{\Psi_t^2}\nabla_{\mathbf{x}}\Psi_t\ \nabla_{\mathbf{x}}\Psi_t^T - \lambda\frac{1}{\Psi_t}\nabla_{\mathbf{xx}}\Psi_t\right)\mathbf{B}_t\mathbf{\Sigma}_{\mathbf{w}}\mathbf{B}_t^T \tag{4.12}$$

The tr of Γ is therefore:

$$\Gamma = \underline{\lambda\frac{1}{\Psi^2}tr\left(\nabla_{\mathbf{x}}\Psi_t^T\mathbf{B}_t\mathbf{\Sigma}_{\mathbf{w}}\mathbf{B}_t\nabla_{\mathbf{x}}\Psi_t\right)} - \lambda\frac{1}{\Psi_t}tr\left(\nabla_{\mathbf{xx}}\Psi_t\mathbf{B}_t\mathbf{\Sigma}_{\mathbf{w}}\mathbf{B}_t^T\right) \tag{4.13}$$

Comparing the underlined terms in (4.11) and (4.55), one can recognize that these terms will cancel under the assumption $\lambda\mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T = \mathbf{B}(\mathbf{x})\mathbf{\Sigma}_{\mathbf{w}}\mathbf{B}(\mathbf{x})^T = \mathbf{\Sigma}(\mathbf{x}_t) = \mathbf{\Sigma}_t$. The resulting PDE is formulated as follows:

$$-\partial_t\Psi_t = -\frac{1}{\lambda}q_t\Psi_t + \mathbf{f}_t^T(\nabla_{\mathbf{x}}\Psi_t) + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t)\mathbf{\Sigma}_t\right) \tag{4.14}$$

with boundary condition: $\Psi_{t_N} = \Psi(\mathbf{x}, t_N) = \exp\left(-\frac{1}{\lambda}\phi(\mathbf{x}_{t_N})\right)$. The partial differential equation (PDE) in (4.14) corresponds to the so called Chapman Kolmogorov PDE, which is of second order and linear. Analytical solutions of even linear PDEs are plausible only in very special cases which correspond to systems with trivial low dimensional dynamics.

In this work we compute the solution of the linear PDE above with the use of the Feynman - Kac lemma (Øksendal 2003). The Feynman- Kac lemma provides a connection between stochastic differential equations and PDEs and therefore its use is twofold. On one side it can be used to find probabilistic solutions of PDEs based on forward sampling of diffusions while on the other side it can be used find solution of SDEs based on deterministic methods that numerically solve PDEs. The solution of the PDE above can be found by evaluating the expectation:

$$\Psi\left(\mathbf{x}, t_i\right) = \left\langle e^{-\int_{t_i}^{t_N} \frac{1}{\lambda} q(\mathbf{x}) dt} \Psi(\mathbf{x}_{t_N}) \right\rangle_{\boldsymbol{\tau}_i} \tag{4.15}$$

on sample paths $\boldsymbol{\tau}_i = (\mathbf{x}_i, ..., \mathbf{x}_{t_N})$ generated with the forward sampling of the diffusion equation $\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)\mathbf{L}d\mathbf{w}$. Under the use of the Feynman Kac lemma the stochastic optimal control problem has been transformed into an approximation problem of a path integral. With a view towards a discrete time approximation, which will be needed for numerical implementations, the solution (4.15) can be formulated as:

$$\Psi\left(\mathbf{x}, t_i\right) = \lim_{dt \to 0} \int p\left(\boldsymbol{\tau}_i | \mathbf{x}_i\right) \exp\left[-\frac{1}{\lambda} \left(\phi(\mathbf{x}(t_N)) + \sum_{j=i}^{N-1} q(\mathbf{x}, t_j) dt \right) \right] d\boldsymbol{\tau}_i \tag{4.16}$$

where $\boldsymbol{\tau}_i = (\mathbf{x}_{t_i}, ..., \mathbf{x}_{t_N})$ is a sample path (or trajectory piece) starting at state $\mathbf{x}_{t_i}$ and the term $p\left(\boldsymbol{\tau}_i | \mathbf{x}_i\right)$ is the probability of sample path $\boldsymbol{\tau}_i$ conditioned on the start state $\mathbf{x}_{t_i}$. Since equation (4.16) provides the exponential cost to go Ψ_{t_i} in state $\mathbf{x}_{t_i}$, the integration above is taken with respect to sample paths $\boldsymbol{\tau}_i = \left(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i+1}}, ..., \mathbf{x}_{t_N}\right)$. The differential term $d\boldsymbol{\tau}_i$ is defined as $d\boldsymbol{\tau}_i = (d\mathbf{x}_{t_i}, ..., d\mathbf{x}_{t_N})$. After the exponentiated value function

$\Psi(\mathbf{x}, t)$ has been approximated, the optimal control are found according to the equation that follows:

$$\mathbf{u}(\mathbf{x}, t) = \lambda \mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T \frac{\nabla_{\mathbf{x}} \Psi(\mathbf{x}, t)}{\Psi(\mathbf{x}, t)} \tag{4.17}$$

Clearly optimal controls in the equation above act such that the stochastic dynamical system visits regions of the state space with high exponentiated values function $\Psi(\mathbf{x}, t)$ while in the optimal control formulation (4.5) controls will move the system towards part of the state space with minimum cost-to-go $V(\mathbf{x}, t)$. This observation is in complete agreement with the exponentiation of value function $\Psi(\mathbf{x}, t) = \exp\left(-\frac{1}{\lambda} V(\mathbf{x}, t)\right)$. Essentially, the resulting value function $\Psi(\mathbf{x}, t)$ can be thought as a probability of the state and thus states with high cost to go $V(\mathbf{x}, t)$ will be less probable(= small $\Psi(\mathbf{x}, t)$) while state with small cost to go will be most probable. In that sense the stochastic optimal control has been transformed from a minimization to maximization optimization problem. Finally the intuition behind the condition $\lambda \mathbf{G}(\mathbf{x}, t) \mathbf{R}^{-1} \mathbf{G}(\mathbf{x}, t)^T = \mathbf{B}(\mathbf{x}, t) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t)^T$ is that, since the weight control matrix $\mathbf{R}$ is inverse proportional to the variance of the noise, a high variance control input implies cheap control cost, while small variance control inputs have high control cost. From a control theoretic stand point such a relationship makes sense due to the fact that under a large disturbance (= high variance) significant control authority is required to bring the system back to a desirable state. This control authority can be achieved with corresponding low control cost in $\mathbf{R}$.

With the goal to find the $\Psi(\mathbf{x}, t)$ in equation (4.16), in the next section we derive the distribution $p(\boldsymbol{\tau}_i | \mathbf{x}_i)$ based on the passive dynamics. This is a generalization of results in (Kappen 2007, Broek et al. 2008).

4.2 Generalized path integral formalism

To develop our algorithms, we will need to consider a more general development of the path integral approach to stochastic optimal control than presented in (Kappen 2007) and (Broek et al. 2008). In particular, we have to address that in many stochastic dynamical systems, the control transition matrix $\mathbf{G}_t$ is state dependent and its structure depends on the partition of the state in directly and non-directly actuated parts. Since only some of the states are directly controlled, the state vector is partitioned into $\mathbf{x} = [\mathbf{x}^{(m)^T} \quad \mathbf{x}^{(c)^T}]^T$ with $\mathbf{x}^{(m)} \in \Re^{k\times 1}$ the non-directly actuated part and $\mathbf{x}^{(c)} \in \Re^{l\times 1}$the directly actuated part. Subsequently, the passive dynamics term and the control transition matrix can be partitioned as $\mathbf{f}_t = [\mathbf{f}_t^{(m)^T} \quad \mathbf{f}_t^{(c)^T}]^T$ with $\mathbf{f}_m \in \Re^{k\times 1}$, $\mathbf{f}_c \in \Re^{l\times 1}$ and $\mathbf{G}_t = [\mathbf{0}_{k\times p} \quad \mathbf{G}_t^{(c)^T}]^T$ with $\mathbf{G}_t^{(c)} \in \Re^{l\times p}$. The discretized state space representation of such systems is given as:

$$\mathbf{x}_{t_{i+1}} = \mathbf{x}_{t_i} + \mathbf{f}_{t_i} dt + \mathbf{G}_{t_i} \mathbf{u}_{t_i} dt + \mathbf{B}_{t_i} d\mathbf{w}_{t_i},$$

or, in partitioned vector form:

$$\begin{pmatrix} \mathbf{x}_{t_{i+1}}^{(m)} \\ \mathbf{x}_{t_{i+1}}^{(c)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{t_i}^{(m)} \\ \mathbf{x}_{t_i}^{(c)} \end{pmatrix} + \begin{pmatrix} \mathbf{f}_{t_i}^{(m)} \\ \mathbf{f}_{t_i}^{(c)} \end{pmatrix} dt + \begin{pmatrix} \mathbf{0}_{k\times p} \\ \mathbf{G}_{t_i}^{(c)} \end{pmatrix} \mathbf{u}_{t_i} dt + \begin{pmatrix} \mathbf{0}_{k\times p} \\ \mathbf{B}_{t_i}^{(c)} \end{pmatrix} d\mathbf{w}_{t_i}. \quad (4.18)$$

Essentially the stochastic dynamics are partitioned into controlled equations in which the state $\mathbf{x}_{t_{i+1}}^{(c)}$ is directly actuated and the uncontrolled equations in which the state $\mathbf{x}_{t_{i+1}}^{(m)}$

is not directly actuated. Since stochasticity is only added in the directly actuated terms (c) of (4.18), we can develop $p\left(\boldsymbol{\tau}_i|\mathbf{x}_i\right)$ as follows.

$$\begin{aligned} p\left(\boldsymbol{\tau}_i|\mathbf{x}_{t_i}\right) &= p\left(\boldsymbol{\tau}_{i+1}|\mathbf{x}_{t_i}\right) \\ &= p\left(\mathbf{x}_{t_N}, ..., \mathbf{x}_{t_{i+1}}|\mathbf{x}_{t_i}\right) \\ &= \Pi_{j=i}^{N-1} p\left(\mathbf{x}_{t_{j+1}}|\mathbf{x}_{t_j}\right), \end{aligned}$$

where we exploited the fact that the start state $\mathbf{x}_{t_i}$ of a trajectory is given and does not contribute to its probability. For systems where the control has lower dimensionality than the state (4.18), the transition probabilities $p\left(\mathbf{x}_{t_{j+1}}|\mathbf{x}_{t_j}\right)$ are factorized as follows:

$$\begin{aligned} p\left(\mathbf{x}_{t_{j+1}}|\mathbf{x}_{t_j}\right) &= p\left(\mathbf{x}_{t_{j+1}}^{(m)}|\mathbf{x}_{t_j}\right) p\left(\mathbf{x}_{t_{j+1}}^{(c)}|\mathbf{x}_{t_j}\right) \\ &= p\left(\mathbf{x}_{t_{j+1}}^{(m)}|\mathbf{x}_{t_j}^{(m)}, \mathbf{x}_{t_j}^{(c)}\right) p\left(\mathbf{x}_{t_{j+1}}^{(c)}|\mathbf{x}_{t_j}^{(m)}, \mathbf{x}_{t_j}^{(c)}\right) \\ &\propto p\left(\mathbf{x}_{t_{j+1}}^{(c)}|\mathbf{x}_{t_j}\right), \end{aligned} \tag{4.19}$$

where we have used the fact that $p\left(\mathbf{x}_{t_{i+1}}^{(m)}|\mathbf{x}_{t_i}^{(m)}, \mathbf{x}_{t_i}^{(c)}\right)$ is the Dirac delta function, since $\mathbf{x}_{t_{j+1}}^{(m)}$ can be computed deterministically from $\mathbf{x}_{t_j}^{(m)}, \mathbf{x}_{t_j}^{(c)}$. For all practical purposes,[1] the transition probability of the stochastic dynamics is reduced to the transition probability of the directly actuated part of the state:

$$p\left(\boldsymbol{\tau}_i|\mathbf{x}_{t_i}\right) = \Pi_{j=i}^{N-1} p\left(\mathbf{x}_{t_{j+1}}|\mathbf{x}_{t_j}\right) \propto \Pi_{j=i}^{N-1} p\left(\mathbf{x}_{t_{j+1}}^{(c)}|\mathbf{x}_{t_j}\right). \tag{4.20}$$

[1]The delta functions will all integrate to 1 in the path integral.

Since we assume that the noise $\boldsymbol{\epsilon}$ is zero mean Gaussian distributed with variance $\boldsymbol{\Sigma}_{\mathbf{w}}$, where $\boldsymbol{\Sigma}_{\mathbf{w}} = \mathbf{L}\mathbf{L}^T \in \Re^{l \times l}$, the transition probability of the directly actuated part of the state is defined as:[2]

$$p\left(\mathbf{x}_{t_{j+1}}^{(c)} | \mathbf{x}_{t_j}\right) = \frac{1}{\left((2\pi)^l \cdot |\boldsymbol{\Sigma}_{t_j}|\right)^{1/2}} \exp\left(-\frac{1}{2} \left\| \mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)} - \mathbf{f}_{t_j}^{(c)} dt \right\|_{\boldsymbol{\Sigma}_{t_j}^{-1}}^2\right), \tag{4.21}$$

where the covariance $\boldsymbol{\Sigma}_{t_j} \in \Re^{l \times l}$ is expressed as $\boldsymbol{\Sigma}_{t_j} = \mathbf{B}_{t_j}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}_{t_j}^{(c)^T} dt$. Combining (4.21) and (4.20) results in the probability of a path expressed as:

$$p\left(\boldsymbol{\tau}_i | \mathbf{x}_{t_i}\right) \propto \frac{1}{\Pi_{j=i}^{N-1} \left((2\pi)^l \|\boldsymbol{\Sigma}_{t_j}|\right)^{1/2}} \exp\left(-\frac{1}{2} \sum_{j=1}^{N-1} \left\| \mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)} - \mathbf{f}_{t_j}^{(c)} dt \right\|_{\boldsymbol{\Sigma}_{t_j}^{-1}}^2\right).$$

Finally, we incorporate the assumption (4.56) about the relation between the control cost and the variance of the noise, which needs to be adjusted to the controlled space as $\boldsymbol{\Sigma}_{t_j} = \mathbf{B}_{t_j}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}_{t_j}^{(c)^T} dt = \lambda \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T} dt = \lambda \mathbf{H}_{t_j} dt$ with $\mathbf{H}_{t_j} = \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T}$. Thus, we obtain:

$$p\left(\boldsymbol{\tau}_i | \mathbf{x}_{t_i}\right) \propto \frac{1}{\Pi_{j=i}^{N-1} \left((2\pi)^l |\boldsymbol{\Sigma}_{t_j}|\right)^{1/2}} \exp\left(-\frac{1}{2\lambda} \sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt\right).$$

[2] For notational simplicity, we write weighted square norms (or Mahalanobis distances) as $\mathbf{v}^T \mathbf{M} \mathbf{v} = \|\mathbf{v}\|_{\mathbf{M}}^2$.

With this formulation of the probability of a trajectory, we can rewrite the the path integral (4.16) as:

$$\Psi_{t_i} = \\ = \lim_{dt\to 0} \int \frac{\exp\left(-\frac{1}{\lambda}\left(\phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt + \frac{1}{2}\sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|^2_{\mathbf{H}_{t_j}^{-1}} dt\right)\right)}{\Pi_{j=i}^{N-1}\left((2\pi)^{l/2}|\mathbf{\Sigma}_{t_j}|^{1/2}\right)} d\boldsymbol{\tau}_i^{(c)} \tag{4.22}$$

Or in a more compact form:

$$\Psi_{t_i} = \lim_{dt\to 0} \int \frac{1}{D(\boldsymbol{\tau}_i)} \exp\left(-\frac{1}{\lambda} S(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i^{(c)}, \tag{4.23}$$

where, we defined

$$S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt + \frac{1}{2}\sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|^2_{\mathbf{H}_{t_j}^{-1}} dt,$$

and

$$D(\boldsymbol{\tau}_i) = \prod_{j=i}^{N-1} \left((2\pi)^{l/2}|\mathbf{\Sigma}_{t_j}|^{1/2}\right).$$

Note that the integration is over $d\boldsymbol{\tau}_i^{(c)} = \left(d\mathbf{x}_{t_i}^{(c)}, ..., d\mathbf{x}_{t_N}^{(c)}\right)$, as the non-directly actuated states can be integrated out due to the fact that the state transition of the non-directly actuated states is deterministic, and just added Dirac delta functions in the integral (cf. Equation (4.19)). Equation (4.23) is written in a more compact form as:

$$\begin{aligned}\Psi_{t_i} &= \lim_{dt\to 0} \int \exp\left(-\frac{1}{\lambda} S(\boldsymbol{\tau}_i) - \log D(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i^{(c)} \\ &= \lim_{dt\to 0} \int \exp\left(-\frac{1}{\lambda} Z(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i^{(c)},\end{aligned} \tag{4.24}$$

where $Z(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \lambda \log D(\boldsymbol{\tau}_i)$. It can be shown (see appendix) that this term is factorized in path dependent and path independent terms of the form:

$$Z(\boldsymbol{\tau}_i) = \tilde{S}(\boldsymbol{\tau}_i) + \frac{\lambda(N-i)l}{2} \log\left(2\pi dt\lambda\right),$$

where

$$\tilde{S}(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log|\boldsymbol{\mathcal{B}}_{t_j}| \tag{4.25}$$

with $\boldsymbol{\mathcal{B}} = \mathbf{B}_{t_j}^{(c)} \mathbf{B}_{t_j}^{(c)^T}$. This formula is a required step for the derivation of optimal controls in the next section. The constant term $\frac{\lambda(N-i)l}{2} \log\left(2\pi dt\lambda\right)$ can be the source of numerical instabilities especially in cases where fine discretization dt of stochastic dynamics is required. However, in the next section, and in a great detail in Appendix A, lemma 1, we show how this term drops out of the equations.

4.3 Path integral optimal controls

For every moment of time, the optimal controls are given as $\mathbf{u}_{t_i} = -\mathbf{R}^{-1}\mathbf{G}_{t_i}^T(\nabla_{x_{t_i}} V_{t_i})$. Due to the exponential transformation of the value function, the equation of the optimal controls can be written as

$$\mathbf{u}_{t_i} = \lambda \mathbf{R}^{-1}\mathbf{G}_{t_i} \frac{\nabla_{\mathbf{x}_{t_i}} \Psi_{t_i}}{\Psi_{t_i}}.$$

After substituting Ψ_{t_i} with (4.24) and canceling the state independent terms of the cost we have:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T \frac{\nabla_{\mathbf{x}_{t_i}^{(c)}} \left(\int e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i^{(c)} \right)}{\int \; e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i^{(c)}} \right),$$

Further analysis of the equation above leads to a simplified version for the optimal controls as

$$\boxed{\mathbf{u}_{t_i} dt = \int P\left(\boldsymbol{\tau}_i\right) \mathbf{u}_L\left(\boldsymbol{\tau}_i\right) d\boldsymbol{\tau}_i^{(c)}}, \tag{4.26}$$

with the probability $P\left(\boldsymbol{\tau}_i\right)$ and local controls $\mathbf{u}_L\left(\boldsymbol{\tau}_i\right)$ defined as

$$\boxed{P\left(\boldsymbol{\tau}_i\right) = \frac{e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)}}{\int e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i}} \tag{4.27}$$

The local control can now be expressed as:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\mathbf{H}_{t_i}^{-1}\mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i},$$

By substituting $\mathbf{H}_{t_i} = \mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T}$ in the equation above, we get our main result for the local controls of the sampled path for the generalized path integral formulation:

$$\boxed{\mathbf{u}_L(\boldsymbol{\tau}_i) = \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i}}. \tag{4.28}$$

Given the local control above the optimal control in (4.26) are now expressed by the equation that follows:

$$\boxed{\mathbf{u}_{t_i} dt = \int P\left(\boldsymbol{\tau}_i\right) \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i} d\boldsymbol{\tau}_i^{(c)}}, \tag{4.29}$$

The equations in boxes (4.29) and (4.27) the solution for the generalized path integral stochastic optimal control problem. The numerical evaluation of the integral above is expressed by the equation

$$\boxed{\mathbf{u}\left(\boldsymbol{\tau}_i\right) dt = \sum_{k=1}^{\#Paths} \tilde{p}^{(k)}\left(\boldsymbol{\tau}_i\right) \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \left(\mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i}^{(k)} \right)} \tag{4.30}$$

The equation above can also be written in the form:

$$\boxed{\mathbf{u}\left(\boldsymbol{\tau}_i\right) dt = \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \sum_{k=1}^{\#Paths} \tilde{p}^{(k)}\left(\boldsymbol{\tau}_i\right) \left(\mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i}^{(k)} \right)} \tag{4.31}$$

- **Given**:
  - The system dynamics $\mathbf{x}_{t_{i+1}} = \mathbf{x}_{t_i} + (\mathbf{f}_{t_i} + \mathbf{G}_t \mathbf{u}_t)\, dt + \mathbf{B}_{t_i} d\mathbf{w}_{t_i}$ (cf. 4.2)
  - The immediate cost $\boldsymbol{\mathcal{L}}_t = q_t + \frac{1}{2}\mathbf{u}_t^T \mathbf{R} \mathbf{u}_t$ (cf. 4.3)
  - A terminal cost term ϕ_{t_N}
  - Trajectory starting at t_i and ending at t_N: $\boldsymbol{\tau}_i = (\mathbf{x}_{t_i}, ..., \mathbf{x}_{t_N})$
  - A partitioning of the system dynamics into (c) controlled and (m) uncontrolled equations, where $n = c + m$ is the dimensionality of the state $\mathbf{x}_t$ (cf. Section 4.2)
- **Optimal Controls**:
  - Optimal controls at every time step t_i: $\mathbf{u}_{t_i} dt = \int P(\boldsymbol{\tau}_i)\, \mathbf{u}_L(\boldsymbol{\tau}_i)\, d\boldsymbol{\tau}_i^{(c)}$
  - Probability of a trajectory: $P(\boldsymbol{\tau}_i) = \frac{e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)}}{\int e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i}$
  - Generalized trajectory cost: $\tilde{S}(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \frac{\lambda}{2}\sum_{j=i}^{N-1} \log|\boldsymbol{\mathcal{B}}_{t_j}|$ where
    - $S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt + \frac{1}{2}\sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt$
    - $\mathbf{H}_{t_j} = \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T}$ and $\boldsymbol{\mathcal{B}} = \mathbf{B}_{t_j}^{(c)} \mathbf{B}_{t_j}^{(c)^T}$
  - Local Controls: $\mathbf{u}_L(\boldsymbol{\tau}_i) = \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \left(\mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i} \right)$.

Table 4.1: Summary of optimal control derived from the path integral formalizm.

Given that this result is of general value and constitutes the foundation to derive our reinforcement learning algorithm in the next section, but also since many other special cases can be derived from it, we summarized all relevant equations in Table 4.1.

The **Given** components of Table 4.1 include a model of the system dynamics, the cost function, knowledge of the system's noise process, and a mechanism to generate trajectories $\boldsymbol{\tau}_i$. It is important to realize that this is a *model-based* approach, as the computations of the optimal controls requires knowledge of $\boldsymbol{\epsilon}_i$. $\boldsymbol{\epsilon}_i$ can be obtained in two ways. First, the trajectories $\boldsymbol{\tau}_i$ can be generated purely in simulation, where the noise

is generated from a random number generator. Second, trajectories could be generated by a real system, and the noise $\boldsymbol{\epsilon}_i$ would be computed from the difference between the actual and the predicted system behavior, that is, $\mathbf{G}_{t_i}^{(c)}\boldsymbol{\epsilon}_i = \dot{\mathbf{x}}_{t_i} - \hat{\dot{\mathbf{x}}}_{t_i} = \dot{\mathbf{x}}_{t_i} - (\mathbf{f}_{t_i} + \mathbf{G}_{t_i}\mathbf{u}_{t_i})$. Computing the prediction $\hat{\dot{\mathbf{x}}}_{t_i}$ also requires a model of the system dynamics.

In the next section we show how our generalized formulation is specialized to different classes of stochastic dynamical systems and we provide the corresponding formula of local controls for each class.

4.4 Path integral control for special classes of dynamical systems

The purpose of this section is twofold. First, it demonstrates how to apply the path integral approach to specialized forms of dynamical systems, and how the local controls in (4.28) simplify for these cases. Second, this section prepares the special case which we will need for our reinforcement learning algorithm in presented in the next chapter.

The generalized formulation of stochastic optimal control with path integrals in Table 4.1 can be applied to a variety of stochastic dynamical systems with different types of control transition matrices. One case of particular interest is where the dimensionality of the directly actuated part of the state is 1D, while the dimensionality of the control vector is 1D or higher dimensional. As will be seen below, this situation arises when the controls are generated by a linearly parameterized function approximator. The control

transition matrix thus becomes a row vector $\mathbf{G}_{t_i}^{(c)} = \mathbf{g}_{t_i}^{(c)T} \in \Re^{1\times p}$. According to (4.28), the local controls for such systems are expressed as follows:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}}{\mathbf{g}_{t_i}^{(c)T}\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}} \left(\mathbf{g}_{t_i}^{(c)T} d\mathbf{w}_{t_i}\right).$$

Since the directly actuated part of the state is 1D, the vector $\mathbf{x}_{t_i}^{(c)}$ collapses into the scalar $x_{t_i}^{(c)}$ which appears in the partial differentiation above. In the case that $\mathbf{g}_{t_i}^{(c)}$ does not depend on $x_{t_i}^{(c)}$, the differentiation with respect to $x_{t_i}^{(c)}$ results to zero and the the local controls simplify to:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}\mathbf{g}_{t_i}^{(c)T}}{\mathbf{g}_{t_i}^{(c)T}\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}} d\mathbf{w}_{t_i}$$

The generalized formula of the local controls (4.28) was derived for the case where the control transition matrix is state dependent and its dimensionality is $\mathbf{G}_t^{(c)} \in \Re^{l\times p}$ with $l < n$ and p the dimensionality of the control. There are many special cases of stochastic dynamical systems in optimal control and robotic applications that belong into this general class. More precisely, for systems having a state dependent control transition matrix that is square ($\mathbf{G}_{t_i}^{(c)} \in \Re^{l\times l}$ with $l = p$) the local controls based on (4.28) are reformulated as:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = d\mathbf{w}_{t_i}. \tag{4.32}$$

Interestingly, a rather general class of mechanical systems such as rigid-body and multi-body dynamics falls into this category. When these mechanical systems are expressed in state space formulation, the control transition matrix is equal to rigid body

inertia matrix $\mathbf{G}_{t_i}^{(c)} = \mathbf{M}(\theta_{t_i})$ (Sciavicco & Siciliano 2000). Future work will address this special topic of path integral control for multi-body dynamics.

Another special case of systems with partially actuated state is when the control transition matrix is state independent and has dimensionality $\mathbf{G}_t^{(c)} = \mathbf{G}^{(c)} \in \Re^{l \times p}$. The local controls, according to (4.28), become:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = \mathbf{R}^{-1}\mathbf{G}^{(c)^T}\left(\mathbf{G}^{(c)}\mathbf{R}^{-1}\mathbf{G}^{(c)^T}\right)^{-1}\mathbf{G}^{(c)}d\mathbf{w}_{t_i}. \tag{4.33}$$

If $\mathbf{G}_{t_i}^{(c)}$ is square and state independent, $\mathbf{G}_{t_i}^{(c)} = \mathbf{G}^{(c)} \in \Re^{l \times l}$, we will have:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) = d\mathbf{w}_{t_i}. \tag{4.34}$$

This special case was explored in (Kappen 2005*a*), (Kappen 2007), (Kappen 2005*b*) and (Broek et al. 2008). Our generalized formulation allows a broader application of path integral control in areas like robotics and other control systems, where the control transition matrix is typically partitioned into directly and non-directly actuated states, and typically also state dependent.

4.5 Itô versus Stratonovich path integral stochastic optimal control

The derivation of the Path Integral for the systems with partitioned state into directly and no directly actuated parts was performed based on the Itô stochastic calculus. In

this section we derive the path integral control for the case of Stratonovich stochastic calculus. We consider the dynamics:

$$dx = f(x,t)dt + g(x)(udt + dw) \tag{4.35}$$

We follow the same argument required to apply the path integral control framework and we come up with the path integral formulation expressed according to Stratonovich caclulus. For a general integration scheme we have shown that the path integral takes the form:

$$P\Big(x_N, t_N | x_0, t_0\Big) = \int \prod_{i=1}^{N-1} \left(\frac{dx_i}{\sqrt{2\pi\delta t}B_i}\right) \times \exp\left(-\sum_{i=1}^{N} \left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t} - f_i + \alpha(\partial_x B_i)B_i\right)^2 + \beta\ (\partial_x f_i)\right]\delta t\right)$$

For the Stratonovich calculus we can chose $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{2}$ and we will have the path integral:

$$P\Big(x_N, t_N | x_0, t_0\Big) = \int \prod_{i=1}^{N-1} \left(\frac{dx_i}{\sqrt{2\pi\delta t}B_i}\right) \exp\left(-\sum_{i=1}^{N} \left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t} - f_i + \frac{1}{2}(\partial_x B_i)B_i\right)^2\right]\delta t\right) \times \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\partial_x f_i)\delta t\right)$$

we can write the equation above in the form:

$$P\Big(x_N, t_N | x_0, t_0\Big) = \int \prod_{i=1}^{N-1} \left(\frac{dx_i}{\sqrt{2\pi\delta t} B_i}\right) \exp\left(-\sum_{i=1}^{N} \left[\frac{1}{2B_i^2}\left(\frac{\delta x}{\delta t} - \tilde{f}_i\right)^2\right] \delta t\right)$$
$$\times \exp\left(-\frac{1}{2}\sum_{i=1}^{N} (\partial_x f_i)\delta t\right)$$

where $\tilde{f}_i = f_i - \frac{1}{2}(\partial_x B_i)B_i$. The derivation of the optimal control for the scalar case follows the same steps as in appendix but with the difference of using $\tilde{f}_i$ instead of f_i and the additional term $\sum_{i=1}^{N}(\partial_x f_i)\delta t$. It can be shown that the optimal control is now formulated as :

$$\begin{aligned} u(x_{t_i}) &= \int p(\tau_i) u_L d\tau_i \\ &= \int p(\tau_i) \left(\dot{x} - \tilde{f}(x)\right) d\tau_i \end{aligned}$$

In the next section we discuss the iterative version of path integral control.

4.6 Iterative path integral stochastic optimal control

In this section, we show how Path Integral Control is transformed into an iterative process, which has several advantages for use on a real robot. The analysis that follows holds for any stochastic dynamical systems that belongs to the class of systems expressed by (4.2). When the iterative path integral control is applied to Dynamic Movement Primitives then the resulting algorithm is the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals (**PI**^{2}).

However, we will leave the discussion for $\mathbf{PI}^2$ for the next chapter and in this section we present the general version of iterative path integral control.

In particular, we start by looking into the expectation (4.15) in the Feynman Kac Lemma that is evaluated over the trajectories $\boldsymbol{\tau}_i = \left(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i+1}}, ..., \mathbf{x}_{t_N}\right)$ sampled with the forward propagation of uncontrolled diffusion $\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)\mathbf{L}d\mathbf{w}$. This sampling approach is inefficient since it is very likely that parts of the state space relevant to the optimal control task may not be reached by the sampled trajectories at once. In addition, it has poor scalability properties when applied to high dimensional robotic optimal control problems. Besides the reason of poor sampling, it is very common in robotics applications to have an initial controller-policy which is manually tuned and found based on experience. In such cases, the goal is to improve this initial policy by performing an iterative process. At every iteration (i) the policy $\delta\mathbf{u}^{(i-1)}$ is applied to the dynamical system to generate state space trajectories which are going to be used for improving the current policy. The policy improvement results from the evaluation of the expectation (4.16) of the Feynman - Kac Lemma on the sampled trajectories and the use of the path integral control formalism to find $\delta\mathbf{u}^{(i)}$. The old policy $\delta\mathbf{u}^{(i-1)}$ is updated according to $\delta\mathbf{u}^{(i-1)} + \delta\mathbf{u}^{(i)}$ and the process repeats again with the generation of the new state space trajectories according to the updated policy. In mathematical terms the iterative version of Path Integral Control is expressed as follows:

$$V^{(i)}(\mathbf{x}) = \min_{\delta\mathbf{u}^{(i)}} J(\mathbf{x}, \mathbf{u}) = \min_{\delta\mathbf{u}^{(i)}} \left\langle \int_{to}^{t_N} \left(q(\mathbf{x}, t) + \delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i)} \right) dt \right\rangle \tag{4.36}$$

subject to the stochastic dynamical constrains:

$$\mathbf{dx} = \left(\mathbf{f}^{(i)}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)}\right) dt + \mathbf{B}(\mathbf{x})\mathbf{L}d\mathbf{w} \tag{4.37}$$

where $\mathbf{f}^{(i)}(\mathbf{x}_t) = \mathbf{f}^{(i-1)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i-1)}$ where $\delta\mathbf{u}^{(i-1)}$ is the control correction found in the previous iteration. The linear HJB equation is now formulated as:

$$-\partial_t \Psi_t^{(i)} = -\frac{1}{\lambda} q_t \Psi_t^{(i)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}}\Psi_t^{(i)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}}\Psi_t^{(i)})\mathbf{\Sigma}\right) \tag{4.38}$$

The solution of PDE above is given by

$$\Psi^{(i)}(\mathbf{x}_t) = \left\langle e^{-\int_{t_i}^{t_N} \frac{1}{\lambda} q(\mathbf{x}) dt} \Psi(\mathbf{x}_{t_N}) \right\rangle_{\boldsymbol{\tau}^{(i)}} \tag{4.39}$$

where $\boldsymbol{\tau}^{(i)} = (\mathbf{x}_t, ..., \mathbf{x}_{t_N})$ are sampled trajectories generated by the diffusion: $\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{B}(\mathbf{x}, t)\mathbf{L}d\mathbf{w}$. The optimal control at iteration (i) is expressed as:

$$\delta\mathbf{u}^{(i)} = \lambda \mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T \frac{\nabla_{\mathbf{x}}\Psi^{(i)}(\mathbf{x}, t)}{\Psi^{(i)}(\mathbf{x}, t)} \tag{4.40}$$

and it is applied to the dynamics $\mathbf{f}^{(i)}(\mathbf{x}_t)$. The application of the new control results in updating the previous control $\delta\mathbf{u}^{(i-1)}$ and creating the new dynamics $\mathbf{f}^{(i+1)}(\mathbf{x}) = \mathbf{f}^{(i)}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)} = \mathbf{f}^{(i-1)}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\left(\delta\mathbf{u}^{(i)} + \delta\mathbf{u}^{(i-1)}\right)$. At the next iteration $(i+1)$ of

the iterative path integral control, the corresponding exponentiated value function $\Psi^{(i+1)}$ is given by the following PDE:

$$-\partial_t \Psi_t^{(i+1)} = -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i+1)T} (\nabla_{\mathbf{x}} \Psi_t^{(i+1)} + \frac{1}{2} tr \left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)}) \mathbf{\Sigma} \right) \tag{4.41}$$

The solution of the PDE is now expressed as:

$$\Psi^{(i+1)}(\mathbf{x}_t) = \left\langle e^{-\int_{t_i}^{t_N} \frac{1}{\lambda} q(\mathbf{x}) dt} \Psi(\mathbf{x}_{t_N}) \right\rangle_{\boldsymbol{\tau}^{(i+1)}} \tag{4.42}$$

where $\boldsymbol{\tau}^{(i+1)} = (\mathbf{x}_t, ..., \mathbf{x}_{t_N})$ are sampled trajectories generated by the diffusion: $\mathbf{dx} = \mathbf{f}^{(i+1)}(\mathbf{x}_t)dt + \mathbf{B}(\mathbf{x})d\omega$.

Our ultimate goal for the iterative path integral control is to find the sufficient conditions so that at every iteration the value function improves $V^{(i+1)}(\mathbf{x}, t) < V^{(i)}(\mathbf{x}, t) < .... < V^{(0)}(\mathbf{x}, t)$. Since in the path integral control formalism we make use of the transformation $\Psi(\mathbf{x}, t) = \exp\left(-\frac{1}{\lambda} V(\mathbf{x}, t)\right)$ it suffices to show that $\Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t) > .... > \Psi^{(0)}(\mathbf{x}, t)$. If the last condition is true then at every (i) iteration the stochastic dynamical system visits to regions of state space with more and more probable states(= states with high $\Psi(\mathbf{x}, t)$). These states correspond to small value function $V(\mathbf{x}, t)$. To find the condition under which the above is true, we proceed with the analysis that follows. Since we know that $\mathbf{f}^{(i+1)}(\mathbf{x}) = \mathbf{f}^{(i)}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)}$ we substitute in (4.41) and we will have that:

$$-\partial_t \Psi_t^{(i+1)} = -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)})\mathbf{\Sigma}\right) + \delta\mathbf{u}^{(i)T}\mathbf{G}^T(\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) \tag{4.43}$$

substitution of $\delta\mathbf{u}$ results in:

$$\begin{aligned} -\partial_t \Psi_t^{(i+1)} &= -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)})\mathbf{\Sigma}\right) \\ &+ \frac{\lambda}{\Psi_t^{(i)}}(\nabla_{\mathbf{x}} \Psi_t^{(i)})^T \mathbf{G}\mathbf{R}\mathbf{G}^T(\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) \end{aligned} \tag{4.44}$$

or in a more compact form:

$$-\partial_t \Psi_t^{(i+1)} = -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)})\mathbf{\Sigma}\right) + F(\mathbf{x}, t)$$

where

$$\boldsymbol{\mathcal{F}}(\mathbf{x}, t) = \frac{\lambda}{\Psi^{(i)}(\mathbf{x}, t)} \nabla_{\mathbf{x}} \Psi^{(i)}(\mathbf{x}, t)^T \ \mathbf{G}\mathbf{R}\mathbf{G}^T \ \nabla_{\mathbf{x}} \Psi^{(i+1)}(\mathbf{x}, t) \tag{4.45}$$

correspond to a force term which is the inner product of the gradients of the value functions at iterations (i) and $(i+1)$ under the metric $\boldsymbol{\mathcal{M}} = \frac{\lambda}{\Psi^{(i)}(\mathbf{x},t)}\mathbf{G}\mathbf{R}\mathbf{G}^T$. Clearly $\boldsymbol{\mathcal{M}} > 0$ since the matrix product $\mathbf{G}\mathbf{R}\mathbf{G}^T > 0$ is positive definite and $\lambda > 0, \Psi(\mathbf{x}, t) > 0$. Comparing the two PDEs at iteration (i) and $(i+1)$ and by using the differential linear operator $\mathscr{A}^{(i)} = -\frac{1}{\lambda} q_t + \mathbf{f}_t^{(i)T} \nabla_{\mathbf{x}} + \frac{1}{2} tr(\mathbf{\Sigma} \nabla_{\mathbf{xx}})$ we have:

$$
\begin{aligned}
-\partial_t \Psi_t^{(i+1)} &= \mathscr{A}^{(i)} \Psi_t^{(i+1)} + \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \\
-\partial_t \Psi_t^{(i)} &= \mathscr{A}^{(i)} \Psi_t^{(i)}
\end{aligned}
\tag{4.46}
$$

under the terminal condition $\Psi_{t_N}^{(i)} = \exp\left(-\frac{1}{\lambda}\phi(\mathbf{x}_{t_N})\right)$ and $\Psi_{t_N}^{(i+1)} = \exp\left(-\frac{1}{\lambda}\phi(\mathbf{x}_{t_N})\right)$. In the next two subsection we study the two PDEs above, with the goal to find the connection between $\Psi^{(i)}$ and $\Psi^{(i+1)}$.

4.6.1 Iterative path integral Control with equal boundary conditions

In this section we will simplify our analysis and we will assume that over the iterations i the boundary conditions of the corresponding PDEs are the same thus $\Psi_{t_N}^{(i)} = \exp\left(-\frac{1}{\lambda}\phi(\mathbf{x}_{t_N})\right),\ \forall i$. Our analysis is fairly intuitive. We claim that $\Psi^{(i+1)} < \Psi^{(i)}$ if $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) > 0\ \forall \mathbf{x}, t$. To see this result we rewrite equation (4.60) in the following form:

$$
\begin{aligned}
-\partial_t \Psi_t^{(i+1)} &= -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T} (\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)}) \boldsymbol{\Sigma} \right) \\
&+ \frac{1}{\lambda} \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T} \Psi_t^{(i+1)}
\end{aligned}
\tag{4.47}
$$

where we used the fact that $\delta \mathbf{u}^{(i+1)} = \lambda \mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T \frac{\nabla_{\mathbf{x}} \Psi^{(i)}(\mathbf{x},t)}{\Psi^{(i+1)}(\mathbf{x},t)}$ or in a more compact form:

$$
-\partial_t \Psi_t^{(i+1)} = -\frac{1}{\lambda} \tilde{q}_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T} (\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)}) \boldsymbol{\Sigma} \right) \tag{4.48}
$$

where the term $\tilde{q} = \tilde{q}(\mathbf{x}, t, \delta\mathbf{u}^{(i)}, \delta\mathbf{u}^{(i+1)})$ is defined as $\tilde{q} = q(\mathbf{x}, t) - \frac{1}{\lambda}\delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i+1)T}$. To find the relation between $\Psi^{(i)}(\mathbf{x}, t)$ and $\Psi^{(i+1)}(\mathbf{x}, t)$ we first transform the PDE above into a forward PDE and then we follow some intuitive arguments. More precisely we assume the transformation $\Psi(\mathbf{x}, t) = \Phi(\mathbf{x}, T - t) = \Phi(\tau)$. Thus we will have that:

$$\partial_t \Psi_t = -\partial_\tau \Phi_\tau \tag{4.49}$$

The PDE at iteration (i) takes now the form:

$$\partial_\tau \Phi_\tau^{(i)} = -\frac{1}{\lambda} q(\mathbf{x}, T - \tau)\Phi_\tau^{(i)} + \mathbf{f}_\tau^{(i)T}(\nabla_{\mathbf{x}}\Phi_\tau^{(i)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}}\Phi_\tau^{(i)})\mathbf{\Sigma}\right) \tag{4.50}$$

with the initial condition $\Phi(\mathbf{x}, 0) = \exp(-\frac{1}{\lambda}\phi(t_N))$. At iteration $(i + 1)$ we will have:

$$\partial_\tau \Phi_\tau^{(i+1)} = -\frac{1}{\lambda} \tilde{q}(\mathbf{x}, T - \tau)\Phi_\tau^{(i+1)} + \mathbf{f}_\tau^{(i)T}(\nabla_{\mathbf{x}}\Phi_\tau^{(i+1)}) + \frac{1}{2} tr\left((\nabla_{\mathbf{xx}}\Phi_\tau^{(i+1)})\mathbf{\Sigma}\right) \tag{4.51}$$

under the initial condition $\Phi(\mathbf{x}, 0) = \exp(-\frac{1}{\lambda}\phi(t_N))$. Clearly there are 3 cases depending on the sign of $F(\mathbf{x}, t)$ and therefore the sign of $\frac{1}{\lambda}\delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i+1)T}$. More precisely we will have that:

- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, T-t) > 0 \Rightarrow \delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i+1)T} > 0$. By comparing (4.50) with (4.51) we see that state cost $\tilde{q}$ subtracted from $\Phi^{(i+1)}$ is smaller than the state cost q subtracted from $\Phi^{(i)}$ and therefore $\Phi^{(i+1)}(\mathbf{x}, T-t) > \Phi^{(i)}(\mathbf{x}, T-t) \Longrightarrow \Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t)$.

- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, T-t) = 0 \Rightarrow \delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i+1)T} = 0$ the two PDEs (4.50) and (4.51) are identical. Therefore under the same boundary condition $\Phi^{(i+1)}(\mathbf{x}, 0) = \Phi^{(i)}(\mathbf{x}, 0)$ we will have that $\Phi^{(i+1)}(\mathbf{x}, T-t) = \Phi^{(i)}(\mathbf{x}, T-t) \Longrightarrow \Psi^{(i+1)}(\mathbf{x}, t) = \Psi^{(i)}(\mathbf{x}, t)$.

- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, T-t) < 0 \Rightarrow \delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i+1)T} > 0$. By comparing (4.50) with (4.51) we see that state cost $\tilde{q}$ subtracted from $\Phi^{(i+1)}$ is smaller than the state cost q subtracted from $\Phi^{(i)}$ and therefore $\Phi^{(i+1)}(\mathbf{x}, T-t) < \Phi^{(i)}(\mathbf{x}, T-t) \Longrightarrow \Psi^{(i+1)}(\mathbf{x}, t) < \Psi^{(i)}(\mathbf{x}, t)$.

4.6.2 Iterative path integral control with not equal boundary conditions

In this section we deal with the more general case in which the boundary conditions for the PDEs in (4.46) are not necessarily equal. To study the relation between $\Psi^{(i+1)}$ and $\Psi^{(i)}$ we define the function $\Delta\Psi^{(i+1,i)} = \Psi^{(i+1)} - \Psi^{(i)}$. Since the two PDEs in (4.46) are linear we can subtract the PDE in Ψ^i from the PDE in Ψ^{i+1} and we will have:

$$-\partial_t \Delta\Psi_t^{(i+1,i)} = \mathscr{A}^{(i)} \Delta\Psi^{(i+1,i)} + \boldsymbol{\mathcal{F}}(\mathbf{x}, t) \tag{4.52}$$

Now we apply the generalized version of the Feynman-Kac lemma and we represent the solution of the PDE above in a probabilistic manner. More precisely we will have:

$$\begin{aligned}\Delta\Psi^{(i+1,i)}(\mathbf{x}, t) = &\left\langle \Delta\Psi^{(i+1,i)}(\mathbf{x}, t_N) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}_s, s) ds\right)\right\rangle \\ &+ \left\langle \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}, s) ds\right) d\theta \right\rangle\end{aligned}$$

We identify 3 cases:

- Clearly in case $\Delta\Psi^{(i+1,i)}(\mathbf{x}, t_N) = 0$ then, if $\boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) > 0 \Rightarrow \Delta\Psi^{(i+1,i)}(\mathbf{x}, t) > 0 \Rightarrow \Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t)$. This case was discussed in the previous subsection in which we came to the same conclusion that $\boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) > 0$ by using more intuitive arguments.

- If $\Delta\Psi^{(i+1,i)}(\mathbf{x}, t_N) < 0$ then the conditions, for $\Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t)$ to be true, are given as follows:

$$
\begin{aligned}
&- \left\langle \Delta\Psi^{(i+1,i)}(\mathbf{x}, t_N) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}_s, s)ds\right)\right\rangle \\
&< \left\langle \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}, s)ds\right)d\theta\right\rangle
\end{aligned}
$$

  The condition above results in : $\left\langle \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}, s)ds\right)d\theta\right\rangle > 0$ which is a necessary but not sufficient condition.

- If $\Delta\Psi^{(i+1,i)}(\mathbf{x}, t_N) > 0$ then the condition $\left\langle \int_t^T \boldsymbol{\mathcal{F}}(\mathbf{x}_\theta, \theta) \exp\left(-\frac{1}{\lambda}\int_t^T q(\mathbf{x}, s)ds\right)d\theta\right\rangle > 0$ becomes the sufficient condition such that $\Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t)$.

4.7 Risk sensitive path integral control

To arrive in the Path integral control formalism for the risk sensitive setting we make use of (2.117) and (2.118) and the transformation $V(\mathbf{x}, t) = -\lambda \log \Psi(\mathbf{x}, t)$. More precisely we will have that:

$$\frac{\lambda}{\Psi_t}\partial_t\Psi = q - \frac{\lambda}{\Psi}(\nabla_{\mathbf{x}}\Psi)^T\mathbf{f} - \underline{\frac{\lambda^2}{2\Psi^2}(\nabla_{\mathbf{x}}\Psi)^T\boldsymbol{\mathcal{M}}(\mathbf{x})(\nabla_{\mathbf{x}}\Psi)} + \frac{\epsilon}{2\gamma}tr\left(\tilde{\Gamma}\right) \tag{4.53}$$

where the term $\tilde{\Gamma}$ is expressed as:

$$\tilde{\Gamma}(\mathbf{x}) = \lambda\frac{1}{\Psi^2}\nabla_{\mathbf{x}}\Psi\ \nabla_{\mathbf{x}}\Psi^T\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T - \lambda\frac{1}{\Psi}\nabla_{\mathbf{xx}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T \tag{4.54}$$

The tr of $\tilde{\Gamma}$ is therefore:

$$\tilde{\Gamma}(\mathbf{x}) = \underline{\lambda\frac{1}{\Psi^2}tr\left(\nabla_{\mathbf{x}}\Psi^T\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}\nabla_{\mathbf{x}}\Psi\right)} - \lambda\frac{1}{\Psi}tr\left(\nabla_{\mathbf{xx}}\Psi\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) \tag{4.55}$$

Comparing the underlined terms in (4.11) and (4.55), one can recognize that these terms will cancel under the assumption:

$$\lambda\boldsymbol{\mathcal{M}}(\mathbf{x}) = \frac{\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \boldsymbol{\Sigma}(\mathbf{x}_t) = \boldsymbol{\Sigma}_t \tag{4.56}$$

which results in:

$$\lambda\left(\mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T - \frac{1}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\tilde{\mathbf{C}}(\mathbf{x})^T\right) = \frac{\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T \tag{4.57}$$

Again since $\frac{\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T$ is positive definite $\forall\epsilon,\gamma > 0$, we will have that:

$$\lambda\left(\mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T - \frac{1}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\tilde{\mathbf{C}}(\mathbf{x})^T\right) > 0 \tag{4.58}$$

The previous equation can be written in the form:

$$\lambda \mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T = \frac{\lambda+\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T \tag{4.59}$$

With this simplification, (4.11) reduces to the following form:

$$-\partial_t \Psi = -\frac{1}{\lambda}q\Psi_t + \mathbf{f}^T(\nabla_{\mathbf{x}}\Psi) + \frac{\epsilon}{2\gamma}tr\left((\nabla_{\mathbf{xx}}\Psi)\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right) \tag{4.60}$$

with boundary condition: $\Psi_{t_N} = \exp\left(-\frac{1}{\lambda}\phi_{t_N}\right)$. The analysis so far results in the following theorem.

Theorem: *The exponentiated value function* $\Psi(\mathbf{x},t) = \exp\left(-\frac{1}{\lambda}V(\mathbf{x},t)\right)$ *of the risk sensitive stochastic optimal control problem defined by* (2.103),(2.104) *is given by the linear and second order PDE:*

$$-\partial_t \Psi = -\frac{1}{\lambda}q\Psi_t + \mathbf{f}^T(\nabla_{\mathbf{x}}\Psi) + \frac{\epsilon}{2\gamma}tr\left((\nabla_{\mathbf{xx}}\Psi)\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right)$$

with terminal condition $\Psi_{t_N} = \exp\left(-\frac{1}{\lambda}\phi_{t_N}\right)$ *iff the following assumption holds*

$$\lambda \mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T = \frac{\lambda+\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T$$

where the parameters $\epsilon, \gamma, \lambda > 0$ *and* $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{L}\mathbf{L}^T$.

Clearly, a quick inspection of (4.41) and (4.60) leads to the conclusion that the PDEs are identical if $\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T$. Given the last assumption and (4.59) the stochastic differential game formulated by (2.120), (2.94) and stochastic risk sensitive

optimal control problem given by (2.103),(2.104) are equivalent. Essentially, condition (4.59) guarantees that the equivalence between differential games and risk sensitivity in optimal control carries over inside the path integral control formalism.

The theorem that follows is the synopsis of our analysis and it is central in this work since it establishes the connection between risk sensitive control and differential game theory under the path integral control formalism. More precisely:

Theorem: *Consider the stochastic differential game expressed by* (2.120) *and* (2.94) *and the risk sensitive stochastic optimal control problem defined by* (2.103) *and* (2.104). *These optimal control problems are equivalent under the path integral formalism. Their common optimal control solution is expressed by:*

$$\mathbf{u}^*(\mathbf{x}) = \lambda \mathbf{R}^{-1}\mathbf{G}^T \frac{\nabla_{\mathbf{x}}\Psi}{\Psi} \tag{4.61}$$

where

$$-\partial_t \Psi = -\frac{1}{\lambda} q \Psi_t + \mathbf{f}^T(\nabla_{\mathbf{x}}\Psi) + \frac{\epsilon}{2\gamma} tr\left((\nabla_{\mathbf{xx}}\Psi)\tilde{\mathbf{C}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}^T\right)$$

with boundary condition $\Psi_{t_N} = \exp\left(-\frac{1}{\lambda}\phi_{t_N}\right)$, *iff the following conditions hold*

i) $\lambda \mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T = \frac{\lambda+\epsilon}{\gamma}\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T$ and

ii) $\tilde{\mathbf{C}}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\tilde{\mathbf{C}}(\mathbf{x})^T = \mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{C}(\mathbf{x})^T$

with the parameters $\gamma, \lambda > 0$ *and* $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ *defined as* $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{L}\mathbf{L}^T$.

4.8 Appendix

This section contains the derivation for the factorization of the cost function $Z(\boldsymbol{\tau}_i)$, into path dependent and path independent terms, the lemmas **L1** and **L2** and one theorem **T1**. The theorem provides the main result of the generalized path integral control formalism expressed by (4.26), (4.27), (4.28). Its proof is based on results proven in the lemmas **L1** and **L2**.

Derivation of the factorization of $Z(\boldsymbol{\tau}_i)$.

We start our derivation from the equation 4.24. Our goal is to factorize the following quantity into path dependent and path independent terms. More precisely we have:

$$Z(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \lambda \log D(\boldsymbol{\tau}_i) \tag{4.62}$$

$$D(\boldsymbol{\tau}_i) = \prod_{j=i}^{N-1} \left((2\pi)^{l/2} |\boldsymbol{\Sigma}_{t_j}|^{1/2} \right).$$

Since $\boldsymbol{\Sigma}_{t_j} = \mathbf{B}_{t_j}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}_{t_j}^{(c)^T} dt = \lambda \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T} dt = \lambda \mathbf{H}_{t_j} dt$ with $\mathbf{H}_{t_j} = \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T}$

$$\begin{aligned}
Z(\boldsymbol{\tau}_i) &= S(\boldsymbol{\tau}_i) + \lambda \log \prod_{j=i}^{N-1} (2\pi)^{n/2} |\boldsymbol{\Sigma}(\mathbf{x}_{t_i})|^{1/2} \\
&= S(\boldsymbol{\tau}_i) + \lambda \sum_{j=i}^{N-1} \log \left((2\pi)^{n/2} |\mathbf{B}(\mathbf{x}, t_j) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t_j)^T dt|^{1/2} \right) \\
&= S(\boldsymbol{\tau}_i) + \lambda \sum_{j=i}^{N-1} \log \left((2\pi)^{n/2} |\mathbf{B}(\mathbf{x}, t_j) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t_j)^T dt|^{1/2} \right)
\end{aligned}$$

$$
\begin{aligned}
&= S(\boldsymbol{\tau}_i) + \lambda \sum_{j=i}^{N-1} \log\left((2\pi)^{n/2} \left| \mathbf{B}(\mathbf{x}, t_j) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t_j)^T dt \right|^{1/2} \right) \\
&= S(\boldsymbol{\tau}_i) + \lambda \sum_{j=i}^{N-1} \log\left(\left| 2\pi \mathbf{B}(\mathbf{x}, t_j) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t_j)^T dt \right|^{1/2} \right) \\
&= S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} tr \log\left(2\pi \mathbf{B}(\mathbf{x}, t_j) \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{B}(\mathbf{x}, t_j)^T dt \right)
\end{aligned}
$$

Here we will assume just for simplicity that $\boldsymbol{\Sigma}_{\mathbf{w}} = \sigma_w^2 I_{n \times n}$.

$$
\begin{aligned}
Z(\boldsymbol{\tau}_i) &= S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} tr \left[\log\left(2\pi \sigma_w^2 I_{n \times n} dt \right) + \log\left(\mathbf{B}(\mathbf{x}, t_j) \mathbf{B}(\mathbf{x}, t_j)^T \right) \right] \\
&= S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} tr \left[\log\left(2\pi \sigma_w^2 I_{n \times n} dt \right) + \log\left(\mathbf{B}(\mathbf{x}, t_j) \mathbf{B}(\mathbf{x}, t_j)^T \right) \right] \\
&= S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} \left[n \log\left(2\pi \sigma_w^2 dt \right) + tr \log\left(\mathbf{B}(\mathbf{x}, t_j) \mathbf{B}(\mathbf{x}, t_j)^T \right) \right] \\
&= S(\boldsymbol{\tau}_i) + \frac{\lambda N n}{2} \log\left(n 2\pi \sigma_w^2 dt \right) + \frac{\lambda}{2} \sum_{j=i}^{N-1} tr \log\left(\mathbf{B}(\mathbf{x}, t_j) \mathbf{B}(\mathbf{x}, t_j)^T \right) \\
&= S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log \left| \mathbf{B}(\mathbf{x}, t_j) \mathbf{B}(\mathbf{x}, t_j)^T \right| + \frac{\lambda (N-i) n}{2} \log\left(2\pi \sigma_w^2 dt \right)
\end{aligned}
$$

Finally the full cost to go is:

$$
Z(\boldsymbol{\tau}_i) = \tilde{S}(\boldsymbol{\tau}_i) + \frac{\lambda N n}{2} \log\left(2\pi \sigma_w^2 dt \right)
$$

where

$$\tilde{S}\left(\boldsymbol{\tau}_i\right) = S\left(\boldsymbol{\tau}_i\right) + \frac{\lambda}{2}\sum_{i=0}^{N-1}\log\left|\boldsymbol{\mathcal{B}}(\mathbf{x}, t_j)\right|$$

where $\boldsymbol{\mathcal{B}}(\mathbf{x}, t_j) = \mathbf{B}(\mathbf{x}, t_j)\mathbf{B}(\mathbf{x}, t_j)^T$ and

$$S\left(\boldsymbol{\tau}_i\right) = \phi_{t_N} + \sum_{j=i}^{N-1}\left(q_{t_j} + \frac{1}{2}\left\|\frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)}\right\|_{\mathbf{H}_{t_j}}^2\right)dt$$

In cases where $\boldsymbol{\Sigma}_{\mathbf{w}} \neq \sigma_w^2 I_{n\times n}$ the results are the same with the equations besides the term $\boldsymbol{\mathcal{B}}(\mathbf{x}, t_j)$ that is now defined as $\boldsymbol{\mathcal{B}}(\mathbf{x}, t_j) = \mathbf{B}(\mathbf{x}, t_j)\boldsymbol{\Sigma}_{\mathbf{w}}\mathbf{B}(\mathbf{x}, t_j)^T$.

Lemma 1

The optimal control solution to the stochastic optimal control problem expressed by (4.1),(4.2) and (4.3) is formulated as:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left[-\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\int \tilde{p}\left(\boldsymbol{\tau}_i\right)\nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i)d\boldsymbol{\tau}_i\right]$$

where $\tilde{p}\left(\boldsymbol{\tau}_i\right) = \frac{\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)}{\int\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}$ ***is a path dependent probability distribution. The term*** $\tilde{S}(\boldsymbol{\tau}_i)$ ***is a path function defined as*** $\tilde{S}(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \frac{\lambda}{2}\sum_{j=i}^{N-1}\log\left|\boldsymbol{\mathcal{B}}(\mathbf{x}, t_j)\right|$ ***that satisfies the following condition*** $\lim_{dt\to 0}\int\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i \in \boldsymbol{\mathcal{C}}^{(1)}$ ***for any sampled trajectory starting from state*** $\mathbf{x}_{t_i}$***. Moreover the term*** $\mathbf{H}_{t_j}$ ***is given by*** $\mathbf{H}_{t_j} = \mathbf{G}_{t_j}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_j}^{(c)T}$ ***while the term*** $S(\boldsymbol{\tau}_i)$ ***is defined according to***

$$S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j}dt + \frac{1}{2}\sum_{j=i}^{N-1}\left\|\frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)}\right\|_{\mathbf{H}_{t_j}}^2 dt.$$

Proof:

The optimal controls at the state $\mathbf{x}_{t_i}$ is expressed by the equation $\mathbf{u}_{t_i} = -\mathbf{R}^{-1}\mathbf{G}_{t_i}\nabla_{\mathbf{x}_{t_i}} V_{t_i}$. Due to the exponential transformation of the value function $\Psi_{t_i} = -\lambda \log V_{t_i}$ the equation of the optimal controls is written as:

$$\mathbf{u}_{t_i} = \lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}\frac{\nabla_{\mathbf{x}_{t_i}}\Psi_{t_i}}{\Psi_{t_i}}.$$

In discrete time the optimal control is expressed as follows:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T\frac{\nabla_{\mathbf{x}_{t_i}}\Psi_{t_i}^{(dt)}}{\Psi_{t_i}^{(dt)}}\right).$$

By using equation (4.24) and substituting $\Psi^{(dt)}(\mathbf{x}_{t_i}, t)$ we have:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T\frac{\nabla_{\mathbf{x}_{t_i}}\int \exp\left(-\frac{1}{\lambda}Z(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}{\int \exp\left(-\frac{1}{\lambda}Z(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}\right).$$

Substitution of the term $Z(\boldsymbol{\tau}_i)$ results in the equation:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T\frac{\nabla_{\mathbf{x}_{t_i}}\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i) - \frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)d\boldsymbol{\tau}_i}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i) - \frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)d\boldsymbol{\tau}_i}\right).$$

Next we are using standard properties of the exponential function that lead to:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T\frac{\nabla_{\mathbf{x}_{t_i}}\left[\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)\exp\left(-\frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)d\boldsymbol{\tau}_i\right]}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)\exp\left(-\frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)d\boldsymbol{\tau}_i}\right).$$

The term $\exp\left(-\frac{\lambda N l}{2}\log\left(2\pi dt\lambda\right)\right)$ does not depend on the trajectory $\boldsymbol{\tau}_i$, therefore it can be taken outside the integral as well as outside the gradient. Thus we will have that:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T \frac{\exp\left(-\frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)\nabla_{\mathbf{x}_{t_i}}\left[\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i\right]}{\exp\left(-\frac{\lambda(N-i)l}{2}\log\left(2\pi dt\lambda\right)\right)\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}\right).$$

The constant term drops from the nominator and denominator and thus we can write:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left(\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T \left[\frac{\nabla_{\mathbf{x}_{t_i}}\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}\right]\right).$$

Under the assumption that term $\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i$ is continuously differentiable in $\mathbf{x}_{t_i}$ and dt we can change order of the integral with the differentiation operations. In general for $\nabla_x \int f(x,y)dy = \int \nabla_x f(x,y)dy$ to be true, $f(x,t)$ should be continuous in y and differentiable in x. Under this assumption, the optimal controls can be further formulated as:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left[\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T \frac{\int \nabla_{\mathbf{x}_{t_i}} \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}\right].$$

Application of the differentiation rule of the exponent results in:

$$\mathbf{u}_{t_i} = \lim_{dt\to 0}\left[\lambda \mathbf{R}^{-1}\mathbf{G}_{t_i}^T \frac{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)\nabla_{\mathbf{x}_{t_i}}\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)d\boldsymbol{\tau}_i}\right].$$

The denominator is a function of $\mathbf{x}_{t_i}$ the current state and thus it can be pushed inside the integral of the nominator:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left[\lambda \mathbf{R}^{-1} \mathbf{G}_{t_i}^T \int \frac{\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i} \nabla_{\mathbf{x}_{t_i}} \left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i) \right) d\boldsymbol{\tau}_i \right].$$

By defining the probability $\tilde{p}\left(\boldsymbol{\tau}_i\right) = \frac{\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i}$ the expression above can be written as:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left[\lambda \mathbf{R}^{-1} \mathbf{G}_{t_i}^T \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \nabla_{\mathbf{x}_{t_i}} \left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i) \right) d\boldsymbol{\tau}_i \right].$$

Further simplification will result in:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left[-\mathbf{R}^{-1} \mathbf{G}_{t_i}^T \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \nabla_{\mathbf{x}_{t_i}} \tilde{S}(\boldsymbol{\tau}_i) d\boldsymbol{\tau}_i \right].$$

We know that the control transition matrix has the form $\mathbf{G}(\mathbf{x}_{t_i})^T = [0^T \quad \mathbf{G}_c(\mathbf{x}_{x_{t_i}})^T]$. In addition the partial derivative $\nabla_{\mathbf{x}_{t_i}} \tilde{S}(\boldsymbol{\tau}_i)$ can be written as:

$$\nabla_{\mathbf{x}_{t_i}} \tilde{S}(\boldsymbol{\tau}_i)^T = [\nabla_{\mathbf{x}_{t_i}^{(m)}} \tilde{S}(\boldsymbol{\tau}_i)^T \quad \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i)^T].$$

By using these equations we will have that:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left(-\mathbf{R}^{-1} [0^T \quad \mathbf{G}_{t_i}^{(c)T}] \int \tilde{p}\left(\boldsymbol{\tau}_o\right) \begin{bmatrix} \nabla_{\mathbf{x}_{t_i}^{(m)}} \tilde{S}(\boldsymbol{\tau}_i) \\ \\ \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) \end{bmatrix} d\boldsymbol{\tau}_i \right).$$

The equation above can be written in the form:

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left(-[0^T \quad \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T}] \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \begin{bmatrix} \nabla_{\mathbf{x}_{t_i}^{(m)}} \tilde{S}(\boldsymbol{\tau}_i) \\ \\ \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) \end{bmatrix} d\boldsymbol{\tau}_i \right).$$

or

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left(-[0^T \quad \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}] \begin{bmatrix} \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \cdot \nabla_{\mathbf{x}_{t_i}^{(m)}} \tilde{S}(\boldsymbol{\tau}_i) d\boldsymbol{\tau}_i \\ \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \cdot \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) d\boldsymbol{\tau}_i \end{bmatrix} \right).$$

Therefore we will have the result

$$\mathbf{u}_{t_i} = \lim_{dt \to 0} \left[-\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T} \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) d\boldsymbol{\tau}_i \right].$$

Lemma 2

Given the stochastic dynamics and the cost in(4.1),(4.2) and(4.3) the gradient of the path function $\tilde{S}(\boldsymbol{\tau}_i)$ ***in (4.25), with respect to the directly actuated part of the state*** $\mathbf{x}_{t_i}^{(c)}$ ***is formulated as:***

$$\nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) = \frac{1}{2dt} \boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1} \right) \boldsymbol{\alpha}_{t_i} - \mathbf{H}_{t_i}^{-1} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)} \right) \boldsymbol{\alpha}_{t_i} - \frac{1}{dt} \mathbf{H}_{t_i}^{-1} \boldsymbol{\alpha}_{t_i} + \frac{\lambda}{2} \nabla_{\mathbf{x}_{t_i}^{(c)}} \log |\boldsymbol{\mathcal{B}}_{t_i}|$$

where $\mathbf{H}_{t_i} = \mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)^T}$ ***and*** $\boldsymbol{\alpha}_{t_j} = \left(\mathbf{x}_{t_{i+1}}^{(c)} - \mathbf{x}_{t_i}^{(c)} - \mathbf{f}_{t_i}^{(c)} dt \right)$***.***

Proof:

We are calculating the term $\nabla_{\mathbf{x}_{t_o}^{(c)}} \tilde{S}(\boldsymbol{\tau}_o)$. More precisely we have shown that

$$\tilde{S}(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt + \frac{1}{2} \sum_{j=i}^{N-1} \| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \|_{\mathbf{H}_{t_j}}^2 dt + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log |\boldsymbol{\mathcal{B}}_{t_j}|.$$

To limit the length of our derivation we introduce the notation $\gamma_{t_j} = \boldsymbol{\alpha}_{t_j}^T \, \mathbf{h}_{t_j}^{-1} \, \boldsymbol{\alpha}_{t_j}$ and $\boldsymbol{\alpha}_{t_j} = \left(\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)} - \mathbf{f}_{t_j}^{(c)} dt \right)$ and it is easy to show that $\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \|_{\mathbf{H}_{t_j}}^2 dt = \frac{1}{dt} \gamma_{t_j}$ and therefore we will have:

$$\tilde{S}(\boldsymbol{\tau}_i) = \phi_{t_N} + \frac{1}{2dt} \sum_{j=i}^{N-1} \gamma_{t_j} + \sum_{t_o}^{t_N} Q_{t_j} dt + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log |\boldsymbol{\mathcal{B}}_{t_j}|.$$

In the analysis that follows we provide the derivative of the 1th, 2th and 4th term of the cost function. We assume that the cost of the state during the time horizon $Q_{t_i} = 0$. In cases that this is not true then the derivative $\nabla_{\mathbf{x}_{t_i}^{(c)}} \sum_{t_i}^{t_N} Q_{t_i} dt$ needs to be found as well. By calculating the term $\nabla_{\mathbf{x}_{t_o}^{(c)}} \tilde{S}(\boldsymbol{\tau}_o)$ we can find the local controls $\mathbf{u}(\boldsymbol{\tau}_i)$. It is important to mention that the derivative of the path cost $S(\boldsymbol{\tau}_i)$ is taken only with respect to the current state $\mathbf{x}_{t_o}$.

The first term is:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}} \left(\phi_{t_N} \right) = 0. \tag{4.63}$$

Derivative of the 2th Term $\nabla_{\mathbf{x}_{t_i}^{(c)}} \left[\frac{1}{2dt} \sum_{i=1}^{N-1} \gamma_{t_i} \right]$ of the cost $S(\boldsymbol{\tau}_i)$.

The second term can be found as follows:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}} \left[\frac{1}{2dt} \sum_{j=i}^{N-1} \gamma_{t_j} \right].$$

The operator $\nabla_{\mathbf{x}_{t_o}^{(c)}}$ is linear and it can massaged inside the sum:

$$\frac{1}{2dt} \sum_{j=i}^{N-1} \nabla_{\mathbf{x}_{t_j}^{(c)}} \left(\gamma_{t_j} \right).$$

Terms that do not depend on $\mathbf{x}_{t_i}^{(c)}$ drop and thus we will have:

$$\frac{1}{2dt}\nabla_{\mathbf{x}_{t_i}^{(c)}}\gamma_{t_i}.$$

Substitution of the parameter $\gamma_{t_i} = \boldsymbol{\alpha}_{t_i}^T \ \mathbf{H}_{t_i}^{-1} \ \boldsymbol{\alpha}_{t_i}$ will result in:

$$\frac{1}{2dt}\nabla_{\mathbf{x}_{t_i}^{(c)}}\left[\boldsymbol{\alpha}_{t_i}^T \ \mathbf{H}_{t_i}^{-1} \ \boldsymbol{\alpha}_{t_i}\right].$$

By making the substitution $\boldsymbol{\beta}_{t_i} = \mathbf{H}_{t_i}^{-1} \ \boldsymbol{\alpha}_{t_i}$ and applying the rule $\nabla\left(\mathbf{u}(\mathbf{x})^T\mathbf{v}(\mathbf{x})\right) = \nabla\left(\mathbf{u}(\mathbf{x})\right)\mathbf{v}(\mathbf{x}) + \nabla\left(\mathbf{v}(\mathbf{x})\right)\mathbf{u}(\mathbf{x})$ we will have that:

$$\frac{1}{2dt}\left[\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\alpha}_{t_i} \ \ \boldsymbol{\beta}_{t_i} + \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i} \ \ \boldsymbol{\alpha}_{t_i}\right]. \tag{4.64}$$

Next we find the derivative of $\boldsymbol{\alpha}_{t_o}$:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\alpha}_{t_i} = \nabla_{\mathbf{x}_{t_i}^{(c)}}\left[\mathbf{x}_{t_{i+1}}^{(c)} - \mathbf{x}_{t_i}^{(c)} - \mathbf{f}_c(\mathbf{x}_{t_i})dt\right].$$

and the result is

$$\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\alpha}_{t_i} = -I_{l\times l} - \nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}dt.$$

We substitute back to (4.64) and we will have:

$$\frac{1}{2dt}\left[-\left(I_{l\times l} + \nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}dt\right) \ \ \boldsymbol{\beta}_{t_i} + \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i} \ \ \boldsymbol{\alpha}_{t_i}\right].$$

$$-\frac{1}{2dt}\left(I_{l\times l}+\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}dt\right)\ \ \boldsymbol{\beta}_{t_i}+\frac{1}{2dt}\ \ \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \boldsymbol{\alpha}_{t_i}.$$

After some algebra the result of $\nabla_{\mathbf{x}_{t_i}^{(c)}}\left(\frac{1}{2dt}\sum_{i=1}^{N-1}\gamma_{t_i}\right)$ is expressed as:

$$-\frac{1}{2dt}\ \ \boldsymbol{\beta}_{t_i}-\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}\ \ \boldsymbol{\beta}_{t_i}+\frac{1}{2dt}\ \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \ \boldsymbol{\alpha}_{t_i}.$$

We continue with further analysis of each one of the terms in the expression above. More precisely we will have:

First Subterm: $-\frac{1}{2dt}\ \boldsymbol{\beta}_{t_i}$

$$\begin{aligned}\left(-\frac{1}{2dt}\ \ \boldsymbol{\beta}_{t_i}\right)&=-\left(\frac{1}{2dt}\ \ \mathbf{H}_{t_i}^{-1}\ \ \boldsymbol{\alpha}_{t_i}\right)\\&=-\frac{1}{2}\ \ \mathbf{H}_{t_i}^{-1}\ \ \boldsymbol{\alpha}_{t_i}\\&=-\frac{1}{2}\ \ \mathbf{H}_{t_i}^{-1}\ \ \left((\mathbf{x}_{t_{i+1}}^{(c)}-\mathbf{x}_{t_i}^{(c)})\frac{1}{dt}-\mathbf{f}_{t_i}^{(c)}\right).\end{aligned}$$

Second Subterm: $-\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}\ \ \boldsymbol{\beta}_{t_i}$

$$\begin{aligned}\left(\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}\ \ \boldsymbol{\beta}_{t_i}\right)&=-\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_c(\mathbf{x}_{t_i})\ \ \boldsymbol{\beta}_{t_i}=-\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_{t_i}^{(c)}\ \ \left(\mathbf{H}_{t_i}^{-1}\ \ \boldsymbol{\alpha}_{t_i}\right)\\&=-\frac{1}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\mathbf{f}_c(\mathbf{x}_{t_i})\ \ \mathbf{H}_{t_i}^{-1}\ \ \boldsymbol{\alpha}_{t_i}\end{aligned}$$

Third Subterm: $\frac{1}{2dt}\ \ \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \ \boldsymbol{\alpha}_{t_i}$

$$\left(\frac{1}{2dt}\ \ \nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \ \boldsymbol{\alpha}_{t_i}\right)=\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \ \left(\frac{1}{2dt}\ \ \boldsymbol{\alpha}_{t_i}\right)=\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i}\ \ \frac{1}{2}\left((\mathbf{x}_{t_{i+1}}^{(c)}-\mathbf{x}_{t_i}^{(c)})\frac{1}{dt}-\mathbf{f}_{t_i}^{(c)}\right).$$

We substitute $\boldsymbol{\beta}_{t_i} = \mathbf{H}_{t_i}^{-1} \quad \boldsymbol{\alpha}_{t_i}$ and write the matrix $\mathbf{H}_{t_i}^{-1}$ in row form:

$$= \nabla_{\mathbf{x}_{t_i}^{(c)}} \left(\mathbf{H}_{t_i}^{-1} \quad \boldsymbol{\alpha}_{t_i} \right) \quad \frac{1}{2dt} \boldsymbol{\alpha}_{t_i} =$$

$$= \quad \nabla_{\mathbf{x}_{t_i}^{(c)}} \left(\begin{bmatrix} \mathbf{H}_{t_i}^{(1)^{-T}} \\ \mathbf{H}_{t_i}^{(2)^{-T}} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{H}_{t_i}^{(l)^{-T}} \end{bmatrix} \quad \boldsymbol{\alpha}_{t_i} \right) \quad \frac{1}{2dt} \boldsymbol{\alpha}_{t_i} = \nabla_{\mathbf{x}_{t_i}^{(c)}} \begin{bmatrix} \mathbf{H}_{t_i}^{(1)^{-T}} & \boldsymbol{\alpha}_{t_i} \\ \mathbf{H}_{t_i}^{(2)^{-T}} & \boldsymbol{\alpha}_{t_i} \\ \cdot & \\ \cdot & \\ \cdot & \\ \mathbf{H}_{t_i}^{(l)^{-T}} & \boldsymbol{\alpha}_{t_i} \end{bmatrix} \frac{1}{2dt} \boldsymbol{\alpha}_{t_i}.$$

We can push the operator $\nabla_{\mathbf{x}_{t_i}^{(c)}}$ insight the matrix and apply it to each element.

$$= \begin{bmatrix} \nabla_{\mathbf{x}_{t_i}^{(c)}}^T \left(\mathbf{H}_{t_i}^{(1)^{-T}} \quad \boldsymbol{\alpha}_{t_i} \right) \\ \nabla_{\mathbf{x}_{t_i}^{(c)}}^T \left(\mathbf{H}_{t_i}^{(2)^{-T}} \quad \boldsymbol{\alpha}_{t_i} \right) \\ \cdot \\ \cdot \\ \cdot \\ \nabla_{\mathbf{x}_{t_i}^{(c)}}^T \left(\mathbf{H}_{t_i}^{(l)^{-T}} \quad \boldsymbol{\alpha}_{t_i} \right) \end{bmatrix} \quad \frac{1}{2dt} \boldsymbol{\alpha}_{t_i}.$$

We again use the rule $\nabla \left(\mathbf{u}(\mathbf{x})^T \mathbf{v}(\mathbf{x}) \right) = \nabla \left(\mathbf{u}(\mathbf{x}) \right) \mathbf{v}(\mathbf{x}) + \nabla \left(\mathbf{v}(\mathbf{x}) \right) \mathbf{u}(\mathbf{x})$ and thus we will have:

$$= \begin{bmatrix} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(1)^{-T}} \quad \boldsymbol{\alpha}_{t_i} + \nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i} \quad \mathbf{H}_{t_i}^{(1)^{-T}}\right)^T \\ \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(2)^{-T}} \quad \boldsymbol{\alpha}_{t_i} + \nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i} \quad \mathbf{H}_{t_i}^{(2)^{-T}}\right)^T \\ \cdot \\ \cdot \\ \cdot \\ \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(l)^{-T}} \quad \boldsymbol{\alpha}_{t_i} + \nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i} \quad \mathbf{H}_{t_i}^{(l)^{-T}}\right)^T \end{bmatrix} \frac{1}{2dt} \boldsymbol{\alpha}_{t_i}.$$

We can split the matrix above into two terms and then we pull out the terms $\boldsymbol{\alpha}_{t_i}$ and $\nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i}$ respectively :

$$\begin{aligned} &= \left(\boldsymbol{\alpha}_{t_i}^T \begin{bmatrix} \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(1)^{-T}} \\ \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(2)^{-T}} \\ \cdot \\ \cdot \\ \cdot \\ \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{(l)^{-T}} \end{bmatrix} + \begin{bmatrix} \mathbf{H}_{t_i}^{(1)^{-T}} \\ \mathbf{H}_{t_i}^{(2)^{-T}} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{H}_{t_i}^{(l)^{-T}} \end{bmatrix} \quad \nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i}^T \right) \quad \frac{1}{2dt} \boldsymbol{\alpha}_{t_i} \\ &= \left(\boldsymbol{\alpha}_{t_i}^T \quad \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1} + \mathbf{H}_{t_i}^{-1} \quad \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i}^T\right) \right) \frac{1}{2dt} \boldsymbol{\alpha}_{t_i}. \\ &= \frac{1}{2dt} \left(\boldsymbol{\alpha}_{t_i}^T \quad \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1} \boldsymbol{\alpha}_{t_i} + \mathbf{H}_{t_i}^{-1} \quad \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i}^T\right) \boldsymbol{\alpha}_{t_i}. \right) \end{aligned}$$

Since $\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\alpha}_{t_i}^T\right) = -I_{l\times l} - \nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)} dt$. the final result is expressed as follows

$$\frac{1}{2dt}\nabla_{\mathbf{x}_{t_i}^{(c)}}\boldsymbol{\beta}_{t_i} \;\; \boldsymbol{\alpha}_{t_i} = \frac{1}{2dt}\left[\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} - \mathbf{H}_{t_i}^{-1}\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)}\right) dt\boldsymbol{\alpha}_{t_i} - \mathbf{H}_{t_i}^{-1}\boldsymbol{\alpha}_{t_i}\right]$$

After we have calculated the 3 sub-terms, the 2th term of the of the derivative of path cost $S(\boldsymbol{\tau}_i)$ can be expressed in the following form:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}}\left(\frac{1}{2dt}\sum_{j=i}^{N-1}\gamma_{t_j}\right) = \frac{1}{2dt}\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} - \mathbf{H}_{t_i}^{-1}\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)}\right)\boldsymbol{\alpha}_{t_i} - \frac{1}{dt}\mathbf{H}_{t_i}^{-1}\boldsymbol{\alpha}_{t_i} \tag{4.65}$$

Next we will find the derivative of the term$\nabla_{\mathbf{x}_{t_i}^{(c)}}\left(\frac{\lambda}{2}\sum_{j=i}^{N-1}\log|\boldsymbol{\mathcal{B}}_{t_j}|\right)$.

Derivative of the Fourth Term $\nabla_{\mathbf{x}_{t_i}^{(c)}}\left(\frac{\lambda}{2}\sum_{j=i}^{N-1}\log|\boldsymbol{\mathcal{B}}_{t_j}|\right)$ of the cost $S(\boldsymbol{\tau}_i)$.

The analysis for the 4th term is given below:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}}\left(\frac{\lambda}{2}\sum_{j=i}^{N-1}\log|\boldsymbol{\mathcal{B}}_{t_j}|\right) = \frac{\lambda}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\log|\boldsymbol{\mathcal{B}}_{t_i}|. \tag{4.66}$$

After having calculated all the derivatives of $\tilde{S}(\boldsymbol{\tau}_i)$ the final result under (4.63),(4.65) and (4.66) takes the form:

$$\nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i) = \frac{1}{2dt}\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} - \mathbf{H}_{t_i}^{-1}\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)}\right)\boldsymbol{\alpha}_{t_i} - \frac{1}{dt}\mathbf{H}_{t_i}^{-1}\boldsymbol{\alpha}_{t_i} + \frac{\lambda}{2}\nabla_{\mathbf{x}_{t_i}^{(c)}}\log|\boldsymbol{\mathcal{B}}_{t_i}|.$$

Theorem

The optimal control solution to the stochastic optimal control problem expressed by (4.1),(4.2),(4.3) is formulated by the equation that follows:

$$\mathbf{u}_{t_i} dt = \int \tilde{p}\left(\boldsymbol{\tau}_i\right) \ \mathbf{u}_L\left(\boldsymbol{\tau}_i\right) \ d\boldsymbol{\tau}_i,$$

where $\tilde{p}\left(\boldsymbol{\tau}_i\right) = \frac{\exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right)}{\int \exp\left(-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i}$ ***is a path depended probability distribution and the term*** $\mathbf{u}\left(\boldsymbol{\tau}_i\right)$ ***defined as*** $\mathbf{u}_L\left(\boldsymbol{\tau}_i\right) = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\left(\mathbf{G}_{t_i}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\right)^{-1}\mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i}$***, are the local controls of each sampled trajectory starting from state*** $\mathbf{x}_{t_i}$***. The term is defined as*** $\mathbf{H}_{t_i} = \mathbf{G}_{t_i}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)^T}$ ***.***

To prove the theorem we make use of the lemma **L2** and we substitute $\nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i)$ in the main result of lemma **L1**. More precisely from lemma **L1** we have that:

$$\mathbf{u}_{t_i} dt = -\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T} dt \int \tilde{p}\left(\boldsymbol{\tau}_i\right)\left(\nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i.$$

$$\begin{aligned}\mathbf{u}_{t_i} dt &= -\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T} dt \int \tilde{p}\left(\boldsymbol{\tau}_i\right)\left(\nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i \\ &= \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\left\langle \nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i) dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}\end{aligned} \tag{4.67}$$

Now we will find the term $\left\langle \nabla_{\mathbf{x}_{t_i}^{(c)}}\tilde{S}(\boldsymbol{\tau}_i) dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}$. More precisely we will have that:

$$\left\langle \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i)dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} = \left\langle \frac{1}{2}\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} - \left\langle \mathbf{H}_{t_i}^{-1} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)}\right) \boldsymbol{\alpha}_{t_i} dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}$$
$$- \left\langle \mathbf{H}_{t_i}^{-1} \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} + \left\langle \frac{\lambda}{2} \nabla_{\mathbf{x}_{t_i}^{(c)}} \log |\boldsymbol{\mathcal{B}}_{t_i}| dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}$$

The first term of the expectation above is calculated as follows:

$$\begin{aligned}
\left\langle \frac{1}{2dt}\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} &= \left\langle \frac{1}{2} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} \boldsymbol{\alpha}_{t_i}^T \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \\
&= \frac{1}{2dt} \left\langle tr\left(\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} \boldsymbol{\alpha}_{t_i}^T \right) \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \\
&= \frac{1}{2dt} tr\left[\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \left\langle \boldsymbol{\alpha}_{t_i} \boldsymbol{\alpha}_{t_i}^T \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \right]
\end{aligned}$$

By taking into account the fact that $\left\langle \boldsymbol{\alpha}_{t_i} \boldsymbol{\alpha}_{t_i}^T \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} = \mathbf{B}_{t_i}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} {\mathbf{B}_{t_i}^{(c)}}^T dt$

$$\begin{aligned}
\left\langle \frac{1}{2dt}\boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} &= \frac{1}{2} tr\left(\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \mathbf{B}_{t_i}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} {\mathbf{B}_{t_i}^{(c)}}^T \right) \\
&= \frac{dt}{2} tr\left(\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1}\right) \mathbf{B}_{t_i}^{(c)} \boldsymbol{\Sigma}_{\mathbf{w}} {\mathbf{B}_{t_i}^{(c)}}^T \right)
\end{aligned}$$

By using the fact that the noise and the controls are related via $\mathbf{\Sigma}_{t_j} = \mathbf{B}_{t_j}^{(c)} \mathbf{\Sigma}_{\mathbf{w}} \mathbf{B}_{t_j}^{(c)^T} dt = \lambda \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T} dt = \lambda \mathbf{H}_{t_j} dt$ with $\mathbf{H}_{t_j} = \mathbf{G}_{t_j}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_j}^{(c)^T}$ we will have:

$$\begin{aligned}
\left\langle \frac{1}{2} \boldsymbol{\alpha}_{t_i}^T \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1} \right) \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} &= \frac{1}{2} tr \left(\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{H}_{t_i}^{-1} \right) \mathbf{B}_{t_i}^{(c)} \mathbf{\Sigma}_{\mathbf{w}} \mathbf{B}_{t_i}^{(c)^T} \right) \\
&= \frac{\lambda}{2} tr \left(\left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \boldsymbol{\mathcal{B}}(\mathbf{x}_{t_i})^{-1} \right) \boldsymbol{\mathcal{B}}(\mathbf{x}_{t_i}) \right) \\
&= \frac{\lambda}{2} \nabla_{\mathbf{x}_{t_i}^{(c)}} \log |\boldsymbol{\mathcal{B}}(\mathbf{x}_{t_i})|^{-1} \\
&= -\frac{\lambda}{2} \nabla_{\mathbf{x}_{t_i}^{(c)}} \log |\boldsymbol{\mathcal{B}}(\mathbf{x}_{t_i})|
\end{aligned}$$

The second term $\left\langle \mathbf{H}_{t_i}^{-1} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)} \right) \boldsymbol{\alpha}_{t_i} dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} = 0$ since $dt \boldsymbol{\alpha}_{t_i} = dt \mathbf{G}_{t_i}^{(c)} d\mathbf{w} \to 0$. We the equation above we will have that:

$$\begin{aligned}
\left\langle \nabla_{\mathbf{x}_{t_i}^{(c)}} \tilde{S}(\boldsymbol{\tau}_i) dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} &= -\left\langle \mathbf{H}_{t_i}^{-1} \left(\nabla_{\mathbf{x}_{t_i}^{(c)}} \mathbf{f}_{t_i}^{(c)} \right) \boldsymbol{\alpha}_{t_i} dt \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} - \left\langle \mathbf{H}_{t_i}^{-1} \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \\
&= -\left\langle \mathbf{H}_{t_i}^{-1} \boldsymbol{\alpha}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \\
&= -\left\langle \mathbf{H}_{t_i}^{-1} \mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i} \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}
\end{aligned}$$

Substituting back to the optimal control we will have that:

$$\mathbf{u}_{t_i} dt = \int \tilde{p}(\boldsymbol{\tau}_i) \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \mathbf{H}_{t_i}^{-1} \mathbf{G}_{t_i}^{(c)} d\mathbf{w}_{t_i} d\boldsymbol{\tau}_i, \tag{4.68}$$

or in a more compact form:

$$\boxed{\mathbf{u}_{t_i} dt = \int \tilde{p}(\boldsymbol{\tau}_i) \ \mathbf{u}_L^{(dt)}(\boldsymbol{\tau}_i) \ d\boldsymbol{\tau}_i,} \tag{4.69}$$

where the local controls $\mathbf{u}_L^{(dt)}(\boldsymbol{\tau}_i)$ are given as follows:

$$\mathbf{u}_L^{(dt)}(\boldsymbol{\tau}_i) = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\mathbf{H}_{t_i}^{-1}\mathbf{G}_{t_i}^{(c)}d\mathbf{w}_{t_i}$$

The local control above can be written in the form:

$$\boxed{\mathbf{u}_L = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\left(\mathbf{G}_{t_i}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\right)^{-1}\mathbf{G}_c d\mathbf{w}_{t_i}.}$$

Therefore the optimal control can now be expressed in the form:

$$\boxed{\mathbf{u}\left(\boldsymbol{\tau}_i\right)dt = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\left(\mathbf{G}_{t_i}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\right)^{-1}\sum_{k=1}^{K}\tilde{p}^{(k)}\left(\boldsymbol{\tau}_i\right)\mathbf{G}_{t_i}^{(c)}d\mathbf{w}_{t_i}^{(k)}} \tag{4.70}$$

Chapter 5

Policy Gradient Methods

In this chapter we are discussing the Policy Gradient (PG) methods which are classified as part of model free reinforcement learning. Our goal is to provide a quick introduction to PGs and review the main assumptions and mathematical tricks and their derivations. Our discussion starts in section 5.1 with the presentation of one of the most simple and widely used PG methods, the so called finite difference algorithm. We continue in section 5.2 with the derivation of the Episodic Reinforce PG method. Our derivation consist of the computation of the PG, the computation of the optimal baseline necessary for reducing the variance of the estimate gradient. In section 5.3 the policy gradient theorem is presented with the derivation of the corresponding gradient and the time optimal baseline. In section 5.4 the concept of the Natural Gradient is presented and its application to reinforcement learning problems is discussed. The resulting algorithm Natural Actor Critic is derived. In the last section we conclude with observations and comments regarding the performance of PG methods.

5.1 Finite difference

In the Finite Difference(FD) method the goal is to optimize a cost function w.r.t. a parameter vector $\boldsymbol{\theta} \in \Re^{p\times 1}$. In reinforcement learning scenarios this parameter vector is used to parametrized the policy. The optimization problem is stated as follows:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

As in all policy gradient algorithms, in FD methods the gradient is estimated $\nabla_{\boldsymbol{\theta}} J$ and the parameter estimates are updated according to the rule $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \nabla_{\boldsymbol{\theta}} J$. To find the gradient, a number of perturbations of the parameters $\delta\boldsymbol{\theta}$ are performed and the Taylor series expansions of the cost function is computed. More precisely we will have:

$$J_i(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \delta\boldsymbol{\theta}_i) = J(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} J^T \delta\boldsymbol{\theta}_i + \boldsymbol{\mathcal{O}}(\delta\theta_{i,j}^2) \quad \forall\ i = 1, 2, ..., M$$

By putting all these equations above for $i = 1, 2, ..., M$ together we will have that:

$$\begin{pmatrix} \Delta J_1(\boldsymbol{\theta}) \\ \\ \\ \Delta J_M(\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} \delta\boldsymbol{\theta}_1^T \\ \\ \\ \delta\boldsymbol{\theta}_M^T \end{pmatrix} \nabla_{\boldsymbol{\theta}} J$$

The equation can be solved with respect to $\nabla_{\boldsymbol{\theta}} J$. More precisely we will have:

$$\nabla_{\boldsymbol{\theta}} J = \left(\Delta\boldsymbol{\Theta}^T \Delta\boldsymbol{\Theta}\right)^{-1} \Delta\boldsymbol{\Theta}^T \Delta\mathbf{J}$$

where $\Delta\boldsymbol{\Theta}^T = (\delta\boldsymbol{\theta}_1, ..., \delta\boldsymbol{\theta}_M) \in \Re^{N\times M}$ and $\Delta\mathbf{J}^T = (\Delta J_1(\boldsymbol{\theta}), ..., \Delta J_1(\boldsymbol{\theta})) \in \Re^{1\times M}$. The estimation of the gradient vector $\nabla_{\boldsymbol{\theta}} J$ requires that the matrix $\Delta\boldsymbol{\Theta}^T\Delta\boldsymbol{\Theta}$ is full rank and therefore invertible.

5.2 Episodic reinforce

(Williams 1992) introduced the episodic REINFORCE algorithm, which is derived from taking the derivative of a cost with respect to the policy parameters. This algorithm has rather slow convergence due to a very noisy estimate of the policy gradient. It is also very sensitive to a reward baseline parameter b_k (see below). Recent work derived the optimal baseline for REINFORCE (cf. (Peters & Schaal 2008*c*)), which improved the performance significantly.

We derive of episodic REINFORCE algorithm by mathematically expressing the cost function under optimization as follows:

$$\boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \int p(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau} \tag{5.1}$$

where $p(\boldsymbol{\tau})$ is the probability of the trajectory $\boldsymbol{\tau} = (\mathbf{x}_0, \mathbf{u}_0 ... \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N)$ of states and controls with $\mathbf{x} \in \Re^{n\times 1}$ and $\mathbf{u} \in \Re^{p\times 1}$ and $R(\boldsymbol{\tau}) = \sum_{t=1}^{N} r(\mathbf{x}_t, \mathbf{u}_t)$ is the cost accumulated over the horizon $T = Ndt$. Due to the Markov property and the dependence of the policy to the state and parameter $\mathbf{x}, \boldsymbol{\theta}$ we will have the following expression for the probability of the trajectory $p(\boldsymbol{\tau})$:

$$p(\boldsymbol{\tau}) = p\left(\mathbf{x}_0\right) \prod_{i=1}^{N-1} p\left(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i\right) p\left(\mathbf{u}_i|\mathbf{x}_i; \boldsymbol{\theta}\right) \tag{5.2}$$

The probability of the trajectory is expressed as the product of the transition probabilities in $p\left(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i\right)$ and the parametrized policy $p\left(\mathbf{u}_i|\mathbf{x}_i; \boldsymbol{\theta}\right)$ where $\boldsymbol{\theta} \in \Re^{q\times 1}$ is the parameter under learning. We would like to find the gradient of $\boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u})$ w.r.t the parameter $\boldsymbol{\theta}$. More precisely we will have that:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) &= \nabla_{\boldsymbol{\theta}} \left(\int p(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau} \right) \\
&= \int \nabla_{\boldsymbol{\theta}} p(\boldsymbol{\tau}) \; R(\boldsymbol{\tau}) d\boldsymbol{\tau} \\
&= \int p(\boldsymbol{\tau}) \; \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}) \; R(\boldsymbol{\tau}) d\boldsymbol{\tau} \\
&= \left\langle \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}) \; R(\boldsymbol{\tau}) \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)}
\end{aligned}$$

where the $\left\langle \right\rangle_{p(\boldsymbol{\tau})}$ is the expectation under the probability metric $p\left(\boldsymbol{\tau}\right)$. The next step is to calculate the term $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau})$.

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}) &= \nabla_{\boldsymbol{\theta}} \left(\log p(\mathbf{x}_0) + \sum_{i=1}^{N-1} \log p\left(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i\right) + \sum_{i=1}^{N-1} \log p\left(\mathbf{u}_i|\mathbf{x}_i; \boldsymbol{\theta}\right) \right) \\
&= \nabla_{\boldsymbol{\theta}} \left(\sum_{i=1}^{N-1} \log p\left(\mathbf{u}_i|\mathbf{x}_i; \boldsymbol{\theta}\right) \right) \\
&= \sum_{i=1}^{N-1} \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u}_i|\mathbf{x}_i; \boldsymbol{\theta}\right)
\end{aligned}$$

Therefore the policy gradient is expressed as:

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \left\langle R(\boldsymbol{\tau}) \sum_{i=1}^{N-1} \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right\rangle_{\tilde{p}(\boldsymbol{\tau}_i)} \tag{5.3}$$

The equation above provide us with an estimate of the true gradient $\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u})$. In order to reduce the variance of this estimate we will incorporate the baseline b_k such that the following expression is minimized:

$$b_k = \operatorname{argmin} \left\langle \left(\left(R(\boldsymbol{\tau}) - b_k\right) \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) - \mu_k \right)^2 \right\rangle$$

where $\mu_k = \left\langle R(\boldsymbol{\tau}) \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right\rangle$. More precisely we will have that:

$$\begin{aligned}
&\partial_{b_k} \Bigg(\Bigg\langle \left(R(\boldsymbol{\tau}) - b_k\right)^2 \left(\sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right)^2 \\
&+ \mu_k^2 - 2\mu_k \left(R(\boldsymbol{\tau}) - b_k\right) \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \Bigg\rangle \Bigg) = \\
&\partial_{b_k} \Bigg(\Bigg\langle \left(R(\boldsymbol{\tau}) - b_k\right)^2 \left(\sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right)^2 \\
&+ \mu_k^2 - 2\mu_k R(\boldsymbol{\tau}) \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \Bigg\rangle \Bigg) = \\
&\partial_{b_k} \Bigg(\Bigg\langle \left(R(\boldsymbol{\tau}) - b_k\right)^2 \left(\sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right)^2 \Bigg\rangle \Bigg) = 0
\end{aligned}$$

where we have used the fact that:

$$\int p(\boldsymbol{\tau}) d\boldsymbol{\tau} = 1 \Rightarrow \nabla_{\boldsymbol{\theta}} \int p(\boldsymbol{\tau}) d\boldsymbol{\tau} = 0 \Rightarrow \left\langle \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}) \right\rangle_{p(\boldsymbol{\tau})} = 0$$

The optimal baseline is defined as:

$$\boxed{b_k = \frac{\left\langle \left(R(\boldsymbol{\tau}) \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right)^2 \right\rangle_{p(\boldsymbol{\tau})}}{\left\langle \sum_{i=1}^{N-1} \partial_{\theta_k} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right\rangle_{p(\boldsymbol{\tau})}}} \tag{5.4}$$

The final expression for the gradient is:

$$\boxed{\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \left\langle \mathbf{diag}\left(R(\boldsymbol{\tau}) - \mathbf{b}\right) \sum_{i=1}^{N-1} \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right\rangle_{p(\boldsymbol{\tau})}} \tag{5.5}$$

where $\mathbf{diag}\left(R(\boldsymbol{\tau}) - \mathbf{b}\right)$ is defined as:

$$\mathbf{diag}\left(R(\boldsymbol{\tau}) - \mathbf{b}\right) = \begin{pmatrix} R(\boldsymbol{\tau}) - b_1 & \ldots & 0 \\ 0 & & 0 \\ 0 & \ldots & R(\boldsymbol{\tau}) - b_n \end{pmatrix} \tag{5.6}$$

Without loss of generality, the policy could be parametrized as follows:

$$\mathbf{u}(\mathbf{x}, \boldsymbol{\theta}) dt = \boldsymbol{\Phi}(\mathbf{x}) \boldsymbol{\theta} dt + \mathbf{B}(\mathbf{x}) d\mathbf{w} \tag{5.7}$$

Under this parameterization we will have that:

$$p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) = \frac{1}{(2\pi)^{m/2} |\mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T|} \exp\left(-\frac{1}{2} \left(\mathbf{u} - \boldsymbol{\Phi}\boldsymbol{\theta}\right)^T \left(\mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T\right)^{-1} \left(\mathbf{u} - \boldsymbol{\Phi}\boldsymbol{\theta}\right) \right)$$

By taking the logarithm of the probability above we will have that:

$$\begin{aligned}\log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right) &= -\log(2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T| - \left(\frac{1}{2}\left(\mathbf{u}-\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\theta}\right)^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\left(\mathbf{u}-\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\theta}\right)\right) \\ &= -\log(2\pi)^{m/2}|\mathbf{B}(\mathbf{x})\mathbf{B}(\mathbf{x})^T| - \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\theta}^T\boldsymbol{\Phi}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{u} \\ &\quad - \frac{1}{2}\mathbf{u}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{u}\end{aligned}$$

Thus $\nabla_{\boldsymbol{\theta}}\log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right) = -\boldsymbol{\Phi}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\Phi}^T\mathbf{B}\mathbf{B}^T\mathbf{u} = \boldsymbol{\Phi}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{B}\boldsymbol{\epsilon}_i$ and the policy gradient will take the form:

$$\boxed{\nabla_{\boldsymbol{\theta}}\boldsymbol{\mathcal{J}}(\mathbf{x},\mathbf{u}) = \left\langle \mathbf{diag}\left(R(\boldsymbol{\tau})-\mathbf{b}\right)\sum_{i=1}^{N-1}\boldsymbol{\Phi}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{B}\boldsymbol{\epsilon}_i\right\rangle_{p(\boldsymbol{\tau})}} \tag{5.8}$$

The result above can take different formulations depending on the parameterization of the policy. Therefore, if $\mathbf{B} = \boldsymbol{\Phi}$ then we will have that:

$$\nabla_{\boldsymbol{\theta}}\boldsymbol{\mathcal{J}}(\mathbf{x},\mathbf{u}) = \left\langle \mathbf{diag}\left(R(\boldsymbol{\tau})-\mathbf{b}\right)\sum_{i=1}^{N-1}\mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{B}\boldsymbol{\epsilon}_i\right\rangle_{p(\boldsymbol{\tau})} \tag{5.9}$$

Before we move to the derivation of the policy gradient theorem it is important to realize that the expectations above are taken with respect to the state space trajectories, These trajectories can be generated by the application of the current policy (policy at every iteration) on the real physical system. In addition one may ask in which cases the expectations above result in zero gradient vector and therefore no further update of . The expectations compute the correlation of the perturbations of the policy parameters with the observed changes in the cost function. Therefore the gradient estimate will approach zero either when no change in the cost function is observed or there is no correlation

between the cost function and the parameter perturbations. In both cases, cost function tuning is of critical importance.

5.3 GPOMDP and policy gradient theorem

In their GPOMDP algorithm, (Baxter & Bartlett 2001) introduced several improvements over REINFORCE that made the gradient estimates more efficient. GPOMDP can also be derived from the policy gradient theorem (Sutton, McAllester, Singh & Mansour 2000, Peters & Schaal 2008*c*), and an optimal reward baseline can be added (cf. (Peters & Schaal 2008*c*))

Under the observation that past rewards do not affect future controls, the reinforce policy gradient can be reformulated as follows:

$$\boxed{\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \left\langle \sum_{i=1}^{N-1} \left(\mathbf{diag}\left(R_i(\boldsymbol{\tau}) - \mathbf{b}_i\right) \left(\nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \right) \right) \right\rangle_{p(\boldsymbol{\tau})}} \tag{5.10}$$

where $R_i(\boldsymbol{\tau}) = \frac{1}{N-i} \sum_{j=i}^{N} r(\mathbf{x}_j, \mathbf{u}_j)$. Given the parameterization of the policy the results above takes the form:

$$\boxed{\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \left\langle \sum_{i=1}^{N-1} \left(\mathbf{diag}\left(R_i(\boldsymbol{\tau}) - \mathbf{b}_i\right) \left(\boldsymbol{\Phi}^T \left(\mathbf{B}\mathbf{B}^T\right)^{-1} \mathbf{B}\boldsymbol{\epsilon}_i \right) \right) \right\rangle_{p(\boldsymbol{\tau})}} \tag{5.11}$$

The term $b_{k,i}$ is the optimal baseline that minimizes the variance of the estimated gradient.

$$\mathbf{diag}\left(R_i(\boldsymbol{\tau}) - \mathbf{b}_i\right) = \begin{pmatrix} R(\boldsymbol{\tau}) - b_{1,i} & \dots & 0 \\ 0 & & 0 \\ 0 & \dots & R(\boldsymbol{\tau}) - b_{n,i} \end{pmatrix} \tag{5.12}$$

The variance of the estimated gradient is expressed as follows:

$$\begin{aligned} &\left\langle \left(\sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) - \mu_k \right)^2 \right\rangle_{p(\boldsymbol{\tau})} = \\ &= \left\langle \left(\sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right)^2 \right\rangle_{p(\boldsymbol{\tau})} \\ &- \left\langle 2\mu_k \sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) + \mu_k^2 \right\rangle \end{aligned}$$

We take the derivative of the expectation above with respect to b_k and set it to zero. More precisely we will have that:

$$\begin{aligned} &\partial_{b_{k,m}} \left\langle \left(\sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) - \mu_k \right)^2 \right\rangle_{p(\boldsymbol{\tau})} = 0 \\ &\partial_{b_{k,m}} \left\langle \left(\sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) \right)^2 \right\rangle_{p(\boldsymbol{\tau})} \\ &- \partial_{b_{k,m}} \left\langle 2\mu_k \sum_{i=1}^{N-1} \left(\partial_{\theta_k} \log p\left(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}\right)\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) \right\rangle_{p(\boldsymbol{\tau})} = 0 \end{aligned}$$

Since $\left\langle \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}) \right\rangle_{p(\boldsymbol{\tau})} = 0$ the expression above takes the form:

$$\left\langle b_{k,m}\ \partial_{\theta_k} \log p\left(\mathbf{u}_m|\mathbf{x}_m;\boldsymbol{\theta}\right) \right\rangle_{p(\boldsymbol{\tau})} = \left\langle R_m(\boldsymbol{\tau}) \Big(\ \partial_{\theta_k} \log p\left(\mathbf{u}_m|\mathbf{x}_m;\boldsymbol{\theta}\right) \Big)^2 \right\rangle_{p(\boldsymbol{\tau})}$$

Thus, the optimal baseline is defined as:

$$\boxed{b_{k,m} = \frac{\left\langle R_m(\boldsymbol{\tau}) \Big(\ \partial_{\theta_k} \log p\left(\mathbf{u}_m|\mathbf{x}_m;\boldsymbol{\theta}\right) \Big)^2 \right\rangle_{p(\boldsymbol{\tau})}}{\left\langle \partial_{\theta_k} \log p\left(\mathbf{u}_m|\mathbf{x}_m;\boldsymbol{\theta}\right) \right\rangle_{p(\boldsymbol{\tau})}}} \tag{5.13}$$

5.4 Episodic natural actor critic

Vanilla policy gradients which follow the gradient of the expected cost function $J(\mathbf{x}, \mathbf{u})$ very often stuck into local minimum. As, it has been demonstrated in supervised learning (Amari 1999) natural gradients are less sensitive in getting trapped to local minima. Methods based on natural gradients do not follow the steepest direction in the parameter space but the steepest direction with respect to Fisher information metric.

One of the most efficient policy gradient algorithm was introduced in (Peters & Schaal 2008*b*), called the Episodic Natural Actor Critic. In essence, the method uses the Fisher Information Matrix to project the REINFORCE gradient onto a more effective update direction, which is motivated by the theory of natural gradients by (Amari 1999). The gradient for the eNAC algorithm takes the form of:

$$\tilde{\nabla} J = \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1}\ \nabla_{\boldsymbol{\theta}} J \tag{5.14}$$

where $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})$ is the Fisher information matrix. To derive the natural actor critic we start from the policy gradient theorem and we will have that:

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \left\langle \sum_{i=1}^{N-1} \left(\nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u}_i | \mathbf{x}_i; \boldsymbol{\theta}\right) \mathbf{diag}\left(R_i(\boldsymbol{\tau}) - b_{k,i}\right) \right) \right\rangle_{p(\boldsymbol{\tau})}$$

The equation above can be also written in the form:

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) = \int p(\mathbf{x}^{'} | \mathbf{x}, \mathbf{u}) \int p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \left(\nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \left(R(\boldsymbol{\tau}) - b_k\right) \right) d\mathbf{u} d\mathbf{x}$$

At this point the term $R(\boldsymbol{\tau}) - b_k$ is approximated with $\log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right)^T \mathbf{w}$. Thus substitution of $R(\boldsymbol{\tau}) - b_k = \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right)^T \mathbf{w}$ results in:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}) &= \int p(\mathbf{x}^{'} | \mathbf{x}, \mathbf{u}) \int p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \left(\nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right)^T \mathbf{w} \right) d\mathbf{u} d\mathbf{x} \\ &= \int p(\mathbf{x}^{'} | \mathbf{x}, \mathbf{u}) \boldsymbol{\mathcal{F}}(\mathbf{x}, \boldsymbol{\theta}) d\mathbf{x} \mathbf{w} \\ &= \boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) \mathbf{w} \end{aligned}$$

where $\boldsymbol{\mathcal{F}}(\mathbf{x}, \boldsymbol{\theta}) = \int p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \left(\nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right) \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{u} | \mathbf{x}; \boldsymbol{\theta}\right)^T \right) d\mathbf{u}$. By substituting the result above to the parameter update law we will have that:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} J = \boldsymbol{\theta}_k + \mathbf{w} \tag{5.15}$$

As we can see the update law is further simplified to just only updating the parameters $\boldsymbol{\theta}$ with $\mathbf{w}$. Thus it is important to compute $\mathbf{w}$. To do so, we consider the Bellman equation in terms of an advantage function $A(\mathbf{x}, \mathbf{u})$ and state value function $V(\mathbf{x})$. More precisely we will have:

$$Q(\mathbf{x}, \mathbf{u}) = A(\mathbf{x}, \mathbf{u}) + V(\mathbf{x}) = r(\mathbf{x}, \mathbf{u}) + \int p(\mathbf{x}'|\mathbf{x}, \mathbf{u}) V(\mathbf{x}') d\mathbf{x}$$

By evaluating the equation above on the trajectory $\left(\mathbf{x}_0^{(j)}, \mathbf{u}_0^{(j)} ... \mathbf{x}_{N-1}^{(j)}, \mathbf{u}_{N-1}^{(j)}, \mathbf{x}_N^{(j)}\right)$ we will have that:

$$\begin{aligned}
&\sum_{i=1}^{N-1} A(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)}) + V(\mathbf{x}_0^{(j)}) = \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)}) + V(\mathbf{x}_N^{(j)}) \\
&\sum_{i=1}^{N-1} \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(j)}|\mathbf{x}_i^{(j)}; \boldsymbol{\theta}\right)^T \mathbf{w} + V(\mathbf{x}_0^{(j)}) - V(\mathbf{x}_N^{(j)}) = \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)}) \\
&\sum_{i=1}^{N-1} \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(j)}|\mathbf{x}_i^{(j)}; \boldsymbol{\theta}\right)^T \mathbf{w} + \Delta V = \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)})
\end{aligned}$$

By combining the equations above for $j = 1, 2, ..., M$ we will have that:

$$\begin{pmatrix} \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(1)}|\mathbf{x}_i^{(1)}; \boldsymbol{\theta}\right)^T, & 1 \\ ... & ... \\ \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(M)}|\mathbf{x}_i^{(M)}; \boldsymbol{\theta}\right)^T, & 1 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \Delta V \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(1)}, \mathbf{u}_i^{(1)}) \\ ... \\ \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(M)}, \mathbf{u}_i^{(M)}) \end{pmatrix}$$

We regress the equation above and get the final result for $\mathbf{w}$ and obtain:

$$\mathbf{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{5.16}$$

where the matrix $\mathbf{X}$ and the vector $\mathbf{Y}$ are defined as follows:

$$\mathbf{X} = \begin{pmatrix} \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(1)}|\mathbf{x}_i^{(1)};\boldsymbol{\theta}\right)^T, & 1 \\ \ldots & \ldots \\ \nabla_{\boldsymbol{\theta}} p\left(\mathbf{u}_i^{(M)}|\mathbf{x}_i^{(M)};\boldsymbol{\theta}\right)^T, & 1 \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(1)}, \mathbf{u}_i^{(1)}) \\ \ldots \\ \sum_{i=1}^{N-1} r(\mathbf{x}_i^{(M)}, \mathbf{u}_i^{(M)}) \end{pmatrix}$$

To find the parameter vector $\mathbf{w} \in \Re^{n\times 1}$, there must be $M > n$ number of trajectories rollouts such that the matrix $\mathbf{X}^T\mathbf{X}$ is full rank and therefore invertible . With the episodic Natural Actor Critic we will conclude our presentation of PG methods. In the next section we discuss the application and comparison of PGs on a LQR optimal control problem.

5.5 Discussion

In this chapter we have reviewed the PG methods with the derivation of estimated corresponding gradients. The work on PG methods for reinforcement learning was an important advancement since it offered an alternative approach to optimal control problems in which either no model is available, or if there is a model, it is a bad approximation. Besides the their advantages, PG methods are in general, not easy to tune since they are very sensitive to exploration noise as well as the cost function design. In the next chapter we compare the PG methods with iterative path integral optimal control in via point tasks.

Chapter 6

Applications to Robotic Control

In this chapter we present the application of iterative path integral stochastic optimal control for the applications of planning and gain scheduling. We start our presentation in section 6.1 with the discussion on Dynamic Movement Primitives (DMPs) which corresponds to nonlinear point or limit cycle attractors with adjustable land scape. The DMPs play an essential role in the application of path integral control to learning robotic tasks. We discuss this role in section 6.2 where the ways in which DMPs are used for representing desired trajectories and for gain scheduling are presented. When the iterative path integral control framework is applied to DMPs the resulting algorithm is the so called the **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$). In section 6.3 we provide all the main equations of ($\mathbf{PI}^2$) and we discuss all the small adjustments required to robotic tasks with the use of the DMPs.

In section 6.4 $\mathbf{PI}^2$ is applied for learning optimal state space trajectories. The evaluations take place on simulated planar manipulators of different DOF and the little dog robot for the task or passing through a target and jumping over a gap respectively. In section 6.5 $\mathbf{PI}^2$ is applied for optimal planning and gain scheduling. The robotic tasks

include via point task with manipulators of various DOFs as well as the task of pushing a door to open with the simulated CBi humanoid robot. In the last section 6.8 we discuss the performance of $\mathbf{PI}^2$ in the aforementioned tasl and we conclude.

6.1 Learnable nonlinear attractor systems

6.1.1 Nonlinear point attractors with adjustable land-scape

The nonlinear point attractor consists of two sets of differential equations, the canonical and transformation system which are coupled through a nonlinearity (Ijspeert, Nakanishi, Pastor, Hoffmann & Schaal submitted),(Ijspeert, Nakanishi & Schaal 2003). The canonical system is formulated as $\frac{1}{\tau}\dot{x}_t = -\alpha x_t$. That is a first - order linear dynamical system for which, starting from some arbitrarily chosen initial state x_0 , e.g., $x_0 = 1$, the state x converges monotonically to zero. x can be conceived of as a phase variable, where $x = 1$ would indicate the start of the time evolution, and x close to zero means that the goal g (see below) has essentially been achieved. The transformation system consist of the following two differential equations:

$$\begin{aligned} \tau\dot{z} =& \alpha_z\beta_z\left(\left(g + \frac{f}{\alpha_z\beta_z}\right) - y\right) - \alpha_z z \\ \tau\dot{y} =& z \end{aligned} \tag{6.1}$$

Essentially, these 3 differential equations code a learnable point attractor for a movement from y_{t_0} to the goal g, where $\boldsymbol{\theta}$ determines the shape of the attractor. $y_t, \dot{y}_t$ denote

the position and velocity of the trajectory, while z_t, x_t are internal states. α_z, β_z, τ are time constants. The nonlinear coupling or forcing term f is defined as:

$$f(x) = \frac{\sum_{i=1}^{N} K\left(x_t, c_i\right) \theta_i x_t}{\sum_{i=1}^{N} K\left(x_t, c_i\right)} (g - y_0) = \boldsymbol{\Phi}_P(x)^T \boldsymbol{\theta} \tag{6.2}$$

The basis functions $K\left(x_t, c_i\right)$ are defined as:

$$K\left(x_t, c_i\right) = w_i = \exp\left(-0.5 h_i (x_t - c_i)^2\right) \tag{6.3}$$

with bandwidth h_i and center c_i of the Gaussian kernels – for more details see (Ijspeert et al. 2003). The full dynamics have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\mathbf{u}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (x, y, z)$ while the controls are specified as $\mathbf{u} = \boldsymbol{\theta} = (\theta_1, ..., \theta_p)^T$.The representation above is advantageous as it guarantees attractor properties towards the goal while remaining linear in the parameters $\boldsymbol{\theta}$ of the function approximator. By varying the parameter $\boldsymbol{\theta}$ the shape of the trajectory changes while the goal state g and initial state y_{t_0} remain fixed. These properties facilitate learning (Peters & Schaal 2008*c*).

6.1.2 Nonlinear limit cycle attractors with adjustable land-scape

The canonical system for the case of limit cycle attractors consist the differential equation $\tau\dot{\phi} = 1$ where the term $\phi \in [0, 2\pi]$ correspond to the phase angle of the oscillator in polar coordinates. The amplitude of the oscillation is assumed to be r. This oscillator produces a stable limit cycle when projected into Cartesian coordinated with $v_1 = r\cos(\phi)$ and $v_2 = r\sin(\phi)$. In fact, it corresponds to form of the (Hopf-like) oscillator equations

$$\tau \dot{v_1} = -\mu \frac{\sqrt{v_1^2 + v_2^2} - r}{\sqrt{v_1^2 + v_2^2}} v_1 - v_2 \tag{6.4}$$

$$\tau \dot{v_2} = -\mu \frac{\sqrt{v_1^2 + v_2^2} - r}{\sqrt{v_1^2 + v_2^2}} v_2 + v_1 \tag{6.5}$$

where μ is a positive time constant. The system above evolve to the limit cycle $v_1 = r\cos(t/\tau + c)$ and $v_2 = r\sin(t/\tau + c)$ with c a constant, given any initial conditions except $[v_1, v_2] = [0, 0]$ which is an unstable fixed point. Therefore the canonical system provides the amplitude signal (r) and a phase signal (ϕ) to the forcing term:

$$f(\phi, r) = \frac{\sum_{i=1}^{N} K(\phi, c_i)\theta_i}{\sum_{i=1}^{N} K(\phi, c_i)} r = \mathbf{\Phi}_R(\phi)^T \boldsymbol{\theta} \tag{6.6}$$

where the basis function $K(\phi, c_i)$ are defined as $K(\phi, c_i) = \exp(h_i(\cos(\phi - c_i) - 1))$. The forcing term is incorporated into the transformation system which is expressed by the equations (6.1). The full dynamics of the rhythmic movement primitives have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\mathbf{u}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (\phi, v_1, v_2, z, y)$ while the controls are specified as $\mathbf{u} = \boldsymbol{\theta} = (\theta_1, ..., \theta_p)^T$. The term g for the case of limit cycle attractors is interpreted as anchor point (or set point) for the oscillatory trajectory, which can be changed to accommodate any desired baseline of the oscillation The complexity of attractors is restricted only by the abilities of the function approximator used to generate the forcing term, which essentially allows for almost arbitrarily complex (smooth) attractors with modern function approximators

6.2 Robotic optimal control and planning with nonlinear attractors

In this section we show how the Path integral optimal control formalism in combination with the point and limit cycle attractors can be used for optimal planning (Theodorou, Buchli & Schaal 2010) and control (Buchli, Theodorou, Stulp & Schaal 2010) of robotic systems in high dimensions. As an example, consider a robotic system with rigid body dynamics (RBD) equations (Sciavicco & Siciliano 2000) using a parameterized policy:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\left(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{v}(\mathbf{q})\right) + \mathbf{M}(\mathbf{q})^{-1}\mathbf{u} \tag{6.7}$$

$$\mathbf{u} = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \tag{6.8}$$

where $\mathbf{M}$ is the RBD inertia matrix, $\mathbf{C}$ are Coriolis and centripetal forces, and $\mathbf{v}$ denotes gravity forces. The state of the robot is described by the joint angles $\mathbf{q}$ and joint velocities $\dot{\mathbf{q}}$. The proportional-Derivative (PD) controller with positive definite gain matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ have the form $\mathbf{K}_P = \mathbf{diag}\left(K_p^{(1)}, K_p^{(2)}, ..., K_p^{(N)}\right)$ and $\mathbf{K}_D = \mathbf{diag}\left(K_d^{(1)}, K_d^{(2)}, ..., K_d^{(N)}\right)$ where $K_p^{(i)}, K_d^{(i)}$ are the proportional and derivative gains for every DOF i. These gains converts a desired trajectory $\mathbf{q}_d, \dot{\mathbf{q}}_d$ into a motor command $\mathbf{u}$. The gains are parameterized as follows:

$$dK_p^{(i)} = \alpha_K\left(\mathbf{\Phi}_P^{(i)T}\left(\boldsymbol{\theta}^{(i)}dt + d\boldsymbol{\omega}^{(i)}\right) - K_p^{(i)}dt\right) \tag{6.9}$$

This equation models the time course of the position gains which are are represented by a basis function $\mathbf{\Phi}_P^{(i)T}\boldsymbol{\theta}^{(i)}$ linear with respect to the learning parameter $\boldsymbol{\theta}^{(i)}$, and these

parameter can be learned with the ($\mathbf{PI}^2$). We will assume that the time constant α_K is so large, that for all practical purposes we can assume that $K_P^{(i)} = \mathbf{\Phi}_p^{(i)T}\left(\boldsymbol{\theta}^{(i)} + \boldsymbol{\epsilon}_t^{(i)}\right)$ holds at all time where $\boldsymbol{\epsilon}_t^{(i)} = \frac{d\boldsymbol{\omega}^{(i)}}{dt}$. In our experiments $\mathbf{K}_D$ gains are specified as $K_d^{(i)} = \xi\sqrt{K_p^{(i)}}$ where ξ is user determined. Alternatively, for the case of optimal planing we could create another form of control structure in which we add for the RBD system (6.7) the following equation:

$$\ddot{\mathbf{q}}_d \quad = \quad \boldsymbol{\mathcal{G}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)(\boldsymbol{\theta} + \boldsymbol{\epsilon}_t) \tag{6.10}$$

where $\boldsymbol{\mathcal{G}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)$ is represented with a point or limit cycle attractor. The control or learning parameter for this case is the parameter $\boldsymbol{\theta}$ in (6.10).

6.3 Policy improvements with path integrals: The ($\mathbf{PI}^2$) algorithm.

After having introduced the nonlinear stable attractors with learnable landscapes which from now on we will call them as Dynamic Movement Primitives(DMPs), in this section we discuss the application of iterative path integral control to DMPs. The resulting algorithm is the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals $\mathbf{PI}^2$. As can be easily recognized, the DMP equations are of the form of our control system (4.2), with only one controlled equation and a one dimensional actuated state. This case has been treated in Section 4.4. The motor commands are replaced with the parameters $\boldsymbol{\theta}$ – the issue of time

dependent vs. constant parameters will be addressed below. More precisely, the DMP equations can be written as:

$$\begin{pmatrix} \dot{x}_t \\ \dot{z}_t \\ \dot{y}_t \end{pmatrix} = \begin{pmatrix} -\alpha x_t \\ y_t \\ \alpha_z(\beta_z(g - y_t) - z_t) \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{1\times p} \\ \mathbf{0}_{1\times p} \\ \mathbf{g}_t^{(c)T} \end{pmatrix} (\boldsymbol{\theta}_t + \boldsymbol{\epsilon}_t) \tag{6.11}$$

The state of the DMP is partitioned into the controlled part $\mathbf{x}_t^{(c)} = y_t$ and uncontrolled part $\mathbf{x}_t^{(m)} = (x_t \ z_t)^T$. The control transition matrix depends on the state, however, it depends only on one of the state variables of the uncontrolled part of the state, i.e., x_t. The path cost for the stochastic dynamics of the DMPs is given by:

$$\begin{aligned}
\tilde{S}(\boldsymbol{\tau}_i) &= \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt + \frac{1}{2} \sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log |\mathbf{H}_{t_j}| \\
&\propto \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2} \sum_{j=i}^{N-1} \left\| \mathbf{g}_{t_j}^{(c)T} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j}) \right\|_{\mathbf{H}_{t_j}^{-1}}^2 \\
&= \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2} \sum_{j=i}^{N-1} \frac{1}{2} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j})^T \mathbf{g}_{t_j}^{(c)} \mathbf{H}_{t_j}^{-1} \mathbf{g}_{t_j}^{(c)T} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j}) \\
&= \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2} \sum_{j=i}^{N-1} \frac{1}{2} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j})^T \frac{\mathbf{g}_{t_j}^{(c)} \mathbf{g}_{t_j}^{(c)T}}{\mathbf{g}_t^{(c)T} \mathbf{R}^{-1} \ \mathbf{g}_t^{(c)}} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j}) \\
&= \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2} \sum_{j=i}^{N-1} \frac{1}{2} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j})^T \mathbf{M}_{t_j}^T \mathbf{R} \mathbf{M}_{t_j} (\boldsymbol{\theta}_{t_j} + \boldsymbol{\epsilon}_{t_j})
\end{aligned} \tag{6.12}$$

with $\mathbf{M}_{t_j} = \frac{\mathbf{R}^{-1} \mathbf{g}_{t_j} \ \mathbf{g}_{t_j}^T}{\mathbf{g}_{t_j}^T \mathbf{R}^{-1} \mathbf{g}_{t_j}}$. $\mathbf{H}_t$ becomes a scalar given by $\mathbf{H}_t = \mathbf{g}_t^{(c)T} \mathbf{R}^{-1} \ \mathbf{g}_t^{(c)}$. Interestingly, the term $\frac{\lambda}{2} \sum_{j=i}^{N-1} \log |\mathbf{H}_{t_j}|$ for the case of DMPs depends only on x_t, which is a

deterministic variable and therefore can be ignored since it is the same for all sampled paths. We also absorbed, without loss of generality, the time step dt in cost terms. Consequently, the fundamental result of the path integral stochastic optimal problem for the case of DMPs is expressed as:

$$\boxed{\mathbf{u}_{t_i} = \int P\left(\boldsymbol{\tau}_i\right) \mathbf{u}\left(\boldsymbol{\tau}_i\right) d\boldsymbol{\tau}_i^{(c)}} \tag{6.13}$$

where the probability $P\left(\boldsymbol{\tau}_i\right)$ and local controls $\mathbf{u}\left(\boldsymbol{\tau}_i\right)$ are defined as

$$P\left(\boldsymbol{\tau}_i\right) = \frac{e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)}}{\int e^{-\frac{1}{\lambda}\tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i}, \quad \mathbf{u}(\boldsymbol{\tau}_i) = \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}\mathbf{g}_{t_i}^{(c)T}}{\mathbf{g}_{t_i}^{(c)T}\mathbf{R}^{-1}\mathbf{g}_{t_i}^{(c)}}\boldsymbol{\epsilon}_{t_i} \tag{6.14}$$

and the path cost given as

$$\tilde{S}(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2}\sum_{j=i}^{N-1} \boldsymbol{\epsilon}_{t_j}^T \mathbf{M}_{t_j}^T \mathbf{R}\mathbf{M}_{t_j}\boldsymbol{\epsilon}_{t_j} \tag{6.15}$$

Note that $\boldsymbol{\theta} = 0$ in these equations, i.e., the parameters are initialized to zero. These equations correspond to the case where the stochastic optimal control problem is solved with one evaluation of the optimal controls (6.13) using dense sampling of the whole state space under the "passive dynamics" (i.e., $\boldsymbol{\theta} = 0$), which requires a significant amount of exploration noise. Such an approach was pursued in the original work by (Kappen 2007, Broek et al. 2008), where a potentially large number of sample trajectories was needed to achieve good results. Extending this sampling approach to high dimensional spaces, however, is daunting, as with very high probablity, we would sample primarily rather

useless trajectories. Thus, biasing sampling towards good initial conditions seems to be mandatory for high dimensional applications.

Thus, we consider only local sampling and an iterative update procedure. Given a current guess of $\boldsymbol{\theta}$, we generate sample roll-outs using stochastic parameters $\boldsymbol{\theta} + \boldsymbol{\epsilon}_t$ at every time step. To see how the generalized path integral formulation is modified for the case of iterative updating, we start with the equations of the update of the parameter vector $\boldsymbol{\theta}$, which can be written as:

$$\begin{aligned}
\boldsymbol{\theta}_{t_i}^{(new)} &= \int P\left(\boldsymbol{\tau}_i\right) \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{t_i})}{{\mathbf{g}_{t_i}}^T\mathbf{R}^{-1}\mathbf{g}_{t_i}} d\boldsymbol{\tau}_i & (6.16)\\
&= \int P\left(\boldsymbol{\tau}_i\right) \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T\boldsymbol{\epsilon}_{t_i}}{{\mathbf{g}_{t_i}}^T\mathbf{R}^{-1}\mathbf{g}_{t_i}} d\boldsymbol{\tau}_i + \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T\boldsymbol{\theta}}{{\mathbf{g}_{t_i}}^T\mathbf{R}^{-1}\mathbf{g}_{t_i}} & (6.17)\\
&= \delta\boldsymbol{\theta}_{t_i} + \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T}{tr\left(\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T\right)}\boldsymbol{\theta} & (6.18)\\
&= \delta\boldsymbol{\theta}_{t_i} + \mathbf{M}_{t_i}\boldsymbol{\theta} & (6.19)
\end{aligned}$$

The correction parameter verctor $\delta\boldsymbol{\theta}_{t_i}$ is defined as $\delta\boldsymbol{\theta}_{t_i} = \int P\left(\boldsymbol{\tau}_i\right) \frac{\mathbf{R}^{-1}\mathbf{g}_{t_i}{\mathbf{g}_{t_i}}^T\boldsymbol{\epsilon}_{t_i}}{{\mathbf{g}_{t_i}}^T\mathbf{R}^{-1}\mathbf{g}_{t_i}} d\boldsymbol{\tau}_i$. It is important to note that $\boldsymbol{\theta}_{t_i}^{(new)}$ is now time dependent, i.e., for every time step t_i, a different optimal parameter vector is computed. In order to return to one single time independent parameter vector $\boldsymbol{\theta}^{(new)}$, the vectors $\boldsymbol{\theta}_{t_i}^{(new)}$ need to be averaged over time t_i.

We start with a first tentative suggestion of averaging over time, and then explain why it is inappropriate, and what the correct way of time averaging has to look like. The tentative and most intuitive time average is:

$$\boldsymbol{\theta}^{(new)} = \frac{1}{N}\sum_{i=0}^{N-1} \boldsymbol{\theta}_{t_i}^{(new)} = \frac{1}{N}\sum_{i=0}^{N-1} \delta\boldsymbol{\theta}_{t_i} + \frac{1}{N}\sum_{i=0}^{N-1} \mathbf{M}_{t_i}\boldsymbol{\theta}$$

Thus, we would update $\boldsymbol{\theta}$ based on two terms. The first term is the average of $\delta\boldsymbol{\theta}_{t_i}$, which is reasonable as it reflects the knowledge we gained from the exploration noise. However, there would be a second update term due to the average over projected mean parameters $\boldsymbol{\theta}$ from every time step – it should be noted that $\mathbf{M}_{t_i}$ is a projection matrix onto the range space of $\mathbf{g}_{t_i}$ under the metric $\mathbf{R}^{-1}$, such that a multiplication with $\mathbf{M}_{t_i}$ can only shrink the norm of $\boldsymbol{\theta}$. From the viewpoint of having optimal parameters for *every time step*, this update component is reasonable as it trivially eliminates the part of the parameter vector that lies in the null space of $\mathbf{g}_{t_i}$ and which contributes to the command cost of a trajectory in a useless way. From the view point of a parameter vector that is *constant and time independent* and that is updated *iteratively*, this second update is undesirable, as the multiplication of the parameter vector $\boldsymbol{\theta}$ with $\mathbf{M}_{t_i}$ in (6.19) and the averaging operation over the time horizon reduces the L_2 norm of the parameters at every iteration, potentially in an uncontrolled way[1]. What we rather want is to achieve convergence when the average of $\delta\boldsymbol{\theta}_{t_i}$ becomes zero, and we do not want to continue updating due to the second term.

The problem is avoided by eliminating the projection matrix in the second term of averaging, such that it become:

$$\boldsymbol{\theta}^{(new)} = \frac{1}{N}\sum_{i=0}^{N-1}\delta\boldsymbol{\theta}_{t_i} + \frac{1}{N}\sum_{i=0}^{N-1}\boldsymbol{\theta} = \frac{1}{N}\sum_{i=0}^{N-1}\delta\boldsymbol{\theta}_{t_i} + \boldsymbol{\theta}$$

The meaning of this reduced update is simply that we keep a component in $\boldsymbol{\theta}$ that is irrelevant and contributes to our trajectory cost in a useless way. However, this irrelevant

[1]To be precise, θ would be projected and continue shrinking until it lies in the intersection of all null spaces of the $\mathbf{g}_{t_i}$ basis function – this null space can easily be of measure zero.

component will not prevent us from reaching the optimal effective solution, i.e., the solution that lies in the range space of $\mathbf{g}_{t_i}$. Given this modified update, it is, however, also necessary to derive a compatible cost function. As mentioned before, in the unmodified scenario, the last term of (6.12) is:

$$\frac{1}{2}\sum_{j=i}^{N-1}(\boldsymbol{\theta}+\boldsymbol{\epsilon}_{t_j})^T\mathbf{M}_{t_j}^T\mathbf{R}\mathbf{M}_{t_j}(\boldsymbol{\theta}+\boldsymbol{\epsilon}_{t_j}) \tag{6.20}$$

To avoid a projection of $\boldsymbol{\theta}$, we modify this cost term to be:

$$\frac{1}{2}\sum_{j=i}^{N-1}(\boldsymbol{\theta}+\mathbf{M}_{t_j}\boldsymbol{\epsilon}_{t_j})^T\mathbf{R}(\boldsymbol{\theta}+\mathbf{M}_{t_j}\boldsymbol{\epsilon}_{t_j}) \tag{6.21}$$

With this modified cost term, the path integral formalism results in the desired $\boldsymbol{\theta}_{t_i}^{(new)}$ without the $\mathbf{M}_{t_i}$ projection of θ.

The main equations of the iterative version of the generalized path integral formulation, called **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$), can be summarized as:

$$P(\boldsymbol{\tau}_i) = \frac{e^{-\frac{1}{\lambda}S(\boldsymbol{\tau}_i)}}{\int e^{-\frac{1}{\lambda}S(\boldsymbol{\tau}_i)}d\boldsymbol{\tau}_i} \tag{6.22}$$

$$S(\boldsymbol{\tau}_i) = \phi_{t_N}+\sum_{j=i}^{N-1}q_{t_j}dt+\frac{1}{2}\sum_{j=i}^{N-1}(\boldsymbol{\theta}+\mathbf{M}_{t_j}\boldsymbol{\epsilon}_{t_j})^T\mathbf{R}(\boldsymbol{\theta}+\mathbf{M}_{t_j}\boldsymbol{\epsilon}_{t_j})dt \tag{6.23}$$

$$\delta\boldsymbol{\theta}_{t_i} = \int P(\boldsymbol{\tau}_i)\,\mathbf{M}_{t_i}\boldsymbol{\epsilon}_{t_i}d\boldsymbol{\tau}_i \tag{6.24}$$

$$[\delta\boldsymbol{\theta}]_j = \frac{\sum_{i=0}^{N-1}(N-i)\;w_{j,t_i}\;[\delta\boldsymbol{\theta}_{t_i}]_j}{\sum_{i=0}^{N-1}w_{j,t_i}(N-i)} \tag{6.25}$$

$$\boldsymbol{\theta}^{(new)} = \boldsymbol{\theta}^{(old)}+\delta\boldsymbol{\theta} \tag{6.26}$$

Essentially, (6.22) computes a discrete probability at time t_i of each trajectory roll-out with the help of the cost (6.23). For every time step of the trajectory, a parameter update is computed in (6.24) based on a probability weighted average over trajectories. The parameter updates at every time step are finally averaged in (6.25). Note that we chose a weighted average by giving every parameter update a weight[2] according to the time steps left in the trajectory and the activation of the kernel in (6.3). This average can be interpreted as using a function approximator with only a constant (offset) parameter vector to approximate the time dependent parameters. Giving early points in the trajectory a higher weight is useful since their parameters affect a large time horizon and thus higher trajectory costs. Other function approximation (or averaging) schemes could be used to arrive at a final parameter update – we preferred this simple approach as it gave very good learning results. The final parameter update is $\boldsymbol{\theta}^{(new)} = \boldsymbol{\theta}^{(old)} + \delta\boldsymbol{\theta}$.

The parameter λ regulates the sensitivity of the exponentiated cost and can automatically be optimized for every time step i to maximally discriminate between the experienced trajectories. More precisely, a constant term can be subtracted from (6.23) as long as all $S(\boldsymbol{\tau}_i)$ remain positive – this constant term [3] cancels in (6.22). Thus, for a given number of roll-outs, we compute the exponential term in (6.22) as

$$\exp\left(-\frac{1}{\lambda}S(\boldsymbol{\tau}_i)\right) = \exp\left(-h\frac{S(\boldsymbol{\tau}_i) - \min S(\boldsymbol{\tau}_i)}{\max S(\boldsymbol{\tau}_i) - \min S(\boldsymbol{\tau}_i)}\right) \tag{6.27}$$

[2]The use of the kernel weights in the basis functions (6.3) for the purpose of time averaging has shown better performance with respect to other weighting approaches, across all of our experiments. Therefore this is the weighting that we suggest. Users may develop other weighting schemes as more suitable to their needs.

[3]In fact, the term inside the exponent results by adding $\frac{h \min S(\boldsymbol{\tau}_i)}{\max S(\boldsymbol{\tau}_i) - \min S(\boldsymbol{\tau}_i)}$, which cancels in (6.22), to the term $-\frac{hS(\boldsymbol{\tau}_i)}{\max S(\boldsymbol{\tau}_i) - \min S(\boldsymbol{\tau}_i)}$ which is equal to $-\frac{1}{\lambda}S(\boldsymbol{\tau}_i)$.

with h set to a constant, which we chose to be $h = 10$ in all our evaluations. The max and min operators are over all sample roll-outs. This procedure eliminates λ and leaves the variance of the exploration noise $\boldsymbol{\epsilon}$ as the only open algorithmic parameter for $\mathbf{PI}^2$. It should be noted that the equations for $\mathbf{PI}^2$ have no numerical pitfalls: no matrix inversions and no learning rates [4], rendering $\mathbf{PI}^2$ to be very easy to use in practice.

The pseudocode for the final $\mathbf{PI}^2$ algorithm for a one dimensional control system with function approximation is given in Table 6.1. A tutorial Matlab example of applying $\mathbf{PI}^2$ can be found at http://www-clmc.usc.edu/software .

6.4 Evaluations of ($\mathbf{PI}^2$) for optimal planning

We evaluated $\mathbf{PI}^2$ in several synthetic examples in comparison with REINFORCE, GPOMDP, eNAC, and, when possible, PoWER. Except for PoWER, all algorithms are suitable for optimizing immediate reward functions of the kind $r_t = q_t + \mathbf{u}_t\mathbf{R}\mathbf{u}_t$. As mentioned above, PoWER requires that the immediate reward behaves like an improper probability. This property is incompatible with $r_t = q_t + \mathbf{u}_t\mathbf{R}\mathbf{u}_t$ and requires some special nonlinear transformations, which usually change the nature of the optimization problem, such that PoWER optimizes a different cost function. Thus, only one of the examples below has a compatible a cost function for all algorithms, including PoWER. In all examples below, exploration noise and, when applicable, learning rates, were tuned for every individual algorithms to achieve the best possible numerically stable performance. Exploration noise was only added to the maximally activated basis function in a motor primitive, and

[4] $\mathbf{R}$ is a user design parameter and usually chosen to be diagonal and invertible.

Table 6.1: Pseudocode of the $\mathbf{PI}^2$ algorithm for a 1D Parameterized Policy (Note that the discrete time step dt was absobed as a constant multiplier in the cost terms).

- **Given**:
  - An immediate cost function $r_t = q_t + \boldsymbol{\theta}_t^T \mathbf{R} \boldsymbol{\theta}_t$
  - A terminal cost term ϕ_{t_N} (cf. 4.25)
  - A stochastic parameterized policy $\mathbf{a}_t = \mathbf{g}_t^T (\boldsymbol{\theta} + \boldsymbol{\epsilon}_t)$
  - The basis function $\mathbf{g}_{t_i}$ from the system dynamics (cf. 4.2)
  - The variance $\Sigma_{\boldsymbol{\epsilon}}$ of the mean-zero noise $\boldsymbol{\epsilon}_t$
  - The initial parameter vector $\boldsymbol{\theta}$
- **Repeat** until convergence of the trajectory cost R:
  - Create K roll-outs of the system from the same start state $\mathbf{x}_0$ using stochstic parameters $\boldsymbol{\theta} + \boldsymbol{\epsilon}_t$ at every time step
  - **For** $k = 1...K$, compute:
    - $P(\boldsymbol{\tau}_{i,k}) = \frac{e^{-\frac{1}{\lambda} S(\boldsymbol{\tau}_{i,k})}}{\sum_{k=1}^{K} [e^{-\frac{1}{\lambda} S(\boldsymbol{\tau}_{i,k})}]}$
    - $S(\boldsymbol{\tau}_{i,k}) = \phi_{t_N,k} + \sum_{j=i}^{N-1} q_{t_j,k} + \frac{1}{2} \sum_{j=i+1}^{N-1} (\boldsymbol{\theta} + \mathbf{M}_{t_j,k} \boldsymbol{\epsilon}_{t_j,k})^T \mathbf{R} (\boldsymbol{\theta} + \mathbf{M}_{t_j,k} \boldsymbol{\epsilon}_{t_j,k})$
    - $\mathbf{M}_{t_j,k} = \frac{\mathbf{R}^{-1} \mathbf{g}_{t_j,k} \ \mathbf{g}_{t_j,k}^T}{\mathbf{g}_{t_j,k}^T \mathbf{R}^{-1} \mathbf{g}_{t_j,k}}$
  - **For** $i = 1...(N-1)$, compute:
    - $\delta \boldsymbol{\theta}_{t_i} = \sum_{k=1}^{K} [P(\boldsymbol{\tau}_{i,k}) \mathbf{M}_{t_i,k} \ \boldsymbol{\epsilon}_{t_i,k}]$
  - Compute $[\delta \boldsymbol{\theta}]_j = \frac{\sum_{i=0}^{N-1} (N-i) \ w_{j,t_i} \ [\delta \boldsymbol{\theta}_{t_i}]_j}{\sum_{i=0}^{N-1} w_{j,t_i} (N-i)}$
  - Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \delta \boldsymbol{\theta}$
  - Create one noiseless roll-out to check the trajectory cost $R = \phi_{t_N} + \sum_{i=0}^{N-1} r_{t_i}$. In case the noise cannot be turned off, i.e., a stochastic system, multiple roll-outs need be averaged.

---

the noise was kept constant for the entire time that this basis function had the highest activation – empirically, this trick helped to improve the learning speed of all algorithms.

6.4.1 Learning Optimal Performance of a 1 DOF Reaching Task

The first evaluation considers learning optimal parameters for a 1 DOF DMP (cf. Equation 6.11). The immediate cost and terminal cost are, respectively:

$$r_t = 0.5 f_t^2 + 5000\ \boldsymbol{\theta}^T \boldsymbol{\theta} \qquad \phi_{t_N} = 10000(\dot{y}_{t_N}^2 + 10(g - y_{t_N})^2) \tag{6.28}$$

with $y_{t_0} = 0$ and $g = 1$ – we use *radians* as units motivated by our interest in robotics application, but we could also avoid units entirely. The interpretation of this cost is that we would like to reach the goal g with high accuracy while minimizing the acceleration of the movement and while keeping the parameter vector short. Each algorithm was run for 15 trials to compute a parameter update, and a total of 1000 updates were performed. Note that 15 trials per update were chosen as the DMP had 10 basis functions, and the eNAC requires at least 11 trials to perform a numerically stable update due to its matrix inversion. The motor primitives were initialized to approximate a 5-th order polynomial as point-to-point movement (cf. Figure 6.1a,b), called a minimum-jerk trajectory in the motor control literature; the movement duration was 0.5 seconds, which is similar to normal human reaching movements. Gaussian noise of $N(0, 0.1)$ was added to the initial parameters of the movement primitives in order to have different initial conditions for every run of the algorithms. The results are given in Figure 6.1. Figure 6.1a,b show the initial (before learning) trajectory generated by the DMP together with the

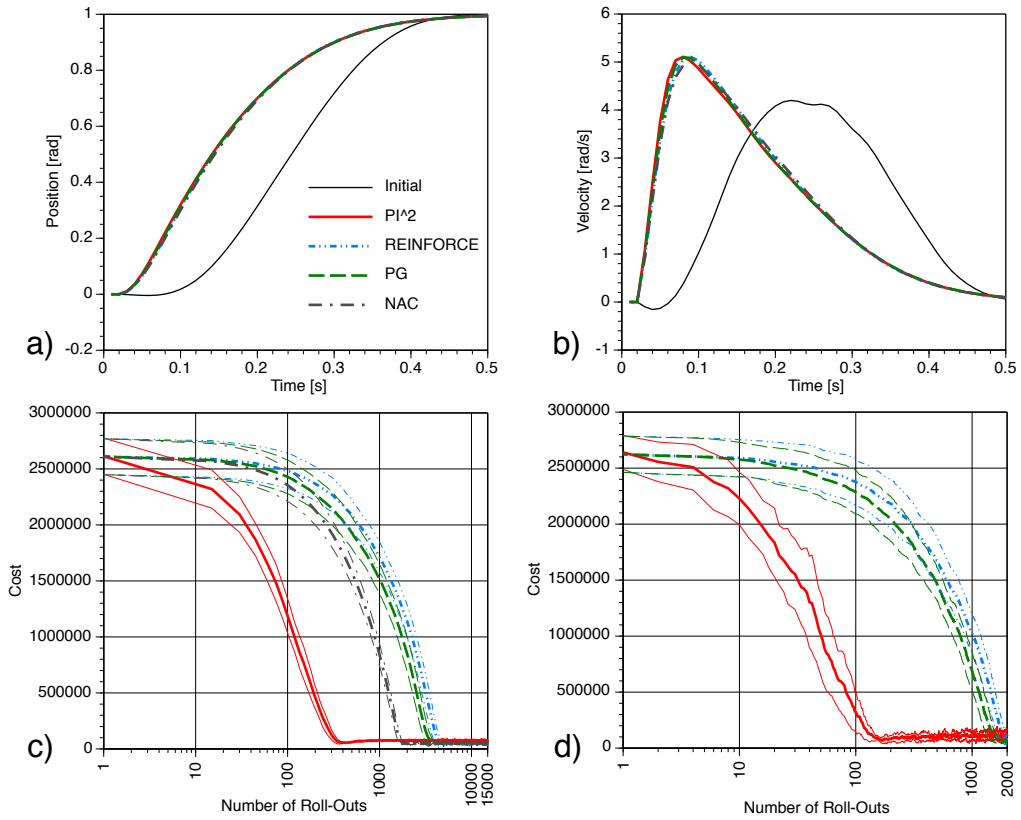


Figure 6.1: Comparison of reinforcement learning of an optimized movement with motor primitives. a) Position trajectories of the initial trajectory (before learning) and the results of all algorithms after learning – the different algorithms are essentially indistuighishable. b) The same as a), just using the velocity trajectories. c) Average learning curves for the different algorithms with 1 std error bars from averaging 10 runs for each of the algorithms. d) Learning curves for the different algorithms when only two roll-outs are used per update (note that the eNAC cannot work in this case and is omitted).

learning results of the four different algorithms after learning – essentially, all algorithms achieve the same result such that all trajectories lie on top of each other. In Figure 6.1c, however, it can be seen that $\mathbf{PI}^2$ outperforms the gradient algorithms by an order of magnitude. Figure 6.1d illustrates learning curves for the same task as in Figure 6.1c, just that parameter updates are computed already after two roll-outs – the eNAC was excluded from this evaluation as it would be too heuristic to stablize its ill-conditioned matrix inversion that results from such few roll-outs. $\mathbf{PI}^2$ continues to converge much faster than the other algorithms even in this special scenario. However, there are some

noticable fluctuation after convergence. This noise around the convergence baseline is caused by using only two noisy roll-outs to continue updating the parameters, which causes continuous parameter fluctuations around the optimal parameters. Annealing the exploration noise, or just adding the optimal trajectory from the previous parameter update as one of the roll-outs for the next parameter update can alleviate this issue – we do not illustrate such little "tricks" in this paper as they really only affect fine tuning of the algorithm.

6.4.2 Learning optimal performance of a 1 DOF via-point task

The second evaluation was identical to the first evaluation, just that the cost function now forced the movement to pass through an intermediate via-point at $t = 300ms$. This evaluation is an abstract approximation of hitting a target, e.g., as in playing tennis, and requires a significant change in how the movement is performed relative to the initial trajectory (Figure 6.2a). The cost function was

$$r_{300ms} = 100000000(G - y_{t_{300ms}})^2 \qquad \phi_{t_N} = 0 \tag{6.29}$$

with $G = 0.25$. Only this single reward was given. For this cost function, the PoWER algorithm can be applied, too, with cost function $\tilde{r}_{300ms} = \exp(-1/\lambda \quad r_{300ms})$ and $\tilde{r}_{t_i} = 0$ otherwise. This transformed cost function has the same optimum as r_{300ms}. The resulting learning curves are given in Figure 6.2 and resemble the previous evaluation: $\mathbf{PI}^2$ outperforms the gradient algorithms by roughly an order of magnitude, while all the gradient algorithms have almost identical learning curves. As was expected from the

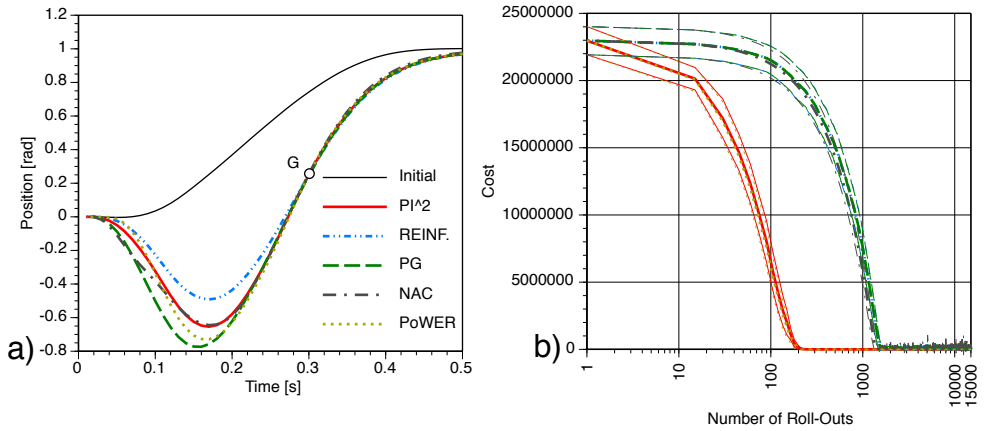


Figure 6.2: Comparison of reinforcement learning of an optimized movement with motor primitives for passing through an intermediate target G. a) Position trajectories of the initial trajectory (before learning) and the results of all algorithms after learning. b) Average learning curves for the different algorithms with 1 std error bars from averaging 10 runs for each of the algorithms.

similarity of the update equations, PoWER and $\mathbf{PI}^2$ have in this special case the same performance and are hardly distinguisable in Figure 6.2. Figure 6.2a demonstrates that all algorithms pass through the desired target G, but that there are remaining differences between the algorithms in how they approach the target G – these difference have a small numerical effect in the final cost (where $\mathbf{PI}^2$ and PoWER have the lowest cost), but these difference are hardly task relevant.

6.4.3 Learning optimal performance of a multi-DOF via-point task

A third evaluation examined the scalability of our algorithms to a high-dimensional and highly redundant learning problem. Again, the learning task was to pass through an intermediate target G, just that a $d = 2, 10$, or 50 dimensional motor primitive was employed. We assume that the multi-DOF systems model planar robot arms, where d links of equal length $l = 1/d$ are connected in an open chain with revolute joints. Essentially, these robots look like a multi-segment snake in a plane, where the tail of the

snake is fixed at the origin of the 2D coordinate system, and the head of the snake can be moved in the 2D plane by changing the joint angles between all the links. Figure 6.3b,d,f illustrate the movement over time of these robots: the initial position of the robots is when all joint angles are zero and the robot arm completely coincides with the x-axis of the coordinate frame. The goal states of the motor primitives command each DOF to move to a joint angle, such that the entire robot configuration afterwards looks like a semi-circle where the most distal link of the robot (the end-effector) touches the y-axis. The higher priority task, however, is to move the end-effector through a via-point $G = (0.5, 0.5)$. To formalize this task as a reinforcement learning problem, we denote the joint angles of the robots as ξ_i, with $i = 1, 2, ..., d$, such that the first line of (6.11) reads now as $\ddot{\xi}_{i,t} = f_{i,t} + \mathbf{g}_{i,t}^T(\boldsymbol{\theta}_i + \boldsymbol{\epsilon}_{i,t})$ – this small change of notation is to avoid a clash of variables with the (x, y) task space of the robot. The end-effector position is computed as:

$$x_t = \frac{1}{d}\sum_{i=1}^{d} cos(\sum_{j=1}^{i} \xi_{j,t}), \quad y_t = \frac{1}{d}\sum_{i=1}^{d} sin(\sum_{j=1}^{i} \xi_{j,t}) \tag{6.30}$$

The immediate reward function for this problem is defined as

$$r_t = \frac{\sum_{i=1}^{d}(d+1-i)\left(0.1 f_{i,t}^2 + 0.5\ \boldsymbol{\theta}_i^T \boldsymbol{\theta}_i\right)}{\sum_{i=1}^{d}(d+1-i)} \tag{6.31}$$

$$\Delta r_{300ms} = 100000000\left((0.5 - x_{t_{300ms}})^2 + (0.5 - y_{t_{300ms}})^2\right) \tag{6.32}$$

$$\phi_{t_N} = 0 \tag{6.33}$$

where Δr_{300ms} is added to r_t at time $t = 300ms$, i.e., we would like to pass through the via-point at this time. The individual DOFs of the motor primitive were initialized as in

the 1 DOF examples above. The cost term in (6.31) penalizes each DOF for using high accelerations and large parameter vectors, which is a critical component to achieve a good resolution of redundancy in the arm. Equation (6.31) also has a weighting term $d+1-i$ that penalizes DOFs proximal to the orgin more than those that are distal to the origin — intuitively, applied to human arm movements, this would mean that wrist movements are cheaper than shoulder movements, which is motivated by the fact that the wrist has much lower mass and inertia and is thus energetically more efficient to move.

The results of this experiment are summarized in Figure 6.3. The learning curves in the left column demonstrate again that **PI**^{2} has an order of magnitude faster learning performance than the other algorithms, irrespective of the dimensionality. **PI**^{2} also converges to the lowest cost in all examples:

Algorithm	2-DOFs	10-DOFs	50-DOFs
PI^2	98000 ± 5000	15700 ± 1300	2800 ± 150
REINFORCE	125000 ± 2000	22000 ± 700	19500 ± 24000
PG	128000 ± 2000	28000 ± 23000	27000 ± 40000
NAC	113000 ± 10000	48000 ± 8000	22000 ± 2000

Figure 6.3 also illustrates the path taken by the end-effector before and after learning. All algorithms manage to pass through the via-point G appropriately, although the path particularly before reaching the via-point can be quite different across the algorithms. Given that **PI**^{2} reached the lowest cost with low variance in all examples, it appears to have found the best solution. We also added a "stroboscopic" sketch of the robot arm for the **PI**^{2} solution, which proceeds from the very right to the left as a function of time. It

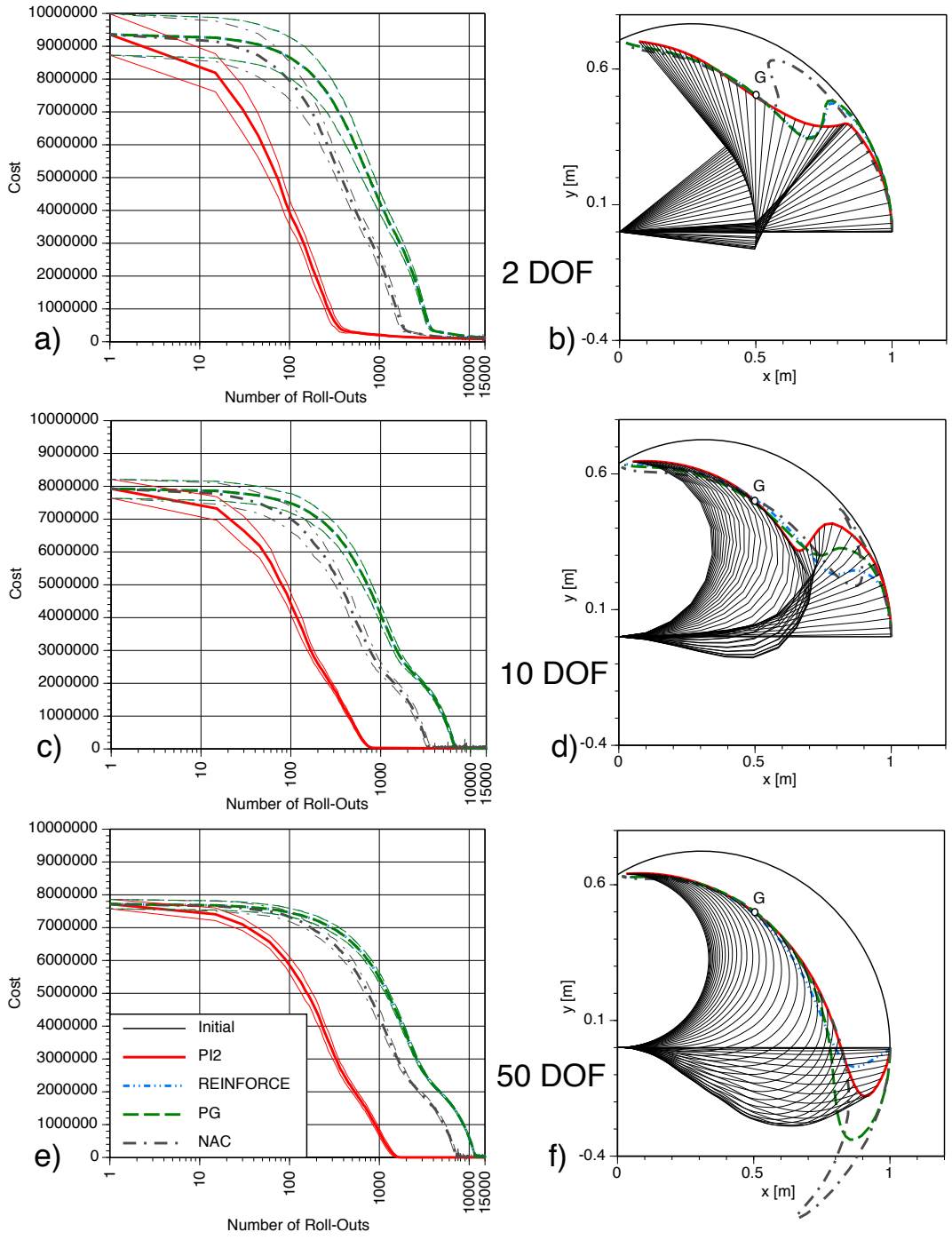


Figure 6.3: Comparison of learning multi-DOF movements (2,10, and 50 DOFs) with planar robot arms passing through a via-point G. a,c,e) illustrate the learning curves for different RL algorithms, while b,d,f) illustrate the end-effector movement after learning for all algorithms. Additionally, b,d,f) also show the initial end-effector movement, before learning to pass through G, and a "stroboscopic" visualization of the arm movement for the final result of $\mathbf{PI}^2$ (the movements proceed in time starting at the very right and ending by (almost) touching the y axis).

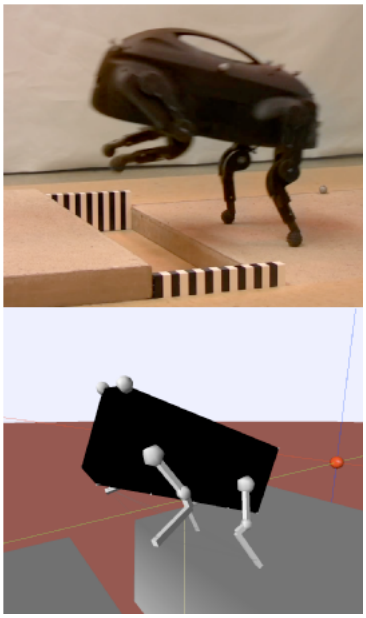
should be emphasized that there were absolutely no parameter tuning needed to achieve the $\mathbf{PI}^2$ results, while all gradient algorithms required readjusting of learning rates for every example to achieve best performance.

6.4.4 Application to robot learning

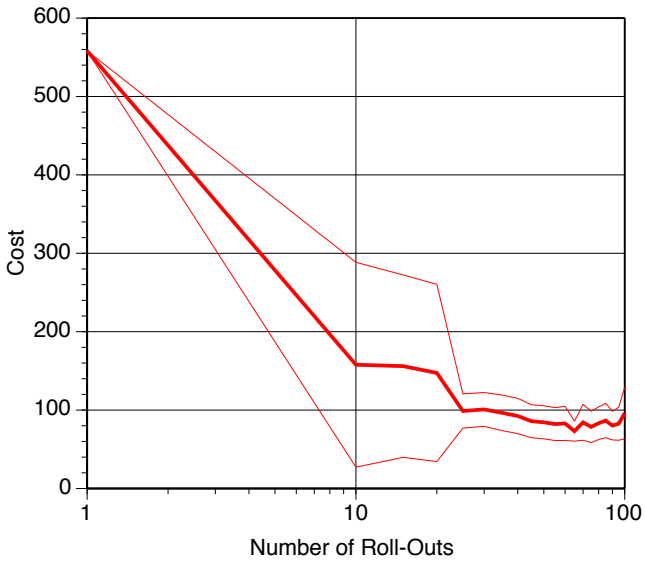
Figure 6.4 illustrates our application to a robot learning problem. The robot dog is to jump across as gap. The jump should make forward progress as much as possible, as it is a maneuver in a legged locomotion competition which scores the speed of the robot – note that we only used a physical simulator of the robot for this experiment, as the actual robot was not available. The robot has three DOFs per leg, and thus a total of $d = 12$ DOFs. Each DOF was represented as a DMP with 50 basis functions. An initial seed behavior (Figure 6.5-top) was taught by learning from demonstration, which allowed the robot barely to reach the other side of the gap without falling into the gap – the demonstration was generated from a manual adjustment of spline nodes in a spline-based trajectory plan for each leg.

$\mathbf{PI}^2$ learning used primarily the forward progress as a reward, and slightly penalized the squared acceleration of each DOF, and the length of the parameter vector. Additionally, a penalty was incurred if the yaw or the roll exceeded a threshold value – these penalties encouraged the robot to jump straight forward and not to the side, and not to fall over. The exact cost function is:

$$r_t = r_{roll} + r_{yaw} + \sum_{i=1}^{d} \left(a_1 f_{i,t}^2 + 0.5 a_2 \, \boldsymbol{\theta}_i^T \boldsymbol{\theta}\right) (a_1 = 1.e-6, a_2 = 1.e-8) \quad (6.34)$$



(a) Real & Simulated Robot Dog



(b) Learning curve for Dog Jump with $\mathbf{PI}^2$ $\pm 1 std$

Figure 6.4: Reinforcement learning of optimizing to jump over a gap with a robot dog. The improvement in cost corresponds to about 15 cm improvement in jump distance, which changed the robot's behavior from an initial barely successful jump to jump that completely traversed the gap with entire body. This learned behavior allowed the robot to traverse a gap at much higher speed in a competition on learning locomotion.

$$r_{roll} = \begin{cases} 100 * (|roll_t| - 0.3)^2, & \text{if } (|roll_t| > 0.3) \\ 0, & otherwise \end{cases} \tag{6.35}$$

$$r_{yaw} = \begin{cases} 100 * (|yaw_t| - 0.1)^2, & \text{if } (|yaw_t| > 0.1) \\ 0, & otherwise \end{cases} \tag{6.36}$$

$$\phi_{t_N} = 50000(goal - x_{nose})^2 \tag{6.37}$$

where $roll, yaw$ are the roll and yaw angles of the robot's body, and x_{nose} is the position of the front tip (the "nose") of the robot in the forward direction, which is the direction towards the *goal*. The multipliers for each reward component were tuned to have a balanced influence of all terms. Ten learning trials were performed initially for the first parameter update. The best 5 trials were kept, and five additional new trials were performed for the second and all subsequent updates. Essentially, this method performs importance sampling, as the rewards for the 5 trials in memory were re-computed with the latest parameter vectors. A total of 100 trials was performed per run, and ten runs were collected for computing mean and standard deviations of learning curves.

(i.e., 5 updates), the performance of the robot was converged and significantly improved, such that after the jump, almost the entire body was lying on the other side of the gap. Figure 6.4 captures the temporal performance in a sequence of snapshots of the robot. It should be noted that applying $\mathbf{PI}^2$ was algorithmically very simple, and manual tuning only focused on generated a good cost function, which is a different research topic beyond the scope of this paper.

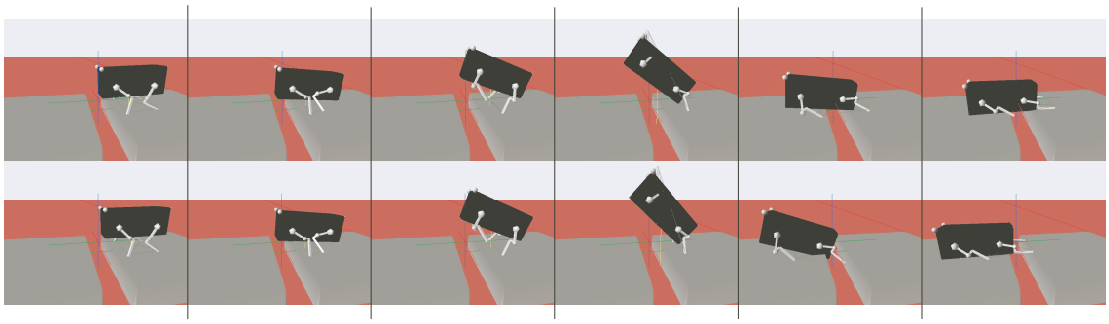


Figure 6.5: Sequence of images from the simulated robot dog jumping over a 14cm gap. Top: before learning. Bottom: After learning. While the two sequences look quite similar at the first glance, it is apparent that in the 4th frame, the robot's body is significantly heigher in the air, such that after landing, the body of the dog made about 15cm more forward progress as before. In particular, the entire robot's body comes to rest on the other side of the gap, which allows for an easy transition to walking.

6.5 Evaluations of (PI^2) on planning and gain scheduling

In the next sections we evaluate the $\mathbf{PI}^2$ on the problems of optimal planning and gain scheduling. In a typical planning scenario the goal is to find or to learn trajectories which minimize some performance criterion. As we have seen in the previous sections at every iteration of the learning algorithm, new trajectories are generated based on which the new planing policy is computed. The new planning policy is used at the next iteration to generated new trajectories which are again used to compute the improved planning policy. The process continues until the convergence criterion is meet.

In this learning process main assumption is the existence of a control policy that is adequate to steer the system such that it follows the trajectories generated at every iteration of the learning procedure. In this section we go one step further and apply $\mathbf{PI}^2$ not only to find the optimal desired trajectories but also to learn control policies that minimize a performance criterion. This performance criterion is a function of kinematic

variables of the underlying dynamics and the strength of control gains that are incorporated in the control policy. Essentially, the goal for the robot is to be able to perform the task with as lower gains as possible.

6.6 Way-point experiments

We start our evaluations with way -point experiments in two simulated robots, the 3DOF Phantom robot and the 6DOF Kuka robot. For both robots, the immediate reward at time step t is given as:

$$r(t) = w_{gain} \sum_i K^i_{P,t} + w_{acc} ||\ddot{\mathbf{x}}|| + w_{subgoal} C(t) \tag{6.38}$$

Here, $\sum_i K^i_{P,t}$ is the sum over the proportional gains over all joints. The reasoning behind penalizing the gains is that low gains lead to several desirable properties of the system such as compliant behavior (safety and/or robustness (Buchli, Kalakrishnan, Mistry, Pastor & Schaal 2009)), lowered energy consumption, and less wear and tear. The term $||\ddot{\mathbf{x}}||$ is magnitude of the accelerations of the end-effector. This quantity is penalized to avoid high-jerk end-effector motion. This penalty is low in comparison to the gain penalty.

The robot's primary task is to pass through an intermediate goal, either in joint space or end-effector space – such scenarios occur in tasks like playing tennis or table tennis. The component of the cost function $C(t)$ that represents this primary task will be described individually for each robot in the next sections. Gains and accelerations are penalized at each time step, but $C(t)$ only leads to a cost at specific time steps along

the trajectory. Finally for both robots, the cost weights are $w_{int} = 2000$, $w_{gain} = 1/N$, $w_{acc} = 1/N$. Dividing the weights by the number of time steps N is convenient, as it makes the weights independent of the duration of a movement.

6.6.1 Phantom robot, passing through waypoint in joint space

The Phantom Premium 1.5 Robot is a 3 DOF, two link arm. It has two rotational degrees of freedom at the base and one in the arm. We use a physically realistic simulation of this robot generated in SL (Schaal 2009), as depicted in Figure 6.6.

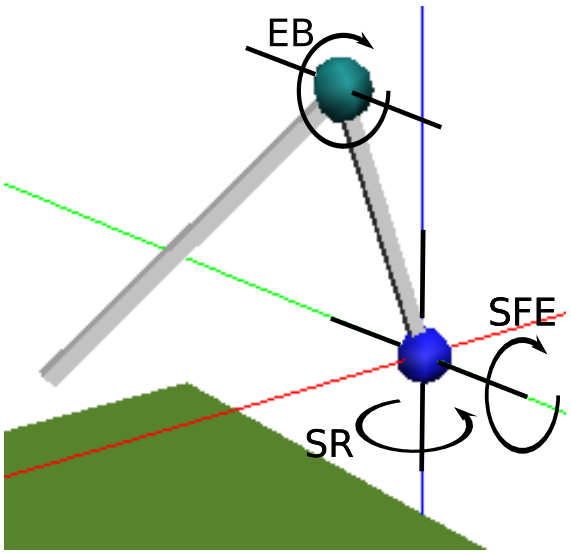


Figure 6.6: 3-DOF Phantom simulation in SL.

The task for this robot is intentionally simple and aimed at demonstrating the ability to tune task relevant gains in joint space with straightforward and easy to interpret data. The duration of the movement is 2.0s, which corresponds to 1000 time steps at 500Hz servo rate. The intermediate goals for this robot are set as follows:

$$C(t) = \delta(t - 0.4) \mid q_{SR}(t) + 0.2 \mid + \delta(t - 0.8) \mid q_{SFE}(t) - 0.4 \mid + \delta(t - 1.2) \mid q_{EB}(t) - 1.5 \mid$$

This penalizes joint SR for not having an angle $q_{SR} = -0.2$ at time $t = 0.4s$. Joints SFE and EB are also required to go through (different) intermediate angles at times $0.8s$ and $1.2s$ respectively.

The initial parameters $\boldsymbol{\theta}^i$ for the reference trajectory are determined by training the DMPs with a minimum jerk trajectory (Zefran, Kumar & Croke 1998) in joint space from $\mathbf{q}_{t=0.0} = [0.0\ 0.3\ 2.0]^T$ to $\mathbf{q}_{t=2.0} = [-0.6\ 0.8\ 1.4]^T$. The function approximator for the proportional gains of the 3 joints is initialized to return a constant gain of $6.0 Nm/rad$. The initial trajectories are depicted as red, dashed plots in Figure 6.8, where the angles and gains of the three joints are plotted against time. Since the task of $\mathbf{PI}^2$ is to optimize both trajectories and gains with respect to the cost function, this leads to a 6-D RL problem. The robot executes 100 parameter updates, with 4 noisy exploration trials per update. After each update, we perform one noise-less test trial for evaluation purposes.

Figure 6.7 depicts the learning curve for the phantom robot (left), which is the overall cost of the noise-less test trial after each parameter update. The joint space trajectory and gain scheduling after 100 updates are depicted as blue, solid lines in Figure 6.8.

From these graphs, we draw the following conclusions:

- $\mathbf{PI}^2$ has adapted the initial minimum jerk trajectories such that they fulfill the task and pass through the desired joint angles at the specified times. These intermediate goals are represented by the circles on the graphs.

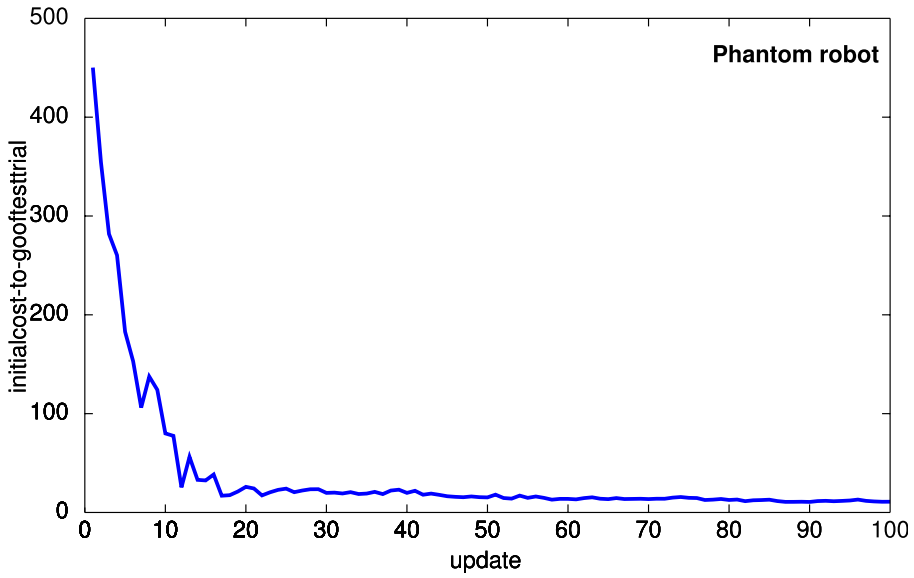


Figure 6.7: Learning curves for the phantom robot.

- Because the magnitude of gains is penalized in general, they are low when the task allows it. After $t = 1.6s$, all gains drop to the minimum value[5], because accurate tracking is no longer required to fulfill the goal. Once the task is completed, the robot becomes maximally compliant, as one would wish it to be.

- When the robot is required to pass through the intermediate targets, it needs better tracking, and therefore higher gains. Therefore, the peaks of the gains correspond roughly to the times where the joint is required to pass through an intermediate point.

- Due to nonlinear effects, e.g., Coriolis and centripedal forces, the gain schedule shows more complex temporal behavior as one would initially assume from specifying three different joint space targets at three different times.

[5]We bounded the gains between pre-specified maximum and minimum values. Too high gains would generate oscillations and can lead to instabilities of the robot, and too low gains lead to poor tracking such that the robot frequently runs into the joint limits.

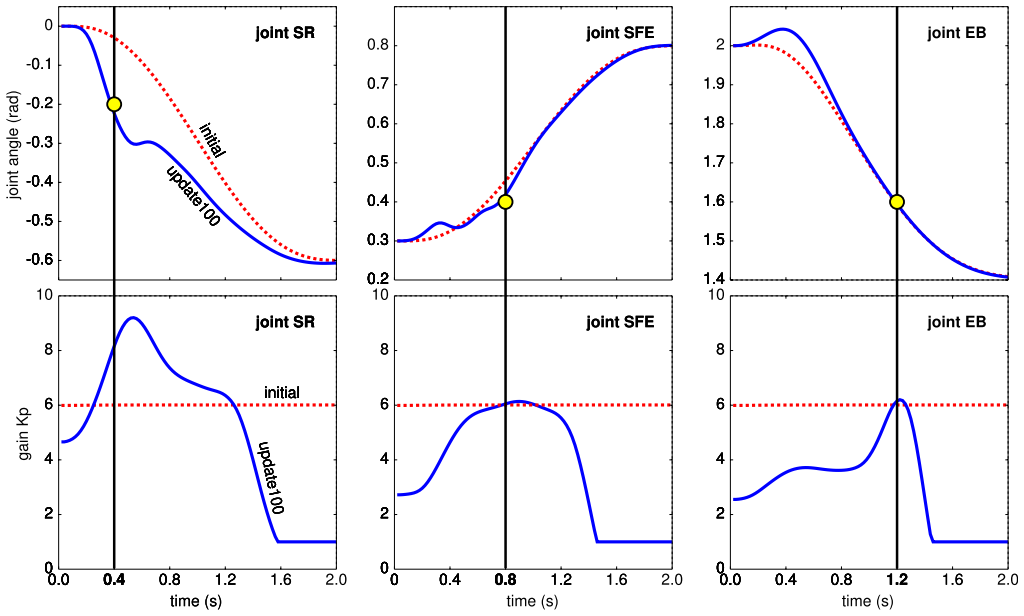


Figure 6.8: Initial (red, dashed) and final (blue, solid) joint trajectories and gain scheduling for each of the three joints of the phantom robot. Yellow circles indicate intermediate subgoals.

In summary, we achieved the objective of variable impedance control: the robot is compliant when possible, but has a higher impedance when the task demands it.

6.6.2 Kuka robot, passing through a waypoint in task space

Next we show a similar task on a 6 DOF anthropomorphic arm, a Kuka Light-Weight Arm. This example illustrates that our approach scales well to higher-dimensional systems, and also that appropriate gains schedules are learned when intermediate targets are chosen in end-effector space instead of joint space.

The duration of the movement is $1.0s$, which corresponds to 500 time steps. This time, the intermediate goal is for the end-effector $\mathbf{x}$ to pass through $[\ 0.7\ 0.3\ 0.1]^T$ at time $t = 0.5s$:

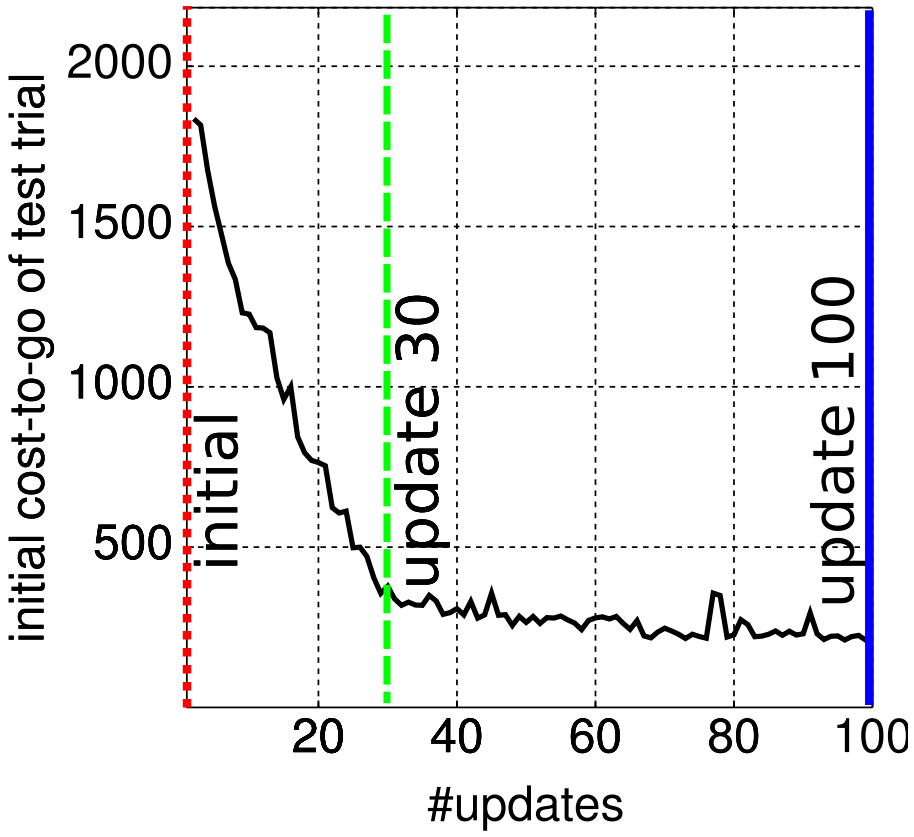


Figure 6.9: Learning curves for the Kuka robot.

$$C(t) = \quad \delta(t - 0.5)(\mathbf{x} - [\ 0.7\ 0.3\ 0.1]^T) \tag{6.39}$$

The six joint trajectories are again initialized as minimum jerk trajectories. As before, the resulting initial trajectory is plotted as red, dashed line in Figure 6.10. The initial gains are set to a constant $[60, 60, 60, 60, 25, 6]^T$. Given these initial conditions, finding the parameter vectors for DMPs and gains that minimizes the cost function leads to a 12-D RL problem. We again perform 100 parameter updates, with 4 exploration trials per update.

The learning curve for this problem is depicted in Figure 6.9. The trajectory of the end-effector after 30 and 100 updates is depicted in Figure 6.10. The intermediate goal at $t = 0.5$ is visualized by circles. Finally, Figure 6.11 shows the gain schedules after 30 and 100 updates for the 6 joints of the Kuka robot.

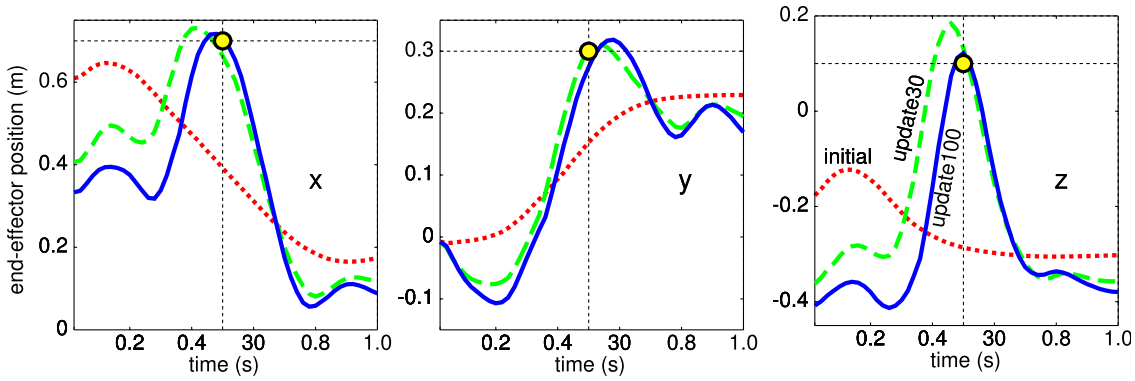


Figure 6.10: Initial (red, dotted), intermediate (green, dashed), and final (blue, solid) end-effector trajectories of the Kuka robot.

From these graphs, we draw the following conclusions:

- **PI**^{2} has adapted joint trajectories such that the end-effector passes through the intermediate subgoal at the right time. It learns to do so after only 30 updates (Figure 6.7).

- After 100 updates the peaks of most gains occur just before the end-effector passes through the intermediate goal (Figure 6.11), and in many cases decrease to the minimum gain directly afterwards. As with the phantom robot we observe high impedance when the task requires accuracy, and more compliance when the task is relatively unconstrained.

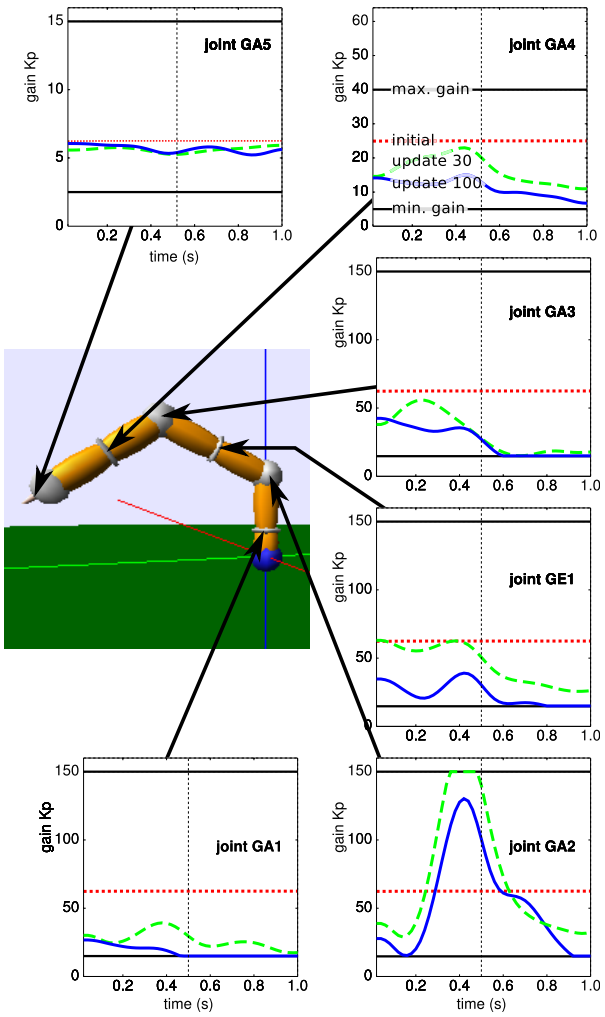


Figure 6.11: Initial (red, dotted), intermediate (green, dashed), and final (blue, solid) joint gain schedules for each of the six joints of the Kuka robot.

- The second joint (GA2) has the most work to perform, as it must support the weight of all the more distal links. Its gains are by far the highest, especially at the intermediate goal, as any error in this DOF will lead to a large end-effector error.

- The learning has two phases. In the first phase (plotted as dashed, green), the robot is learning to make the end-effector pass through the intermediate goal. At

this point, the basic shape of the gain scheduling has been determined. In the second phase, $\mathbf{PI}^2$ fine tunes the gains, and lowers them as much as the task permits.

6.7 Manipulation task

6.7.1 Task 2: Pushing open a door with the CBi humanoid

In this task, the simulated CBi humanoid robot (Cheng, Hyon, Morimoto, Ude, Hale, Colvin, Scroggin & Jacobsen 2007) is required to open a door. This robot is accurately simulated with the SL software (Schaal 2009). For this task, we not only learn the gain schedules, but also improve the planned joint trajectories with$\mathbf{PI}^2$simultaneously.

Regarding the initial trajectory in this task, we fix the base of the robot, and consider only the 7 degrees of freedom in the left arm. The initial trajectory before learning is a minimum jerk trajectory in joint space. In the initial state, the upper arm is kept parallel to the body, and the lower arm is pointing forward. The target state is depicted in Figure 6.12. With this task, we demonstrate that our approach can not only be applied to imitation of observed behavior, but also to manually specify trajectories, which are fine-tuned along with the gain schedules.

The gains of the 7 joints are initialized to $1/10th$ of their default values. This leads to extremely compliant behavior, whereby the robot is not able to exert enough force to overcome the static friction of the door, and thus cannot move it. The minimum gain for all joints was set to 5. Optimizing both joint trajectories and gains leads to a 14-dimensional learning problem.

The terminal cost is the degree to which the door was opened, i.e. $\phi_{t_N} = 10^4 \cdot (\psi_{max} - \psi_N)$, where the maximum door opening angle ψ_{max} is $0.3rad$ (it is out of reach otherwise). The immediate cost for the gains is again $q_t = \frac{1}{N} \sum_{i=1}^{3} K_P^i$. The sum of the gains of all joints is divided by the number of time steps of the trajectory N, to be independent of trajectory duration. The cost for the gains expresses our preference for low gain control.

The variance of the exploration noise for the gains is again $10^{-4}\gamma^n$, and for the joint trajectories $10\gamma^n$, both with decay parameter $\lambda = 0.99$ and n the number of updates. The relatively high exploration noise for the joint trajectories does not express less exploration per se, but is rather due to numerical differences in using the function approximator to model the gains directly rather than as the non-linear component of a DMP. The number of executed and reused 'elite' roll-outs is both 5, so the number of roll-outs on which the update is performed is $K = 10$.

Figure 6.12 (right) depicts the total cost of the noise-less test trial after each update. The costs for the gains are plotted separately. When all of the costs are due to gains, i.e. the door is opened completely to ψ_{max} and the task is achieved, the graphs of the total cost and that of the gains coincide. Here, it can be clearly seen that the robot switches to high-gain control in the first 6 updates (costs of gains go up) to achieve the task (cost of not opening the door goes down). Then, the gains are lowered at each update, until they are lower than the initial values. The joint trajectories and gain schedules after 0, 6 and 100 updates are depicted in Figure 6.13.

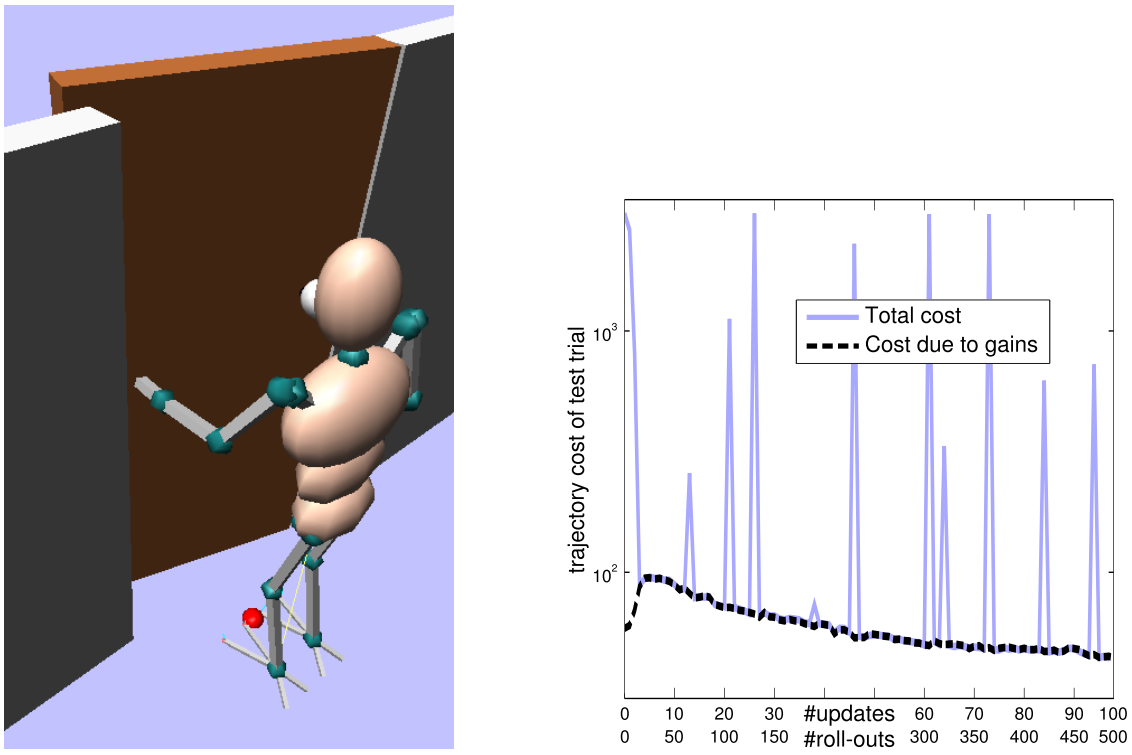


Figure 6.12: Left: Task scenario. Right: Learning curve for the door task. The costs specific to the gains are plotted separately.

6.7.2 Task 3: Learning tasks on the PR2

In (Pastor, Kalakrishnan, Chitta, Theodorou & Schaal 2011) $\mathbf{PI}^2$ was used on PR2 robot for learning how to perform two manipulations tasks: learning billiard and rolling a box with chopsticks. In this section, we leave the details of the application of $\mathbf{PI}^2$ and we focus on the design of the cost function for these two tasks. A more thorough and in detailed discussion on the application of $\mathbf{PI}^2$ on the PR2 can be found in (Pastor et al. 2011).

For the first task of learning to play billiard the critical issue is to find the states which are relevant. These state are illustrated in 6.14 and they consist of the cue roll, pitch, yaw, the elbow posture and the cue tip offset. The cost function used is minimizes large cue displacements, time to the target and distance to the target. Thus:

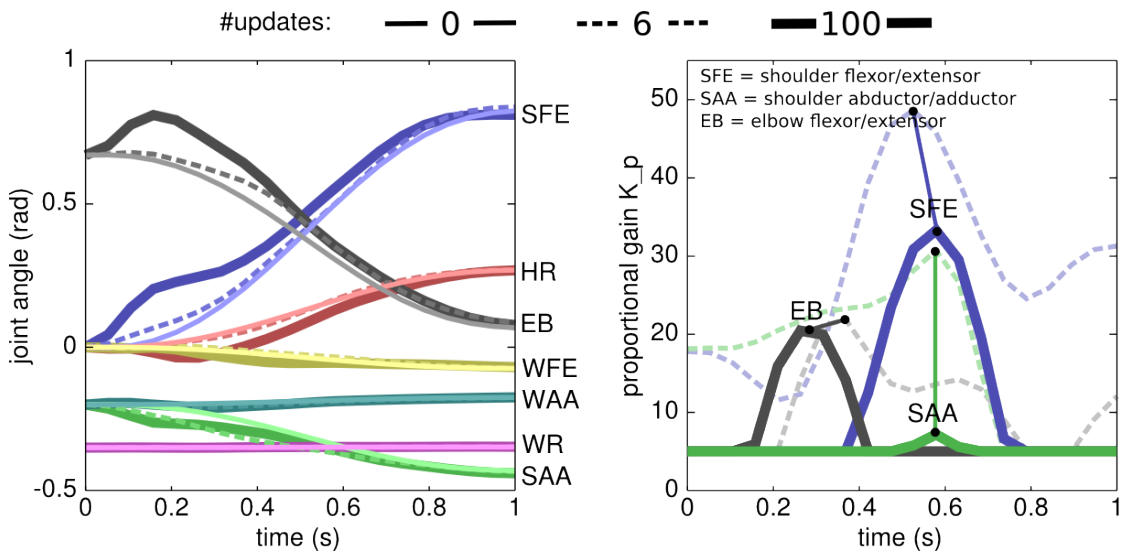


Figure 6.13: Learned joint angle trajectories (center) and gain schedules (right) of the CBi arm after 0/6/100 updates.

$$q(\mathbf{x}) = w_1 \times (\text{Displacement}) + w_2 \times (\text{Time to Target}) + w_3 \times (\text{Distance to Target})$$

For the second tasks, the goal for the robot is to learn to flip the box by using chopsticks. The state dependent cost function for this particular task penalizes high box accelerations measured by an IMU insight the box, high forces measured in the tactile sensors of PR2 robot and high arm accelerations measured by an accelerometer at each gripper. The terminal cost penalizes deviation from the desired state which is the one with the boxed flipped. Thus the cost function is expressed as:

$$q(\mathbf{x}) = w_1 \times (\text{Box acceleration}) + w_2 \times (\text{Force}) + w_3 \times (\text{Gripper acceleration})$$

$$\phi(\mathbf{x}_{T_N}) = w_4 \times (\text{Terminal state error})$$

In figure 6.15 the initial policy and the final policy are illustrated. As we can see the robot learns how to succesfully flip the box

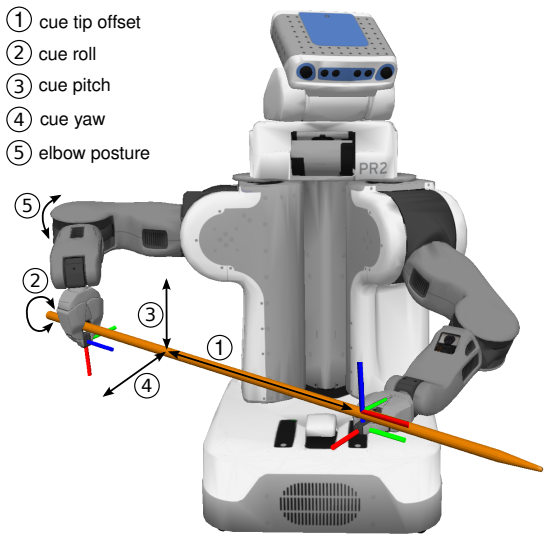


Figure 6.14: Relevant states for learning how to play billiard.

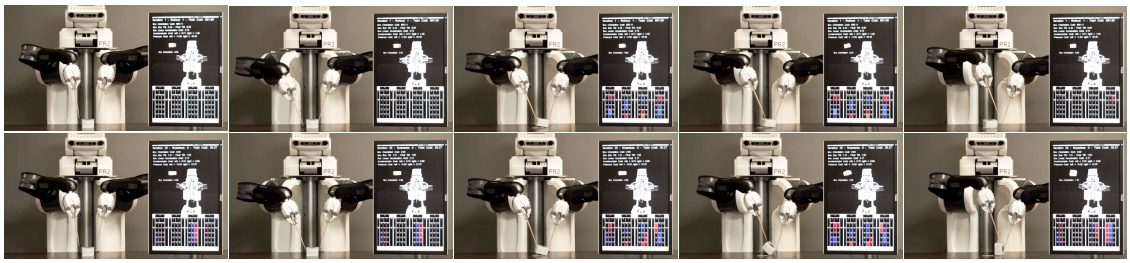


Figure 6.15: Initial and final policies for rolling the box.

6.8 Discussion

We have applied applied the **PI**^{2} algorithm, which is modified version of iterative path integral control, to the problems of optimal planning and control for robotic tasks. The

DMPs which correspond to nonlinear dynamical systems with adjustable land scape play an essential role for the representation of kinematic trajectories and control gains.

The key results of the path integral control formalism, which were presented in Table 4.1 and Section 4.4, consider how to compute the optimal controls for a general class of stochastic control systems with state-dependent control transition matrix. One important class of these systems can be interpreted in the framework of reinforcement learning with parameterized policies. For this class, we derived Policy Improvement with Path Integrals ($\mathbf{PI}^2$) as a novel algorithm for learning a parametrized policy. $\mathbf{PI}^2$ inherits its sound foundation in first order principles of stochastic optimal control from the path integral formalizm. It is a probabilistic learning method without open algorithmic tuning parameters, except for the exploration noise. In our evaluations, $\mathbf{PI}^2$ outperformed gradient algorithms significantly. It is also numerically simpler and has easier cost function design than previous probabilistic RL methods that require that immediate rewards are pseudo-probabilities. The similarity of $\mathbf{PI}^2$ with algorithms based on probability matching indicates that the principle of probability matching seems to approximate a stochastic optimal control framework. Our evaluations demonstrated that $\mathbf{PI}^2$ can scale to high dimensional control systems, unlike many other reinforcement learning systems.

The mathematical structure of the $\mathbf{PI}^2$ algorithm makes it suitable to optimize simultaneously both reference trajectories and gain schedules. This is similar to classical DDP. We evaluated our approach on two simulated robot systems, which posed up to 14 dimensional learning problems in continuous state- action spaces. The goal was to learn compliant control while fulfilling kinematic task constraints, like passing through an intermediate target. The evaluations demonstrated that the algorithm behaves as expected:

it increases gains when needed, but tries to maintain low gain control otherwise. The optimal reference trajectory always fulfilled the task goal. Learning speed was rather fast, i.e., within at most a few hun- dred trials, the task objective was accomplished. From a machine learning point of view, this performance of a reinforcement learning algorithm is very fast. The PI2 algorithms inherits the properties of all trajectory-based learning algorithms in that it only finds locally optimal solutions. For high dimensional robotic system, this is unfortunately all one can hope for, as exploring the entire state- action space in search for a globally optimal solution is impossible.

We continue our discussion in the next subsections with some issues that deserve more detailed discussions.

6.8.1 Simplifications of $\mathbf{PI}^2$.

In this section we discuss simplifications of $\mathbf{PI}^2$. The discussions starts with research directions that may allows us to remove the assumption between control weight matrix and variance of the noise. Moreover, we show how $\mathbf{PI}^2$ could be used as model based, semi model based of model free way. Finally, we discuss some rules for cost function design as well as how $\mathbf{PI}^2$ handles hidden states in the state vector and arbitrary states in the cost function.

6.8.2 The assumption $\lambda \mathbf{R}^{-1} = \mathbf{\Sigma}_{\boldsymbol{\epsilon}}$

In order to obtain linear 2^{nd} order differential equations for the exponentially transformed HJB equations, the simplification $\lambda \mathbf{R}^{-1} = \mathbf{\Sigma}_{\boldsymbol{\epsilon}}$ was applied. Essentially, this assumption couples the control cost to the stochasticity of the system dynamics, i.e., a control with

high variance will have relatively small cost, while a control with low variance will have relatively high cost. This assumption makes intuitively sense as it would be mostly unreasonable to attribute a lot of cost to an unreliable control component. Algorithmically, this assumption transforms the Gaussian probability for state transitions into a quadratic command cost, which is exactly what our immediate reward function postulated. Future work may allow removing this simplification by applying nonlinear versions of the Feynman-Kac Lemma.

6.8.3 Model-based, Hybrid, and Model-free Learning

Stochastic optimal control with path integrals makes a strong link to the dynamic system to be optimized – indeed, originally, it was derived solely as model-based method. As this paper demonstrated, however, this view can be relaxed. The roll-outs, needed for computing the optimal controls, can be generated either from simulating a model, or by gathering experience from an actual system. In the latter case, only the control transition matrix of the model needs be known, such that we obtain a hybrid model-based/model-free method. In this work, we even went further and interpreted the stochastic dynamic system as a parameterized control policy, such that no knowledge of the model of the control system was needed anymore – i.e., we entered a model-free learning domain. It seems that there is a rich variety of ways how the path integral formalism can be used in different applications.

Further simplifications of $\mathbf{PI}^2$ can be consider if one substitutes the optimal controls to stochastic dynamics. More precisely the optimal controls are expressed as:

$$\mathbf{u}\left(\boldsymbol{\tau}_i\right) dt = \mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\left(\mathbf{G}_{t_i}^{(c)}\mathbf{R}^{-1}\mathbf{G}_{t_i}^{(c)T}\right)^{-1}\sum_{k=1}^{K}\tilde{p}^{(k)}\left(\boldsymbol{\tau}_i\right)\mathbf{G}_{t_i}^{(c)}d\mathbf{w}_{t_i}^{(k)} \tag{6.40}$$

When the controls above are applied to the stochastic dunamics then they have to be multiplied by the matrix $\mathbf{G}_{t_i}^{(c)}$. This multiplication results in:

$$\mathbf{G}_{t_i}^{(c)}\mathbf{u}\left(\boldsymbol{\tau}_i\right) dt = \sum_{k=1}^{K}\tilde{p}^{(k)}\left(\boldsymbol{\tau}_i\right)\mathbf{G}_{t_i}^{(c)}d\mathbf{w}_{t_i}^{(k)} \tag{6.41}$$

The equation above suggests simplifications of $\mathbf{PI}^2$ which will be explored. As the evaluations in this chapter show, $\mathbf{PI}^2$, in its current form, has amazingly robust performance in a variety of learning robotic control tasks.

6.8.4 Rules of cost function design

The cost functions allowed in our formulations can have arbitrary state cost, but need quadratic command cost. This is somewhat restrictive, although the user can be flexible in what is defined as a command. For instance, the dynamic movement primitives (6.11) used in this paper can be written in two alternative ways:

$$\frac{1}{\tau}\dot{z}_t = f_t + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_t) \tag{6.42}$$

or

$$\frac{1}{\tau}\dot{z}_t = \begin{bmatrix}\mathbf{g}_t^T & f_t\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{\theta} \\ \\ 1\end{bmatrix} + \tilde{\boldsymbol{\epsilon}}_t\right) \tag{6.43}$$

where the new noise vector $\tilde{\boldsymbol{\epsilon}}_t$ has one additional cofficient. The second equation treats f_t as another basis function whose parameter is constant and is thus simply not updated. Thus, we added f_t to the command cost instead of treating it as a state cost.

We also numerically experimented with violations of the clean distinction between state and command cost. Equation (6.23) could be replaced by a cost term, which is an arbitrary function of state and command. In the end, this cost term is just used to differentiate the different roll-outs in a reward weighted average, similarly as in (Peters & Schaal 2008*a*, Kober & Peters 2009). We noticed in several instances that $\mathbf{PI}^2$ continued to work just fine with this improper cost formulation.

Again, it appears that the path integral formalism and the $\mathbf{PI}^2$ algorithm allow the user to exploit creativity in designing cost functions, without absolute need to adhere perfectly to the theoretical framework.

6.8.5 Dealing with hidden state

Finally, it is interesting to consider in how far $\mathbf{PI}^2$ would be affected by hidden state. Hidden state can either be of stochastic or deterministic nature, and we consider hidden state as adding additional equations to the system dynamics (4.2).

Section 4.2 already derived that deterministic hidden states drop out of the $\mathbf{PI}^2$ update equations – these components of the system dynamics were termed "uncontrolled" equations.

More interesting are hidden state variables that have stochastic differential equations, i.e., these equations are uncontrolled but do have a noise term and a non-zero corresponding coefficient in $\mathbf{G}_t$ in equation (4.2), and these equations are coupled to the other

equations through their passive dynamics. The noise term of these equations would, in theory, contribute terms in Equation (6.23), but given that neither the noise nor the state of these equations are observable, we will not have the knowledge to add these terms. However, as long as the magnitude of these terms is small relative to the other terms in Equation (6.23), $\mathbf{PI}^2$ will continue to work fine, just a bit sub-optimally. This issue would affect other reinforcement learning methods for parameterized policies in the same way, and is not specific to $\mathbf{PI}^2$.

6.8.6 Arbitrary states in the cost function

As a last point, we would like to consider which variables can actually enter the cost functions for $\mathbf{PI}^2$. The path integral approach prescribes that the cost function needs to be a function of the state and command variables of the system equations (4.2). It should be emphasized that the state cost q_t can be any deterministic function of the state, i.e., anything that is predictable from knowing the state, even if we do not know the predictive function. There is a lot of flexibiliy in this formulation, but it is also more restrictive than other approaches, e.g., like policy gradients or the PoWER algorithm, where arbitrary variables can be used in the cost, no matter whether they are states or not.

We can think of any variable that we would like to use in the cost as having a corresponding differentential equation in the system dynamics (4.2), i.e., we simply add these variables as state variables, just that we do not know the analytical form of these equations. As in the previous section, it is useful to distinguish whether these states have deterministic or stochastic differential equations.

If the differential equation is deterministic, we can cover the case with the derivations from Section 4.2, i.e., we consider such an equation as uncontrolled deterministic differential equation in the system dynamics, and we already know that we can use its state in the cost without any problems as it does not contribute to the probability of a roll-out.

If the differential equation is stochastic, the same argument as in the previous section applies, i.e., the (unknown) contribution of the noise term of this equation to the exponentiated cost (6.23) needs to be small enough for $\mathbf{PI}^2$ to work effectively. Future work and empirical evaluations will have to demonstrate when these issues really matter – so far, we have not encountered problems in this regard.

Chapter 7

Neuromuscular Control

Neuromuscular control or control of bio-mechanical models is one of the areas, in which the optimal control theory has been applied with significant contributions. These contributions are related to a better understanding of bio-mechanical and neuromuscular structure in terms of its functionality and design. In this chapter we are discussing the main characteristic of bio-mechanical systems and we investigate the main challenges in modeling such systems. More precisely, in section 7.1 we present the main differences between torque driven and tendon driven systems and we discuss the alternative use of control theory. In section 7.2 we have a literature review on the skeletal mechanics modeling approaches. In section 7.3 we discuss various modeling choices regarding the high dimensionality and redundancy of neuromuscular systems.

We continue with the section 7.4 which reviews previous work on musculotendon routing models. In section 7.5 the application of optimal control to psychophysical and bio-mechanical models is discussed. In the last section 8.5 we conclude with the main points of this chapter.

7.1 Tendon driven versus torque driven actuation

The gap in the functionality and robustness between robotic and human hands has its origins in our lack of understanding of design principles based on control theoretic ideas applicable to complex biomechanical structures such as the hand. From the control theoretic standpoint, the control of a highly dimensional and nonlinear stochastic plant of the complexity of a robotic or biomechanical hand is not an easy task–which also makes it difficult to understand the neuromuscular control of the hand. To appreciate the high dimensionality, it is enough to consider that more than 35 tendons must be controlled by the nervous system (Freivalds 2000). Some critical questions that remain open are:

- What strategies does the nervous system use for moving the finger given the geometrical and mechanical characteristics of the muscular-tendon-bone structure? How sensitive these strategies are with respect to variations in the underlying dynamics and moment arm geometry?

There are few important differences between torque driven and tendons driven biomechanical structures. In particular, in tendon driven systems, the number of control variables is usually higher than the number of corresponding controls in torque driven systems. For example, for the case of the index finger, there are 7 actuating tendons which produce the required torque around the 3 joins, while in torque actuated mechanical fingers systems, 3 torque based control variables are sufficient to produce planar movements. An additional component is that, the tendon actuation is constrained since tendons can only pull and not push while in most robotic systems that are torque driven, the control variables can take negative and positive values to generate negative or positive torques

around joins. The limits or control constrains for the case of torque driven control systems are due to torque saturation. Clearly the actuation mechanism is different in tendon driven and torque driven dynamical systems. A step towards understanding the role of each tendon for the production of a movement is to control a bio-mechanical model and discover the underlying control strategies.

In order to apply a control theoretic approach, a model of the underlying neuromuscular dynamics is required. This model is usually built based on the knowledge of the physiology and anatomy of the bio-mechanical system under investigation, and it is, with no doubt, an approximation of the true dynamics. Given this " acceptable " model a control theoretic approach is used to generated the desired behavior. The main goal in this form of scientific reasoning is to generate with the use of control the same dynamic behavior with the one observed experimentally. Provided that both the experimenter and the theoretician trust the bio-mechanical model the claim is that the underlying control strategies matches the one that was used to generated the desired behavior in simulation and thus these control strategies is what the nervous systems may implement.

Clearly this is one way of making use of control theory which relies on the assumption that the model captures the main characteristics of the bio-mechanics and it is an acceptable approximation. Nevertheless, there are examples and cases of bio-mechanical systems for which there is no such a good model or if there is, then it is very sensitive with respect to parameter variations. In such cases, the use of control theory could be twofold. On one side it can be used as a verification tool of every proposed model as a candidate while one the other side it can be used to explore the sensitivity of the model with respect to critical parameters.

We will leave this discussion for the next chapter and in the next sections we are focusing on previous effort of bio-mechanical modeling based on the characteristics of skeletal mechanics and the redundancy and high dimensionality of neuromuscular systems.

7.2 Skeletal Mmechanics

In neuromuscular function studies, skeletal segments are generally modeled as rigid links connected to one another by mechanical pin joints with orthogonal axes of rotation. These assumptions are tenable in most cases, but their validity may depend on the purpose of the model. Some joints like the thumb carpometacarpal joint, the ankle and shoulder joints are complex and their rotational axes are not necessarily perpendicular [46][48], or necessarily consistent across subjects (Hollister, Buford, Myers, Giurintano & Novick 1992), (Santos & Valero-Cuevas 2006), (Cerveri, De Momi, Marchente, Lopomo, Baud-Bovy, Barros & Ferrigno 2008). Assuming simplified models may fail to capture the real kinematics of these systems (Valero-Cuevas, Johanson & Towles 2003). While passive moments due to ligaments and other soft tissues of the joint are often neglected, at times they are modeled as exponential functions of joint angles (Yoon & Mansour 1982), (Hatze 1997) at the extremes of range of motion to passively prevent hyper-rotation. In other cases, passive moments well within the range of motion could be particularly important in the case of systems like the fingers (Esteki & Mansour 1996), (Sancho-Bru, Prez-Gonzlez, Vergara-Monedero & Giurintano 2001) where skin, fat and hydrostatic pressure tend to resist flexion. Modeling of contact mechanics could be important for joints like the knee and the ankle where there is significant loading on the articulating

surfaces of the bones, and where muscle force predictions could be affected by contact pressure. Joint mechanics are also of interest for the design of prostheses, where the knee or hip could be simulated as contact surfaces rolling and sliding with respect to each other (Rawlinson & Bartel 2002),(Rawlinson, Furman, Li, Wright & Bartel 2006). Several studies estimate contact pressures using quasi-static models with deformable contact theory (e.g., (Wismans, Veldpaus, Janssen, Huson & Struben 1980),(Blankevoort, Kuiper, Huiskes & Grootenboer n.d.)). But these models fail to predict muscle forces during dynamic loading. Multibody dynamic models with rigid contact fail to predict contact pressures (Piazza & Delp 2001).

7.3 Dimensionality and redundancy

The first decision to be made when assembling a musculoskeletal model is to define dimensionality of the musculoskeletal model (i.e., number of kinematic degrees-of- freedom and the number of muscles acting on them). If the number of muscles exceeds the minimal number required to control a set of kinematic DOF, the musculoskeletal model will be redundant for some sub-maximal tasks. The validity and utility of the model to the research question will be affected by the approach taken to address muscle redundancy. Most musculoskeletal models have a lower dimensionality than the actual system they are simulating because it simplifies the mathematical implementation and analysis, or because a low-dimensional model is thought sufficient to simulate the task being analyzed. Kinematic dimensionality is often reduced to limit motion to a plane when simulating arm motion at the level of the shoulder(Abend, Bizzi & Morasso 1982),(Mussa-Ivaldi, Hogan

& Bizzi 1982), when simulating fingers flexing and extending (Dennerlein, Diao, Mote & Rempel 1998) or when simulating leg movements during gait (Olney, Griffin, Monga & McBride 1991). Similarly, the number of independently controlled muscles is often reduced (An, Chiao, Cooney & Linscheid 1985) for simplicity, or even made equal to the number of kinematic degrees-of-freedom to avoid muscle redundancy (Harding, Brandt & Hillberry 1993). While reducing the dimensionality of a model can be valid in many occasions, one needs to be careful to ensure it is capable of replicating the function being studied. For example, an inappropriate kinematic model can lead to erroneous predictions (Valero-Cuevas, Towles & Hentz 2000), (Jinha, Ait-Haddou, Binding & Herzog 2006), or reducing a set of muscles too severely may not be sufficiently realistic for clinical purposes. A subtle but equally important risk is that of assembling a kinematic model with a given number of degrees of freedom, but then not considering the full kinematic output. For example, a three-joint planar linkage system to simulate a leg or a finger has three kinematic DOF at the input, and also three kinematic degrees of freedom at the output: the x and y location of the endpoint plus the orientation of the third link. As a rule, the number of rotational degrees- of-freedom (i.e., joint angles) maps into as many kinematic degrees-of-freedom at the endpoint (Murray, Li & Sastry 1994). Thus, for example, studying muscle coordination to study endpoint location with- out considering the orientation of the terminal link can lead to variable results. As we have described in the literature (Valero-Cuevas, Zajac & Burgar 1998), (Valero-Cuevas 2009), the geometric model and Jacobian of the linkage system need to account for all input and output kinematic degrees- of-freedom to properly represent the mapping from muscle actions to limb kinematics and kinetics.

7.4 Musculotendon routing

Next, we need to select the routing of the musculotendon unit consisting of a muscle and its tendon in series (Zajac 1989), (Zajac 1992). The reason we speak in general about musculo-tendons (and not simply tendons) is that in many cases it is the belly of the muscle that wraps around the joint (e.g., gluteus maximus over the hip, medial deltoid over the shoulder). In other cases, however, it is only the tendon that crosses any joints as in the case of the patellar tendon of the knee or the flexors of the wrist. In addition, the properties of long tendons affect the overall behavior of muscle like by stretching out the force- length curve of the muscle fibers (Zajac 1989). Most studies assume correctly that musculotendons insert into bones at single points or multiple discrete points (if the actual muscle attaches over a long or broad area of bone). Musculo-tendon routing defines the direction of travel of the force exerted by a muscle when it contracts. This defines the moment arm r of a muscle about a particular joint, and determines both the excursion δs the musculo-tendon will undergo as the joint rotates an angle $\delta\theta$ defined by the equation, $\delta s = r\delta\theta$, as well as the joint torque at that joint due to the muscle force fm transmitted by the tendon $\tau = r \cdot f_m$ where r is the minimal perpendicular distance of the musculo-tendon from the joint center for the planar (scalar) case (Zajac 1992). For the three dimensional case the torque is calculated by the cross product of the moment arm with the vector of muscle force $\tau = r \times f_m$. In todays models, musculo-tendon paths are modeled and visualized either by straight lines joining the points of attachment of the muscle; straight lines connecting via points attached to specific points on the bone which are added or removed depending on joint configuration (Garner & Pandy 2000) or as cubic splines

with sliding and surface constraints (Blemker & Delp 2005.). Several advances also allow representing muscles as volumetric en- tities with data extracted from imaging studies (Blemker & Delp 2005.) ,(S. S. Blemker & Delp 2007), and defining tendon paths as wrapping in a piecewise linear way around ellipses defining joint locations (R. Davoodi & Loeb 2003), (Delp & Loan 2007). The path of the musculotendon in these cases is defined based on knowledge of the anatomy. Sometimes, it may not be necessary to model the musculotendon paths but obtaining a mathematical expression for the moment arm (r) could suffice. The moment arm is often a function of joint angle and can be obtained by recording incremental tendon excursions (δs) and corresponding joint angle changes ($\delta\theta$) in cadaveric specimens.

7.5 Discussion

The use of stochastic optimal control theory as conceptual tool towards understanding neuromuscular behavior was proposed in, for example, (He, Levine & Loeb 1991), (Harris & Wolpert 1998), (Todorov 2004). In that work, a stochastic optimal control framework for systems with linear dynamics and control-dependent noise was used to understand the variability profiles of reaching movements. The influential work by (Todorov 2004) established the minimal intervention principle in the context of optimal control. The minimal interven- tion principle was developed based on the characteristics of stochastic optimal controllers for systems with multiplicative noise in the control signals.

The LQR and LQG optimal control methods have been mostly tested on linear dynamical systems for modeling sen- sorimotor behavior; e.g, in reaching tasks, linear models

were used to describe the kinematics of the hand trajectory (Harris & Wolpert 1998), (Todorov & Jordan 2002). In neuromuscular modeling, however, linear models cannot capture the nonlinear behavior of muscles and multi- body limbs. In (Li & Todorov 2004), an Iterative Linear Quadratic Regulator (ILQR) was first introduced for the optimal control of nonlin- ear neuromuscular models. The proposed method is based on linearization of the dynamics. An interesting component of this work that played an influential role in the studies on optimal control methods for neuromuscular models was the fact that there was no need for a pre-specified desired trajectory in state space.

By contrast, most approaches for neuromuscular optimization that use classical control theory (see Section VI) require target time histories of limb kinematics, kinetics and/or muscle activity. In (Todorov 2005) the ILQR method was extended for the case of nonlinear stochastic systems with state and control dependent noise. The proposed algorithm is the Iterative Linear Quadratic Gaussian Regulator (iLQG). This extension allows the use of stochastic nonlinear models for muscle force as a function of fiber length and fiber velocity. Figure 6 illustrates the application of LQG to our arm model (Section II). Further theoretical developments in (Li & Todorov 2006) and (Todorov 2007) allowed the use of an Extended Kalman Filter (EKF) for the case of sensory feedback noise. The EKF is an extension of the Kalman filter for nonlinear systems.

The has been only few examples of studies in the area of the biomechanics of the index finger which try to identity the underlying control signals for the case of movement and force production, either these signals corresponds to neural commands or tensions applied on the tendons. More precisely on the experimental side, the work in (Venkadesan & Valero-Cuevas 2008*b*) investigated the neural control of contact transition between

motion and force during tapping. On the theoretical side the study in (Venkadesan & Valero-Cuevas 2008*a*) has found that such transitions from motion to well-directed contact force are a fundamental part of dexterous manipulation, and that such tasks are likely controlled optimally. Moreover, one of the main assumptions in (Venkadesan & Valero-Cuevas 2008*a*) is that the underlying control strategy of the finger is considered to be open loop. In addition, the model used is a torque driven model while the neuromuscular delays are modeled as activation contraction dynamics at the level of the torques driving the 3 joints of the index finger. Even though with this simple model the optimality principles of the motion to force transition for the task of tapping were investigated, an open loop control strategy would have failed in tasks such as object manipulation where feedback control is critical requirement for successfully performing the manipulation task. Furthermore, since only 3 sets of differential equation that model the activation contraction dynamics are considered, the full structure and redundancy of the index finger is not explored and the system under investigation remains in nature torque driven.

In this chapter we have reviewed previous work on bio-mechanical modeling by touching the critical issues of skeletal mechanics, muscle redundancy and musculotendon routing as well as on application of optimal control theory to psychophysical and neuromuscular models. We have provided the main differences between torque driven and tendon driven systems. We have discussed the role of the use of control theory into bio-mechanical models not only as a tool that provides insights regarding the underlying control strategies put also as a way to verify bio-mechanical models through a sensitivity analysis.

Following this line of reasoning, in the next chapter we apply the optimal control theory to two tendon driven models of the index finger.

Chapter 8

Control of the index finger

In this chapter we apply the iterative optimal control algorithm on two bio-mechanical models of the index finger and we compare the resulting behavior. The bio-mechanical models share the same multi-body dynamics but they differ in the tendon geometry since they incorporate different moment arm matrices found in (Valero-Cuevas et al. 1998) and (An, Ueba, Chao, Cooney & Linscheid 1983). As it is illustrated, the different moment arm matrices play important role in the actuation capabilities of each model of the index finger which become obvious as we compare the underlying tension profiles for the case a flexing and a tapping movement.

The remaining of this chapter is organized as follows: in section 8.1 we provide a short introduction for the biomechanics of the index finger while in section 8.2 we discuss the iterative linear quadratic regulator which is the optimal control algorithm used for our simulations. In section 8.3 we provide the multi-body dynamics and in 8.4 we compare our results on the optimal control of the index finger between the two models of the moment arm matrices. The moment arm models and the optimal control algorithm are tested on the tasks of flexing and tapping with the index finger.

8.1 Index fingers biomechanics

The skeleton of the human index finger consist of 3 joints connected with 3 rigid links. The two joints (proximal interphalangeal (PIP) and the distal interphalangeal (DIP)) are described as hinge joints that can generate both flexion-extension. The metacarpophalangeal joint (MCP) is a saddle joint and it can generated flexion-extension as well as abduction-adduction.

Fingers have at least 6 muscles, and the index finger is controlled by 7. Starting with the flexors, the index finger has the Flexor Digitorum Profundus (FDS) and the Flexor Digitorum Superficialis (FDP). The the Radial Interosseous (RI) acts on the MCP joint. Lastly, the extensor mechanism acts on all three joints. It is an interconnected network of tendons driven by two extensors Extensor Communis (EC) and the Extensor Indicis (EI), and the Ulnar Interosseous (UI) and Lumbrical (LU). There are also 4 passive tendon elements that complete this network. These passive tendons are the Terminal Extensor (TE), the Radial Band (RB) the Ulnar Band (UB) and the Extensor Slip (ES).

Active tendons are connected to muscles and therefore they directly actuate the finger. Passive tendons are connected with other tendons(active) and ligaments and their role for the case of the index finger is to transform the applied tensions to the distal join. In our work we will consider only the active tendons.

8.2 Iterative stochastic optimal control

We consider the nonlinear dynamical system described by the stochastic differential equation that follows:

$$\mathbf{dx} = f(\mathbf{x}, \mathbf{u})dt + F(\mathbf{x}, \mathbf{u})d\mathbf{w}$$

where $\mathbf{x} \in \Re^{n\times 1}$ is the state, $\mathbf{u} \in \Re^{m\times 1}$ is the control and $\mathbf{w} \in \Re^{p\times 1}$ Brownian motion noise with variance $\sigma^2 I_{p\times p}$. The stochastic differential equation above corresponds to a rather general class of dynamical systems which are found in robotics and biomechanics. The term $h(\mathbf{x}(T))$ is the terminal cost in the cost function while the $\ell\left(\tau, \mathbf{x}(\tau), \pi(\tau, \mathbf{x}(\tau))\right)$ is the instantaneous cost rate which is a function of the state $\mathbf{x}$ and control policy $\pi(\tau, \mathbf{x}(\tau))$. The cost-to - go $v^{\pi}(\mathbf{x}, t)$ is defined as the expected cost accumulated over the time horizon $(t_0, ..., T)$ starting from the initial state $\mathbf{x}_t$ to the final state $\mathbf{x}(T)$.

$$v^{\pi}(\mathbf{x}, t) = E\left[h(\mathbf{x}(T)) + \int_{t_0}^{T} \ell\left(\tau, \mathbf{x}(\tau), \pi(\tau, \mathbf{x}(\tau))\right) d\tau\right]$$

The expectation above is taken over the noise ω. We next discretize the deterministic dynamics and therefore we will have $\bar{\mathbf{x}}_{t_{k+1}} = \bar{\mathbf{x}}_{t_k} + \Delta t f(\bar{\mathbf{x}}_{t_k}, \bar{\mathbf{u}}_{t_k})$. Furthermore the deterministic dynamics are linearized according to the equation that follows around $\bar{\mathbf{x}}_{t_k}$

$$\delta\mathbf{x}_{t_{k+1}} + \bar{\mathbf{x}}_{t_{k+1}} = \bar{\mathbf{x}}_{t_k} + \delta\bar{\mathbf{x}}_{t_k} + \Delta t f(\bar{\mathbf{x}}_{t_k} + \delta\bar{\mathbf{x}}_{t_k}, \bar{\mathbf{u}}_{t_k} + \delta\bar{\mathbf{u}}_{t_k})$$

The first order approximation of the nonlinear dynamics leads the linearized dynamics:

$$\delta \mathbf{x}_{t_{k+1}} = A_k \mathbf{x}_{t_k} + B_k \delta \mathbf{u}_{t_k} + \Gamma_k \left(\delta \mathbf{u}_{t_k} \right) \boldsymbol{\xi}_{t_k}$$

where Γ_k is the noise transition matrix that is control depended and it is defined as follows:

$$\Gamma_k \left(\delta \mathbf{u}_{t_k} \right) = \left[\begin{array}{ccc} \mathbf{c}_{1,k} + C_{1,k} \delta \mathbf{u}_{t_k} & \cdots & \mathbf{c}_{p,k} + C_{p,k} \delta \mathbf{u}_{t_k} \end{array} \right]$$

with $\mathbf{c}_{i,k} = \sqrt{dt} F^{(i)}$ and $C_{i,k} = \sqrt{dt} \partial F^{(i)} / \partial \delta \mathbf{u}$. The state and control transition matrices are expressed as: $A_k = I + dt \partial \mathbf{f} / \partial \mathbf{x}$ and $B_k = dt \partial \mathbf{f} / \partial \mathbf{u}$. The quadratic approximation of the cost function is given as follows:

$$\begin{aligned} Cost_k = q_k + \delta \mathbf{x}_{t_k}^T \mathbf{q} + \frac{1}{2} \delta \mathbf{x}_{t_k}^T Q_k \mathbf{x}_{t_k} \\ + \delta \mathbf{u}_{t_k}^T \mathbf{r} + \frac{1}{2} \delta \mathbf{u}_{t_k}^T R_k \mathbf{u}_{t_k} + \delta \mathbf{x}_{t_k}^T P_k \mathbf{u}_{t_k} \end{aligned} \tag{8.1}$$

where the terms : $q_k, \mathbf{q}_k \in \Re^{n \times 1}, Q_k \in \Re^{n \times n}, \mathbf{r}_k \in \Re^{m \times 1}, R_k \in \Re^{m \times m}, P_k \in \Re^{n \times m}$ are defined as:

$$q_k = dt \;\; \ell; \quad \mathbf{q}_k = dt \; \partial \ell / \partial \mathbf{x} \tag{8.2}$$

$$Q_k = dt \; \partial^2 \ell / \partial \mathbf{x} \partial \delta \mathbf{x}; \quad P_k = dt \; \partial^2 \ell / \partial \mathbf{u} \partial \mathbf{x} \tag{8.3}$$

$$\mathbf{r}_k = dt \; \partial \ell / \partial \delta \mathbf{u}; \quad R_k = dt \; \partial^2 \ell / \partial \mathbf{u} \partial \mathbf{u} \tag{8.4}$$

The cost to go $v_k(\delta\mathbf{x})$ is quadratic of the state and therefore it has the form:

$$v_k(\delta\mathbf{x}) = s_k + \mathbf{s}_{k+1}^T\delta\mathbf{x} + \delta\mathbf{x}^T S_{k+1}\delta\mathbf{x} \tag{8.5}$$

Where the terms $s_k, \mathbf{s}_{k+1}$ and S_{k+1} are backward propagated from the terminal or goal state to the initial state. More precisely starting with the terminal conditions $s_{k+1} = q_T, \mathbf{s}_{k+1} = \mathbf{q}_T$ and $S_{k+1} = Q_T$, for $k = T-1$ we find the following terms:

$$\begin{aligned}
\mathbf{g} &= \mathbf{r}_k + B_k^T s_{k+1} + \sigma^2 \sum_i C_{i,k}^T S_{k+1} c_{i,k} \\
G &= P_k + B_k^T S_{k+1} A_k \\
H &= \sigma^2 \sum_i C_{i,k}^T S_{k+1} C_{i,k} + B_k^T S_{k+1} B_k + R_k
\end{aligned} \tag{8.6}$$

By using the terms above the we can now calculate the correction in the control policy $\delta\mathbf{u}_{t_k}$ is formulated as $\delta\mathbf{u}_{t_k} = -H^{-1}(\mathbf{g} + G\delta\mathbf{x}_{t_k})$ or in a more compact form $\delta\mathbf{u}_{t_k} = l_k + L_k\delta\mathbf{x}_{t_k}$ where $l_k = -H^{-1}\mathbf{g}$ and $L_k = -H^{-1}G$. As we can see the correction in the control policy consist of an open loop gain l_k and a close loop gain L_k which guarantees local stability around the point of linearization of the nonlinear dynamics. Since the open and close loop gains l_k and L_k have been specified the next step is the backward propagation of the terms $s_k, \mathbf{s}_{k+1}$ and S_{k+1}. This backward propagation is expressed by the equations that follow:

$$
\begin{aligned}
S_k &= Q_k + A_k^T S_{k+1} A_k + L_k^T H L_k + L^k G + G^T L_k \\
\mathbf{s}_k &= \mathbf{q}_k + A_k^T \mathbf{s}_{k+1} + L_k^T H l_k + G^T L_k + L_k^T \mathbf{g} \\
s_k &= q_k + s_{k+1} + \frac{1}{2}\sigma^2 \sum_i c_{i,k}^T S_{k+1} c_{i,k} + \frac{1}{2} l_k^T H l_k + l_k^T \mathbf{g}
\end{aligned}
\tag{8.7}
$$

The control policy at the next iteration is given by the adding the correction $\delta \mathbf{u}_{t,\ldots,T}^{(i)}$ in the control policy of the current iteration. Therefore we will have that $\mathbf{u}_{t,\ldots,T}^{(i+1)} = \mathbf{u}_{t,\ldots,T}^{(i)} + \gamma \cdot \delta \mathbf{u}_{t,\ldots,T}^{(i)}$ where γ is the step size. Using the updated control policy $\mathbf{u}_{t,\ldots,T}^{(i+1)}$ and by propagating the nonlinear dynamics a new trajectory is generated in state space. The linear and quadratic approximation of the dynamics and cost are found and the algorithms is repeated again until convergence. The control law $\delta \mathbf{u}_{t_k} = -H^{-1}\left(\mathbf{g} + G\delta \mathbf{x}_{t_k}\right)$ is the optimal one for as long as the matrix H is positive definite. The cost-to -go function $v_\pi\left(\delta \mathbf{x}\right)$ depends on the control law $\delta \mathbf{u}_k = \boldsymbol{\pi}_k\left(\delta \mathbf{x}\right)$ through the term $\alpha(\delta \mathbf{x}, \delta \mathbf{u}) = \delta \mathbf{u}^T(\mathbf{g} + G\delta \mathbf{x}) + \frac{1}{2}\delta \mathbf{u}^T H \delta \mathbf{u}$. Therefore minimization of the cost to go function is equivalent to the minimization of the quadratic function $\alpha(\delta \mathbf{x}, \delta \mathbf{u})$ which is convex iff the its Hessian $H > 0$. In highly dimensional dynamical systems H might loose its positive definiteness. In such cases we follow an approach similar to Levenberg-Marquardt : (1) compute the eigenvalue decomposition of H, $[V, D] = eig(H)$(2) replace all the negative elements of the diagonal matrix with 0 (3) add a small positive number λ to the diagonal of D (4) set $H = VDV^T$ using the modified diagonal matrix D from the steps (2) and (3). For our simulation we need to constrain the controls $\mathbf{u}$ since the control variable of our index finger model corresponds to neural activation that is always positive. To avoid violating

Table 8.1: Pseudocode of the iLQG algorithm

- **Given**:
  - An immediate cost function $\ell(\mathbf{x}, \mathbf{u})$
  - A terminal cost term ϕ_{t_N}.
  - The stochastic dynamics $\mathbf{dx} = f(\mathbf{x}, \mathbf{u})dt + F(\mathbf{x}, \mathbf{u})\mathbf{d}\omega$
- **Repeat** until convergence:
  - Given a trajectory in states and controls $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ find the approximations A_t, B_t, Γ_t and $\ell_o, \ell_{\mathbf{x}}, \ell_{\mathbf{xx}}, \ell_{\mathbf{uu}}, \ell_{\mathbf{ux}}$ around these trajectories.
  - Compute all the terms H, G and g according to equations (8.6).
  - Back-propagate the quadratic approximation of the value function based on the equations (8.7).
  - Compute $\delta\mathbf{u}_{t_k} = -H^{-1}\left(\mathbf{g} + G\delta\mathbf{x}_{t_k}\right)$
  - Update controls $\mathbf{u}^*_{new} = \mathbf{u}^*_{old} + \gamma \cdot \delta\mathbf{u}^*$
  - If $\mathbf{u}^*_{new} < \mathbf{u}_c$ then reduce γ to γ_c so that the constraint is not violated and find the controls $\mathbf{u}^*_{new} = \mathbf{u}^*_{old} + \gamma_c \cdot \delta\mathbf{u}^*$
  - Get the new optimal trajectory x^* by propagating the nonlinear dynamics $\mathbf{dx} = f(\mathbf{x}, \mathbf{u}^*)dt + F(\mathbf{x}, \mathbf{u}^*)\mathbf{d}\omega$.
  - Set $\bar{\mathbf{x}} = \mathbf{x}^*$ and $\bar{\mathbf{u}} = \mathbf{u}^*_{old} = \mathbf{u}^*_{new}$ and repeat.

the control constrains the step size γ is reduced until the constrain is not violated. The iLQG algorithm in a pseudocode form is illustrated in table (8.1).

8.3 Multi-body dynamics

The full model of the index finger is given by the equations that follow:

$$\ddot{\boldsymbol{\theta}} = -\mathbf{I}\left(\boldsymbol{\theta}\right)^{-1}\mathbf{C}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) + \mathbf{B}\dot{\boldsymbol{\theta}} + \mathbf{I}\left(\theta\right)^{-1}\mathbf{T} \tag{8.8}$$

$$\mathbf{T} = \mathbf{M}(\theta)\ \mathbf{F} \tag{8.9}$$

$$\dot{\mathbf{F}} = -\frac{1}{\tau}(\mathbf{F} - \mathbf{u}) \tag{8.10}$$

where $\mathbf{I} \in \Re^{3\times3}$ is the inertial matrix, $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \Re^{3\times1}$ is matrix of Coriolis and centripetal forces and $\mathbf{B} \in \Re^{3\times3}$ is the damping matrix. The matrix $\mathbf{M} \in \Re^{3\times7}$ is the moment-arm matrix, $\mathbf{T} \in \Re^{3\times1}$ is the torque vector, $\mathbf{F} \in \Re^{7\times1}$ is the force-tension on the tendons and $\mathbf{u}$ is the control vector. Equation (8.10) is used to model delays in the generation of tensions on the tendons. For our simulations we have excluded the abduction-adduction movement at MCP joint and we examine planar movements and we investigate the necessary length and velocity profiles of the tendons for producing such movements. Therefore, the state space formulation of our model has dimensionality of 13, corresponding to 6 states related to joint space kinematics (angles and velocities) and 7 states for the tensions applied on the 7 active tendons. The quantities $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ are vectors of dimensionality $\boldsymbol{\theta} \in \Re^{3\times1}$,$\dot{\boldsymbol{\theta}} \in \Re^{3\times1}$ defined as $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ and $\dot{\boldsymbol{\theta}} = \left(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3\right)$. The inertia $\mathbf{I}(\boldsymbol{\theta})$terms of the forward dynamics are given as follows:

$$\begin{aligned}
I_{11} &= I_{31} + \mu_1 + \mu_2 + 2\mu_4 \cos\theta_2 \\
I_{21} &= I_{22} + \mu_4 \cos\theta_2 + \mu_6 \cos(\theta_2 + \theta_2) \\
I_{22} &= I_{33} + \mu_2 + 2\mu_5 \cos\theta_3 \\
I_{31} &= I_{32} + \mu_6 \cos(\theta_3 + \theta_3) \\
I_{33} &= \mu_3
\end{aligned}$$

while the term of coriolis and centripetal forces $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is formulated as follows:

$$
\begin{aligned}
C_1 &= \mu_4 \sin\theta_2 \left[-\dot{\theta}_2 \left(2\dot{\theta}_1 + \dot{\theta}_2\right)\right] + \mu_5 \sin\theta_3 \left[-\dot{\theta}_3 \left(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3\right)\right] \\
&\quad - \mu_6 \sin\left(\theta_2 + \theta_3\right) \left(\dot{\theta}_2 + \dot{\theta}_3\right) \left(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right) \\
C_2 &= \mu_5 \sin\theta_2 \dot{\theta}_1^2 - \mu_5 \sin\theta_3 \left[\dot{\theta}_3 \left(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right)\right] \\
&\quad + \mu_6 \sin\left(\theta_2 + \theta_3\right) \dot{\theta_1}^2 \\
C_3 &= \mu_5 \sin\theta_3 \left(\dot{\theta}_1 + \dot{\theta}_2\right) + \mu_6 \sin\left(\dot{\theta}_2 + \dot{\theta}_3\right) \dot{\theta}_1^2
\end{aligned}
$$

The terms μ_1, μ_2, μ_3 are functions of the masses $(m_1, m_2, m_3) = (0.05, 0.04, 0.03)\, Kgr$ and the lengths $(l_1, l_2, l_3) = (0.0508, 0.0254, 0.01905)\, m$ of the 3 bones of the index finger. They are specified as $mu_1 = (m_1 + m_2 + m_3)\,, \mu_1 = (m_1 + m_2 + m_3)\, l_1^2, \mu_3 = m_3 l_3^2, \mu_4 = (m_2 + m_3)\, l_1 l_2, \mu_5 = m_3 l_2 l_3$ and $\mu_6 = m_3 l_1 l_3$.

8.4 Effect of the moment arm matrices in the control of the index finger

In this section we apply optimal control framework to a bio-mechanical model of the index and we are testing the effect of different moment arm matrices in the control of the index finger. In our analysis we used the moment arm matrix suggested in (An et al. 1983) and(Valero-Cuevas et al. 1998) . We apply iLQG optimal control to generate the two movements and we compare the behaviors of the two models.

8.4.1 Flexing movement

The first movement is a flexion movement around the PIP and DIP joins while the MCP join remains almost constant. The initial posture is at $\theta_0 = (0, 0, \pi/10,)$ and the terminal posture is at $\theta_N = (0, \pi/2, \pi/12,)$ while the time horizon of the movement is $T_N = 400ms$ The cost function is tuned such that it penalizes only terminal errors with respect to the target posture and control cost. Therefore, we do not pre-specify any desire trajectory that would have imposed extra state depended terms in the cost function.

The flexion and tapping movement correspond to control problems where the goal is to bring the dynamics form an initial state to a target state. The iterative optimal control algorithms provide us with the optimal control sequence $\mathbf{u}$, a set of locally optimal closed loop gains $\mathbf{L}$ and an the locally optimal state space trajectory. This trajectory in treated as a desired trajectory that is followed by the dynamics with the use of the open loop control $\mathbf{u}$ and the feedback policies $\mathbf{L}$. Essentially, we leave the optimization procedure to come-up with each one desired trajectory.

An alternative to this approach would be to record join kinematic trajectories and then use these trajectories in the cost function. In particular, in this scenario we would have to impose extra terms in the cost function which penalize errors with respect to any deviation from the desired trajectory. In both cases scenarios, the iterative optimal controller is a tracking controller, the deference between the two cases in whether or not the desired trajectory is pre-specified or it is the outcome of the optimization procedure. In the figures that follows the postures, the kinetics of the tendons and the underlying tension profiles are illustrated.

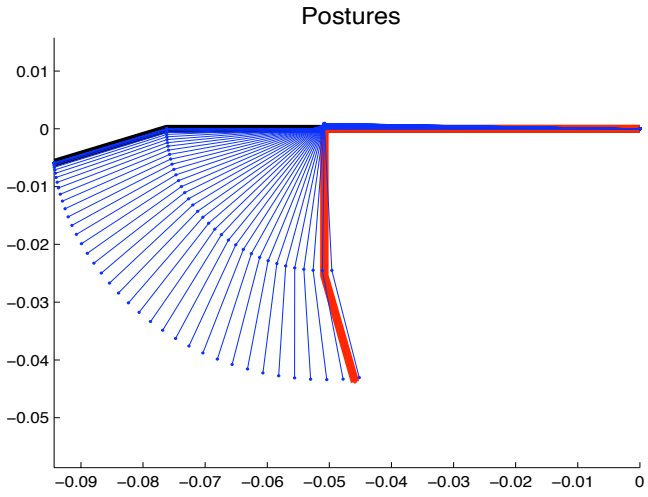


Figure 8.1: Flexing Movement: Sequence of postures generated when the first model of moment arm matrix is used and the iLQG is applied

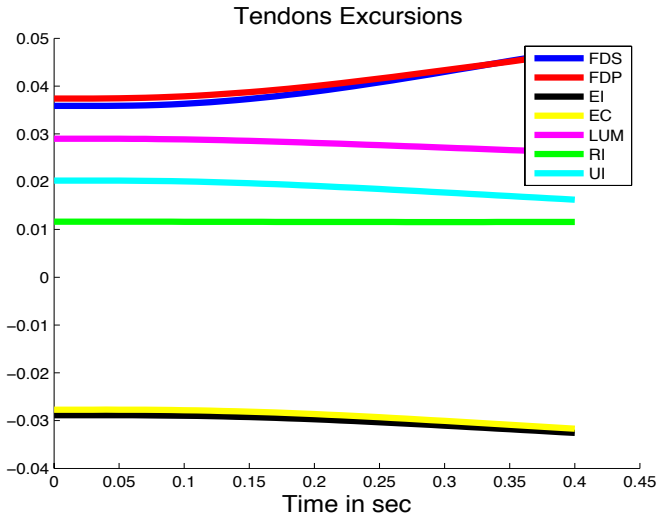


Figure 8.2: Flexing Movement: Tendon excursions for the right index finger during the flexing movement when the first model of moment arm matrix.

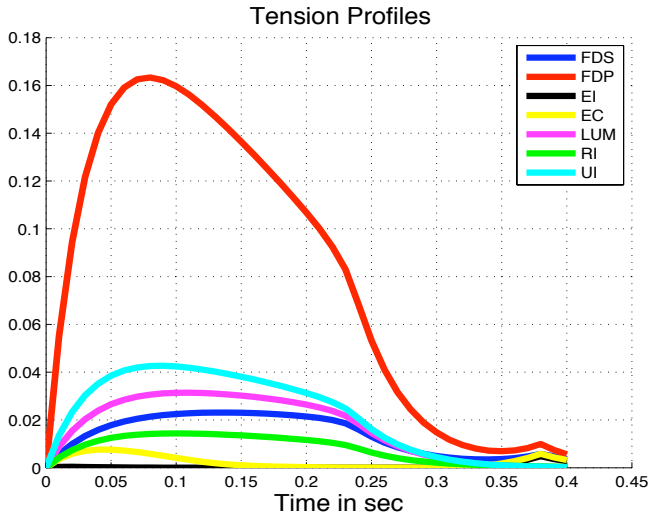


Figure 8.3: Flexing Movement: Tension profiles applied to the right index finger when the first model of moment arm matrix by is used.

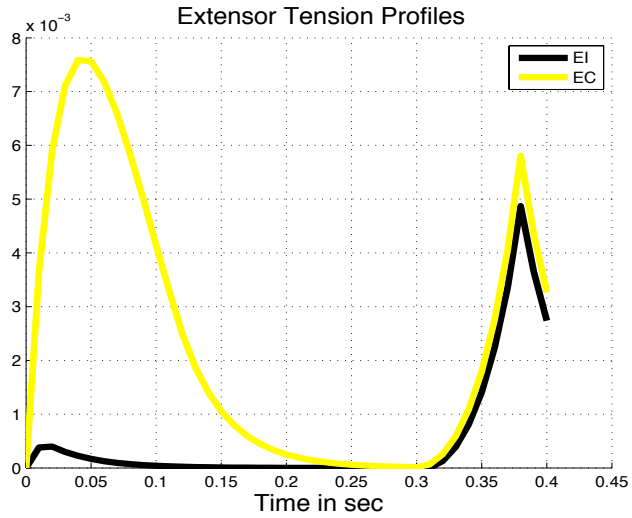


Figure 8.4: Flexing Movement: Extensor tension profiles applied to the right index finger when the first model of moment arm matrix is used.

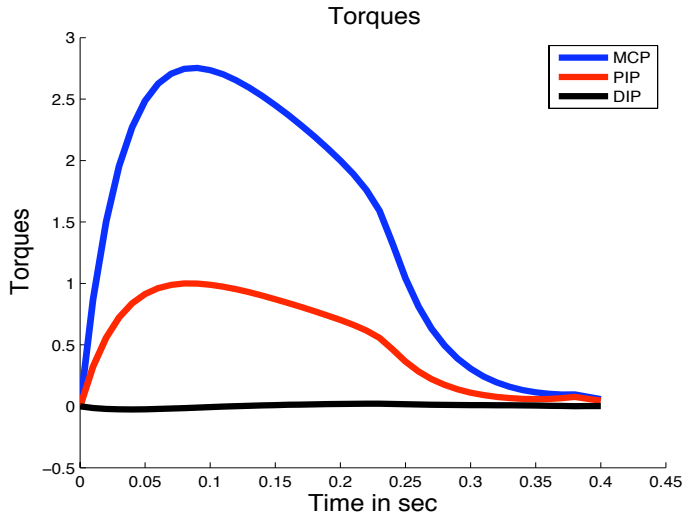


Figure 8.5: Flexing Movement: Generated torques at MCP, PIP and DIP joins of the right index finger when the first model of moment arm matrix is used.

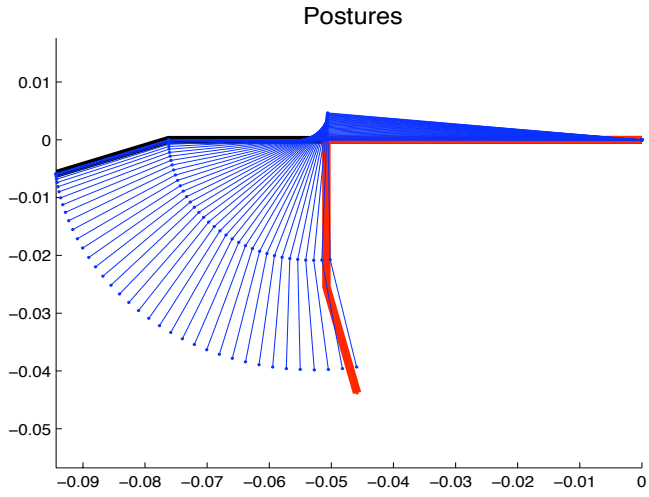


Figure 8.6: Flexing Movement: Sequence of postures generated when the second model of moment arm matrix is used and the iLQG is applied.

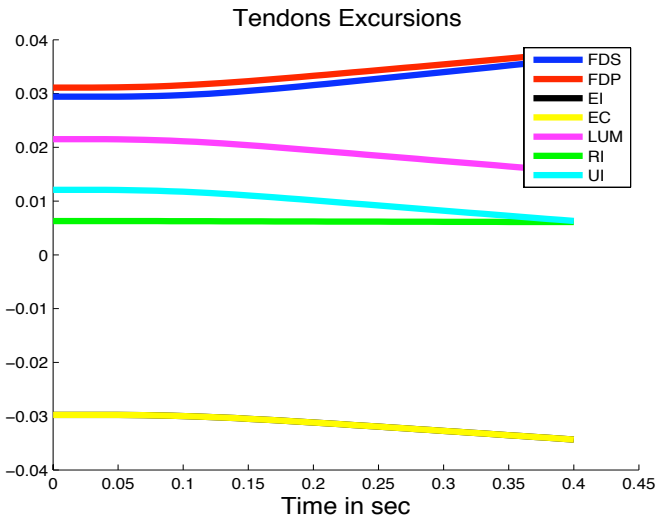


Figure 8.7: Flexing Movement: Tendon excursions for the right index finger during the flexing movement when the second model of moment arm matrix..

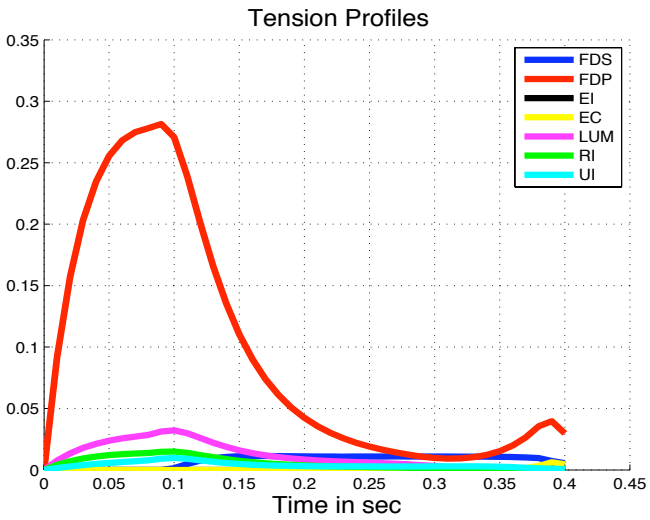


Figure 8.8: Flexing Movement: Tension profiles applied to the right index finger when the second model of moment arm matrix by is used.

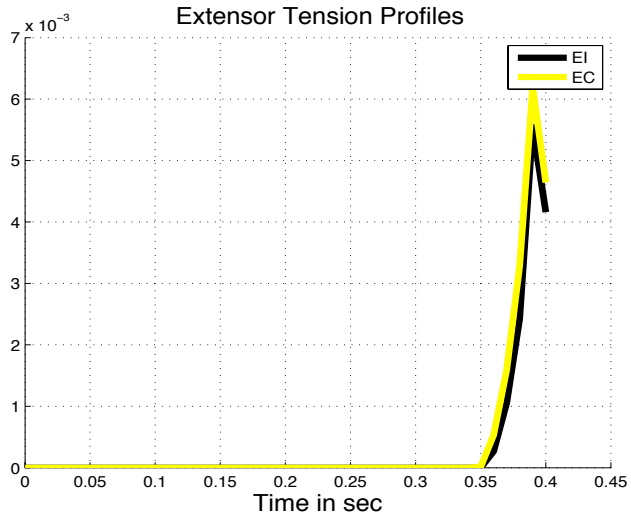


Figure 8.9: Flexing Movement: Extensor tension profiles applied to the right index finger when the second model of moment arm matrix is used.

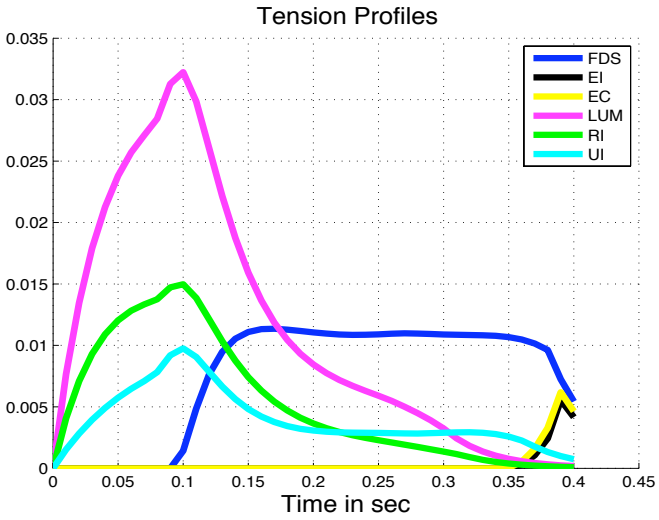


Figure 8.10: Flexing Movement: Flexors tension profiles applied to the right index finger when the second model of moment arm matrix is used.

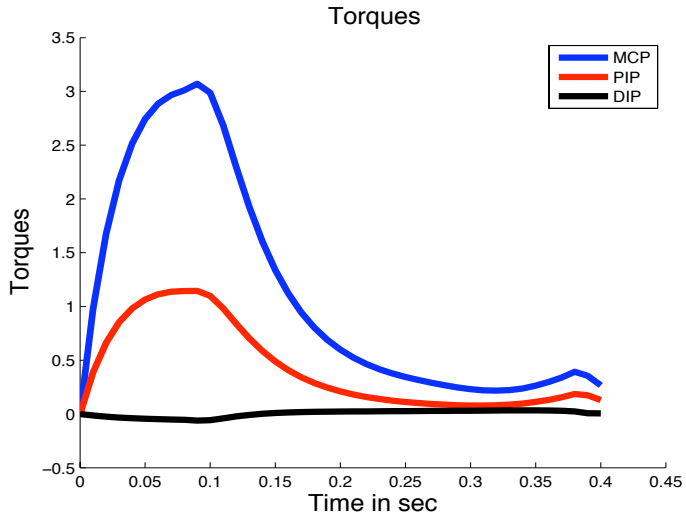


Figure 8.11: Flexing Movement: Generated torques at MCP, PIP and DIP joins of the right index finger when the second model of moment arm matrix is used.

There are few important observations regarding the kinematic behaviors and the underlying tension profiles when the two different moment arm matrices are used. More precisely:

- Figures 8.1 and 8.6 illustrate the sequence of postures for the two moment arms. In both cases the iLQG succeeds in bringing the finger to the desired posture. When the moment arm by (Valero-Cuevas et al. 1998) is used then there is a small rotation at the MCP joint which is not observed for the case when the moment arm matrix by (An et al. 1983) is used.

- In figures 8.2 and 8.7 the tendon excursions are illustrated. More precisely in both cases we see that the tendon flexors FDP and FDS move inwards and therefore the

corresponding tendons are flexing as expected. Correspondingly the tendon excursions EC and EI for are moving outwards and thus operate as expected. Moreover the tendons LUM, RI and UI move outwards as it is illustrated in the two figures.

- In figures 8.3,8.4 and 8.8,8.9 the tensions applied on the 7 tendons to generate the flexing movement are shown. Clearly for the case of the first moment arm there is a synchronized burst of activity since all the tensions are reaching their maximum tensions during the time window between 0ms and 0.2 ms. For the case of the second moment arm, the results in 8.8, do not illustrated a burst of activity but they rather suggest a different mechanism which is characterized by a higher tensions in the FDP tendon with respect to the rest tendons, and a delay in the activation of the FDS and EI, EC tendons as it is shown in figure 8.9.

- The torque profiles are illustrated in figures 8.5 and 8.11. As it is illustrated the torque profiles are very similar since in both cases the highest torque is generated around the MCP join and the smallest around the DIP join. The torques applied at the MCP and DIP join for the first moment arm reach a smaller pick than the corresponding pick reached by MCP and DIP torques for the second moment arm matrix. Furthermore the torques for the first moment arm 8.5 are changing over time in smoother fashion than the torques in 8.11.

In the next subsection we will continue our sensitivity analysis for the case of the tapping movement and we are testing again the two moment arm matrices.

8.4.2 Tapping Movement

The second movement corresponds to tapping with the index finger. The initial posture is at $\theta_0 = (5\pi/6, \pi/2, \pi/10,)$ and the terminal posture is at $\theta_0 = (7\pi/6, \pi/4, \pi/12,)$ while the time horizon of the movement is 300ms. The cost function is tuned such that it penalizes only terminal errors with respect to the target posture and control cost. In the figures that follows the postures, the kinetics of the tendons and the underlying tension profiles are illustrated.

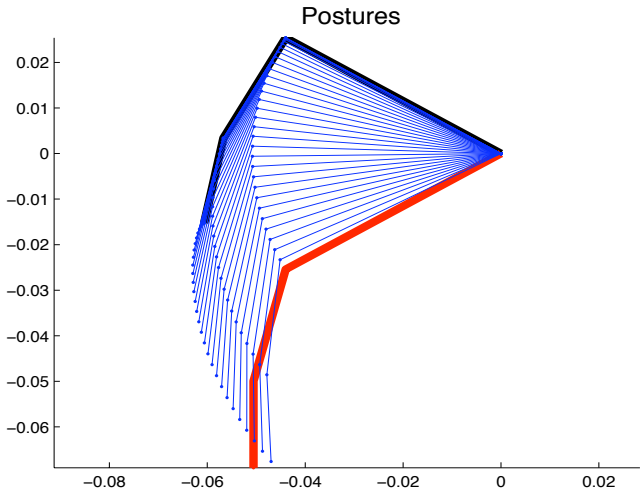


Figure 8.12: Tapping Movement: Sequence of postures generated when the first model of moment arm matrix is used and the iLQG is applied.

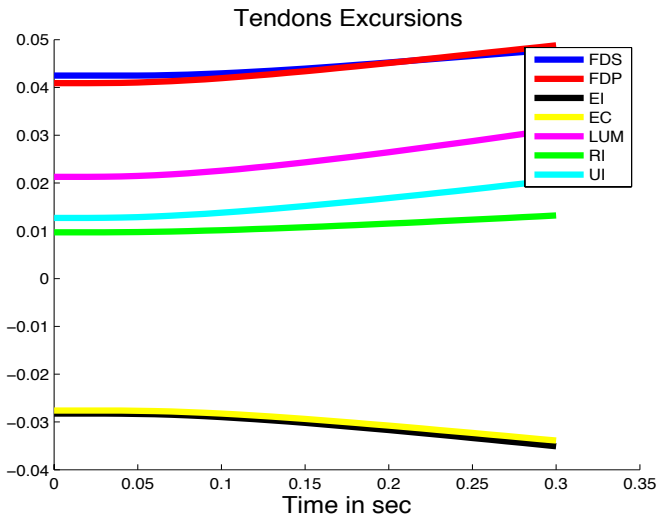


Figure 8.13: Tapping Movement: Tendon excursions for the right index finger during the flexing movement when the first model of moment arm matrix.

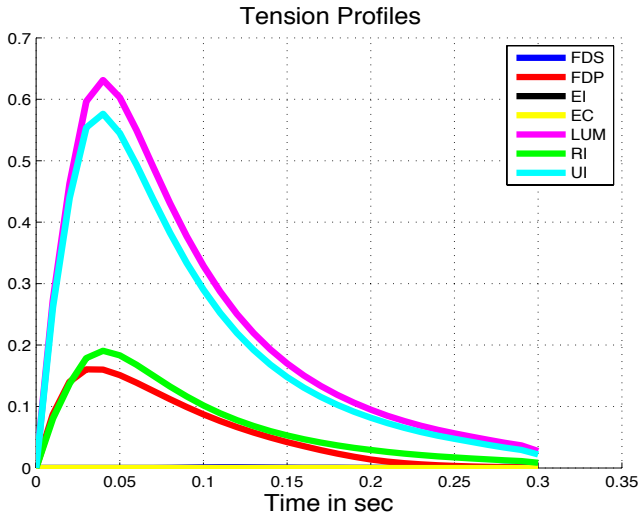


Figure 8.14: Tapping Movement: Tension profiles applied to the right index finger when the first model of moment arm matrix by is used.

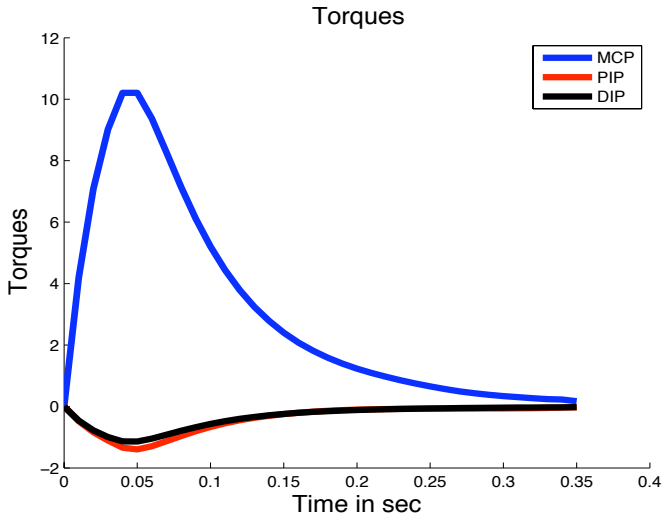


Figure 8.15: Tapping Movement: Generated torques at MCP, PIP and DIP joins of the right index finger when the first model of moment arm matrix is used.

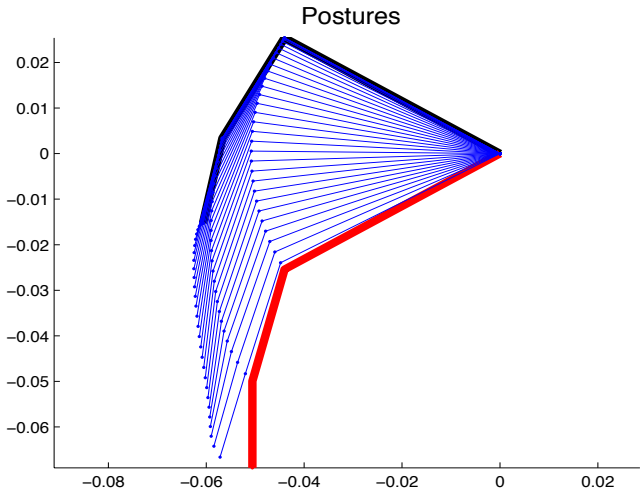


Figure 8.16: Tapping Movement: Sequence of postures generated when the second model of moment arm matrix is used and the iLQG is applied.

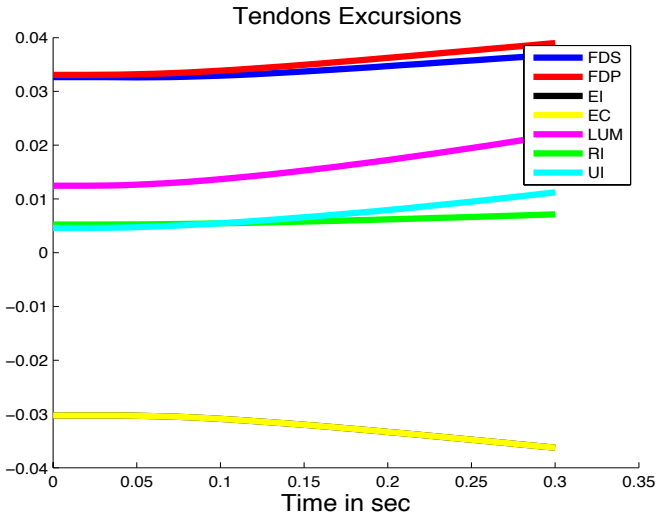


Figure 8.17: Tapping Movement: Tendon excursions for the right index finger during the flexing movement when the second model of moment arm matrix.

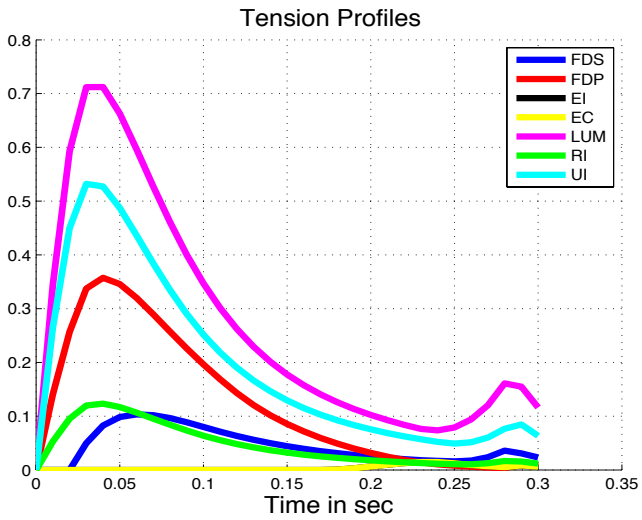


Figure 8.18: Tapping Movement: Tension profiles applied to the right index finger when the second model of moment arm matrix by is used.

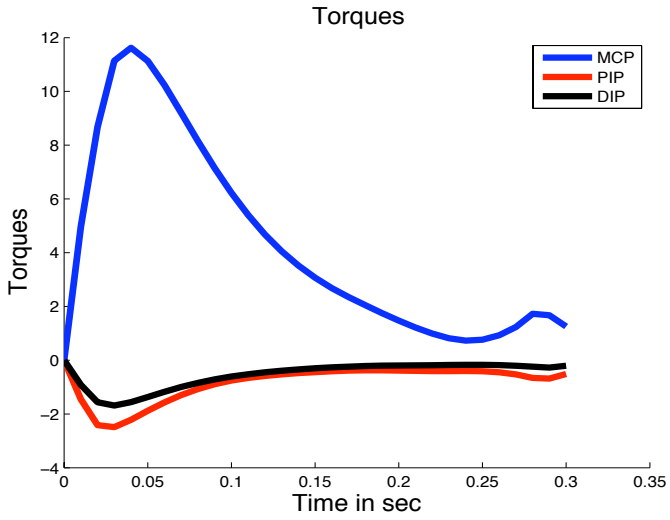


Figure 8.19: Tapping Movement: Generated torques at MCP, PIP and DIP joins of the right index finger when the second model of moment arm matrix is used.

There are few important observations regarding the kinematic behaviors and the underlying tension profiles when the two different moment arm matrices are used. More precisely:

- In figures 8.12 and 8.16 the sequence of the postures for the tapping movement for the cases of the two moment arm matrices is illustrated. In both cases the finger reaches the desired posture with some small error. It is important to mention that there is not any desired trajectory encoded in the cost function and therefore there is only a penalty at the terminal state which is the desired terminal posture. In addition, the dynamical systems are tendon driven and therefore the tensions and activation variable should be positive. Even though these hard constrains in

controls challenge the feasibility of the optimization problem, iLQG succeeds to bring the system close to the desired state.

- The tendon excursions are shown in 8.13 and 8.17. Clearly the flexor tendon FDP and FDS flex since they move inwards and the extensor tendons EC and EI extend because they move outwards. The LU, RI and UI are moving inwards and therefore they are acting as flexors for this specific tapping movement.

- In figures 8.14 and the tension profiles are shown. The difference in the use of the moment arm matrices is more apparent in the comparison of the underlying tensions. In both cases there is a synchronization of the time during which tensions are reaching their maximum value. However for the case of the first moment arm there tension at the FDS is very small. In addition for the case of the second moment arm in there is a pick of activation at 30ms before the end of the movement.

- The torque profiles are shown in 8.19 and 8.15. In both cases the highest positive torque is applied at the MCP join and the smallest negative torque at the DIP and PIP joins. Furthermore for the case of the second moment arm there is a pick of the MCP torque just 30ms before the end of the movement.

8.5 Discussion

The results above suggest that the application of the optimal control to the index finger provides different results for the cases of the two moment arm matrices. This is not a surprising result since as we have mentioned, optimal control is a constrained optimization problem with the characteristic that its constrains correspond to dynamical systems.

When different moment arm matrices are used, the underlying dynamics differ and thus the constrains of the constrained optimization problem change. This changes result in different local optimal solutions. An interesting component is that the observed differences between the two cases are more highlighted in the underlying tension profiles.

The underlying optimization in the iterative optimal control method used in this chapter, was formulated without the existence of a desired trajectory. Only a terminal desired state was used as the goal state in both movements. The outcome of the application of the optimal control is a desired optimal state trajectory $\mathbf{x}_1^*, ..., \mathbf{x}_T^*$, a feedforward optimal command $\mathbf{u}_1^*, ..., \mathbf{u}_{T-1}^*$ and locally optimal gains $\mathbf{L}_1, ..., \mathbf{L}_{T-1}$. Thus even though no desired trajectory was initially used for the design of the cost function, the resulting policy is a feedback policy which has as a desired trajectory the one that is provided by the optimization and it is the optimal $\mathbf{x}_1^*, ..., \mathbf{x}_T^*$. Consequently, for the case of nonlinear systems even though no initial trajectory is used as a desired one, the resulting controller is a tracking controller.

In this chapter, with the application of the optimal control framework on the two bio-mechanical models of the index finger, we have observed the sensitivity of the predictions with respect to model changes. These sensitivity suggest the need for verification and model checking of the bio-mechanical models under consideration.

Chapter 9

Conclusions and future work

In this thesis a new method for learning control in high dimensional state spaces has been proposed based on the framework of path integral control. On the bio-mechanical side, models of the index finger were tested for two tasks and the results where compared. In the next section we give the outline of this thesis based on the aforementioned projects and we discuss future research and extensions of current work.

9.1 Path integral control and applications to learning and control in robotics

One of the main contributions of this thesis is the derivation of path integral control for the class of nonlinear dynamical system affine in control and noise. Furthermore, this thesis suggests the iterative version of the path integral stochastic optimal control framework. The outcome of this version is a new formalizm the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals ($\mathbf{PI}^2$) capable of scaling in high dimensional learning control problems. The advantages and characteristics of $\mathbf{PI}^2$ could summarized as follows:

- With respect to other gradient based methods, in $\mathbf{PI}^2$ and in path integral control the gradient is calculated based on the weighted averaged of the local controls or local changes in the policy. These weights are given by the exponentiation of the variable $-S(\mathbf{x})$ where $S(\mathbf{x})$ is proportional of the cost of the each path. Thus, paths with high cost will have very low probability and therefore low weight while paths with low cost will have high probability. Consequently, the gradient or optimal change in policy is given by the convex combination of the local control or local changes in the policy. This calculation has obvious robustness against exploration noise.

- Since the gradient is calculated based on the convex combination of local policies, the optimality is with respect to these sampled local policies. Therefore, the question is how $\mathbf{PI}^2$ explores the state space. Exploration comes as an outcome of the iterative version of path integral stochastic optimal control. Essentially with the iterative version and the update of the parameterized policy, the local policies at the every iteration yield trajectories with lower cost than the local policies at the previous iteration.

- An essential characteristic of path integral control is that the solution of the backward Chapman Kolmogorov equation is found with forward sampling of the corresponding SDE. This characteristic comes from the direct application of the Feynman Kac lemma. Moreover, it allows us to perform sampling by executing trials on the real physical system with forward propagation of its dynamics and accumulation of the observed cost.

- Finally, in the path integral control framework, the optimal control is transformed from a minimization to a maximization problem. The exponentiation of the value function results in a new value function $\Psi(\mathbf{x})$ which has a probabilistic meaning. This probabilistic nature appears again in the final form of the optimal control as the expectation over the local controller evaluated under the probability metric of $p = \frac{e^{-S(\mathbf{x})}}{\int e^{-S(\mathbf{x})} dx}$.

9.2 Future work on path integral optimal control

The extensions of path integral control are related to different noise distributions as well as to more general classes of stochastic systems. Examples are the cases where the stochastic dynamics are not affine in controls but they are affine only in the noise term. In addition stochastic dynamics with Wiener and Poisson noise terms are also of interest. In the next three subsections we discuss these extensions of path integral control.

9.2.1 Path integral control for systems with control multiplicative noise

So far path integral stochastic optimal control has been applied to stochastic dynamical systems with state multiplicative noise. If one considers control multiplicative noise then the underlying HJB equation can also be derived. In this case however the resulting HJB equation can not be transformed to a linear PDE and therefore the application of the Feynman Kac lemma may not be possible. To avoid this obstacle one could formulate the stochastic optimal control problem as follows:

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{x}) = \min_{\mathbf{u}} E\left[\exp\left(-\int_{t_0}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x}, \mathbf{u})\right)\right] dt$$

subject to the stochastic dynamics with state and control multiplicative noise:

$$\mathbf{dx} = \mathbf{F}(\mathbf{x}, \mathbf{u})dt + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{d}\omega$$

The transition probability for the derivation of the path integral is now given as:

$$\left\langle \delta[\mathbf{x}_i - \boldsymbol{\phi}(t_i; \mathbf{x}_{i-1}, t_{i-1}]\right\rangle = \int \frac{d\boldsymbol{\omega}}{(2\pi)^n} \exp\left(j\boldsymbol{\omega}^T \boldsymbol{\mathcal{A}}\right) \exp\left(-\frac{1}{2}\boldsymbol{\omega}^T \boldsymbol{\mathcal{B}}\boldsymbol{\omega} dt\right)$$

where $\boldsymbol{\mathcal{A}} = \mathbf{x}(t_i) - \mathbf{x}(t_{i-1}) - \mathbf{F}(\mathbf{x}, \mathbf{u})dt$ and $\boldsymbol{\mathcal{B}} = \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{B}(\mathbf{x}, \mathbf{u})^T$. With respect to path integral control, there is no need for the derivation of the HJB equation. Thus, one could derive the path integral for the stochastic dynamics and then find the gradient of the cost function.

9.2.2 Path integral control for markov jump diffusions processes.

Markov jump diffusion processes are important in applications of stochastic optimal control in financial engineering, economics as well in systems biology. Many phenomena in these fields could be modeled as jump diffusion processes due to sudden changes or jumps observed, in markets and the dynamic behavior of micro-organisms such as cells. In addition, in robotics, Markov jump diffusions could model contact phenomena of walking robots with the ground. Thus, extending the path integral control framework to Markov

jump diffusion processes is of our interest. A Markov jump diffusion is expressed by the equation:

$$\mathbf{dx} = \mathbf{F}(\mathbf{x}, \mathbf{u})dt + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{d}\omega + \mathbf{h}(\mathbf{x}, t)d\mathbf{P}(t)$$

where $\mathbf{F}(\mathbf{x}, \mathbf{u}) \in \Re^{n\times 1}$ is the drift term, $\mathbf{B}(\mathbf{x}, \mathbf{u}) \in \Re^{n\times m}$ is the diffusion term and $\mathbf{h}(\mathbf{x}, t) \in \Re^{n\times l}$ is the poisson process coefficient. The HJB equation for the case of markov diffusions processes is a PDE equation with an additional integral term that corresponds to the poisson distributed stochastic term $d\mathbf{P}$. It is an open question weather or not the path integral control framework could be derived for the cases of Markov jump diffusion processes and certainly it is a topic of current and future research.

9.2.3 Path integral control for generalized cost functions

In this work, the cost functions under optimization have no cross terms between control and state dependent terms. However, one may consider a more general case of cost function in which besides the state dependent and control dependent term, there is an additional term that is the projection of controls on the space of the state. These cost functions have the form:

$$\boldsymbol{\mathcal{L}}_t = \boldsymbol{\mathcal{L}}(\mathbf{x}_t, \mathbf{u}_t, t) = q_0\left(\mathbf{x}, t\right) + \mathbf{q}_1(\mathbf{x}, t)^T\mathbf{u} + \frac{1}{2}\mathbf{u}_t^T\mathbf{R}\mathbf{u}_t$$

For these cost functions one can show that the optimal controls are expressed as follows:

$$\mathbf{u}(\mathbf{x},t) = -\mathbf{R}^{-1}\left(\mathbf{q}_1(\mathbf{x},t) + \mathbf{G}(\mathbf{x})^T\nabla_{\mathbf{x}}V(\mathbf{x},t)\right)$$

The linear HJB for this case is expressed as:

$$-\partial_t\Psi_t = -\frac{1}{\lambda}\tilde{q}_0\Psi_t + \tilde{\mathbf{f}}_t^T(\nabla_{\mathbf{x}}\Psi_t) + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t)\mathbf{\Sigma}_t\right)$$

where the terms $\tilde{q}_0$ and $\tilde{\mathbf{f}}$ are given as follows:

$$\tilde{q}_0(\mathbf{x},t) = q_0(\mathbf{x},t) - \frac{1}{2}\mathbf{q}_1(\mathbf{x},t)^T\mathbf{R}^{-1}\mathbf{q}_1(\mathbf{x},t), \quad \tilde{\mathbf{f}}(\mathbf{x},t) = \mathbf{f}(\mathbf{x},t) - \mathbf{G}(\mathbf{x},t)\mathbf{R}^{-1}\mathbf{q}_1(\mathbf{x},t)$$

Under the logarithmic transformation the optimal controls are defined by the equation:

$$\mathbf{u}(\mathbf{x},t) = -\mathbf{R}^{-1}\left(\mathbf{q}_1(\mathbf{x},t) - \lambda\mathbf{G}(\mathbf{x})^T\frac{\nabla_{\mathbf{x}}\Psi(\mathbf{x},t)}{\Psi(\mathbf{x},t)}\right)$$

It is a topic of future research to investigate the differences in the resulting optimal policies when this type of cost functions is used.

9.3 Future work on stochastic dynamic programming

Another contribution of this thesis is the derivation of the stochastic version of Differential Dynamic Programming (SDDP) for the cases of stochastic dynamical systems with state and control multiplicative noise. There are many possible extensions and research topics for SDDP which are summarized as follows:

- Further applications of SDDP to stochastic dynamical systems and extension for the case of constraints in state and controls.

- So far we have derived the SDDP by using the Itô calculus. It is of our interest to investigate how different discretization schemes based on Stratonovich or other stochastic calculus could affect the convergence of SDDP. This is important for systems where the noise is control and state multiplicative.

- Extensions of SDDP to the cases of partial observability and the addition of a extended second order truncated Kalman filter. The resulting algorithm can be thought as a version of nonlinear LQG design in which state space dynamics are expended up to the second order for both estimation and control.

9.4 Future work on neuromuscular control

The application of optimal control methods to identify the underlying tension profiles for the index finger reveals that the results depend on the model. We have used two different moment arm models that distribute the forces applied on the index finger in a different way. The question of, which moment arm model is the most appropriate one, is open and difficult to answer since it requires a experiments in which access to tendon tensions is possible.

A topic for future research is to develop methods which could be used to verify biomechanical models before optimal control is applied. A possible way to verify models of the index finger biomechanics would be to record trajectories of finger movements in

humans, and then test weather or not the candidate models satisfy the local controllability condition when they are linearized on the recorded trajectories. But even if the controllability condition is satisfied that does not mean that the tested model is a good candidate due to the fact that controls are constrained. More precisely if neural activity is treated as the control variable then it is bounded between 0 and 1 while in cases where the forces produced by the muscles is treated as controls then these control variables have to be positive. Thus, the controllability condition is a necessary but not a sufficient condition for the case of constrained controls.

Future research will investigate the application of alternative methods of optimal control such as the Pseudospectral methods. In Pseudospectral methods, the optimal trajectory and control are represented as polynomial functions of time. These methods can handle hard constrains in control and state however they provide open loop optimal policies and not feedback policies. Moreover, they are mostly applicable to deterministic and not stochastic systems. It is really an open question how Pseudospectral methods compare to iterative methods and how they could be applied to bio-mechanical models.

Bibliography

Abend, W., Bizzi, E. & Morasso, P. (1982), 'Human arm trajectory formation', *Brain* **105**(Pt 2), 331–348.

Amari, S. (1999), 'Natural gradient learning for over- and under-complete bases in ica', *Neural Computation* **11**(8), 1875–83.

An, K. N., Chiao, E. Y., Cooney, W. P. & Linscheid, R. L. (1985), 'Forces in the normal and abnormal hand', *Journal of Orthopaedic Research,* **3**, 202 – 211.

An, K., Ueba, Y., Chao, E., Cooney, W. & Linscheid, R. (1983), 'Tendon excursion and moment arm of index finger muscles', *Journal of Biomechanics* **16**(6), 419 – 425.

Basar, T. (1991), *Time Consistency and robustness of equilibria in noncooperative dynamic games*, Springer Verlag, North Holland.

Basar, T. & Berhard, P. (1995), *H-infinity Optimal Control and Related Minimax Design*, Birkhauser, Boston.

Baxter, J. & Bartlett, P. L. (2001), 'Infinite-horizon policy-gradient estimation', *Journal of Artificial Intelligence Research* **15**, 319–350.

Bellman, R. & Kalaba, R. (1964), *Selected Papers On mathematical trends in Control Theory*, Dover Publications.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blankevoort, L., Kuiper, J., Huiskes, R. & Grootenboer, H. (n.d.), 'Articular contact in a three-dimensional model of the knee', *Journal of Biomechanics* .

Blemker, S. S. & Delp, S. L. (2005.), 'Three-dimensional representation of complex muscle architectures and geometries,', *Annals of Biomedical Engineering,* **33**(5), 661 – 773.

Broek, B. V. D., Wiegerinck, W. & Kappen., H. J. (2008), 'Graphical model inference in optimal control of stochastic multi-agent systems', *Journal of Artificial Intelligence Research* **32**(1), 95–122.

Buchli, J., Kalakrishnan, M., Mistry, M., Pastor, P. & Schaal, S. (2009), compliant quadruped locomotion over rough terrain, *in* 'intelligent robots and systems, 2009. iros 2009. ieee/rsj international conference on'.
URL: *http://www-clmc.usc.edu/publications/B/buchli-IROS2009.pdf*

Buchli, J., Theodorou, E., Stulp, F. & Schaal, S. (2010), Variable impedance control - a reinforcement learning approach, *in* 'Robotics: Science and Systems Conference (RSS)'.

Cerveri, P., De Momi, E., Marchente, M., Lopomo, N., Baud-Bovy, G., Barros, R. M. L. & Ferrigno, G. (2008), 'In vivo validation of a realistic kinematic model for the trapezio-metacarpal joint using an optoelectronic system', *ANNALS OF BIOMEDICAL ENGINEERING* **36**(7), 1268–1280.

Cheng, G., Hyon, S., Morimoto, J., Ude, A., Hale, J., Colvin, G., Scroggin, W. & Jacobsen, S. C. (2007), 'Cb: A humanoid research platform for exploring neuroscience', *Journal of Advanced Robotics* **21**(10), 1097–1114.

Chirikjian, S. G. (2009), *Stochastic Models, Information Theory, and Lie Groups.*, Vol. I, Birkhăuser.

Dayan, P. & Hinton, G. (1997), 'Using em for reinforcement learning', *Neural Computation* **9**.

Deisenroth, M. P., Rasmussen, C. E. & Peters, J. (2009), 'Gaussian process dynamic programming', *Neurocomputing* **72**(7–9), 1508–1524.

Delp, S. L. & Loan, J. P. (2007), 'A graphics-based software system to develop and analyze models of musculoskeletal structures,', *Computers in Biology and Medicine* **25**(1), 21 – 34.

Dennerlein, J. T., Diao, E., Mote, C. D. & Rempel, D. M. (1998), 'Tensions of the flexor digitorum superficialis are higher than a current model predicts', *Journal of Biomechanics* **31**(4), 295 – 301.

Dorato, P., Cerone, V. & Abdallah, C. (2000), *Linear Quadratic Control: An Introduction*, Krieger Publishing Co., Inc., Melbourne, FL, USA.

Doyle, J. (1978), 'Guaranteed margins for lqg regulators', *Automatic Control, IEEE Transactions on* **23**(4), 756 – 757.

Esteki, A. & Mansour, J. M. (1996), 'An experimentally based nonlinear viscoelastic model of joint passive moment', *Journal of Biomechanics* **29**(4), 443 – 450.

Feynman, P. R. & Hibbs, A. (2005), *Quantum Mechanics and Path Integrals*, Dover - (Emended Edition).

Fleming, W. H. & Soner, H. M. (2006), *Controlled Markov Processes and Viscosity Solutions*, Applications of aathematics, 2nd edn, Springer, New York.

Freivalds, A. (2000), *Biomechanics of the upper limbs: mechanics, modeling, and Musculoskeletal injures*, 1rd edn, CRC Press.

Friedman, A. (1975), *Stochastic Differential Equations And Applications*, Academic Press.

Gardiner, C. (2004), *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, Spinger.

Garner, B. A. & Pandy, M. G. (2000), 'The obstacle-set method for representing muscle paths in musculoskeletal models,', *Computer methods in biomechanics and biomedical engineering* **3**(1), 1 – 30.

Ghavamzadeh, M. & Yaakov, E. (2007), Bayesian actor-critic algorithms, *in* 'ICML '07: Proceedings of The 24th International Conference on Machine Learning', pp. 297–304.

Harding, D., Brandt, K. & Hillberry, B. (1993), 'Finger joint force minimization in pianists using optimization techniques', *Journal of Biomechanics* **26**(12), 1403 – 1412.

Harris, C. M. & Wolpert, D. M. (1998), 'Signal-dependent noise determines motor planning', *Nature* **394**, 780–784.

Hatze, H. (1997), 'A three-dimensional multivariate model of passive human joint torques and articular boundaries', *Clinical Biomechanics* **12**(2), 128 – 135.

He, J., Levine, W. & Loeb, G. (1991), 'Feedback gains for correcting small perturbations to standing posture', *Automatic Control, IEEE Transactions on* .

Hollister, A., Buford, W. L., Myers, L. M., Giurintano, D. J. & Novick, A. (1992), 'The axes of rotation of the thumb carpometacarpal joint.', *Journal of Orthopaedic Research* **10**(3), 454–460.

Ijspeert, A., Nakanishi, J., Pastor, P., Hoffmann, H. & Schaal, S. (submitted), 'learning nonlinear dynamical systems models'.
URL: *http://www-clmc.usc.edu/publications/I/ijspeert-submitted.pdf*

Ijspeert, A., Nakanishi, J. & Schaal, S. (2003), Learning attractor landscapes for learning motor primitives, *in* S. Becker, S. Thrun & K. Obermayer, eds, 'Advances in Neural Information Processing Systems 15', Cambridge, MA: MIT Press, pp. 1547–1554.

Jacobson, D. H. (1973), 'Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games', *IEEE Transactions of Automatic Control* **AC - 18**, 124–131.

Jacobson, D. H. & Mayne, D. Q. (1970), *Differential dynamic programming*, American Elsevier Pub. Co., New York,.

James, M. R., Baras, J. & Elliot, R. (1994), 'Risk sensitive control of dynamic games for partially observed discrete - time nonlinear systems', *IEEE Transactions of Automatic Control* **AC - 39(4)**, 780–792.

Jetchev, N. & Toussaint, M. (2009), Trajectory prediction: learning to map situations to robot trajectories, *in* 'ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning', pp. 449–456.

Jinha, A., Ait-Haddou, R., Binding, P. & Herzog, W. (2006), 'Antagonistic activity of one-joint muscles in three-dimensions using non-linear optimisation', *Mathematical Biosciences* **202**(1), 57 – 70.

Kalman, R. (1964), 'When is a linear control system optimal?', *ASME Transactions, Journal of Basic Engineering* **86**, 51–60.

Kappen, H. J. (2005*a*), 'Linear theory for control of nonlinear stochastic systems', *Phys. Rev. Lett.* **95**, 200201.

Kappen, H. J. (2005*b*), 'Path integrals and symmetry breaking for optimal control theory', *Journal of Statistical Mechanics: Theory and Experiment* (11), P11011.

Kappen, H. J. (2007), An introduction to stochastic control theory, path integrals and reinforcement learning, *in* J. Marro, P. L. Garrido & J. J. Torres, eds, 'Cooperative Behavior in Neural Systems', Vol. 887 of *American Institute of Physics Conference Series*, pp. 149–181.

Karatzas, I. & Shreve, S. E. (1991), *Brownian Motion and Stochastic Calculus (Graduate Texts in Mathematics)*, 2nd edn, Springer.

Kober, J. & Peters, J. (2009), Learning motor primitives in robotics, *in* D. Schuurmans, J. Benigio & D. Koller, eds, 'Advances in Neural Information Processing Systems 21', Cambridge, MA: MIT Press, Vancouver, BC, Dec. 8-11.

Lau, A. W. C. & Lubensky, T. C. (2007), 'State-dependent diffusion: thermodynamic consistency and its path integral formulation'.
URL: *http://arxiv.org/abs/0707.2234*

Leitmann, G. (1981), *The Calculus Of Variations and Optimal Control*, Plenum Press, New York.

Li, W. & Todorov, E. (2004), Iterative linear quadratic regulator design for nonlinear biological movement systems, *in* 'ICINCO (1)', pp. 222–229.

Li, W. & Todorov, E. (2006), An iterative optimal control and estimation design for nonlinear stochastic system, *in* 'Decision and Control, 2006 45th IEEE Conference on', pp. 3242 –3247.

Morimoto, J. & Atkeson, C. (2002), Minimax differential dynamic programming: An application to robust biped walking, *in* 'In Advances in Neural Information Processing Systems 15', MIT Press, Cambridge, MA.

Morimoto, J. & Doya, K. (2005), 'Robust reinforcement learning', *Neural Comput.* **17**(2).

Murray, R. M., Li, Z. & Sastry, S. S. (1994), *A Mathematical Introduction to Robotic Manipulation*, 1 edn, CRC.

Mussa-Ivaldi, A., Hogan, N. & Bizzi, E. (1982), 'Neural, mechanical, and geometric factors subserving arm posture in humans', *Journal of Neuroscience* **5**, 331–348.

Nobel-Lectures (1965), *Physics 1922-1941*, Elsevier Publishing Company, Amsterdam.

Nobel-Lectures (1972), *Physics 1963-1970*, Elsevier Publishing Company, Amsterdam.

Øksendal, B. K. (2003), *Stochastic Differential Equations : An Introduction with Applications*, 6th edn, Springer, Berlin; New York.

Olney, S. J., Griffin, M. P., Monga, T. N. & McBride, I. D. (1991), 'Work and power in gait of stroke patients', *Archives of physical medicine and rehabilitation,* **72**(5), 309 – 314.

Pastor, P., Kalakrishnan, M., Chitta, S., Theodorou, E. & Schaal, S. (2011), skill learning and task outcome prediction for manipulation, *in* '2011 IEEE international conference on Robotics and Automation'.

Peters, J. (2007), Machine Learning of Motor Skills for Robotics., PhD thesis, University of Southern California.

Peters, J. & Schaal, S. (2008*a*), 'Learning to control in operational space', *International Journal of Robotics Research* **27**, 197–212.

Peters, J. & Schaal, S. (2008*b*), 'Natural actor critic', *Neurocomputing* **71**(7-9), 1180–1190.

Peters, J. & Schaal, S. (2008*c*), 'Reinforcement learning of motor skills with policy gradients', *Neural Networks* **21**(4), 682–97.

Piazza, S. J. & Delp, S. L. (2001), 'Three-dimensional dynamic simulation of total knee replacement motion during a step-up task', *Journal of Biomechanical Engineering* **123**(6), 599–606.

Pontryagin, L., Boltyanskii, V., Gamkrelidze, R. & Mishchenko, E. (1962), *The mathematical theory of Optimal Processes*, Pergamon Press, New York.

R. Davoodi, I. E. B. & Loeb, G. E. (2003), 'Advanced modeling environment for developing and testing fes control systems,', *Medical Engineering and Physics* **25**(1), 3 – 9.

Rawlinson, J. J. & Bartel, D. L. (2002), 'Flat medial-lateral conformity in total knee replacements does not minimize contact stresses', *Journal of Biomechanics* **35**(1), 27 – 34.

Rawlinson, J. J., Furman, B. D., Li, S., Wright, T. M. & Bartel, D. L. (2006), 'Retrieval, experimental, and computational assessment of the performance of total knee replacements', *Journal of Orthopaedic Research* **24**(7), 1384 – 1394.

Ross, D. (2009), *Aristotle: The Nicomachean Ethics*, Oxford University Press.

Runolfsson, T. (1994), 'The equivalence between infinite horizon control of stochastic systems with exponential of integral performance index and stochastic differential games', *IEEE Transactions of Automatic Control* **39**, 1551–1563.

Russell, S. & Norvig, P. (2003), *Artificial Intelligence: A Modern Approach*, second edn, Prentice Hall.

S. S. Blemker, D. S. Asakawa, G. E. G. & Delp, S. L. (2007), 'Image- based musculoskeletal modeling: applications, advances, and future opportunities,', *Journal of Magnetic Resonance Imaging,* **25**(2), 441 – 451.

Safonov, M. G. & Athans, M. (1976), Gain and phase margin for multiloop lqg regulators, *in* 'Decision and Control including the 15th Symposium on Adaptive Processes, 1976 IEEE Conference on', Vol. 15, pp. 361 –368.

Sancho-Bru, J. L., Prez-Gonzlez, A., Vergara-Monedero, M. & Giurintano, D. (2001), 'A 3-d dynamic model of human finger for studying free movements', *Journal of Biomechanics* **34**(11), 1491 – 1500.

Santos, V. & Valero-Cuevas, F. (2006), 'Reported anatomical variability naturally leads to multimodal distributions of denavit-hartenberg parameters for the human thumb', *Biomedical Engineering, IEEE Transactions on* **53**(2), 155 –163.

Saridis, G. (1996), *Stochastic Processed, Estimation and Control. The Entropy approach*, John Wiley and Sons, New York.

Schaal, S. (2009), the sl simulation and real-time control software package, Technical report.
URL: *http://www-clmc.usc.edu/publications/S/schaal-TRSL.pdf*

Schulz, M. (2006), *Control Theory in Physics and other Fields of Science. Concepts, Tools and Applications*, Spinger.

Sciavicco, L. & Siciliano, B. (2000), *Modelling and Control of Robot Manipulators*, Advanced textbooks in control and signal processing, Springer, London ; New York.

Stengel, R. F. (1994), *Optimal Control and Estimation*, Dover books on advanced mathematics, Dover Publications, New York.

Sutton, R. S. & Barto, A. G. (1998), *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*, The MIT Press.

Sutton, R. S., McAllester, D., Singh, S. & Mansour, Y. (2000), Policy gradient methods for reinforcement learning with function approximation, *in* 'Advances in Neural Information Processing Systems 12', MIT Press, pp. 1057–1063.

Theodorou, E., Buchli, J. & Schaal, S. (2010), 'A generalized path integral control approach to reinforcement learning.', *Journal of Machine Learning Research* p. 3137?3181.

Theodorou, E., T.-Y. T. E. (2010), 'stochastic differential dynamic programming'.

Todorov, E. (2004), 'Optimality principles in sensorimotor control.', *Nature neuroscience* **7**(9), 907–915.

Todorov, E. (2005), 'Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system', *Neural Computation* **17**(5), 1084.

Todorov, E. (2007), Linearly-solvable markov decision problems, *in* B. Scholkopf, J. Platt & T. Hoffman, eds, 'Advances in Neural Information Processing Systems 19 (NIPS 2007)', Cambridge, MA: MIT Press, Vancouver, BC.

Todorov, E. (2008), General duality between optimal control and estimation, *in* 'Decision and Control, 2008. CDC 2008. 47th IEEE Conference on', pp. 4286 –4292.

Todorov, E. & Jordan, M. I. (2002), 'Optimal feedback control as a theory of motor coordination.', *Nature neuroscience* **5**(11), 1226–1235.
URL: *http://dx.doi.org/10.1038/nn963*

Toussaint, M. & Storkey, A. (2006), 'Probabilistic inference for solving discrete and continuous state markov decision processes'.

Valero-Cuevas, F. J. (2009), 'A mathematical approach to the mechanical capabilities of limbs and fingers', **629**, 619–633.

Valero-Cuevas, F. J., Johanson, M. E. & Towles, J. D. (2003), 'Towards a realistic biomechanical model of the thumb: the choice of kinematic description may be more critical than the solution method or the variability/uncertainty of musculoskeletal parameters.', *J Biomech* **36**(7), 1019–1030.

Valero-Cuevas, F. J., Towles, J. D. & Hentz, V. R. (2000), 'Quantification of fingertip force reduction in the forefinger following simulated paralysis of extensor and intrinsic muscles', *Journal of Biomechanics* **33**(12), 1601 – 1609.

Valero-Cuevas, F. J., Zajac, F. E. & Burgar, C. G. (1998), 'Large index-fingertip forces are produced by subject-independent patterns of muscle excitation', *Journal of Biomechanics* **31**(8), 693 – 703.

Venkadesan, M. & Valero-Cuevas, F. (2008*a*), 'Effects of time delays on controlling contact transitions', *Royal Society* .

Venkadesan, M. & Valero-Cuevas, F. (2008*b*), 'Neural control of motion- to force transitions with the fingertip', *The journal of Neuroscience* **28**(6), 1366–1373.

Vlassis, N., Toussaint, M., Kontes, G. & S., P. (2009), 'Learning model-free control by a monte-carlo em algorithm', *Autonomous Robots* **27**(2), 123–130.

Whittle, P. (1990), *Risk Sensitive Optimal Control*, Wiley.

Whittle, P. (1991), 'Risk sensitive optimal linear quadratic gaussian control', *Adv. Appl. Probability* **13**, 746 – 777.

Williams, R. J. (1992), 'Simple statistical gradient-following algorithms for connectionist reinforcement learning', *Machine Learning* **8**, 229–256.

Wismans, J., Veldpaus, F., Janssen, J., Huson, A. & Struben, P. (1980), 'A three-dimensional mathematical model of the knee-joint', *Journal of Biomechanics* **13**(8), 677 – 679, 681–685.

Yoon, Y. & Mansour, J. (1982), 'The passive elastic moment at the hip', *Journal of Biomechanics* **15**(12), 905 – 910.

Zajac, F. E. (1989), 'Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control.', *Crit. Rev. Biomed. Eng.* **17**(4), 350 – 411.

Zajac, F. E. (1992), 'How musculotendon architecture and joint geometry affect the capacity of muscles to move and exert force on objects: a review with application to arm and forearm tendon transfer design.', *J. Hand. Surg. Am.* **17**(5), 799 – 804.

Zefran, M., Kumar, V. & Croke, C. (1998), 'On the generation of smooth three-dimensional rigid body motions', *IEEE Transactions on Robotics and Automation* **14**(4), 576–589.