Stochastic Optimal Control with Learned Dynamics Models

Djordje Mitrovic

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
2010

Abstract

The motor control of anthropomorphic robotic systems is a challenging computational task mainly because of the high levels of redundancies such systems exhibit. Optimality principles provide a general strategy to resolve such redundancies in a task driven fashion. In particular closed loop optimisation, i.e., *optimal feedback control (OFC)*, has served as a successful motor control model as it unifies important concepts such as costs, noise, sensory feedback and internal models into a coherent mathematical framework.

Realising OFC on realistic anthropomorphic systems however is non-trivial: Firstly, such systems have typically large dimensionality and nonlinear dynamics, in which case the optimisation problem becomes computationally intractable. Approximative methods, like the *iterative linear quadratic gaussian (ILQG)*, have been proposed to avoid this, however the transfer of solutions from idealised simulations to real hardware systems has proved to be challenging. Secondly, OFC relies on an accurate description of the system dynamics, which for many realistic control systems may be unknown, difficult to estimate, or subject to frequent systematic changes. Thirdly, many (especially biologically inspired) systems suffer from significant state or control dependent sources of noise, which are difficult to model in a generally valid fashion. This thesis addresses these issues with the aim to realise efficient OFC for anthropomorphic manipulators.

First we investigate the implementation of OFC laws on anthropomorphic hardware. Using ILQG we optimally control a high-dimensional anthropomorphic manipulator without having to specify an explicit inverse kinematics, inverse dynamics or feedback control law. We achieve this by introducing a novel cost function that accounts for the physical constraints of the robot and a dynamics formulation that resolves discontinuities in the dynamics. The experimental hardware results reveal the benefits of OFC over traditional (open loop) optimal controllers in terms of energy efficiency and compliance, properties that are crucial for the control of modern anthropomorphic manipulators.

We then propose a new framework of *OFC with learned dynamics (OFC-LD)* that, unlike classic approaches, does not rely on analytic dynamics functions but rather updates the internal dynamics model continuously from sensorimotor plant feedback. We demonstrate how this approach can compensate for unknown dynamics and for complex dynamic perturbations in an online fashion.

A specific advantage of a learned dynamics model is that it contains the stochastic information (i.e., noise) from the plant data, which corresponds to the uncertainty in the system. Consequently one can exploit this information within OFC-LD in order to produce control laws that minimise the uncertainty in the system. In the domain of antagonistically actuated systems this approach leads to improved motor performance, which is achieved by co-contracting antagonistic actuators in order to reduce the negative effects of the noise. Most importantly the shape and source of the noise is unknown a priory and is solely learned from plant data. The model is successfully tested on an antagonistic *series elastic actuator (SEA)* that we have built for this purpose.

The proposed OFC-LD model is not only applicable to robotic systems but also proves to be very useful in the modelling of biological motor control phenomena and we show how our model can be used to predict a wide range of human impedance control patterns during both, stationary and adaptation tasks.

Acknowledgements

There are a number of people whom I wish to thank for their support, help and advice during my PhD.

First, I would like to thank my supervisor, *Sethu Vijayakumar*, for his great inspiration and support both professionally and personally and for giving me so many opportunities and ideas to conduct truly fascinating research.

Second, I would like to thank *Stefan Klanke* for his close help on all levels of my research. He took always the time to discuss interesting research directions with me and especially his technical help and editorial advice was much valued for me.

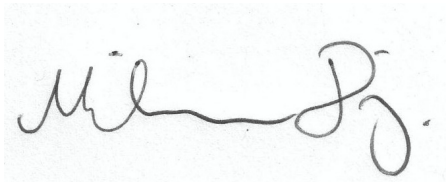
Third, my thanks go to my numerous collaborators, *Rieko Osu* and *Mitsuo Kawato* from ATR labs in Kyoto, *Takamitsu Matsubara* and *Sho Nagashima* from NAIST, and *Patrick van der Smagt* from DLR.

Furthermore, special thanks go to *Jun Nakanishi* for giving me useful advice during the thesis write-up and for proof-reading my thesis.

Last but not least, I would like to express my gratitude to my dear girlfriend *Katja Ammon*, who enriched my life for over a decade and who kept me going in stressful times.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.



(*Djordje Mitrovic*)

Table of Contents

List of Notation

Below is a list of symbols and abbreviations used throughout this thesis (unless noted differently in the text). We use the convention of bold upper-case, **A**, for matrices, bold lower-case letters, **a**, for vectors and normal weighted font, *a*, for scalars. Entries of the form $f(\cdot)$ denote that an argument should be supplied to the function f.

Symbol	
$\mathbf{x}$	State space variable.
$\mathbf{u}$	Control variable.
$\pi(\cdot)$	Policy mapping from states to actions.
$J(\cdot)$	Cost function or performance index.
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Position, velocity and acceleration in joint space.
$\boldsymbol{\tau}$	Torque in joint space.
t	Time (continuous).
a_k	Value of variable a at discrete time step k .
T	Duration in time (e.g., of a trajectory).
$\mathbf{I}$	Identity matrix.
$N(\mu, \sigma)$	Gaussian distribution with mean μ and standard deviation σ.
$\langle f(x) \rangle$	Expectation value of function $f(x)$ in variable x.
$\nabla_{\mathbf{x}}$	Gradient operator with respect to variable $\mathbf{x}$.
$\mathbf{a} \cdot \mathbf{b}$	Dot product of the vectors $\mathbf{a}$ and $\mathbf{b}$.
$\mathbf{a} \times \mathbf{b}$	Cross product of the vectors $\mathbf{a}$ and $\mathbf{b}$.
$[x]_+$	Compact notation for $\max(0, x)$.

Abbreviations

CV	Calculus of variations.
CNS	Central nervous system.
DDP	Differential dynamic programming.
DoF	Degrees of freedom.
FIR	Finite impulse response.
FF	Force field.
ILC	Iterative learning controller.
ILQR	Iterative linear quadratic regulator.
ILQG	Iterative linear quadratic Gaussian.
ILQG–LD	Iterative linear quadratic Gaussian with learned dynamics.
KV	Kinematic variability.
LQR	Linear quadratic regulator.
LQG	Linear quadratic Gaussian.
LWL	Locally weighted learning.
LWPR	Locally weighted projection regression.
MPC	Model predictive control.
nMSE	Normalised mean squared error.
OC	Optimal control.
ODE	Ordinary differential equation.
OFC	Optimal feedback control.
OFC-LD	Optimal feedback control with learned dynamics.
P(I)D	Proportional (integral) derivative.
PLS	Partial least squares.
RL	Reinforcement learning.
SEA	Series elastic actuator.
SOC	Stochastic optimal control.

Chapter 1

Introduction and motivation

Humans and other biological systems are very adept at performing fast and complicated control tasks while being fairly robust to noise and perturbations. This unique combination of accuracy, robustness and adaptability is extremely appealing in any autonomous robotic system. The human motion apparatus is by nature a highly redundant system and modern anthropomorphic robots, designed to mimic human behaviour and performance, typically exhibit large *degrees of freedom (DoF)* in the kinematics domain (i.e., joints) and in the dynamics domain (i.e., actuation). Examples of such robots are depicted in Fig. 1.1. However often the additional flexibility comes with the price of an increased control costs. For example, if we want to perform a presumably simple reaching task with a redundant robotic arm, i.e, from a start configuration to a target position (x,y,z) in Euclidean task space, multiple levels of redundancies need to be resolved: typically there will be multiple possible trajectories in *task space* leading to the same target. Each of these task space routes again can be achieved with a multitude of configurations in *joint angle space*. On a dynamics level, in the case of redundant actuation, each joint angle trajectory can be realised with different levels of *muscle co-contractions*. Therefore for redundant systems producing even the simplest movement involves an enormous amount of information processing and a controller has to make a choice from a very large space of possible controls. Therefore an important question to answer is how to resolve this redundancy and how to make a particular control choice?

Optimal control theory (Stengel, 1994) answers this question by postulating that a particular choice is made because it is the optimal solution to a specific task. The objective in an optimal control problem is it to minimise the value of a cost function which represents the performance criteria that a motion system should adhere. Exam-

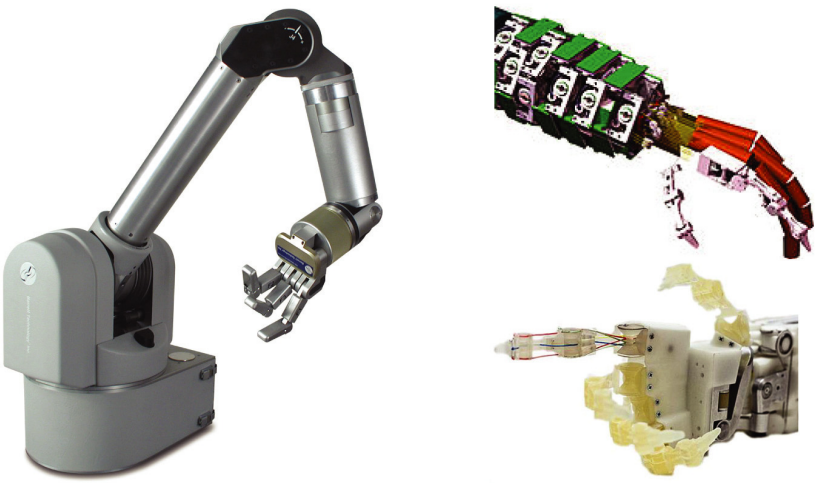


Figure 1.1: Examples of modern anthropomorphic robots mimicking human morphology in the kinematics and the dynamics. These systems are known to exhibit large degrees of freedom. Left: The Barrett WAM is a joint torque controlled anthropomorphic manipulator with 7 kinematic DoF in the arm and 9 kinematic DoF in the hand. Right: The antagonistically actuated hand-arm system developed at the German Aerospace Center (DLR). This system has more than 40 motors and about 25 kinematic degrees of freedom.

ples for such criteria could be, energy consumption, distance to a target or duration of the movement. This approach stands in vast contrast to traditional control, which typically is divided into trajectory planning, finding an inverse kinematic solution and final tracking of a trajectory (An et al., 1988). Therefore optimal control provides a principled and mathematically coherent framework to resolve redundancies in an optimal fashion with respect to the task at hand.

Generally speaking we can distinguish two kinds of optimal control problems, *open loop* and *closed loop* problems. The solution to an open loop problem is an optimal control trajectory. The solution to a closed loop problem is an optimal control law, i.e., a functional mapping from states to optimal controls. Assuming deterministic dynamics (i.e., no unknown perturbations or noise) open-loop control will produce a sequence of optimal motor signals or limb states. However if the system leaves the optimal path it must be corrected for example with a hand tuned PID controller, which most likely will lead to suboptimal behaviour, because the feedback gains have not been incorporated into the optimisation process. Stable optimal performance can only be guaranteed by constructing an *optimal feedback law* that produces a mapping from states to actions by all sensory data available. Therefore in such a closed loop

optimisation, which is also known as *optimal feedback controller (OFC)*, there is no separation between the trajectory planning and trajectory execution for the completion of a task.

Recently OFC has received large attention in the study of biological motor control systems, that are known to suffer from large sensorimotor noise and delays (Faisal et al., 2008). Indeed from a biological point of view optimality is very well motivated as the sensorimotor system can be understood as a result of *natural optimisation*, i.e., evolution, learning, adaptation. Specifically the stochastic OFC model proposed by Todorov and Jordan (2002), which assumes that the policy is optimal with respect to the expectation over the noise of the objective function value, has been a very promising approach. Its fundamental assumption is that the *central nervous system (CNS)* is aware of the system noise and plans its actions to minimise the objective value, which typically consists of task error, end point stability and control effort. Under such a cost function and an appropriate arm model these OFC models have shown to predict the main characteristics of human motion, such as bell-shaped velocity profiles, curved trajectories, goal-directed corrections (Liu and Todorov, 2007), multi-joint synergies (Todorov and Ghahramani, 2004) and variable but successful motor performance. The OFC framework is currently viewed as the predominant theory for interpreting volitional biological motor control (Scott, 2008), since it unifies motor costs, expected rewards, internal models, noise and sensory feedback into a coherent mathematical framework (Shadmehr and Krakauer, 2008).

The aim of this thesis is to transfer biological optimal motor control strategies, more specifically OFC, to artificial limb systems (i.e., arms, legs). Doing so will not only improve performance of humanoid robotic applications but also can help to broaden our understanding of the control principles behind human motor performance. Despite the appeal of OFC as a motor control strategy for high dimensional systems, in its current form it has certain issues that we specifically address in this thesis.

Scaling of OFC to high dimensional, nonlinear hardware systems

Many optimal motor control models in robotics have focused on open loop optimisation whereas closed loop optimal control found little attention. In fact we are not aware of any OFC implementations on a large DoF system. The reasons for this are twofold: (i) It is computationally much more difficult to obtain OFC laws as opposed

to open loop solutions, especially for high dimensional and nonlinear systems[1]. It is very complicated to solve a closed loop problem since the information represented by the optimal value function is essentially equal to the information obtained by solving a two point boundary ordinary differential equation from *each* point in state space. One way to avoid computational problems in practice is to use approximative methods as discussed in detail in Chapter 2. Approximative optimal control methods such as *differential dynamic programming (DDP)* (Dyer and McReynolds, 1970; Jacobson and Mayne, 1970) and the *iterative linear quadratic Gaussian (ILQG)* (Todorov and Li, 2005) iteratively compute an optimal trajectory together with a locally valid feedback law and therefore are not directly subject to the curse of dimensionality. Previous work largely has focused on the theoretical aspects in idealised simulated scenarios (Li, 2006; Tassa et al., 2007; Mitrovic et al., 2008b) or on fairly simplistic robotic devices (Morimoto and Atkeson, 2003). (ii) To successfully implement OFC laws on a robotic system one needs to identify an accurate dynamics model of the real system incorporating real world constraints such as joint angle limits, maximal joint velocities and applicable controls. Furthermore the dynamics model requires additional modelling effort due to discontinuities that real systems suffer from and which impair numerical stability of approximative OFC methods (Chapter 3).

Adaptation paradigm within OFC

Traditionally optimal control methods rely on analytic dynamics formulations that model the behaviour of the controlled system. A characteristic property of anthropomorphic systems is their lightweight and flexible-joint construction which is a key ingredient to achieve compliant human-like motion. However such a morphology complicates analytic dynamics calculations and unforeseen changes in the plant dynamics are even harder to model. A solution to this shortcoming is to apply online supervised learning methods to extract dynamics models driven by data from the movement system itself. This enables the controller to adapt “on the fly” to changes in dynamics due to wear and tear or external perturbations. Such adaptation methods have been studied previously in robot control (Vijayakumar et al., 2002; D’Souza et al., 2001; Conradt et al., 2000) but have not found much attention in the perspective of the optimal control framework. Indeed the ability to adapt to systematic perturbations is a key feature of biological motion systems and enabling optimal control to be adaptive is a valuable

[1] If the plant dynamics is linear and the cost function is quadratic the optimisation problem is convex and can be solved analytically as in LQR and LQG.

theoretical test-bed for human adaptation experiments (Chapter 4).

Optimally exploiting stochastic information in the dynamics

The human sensorimotor system exhibits highly stochastic characteristics due to various cellular and behavioral sources of variability (Faisal et al., 2008) and a complete motor control theory must contend with the detrimental effects of *signal dependent noise (SDN)* on task performance. One specific example of such a control strategy that takes into account the variability of the motor system is *impedance control* (Hogan, 1984). By co-contracting antagonistic muscle pairs of limbs, the joint stiffness can be increased leading to a reduction of kinematic perturbations. The fact that humans perform very well under very noisy conditions, for example by modulating joint impedance, raises the question if and how this can be achieved within the OFC framework. How to do this is not obvious at all given the fact that biological OFC models usually minimise for control effort, a principle that contradicts muscle co-contraction entirely. A generic way to incorporate stochastic information, without having prior information about source or shape of it, is to use as before a learning framework that can acquire localised (i.e., state or control dependent) stochastic information of the dynamics. This offers a principled strategy of exploiting any kind of stochastic dynamics information and incorporating it into our optimisation. We will show that this leads to improved control performance in robotic systems that suffer from external sources of noise (Chapter 5). We will also demonstrate that this model can predict and conceptually explain impedance control behaviours observed in humans (Chapter 6).

Thesis outline

Next, we provide an outline of the thesis summarising the content of each chapter. Below each chapter description we highlight the original contributions made and we give references to our work that has been published during the course of research.

In **Chapter 2** we give a short introduction to the vast subject of optimal control theory. We review the relevant literature on optimal control with a specific emphasis on motor control problems for high dimensional movement systems. We then motivate the use of approximate optimal control methods and elaborate upon the recently introduced ILQG algorithm, which in the consequent chapters will be used to compute optimal control solutions.

Original contributions:

- *Review of optimal control methods relevant for the control of non-linear and high dimensional systems.*
- *Implementation of ILQG algorithm scalable to large DoF.*

In **Chapter 3** we extend the ILQG algorithm in order to be able to optimally control a real robotic manipulator with large DoF. We show the beneficial properties of this control strategy over traditional controllers in terms of energy efficiency and compliance. These properties are crucial for the control of (mobile) anthropomorphic robots, designed to interact safely in a human environment.

Original contributions:

- *Extension of ILQG in order to incorporate real world constraints and discontinuities in the dynamics.*
- *First ILQG implementation on a real high-dimensional manipulator.*
- *Thorough experimental evaluation, highlighting the plausibility and benefits of OFC over traditional approaches for the control of anthropomorphic robots.*

Related publications:

- *Mitrovic, D., Nagashima, S., Klanke, S., Matsubara, T. and Vijayakumar, S. (2010). Optimal feedback control for anthropomorphic manipulators. In Proceedings of the International Conference on Robotics and Automation (ICRA).*

In **Chapter 4** we address the problem of unknown and changing dynamics (e.g., systematic perturbation, added tool) within the framework of optimality. We propose to combine the OFC framework with a learning methodology for the forward dynamics of the controlled system. We evaluate our proposed method extensively on simulated arms, which exhibit large redundancies, both, in kinematics and in the actuation. We further demonstrate how our approach can compensate for complex dynamic perturbations in an online fashion.

Original contributions:

- *Proposed new ILQG with learned dynamics (ILQG–LD) enabling adaptation within the theory of optimal control.*
- *We show that ILQG-LD scales to high dimensional systems, that it does not sacrifice accuracy and leads to computationally more efficient solutions.*
- *Linking of our model predictions to the study of human adaptation experiments.*

Related publications:

- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2010). Adaptive optimal feedback control with learned internal dynamics models. From Motor Learning to Interaction Learning in Robots, Springer*.
- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2008). Adaptive optimal control for redundantly actuated arms. In Proceedings of the International Conference on the Simulation of Adaptive Behavior (SAB).*
- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2008). Optimal control with adaptive internal dynamics models. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO).*
- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2007). Optimal control with adaptive internal dynamics models. NIPS Workshop: Robotics Challenges for Machine Learning*.

In **Chapter 5** we propose an optimal control strategy under the premise of stochastic dynamics. We present an approach that improves performance by incorporating *learned stochastic information* of the plant, without making prior assumptions about the shape or source of the noise. To test our method we present how the optimal impedance control strategy emerges from minimising stochastic uncertainties of the learned dynamics model. We use an antagonistic robot that we have built specifically to test our control model.

Original contributions:

- *Review of relevant literature of variable impedance actuation with a specific focus on series elastic actuators (SEA).*

- *Design, construction and control of a novel antagonistic SEA, characterised by a simple mechanical setup.*

- *Implementation of impedance control, based on learned stochastic information, which leads to improved control performance over deterministic optimal controllers.*

Related publications:

- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2010). Learning impedance control of antagonistic systems based on stochastic optimisation principles. (to appear in The International Journal of Robotics Research (IJRR)).*

- *Mitrovic, D., Klanke, S., and Vijayakumar, S. (2010). Exploiting sensorimotor stochasticity for learning control of variable impedance actuators. (under review).*

In **Chapter 6** we investigate aspects of human impedance control, which are often used as benchmark for artificial systems. Based on the stochastic OFC-LD framework developed in the previous chapters we formulate a principled strategy for impedance control in human limb reaching tasks. We show that this biologically well motivated model is capable of conceptually explaining a wide range of human impedance control patterns.

Original contributions:

- *Review and critical evaluation of relevant literature for impedance control in the field of biological motor control.*

- *Mathematical formulation of a plausible model of kinematic variability in human limbs. The formulation avoids highly complex simulations and high dimensional state space representations.*

- *Comparative evaluation of our model predictions against humans impedance control patterns from both stationary and adaptation experiments.*

Related publications:

- *Mitrovic, D., Klanke, S., Osu, R., Kawato, M., and Vijayakumar, S. (2010). A computational model of limb impedance control based on principles of internal model uncertainty. (to appear in PLoS One).*
- *Selected talk at: Computational principles of sensorimotor learning, Kloster Irsee, Germany, September 2009.*

In **Chapter 7** we give final conclusions and propose directions for future work.

Chapter 2

Optimal feedback control for high-dimensional movement systems

In this chapter we discuss the background of optimal control theory relevant to the motor control problems that we wish to address. We first provide the basic problem formulation of *optimal control (OC)*. Next we will give a brief overview of OC theory along with some historical notes and we elaborate briefly on the most important approaches to OC. After the basic overview we will discuss the relevant optimality approaches in biological motor control and we will explain the OFC theory for motor coordination. In the third part we will discuss specific OC methods that are particularly well suited for the OFC of high-dimensional limb systems. Our method of choice is ILQG, which will be explained in detail and accompanied with some implementation remarks. At last we will give a brief outlook on current directions of research of OFC for high dimensional systems.

2.1 Introduction

Controlling a system in "the best way possible" is desirable in many applications in a variety of fields, such as aerospace flight, robotics, bioengineering, process control, finance or management sciences. The objective of OC can be summarised in a single sentence as follows:

> *OC is the process of determining control and corresponding state trajectories for a given dynamical system over a period of time in order to minimise a performance index.*

Optimal control - problem formulation

The formulation of an optimal control problem requires following two elements as *input*:

1. **Mathematical model of the controlled system.** This is often referred to as *state space model, process dynamics* or simply *dynamics*[1]. If we assume the state of the system to be represented as $\mathbf{x}$ and the control as $\mathbf{u}$, we can describe a general dynamics function in form of a stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\xi \quad , \quad \xi \sim N(0,\mathbf{I}). \tag{2.1}$$

   Here $d\xi$ is a Gaussian noise process and $\mathbf{F}(\cdot)$ is the so called diffusion coefficient, which indicates how strongly the noise affects which parts of the state and control space. To study deterministic systems we can set $\mathbf{F}(\mathbf{x},\mathbf{u}) = 0$ and the dynamics reduces to $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$.

2. **Performance index.** This is also called *cost function* or *objective function,* and it describes the criteria that one aims to optimise for[2]. The cost function in deterministic form can be written as

$$J(\mathbf{x}(0),\mathbf{u}(\cdot)) = h(\mathbf{x}(T)) + \int_0^T l(\mathbf{x}(t),\mathbf{u}(t),t)dt \tag{2.2}$$

   or

$$J(\mathbf{x}(0),\mathbf{u}(\cdot)) = \lim_{T\to\infty} \frac{1}{T} \int_0^T l(\mathbf{x}(t),\mathbf{u}(t),t)dt, \tag{2.3}$$

   for a task with a *finite* or *infinite horizon* respectively. Apart from the optional final cost $h(\cdot) \geq 0$, which is only evaluated at the final state $\mathbf{x}(T)$, the criterion integrates a cost rate $l(\mathbf{x},\mathbf{u}) \geq 0$ over the course of the movement. That cost may depend on both the system's state $\mathbf{x}$ and control commands $\mathbf{u}$, where the initial state of the system is given as $\mathbf{x}(0)$, and $\mathbf{x}(t)$ evolves according to the system dynamics by applying the commands $\mathbf{u}(t)$. Note that in the case of stochastic dynamics one minimises the *expected* cost, this means we put expectation brackets around the integrals and $h(\cdot)$ in (2.2) and (2.3).

[1] For robotic manipulators this corresponds to the *forward dynamics*.

[2] In the *reinforcement learning (RL)* literature it is common to maximise *reward functions* rather than minimising costs and in general positive costs can be transformed into negative rewards.

In addition to the dynamics and cost function sometimes the *boundary conditions* on states and admissible controls need to be specified as well.

The *output* of OC is an optimal policy π^* that minimises the overall cost

$$\pi^* = \min_{\pi}(J(\mathbf{x}(0), \mathbf{u}(\cdot)) \tag{2.4}$$

subject to the defined dynamics (2.1), the initial state $\mathbf{x}(0) = \mathbf{x}_0$ and target state $\mathbf{x}(T) = \mathbf{x}_T$ and potential additional constraints. In open loop control π^* is a single sequence independent of the states $\mathbf{x}$, whereas in closed loop control π^* represents a control law that depends on the current state. Please note that the OC problem can be formulated for discrete state and time domains (Todorov, 2006).

2.2 Background on optimal control

Optimal control theory has received great attention since the 1950s in many fields in science and engineering. There is a common misconception that optimal control theory has its origins in *dynamic programming (DP)* developed in the 1950 even though OC problems have been studied for over three centuries by mathematicians. Next we will give a brief historical overview with the most important findings in OC. The aim is to summarise the most important developments over the immense body of literature available on that topic. For further (historical) details of OC theory we refer the reader to the review papers by Sussmann and Willems (1997) and Bryson (1996) or some well known standard textbooks on OC theory (Kirk, 1970; Bryson and Ho, 1975; Stengel, 1994; Bertsekas, 1995; Dyer and McReynolds, 1970). The book chapter of Todorov (2006) provides a compact and timely overview and the most relevant mathematical background on OC theory.

2.2.1 History and relevant approaches

Optimal control has its origins in the *calculus of variations (CV)* starting in the 17th century (Bernoulli, Newton, Fermat). Presumably one of the most famous optimal control problems from this time is the "brachystochrone" (i.e., shortest-time curve), which can be solved using CV. It states the problem of finding the shape of a wire, mounted between two points, such that a (frictionless) ball sliding on the wire, accelerated by gravity, traverses the endpoint in minimal time (Sussmann and Willems, 1997). In other words, for which "wire function" $f(x)$ is the descent time T minimised? Solving this problem corresponds to expressing the time of descent T as an

integral involving $f(x)$ and then finding the $f(x)$ that minimises T, which is achieved via the Euler-Lagrange equations. The theory of CV was developed further in the 18th and 19th century by Euler, Lagrange, Jacobi, Hamilton and others and in the early 20th century Bolza (1909) and Bliss (1946) gave the CV the rigorous mathematical structure known today. This was later extended by McShane (1939), which ultimately lead to the development of *Pontryagin's maximum principle* (Pontryagin et al., 1961). The maximum principle is based on the fundamental idea that an optimal trajectory should have neighbouring solutions, i.e., curves that are "slightly off" the optimal solution, that do not lead to smaller costs. Therefore the "derivatives" (or small variations) of the cost function taken along the optimal trajectory should be zero. The maximum principle is typically written in a compact form in terms of *adjoint variables* λ and a *Hamiltonian* function

$$H(\mathbf{x}(t),\mathbf{u}(t),\lambda(t),t) = l(\mathbf{x}(t),\mathbf{u}(t),t) + \lambda^T \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)). \tag{2.5}$$

Using this notation the standard finite horizon cost function can be written as

$$J(\mathbf{x}(0),\mathbf{u}(\cdot)) = h(\mathbf{x}(T)) + \int_0^T \left[H(\mathbf{x}(t),\mathbf{u}(t),\lambda(t),t) - \lambda^T \dot{\mathbf{x}}\right] dt. \tag{2.6}$$

We require that the effect of control variations on the cost is zero at all times, which is reflected in the three optimality conditions known as the maximum principle: If $\{\bar{\mathbf{x}}(t),\bar{\mathbf{u}}(t) : 0 \leq t \leq T\}$ is an optimal trajectory obtained by initialising $\mathbf{x}(0)$ and controlling the system optimally until T, then the following three conditions must hold:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \frac{\partial}{\partial \lambda} H(\bar{\mathbf{x}}(t),\bar{\mathbf{u}}(t),\lambda(t),t) \\ -\dot{\lambda}(t) &= \frac{\partial}{\partial \bar{\mathbf{x}}} H(\bar{\mathbf{x}}(t),\bar{\mathbf{u}}(t),\lambda(t),t) \\ 0 &= \frac{\partial}{\partial \bar{\mathbf{u}}} H(\bar{\mathbf{x}}(t),\bar{\mathbf{u}}(t),\lambda(t),t). \end{aligned} \tag{2.7}$$

It is obvious that the maximum principle provides *necessary conditions* for optimality, i.e., it identifies only candidates for optimal solutions and from the maximum principle alone it is often not possible to conclude whether a trajectory is optimal. OC methods based on the maximum principle are in the literature often referred to as *local methods*, *trajectory based methods* or *open loop methods*. Finding the optimal trajectory of (nonlinear) systems corresponds to solving the set of *ordinary differential equations (ODE)* in (2.7) - usually via numerical methods such as shooting, relaxation, or gradient descent (Stoer and Bulirsch, 1980). However the obtained solutions are local,

not described in closed analytic form and furthermore do not generalise (easily) to stochastic problems.

In the mid 1950 the introduction of *digital computers* allowed for solutions of more complex optimal control problems. Driven by this development, the originally rather theoretical study of OC, was now complemented by more *algorithmic* and *numerical* approaches designed for implementations in computer programs. The development of *dynamic programming (DP)* by Bellman and coworkers marked another important milestone in modern OC (Bellman, 1957). Based on the Hamilton-Jacobi theory, which had been developed 100 years earlier, Bellman introduced the notion of an "optimal value function" $V(\mathbf{x}(t),t)$, which is also known as *cost to go function*. It represents the accumulated value of the performance index starting at state $\mathbf{x}$ at time t progressing optimally towards the final state. The fundamental idea of DP is the so called *principle of optimality*, which states that optimal trajectories remain optimal for intermediate points in time. Going from time t to $t+dt$, this can be formalised as

$$V(\mathbf{x}(t),t) = \min_{\mathbf{u}}\{l(\mathbf{x}(t),\mathbf{u}(t),t)dt + V(\mathbf{x}(t+dt),t+dt)\}. \tag{2.8}$$

Based on this principle one can derive a partial differential equation known as *Hamilton-Jacobi-Bellman equation* for the continuous case[3]

$$\dot{V}(\mathbf{x}(t),t) + \min_{\mathbf{u}}\{\nabla_{\mathbf{x}} V(\mathbf{x}(t),t)\cdot\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) + l(\mathbf{x}(t),\mathbf{u}(t),t)\} = 0, \tag{2.9}$$

subject to the terminal condition

$$V(\mathbf{x}(T),T) = h(\mathbf{x}(T)). \tag{2.10}$$

This partial differential equation represents *sufficient* conditions for optimality. The HJB equation can be similarly formulated under the assumption of stochastic dynamics (Todorov, 2006). Based on this theory it was possible to construct *groups of optimal paths* (as opposed to single optimal paths in Pontryagin's maximum principle) and to associate a control function $\pi^* = \mathbf{u}(\mathbf{x},t)$, which represents the optimal feedback control in state $\mathbf{x}$ at time t. Therefore often dynamic programming is also referred to as *nonlinear optimal feedback control,* which can be formulated in both deterministic and stochastic settings.

A notable milestone in OC theory was the formulation of the *linear quadratic regulator (LQR)* (Kalman, 1960a; Stengel, 1994), which describes a *linear feedback of the state variables* for a system with linear dynamics ($\mathbf{f}(\mathbf{x},\mathbf{u}) = \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{u}$) and quadratic

[3]For the discrete case we get the Bellman equation.

performance index (in both $\mathbf{x}$ and $\mathbf{u}$). Under the additional assumption of Gaussian noise the stochastic extension of LQR, namely the *linear quadratic Gaussian (LQG) compensator*, was introduced (Athans, 1971). The advantage of the LQ formalism is (i) that the solutions can be obtained analytically via the so called Ricatti equations and (ii) that they provide a globally valid (potentially time dependent) feedback control law. Suppose following LQR example

$$\begin{aligned} \text{dynamics:} \quad & d\mathbf{x} = (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u})dt \\ \text{cost rate:} \quad & l(\mathbf{x},\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{final cost:} \quad & h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}^f\mathbf{x}, \end{aligned} \tag{2.11}$$

where $\mathbf{R}$ is symmetric positive definite, $\mathbf{Q}$ and $\mathbf{Q}^f$ are symmetric and $\mathbf{u}$ is unconstrained. This leads to a global optimal control law

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{V}(t) \tag{2.12}$$

where $\mathbf{V}$ is found with the continuous Ricatti differential equation

$$-\dot{\mathbf{V}}(t) = \mathbf{A}^T\mathbf{V}(t) + \mathbf{V}(t)\mathbf{A} - \mathbf{V}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{V}(t) + \mathbf{Q}. \tag{2.13}$$

by initialising it with $\mathbf{V}(T) = \mathbf{Q}^f$ and integrating the ODE backwards in time.

Solving the HJB equations for systems that do not fit into the linear quadratic methodology, involves a discretisation of the state and action space. In practice this is difficult to obtain for realistic problems with continuous state and action space. On the one hand, tiling the state-action space too sparse will lead to poor representation of the underlying plant dynamics. On the other hand, a very fine discretisation leads to a combinatorial explosion of the problem and therefore is not viable for large DoF systems. Bellman called this problem the "curse of dimensionality" and in attempts to avoid that problem some research has been carried out on random sampling in a continuous state and action space (Thrun, 2000), and it has been suggested that sampling can avoid the curse of dimensionality if the underlying problem is simple enough (Atkeson, 2007), as is the case if the dynamics and cost functions are very smooth.

One way to avoid the curse of dimensionality is to restrict the state space to a region that is close to a nominal optimal trajectory. In the neighbourhood of such trajectories the DP problem can be approximated locally using the LQ formalism, which can be solved analytically as described in (2.12) and (2.13). The idea is to compute an optimal trajectory together with a locally valid feedback law and then iteratively improve

this nominal solution until convergence. In some sense this approach can be understood as a *hybrid* between open loop and closed loop OC methods. Well-known examples of such iterative methods are *differential dynamic programming (DDP)* (Dyer and McReynolds, 1970; Jacobson and Mayne, 1970) or the more recent *iterative linear quadratic Gaussian (ILQG)* (Todorov and Li, 2005), which will serve as solution technique of choice in this thesis and will be explained in detail in Section 2.3. However first we wish to discuss in the next section how OC theory has been used to create biological motor control models.

2.2.2 Optimality principles in biological motor control

Computational models provide a very useful tool to model the motor functions observed in biological systems. There are numerous computational models that aim to explain biological movement (Shadmehr and Wise, 2005) and optimality principles are amongst the most successful ones. The main advantage is that optimality describes task goals in form of intuitive high level objective functions and the actual movement plan and execution (including redundancy resolution) falls out of the optimisation. This provides a principled "task driven" approach to the study of observed actions, which is formulated in the mathematically coherent and well understood framework of OC. Indeed human motion, for example in visually guided arm reaching tasks, are highly stereotyped and researchers have been intrigued to discover the principles behind this action selection. From a biological perspective it is well justified to assume *optimisation* as the underlying principle since our sensorimotor system is a result of constant performance improvement via evolution, learning and adaptation. However understanding what constitutes the choice of cost function and optimisation variables is nontrivial and this question has played a central role in the biological motor control community for nearly three decades. Here we give an overview of the most important OC findings of biological motor control with a focus on limb reaching tasks. A review of optimality principles with special focus on biological motor control can be found in Engelbrecht (2001).

The biological optimal control models discussed in this section make certain simplifying assumptions and abstractions in order to be computationally tractable. For example, typically idealised point-to-point reaching tasks are considered, which can easily be reproduced in psychophysical experiments with human subjects. However predicting a wide range of human motion outside of idealised lab settings is very diffi-

cult to achieve using only standard optimal control methods.

Open loop models

Traditionally many optimal control models use open loop optimisation in which a motor task is separated into trajectory planning and execution of the motor plan. In this decoupled setting the trajectory is being optimised with the constraints of point boundary conditions (for targets or via point) in the system dynamics. The obtained optimal trajectories then are tracked with some separate feedback mechanism.

Based on the observation that humans produce smooth point-to-point movement in Cartesian space, which is also improved with practice, it was proposed that the goal of motor coordination is to produce as smooth hand trajectories as possible. In order to achieve this Flash and Hogan (1985) presented the *minimum jerk* model in which the cost function depends on jerk, which is defined as the rate of change of accelerations. Mathematically this was defined as the squared first derivative of the Cartesian hand acceleration

$$J = \frac{1}{2}\int_0^T \left(\left(\frac{d^3x(t)}{dt^3} \right)^2 + \left(\frac{d^3y(t)}{dt^3} \right)^2 \right) dt. \tag{2.14}$$

Here $x(t), y(t)$ denote the Cartesian coordinates of the hand position at time t and T denotes the final time step. This formulation is independent of the dynamics of the motion system and the optimal trajectory is determined only from the kinematics. Minimum jerk trajectories are straight-lines in task space and follow a bell-shaped velocity profile, properties that match to empirical biological data, recorded in fast reaching movements. However the model fails to explain curved Cartesian trajectories observed in wider movement ranges and it is not clear why people should aim for generating smooth movements in the first place.

An alternative model, namely the *minimum torque-change* model, was proposed by Uno et al. (1989). Here the cost function is setup to minimise the rate of change of the torques, and therefore depends on the dynamics of the system rather than only on kinematic properties. For a system with n-joints the minimum torque-change cost function is defined as

$$J = \frac{1}{2}\int_0^T \sum_{i=1}^{n} \left(\frac{d\boldsymbol{\tau}_i}{dt} \right)^2 dt \tag{2.15}$$

where $\frac{d\boldsymbol{\tau}_i}{dt}$ represents the rate of torque-change in the i-th joint. The notion of minimum torque change implies smooth trajectories and to some extent also a low energy consumption as excessively large commands are avoided. This notion also is well mo-

tivated from a biomechanical point of view as unnecessary wear and tear of the muscular system would be avoided. However even though minimum-jerk and minimum torque-change models are capable of predicting many aspects of biological motion the question remains how jerk or torque-change could be integrated by the CNS.

A major drawback of the described OC models is that they are fundamentally incapable of explaining motor variability. The *minimum end point variance (MV)* approach (Harris and Wolpert, 1998) of eye and arm movement incorporates an assumption that the motor control signals are corrupted by noise, the variance of which is proportional to the size of the control signal. In this model the variance of the movement end point is minimised over a short time period after the movement. The cost function of MV is defined as

$$J = \int_{T}^{\hat{T}} \langle (x(t) - \langle x(t) \rangle)^2 \rangle dt \tag{2.16}$$

where $x(t)$ denotes the system position, T the movement time $\hat{T}$ the post movement period, and $\langle \cdot \rangle$ the expectation over the control noise. Therefore, in the presence of control dependent noise movements that require large motor signals (e.g., fast movements) increase the noise in the systems and lead to deviations of the desired trajectory. In contrast, slow motions would keep the noise level low and lead to more precise motion. Therefore signal-dependent noise inherently can be linked to the speed-accuracy trade-off as described by *Fitts' Law* (Fitts, 1954). Most importantly the minimum end point variance model is able to predict variability patterns in the human motor system. The cost function does not rely on complex mathematical parameters, such as torque change and jerk, but rather on the variance of the final position or the consequences of this accuracy. These cost parameters are assumed to be directly available to the CNS through vision and proprioception.

For longer movement durations the MV model shows to be less reliable since, as can be expected from all open loop methods, they ignore online sensory feedback in the optimisation. This fundamental drawback motivated the study of closed loop optimisation mechanisms for biological motor control as described in the next section.

Closed loop models - Optimal Feedback Control (OFC)

As mentioned earlier the sensorimotor system suffers from a multitude of noise sources which demands for closed loop optimisation techniques in which there is no explicit separation between the planning and motor execution for the completion of a task. Todorov and Jordan (2002) recognised that the presence of feedback is a key com-

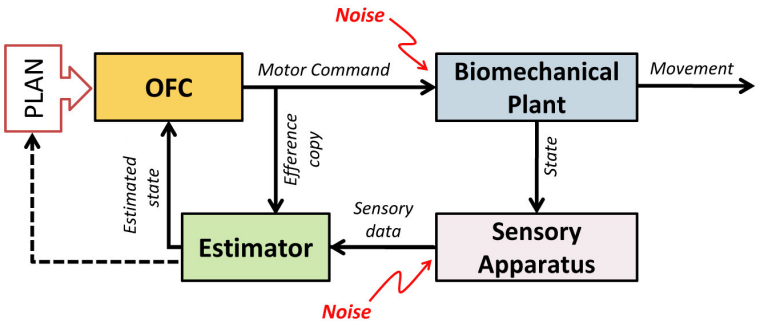


Figure 2.1: Schematic of the optimal feedback control for biological movement systems (Todorov and Jordan, 2002). The movement plan is described as a cost function and the optimal feedback controller produces the desired optimal motor commands to control the plant. This optimal controller incorporates current state estimations, which are achieved through a combination of feedback signals and efferent copies from a forward internal model converting motor commands to state variables.

ponent for motor coordination, and they proposed a closed loop mechanism, namely *optimal feedback control (OFC)*, which has found large support in the biological motor control community (Todorov, 2004; Shadmehr and Krakauer, 2008). For a review please see Scott (2004). This formulation was a breakthrough because it was able to link the most important biological motor control concepts, such as costs, noise, expected rewards, sensory feedback and internal models into a coherent mathematical framework (Shadmehr and Krakauer, 2008). The OFC framework as proposed by Todorov and Jordan (2002) is depicted in Fig. 2.1.

OFC with a performance index that minimises kinematic error and control effort (i.e., energy consumption) predicts a multitude of human reaching movement patterns, e.g., bell-shaped velocity patterns, trajectory curvatures, goal-directed corrections (Liu and Todorov, 2007), multi-joint synergies (Todorov and Ghahramani, 2004) and variable but successful motor performance. Furthermore OFC has also been successfully in various other biological motor behaviours other than reaching, such as spinal reflexes in the limbs of cats during perturbations (He et al., 1991), human postural balance (Kuo, 1995) or bimanual coordination tasks (Diedrichsen, 2007).

A key property of OFC is that errors are corrected by the controller only if they adversely affect the task performance, otherwise they are neglected. In other words, if the system experiences perturbations in the nullspace of the system they will be neglected by the feedback controller. Todorov and Jordan (2003) called this the *minimum intervention principle*, which is an important property especially in systems that

suffer from control dependent noise, since task-irrelevant correction could destabilise the system beside expending additional control effort. In Chapter 3 we will present the advantageous properties of the minimum intervention principle for the control of anthropomorphic robots.

While the theoretical importance of OFC to biological motor control is without doubt significant, the available computational methods for solving OFC problems still are not capable of efficiently determining globally valid optimal control laws for large DoF nonlinear systems. In practice the controlled systems are often approximated by linear dynamics in order to make the problem computationally tractable. For example in Diedrichsen (2007) or in Liu and Todorov (2007) the authors successfully used OFC under linear dynamics assumptions to explain human reaching experiments on a high level of observation (i.e., end-effector trajectories, velocity profiles). However if we wish to analyse more detailed effects (e.g., single muscle-signals, co-contraction in joints) for biomechanical systems linearity assumptions in the dynamics may have simplification-artifacts, the effects of which are hard to predict. Furthermore, since we aim to transfer OFC to realistic robotic systems, linear dynamics prove to be a serious limitation.

The next section discusses iterative optimal feedback control methods that can be applied to achieve computationally efficient (near) optimal behaviour for highly complex systems.

2.3 Iterative optimal control methods

Biologically inspired systems usually have large DoF, typically are highly non-linear and cannot be represented to fit in the linear quadratic framework. We therefore resort to algorithms that compromise between open loop and closed loop optimisation, that is, algorithms which iteratively compute an optimal trajectory together with a locally valid feedback law.

Differential dynamic programming (DDP) (Dyer and McReynolds, 1970; Jacobson and Mayne, 1970) is a well-known successive approximation technique for solving nonlinear deterministic dynamic optimisation problems. This method uses second order approximations to perform dynamic programming in the neighbourhood of a nominal trajectory. We briefly introduce this method as a prototypical example for an iterative OFC method: Following the principle of optimality the algorithm uses a value function to generate optimal solutions. Given a control sequence $\mathbf{u}$ and a state

sequence $\mathbf{x}$ the value function (in the discrete time case) is defined as

$$V(\mathbf{x}_k, k) = h(\mathbf{x}_T) + \sum_{k}^{T-1} l(\mathbf{x}_k, \mathbf{u}_k, k). \tag{2.17}$$

It represents the accumulated future cost $l(\mathbf{x}_k, \mathbf{u}_k, k)$ from time k to the final cost $h(\mathbf{x}_T)$. In DDP solutions are obtained by iteratively improving nominal trajectories and each iteration performs a succession of *backward* and *forward* sweeps in time. First, DDP is initialised using a nominal control sequence $\bar{\mathbf{u}}$ which also determines a corresponding nominal state sequence $\bar{\mathbf{x}}$ through the use of the dynamics function $d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. In the *backward* iteration a control law is obtained by approximating the time-dependent value function along the current nominal trajectory $\bar{\mathbf{x}}$. The approximation is achieved by maintaining a second order local model of a so called *Q-function*

$$Q(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = l(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) + V_{k+1}(\mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k)). \tag{2.18}$$

More specifically the quadratic approximation of the Q-function can be formulated as

$$Q(\bar{\mathbf{x}}_k + \delta\bar{\mathbf{x}}, \bar{\mathbf{u}}_k + \delta\bar{\mathbf{u}}) \approx Q_0 + Q_{\mathbf{x}}\delta\mathbf{x} + Q_{\mathbf{u}}\delta\mathbf{u} + \frac{1}{2}[\delta\mathbf{x}^T \ \delta\mathbf{u}^T] \begin{bmatrix} Q_{\mathbf{xx}} & Q_{\mathbf{xu}} \\ Q_{\mathbf{ux}} & Q_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}, \tag{2.19}$$

where the vector subscripts indicate partial derivatives. Please note that the Q-function is described in terms of deviations $\delta\mathbf{x}$ and $\delta\mathbf{u}$ of the nominal trajectory. After obtaining the local model of Q, the minimising $\delta\mathbf{u}$ can be found directly as

$$\delta\bar{\mathbf{u}} = \min_{\delta\mathbf{u}} \{Q(\bar{\mathbf{x}}_k + \delta\bar{\mathbf{x}}, \bar{\mathbf{u}}_k + \delta\bar{\mathbf{u}})\} = -Q_{\mathbf{uu}}^{-1}(Q_{\mathbf{u}} + Q_{\mathbf{ux}}\delta\mathbf{x}). \tag{2.20}$$

Next, in the *forward* run an *improved* nominal control sequence of the form $\bar{\mathbf{u}}^{new} = \bar{\mathbf{u}} + \delta\bar{\mathbf{u}}$ can be obtained and the next iteration of DDP begins. Iterations are repeated until the cost cannot be reduced anymore. DDP has second-order convergence and is numerically more efficient than implementations of Newton's method (Murray and Yakowitz, 1984).

A more recent algorithm is the *iterative linear quadratic regulator (ILQR)* (Li and Todorov, 2004). This algorithm uses iterative linearisation of the nonlinear dynamics around a nominal trajectory, and solves a locally valid LQR problem to iteratively improve the trajectory. However, this method is still deterministic and cannot deal with control constraints or non-quadratic cost functions. A recent extension to ILQR, the *iterative linear quadratic Gaussian (ILQG)* framework (Todorov and Li, 2005),

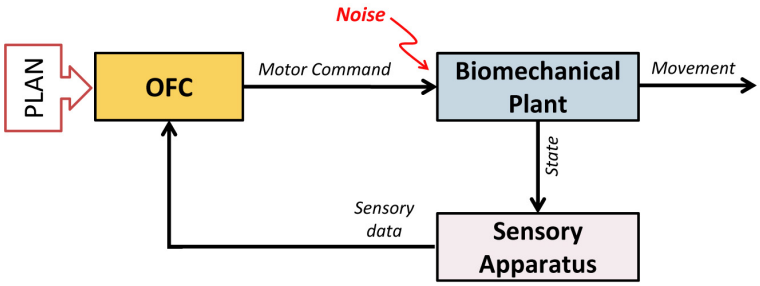


Figure 2.2: Schematic of the OFC under the assumption of full observability, i.e., the sensor readings are not corrupted by any noise.

allows to model nondeterministic dynamics by incorporating a Gaussian noise model. Furthermore it supports control constraints such as non-negative muscle activations or upper control boundaries. The ILQR/ILQG framework is shown to be computationally significantly more efficient than DDP (Li and Todorov, 2004). It has also been previously tested on biological motion systems and therefore is the approach for us to investigate further.

While the original OFC problem was formulated for partially observable systems, i.e., the state observations are corrupted by noise and need an optimal estimation process, here we will take a simplified view and assume fully observable dynamics as depicted in Fig. 2.2. The reason for this is that the well known duality of optimal control and estimation (i.e., Kalman filter) established in the linear quadratic case (Kalman, 1960b) is difficult to transfer to non-linear systems (Todorov, 2008). Furthermore in this thesis we study robotic systems that do not suffer from significant observation noise and the used sensors (i.e., joint angle potentiometers) have very high accuracy.

2.3.1 Iterative Linear Quadratic Gaussian - ILQG

This section explains the ILQG framework based on the description given in Todorov and Li (2005). We consider reaching movements of manipulators as a finite time horizon problems of length $T = k\Delta t$ seconds, where k are the discretisation steps and Δt is the simulation rate. For optimising and carrying out a movement, one also has to define a cost function (where also the desired final state is encoded). The expected accumulated cost when following policy π from time t to T is

$$v^{\pi}(\mathbf{x}(t),t) = \left\langle h(\mathbf{x}(T)) + \int_t^T l(\mathbf{x}(\tau),\pi(\mathbf{x}(\tau),\tau),\tau)d\tau \right\rangle. \tag{2.21}$$

ILQG then finds the control law π^* with minimal $v^{\pi}(0,x_0)$ by iterating in 4 steps until convergence:

Step 1: One starts with an initial time-discretised control sequence $\bar{\mathbf{u}}_k \equiv \bar{\mathbf{u}}(k\Delta t)$, which can be chosen arbitrarily (e.g., gravity compensation, or zero sequence). The initial control sequence is applied to the deterministic forward dynamics to retrieve an initial trajectory $\bar{\mathbf{x}}_k$, where

$$\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \Delta t\, \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k). \tag{2.22}$$

Step 2: By linearising the discretised dynamics (2.1) around $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$ and by subtracting (2.22), one obtains a dynamics equation for the deviations $\delta\mathbf{x}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$ and $\delta\mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$:

$$\begin{aligned} \delta\mathbf{x}_{k+1} &= \mathbf{A}_k\delta\mathbf{x}_k + \mathbf{B}_k\delta\mathbf{u}_k + \mathbf{C}_k(\delta\mathbf{u}_k)\boldsymbol{\xi}_k && (2.23)\\ \mathbf{A}_k &= \mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\bar{\mathbf{x}}_k} && (2.24)\\ \mathbf{B}_k &= \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k} && (2.25) \end{aligned}$$

The last summand in (2.23) represents the case when we assume a dynamics model with noise. $\boldsymbol{\xi}_k$ is randomly drawn from a zero mean Gaussian with covariance $\Omega^{\xi} = I$. $\mathbf{F}^{[i]}$ represents the i-th column of the matrix $\mathbf{F}$.

$$\begin{aligned} \mathbf{C}_k(\partial\mathbf{u}_k) &= [\mathbf{c}_{1,k} + \mathbf{C}_{1,k}\partial\mathbf{u}_k,\ \ldots\ , \mathbf{c}_{p,k} + \mathbf{C}_{p,k}\partial\mathbf{u}_k] && (2.26)\\ \mathbf{c}_{i,k} &= \sqrt{\Delta t}\mathbf{F}^{[i]}; \quad \mathbf{C}_{i,k} = \sqrt{\Delta t}\frac{\partial \mathbf{F}^{[i]}}{\partial \mathbf{u}} \end{aligned}$$

The variance of Brownian motion is known to grow linearly with time and therefore the standard deviation of the discrete-time noise scales as $\sqrt{\Delta t}$.

Similarly to the linearised dynamics in (2.23) one can derive an approximate cost function which is quadratic in $\delta\mathbf{u}$ and $\delta\mathbf{x}$ such that

$$cost_k = q_k + \delta\mathbf{x}_k^T\mathbf{q}_k + \frac{1}{2}\delta\mathbf{x}_k^T\mathbf{Q}_k\delta\mathbf{x}_k + \delta\mathbf{u}_k^T\mathbf{r}_k + \frac{1}{2}\delta\mathbf{u}_k^T\mathbf{R}_k\delta\mathbf{u}_k + \delta\mathbf{u}_k^T\mathbf{P}_k\delta\mathbf{x}_k \tag{2.27}$$

where

$$\begin{aligned} q_k &= \Delta t\ l; & \mathbf{q}_k &= \Delta t\ \frac{\partial l}{\partial \mathbf{x}} \\ \mathbf{Q}_k &= \Delta t \frac{\partial^2 l}{\partial \mathbf{x}\partial \mathbf{x}}; & \mathbf{P}_k &= \Delta t \frac{\partial^2 l}{\partial \mathbf{u}\partial \mathbf{x}} \\ \mathbf{r}_k &= \Delta t \frac{\partial l}{\partial \mathbf{u}}; & \mathbf{R}_k &= \Delta t \frac{\partial^2 l}{\partial \mathbf{u}\partial \mathbf{u}}. \end{aligned} \tag{2.28}$$

Thus, in the vicinity of the current trajectory $\bar{\mathbf{x}}$, the two approximations (2.23) and (2.27) form a "local" LQG problem, which can be solved analytically and yields an affine control law

$$\delta \mathbf{u}_k = \pi_k(\delta \mathbf{x}) = \mathbf{l}_k + \mathbf{L}_k \delta \mathbf{x}_k. \tag{2.29}$$

This control law has special form: since it is defined in terms of deviations of a nominal trajectory and since it needs to be implemented iteratively it consists of an open loop component $\mathbf{l}_k$ and a feedback-component $\mathbf{L}_k \delta \mathbf{x}_k$.

Step 3: To compute the mentioned control law relative to the current nominal trajectory, the optimal cost to go function is approximated iteratively backwards in time

$$v_k(\delta \mathbf{x}) = s_k + \delta \mathbf{x}^T \mathbf{s}_k + \tfrac{1}{2} \delta \mathbf{x}^T \mathbf{S}_k \delta \mathbf{x}. \tag{2.30}$$

At the final time step K (i.e., the first step of the backwards iteration) the cost to go parameters are defined by $\mathbf{S}_K = \mathbf{Q}_K, \mathbf{s}_K = \mathbf{q}_K, s_k = q_k$. If $k < K$ the parameters are recursively updated in following 3 sub-steps (a-c).

For each time step:

- **a)** Compute shortcuts $\mathbf{g}, \mathbf{G}, \mathbf{H}$, by

$$\begin{aligned} \mathbf{g} &= \mathbf{r}_k + \mathbf{B}_k^T \mathbf{s}_{k+1} \sum_i \mathbf{C}_{i,k}^T \mathbf{S}_{k+1} \mathbf{c}_{i,k} \\ \mathbf{G} &= \mathbf{P}_k + \mathbf{B}_k^T \mathbf{S}_{k+1} \mathbf{A}_k \\ \mathbf{H} &= \mathbf{R}_k + \mathbf{B}_k^T \mathbf{S}_{k+1} \mathbf{B}_k + \sum_i \mathbf{C}_{i,k}^T \mathbf{S}_{k+1} \mathbf{C}_{i,k} \end{aligned} \tag{2.31}$$

  .

- **b)** Find affine control law by minimising:

$$a(\delta \mathbf{u}, \delta \mathbf{x}) = \delta \mathbf{u}^T (\mathbf{g} + \mathbf{G} \delta \mathbf{x}) + \frac{1}{2} \delta \mathbf{u}^T \mathbf{H} \delta \mathbf{u} \tag{2.32}$$

  with respect to $\delta \mathbf{u}$ leading to

$$\delta \mathbf{u} = \pi_k(\delta \mathbf{x}) = -H^{-1}(\mathbf{g} + \mathbf{G} \delta \mathbf{x}). \tag{2.33}$$

In reference to the proposed control law (2.29), the open loop component corresponds to $\mathbf{l} = -H^{-1}\mathbf{g}$ and the feedback component to $\mathbf{L} = -H^{-1}\mathbf{G}$ repsectively. Please note that in (2.33), H is a modified version of the Hessian $\mathbf{H}$ such that there are no negative eigenvalues in H, which would make the cost function (arbitrarily) negative. For details about the modification please refer to Todorov and Li (2005).

- **c)** Update the cost to go approximation parameters:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{Q}_k + \mathbf{A}_k^T \mathbf{S}_{k+1} \mathbf{A}_k - \mathbf{G}^T \mathbf{H}^{-1} \mathbf{G} \\ \mathbf{s}_k &= \mathbf{q}_k + \mathbf{A}_k^T \mathbf{s}_{k+1} - \mathbf{G}^T \mathbf{H}^{-1} \mathbf{g} \\ s_k &= q_k + s_{k+1} + \frac{1}{2} \sum_i \mathbf{c}_{i,k}^T \mathbf{S}_{k+1} \mathbf{c}_{i,k} - \frac{1}{2} \mathbf{g}^T \mathbf{H}^{-1} \mathbf{g}. \end{aligned} \tag{2.34}$$

Step 4: After having found the affine control law $\pi(\delta\mathbf{x})$, we apply it to the linearised dynamics (2.23) obtaining the optimal control deviations $\delta\mathbf{u}_k$ for each time step k from the nominal sequence $\bar{\mathbf{u}}_k$. We then obtain the new "improved" torque sequence as follows $\bar{\mathbf{u}}_k = \bar{\mathbf{u}}_k + \delta\mathbf{u}_k$. At last we apply $\bar{\mathbf{u}}$ to the system dynamics (2.1) and compute the total cost along the trajectory. If the resulting cost has converged (i.e., is not decreasing) ILQG is finished. Otherwise we repeat **Step 1** and begin a new iteration with the new control sequence $\bar{\mathbf{u}}$. Within the the main loop of ILQG a factor λ is maintained used for a modified Levenberg-Marquardt optimisation.

After convergence ILQG returns an optimal control sequence $\bar{\mathbf{u}}$, a corresponding state sequence $\bar{\mathbf{x}}$ as well as the optimal feedback control law $\mathbf{L}$. Appendix A elaborates on the ILQG algorithm in form of a commented *MATLAB* source code.

2.3.2 Implementation aspects

In this thesis we wish to study finite time horizon problems of nonlinear, potentially stochastic, dynamic systems under a variety of cost functions. For such scenarios ILQG is very well suited as it does not rely on quadratic cost function formulations. Therefore one can easily for example define targets in *task space* through the use of the forward kinematics function, which is typically non-quadratic. Due to the approximative nature however both, dynamics function and cost function need to have certain smoothness properties, i.e., they must not be discontinuous or contain very steep step-

like properties. One should also note that the current implementation of ILQG assumes costs to be ≥ 0.

In our implementation we use finite differences to compute the gradients of the dynamics. While this approach offers more flexibility due to its general applicability, it imposes significant higher computational costs especially for high-dimensional systems. We will come back to this issue in Chapter 4.

Obtaining good optimisation results with ILQG requires significant practical experience and a good understanding of the optimisation task to be solved. More specifically the algorithm has many open parameters, such as reaching time, cost function parameters, convergence constants or simulation parameters, which all influence the optimisation outcome. In many cases the obtained results, due to its local nature of ILQG, depend also on the chosen initial control sequence $\bar{\mathbf{u}}$.

The last implementation aspect worth mentioning is the way ILQG handles control boundaries. As shown in Appendix A.2, ILQG truncates the controls to the defined boundary value and sets the feedback gains to zero, whenever the control boundaries are reached. While for deterministic systems this conservative approach is reasonable, in the case of stochastic systems one would try to avoid zero feedback gains as this may lead to undesirable outcomes. Therefore in stochastic settings one tries to avoid to reach the control boundaries by setting the weights in the cost function on control cost accordingly.

2.3.3 Beyond ILQG

There are a number of other approximative methods that have been proposed in recent years. For example in Theodorou et al. (2010b) the authors propose *stochastic DDP (SDDP)* by explicitly deriving the second order expansions of the cost to go function for systems with state and/or control dependent noise of the form $d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\xi$. These derivations show that standard DDP can be understood as a special instance of SDDP. From a practical perspective SDDP has so far only been applied to one-dimensional systems in simulation and in the current form it does not support neither state nor control constraints. However constrained versions of standard (deterministic) DDP have been proposed previously (Yakowitz, 1986; Lin and Arora, 1991).

Local OFC methods like ILQG or SDDP have certain limitations when it comes to the study of stochastic systems. As shown in Todorov and Tassa (2009) for systems

that suffer from additive noise, i.e., the dynamics is of the form $d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}d\xi$, second order methods cannot be employed as they are blind to such additive noise. It turns out that, for such systems, the value function approximations are state and control independent and therefore have no direct effect on the optimisation process (for details see Todorov and Tassa (2009)). This drawback motivated the development of an algorithm called *iterative local dynamic programming (ILDP)*, which allows for higher order approximations. In essence ILDP, like ILQG and DDP, is based on solving approximate dynamic programming in the neighbourhood of nominal trajectories and then iteratively improving the solutions. However ILDP uses general basis functions rather than quadratic functions to approximate the value function. More specifically the value function approximations are achieved using a collocation method with samples taken around the nominal trajectory. With the correct choice of basis function parameters one can find approximative OFC solutions iteratively that address the problem of "blindness" towards additive noise. As with all methods discussed in this section the application seems non-trivial and has been limited in the literature to idealised low-dimensional simulation scenarios. In particular the sampling based approach (i.e., collocation cloud) in ILDP seems to require a potentially large number of samples and an intelligent choice of the model parameters in order to give accurate optimisation results (Todorov and Tassa, 2009).

Motivated by the intractability of solving general stochastic control problem, Kappen (2005) discovered a class of continuous non-linear stochastic control problems that can be solved efficiently using concepts from statistical physics. For settings in which the controls act linearly and additively and the control costs are quadratic, the finite time horizon optimal control problem reduces to the computation of *path integrals*. The framework of stochastic optimal control with path integrals has been successfully used for example to model animal behavior and learning (Kappen, 2007) and recently was extended to the optimal control of robotic systems (Theodorou et al., 2010a).

2.4 Discussion

In this chapter we have discussed the main aspects of OC theory. After having defined the problem of OC we have provided a historical and topical overview of the most important findings in OC. More specifically we elaborated upon the difference between open loop and closed loop OC methods. The latter, also known as OFC is of special importance for the control of stochastic systems and has recently found large atten-

tion in the biological motor control community. We then extended our discussion to iterative OFC methods, which avoid the computational problems that global methods face. In particular we focused on ILQG, which is the OFC technique used throughout this thesis. We explained the algorithm in more detail along with some practical implementation remarks.

Chapter 3

Optimal feedback control for anthropomorphic manipulators

In this chapter we address the problem related to the control of movement in large *degree of freedom (DoF)* anthropomorphic manipulators, with specific emphasis on (target) reaching tasks. This is challenging mainly due to the large redundancies that such systems exhibit and we wish to employ OFC as a principled strategy to resolve such redundancies and to control the system in a compliant and energy efficient manner.

3.1 Introduction

Prototypic control architectures for robotic manipulators consists of three components (An et al., 1988): (i) The *planning* of a trajectory in task space, (ii) the transformation of the plan into joint angle space using an *inverse kinematics* mapping and (iii) the *control*, i.e., the execution of the movement plan on the robot. Optimal control has been used previously on robotic manipulators and in such scenarios optimisation is restricted to the selection process and the redundancy resolution of the movement trajectory with respect to some optimisation criteria (e.g., Hollerbach and Suh (1985); Sahar and Hollerbach (1986); Nakamura (1990); for a review see Nenchev (1989)). Therefore a common approach is to use open loop optimisation as defined in Chapter 2, which means that only the kinematics are resolved using optimal control, whereas control is achieved via traditional feedback or feedforward control methods. It is worth mentioning that alternative reactive control strategies have been previously proposed in robotics. One prominent example is the so called *navigation function* approach pro-

posed by Koditschek and Rimon (1990). The idea is to specify potential functions that encode for a known target configuration with low potential value and for known obstacles with higher potential values. Assuming a unique global minimum at the target, the robot can reach the target by following the negative gradient of the potential surface. If the potential function is formulated appropriately its negative gradient can serve as low level feedback control law solving the path planning and control problem simultaneously.

In the previous chapter we have discussed the beneficial properties of OFC as a principled motor control strategy for highly redundant systems. An interesting question therefore is if OFC schemes could be applied to control anthropomorphic manipulators and if this brings any advantages in comparison to traditional control schemes. To date to the best of our knowledge there are no reported implementations of OFC on real robotic manipulator systems and in the following we try to identify potential reasons for this.

Many manipulators control scenarios have been motivated from the viewpoint of industrial applications. Intuitive examples for such are robots working in manufacturing lines performing tasks like welding or assembly. Here the main objectives of the motion plans are *precision*, *repeatability* and *speed*, requirements which can be achieved very well with kinematic control methods. Features like *compliance* and *energy consumption* often are secondary as industrial robots usually operate in controlled environments with continuous energy supply. Indeed kinematics models are much easier to identify than accurate dynamics model as would be required in OFC. In fact for many robotic manipulators the inertial parameters are unknown (even to the manufacturers) and apart from some notable exceptions most commercial manipulators do not allow for direct torque control. An accurate system identification is very difficult (An et al., 1988) as the details of robot dynamics, such as nonlinear friction properties or detailed motor dynamics and gearing, are hard to model in simulation and obtained optimisation results may not transfer well to the real robot. Another aspect is the lack of suitable methods for computing OFC laws for nonlinear high dimensional systems. Even though approximative OFC methods like DDP have been around since the 1970s, they somehow did not find wide spread attention in the robotics community. Only recently these approximative methods have been “rediscovered” in the domain of anthropomorphic manipulators. However, so far they only have been studied in idealised simulation scenarios (Liu and Todorov, 2009; Li, 2006; Todorov et al., 2005). In other robotics domains, such as biped walking or swimming the role of (passive)

dynamic properties is more central and closed loop OC methods based for example on RL or on approximative methods like DDP and ILQG have been applied successfully in simulation and on real robotic systems (Tassa et al., 2007; Tedrake, 2004; Morimoto and Atkeson, 2003).

For anthropomorphic manipulators the control criteria are different compared to typical industrial settings. Suppose we wish to control a humanoid robot to reach towards a bottle of water on a table (at which people are dining). Important constraints for this movement plan are *reaching time*, the *end-effector position/orientation* at the target and a the *arm stability* at the end of the motion (i.e., arm stops at target). For most daily life task like this one, exact trajectory tracking is not a crucial aspect as long as the robot's physical and environmental limits like joint angles, self collision and known obstacles are satisfied. To account for possible unpredictable obstacles the movement should be compliant such that collisions are not destructive, neither for a human nor for the robot. To achieve compliance one typically aims to control the robot with low corrective gains and an accurate feed-forward plan.

Here we show that we can achieve such a control strategy for redundant anthropomorphic manipulators using OFC theory under the basic premise of minimal energy consumption and compliance. To achieve optimal controls that are applicable on real systems we introduce a specific cost function that accounts for the physical constraints of the controlled plant. Using ILQG we then can optimally control a high-dimensional anthropomorphic robot without having to specify an explicit inverse kinematics, inverse dynamics or feedback control law. Another beneficial property of such OFC, that typically minimise for task error and energy consumption, is that errors are only corrected by the controller if they adversely affect the task performance, otherwise they are neglected (minimum intervention principle (Todorov and Jordan, 2003)). Therefore redundant degrees of freedom, often a nuisance for kinematic path planning, in OFC are actually exploited in order to decrease the cost. This is an important property especially for systems that demand low energy consumption and compliant motion, such as mobile humanoid robots interacting safely in a human environment.

In the experimental section we apply the local OFC law to the *Barrett WAM*[1], a modern anthropomorphic manipulator, and we highlight the benefits of the OFC motor control strategy over traditional (open loop) optimal controllers: The presented approach proves to be significantly more energy efficient and compliant, while being accurate with respect to the task at hand. To the best of our knowledge this is the first

[1]WAM stands for Whole Arm Manipulator.

OFC implementation on a real high-dimensional (redundant) manipulator.

3.2 Robot model and control

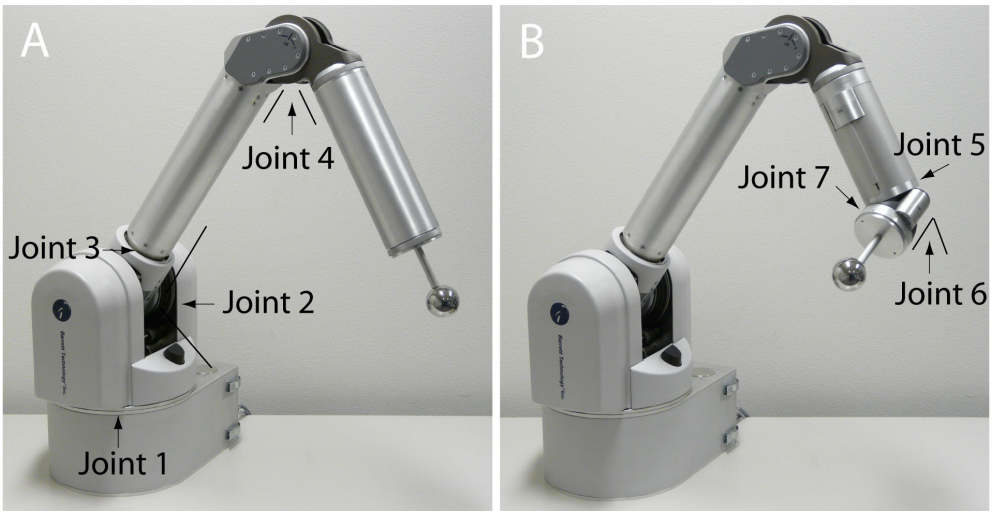


Figure 3.1: The anthropomorphic manipulator (Barrett WAM) used for our experiments. (A): 4 DoF setup; (B): 7 DoF setup (with wrist attached).

In this chapter we use the WAM (Barrett Technology Inc.) (Fig. 3.1) as an implementation platform. The WAM is a cable driven 7 DoF anthropomorphic manipulator (4 DoF without wrist), with a reach of about $1m$ and a payload of $4kg$. The platform is well suited for implementing dynamics model based control (like OFC) since the inertial parameters are publicly available and motor torques can be directly commanded to the WAM. On the sensing side the platform has joint position encoders but offers no joint torque or other external sensors.

3.2.1 Reaching with ILQG

Let $\mathbf{x}_t$ denote the state of a plant and $\mathbf{u}_t$ the applied control signal (i.e., joint torque) at time t. The state consists of the joint angles $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ of the robot. We can express the system dynamics in deterministic form as

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})dt. \tag{3.1}$$

OFC can also be formalised for stochastic dynamics with partial observability (Li and Todorov, 2007). However we will ignore stochasticity in this chapter as we would

require a system noise and estimation noise model of the real hardware. The WAM has negligibly small control and sensor noise, i.e., the same movement plans can be performed with high fidelity and repeatability. We will discuss stochastic optimisation scenarios in Chapters 5 and 6.

We study reaching movements of a manipulator as a finite time horizon problem of length $K = k\Delta t$ seconds. Typical values are between $k = 100$ and $k = 200$ discretisation steps with a simulation rate of $\Delta t = 0.01$. We assume that we have identified an accurate forward dynamics model $\mathbf{f}(\mathbf{x}, \mathbf{u})$ of our plant (see Section 3.2.2). We define a discrete cost function v encoding a task, where the manipulator has to move and stop at the target using a minimal amount of energy (Todorov and Li, 2005):

$$\begin{aligned} v &= w_p \,|\, \mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + \\ &\quad \Delta t \sum_{k=0}^{K} \underbrace{w_e \,|\, \mathbf{u}_k \,|^2}_{=v_k}. \end{aligned} \tag{3.2}$$

The factors for the target position accuracy (w_p), the stopping condition (w_v), and the energy term (w_e) weight the importance of each component. Here we approximate energy with joint torque consumption, and we ignore the efficiency factors of the motors. Further $\mathbf{r}(\mathbf{q})$ denotes the forward kinematics and $\mathbf{r}_{tar}$ the Cartesian coordinates of our reaching target. The choice of the cost function determines the behaviour of the system and encodes the task. Inspired by the study of biological systems where metabolic cost is crucial, the minimum energy criterion (Nelson, 1983) is also a very appealing strategy for mobile robots since battery life is limited. Furthermore minimum energy implies smooth trajectories, reduced stress on the actuators, and joint compliance through low corrective gains, which is also desired in our application.

With the dynamics and the cost function identified the OFC control law can be computed using ILQG. Fig. 3.2 shows the ILQG control scheme. Its components consist of an open loop torque sequence $\bar{\mathbf{u}}$, a corresponding state sequence $\bar{\mathbf{x}}$ and a locally valid optimal feedback control law $\mathbf{L}$, which in essence is a time dependent sequence of PD gains. Denoting the plant's true state by $\mathbf{x}$, at each time step k, the feedback controller calculates the required correction to the control signal as

$$\delta\mathbf{u}_k = \mathbf{L}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k) \tag{3.3}$$

and we then use the final control signal

$$\mathbf{u}_k = \bar{\mathbf{u}}_k + \delta\mathbf{u}_k \tag{3.4}$$

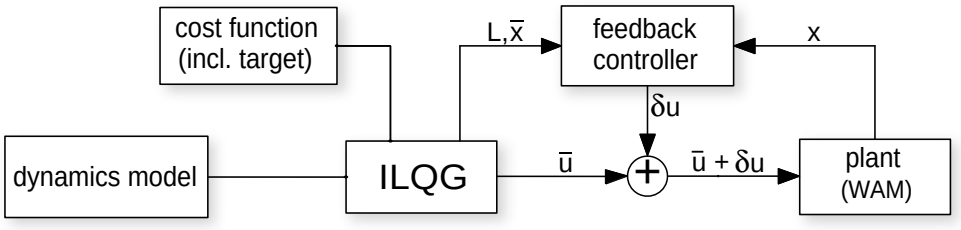


Figure 3.2: The OFC control scheme produced by ILQG used to control the Barrett WAM.

to control the manipulator.

For the control of robotic manipulators ILQG has several desirable properties: (i) ILQG resolves kinematic redundancies automatically, i.e., no explicit inverse kinematics method is required. (ii) It produces a feedforward control sequence with corresponding optimal feedback control law. As we see later this allows us to achieve highly compliant movement plans that are still accurate w.r.t. the task at hand. (iii) We can specify a specific motion task in an "intuitive" high level formulation in the cost function.

At this point one should note that there is no guarantee that ILQG will converge to a global minimum. In fact experience from practice shows that the initial control sequence often affects the final outcome of the algorithm. From a computational perspective the dynamics linearisation steps in the ILQG algorithm loop prove to be the computational bottleneck. This process requires the partial derivatives $\partial\mathbf{f}(\mathbf{x},\mathbf{u})/\partial\mathbf{x}$ and $\partial\mathbf{f}(\mathbf{x},\mathbf{u})/\partial\mathbf{u}$, which are computed, in a generally applicable case[2], using finite differences.

3.2.2 Manipulator dynamics function

We model the non-linear dynamics of our plant using standard equations of motion where the joint torques $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}). \tag{3.5}$$

As before $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the joint angles and joint velocities respectively; $\mathbf{M}(\mathbf{q})$ is the N-dimensional symmetric joint space inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ accounts for Coriolis and centripetal effects, $\mathbf{b}(\dot{\mathbf{q}})$ describes the joint friction, and $\mathbf{g}(\mathbf{q})$ defines the gravity loading depending on the joint angles $\mathbf{q}$ of the manipulator. The kinematic and dynamic

[2] In this work we also follow this approach.

parameters are provided by the robot manufacturer as summarised in Appendix B.

3.2.3 Avoiding discontinuities in the dynamics

The WAM exhibits significant frictional joint torques $\mathbf{b}(\dot{\mathbf{q}})$, which had to be estimated separately. Joint friction is usually modelled to consist of a *static*[3] and *kinetic* Coulomb component as well as of a *viscous* friction component (Fig. 3.3). The Coulomb friction model is discontinuous and therefore has no derivatives defined at $\dot{\mathbf{q}} = 0$, which is problematic because internally ILQG relies on derivatives to improve the control law. Furthermore, very steep gradients (as occurring in step-like functions) can sometimes have a bad impact on the convergence speed of the algorithm. We overcome

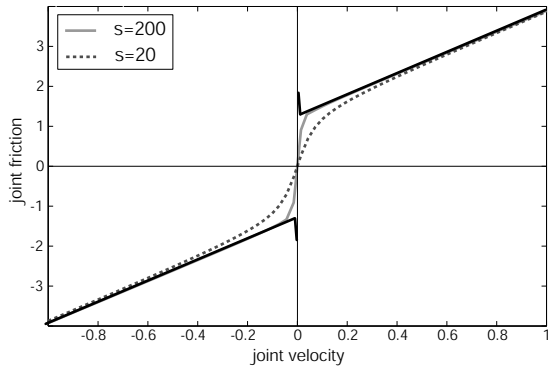


Figure 3.3: Approximative continuous friction model. Solid black line represents the theoretical discontinuos Coulomb friction. For an (example) steepness parameter of $s = 200$ the derivatives at the start condition ($\dot{\mathbf{q}} = 0$) become too large and ILQG diverges whereas for $s = 20$ it successfully converges.

this problem in practice by ignoring the static Coulomb friction and by approximating the kinetic Coulomb and viscous friction in each joint with the following smooth and continuous sigmoid function

$$b(\dot{q}) = b_c \arctan(s\dot{q}) \frac{2}{\pi} + B\dot{q}, \tag{3.6}$$

where s indicates the "steepness" of the fitted arctan function (Fig. 3.3), b_C is the kinetic Coulomb friction, and B is the viscous friction coefficient. We heuristically identified the steepness parameter as $s = 20$ (for all joints) such that it led to overall stable ILQG convergence while providing sufficient modelling accuracy.

We then used the constant angular velocity motion test (Mayeda et al., 1984) as an identification method for the viscous friction coefficient and the kinetic Coulomb

[3] Also called "stiction".

friction. When a small step input torque $\boldsymbol{\tau}^*(i)$ is applied to the target joint i while keeping the other joints fixed, $\dot{\mathbf{q}}_i$ converges to some constant angular velocity as $t \to \infty$ by the effect of the damping torque. By executing the test motions ten times with various values of $\boldsymbol{\tau}^*(i)$ for each joint, B and b_C can be easily estimated by a least-square method. Table 3.1 shows the obtained results for all joints.

Joint	i=1	i=2	i=3	i=4	i=5	i=6	i=7
$\mathbf{B}(i)$	1.142	0.946	0.309	0.255	0.025	0.039	0.004
$\mathbf{b}_c(i)$	2.516	2.581	2.038	0.956	0.323	0.315	0.066

Table 3.1: Estimated joint friction parameters for the WAM.

Since we are commanding joint torques ($\boldsymbol{\tau} = \mathbf{u}$) the deterministic forward dynamics used in ILQG takes the form

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\left(\mathbf{u} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{b}(\dot{\mathbf{q}})\right). \tag{3.7}$$

3.2.4 Incorporating real world constraints into OFC

The model based control of real hardware must obey several physical constraints which need to be incorporated into the OFC framework. These correspond to the physical boundaries of the manipulator, namely the maximally applicable motor torques ($\mathbf{u}_{min}, \mathbf{u}_{max}$), the joint angle limits ($\mathbf{q}_{min}, \mathbf{q}_{max}$), and the maximally executable joint velocities ($\dot{\mathbf{q}}_{min}, \dot{\mathbf{q}}_{max}$). ILQG handles the control constraints during optimisation by enforcing control boundaries on $\bar{\mathbf{u}}$ and by modifying the feedback gain matrix $\mathbf{L}_k$ (i.e., setting $\mathbf{L}_k$ to zero) whenever an element of the control sequence lies on the constraint boundary (see Appendix A.2, code line 19). Applied to the hardware however we found that control constraints are rarely violated whereas state constraints are much more critical (Fig. 3.4) and ILQG does not handle those constrains in any form. We therefore propose to incorporate the joint angle and joint velocity boundaries as opti-

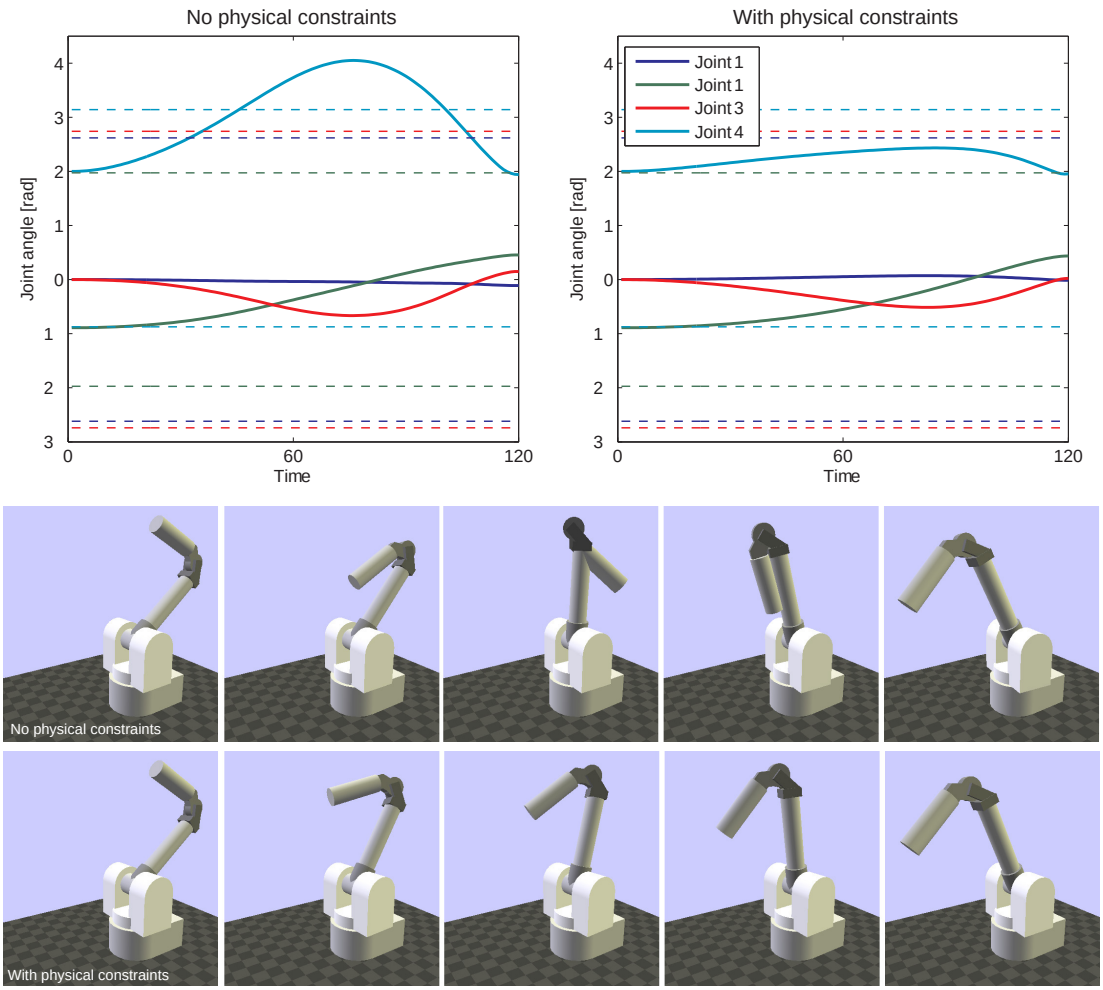


Figure 3.4: Comparison of ILQG results obtained *without* and *with* physical constraint terms $P(\mathbf{q})$ and $V(\dot{\mathbf{q}})$. The unconstrained solution violates the physical limits which would lead to a self collision applied to the WAM (top row of simulation screenshots).

misation constraints into the running cost in (3.2) as

$$\begin{aligned} v &= w_p \,|\, \mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + \\ &\quad \Delta t \sum_{k=0}^{K} \left(w_e \,|\, \mathbf{u}_k \,|^2 + P(\mathbf{q}_k) + V(\dot{\mathbf{q}}_k) \right) \end{aligned} \tag{3.8}$$

$$P(\mathbf{q}) = w_{pb} \sum_{i=1}^{4} \left([q_i - q_i^{max}]_+^2 + [q_i^{min} - q_i]_+^2 \right) \tag{3.9}$$

$$V(\dot{\mathbf{q}}) = w_{vb} \sum_{i=1}^{4} \left([\dot{q}_i - \dot{q}_i^{max}]_+^2 + [\dot{q}_i^{min} - \dot{q}_i]_+^2 \right). \tag{3.10}$$

For the joint angle boundaries (w_{pb}), and the joint velocity boundaries (w_{vb}) we use following notational convention $[x]_+ = \max(0, x)$ given that for each joint ($q_i^{min} < 0 < q_i^{max}$) and ($\dot{q}_i^{min} < 0 < \dot{q}_i^{max}$).

Another issue that needs to be addressed is the correct initialisation of the robot's joint torque state. Before starting the optimal reaching movement the robot is assumed to be in a stationary state, which is achieved by applying the torque sequence $\mathbf{u}_{init}$. For the WAM the gravity compensation is known and we therefore set $\mathbf{u}_{init} = \mathbf{g}(\mathbf{q}_0)$. At reaching start time $k = 0$ the torque sequences of the plant and the ILQG result should be equal, i.e., $\mathbf{g}(\mathbf{q}_0) = \bar{\mathbf{u}}_0$. However there is no way to "tell" ILQG what the initial torques should be at time $k = 0$ and therefore a "torque jump" will be commanded to the plant. A similar effect arises at the end of the motion ($k = K$) where we usually transfer from reaching back to gravity compensation and it would be desirable that $\mathbf{g}(\mathbf{q}_K) = \bar{\mathbf{u}}_K$. Such torque transition errors must be avoided since they destabilise the plant and produce high stresses on the actuators, which contradicts the energy efficient control law that we want to achieve. Therefore an additional constraint is required in order to avoid excessively large motor jumps at the beginning and end of the reaching. In theory one can partially avoid that problem by modelling the underlying motor dynamics that define the maximal change in motor control signals. However this approach blows up the state space by at least N additional states, which in consequence increases the computational load on ILQG. We solve this problem alternatively by (i) using the gravity compensation torques $\mathbf{g}(\mathbf{q_0})$ as initial torque sequence and by (ii) introducing an additional "soft-start" and "soft-stop" constraint into the cost function. We extend our cost function v by another time-dependent term in the running cost

$$
\begin{aligned}
v^* &= v + \Delta t \sum_{k=0}^{K} w_v^* \,|\, \dot{\mathbf{q}}_k \,|^2 \qquad (3.11)\\
w_v^* &= \begin{cases} 1 - \frac{k}{K_s} & , \textbf{if} \quad k < K_s \\ 1 - \frac{K-k}{K_s} & , \textbf{if} \quad k > (K - K_s) \\ 0 & , \textbf{otherwise} \end{cases} .
\end{aligned}
$$

Therefore the reaching has now became closer to a hold-reach-hold task, where K_s determines the transition smoothness, which is formulated in terms of joint angle velocities, irrespective of the current arm position at start and end.

3.3 Results

In this section we discuss the results from controlling the WAM using the proposed (local) optimal feedback controller. We study two setups: First we use the 4 DoF setup to show the basic concepts and compare the results to other trajectory planners in terms

of task achievement, compliance and energy consumption. The second setup contains all 7 DoF and here we will highlight the scalability of our approach and we present a set of control applications.

3.3.1 OFC with 4 DoF

We study movements for the fixed motion duration of 1.6 seconds, which we discretise into $K = 800$ steps (Δt = 0.002 sec) corresponding to the robot's operation rate of $500Hz$. The manipulator starts at an initial position $\mathbf{q}_0$ and reaches towards several targets $\mathbf{t}$, defined in end effector task space coordinates (x,y,z) in meters. During movement we wish to minimise the proposed cost function (3.11).

We set the arm's start position as $\mathbf{q}_0 = (0,-0.89,0,2.0)^T$ *rad* and 3 reference targets (left, middle, right): $\mathbf{t}_{left} = (0.55,0.3,0.25)$, $\mathbf{t}_{center} = (0.55,0,0.25)$, $\mathbf{t}_{right} = (0.55,-0.3,0.25)$. These targets represent a reasonably far distance (center 0.73m; left/right 0.79m) for a reaching time of 1.6 sec. We used following cost parameters: $wp = 50000$, $w_v = 5000$, $w_e = 1$, $w_{pb} = 50$, $w_{vb} = 100$, $K_s = 10$. Next we found the optimal solutions using ILQG for the three targets. Using the gravity compensation as initial trajectory, the algorithm converged after following number of iterations: $iter_{center} = 63$, $iter_{left} = 89$,$iter_{right} = 68$. The ILQG results are applied to the WAM following the control law defined in (3.4). The matrix $\mathbf{L}_k = [\mathbf{L}_P\ \mathbf{L}_D]_k$ is the 4×8 time-dependent locally valid control law that contains the optimal PD gains for each time step k. These gains follow the minimum intervention principle: As can be seen in the center panel of Fig. 3.5, on the example of $\mathbf{t}_{center}$, the L gains take into account the nature of the specified task in the cost function. Therefore the gains are very low up to about 500 time steps and then grow towards the end of the motion where task accuracy and stability are more important[4]. This trade-off between energy preservation and reaching task achievement is present in all joints. Notably the L-matrix is diagonally dominant with an additional coupling between joints 1 and joint 3. This coupling appears due to the redundancy those joint have for the task of reaching to the center targets, e.g., perturbations in joint 1 can partly be corrected by joint 3 and vice versa. A comparison between the desired optimal trajectories (dashed lines in Fig. 3.5) and the feedback corrected trajectories on the plant (solid lines) indicate that the modelled forward dynamics is fairly accurate, especially for joint 2 that exhibits the largest joint torques. We can observe effects of "unmodelled dynamics", which can be attributed to

[4]Generally the L-gains are significantly smaller than the WAM factory-default PD gains which are $P = 2000$ and $D = 500$ for each joint.

the errors in the friction estimation (see section 3.2.3).

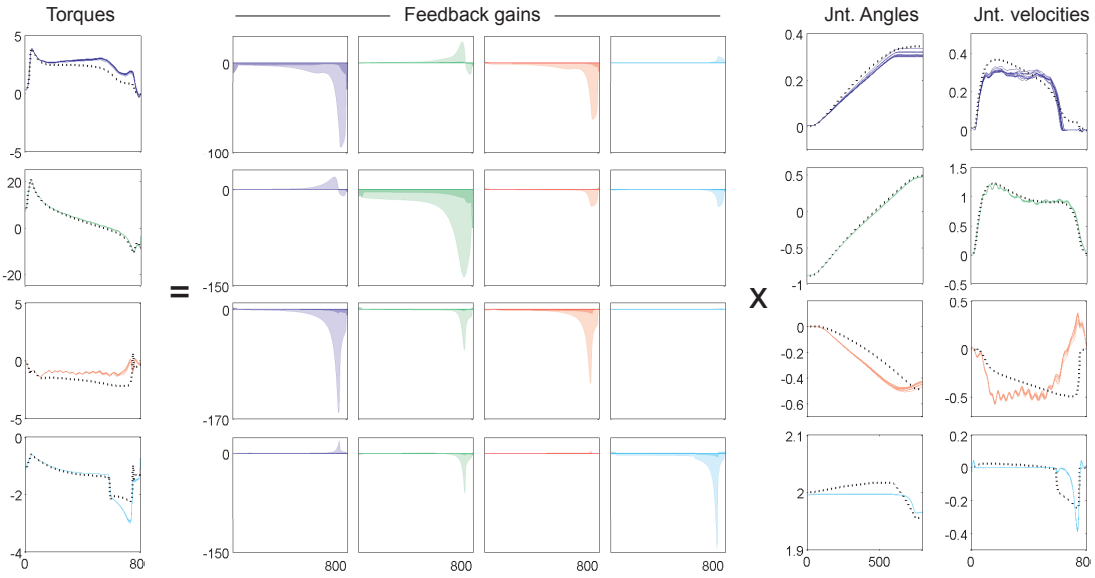


Figure 3.5: Results of the ILQG control law (3.4) applied to the WAM with 4 DoF for a reaching to the center target $\mathbf{t}_{center}$. The dashed lines represent the optimal desired trajectories produced by ILQG and the solid lines show the 10 trajectories recorded from the WAM. The shaded areas depict the feedback gain matrix (L), where brighter shadings indicate the position gains (P) and the darker areas depict the velocity-gains (D).

As mentioned in the introduction in our OFC control strategy the primary aim is task achievement in target reaching and the energy preservation during control, whereas the exact trajectory tracking here is no performance criterion. As we show next these properties allow us (i) to use less energy than other (open loop) optimal control algorithms, and (ii) to be compliant during motion.

We compare the ILQG results against an open loop optimiser following the minimum jerk optimisation criterion (Flash and Hogan, 1985). We use the same start and end position as before and the optimisation gives us a optimal kinematic trajectory $\mathbf{x}^*$, which must be tracked using a (typically) hand tuned PD feedback control law

$$\mathbf{u}_k^{plant} = \mathbf{P} \cdot (\mathbf{q}_k^* - \mathbf{q}_k) + \mathbf{D} \cdot (\dot{\mathbf{q}}_k^* - \dot{\mathbf{q}}_k)\,. \tag{3.12}$$

We used two sets of PD gains: (a) The high gain factory default values of the Barrett WAM controller: $P = 2000$, $D = 500$. (b) The maximal diagonal values of the ILQG feedback gain matrix $\mathbf{L}$: $\mathbf{P} = max(diag(\mathbf{L}_P))$, $\mathbf{D} = max(diag(\mathbf{L}_D))$.

For a better comparison with the ILQG control paradigm we also ran the minimum jerk results with a feed-forward torque sequence that we computed using the inverse

dynamics

$$\mathbf{u}_k^{plant} = \boldsymbol{\tau}_k(\mathbf{q}_k^*, \dot{\mathbf{q}}_k^*, \ddot{\mathbf{q}}_k^*) + \mathbf{P} \cdot (\mathbf{q}_k^* - \mathbf{q}_k) + \mathbf{D} \cdot (\dot{\mathbf{q}}_k^* - \dot{\mathbf{q}}_k)\,. \tag{3.13}$$

As before we used: (c) the standard WAM gains and (d) $max(\mathbf{L})$. Fig. 3.6 summarises the results.

As expected, in terms of accuracy the high gain feedforward minimum jerk trial (c) is the most accurate. However it achieves this performance with the price of a fairly high energy consumption, i.e., 25% higher than ILQG. Due to the high corrective gains its compliance is reduced making the robot much more destructive in the case of unexpected collisions. In summary ILQG offers the best trade-off between end-point accuracy and energy consumption. The ILQG results for all targets are accumulated in Table 3.2. The reaching produced by ILQG is very compliant allowing an interaction at all times with the robot (Fig. 3.7 and accompanying video). The compliant behaviour is a result of the feedback gains L that are very low during the whole motion (i.e., compliant) and only ramp up by a small amount towards the end of the motion near the target. Applied to the balloon experiment (happy face), this leads to the arm bouncing off the "springy" balloon. After the reaching ($1.6sec$) the arm is stopped. There is no collision detection in the control loop.

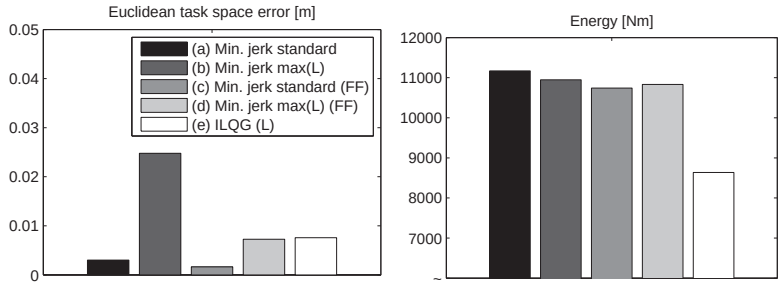


Figure 3.6: Comparison of min jerk (conditions (1)-(4) and ILQG (5) - Results are averaged over 10 reaches to the center target ($\mathbf{t}_{center}$). Left: Euclidean distance (error) between target and end-point. Right: "Energy" consumption computed as sum of all torques in all joints over the entire trajectory: $\sum_{i=1}^{4}\sum_{k=1}^{K}|\mathbf{u}_k^{plant}(i)|$.

3.3.2 Scaling to 7 DoF

In this section we demonstrate the scaling and redundancy resolution abilities of ILQG. We demonstrate results on two types of reaching tasks:

ILQG	Euclidean target error $[mm]$	Energy $[N \cdot m]$
$\mathbf{t}_{center}$	7.576 ± 0.148	$8.637 \cdot 10^3 \pm 0.015 \cdot 10^3$
$\mathbf{t}_{left}$	9.707 ± 0.233	$9.746 \cdot 10^3 \pm 0.020 \cdot 10^3$
$\mathbf{t}_{right}$	9.213 ± 0.273	$8.994 \cdot 10^3 \pm 0.031 \cdot 10^3$

Table 3.2: ILQG results (mean $\pm$ std over 10 trials) onf 4 DoF WAM for 3 reference targets.

Example: Task space reaching without orientation

We repeated the reaching experiment from the section with the 7 DoF setup. Initial position, targets, reaching time and cost function parameters were the same as before. Unlike before we now have 4 redundant degrees of freedom, since we only look at (x,y,z) coordinates and ignore the end-effector orientation. ILQG successfully converges after $iter_{center} = 74$, $iter_{left} = 80$, $iter_{right} = 71$ iterations and resolves the kinematic redundancies. As shown in Table 3.3 the reaching performance is comparable to the 4 DoF case. Notably the 7 DoF setup has a larger energy consumption as the 4 DoF arm. We attribute this to the higher weight of the additional motors and gearing ($+2kg$) that are the added at the last 3 joints.

ILQG	Euclidean target error $[mm]$	Energy $[N \cdot m]$
$\mathbf{t}_{center}$	5.494 ± 0.150	$16.254 \cdot 10^3 \pm 0.023 \cdot 10^3$
$\mathbf{t}_{left}$	6.891 ± 0.175	$17.690 \cdot 10^3 \pm 0.017 \cdot 10^3$
$\mathbf{t}_{right}$	9.210 ± 0.156	$16.272 \cdot 10^3 \pm 0.021 \cdot 10^3$

Table 3.3: ILQG results (mean $\pm$ std over 10 trials) on 7 DoF WAM for 3 reference targets.

Example: Task space reaching with orientation

Many manipulator tasks, for example pick and place tasks, require a specification of the end-effector orientation. Therefore instead of defining the reaching target in task space coordinates (x,y,z) only (3.11), we additionally specify the desired end-effector rotation as *yaw*, *pitch*, *roll* (y,p,r). We set two reaching tasks towards $\mathbf{t}_{center}$ with different end-effector orientations, one pointing horizontally to the front and the other

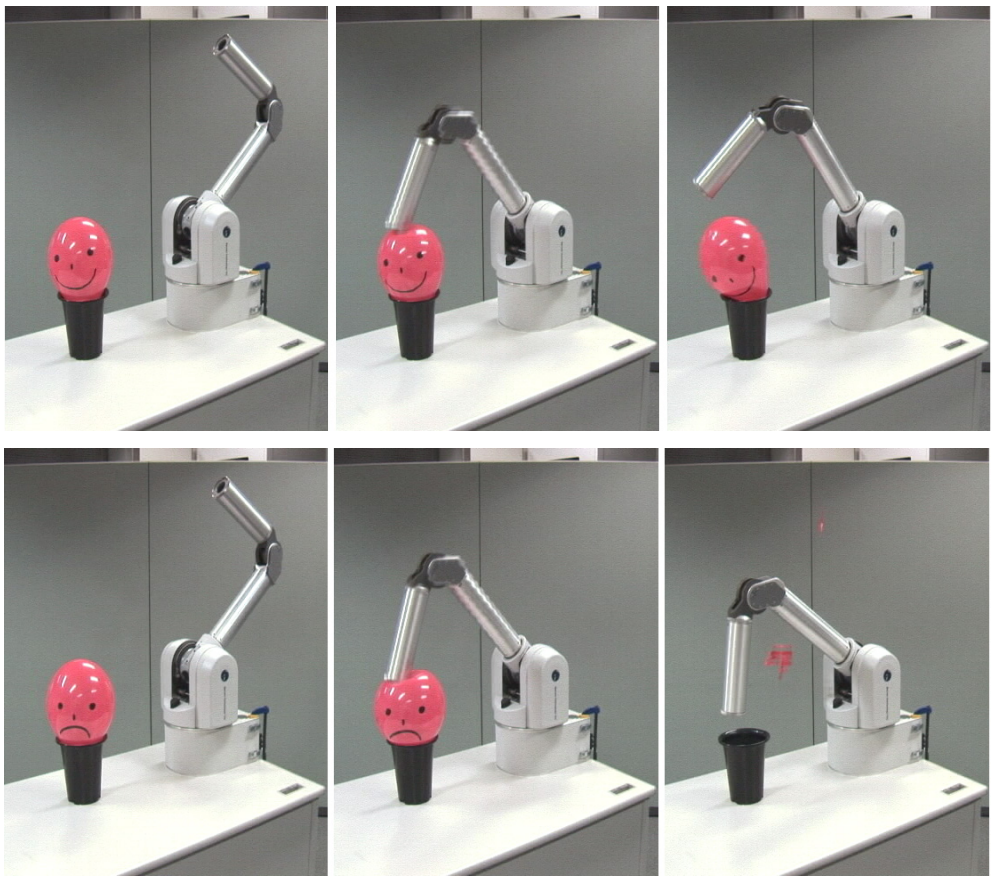


Figure 3.7: Demonstration of the compliance of ILQG motion. Top row: The reaching with ILQG towards the center of the black bucket. The obstacle (balloon) cannot get damaged by the robot, due to the compliant motion of ILQG. In contrast the minimum jerk planning using standard gains is not compliant and therefore highly destructive.

pointing vertically down. The 4×4 transformation matrices of the targets are

$$\mathbf{t}_{front} = \begin{pmatrix} 0 & 0 & 1.0 & 0.45 \\ 0 & 1.0 & 0 & 0 \\ -1.0 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{t}_{down} = \begin{pmatrix} -1.0 & 0 & 0 & 0.45 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & -1.0 & 0.25 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Applied to the WAM (Fig. 3.8), it successfully reached the targets with high accuracy in position and orientation as can be seen from the WAM's end-effector transformation

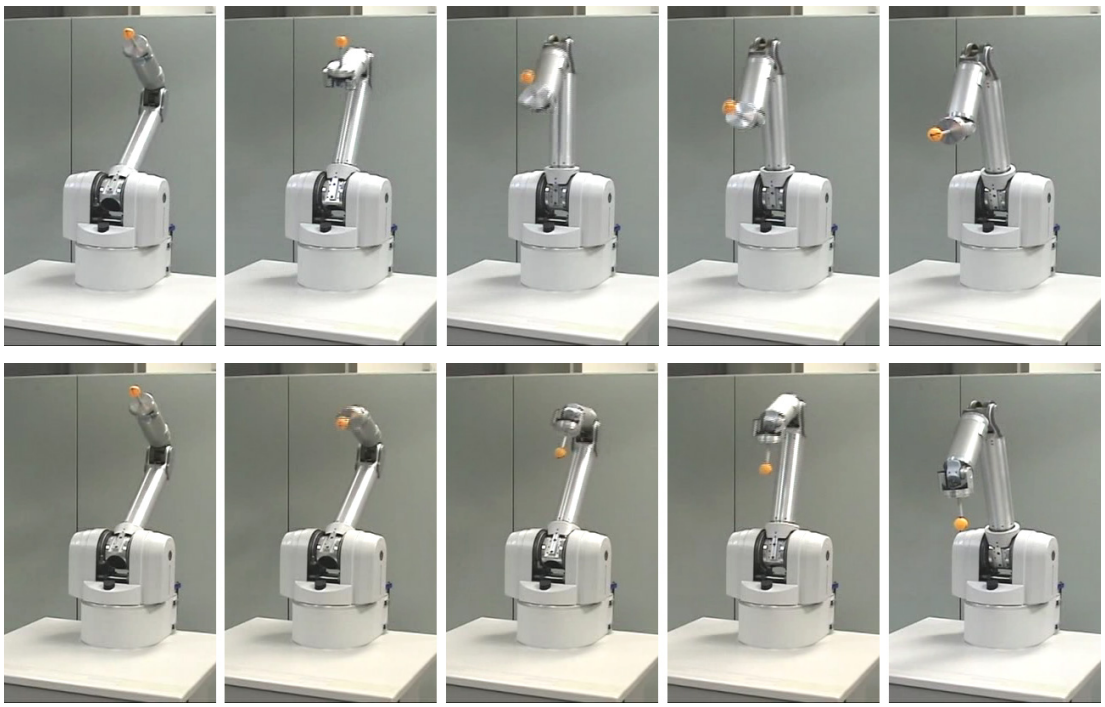


Figure 3.8: ILQG reaching of 7 DoF WAM for targets with two different end point orientations. Top row: $\mathbf{t}_{front}$; bottom row: $\mathbf{t}_{down}$

matrix of the two targets (averaged over 10 trials).

$$\mathbf{r}(\mathbf{q}_K^{front}) = \begin{pmatrix} 0.0032 & 0.0020 & 1.0000 & 0.4585 \\ -0.0005 & 1.0000 & -0.0020 & -0.0003 \\ -1.0000 & -0.0005 & 0.0032 & 0.2631 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

$$\mathbf{r}(\mathbf{q}_K^{down}) = \begin{pmatrix} -1.0000 & -0.0052 & 0.0017 & 0.4448 \\ -0.0052 & 1.0000 & -0.0028 & -0.0018 \\ -0.0017 & -0.0028 & -1.0000 & 0.2502 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Table 3.4 summarises the performance over 10 trials.

3.3.3 Reducing the computational costs of ILQG

For any motion control algorithm, real-time planning is desireable and computational costs therefore play an important role. Given the high operation rate of the WAM (500 Hz), we face serious limitations due to the computational efficiency of iterative methods. These scale linearly with the number of time steps, linearly with the number

ILQG	$\mathbf{t}_{front}$
Euclidean target error $[mm]$	14.672 ± 0.501
Yaw error $[rad]$	0.0024 ± 0.0011
Pitch error $[rad]$	0.0014 ± 0.0011
Roll error $[rad]$	0.0013 ± 0.0005
Energy $[N \cdot m]$	$12.485 \cdot 10^3 \pm 0.0431 \cdot 10^3$
ILQG	$\mathbf{t}_{down}$
Euclidean target error $[mm]$	5.538 ± 0.329
Yaw error $[rad]$	0.0035 ± 0.0027
Pitch error $[rad]$	0.0019 ± 0.0012
Roll error $[rad]$	0.0064 ± 0.0053
Energy $[N \cdot m]$	$13.271 \cdot 10^3 \pm 0.677 \cdot 10^3$

Table 3.4: ILQG results (mean $\pm$ std over 10 trials) on 7 DoF WAM for 2 reference targets with orientation constraints .

of iterations and in the number of input dimensions $\mathbf{x} = (\mathbf{q}; \dot{\mathbf{q}})$ and $\mathbf{u}$ (i.e., $3N$ for an N DoF robot).

Typical ILQG simulations produce accurate optimisation results with $dt = 0.01$. Therefore when calculating the ILQG control law we do this initially with $dt = 0.01$ to quickly obtain an optimal control sequence $\bar{\mathbf{u}}$ of length $n = 160$. We then subsample this optimal torque sequence to get longer control sequence $\bar{\mathbf{u}}_{ext}$ of length $n = 800$. This sequence serves as the new initial control sequence of ILQG with $dt = 0.005$ and since the sequence is located near the optimal solution already, ILQG converges after only 2 to 4 iterations on average.

In order to reduce the finite differences calculations one can employ analytic derivatives of the dynamics function. Another practical speed-up approach limits the required number of iterations by remembering previously calculated optimal trajectories, which then can be used as an "initialisation library" near an expected optimum, performed for example with a nearest neighbour search. A similar approach has proved to create a optimal behaviour in a real-time simulated swimmer (Tassa et al., 2007). Applying the above mentioned speed-up methods we were able to perform complete ILQG computations for the 7 DoF WAM in the range of 2 to 5 seconds on a regular Notebook (*Intel Core 2* $1.8GHz$). Notably these solutions were obtained using finite differences

instead of analytic derivatives.

3.4 Discussion

In this chapter we proposed to use OFC for the control of an anthropomorphic manipulator. We developed a locally valid optimal feedback controller for the WAM, which we achieved by incorporating the physical constraints of the robot into the performance index of the optimal controller. We further elaborated on the problems and solutions of discontinuous dynamics as they occur on real hardware. We successfully tested our control method on the manipulator and demonstrated the practical benefits of this control strategy, that unifies motion planning, redundancy resolution and compliant control as a result of a single algorithm.

Based on our results one could argue that the energy comparison with minimum jerk is not really "fair" since the latter is not designed to be safe. We chose minimum jerk, which is a well-known open loop optimiser, as a baseline comparison in terms of accuracy and energy consumption. By doing so we aimed to highlight some of the benefits of the OFC scheme w.r.t. traditional open loop optimisers. Clearly there are other viable routes to generate, for example a compliant minimum jerk reaching by hand-tuning the PD gains or using an accurate feedforward mechanism. With kinematic planners the physical limitations such as maximum allowed torques or velocities are not accounted for (per se), whereas in OFC, using our new cost function, this is all handled in one go. Notably the applicability of this OFC control law on a real robotic plant has not been shown before and these novel experimental results are meant to provide an alternate, more systematic approach to tuning the feedback gains *selectively* in task dependent dimensions - a choice that may not be evident for complex tasks. Furthermore we believe that the results are not only interesting for the application of ILQG. The proposed extensions to the ILQG scheme are applicable to other dynamics model based optimal controllers, e.g., *model predictive control (MPC)*(Garcia et al., 1989) or RL schemes (Sutton and Barto, 1998). Indeed an implementation if ILQG in a model predictive fashion could serve as a good alternative to our current implementation, which compute the *entire* trajectory plus *optimal feedback controller* a priory. However in its current form it seems unlikely that we could recompute ILQG solutions at a rate of $500Hz$ or similar high frequencies.

As for any model-based control method, the dynamics model is the bottleneck in terms of task achievement and accuracy. Even though we sacrificed model accuracy

to achieve numerical stability within ILQG, we still were able to achieve reasonable reaching accuracy that can be thought to be sufficient for most daily life tasks. However the problem of discontinuous dynamics remains a fundamental limitation of ILQG and other local optimisation techniques. In the case of point-to-point reaching movements this problem is especially severe as the dynamics are discontinuous typically at the start and the end of the motion where velocities are zero. Indeed the modelling and simulation of rigid body dynamics under friction and contacts is a broad and active area of research (Pfeiffer and Glocker, 1996; Stewart, 2000) and there have been attempts to formulate optimal control problems of discontinuous dynamics (Stewart and Anitescu, 2010). Very recently Tassa and Todorov (2010) presented a (preliminary) method of smoothing discontinuous dynamics that involve contacts and friction. Their approach is based on the fundamental assumption that discontinuities can be modelled as noise and that stochastic dynamics are inherently smooth (on average) and therefore differentiability issues disappear. The authors further present initial simulation results with DDP on a task involving collisions using the smoothed dynamics, which suggests this could be a viable route for the future.

A complete treatment of a control methodology should generally involve stability analysis. Because of the difficulties of stability analysis, in particular for control involving nonlinear dynamics, here we can only *qualitatively* observe that ILQG leads to stable behaviour on the WAM. This is even in the presence of manual perturbations (see accompanying video). Indeed we are not aware of any work discussing stability within the ILQG framework. In terms of *stability of the convergence* of ILQG the outcome of the iterations depend on the initial conditions as well as the cost function parameters. As a good practice we propose to use gravity compensation as initial torques.

In the current work we have restricted our attention to the study of reaching tasks. However OFC is easily extendable to more complex tasks such as throwing and hitting (see accompanying video), via point tasks, or walking (Morimoto and Atkeson, 2003). In Chapter 5 we will discuss extensions of OFC for the control of a redundantly actuated (e.g., variable impedance) manipulators under the assumption of stochastic plant dynamics.

Chapter 4

Optimal feedback control with learned dynamics

In the previous chapters we have shown that OFC is an attractive control strategy for anthropomorphic manipulators. We have highlighted the redundancy resolution capabilities of this framework, which lead to compliant and energy efficient movement plans on an anthropomorphic robot. The discussions so far assumed that the dynamics of the controlled system is given as an analytic rigid body dynamics model and that this model does not change over time, i.e., it is *stationary*. Next we wish to study optimal feedback control scenarios in which the dynamics may be unknown or subject to systematic changes.

4.1 Introduction

A characteristic property of modern anthropomorphic systems, besides their large redundancies, is a lightweight and flexible-joint construction which is a key ingredient for achieving compliant human-like motion (Pratt et al., 1995; Hirzinger et al., 2001; Zinn et al., 2004). Well known examples of such "soft robots" are based on *pneumatic artificial muscles (PAM)* (Daerden, 1999), which are inherently flexible due to the compressibility of air. Other common flexible robot designs are based on *series elastic actuators (SEA)*[1] (e.g., Vanderborght et al. (2009)), which introduce flexibility into the actuation mechanism by linking the (stiff) DC motor to the driven joint via adjustable elastic elements, (i.e., springs). Motivated by the structure of mammal muscles PAMs and SEAs are often setup in antagonistic architectures, and therefore

[1] We will discuss SEA and its control in detail in Chapter 5.

introduce additional redundancies.

Often when flexibility or nonlinearities are introduced to a system's morphology the identification of accurate dynamics becomes more difficult. Rigid body assumptions usually can serve only as a crude approximation of real systems and the exact inertial parameters may be unknown. Furthermore, even if we were able to identify the dynamics accurately, a rigid model is not capable to account for *unforeseen* changes in the plant dynamics that could occur for example due to a change in environmental conditions, wear and tear or morphological changes. Incorporating such changes into the control framework is desirable. However to achieve this in a generally valid fashion, i.e., without prior knowledge of the source or shape of the change, is nontrivial when analytic dynamics are used.

In order to overcome the limitations of (rigid) analytical dynamics models we can employ online supervised learning methods to extract dynamics models driven by data from the movement system itself. During operation robots produce a vast amount of movement data such as joint angles ($\mathbf{q}$), velocities ($\dot{\mathbf{q}}$), accelerations ($\ddot{\mathbf{q}}$) and applied motor commands ($\mathbf{u}$). *Learning of dynamics* in essence corresponds to the problem of high-dimensional function approximation or regression, this is given an input $\mathbf{x} \in \mathbb{R}^n$ and a target $\mathbf{y} \in \mathbb{R}^m$ we learn the mapping that describes the relationship from input to target from samples. This can be achieved with a multitude of supervised learning algorithms as long as they can approximate *nonlinear* functions and as long as learning can be performed incrementally, i.e., in an online fashion. Learning the dynamics online enables the controller to adapt *on the fly* to changes in dynamics conditions without having to store all incoming data explicitly. Please note that in the engineering literature learning the dynamics is often referred to as *system identification*. We will use the term *learning* to conform with the nomenclature of modern statistical machine learning. Learning can often be achieved by fitting for example rigid body dynamics parameters to data that has been collected from the robot. Here however we are interested in *non-parametric* learning methods, as we do not assume a specific (parametric) form of the dynamics function a priory.

The concept of dynamics learning has been studied extensively in robot control (Vijayakumar et al., 2002; Conradt et al., 2000; D'Souza et al., 2001; Nguyen-Tuong et al., 2008a). A common approach hereby is to learn a *direct inverse dynamics* mapping $\mathbf{u} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ in order to improve trajectory tracking performance by predicting required motor command for a desired target state (An et al., 1988). A more task oriented approach is *feedback error learning* (Kawato, 1987), which uses the results of compos-

ite controller predictions and actual sensory readings to build an error-correcting model of the dynamics[2]. In redundantly actuated manipulators (e.g., antagonistic SEA) the inverse dynamics mapping may not be unique, i.e., there are multiple muscle commands that produce the same joint torque. *Distal supervised learning* (Jordan and Rumelhart, 1992) can account for that problem by first learning a forward dynamics model, which is not ill-defined. The output of that forward model $\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ then can be used as training input together with the desired states $\mathbf{q}_d$ and $\dot{\mathbf{q}}_d$ to an inverse model, i.e., $\mathbf{u} = \mathbf{g}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}})$. The output of the inverse model $\mathbf{u}$ is then used again to train the forward model using sensory readings of the resulting states $\mathbf{q}$ and $\dot{\mathbf{q}}$. With this composite learning loop one can, after the learning has converged, resolve between inverse dynamics solutions.

The mentioned dynamics learning examples are typically based on some form of desired trajectory that needs to be tracked accurately. For optimal control learned inverse dynamics models could be used to improve performance in open loop optimal controllers (see Chapter 3). However in OFC both planning and control are achieved through the optimisation process and therefore inverse dynamics models play no direct role. Instead forward dynamics models are used to find the optimal trajectory and control law. Some work has been proposed on learning of non-linear forward dynamics models in the context of adaptive control theory (Nakanishi et al., 2005; Choi and Farrell, 2000). On the whole forward dynamics have found more attention in the study of biological motor control[3] and the theory of internal models in the *central nervous system (CNS)* (Miall and Wolpert, 1996; Kawato, 1999).

The classic OFC framework is formulated using analytic dynamics models and by combining OFC with dynamics learning we can create a powerful and principled control strategy for the biomorphic based highly redundant actuation systems that are currently being developed. From a biological point of view, enabling OFC to be adaptive would allow us to investigate the role of optimal control in human adaptation scenarios. Indeed adaptation, for example towards external perturbations, is a key property of human motion and is a very active area of research since more than two decades (Shadmehr and Mussa-Ivaldi, 1994; Shadmehr and Wise, 2005).

The remainder of this chapter is structured as follows: In the next section we will elaborate upon the proposed adaptive OFC scheme and the involved online learning

[2]Feedback error learning was originally motivated from a biological perspective in order to establish a computational motor learning model for internal models in the central nervous system (CNS).

[3]Alongside inverse dynamics models.

mechanism, i.e., we combine the ILQG framework with a learning of the forward dynamics. Section 4.3 contains an in depth experimental evaluation of our framework in simulation: we compare the optimal solutions obtained with classic ILQG using *analytic dynamics* and those obtained using ILQG with a *learned dynamics* model. We show that using learned dynamics in ILQG does not suffer from significant losses in accuracy, energy efficiency or ILQG convergence behaviour. We further highlight the online adaptation capabilities of our method under a variety of systematic external perturbations. All evaluations are performed on (i) high DoF joint torque controlled and (ii) on antagonistically actuated robots in simulation. In Section 4.4 we place our adaptive OFC framework in perspective to other (traditional) adaptive control methods and we conclude this chapter with a discussion section.

4.2 Adaptive optimal feedback control

As mentioned earlier a major shortcoming of ILQG (and other OFC methods) is the dependence on an analytic form of the system dynamics, which often may be unknown or subject to change. We overcome this limitation by learning an adaptive internal model of the system dynamics using an online, (non-parametric) supervised learning method. We consequently use the learned model to derive an ILQG formulation that is computationally efficient, reacts optimally to transient perturbations, and most notably adapts to systematic changes in plant dynamics. We name this algorithm *ILQG with learned dynamics (ILQG–LD)* and *OFC-LD* in the more general case.

The idea of learning the system dynamics in combination with iterative optimisations of trajectory or policy has been explored previously in the literature, e.g., for learning to swing up a pendulum (Atkeson and Schaal, 1997) using some prior knowledge about the form of the dynamics (i.e., parametric dynamics model). Similarly, (Abbeel et al., 2006) proposed a hybrid reinforcement learning algorithm, where a policy and an internal model get subsequently updated from “real life” trials. In contrast to their method, we employ a second-order optimisation method, and we refine the control law solely from the internal model. So in our case learning is restricted to acquiring and changing the dynamics and no learning is involved in planning and control. To our knowledge, learning dynamics in conjunction with control optimisation has not been studied in the light of adaptability to changing plant dynamics.

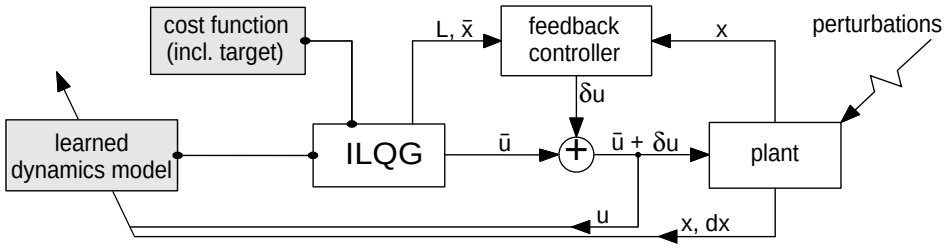


Figure 4.1: Illustration of our ILQG–LD learning and control scheme.

4.2.1 ILQG with Learned Dynamics (ILQG–LD)

In order to eliminate the need for an analytic dynamics model and to make ILQG adaptive, we wish to learn an approximation $\tilde{\mathbf{f}}$ of the real plant forward dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. As before the state consists of joint angles and velocities, $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]^T$. Assuming our model $\tilde{\mathbf{f}}$ has been coarsely pre-trained, for example by motor babbling, we can refine that model in an online fashion as shown in Fig. 4.1. For optimising and carrying out a movement, we have to define a cost function (where also the desired final state is encoded), the start state, and the number of discrete time steps because the ILQG algorithm in its current form requires a specified final time. Given an initial torque sequence $\bar{\mathbf{u}}_k^0$, the ILQG iterations can be carried out as described in Section 2.3.1 of Chapter 2, but now utilising the learned model $\tilde{\mathbf{f}}$. This yields a locally optimal control sequence $\bar{\mathbf{u}}_k$, a corresponding desired state sequence $\bar{\mathbf{x}}_k$, and feedback correction gain matrices $\mathbf{L}_k$. Denoting the plant's true state by $\mathbf{x}$, at each time step k, the feedback controller calculates the required correction to the control signal as $\delta\mathbf{u}_k = \mathbf{L}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k)$. We then use the final control signal $\mathbf{u}_k = \bar{\mathbf{u}}_k + \delta\mathbf{u}_k$, the plant's state $\mathbf{x}_k$ and its change $d\mathbf{x}_k$ to update our internal forward model $\tilde{\mathbf{f}}$. As we show in Section 4.3, we can thus account for (systematic) perturbations and also bootstrap a dynamics model from scratch.

Please note that for the proposed ILQG–LD architecture the updating of the dynamics model is taking place on a trial-by-trial basis. This computation of an ILQG solution is strictly speaking an offline process and decoupled from the dynamics update. The logical order is to (i) compute ILQG, (ii) run it on the plant, (iii) use the plant data to update $\tilde{\mathbf{f}}$. For the next trial we go to step (i) and compute ILQG and so forth. Apparently one could implement an ILQG-LD solution that is "truly online". This could be achieved for example using a MPC style controller, where after each applied motor command the dynamics model is updated and the optimal solution is recomputed for the remaining time horizon. We believe that keeping this decoupled for now is more sensible in order to demonstrate learning and adaptation effects. Fur-

thermore, since the focus of this chapter is on utilising dynamics learning within ILQG and its implications to adaptivity, we do not utilise an explicit noise model $\mathbf{F}$ for the sake of clarity of results. We also do not include any model for estimating the state, that is, we assume that noise-free measurements of the system are available (full observability). However an ILQG implementation for systems with partial observability has been been developed recently (Li and Todorov, 2007).

4.2.2 Learning the forward dynamics

In general various machine learning algorithms could be applied to learn the dynamics approximation $\tilde{\mathbf{f}}$. If we assume prior knowledge about the dynamics structure one can use parametric models. For example, if we assume rigid body dynamics of a manipulator one can estimate the inertial parameters by collecting training data and fitting the model parameters using standard regression techniques (An et al., 1988). If we consider highly nonlinear potentially unknown dynamics structures (as we do here) non-parametric approaches are more favourable, as discussed next.

A major distinction can be made between *global* and *local* learning approaches. An example of a global learning method is a neural network with a sigmoid activation function. Such neural networks are characterised by activation functions that respond to inputs from the whole input space, which generally leads to the problem of negative interference (Schaal, 2002). If a neural network is trained using data from one specific region in the high-dimensional input space, its future predictions will be accurate in these regions. However, if the system is then trained with new data from another region, the input distribution changes and the parameters are adjusted. This may result in the loss of the previously learned regions. This problem could be solved by storing all the produced data, and always retraining the network using the complete data set. In many cases[4] it is undesirable to store the whole incoming data streams produced by high-dimensional robotic systems and furthermore the re-training of the network may be computationally not feasible for real time applications. Another issue with global learning arises with the adjustment of meta-parameters such as the number of neurons or hidden layers, which cannot be adapted during learning. This makes global learning inflexible with respect to change in the dynamics of the system.

Local learning methods, in contrast, represent a function by using small simplistic patches - e.g. first order polynomials. The range of these local patches is determined

[4]For example in compact and mobile robotics.

by weighting kernels, and the number and parameters of the local kernels are adapted during learning to represent the non-linear function. Because any given training sample activates only a few patches, local learning algorithms are robust against global negative interference. This ensures the flexibility of the learned model towards changes in the dynamics properties of the arm (e.g. load, material wear, and different motion). A local learning strategy therefore seems appropriate for sequential data streams from different, previously unknown input regions, which is the case in robot motion systems.

Locally weighted learning

Locally weighted learning (LWL) methods are a group of nonparametric local learning algorithms that in the past have showed to perform well in the context of (online) motor learning scenarios (Atkeson et al., 1997).

Robot motion data is produced in high frequencies, and typically contains areas with highly non-linear characteristics. Furthermore the domain of real-time robot control demands certain properties of a learning algorithm, namely fast learning rates and high computational efficiency for predictions and updates if the model is trained incrementally. Such an incremental LWL method has been proposed by Schaal and Atkeson (1998), namely the *receptive field weighted regression (RFWR)* algorithm. RFWR allocates the required model resources in a data driven fashion, which makes it very useful for learning online motion data and moreover allows the learned model to be adapted to changes in the dynamics in real-time. Another characteristic property of anthropomorphic robot data is their high dimensionality with many irrelevant and redundant input dimensions. It has been shown in the past that high dimensional motion data often can be locally represented by low dimensional distributions. Therefore local models could be allocated on a low dimensional manifold and still make accurate predictions. *Locally weighted projection regression (LWPR)* (Vijayakumar et al., 2005) exploits this fact by finding locally low dimensional projections that are used to perform more efficient local function approximation. LWPR is extremely robust and efficient for incremental learning of nonlinear models in high dimensions and it will serve as our regression tool of choice.

LWPR

During LWPR training, the parameters of the local models (locality and fit) are updated using incremental *partial least squares (PLS)*. PLS projects the input on a small

number of directions in input space along the directions of maximal correlation with the output and then performs linear regression on the projected inputs. This makes LWPR suitable for high dimensional input spaces. Local models can be pruned or added on an as-need basis, for example, when training data is generated in previously unexplored regions. Usually the areas of validity (also termed its *receptive field*) of each local model are modelled by Gaussian kernels, so their activation or response to a query vector $\mathbf{z} = (\mathbf{x}^T, \mathbf{u}^T)^T$ (combining the *state* and *control* inputs of the forward dynamics **f**) is given by

$$w_k(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{c}_k)^T \mathbf{D}_k (\mathbf{z}-\mathbf{c}_k)\right), \tag{4.1}$$

where $\mathbf{c}_k$ is the centre of the k^{th} linear model and $\mathbf{D}_k$ is its distance metric, which determines the shape and size of the region of validity. The distance metric is updated using a gradient descent based optimisation methods (for details see Vijayakumar et al. (2005)).

Treating each output dimension[5] separately for notational convenience, and ignoring the details about the underlying PLS computations (Klanke et al., 2008), the regression function can be written as

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}), \tag{4.2}$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^T(\mathbf{z}-\mathbf{c}_k), \tag{4.3}$$

where b_k^0 and $\mathbf{b}_k$ denote the offset and slope of the k-th model, respectively.

We mentioned previously that for online scenarios it is not desirable to store all of the received data. So the algorithm must be able to perform the projections, local regressions, and distance metric adaptations in an incremental fashion without storing the data. Therefore LWPR keeps a number of variables that hold *sufficient statistics* for the algorithm to perform the required calculations incrementally.

One of these sufficient statistics contains a forgetting factor λ (Vijayakumar et al., 2005), which balances the trade-off between preserving what has been learned and quickly adapting to the non-stationarity. The forgetting factor can be tuned to the expected rate of external changes. In order to provide some insight, LWPR internally uses update rules within each receptive field of the form $E_{new} = \lambda \cdot E_{old} + w \cdot e_{cur}$. In this example, E is the sufficient statistics for the squared prediction error, and e_{cur} is

[5] In the case of learning forward dynamics, the target values are the joint accelerations. We effectively learn a separate model for each joint.

the error from the current training sample alone, but the same principle applies for other quantities such as the correlation between input and output data. In this way, after N updates to a receptive field, the original value of the sufficient statistics has been down-weighted (or forgotten) by a factor of λ^N. As we will see later, the factor λ can be used to model biologically realistic adaptive behaviour to external force-fields.

Furthermore the statistical parameters of the LWPR regression models provide access to the confidence intervals, here termed *confidence bounds*, of new prediction inputs. In LWPR the predictive variances are assumed to evolve as an additive combination of the variances within a local model and the variances independent of the local model. The predictive variance estimates $\sigma^2_{pred,k}$ for the k-th local model can be computed in analogy with ordinary linear regression. Similarly one can formulate the global variances σ^2 across models. In analogy to (4.2) LWPR then combines both variances additively to form the confidence bounds given by

$$\sigma^2_{pred} = \frac{1}{W^2}\left(\sum_{k=1}^{K} w_k(\mathbf{z})\sigma^2 + \sum_{k=1}^{K} w_k(\mathbf{z})\sigma^2_{pred,k}\right). \tag{4.4}$$

The local nature of LWPR leads to the intuitive requirement that only receptive fields that actively contribute to the prediction (e.g., large linear regions) are involved in the actual confidence bounds calculation. Large confidence bound values typically evolve if the training data contains much noise or other sources of variability such as changing output distributions. Further regions with sparse or no training data, i.e. unexplored regions, show large confidence bounds compared to densely trained regions. The confidence bounds will be used in Chapters 5 and 6 in the context of stochastic OFC.

Fig. 4.2 depicts the learning concepts of LWPR graphically on a learned model with one input and one output dimension. The noisy training data was drawn from an example function that becomes more linear and more noisy for larger $z-$values. Furthermore in the range $z = [5..6]$ no data was sampled for training to show the effects of sparse data on LWPR learning. One can observe that the size and number of the receptive fields (green ellipses on bottom of plot) are adapted according to the nonlinearity of the function, i.e, nonlinear regions have smaller and more receptive fields and linear regions fewer but wider ones.

The discussed LWPR algorithm is capable of approximating functions between high dimensional input and output data, as it is produced from robot motion tasks. It can approximate non-linear functions without any prior knowledge, since it uses data from a local neighbourhood only, while not competing with other models. The major strength of LWPR is that it uses incremental calculation methods and therefore does

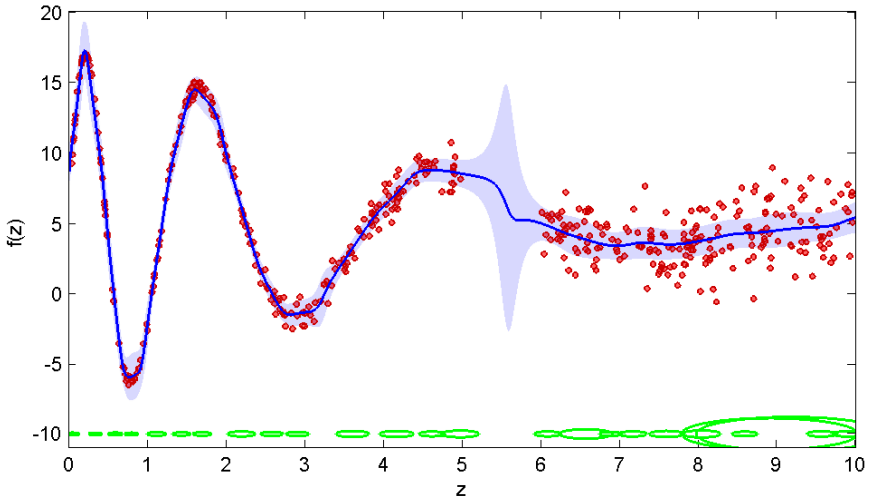


Figure 4.2: Typical regression function (blue continuous line) using LWPR. The dots indicate a representative training data set. The receptive fields are visualised as ellipses drawn at the bottom of the plot. The shaded region represents the confidence bounds around the prediction function. The confidence bounds grow between $\mathbf{z} = [5..6]$ (no training data) and generally towards larger $\mathbf{z}$ values (noise grows with larger values).

not require the input data to be stored. This allows LWPR to learn over an unrestricted period of time. Furthermore the algorithm can cope with highly redundant and irrelevant data input dimensions, because it uses an incremental version of PLS, which automatically determines the required number of projections and the appropriate projection directions.

Other algorithms such as *gaussian process regression (GPR)* (Rasmussen and Williams, 2006) or *support vector regression (SVR)* (Müller et al., 2001) could also be applied for dynamics learning. Nguyen-Tuong et al. (2008b) compared LWPR, GPR and SVR inverse dynamics learning performances where training was performed offline (i.e., batch learning). Data was collected from simulation and from a real SARCOS[6] manipulator. The authors claim higher prediction accuracy using GPR and SVR compared to LWPR. However for LWPR the learning frequency was higher and prediction times were faster. Unlike LWPR, GPR and SVR have the advantage that they do not have a large number of open parameters and therefore little "manual tuning" is required. More fundamentally, basic implementations of GPR and SVR are not designed for online learning, but modified versions of SVR (Vijayakumar and Wu, 1999) and of GPR (Csato and Opper, 2002; Nguyen-Tuong et al., 2008c) are possible enabling online

[6] www.sarcos.com

learning settings. Flentge (2006) proposed the *Locally Weighted Interpolating Growing Neural Gas (LWIGNG)* algorithm, which is based on the principles of growing neural gas and locally weighted learning. The algorithm seems to be well suited to deal with changing target functions and the authors claim comparable performance to LWPR. However LWIGNG has only been tested on toy data and its suitability for high dimensional movement data has not been addressed yet.

Despite the many potential routes for dynamics learning, we believe that LWPR is one of the most suitable methods available, which delivers very efficient and solid learning performance. As in any learning method there are some practical parameter tuning issues that need to be taken into account. A good practical guide for LWPR learning can be obtained from the supplementary documentation[7] of a recently proposed efficient Matlab/C implementation (Klanke et al., 2008), which has been successfully applied within and outside our research group (Castellini et al., 2008; Salaün et al., 2010).

4.2.3 Reducing the computational cost

So far, we have shown how the problem of unknown or changing system dynamics can be addressed within ILQG–LD. Another important issue to discuss is the computational complexity. The ILQG framework has been shown to be one of the most effective locally optimal control method in terms of convergence speed and accuracy (Li, 2006). Nevertheless the computational cost of ILQG remains daunting even for simple movement systems, preventing their application to real-time optimal motion planning for large DoF systems. A large part of the computational cost arises from the linearisation of the system dynamics, which involves repetitive calculation of the system dynamics' derivatives $\partial\mathbf{f}/\partial\mathbf{x}$ and $\partial\mathbf{f}/\partial\mathbf{u}$. When the analytical form of these derivatives is not available, they must be approximated using finite differences. The computational cost of such an approximation scales linearly with the sum of the dimensionalities of $\mathbf{x} = (\mathbf{q}; \dot{\mathbf{q}})$ and $\mathbf{u} = \boldsymbol{\tau}$ (i.e., $3N$ for an N DoF joint torque controlled robot). In simulations, our analysis show that for the 2 DoF manipulator, 60% of the total ILQG computations can be attributed to finite differences calculations. For a 6 DoF arm, this rises to 80%.

Within our ILQG–LD scheme, we can avoid finite difference calculations and rather use the analytic derivatives of the learned model, as has similarly been proposed

[7] www.ipab.inf.ed.ac.uk/slmc/software/lwpr/lwpr_doc.pdf

in (Atkeson et al., 1997). Differentiating the LWPR predictions (4.2) with respect to $\mathbf{z} = (\mathbf{x}; \mathbf{u})$ yields terms

$$\frac{\partial \tilde{f}(\mathbf{z})}{\partial \mathbf{z}} = \frac{1}{W} \sum_k \left(\frac{\partial w_k}{\partial \mathbf{z}} \psi_k(\mathbf{z}) + w_k \frac{\partial \psi_k}{\partial \mathbf{z}} \right) - \frac{1}{W^2} \sum_k w_k(\mathbf{z}) \psi_k(\mathbf{z}) \sum_l \frac{\partial w_l}{\partial \mathbf{z}} \tag{4.5}$$

$$= \frac{1}{W} \sum_k (-\psi_k w_k \mathbf{D}_k(\mathbf{z} - \mathbf{c}_k) + w_k \mathbf{b}_k) + \frac{\tilde{f}(\mathbf{z})}{W} \sum_k w_k \mathbf{D}_k(\mathbf{z} - \mathbf{c}_k) \tag{4.6}$$

for the different rows of the Jacobian matrix $\begin{pmatrix} \partial \tilde{\mathbf{f}}/\partial \mathbf{x} \\ \partial \tilde{\mathbf{f}}/\partial \mathbf{u} \end{pmatrix} = \frac{\partial}{\partial \mathbf{z}} (\tilde{f}_1, \tilde{f}_2, \ldots \tilde{f}_N)^T$.

Table 4.1 illustrates the computational gain (mean CPU time per ILQG iteration) across 3 test manipulators – highlighting added benefits for more complex systems. On a notebook running at 1.6 GHz, the average CPU times for a *complete* ILQG trajectory using the analytic method are 0.8 sec (2 DoF), 1.9 sec (6 DoF), and 9.8 sec (12 DoF), respectively. Note that LWPR is a highly parallelisable algorithm: Since the local models learn independently of each other, the respective computations can be distributed across multiple processors or processor cores, which can yield a further significant performance gain (Klanke et al., 2008).

Table 4.1: CPU time for one ILQG–LD iteration using LWPR (sec).

	finite differences	**analytic Jacobian**	**improvement factor**
2 DoF	0.438	0.193	2.269
6 DoF	4.511	0.469	9.618
12 DoF	29.726	1.569	18.946

The performance gain apparently is correlated to the number of receptive fields in $\tilde{\mathbf{f}}$. It is known that querying points are potentially expensive since every local model could contribute to the output. Therefore both memory and computation costs increases with the "size" of the model. This problem could be tackled by using nearest neighbour type algorithms (e.g., KD-trees) to find only those receptive fields that contribute significantly to the prediction (Atkeson et al., 1997).

4.3 Results

In this section we evaluate ILQG–LD in several setups with increasing complexity. We start with joint torque controlled manipulator setups first, which will be analysed under

stationary and non-stationary conditions. We then present ILQG–LD results from an antagonistic humanoid arm model which embodies the challenge of large redundancies in the dynamics domain.

All simulations are performed with the *Matlab Robotics Toolbox* (Corke, 1996). This simulation model computes the non-linear plant dynamics using standard equations of motion. For an N-DoF manipulator the joint torques $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \tag{4.7}$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the joint angles and joint velocities respectively; $\mathbf{M}(\mathbf{q})$ is the N-dimensional symmetric joint space inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ accounts for Coriolis and centripetal effects, $\mathbf{b}(\dot{\mathbf{q}})$ describes the viscous and Coulomb friction in the joints, and $\mathbf{g}(\mathbf{q})$ defines the gravity loading depending on the joint angles $\mathbf{q}$ of the manipulator.

We study movements for a fixed motion duration of one second, which we discretise into $K = 100$ steps (Δt = 0.01s). The manipulator starts at an initial position $\mathbf{q}_0$ and reaches towards a target $\mathbf{q}_{tar}$. During movement we wish to minimise the energy consumption of the system. We therefore use the cost function

$$v = w_p \,|\, \mathbf{q}_K - \mathbf{q}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + w_e \sum_{k=0}^{K} |\, \mathbf{u}_k \,|^2 \Delta t, \tag{4.8}$$

where the factors for the target position accuracy (w_p), for the zero end-point velocity (w_v), and for the energy term (w_e) weight the importance of each component. We compare the control results of ILQG–LD and ILQG with respect to the number of iterations, the end point accuracy and the generated costs. In the following we will refer to *cost* as total cost defined in (4.8) and to *running cost* to the energy consumption only, i.e., the summation term in (4.8).

4.3.1 Planar arm with 2 torque-controlled joints

The first setup (Fig. 4.3, left) is a horizontally planar 2 DoF manipulator similar to the one used in (Todorov and Li, 2005). The arm is controlled by directly commanding joint torques. This low DoF system is ideal for performing extensive (quantitative) comparison studies and to test the manipulator under controlled perturbations and force fields during planar motion.

Stationary dynamics

First, we compared the characteristics of ILQG–LD and ILQG (both operated in open loop mode) in the case of stationary dynamics without any noise in the 2 DoF plant.

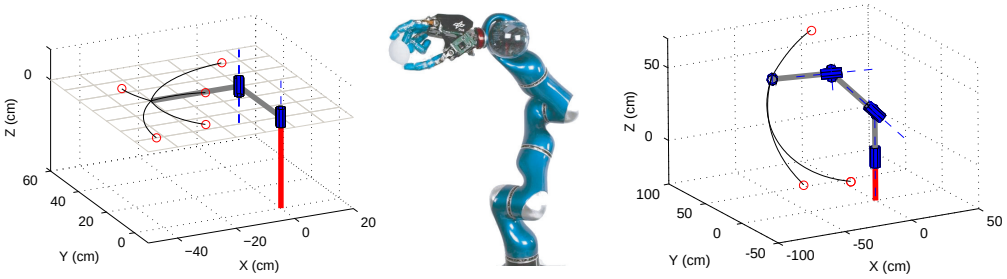


Figure 4.3: Two different joint-torque controlled manipulator models with selected targets (circles) and ILQG generated trajectories as benchmark data. All models are simulated using the Matlab Robotics Toolbox. Left: 2 DoF planar manipulator model; Middle: picture of the Kuka Light-Weight Robot arm (LWR); Right: Simulated 6 DoF LWR model (without hand).

Fig. 4.4 shows three trajectories generated by learned models of different predictive quality, which is reflected by the different normalised mean square errors (nMSE) on test data. The nMSE is defined as $nmse(y,\tilde{y}) = \frac{1}{n\sigma_y^2}\sum_{i=1}^{n}(y_i-\tilde{y}_i)^2$ where y is the desired output data set of size n and $\tilde{y}$ represents the LWPR predictions. The nMSE takes into account the output distribution of the data (variance σ_y^2 in the data) and therefore produces a "dimensionless" error measure. As one would expect, the quality of the model plays an important role for the final cost, the number of ILQG–LD iterations, and the final target distances (cf. the table within Fig. 4.4). For the final learned model, we observe a striking resemblance with the analytic ILQG performance.

Next, we carried out a reaching task to 5 reference targets covering a wide operating area of the planar arm. To simulate control dependent noise, we contaminated the commands $\mathbf{u}$ just before feeding them into the plant, using Gaussian noise with 50% of the variance of the signal $\mathbf{u}$. We then generated motor commands to move the system towards the targets, both with and without the feedback controller. As expected, closed loop control (utilising gain matrices $\mathbf{L}_k$) is superior to open loop operation regarding reaching accuracy. Fig. 4.5 depicts the performance of ILQG–LD and ILQG under both control schemes. Averaged over all trials, both methods show similar endpoint variances and behaviour which is statistically indistinguishable.

Non-stationary dynamics

A major advantage of ILQG–LD is that it does not rely on an accurate analytic dynamics model; consequently, it can adapt *on the fly* to external perturbations and to changes in the plant dynamics that may result from altered morphology or wear and tear. We

ILQG–LD	(L)	(M)	(H)	ILQG
No. of training points	111	146	276	–
Prediction error (nMSE)	0.80	0.50	0.001	–
Iterations	19	17	5	4
Cost	2777.36	1810.20	191.91	192.07
Eucl. target distance (cm)	19.50	7.20	0.40	0.01

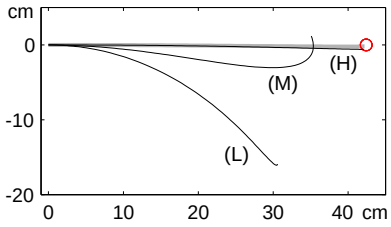
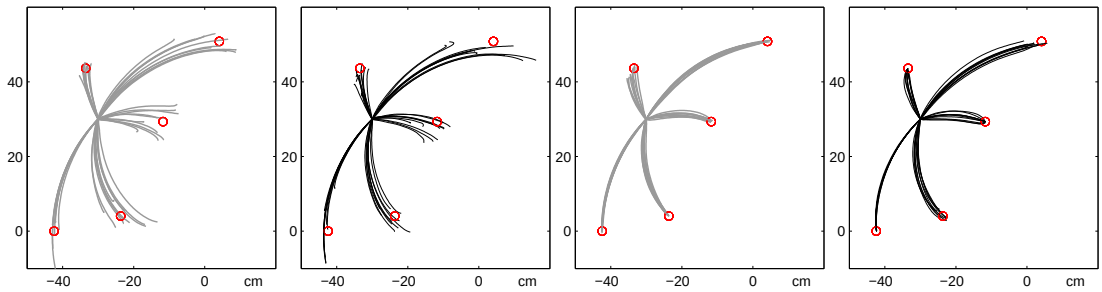


Figure 4.4: Behaviour of ILQG–LD for learned models of different quality: (L)-Low, (M)-Medium, (H)-High. Plot: Trajectories in task space produced by ILQG–LD (black lines) and ILQG (grey line).

carried out adaptive reaching experiments in our simulation similar to the human manipulandum experiments in (Shadmehr and Mussa-Ivaldi, 1994). First, we generated a constant unidirectional *force field (FF)* acting perpendicular to the reaching movement (see Fig. 4.6). Using the ILQG–LD models from the previous experiments, the manipulator gets strongly deflected when reaching for the target because the learned dynamics model cannot account for the "spurious" forces. However, using the resultant deflected trajectory (100 data points) as training data, updating the dynamics model online brings the manipulator nearer to the target with each new trial. We repeated this procedure until the ILQG–LD performance converged successfully. At that point, the internal model successfully accounts for the change in dynamics caused by the FF. Then, removing the FF results in the manipulator overshooting to the other side, compensating for a non-existing FF. Just as before, we re-adapted the dynamics online over repeated trials.

Fig. 4.6 summarises the results of the sequential adaptation process just described. The closed loop control scheme clearly converges faster than the open loop scheme, which is mainly due to the OFC's desirable property of always correcting the system towards the target. Therefore, it produces more relevant dynamics training data. Furthermore, we can accelerate the adaptation process significantly by tuning the forgetting factor λ, allowing the learner to weight the importance of new data more strongly (Vijayakumar et al., 2005). A value of $\lambda = 0.95$ produces significantly faster adapta-



open loop ILQG $d = 2.71 \pm 1.84$ cm | open loop ILQG–LD $d = 3.20 \pm 1.60$ cm | closed loop ILQG $d = 0.19 \pm 0.001$ cm | closed loop ILQG–LD $d = 0.18 \pm 0.001$ cm

Figure 4.5: Illustration of the target reaching performances for the planar 2 DoF in the presence of strong control dependent noise, where d represents the average Euclidean distance to the five reference targets.

tion results than the default of $\lambda = 0.999$. As a follow-up experiment, we made the force field dependent on the velocity $\mathbf{v}$ of the end-effector, i.e. we applied a force

$$\mathbf{F} = \mathbf{B}\mathbf{v}, \quad \text{with} \quad \mathbf{B} = \begin{pmatrix} 0 & 50 \\ -50 & 0 \end{pmatrix} Nm^{-1}s \tag{4.9}$$

to the end-effector. The results are illustrated in Fig. 4.7: For the more complex FF, more iterations are needed in order to adapt the model, but otherwise ILQG–LD shows a similar behaviour as for the constant FF. Interestingly, the overshoot behaviour depicted in Fig. 4.6 and 4.7 has been observed similarly in human adaptation experiments where it was referred to as “after effects” (Shadmehr and Mussa-Ivaldi, 1994). We believe this to be an interesting insight for future investigation of ILQG–LD and its role in modelling sensorimotor adaptation data in the (now extensive) human reach experimental paradigm (Shadmehr and Wise, 2005).

4.3.2 Anthropomorphic 6 DoF robot arm

Our next experimental setup is a 6 DoF manipulator (Fig. 4.3, right), the physical parameters (i.e., link inertia, mass, etc.) of which are a faithful model of the first 6 links of the *Kuka Light-Weight Robot* (LWR).

Using this arm, we studied reaching targets specified in *Cartesian* coordinates $\mathbf{r} \in \mathbb{R}^3$ in order to highlight the redundancy resolution capability and trial-by-trial variability in large DoF systems. We set up the cost function as

$$v = w_p \,|\, \mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + w_e \sum_{k=0}^{K} |\, \mathbf{u}_k \,|^2 \Delta t, \tag{4.10}$$

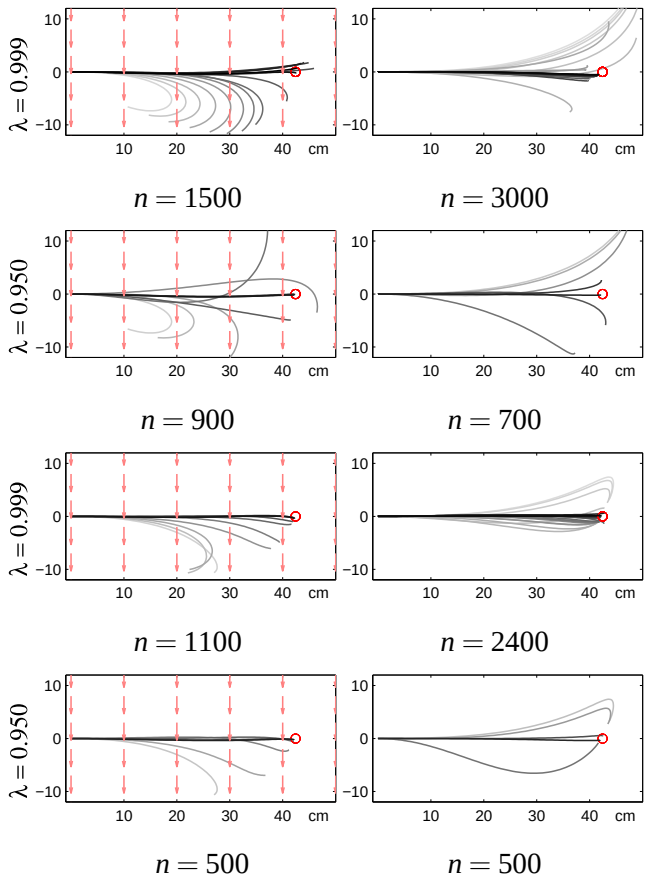


Figure 4.6: Illustration of adaptation experiments for open loop (rows 1,2) and closed loop (rows 3,4) ILQG–LD. Arrows depict the presence of a (constant) force field; n represents the number of training points required to successfully update the internal LWPR dynamics model. Darker lines indicate better trained models, corresponding to later trials in the adaption process. Right column: re-adaptation process after the force field is switched off.

where $\mathbf{r}(\mathbf{q})$ denotes the end-effector position as calculated from forward kinematics. It should be noted that for the specific kinematic structure of this arm, this 3D position depends only on the first 4 joint angles. Joints 5 and 6 only change the orientation of the end-effector[8], which does not play a role in our reaching task and correspondingly in the cost function. In summary, our arm has *one redundant* and further *two irrelevant* degrees of freedom for this task.

Similar to the 2 DoF experiments, we bootstrapped a forward dynamics model through extensive data collection (i.e., motor babbling). Next, we used ILQG–LD (closed loop, with noise) to train our dynamics model online until it converged to sta-

[8]The same holds true for the 7th joint of the original LWR arm.

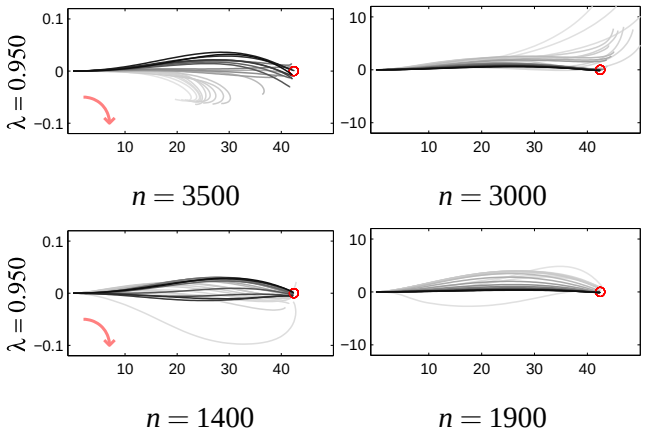


Figure 4.7: Adaptation to a velocity-dependent force field (as indicated by the bent arrow) and re-adaptation after the force field is switched off (right column). Top: open loop. Bottom: closed loop.

Table 4.2: Comparison of the performance of ILQG–LD and ILQG for controlling a 6 DoF robot arm. We report the number of iterations required to compute the control law, the average running cost, and the average Euclidean distance **d** to the three reference targets.

	ILQG			ILQG–LD		
Targets	**Iter.**	**Run. cost**	**d (cm)**	**Iter.**	**Run. cost**	**d (cm)**
(a)	51	**18.50** $\pm$ 0.13	**2.63** $\pm$ 1.63	51	**18.32** $\pm$ 0.55	**1.92** $\pm$ 1.03
(b)	61	**18.77** $\pm$ 0.25	**1.32** $\pm$ 0.69	99	**18.65** $\pm$ 1.61	**0.53** $\pm$ 0.20
(c)	132	**12.92** $\pm$ 0.04	**1.75** $\pm$ 1.30	153	**12.18** $\pm$ 0.03	**2.00** $\pm$ 1.02

ble reaching behaviour. Fig. 4.8 depicts reaching trials, 20 for each reference target, using ILQG–LD with the final learned model. Table 4.2 quantifies the performance. The targets are reached reliably and no statistically significant differences can be spotted between ILQG–LD and ILQG. An investigation of the trials in *joint angle* space also shows similarities. Fig. 4.9 depicts the 6 joint angle trajectories for the 20 reaching trials towards target (c). Please note the high variance of the joint angles especially for the irrelevant joints 5 and 6, which nicely show that task irrelevant errors are not corrected unless they adversely affect the task (minimum intervention principle of OFC). Moreover, the joint angle variances (trial-by-trial variability) between the ILQG–LD and ILQG trials are in a similar range, indicating an equivalent corrective behaviour

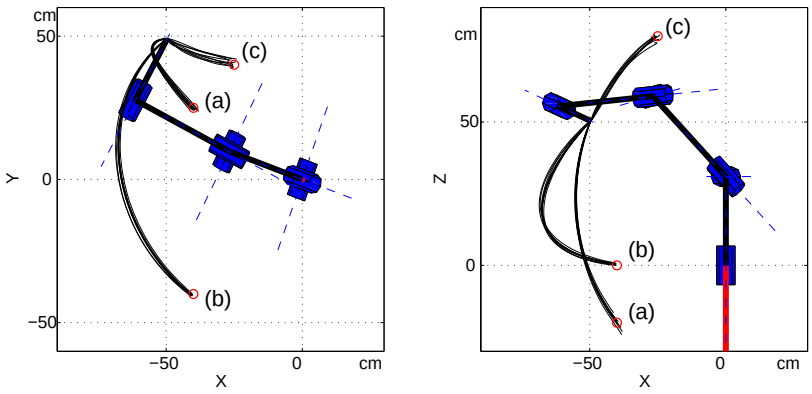


Figure 4.8: Illustration of the trial-by-trial variability of the 6-DoF arm when reaching towards target (a,b,c). Left: top-view, right: side-view.

– the shift of the absolute variances can be explained by the slight mismatch between the learned and analytical dynamics. We can conclude from our results that ILQG–LD scales up very well to 6 DoF, not suffering from any losses in terms of accuracy, cost or convergence behaviour. Furthermore, its computational cost is significantly lower than the one of ILQG.

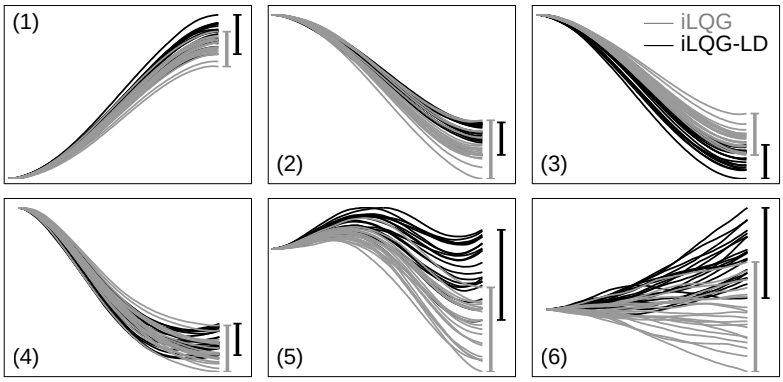


Figure 4.9: Illustration of the trial-by-trial variability in the joint angles (1–6) over time when reaching towards target (c). Grey lines indicate ILQG, black lines stem from ILQG–LD.

4.3.3 Antagonistic planar arm

In order to analyse ILQG–LD in a dynamically redundant scenario, we studied a two DoF planar human arm model, which is actuated by four single-joint and two double-joint antagonistic muscles (Fig. 4.10, left). The arm model described in this section is

based on (Katayama and Kawato, 1993). Although kinematically simple, the system is over-actuated and therefore an interesting testbed for our control scheme, because large redundancies in the dynamics have to be resolved. The dimensionality of the control signals makes adaptation processes (e.g., to external force fields) quite demanding. Indeed this arm poses a harder learning problem than the 6-DoF manipulator of the previous section, because the muscle-based actuation makes the dynamics less linear.

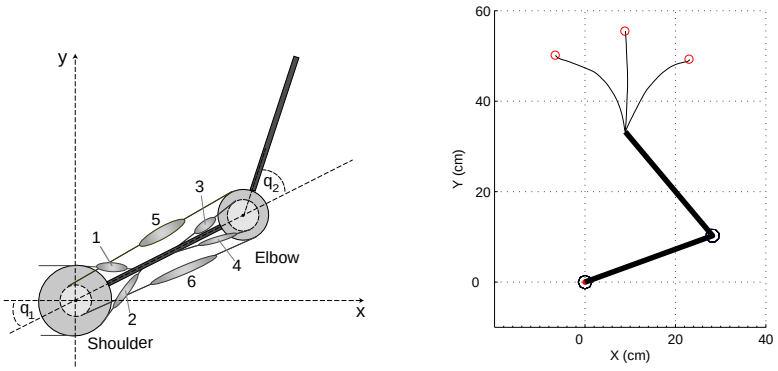


Figure 4.10: Left: Human arm model with 6 muscles (adapted from (Katayama and Kawato, 1993)). Right: Same arm model with selected targets (circles) and ILQG generated trajectories as benchmark data. The physics of the model is simulated using the Matlab Robotics Toolbox (Corke, 1996).

As before the dynamics of the arm is in part based on standard equations of motion, given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}. \tag{4.11}$$

Given the antagonistic muscle-based actuation, we cannot command joint torques directly, but rather we have to calculate effective torques from the muscle activations $\mathbf{u}$. For the present model the corresponding transfer function is given by

$$\boldsymbol{\tau}(\mathbf{q},\dot{\mathbf{q}},\mathbf{u}) = -\mathbf{A}(\mathbf{q})^{\mathbf{T}}\mathbf{t}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u}), \tag{4.12}$$

where $\mathbf{A}$ represents the moment arm. For simplicity, we assume $\mathbf{A}$ to be constant and independent of the joint angles $\mathbf{q}$:

$$\mathbf{A}(\mathbf{q}) = \mathbf{A} = \begin{pmatrix} a_1 & -a_2 & 0 & 0 & a_5 & -a_6 \\ 0 & 0 & a_3 & -a_4 & a_7 & -a_8 \end{pmatrix}^T. \tag{4.13}$$

The muscle lengths $\mathbf{l}$ depend on the joint angles $\mathbf{q}$ through the affine relationship $\mathbf{l} = \mathbf{l}_m - \mathbf{A}\mathbf{q}$, which also implies $\dot{\mathbf{l}} = -\mathbf{A}\dot{\mathbf{q}}$. The term $\mathbf{t}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u})$ in (4.12) denotes the muscle

tension, for which we follow the Kelvin-Voight model (Özkaya and Nordin, 1991) and define:

$$\mathbf{t}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u}) = \mathbf{k}(\mathbf{u})\big(\mathbf{l}_r(\mathbf{u}) - \mathbf{l}\big) - \mathbf{b}(\mathbf{u})\dot{\mathbf{l}}. \tag{4.14}$$

Here, $\mathbf{k}(\mathbf{u})$, $\mathbf{b}(\mathbf{u})$, and $\mathbf{l}_r(\mathbf{u})$ denote the muscle stiffness, the muscle viscosity and the muscle rest length, respectively. Each of these terms depends linearly on the motor commands $\mathbf{u}$, as given by

$$\mathbf{k}(\mathbf{u}) = diag(\mathbf{k}_0 + k\mathbf{u}), \qquad \mathbf{b}(\mathbf{u}) = diag(\mathbf{b}_0 + b\mathbf{u}), \qquad \mathbf{l}_r(\mathbf{u}) = \mathbf{l}_0 + r\mathbf{u}. \tag{4.15}$$

The elasticity coefficient k, the viscosity coefficient b, and the constant r are given from the muscle model. The same holds true for $\mathbf{k}_0$, $\mathbf{b}_0$, and $\mathbf{l}_0$, which are the intrinsic elasticity, viscosity and rest length for $\mathbf{u} = \mathbf{0}$, respectively. For the exact values of these coefficients please refer to (Katayama and Kawato, 1993). ILQG has been applied previously to similar antagonistic arm models, that are slightly more complex. Most notably, non-constant moment arms $\mathbf{A}(\mathbf{q})$, stochastic control signals, and a muscle activation dynamics which increase the dimensionality of the state space have been used (Li, 2006).

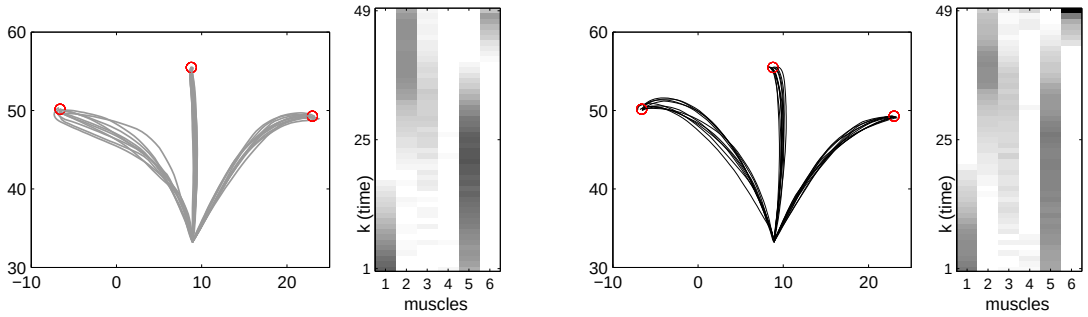
Please note that in contrast to standard torque-controlled robots, in our arm model the dynamics (4.11) is *not* linear in the control signals, since $\mathbf{u}$ enters (4.14) quadratically. We follow the same cost function (4.8) as before and the same fixed motion duration of one second. Here we discretise the time into $K = 50$ steps (Δt = 0.02s).

Stationary dynamics

In order to make ILQG–LD converge for our three reference targets we coarsely pre-trained our LWPR model with a focus on a wide coverage of the workspace. The training data are given as tuples consisting of $(\mathbf{q},\dot{\mathbf{q}},\mathbf{u})$ as inputs (10 dimensions in total), and the observed joint accelerations $\ddot{\mathbf{q}}$ as the desired two-dimensional output. We stopped training once the normalised mean squared error (nMSE) in the predictions reached ≤ 0.005. At this point LWPR had seen $1.2 \cdot 10^6$ training data points and had acquired 852 receptive fields, which is in accordance with the previously discussed high non-linearity of the plant dynamics.

We carried out a reaching task to the 3 reference targets (Fig. 4.10, right) using the feedback controller (feedback gain matrix $\mathbf{L}$) that falls out of ILQG(-LD). To compare the stability of the control solution, we simulated control dependent noise by contaminating the muscle commands $\mathbf{u}$ just before feeding them into the plant. We applied Gaussian noise with 50% of the variance of the signal $\mathbf{u}$.

Fig. 4.11 depicts the generated control signals and the resulting performance of ILQG–LD and ILQG over 20 reaching trials per target. Both methods show similar endpoint variances and trajectories which are in close match. As can be seen from the visualisation of the control sequences, antagonistic muscles (i.e., muscle pairs 1/2, 3/4, and 5/6 in Fig. 4.10, left) are never activated at the same time. This is a direct consequence of the cost function, which penalises co-contraction as a waste of energy. Table 4.3 quantifies the control results of ILQG–LD and ILQG for each target with respect to the number of iterations, the generated running costs and the end point accuracy.



a) ILQG b) ILQG–LD

Figure 4.11: Illustration of an optimised control sequence (left) and resulting trajectories (right) when using a) the known analytic dynamics model and b) the LWPR model learned from data. The control sequences (left target only) for each muscle (1–6) are drawn from bottom to top, with darker grey levels indicating stronger muscle activation.

Table 4.3: Comparison of the performance of ILQG–LD and ILQG with respect to the number of iterations required to compute the control law, the average running cost, and the average Euclidean distance to the three reference targets (left, center, right).

	ILQG			**ILQG–LD**		
Targets	**Iter.**	**Run. cost**	**d (cm)**	**Iter.**	**Run. cost**	**d (cm)**
Center	19	**0.0345** $\pm$ 0.0060	**0.11** $\pm$ 0.07	14	**0.0427** $\pm$ 0.0069	**0.38** $\pm$ 0.22
Left	40	**0.1873** $\pm$ 0.0204	**0.10** $\pm$ 0.06	36	**0.1670** $\pm$ 0.0136	**0.21** $\pm$ 0.16
Right	41	**0.1858** $\pm$ 0.0202	**0.57** $\pm$ 0.49	36	**0.1534** $\pm$ 0.0273	**0.19** $\pm$ 0.12

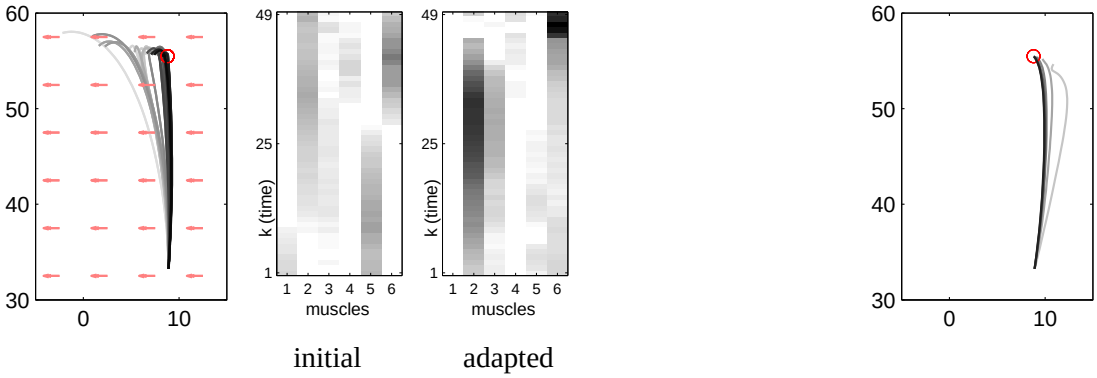


Figure 4.12: Left: Adaptation to a unidirectional constant force field (indicated by the arrows). Darker lines indicate better trained models. In particular, the left-most trajectory corresponds to the "initial" control sequence, which was calculated using the LWPR model (from motor babbling) *before* the adaptation process. The fully "adapted" control sequence results in a nearly straight line reaching movement. Right: Resulting trajectories during re-adaptation after the force field has been switched off (i.e., after effects).

Adaptation results

As before we carried out adaptive reaching experiments (towards the center target) and we generated a constant unidirectional force field (FF) acting perpendicular to the reaching movement (see Fig. 4.12). Using the ILQG–LD model from the previous experiment, the manipulator gets strongly deflected when reaching for the target because the learned dynamics model cannot yet account for the "spurious" forces. However, using the resultant deflected trajectory as training data, updating the dynamics model online brings the manipulator nearer to the target with each new trial. In order to produce enough training data, as is required for a successful adaptation, we generated 20 slightly jittered versions of the optimised control sequences, ran these on the plant, and trained the LWPR model with the resulting 50 samples each. We repeated this procedure until the ILQG–LD performance converged successfully, which was the case after 27000 training samples. At that point, the internal model successfully accounted for the change in dynamics caused by the FF. Then, we switched off the FF while continuing to use the adapted LWPR model. This resulted in an overshooting of the manipulator to the other side, trying to compensate for non-existing forces. Just as before, we re-adapted the dynamics online over repeated trials. The arm reached the target again after 7000 training points. One should note that compared to the initial

global motor babbling, where we required $1.2 \cdot 10^6$ training data points, for the *local* (re-)adaptation we need only a fraction of the data points.

Fig. 4.12 summarises the results of the sequential adaptation process just described. Please note how the optimised *adapted* control sequence contains considerably stronger activations of the extensor muscles responsible for pulling the arm to the right (denoted by "2" and "6" in Fig. 4.10), while still exhibiting practically no co-contraction.

4.4 Relation to other adaptive control methods

Adaptability of controllers towards changes in system dynamics are desired in many applications and a general distinction between non-adaptive and adaptive controllers can be made as described by Dumont and Huzmezan (2002): *"A non-adaptive controller is based solely on a-priori information whereas an adaptive controller is based also on a posteriori information.*" Non-adaptive controllers therefore either ignore posterior (online) information entirely (e.g., classic optimal control such as LQR) or they find control formulations in advance that guarantee stability and performance in the presence of potential changes (e.g., robust control). Adaptive controllers, in contrast, use movement data from the system to update the system properties or the controller to achieve adaptation.

A plethora of work on adaptive control architectures has been published and here we only provide a brief overview in order to put ILQG–LD into perspective. We distinguish three kinds of adaptive controllers: (i) classic adaptive controllers, (ii) iterative learning controllers, (iii) RL controllers. Next we will shortly elaborate on those approaches.

Classic adaptive control has its origins in the 1950's in the development of adaptive flight control and autopilot systems[9]. The main objective of such adaptive controllers is to automatically and continuously update the control parameters and/or the plant model to improve control performance (usually tracking of a desired trajectory) (Narendra and Annaswamy, 1989). *Direct adaptive controllers* update or identify the controller parameters (e.g., PID gains) directly. An example is the *model reference adaptive controller (MRAC)*, which updates the controller in order to match the real system output with the one from the dynamics model. *Indirect adaptive controllers* adapt the system model parameters and then calculate the controller parameters based

[9]Here parameters like plane (fuel) mass or environmental conditions change significantly over time.

on the system model estimates. Well known examples of indirect methods are *self-tuning (ST)* regulators. Another classic adaptive method worth mentioning is *gain scheduling*. Hereby adaptation is achieved by switching or merging between a set of typically linear controllers. Gain scheduling usually requires a lot of prior knowledge about the uncertainty and therefore sometimes is not classified as adaptive controller. A timely review on classic adaptive controllers can be found in Dumont and Huzmezan (2002). A characteristic property of classic adaptive controllers is that their update rules depend on very short-term history, i.e, they react to sudden changes and have no long term memory. This is a significant difference to the ILQG–LD scheme, which is based on long term memory.

In contrast to classic adaptive control methods, *iterative learning controllers (ILC)* remember long term changes in the system, i.e, they remember the previous states and appropriate responses. ILC is a control methodology that is based on the assumption that a system is performing actions repeatedly in a fixed time interval and that the transient performance of the system (i.e., tracking error) can be improved over multiple trials. A prototypical control law in ILC would look as follows: $\mathbf{u}_{k+1} = \mathbf{u}_k + K\mathbf{e}_k$, where $\mathbf{u}_k$ is the control law and $\mathbf{e}_k$ is the tracking error during the k-th trial respectively (Arimoto et al., 1984). The parameter K depicts the design parameter and the objective in ILC is to converge to a zero tracking error. The motivation for ILC from a control theoretical perspective is that it can be used to achieve "perfect" tracking performance (by learning from many trials of the same task), even though the dynamics model may be uncertain. ILC has been studied extensively in the recent years with many applications to robotic manipulators. A technical introduction to ILC can be found in Owens and Hätönen (2005) and for an extensive literature survey on that topic we refer the reader to Ahn et al. (2007). ILC and ILQG–LD have similarities in that they both perform learning trails iteratively and that they operate on a fixed time horizon. However the underlying control objectives are fundamentally different in that ILQG–LD solves an closed loop optimal control problem for both, planning and control whereas ILC only corrects tracking errors w.r.t. some given trajectory[10].

The last group of adaptive control algorithms that we wish to mention is *reinforcement learning (RL)* (Sutton and Barto, 1998). Unlike the previous two approaches, which have their origins in classic control theory, RL is often related to the machine learning, psychology or neuroscience community. In RL an agent is learning by interacting within an environment in a *trial and error* fashion. Unlike in the classic

[10]Many ILC approaches use optimality principles to achieve a reduction in the tracking error.

adaptive control or in the ILC, in RL the agent is not being instructed which actions to take. For each applied action in a current state, the agent receives a numerical reward, which describes the success of the applied action's outcome. The agent then learns from the consequences of its actions to select those that maximise the accumulated reward over time. In RL the dynamics model may be given (model-based) or not (model-free), however the expected reward function is always learned from previous trials. In contrast for classic OFC the cost function and the dynamics model are given and the control law (or cost-to-go) is defined at all times. In some sense methods based on DP, which is the most general way to solve OFC problems, can be understood as a special case of RL where *all* relevant information is given upfront and no learning is involved. But how does ILQG–LD relate to RL then? The situation in our framework is similar to classic OFC with analytic dynamics. In our case the dynamics is acquired from data after each trial. However the cost function is always given and finding an optimal control law involves no learning.

In conclusion ILQG–LD can be classified as an adaptive control method because it uses, similarly to ILC, trial-by-trial data to improve performance. However unlike ILC which is concerned with stability and tracking performance towards a desired trajectory, ILQG–LD also compute an optimal control law and therefore can be linked conceptually to the problem of RL.

4.5 Discussion

In this chapter we introduced ILQG–LD, a method that realises adaptive optimal feedback control by incorporating a learned dynamics model into the ILQG framework. Most importantly, we carried over the favourable properties of ILQG to more realistic control problems where the analytic dynamics model is often unknown, difficult to estimate accurately or subject to changes. As with ILQG control, redundancies are implicitly resolved by the OFC framework through a cost function, eliminating the need for a separate trajectory planner and inverse kinematics/dynamics computation.

Utilising the derivatives (4.6) of the learned dynamics model $\tilde{\mathbf{f}}$ avoids expensive finite difference calculations during the dynamics linearisation step of ILQG. This significantly reduces the computational complexity, allowing the framework to scale to larger DoF systems. We empirically showed that ILQG–LD performs reliably in the presence of noise and that it is adaptive with respect to systematic changes in the dynamics; hence, the framework has the potential to provide a unifying tool for modelling

(and informing) non-linear sensorimotor adaptation experiments even under complex dynamic perturbations.

Most limitations of the proposed framework can be associated with issues related to local learning methods. While local learning assures robustness towards negative inference they are known to have limited generalisation capabilities, especially when many narrow receptive fields are used in the case of highly nonlinear dynamics. Furthermore the successful practical use of LWPR requires some amount of experience and good intuition in terms of parameter tuning and best learning practices. Another limitation is that the amount of data required to cover the whole dynamics space grows exponentially with the number of input dimensions. A potential route of improvement could be a combination of LWPR learning with an analytic model. In such a scenario the combined dynamics could for example be written as $\mathbf{f}_t(\mathbf{x},\mathbf{u}) = \mathbf{f}_a(\mathbf{x},\mathbf{u}) + \tilde{\mathbf{f}}_{err}(\mathbf{x},\mathbf{u})$, where $\mathbf{f}_a$ is a crude rigid body dynamics model, $\tilde{\mathbf{f}}_{err}(\mathbf{x},\mathbf{u})$ is a learned error dynamics model between real dynamics data and the dynamics prediction $\mathbf{f}_a$. Like this one could potentially avoid convergence problems in ILQG–LD due to regions where sparse or no data has been collected, because it would be covered by the analytic part of the dynamics.

In simulation, if the dynamics are not excessively high-dimensional and the dynamics are well-tempered, we can obtain good learning and control results. On real hardware systems the situation can be expected to be more difficult. Collecting vast amounts of data on real systems is demanding and prone to hardware failures. Furthermore the noise issue on real systems is more severe. Many manipulators are only equipped with position sensors and to acquire joint velocities and accelerations by numerical differentiation one automatically introduces significant sources of noise, which need additional filtering efforts. Furthermore per se the LWPR learning does not “solve” the problem of discontinuities in the dynamics due to friction as discussed in Chapter 3. Depending on the chosen learning parameters LWPR may overfit discontinuities; this would produce steep gradients causing convergence problems in ILQG–LD. On the other hand LWPR could also oversmooth the dynamics and therefore loosing predictive quality. The real problem here is that it is difficult to predict what the learning will do in such situations.

Despite certain drawbacks of the local learning methodology the combination of locally weighted learning with local OFC seems a reasonable approach since both algorithms (ILQG and LWPR) rely mainly on localised information.

As we show in the next chapters ILQG–LD may serve as a promising route to ac-

quire learned stochasticity models in order to improve control performance of redundantly actuated systems. Furthermore we can exploit this framework for understanding OFC and its link to biological motor control.

Chapter 5

Exploiting stochastic information for improved motor performance

In this chapter, we extend our focus on issues related to adaptive motor control of antagonistically actuated robots. Modern anthropomorphic robotic systems increasingly employ variable impedance actuation in order to achieve robustness to uncertainty, superior agility and efficiency that are hallmarks of biological systems. Controlling and modulating impedance profiles such that it is optimally tuned to the controlled plant is crucial to realise these benefits. We propose a methodology to generate optimal control commands for variable impedance actuators under a prescribed trade-off of task accuracy and energy cost. Hereby we extend our OFC-LD framework to incorporate *both* the process dynamics as well as the stochastic properties of the plant. This enables us to prescribe an optimal impedance and command profile (i) tuned to the hard-to-model stochastic characteristics of a plant and (ii) adapt to the systematic changes such as a change in load. To evaluate the scalability of our framework to real hardware, we build a novel antagonistic *series elastic actuator (SEA)* characterised by a simple mechanical architecture. We present results on this hardware that highlight how impedance modulation profiles tuned to the plant dynamics emerge from the first principles of optimisation. Furthermore, we illustrate how *changes* in plant dynamics and stochastic characteristics (e.g., while using a power tool) can be accounted for by using this adaptation paradigm, achieving clear performance gains over classical methods that ignore or are incapable of incorporating this information.

5.1 Introduction

Humans have remarkable abilities in controlling their limbs in a fashion that outperforms most artificial systems in terms of versatility, compliance and energy efficiency. The fact that biological motor systems suffer from significant noise, sensory delays and other sources of stochasticity (Faisal et al., 2008) makes its performance even more impressive. Therefore, it comes as no surprise that biological motor control is often used as a benchmark for robotic systems. Biological motor control characteristics, on the one hand, are a result of the inherent biophysical properties of human limbs and on the other hand, are achieved through a framework of learning and adaptation processes (Wolpert et al., 1995; Kawato, 1999; Davidson and Wolpert, 2005). These concepts can be transferred to robotic systems by (i) developing appropriate anthropomorphic hardware and (ii) by employing learning mechanisms that support motor control in the presence of noise and perturbations (Mitrovic et al., 2008a).

Antagonistic actuator designs are based on the biological principle of opposing muscle pairs (see Chapter 4, Section 4.3.3). Therefore, the joint torque motors, for example, of a robotic arm are replaced by opposing actuators, typically coupled via mechanical springs (Pratt and Williamson, 1995). Such *series elastic actuators (SEA)* have found increasing attention over the last decades (Vanderborght et al., 2009) as they provide several beneficial properties over classic joint torque actuated systems:

i. *Impedance control & variable compliance:* Through the use of antagonistic actuation, the system is able to vary co-contraction levels which in turn change the system's mechanical properties – this is commonly referred to as *impedance control* (Hogan, 1984). Impedance in a mechanical system is defined as a measure of force response to a motion exerted on the system and is made of constituent components such as *inertia*, *damping*, and *stiffness*. In general SEAs can only vary stiffness of a system and achieving variable damping is technically challenging (e.g., Laffranchi et al. (2010)). Consequently, in this chapter, when we refer to *impedance control*, we will solely address a *change in stiffness* and ignore variable damping or variable inertia. Antagonistic actuation introduces an additional degree of freedom in the limb dynamics, i.e., the same joint torque can be achieved by different muscle activations. This means a low co-contraction leads to low joint impedance whereas a high co-contraction increases the joint impedance. This degree of freedom can be used beneficially in many motion tasks, especially those involving manipulation or interaction with

tools. It has been shown through many neurophysiological studies (e.g. in Burdet et al. (2001)) that humans are capable of modulating this impedance in an optimal way with respect to the task demands, trading off selectively against energy consumption. For example, when you use a drilling machine to drill holes into a wall, you will *learn* to co-contract your muscles such that the random perturbations of the drilling has minimal impact on your task. We will discuss impedance control in humans in more detail in Chapter 6. Furthermore, the ability to vary the impedance (and therefore the compliance) of joints plays a crucial role in robot safety (Zinn et al., 2004). In general, impedance modulation is an efficient way to control systems that suffer from noise, disturbances or sensorimotor delays.

ii. *Energy efficiency & energy storage:* By appropriately controlling the SEA, one can take into account the passive properties of the springs and produce control strategies with low energy demands. A well known example is walking, where the spring properties combined with an ideal actuation timing can be used to produce energetically efficient gaits (Collins and Ruina, 2005; Collins and Kuo, 2010). Furthermore, SEAs have impressive energy storage and fast discharge capabilities, enabling "explosive" behaviours such as throwing a ball (Wolf and Hirzinger, 2008) – which is quite hard to achieve with regular joint torque actuators. Therefore series elasticity can amplify power and work output of an actuator, which is important in the fabrication of light-weight but powerful robotic or prosthetic devices (Paluska and Herr, 2006).

A disadvantage of antagonistic actuation is that it imposes higher demands on the redundancy resolution capabilities of a motor controller. As discussed earlier optimality principles have successfully been used in biological (Flash and Hogan, 1985; Todorov, 2004; Scott, 2004) and in artificial systems (Nakamura and Hanafusa, 1987; Cortes et al., 2001) as a principled strategy to resolve redundancies in a way that is beneficial for the task at hand. More specifically, stochastic OFC appears to be an especially appealing theory as it studies optimality principles under the premise of noisy and uncertain dynamics. Another important aspect when studying stochastic systems is how the information, for example, about noise or uncertainty is obtained without prior knowledge. Supervised learning methods like LWPR can provide a viable solution to this problem as they can be used to extract information from the plant's sensorimotor data directly.

Here, we use a control strategy for antagonistic systems which is based on stochastic optimal control theory under the premise of a minimal energy cost. Using the proposed OFC-LD framework enables us (i) to adapt to systematic changes of the plant and more crucially (ii) extract its stochastic properties. *Stochastic properties* or *stochastic information* refers to noise or random perturbations of the controlled system that cannot be modelled deterministically. By incorporating this stochastic information into the optimisation process, we show how impedance modulation and co-contraction behaviour emerges as an optimal control strategy from first principles.

In the next section, we present a new antagonistic actuator, which serves as our implementation test-bed for studying impedance control in the presence of stochasticity and which, compared to previous antagonistic designs, has a much simpler mechanical design. Using the local learning framework we then propose a systematic methodology for incorporating dynamic and stochastic plant information into the optimal control framework, resulting in a scheme that improves performance significantly by exploiting the antagonistic redundancy of our plant. Our claims are supported by a number of experimental evaluations on real hardware in Section 5.4. We conclude this chapter with a discussion and an outlook.

5.2 A novel antagonistic actuator design for impedance control

To study impedance control, we developed an antagonistic joint with a simple mechanical setup. Our design is based on the SEA approach in which the driven joint is connected via spring(s) to a stiff actuator (e.g., a servo-motor). A variety of SEA designs have been proposed (for a recent review see Vanderborght et al. (2009)), which we here classify into *pseudo-antagonistic* and *antagonistic* setups. Pseudo antagonistic SEA have one or multiple elastic elements which are connected between the driving motor and the driven joint. The spring tension and therefore the joint stiffness is regulated using a mechanism equipped with a second actuator. Antagonistic SEA have one motor per opposing spring and the stiffness is controlled through a combination of both motor commands. Therefore, in antagonistic designs, the relationship between motor commands and stiffness must be resolved by the controller. This additional computational cost is the trade-off for a biologically plausible architecture.

For antagonistic SEA, nonlinearity of the springs is essential to obtain a variable

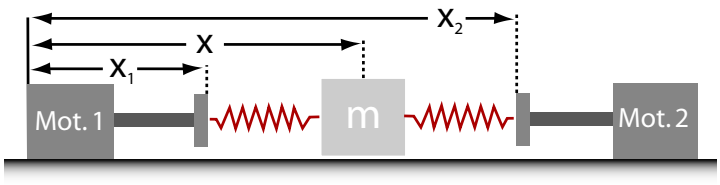


Figure 5.1: Schematic to demonstrate the problem of linear springs in an antagonistic setting.

compliance (van Ham et al., 2009). Because forces produced through springs with linear tension to force characteristics tend to cancel out in an antagonistic setup, an increase in the tension of both springs (i.e., co-contraction) does not change the stiffness of the system. For example, consider the simple antagonistic setting depicted in Fig. 5.1, which consists of a horizontally movable mass connected to two linear actuators via two identical linear springs each with spring constant κ and rest length zero. By actuating the motors 1 and 2 we can change the length x_1 and x_2 of each spring. Following *Hooke's law* the sum of forces acting on the mass is

$$F = -\kappa(x - x_1) + \kappa(x_2 - x) = -2\kappa x + \kappa(x_1 + x_2) \tag{5.1}$$

and consequently the stiffness becomes

$$K = \frac{dF}{dx} = -2\kappa, \tag{5.2}$$

which means it is independent of the motor commands. Therefore co-contracting does not change the stiffness of mass m. If the linear springs are replaced with quadratic ones we get the total force acting on m

$$F = -\kappa(x - x_1)^2 + \kappa(x - x_2)^2 = 2\kappa x(x_1 - x_2) + \kappa(x_1^2 - x_2^2) \tag{5.3}$$

and consequently the stiffness is

$$K = \frac{dF}{dx} = 2\kappa(x_1 - x_2), \tag{5.4}$$

meaning it scales linear with the level of co-contraction.

Now commercially available springs usually have linear tension to force characteristics and consequently most antagonistic SEA require relatively *complex mechanical structures* to achieve such a non-linear tension to force curve (Hurst et al., 2004; Migliore et al., 2005; Tonietti et al., 2005). A graphical summary can be found in Fig. 5.2, which shows a classification of SEA originally proposed by Vanderborght et al.

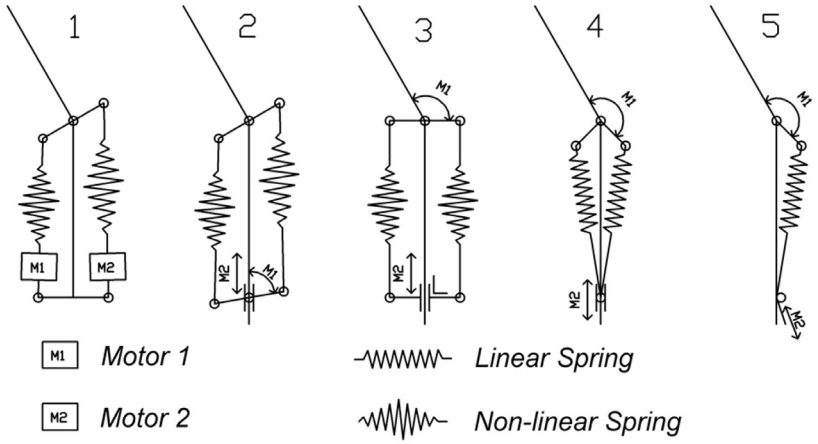


Figure 5.2: Different SEA designs reproduced from Vanderborght et al. (2009). Antagonistic designs (1,2,3) require a nonlinear spring mechanism while pseudo-antagonistic (4,5) can work with linear springs.

(2009). These mechanisms typically increase construction and maintenance effort but also can complicate the system identification and controllability, for example, due to added mechanical couplings, drag and friction properties. We directly addressed this aspect in our design of the SEA, which primarily aims to achieve variable stiffness characteristics using a simple mechanical setup.

5.2.1 Variable stiffness with linear springs

Here we propose a SEA design which does not rely on complex mechanisms to achieve variable stiffness but achieves the desired properties through a specific geometric arrangement of the springs. While the emphasis of this thesis is not on the mechanical design of actuators, we will explain the essential dynamic properties of our testbed. Fig. 5.3 shows a sketch of the robot, which is mounted horizontally and consists of a single joint and two antagonistic servomotors that are connected to the joint via *linear springs*. The springs are mounted with a moment arm offset a at the joints and an offset of L at the motors. Therefore, the spring endpoints move along circular paths at the joints and at the motors. Under the assumption that the servo motors are infinitely stiff, we can calculate the torque τ acting on the arm as follows. Let $\mathbf{s}_1$ denote the vector from point C to A, and $\mathbf{s}_2$ the vector from D to B, and s_1 and s_2 their respective length.

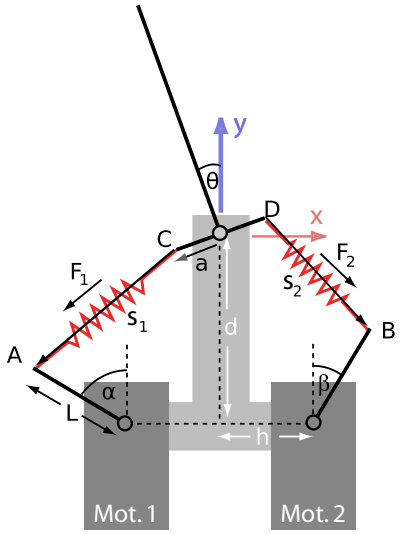


Figure 5.3: Schematic of the variable stiffness actuator. The robot dimension are: $a = 15mm$, $L = 26mm$, $d = 81mm$, $h = 27mm$. The spring rest length is $s_0 = 27mm$.

Putting the origin of the coordinate system at the arm joint, we have

$$\mathbf{s}_1 = \begin{pmatrix} -h - L\sin\alpha \\ -d + L\cos\alpha \\ 0 \end{pmatrix} - \underbrace{\begin{pmatrix} -a\cos\theta \\ -a\sin\theta \\ 0 \end{pmatrix}}_{=\mathbf{a}_1}, \quad \mathbf{s}_2 = \begin{pmatrix} h + L\sin\beta \\ -d + L\cos\beta \\ 0 \end{pmatrix} - \underbrace{\begin{pmatrix} a\cos\theta \\ a\sin\theta \\ 0 \end{pmatrix}}_{=\mathbf{a}_2} \tag{5.5}$$

Denoting the spring constant by κ and the rest length by s_0, this yields forces

$$\mathbf{F}_1 = \kappa(s_1 - s_0)\frac{\mathbf{s}_1}{s_1} \quad \text{and} \quad \mathbf{F}_2 = \kappa(s_2 - s_0)\frac{\mathbf{s}_2}{s_2}. \tag{5.6}$$

Given the motor positions α and β and the arm position θ, the torque generated by the springs is

$$\tau(\alpha, \beta, \theta) = \hat{\mathbf{z}}^T(\mathbf{F}_1 \times \mathbf{a}_1 + \mathbf{F}_2 \times \mathbf{a}_2), \tag{5.7}$$

where $\hat{\mathbf{z}}^T$ denotes the three dimensional basis vector $(0,0,1)^T$. To calculate the equilibrium position θ_{eq} for given motor positions α and β, we need to solve $\tau(\alpha, \beta, \theta_{eq}) = 0$, which in practice we do by numerical optimisation. At this position, we can calculate the joint stiffness as

$$K(\alpha, \beta) = \frac{\partial}{\partial\theta}\tau(\alpha, \beta, \theta)\Big|_{\theta=\theta_{eq}}. \tag{5.8}$$

Please note that K depends linearly on the spring stiffness κ, but that the geometry of the arm induces a nonlinear dependency on α and β. Fig. 5.4 shows the computed profiles of the equilibrium position and stiffness, respectively.

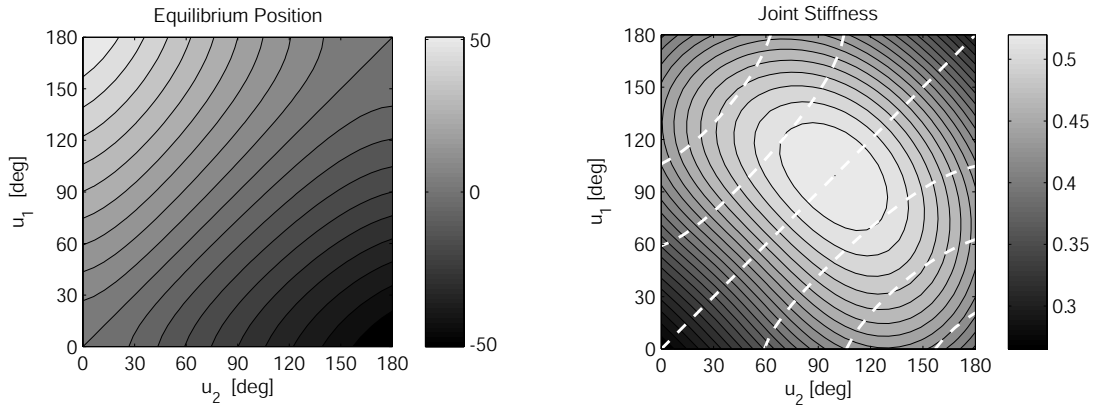


Figure 5.4: Left: Equilibrium position as a function of the motor positions (in degrees), with contour lines spaced at 5 degree intervals. Right: Stiffness profile of the arm, as calculated from (5.8). The maximum achievable stiffness is 150% of the intrinsic spring stiffness.

Further denoting the arm's inertia around the z-axis by I_z and a damping torque given by $\tau(\dot{\theta}) = -D\dot{\theta}$, the dynamics equation can be analytically written as:

$$I_z\ddot{\theta} = \tau(\alpha, \beta, \theta) - D\dot{\theta}. \tag{5.9}$$

5.2.2 Actuator hardware

Fig. 5.5 depicts our prototype SEA hardware implementation of the discussed design. For actuation, we employ two servo motors (Hitec HSR-5990TG), each of which is connected to the arm via a spring mounted on two low friction ball bearings. To avoid excessive oscillations, the joint is attached to a rotary viscous damper. The servos are controlled using 50 Hz PWM signals by an Arduino Duemilanove microcontroller board (Atmel ATmega328). That board also measures the arm's joint angle θ with a contact-free rotary position encoder (Melexis MLX90316GO), as well as its angular acceleration $\ddot{\theta}$ using a LilyPad accelerometer (Analog Devices ADXL330). Finally, we also measure the servo motor positions by feeding a signal from their internal potentiometer to the AD converters of the Arduino. While the operating frequency is limited to 50 Hz due to the PWM control, all measurements are taken at a 4x higher frequency and averaged on the board to reduce the amount of noise, before sending the results to a PC via a serial connection (RS232/USB).

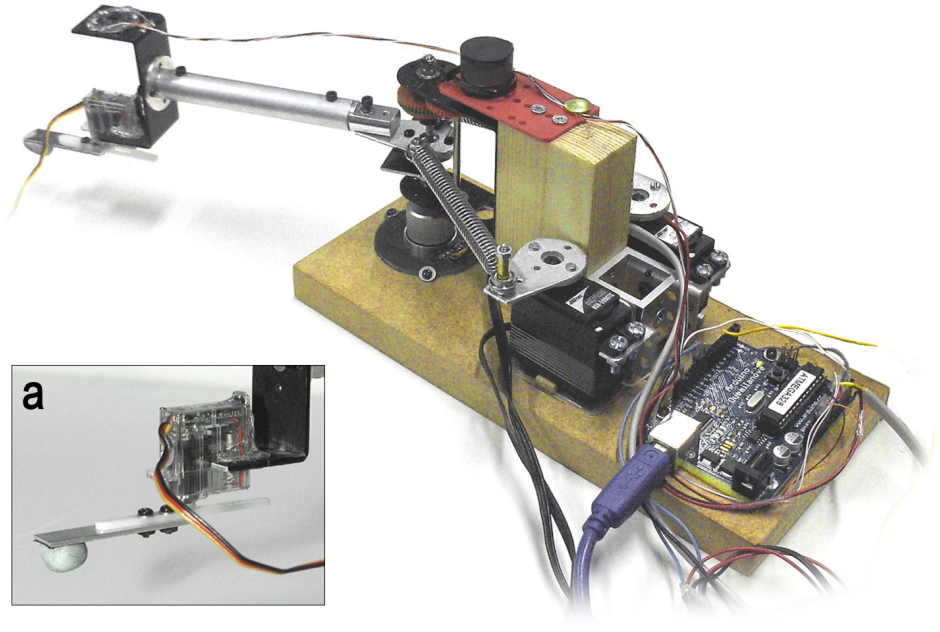


Figure 5.5: Photograph of our antagonistic robot. Inset panel (a): Separate servo motor mounted at the end of the arm to create stochastic perturbations (see Section 5.4.2).

5.2.3 System identification

Apart from measuring the exact dimensions ($L = 2.6cm,\ a = 1.5cm,\ h = 2.7cm,\ d = 8.1cm$) of the robot, and the stiffness constant of the spring (κ=424 N/m), system identification consists of a series of steps, each of which involves a least-squares fit between known and actually measured quantities.

1. *Identify servo motor dynamics:* The servo motors are controlled by sending the desired position (encoded as a PWM signal), which we refer to as u_1 and u_2 for motor 1 and 2, respectively. Even though the servo motor we use are very accurate, they need some time to reach the desired position, and therefore we model the true motor positions (α, β) as a low-pass filtered version of the commands (u_1, u_2) using a finite impulse response (FIR) filter, i.e.,

$$\alpha[n] = (h * u_1)[n] + \varepsilon[n] = \sum_{k=0}^{K} h[k]u_1[n-k] + \varepsilon[n] \tag{5.10}$$

and similar for β and u_2. Please note that square brackets $[\cdot]$ in this chapter denote discrete time indices. The term $\varepsilon[t]$ denotes a noise component of the true motor position that cannot be modelled with the FIR filter. By using the internal potentiometer of the servo-motors, we can measure the actual motor

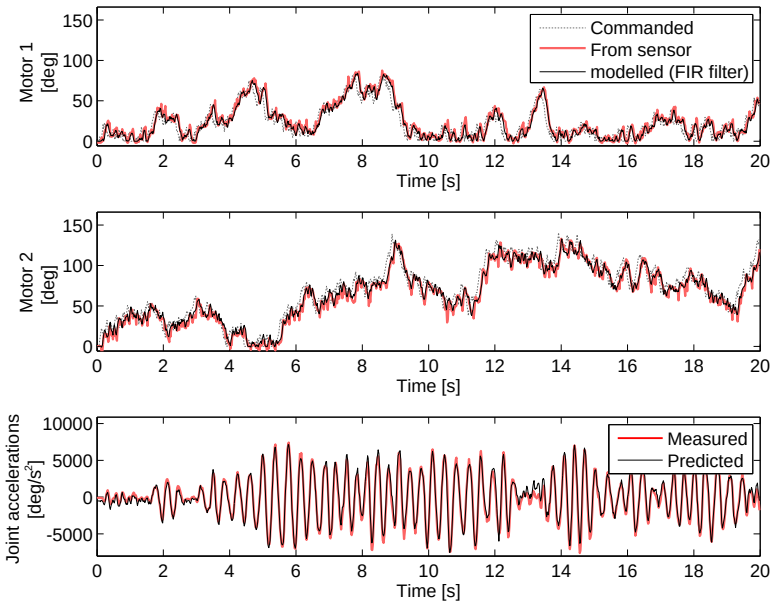


Figure 5.6: Comparison of prediction performance of estimated motor dynamics (top and middle) and of arm dynamics (bottom) for an independent test data set.

positions to identify the filter coefficients h_i using a least squares fit, that is, by minimising $\sum_t (\alpha[n] - (h * u_1)[n])^2$ with respect to h_i. We retrieved a good fit of the motor dynamics (cf. Fig. 5.6) using a FIR filter of 7 steps length, with estimated coefficients $h = [0, 0, 0, 0.0445, 0.2708, 0.3189, 0.3658]$.

2. *Calibrate position sensor:* Tests with the position sensor revealed linear position characteristics. By moving the arm physically to several pre-defined and geometrically measured positions, we determined the sensor's offset and slope.

3. *Calibrate acceleration sensor:* We matched the accelerations measured with the accelerometer with accelerations derived from the position sensor (using finite differences).

4. *Collect training data and fit parameters:* We carried out motor babbling (any excitation movements are applicable) on the servos and measured the resulting arm positions, velocity, and accelerations. Taking into account the estimated motor dynamics using the fitted filter, we estimated the arm's inertia ($I_z = 1.28 \cdot 10^{-3}$) and viscous damping ($D = 0.65$) coefficient using least squares from (5.9).

On a large independent test set of $S_{test} = 300000$ data points the motor prediction produces a *normalised mean squared error (NMSE)* of $e_{nmse} = 1.85\%$. Fig. 5.6 shows

an example prediction performance for a sequence of random motor commands (20 seconds from the test set S_{test}) using the estimated dynamics model.

5.3 Stochastic optimal control

For a system with deterministic (and accurately modelled) dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, it is sufficient to find the optimal *open loop* sequence of commands $\mathbf{u}(t)$ and the associated trajectory $\mathbf{x}(t)$ that minimises the cost function J, which can usually be obtained by solving a two point boundary difference/differential equation derived by applying *Pontryagin's maximum principle* (Stengel, 1994). In practice, in the presence of small perturbations or modelling errors, the optimal open loop sequence of commands can be run on the real plant together with a simple PD controller that corrects deviations from the planned trajectory. However, those corrections will usually not adhere to the optimality criterion, and the resulting cost J will be higher.

Alternatively, we can try to incorporate stochasticity, e.g., as dynamics model

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})dt + \mathbf{F}(\mathbf{x}, \mathbf{u})d\xi \quad , \quad \xi \sim N(0, \mathbf{I}) \tag{5.11}$$

directly into the optimisation process and minimise the *expected* cost. Here, $d\xi$ is a Gaussian noise process and $\mathbf{F}(\cdot)$ indicates how strongly the noise affects which parts of the state and control space. A well studied example of this case is the LQG problem, which stands for *linear* dynamics ($\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$), *quadratic* cost (in both $\mathbf{x}$ and $\mathbf{u}$), and *Gaussian* noise ($\mathbf{F}$ is constant). A solution to these class of problems is the *optimal feedback controller* (OFC), that is, a policy $\mathbf{u} = \pi(\mathbf{x}, t)$ that calculates the optimal command $\mathbf{u}$ based on feedback $\mathbf{x}$ from the real plant. In the LQG case, the solution is a linear feedback law $\mathbf{u} = \mathbf{L}_t\mathbf{x}$ with pre-computed gain matrices[1] $\mathbf{L}_t$ (Stengel, 1994).

Solving OFC problems for more complex systems (non-linear dynamics, non-quadratic cost, varying noise levels $\mathbf{F}$) is a difficult computational task. A general way to solve OFC problems for non linear quadratic problems is *Dynamic Programming (DP)* (Bellman, 1957), which suffers from the curse of dimensionality (see Chapter 2). For example, lets consider a discretisation of 100 steps for each variable of the state and action space. In the case of the presented SEA, this corresponds to a state space dimensionality $n = 2$, for positions and velocities, and action space dimensionality $m = 2$, for the two motors[2]. Even for this low dimensional system the possible

[1] For the infinite-horizon case, the matrix is constant.

[2] Please note that we have ignored any motor dynamics.

combinations of states and actions that DP needs to evaluate and store in order to find the optimal control law are $p = 100^4 = 100000000$. One way to avoid the curse of dimensionality is to restrict the state space to a region that is close to a nominal optimal trajectory. Therefore we will as before resort to ILQG as a solution technique.

5.3.1 Modelling dynamics and noise through learning

Analytical dynamics formulations as described in Section 5.2.1 or in (5.9) have the tremendous advantage of being compact and fast to evaluate numerically, but they also suffer from drawbacks. First, their accuracy is limited to the level of detail put into the physical model. For example, our model is based on the assumption that the robot is completely symmetric, that both motors are perfectly calibrated, and that the two springs are identical, but in reality we cannot avoid small errors in all of these. Second, the analytical model does not provide obvious ways to model changes in the dynamics, such as from wear and tear, or more *systematic* changes due to the weight of an added tool.

While these problems can to some extend be alleviated by a more involved and repeated system identification process, the situation is more difficult if we consider the noise model $\mathbf{F}(\cdot)$, or the *stochastic* changes to the dynamics. For example, an arm might be randomly perturbed by tool interactions such as when drilling into a wall, with stronger effects for certain postures, and milder effects for others. It is not obvious how one can model state dependent noise analytically.

We therefore propose to include a supervised learning component and to acquire both the dynamics and the noise model in a data-driven fashion (Fig. 5.7). Our method of choice as before is LWPR, because that algorithm allows us to adapt the models incrementally and online, and most importantly it is able to reflect heteroscedastic[3] noise in the training data through localised confidence intervals around its predictions. More details about learning with LWPR can be found in Section 4.2.2 of Chapter 4.

In order to simplify the presentation as much as possible, and also due to technical challenges of operating on the real hardware (see discussion in Section 5.5), in this work we only learn the stochastic mapping $f(\mathbf{u})$ from motor positions to joint angle θ, not taking into account velocities and accelerations. During stationary conditions and in the absence of perturbations, this mapping reflects the equilibrium position of

[3]Heteroscedastic noise has different variances across the state and action space. For example, the variance of the noise can scale with the magnitude of the control signal $\mathbf{u}$, which is also called signal dependent noise.

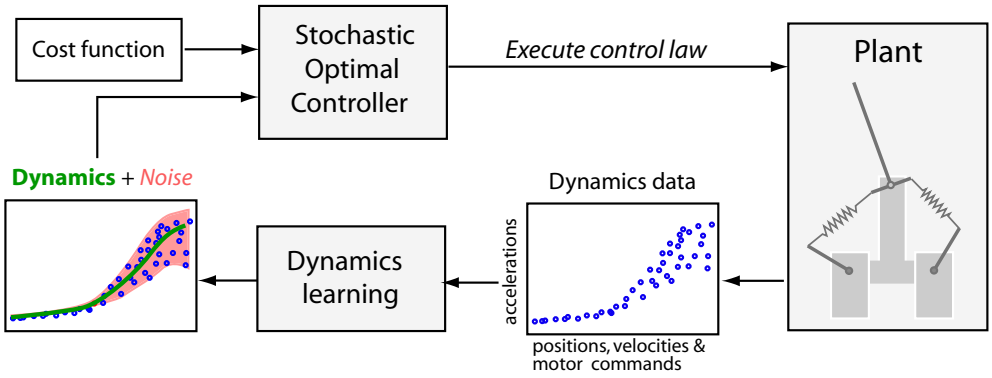


Figure 5.7: Schematic diagram of our proposed combination of stochastic optimal control (SOC) and learning. The dynamics model used in SOC is acquired and constantly updated with data from the plant. The learning algorithm extracts the dynamics as well as stochastic information contained (noise model from confidence intervals). SOC takes into account both measures in the optimisation.

the arm (Fig. 5.4, left). In correspondence to the general dynamics equation (5.11), here the state $\mathbf{x} = \theta_{eq}$ represents the current equilibrium position, $\mathbf{u}$ the applied motor action, and $d\mathbf{x}$ the resulting change in equilibrium position. Therefore the reduced dynamics used here, only depends on the control signals, i.e.,

$$dx = f(\mathbf{u})dt + F(\mathbf{u})d\xi \quad , \quad \xi \sim N(0,1). \tag{5.12}$$

Learning this mapping from data, we can directly account for asymmetries. More interestingly, when we collect data from the perturbed system, we can acquire a model of the arm's kinematic variability as a function of the motor positions.

We use this learned model $\tilde{f}$ in two ways: first, in (slow) position control tasks (Section 5.3.2), and in conjunction with full analytic dynamics models for dynamic reaching tasks (Section 5.3.3).

5.3.2 Energy optimal equilibrium position control

Consider the task of holding the arm at a certain position $\hat{\theta}$, while consuming as little energy as possible. Let us further assume that we have no feedback from the system[4], but that the arm is perturbed randomly. We can state this mathematically as the minimisation of a cost

$$J = \left\langle w_p (f(\mathbf{u}) - \hat{\theta})^2 + |\mathbf{u}|^2 \right\rangle, \tag{5.13}$$

[4] Alternatively, assume the feedback loop is so slow that it is practically unusable.

where w_p is a factor that weights the importance of being at the right position against the energy consumption which for simplicity we model by $|\mathbf{u}|^2$. Taking into account that the motor commands $\mathbf{u}$ are deterministic, and decomposing the expected position error into an error of the mean plus the variance, we can write the expected cost J as

$$J = w_p(\langle f(\mathbf{u})\rangle - \hat{\theta})^2 + w_p\left\langle \left(f(\mathbf{u}) - \langle f(\mathbf{u})\rangle \right)^2 \right\rangle + |\mathbf{u}|^2, \tag{5.14}$$

which based on the LWPR learned model becomes

$$J = w_p(\tilde{f}(\mathbf{u}) - \hat{\theta})^2 + w_p\sigma^2(\mathbf{u}) + |\mathbf{u}|^2. \tag{5.15}$$

Here $\tilde{f}(\mathbf{u})$ and $\sigma(\mathbf{u})$ denote the prediction and the one-standard-deviation based confidence interval of the LWPR model of $f(\mathbf{u})$. The constant w_p represents the importance of the accuracy demand in our task. We then can easily minimise J with respect to $\mathbf{u} = (u_1, u_2)^T$ numerically, taking into account the box constraints $0^\circ \leq u_i \leq 180^\circ$ [5].

5.3.3 Dynamics control with learned stochastic information

Equilibrium position control is ignorant about the dynamics of the arm, that is, going from one desired position to the next might induce swinging movements, which are not damped out actively. Proper dynamics control should take these effect into account and optimise the command sequence accordingly. What follows is a description of how we model the full dynamics of the arm, that is, the combination of the dynamics of the joint and the motors.

The state vector $\mathbf{x}[k]$ of our system at time k consists of the joint angle $x_1[k] = \theta[k]$ and joint velocity $x_2[k] = \dot{\theta}[k]$ as well as 12 *additional* state variables, which represent the command history of the two motors, i.e., the last 6 motor commands that were applied to the system. The state vector therefore is

$$\mathbf{x}[k] = (\theta[k], \dot{\theta}[k], u_1[k-1], \ldots, u_1[k-6], u_2[k-1], \ldots, u_2[k-6])^T, \tag{5.16}$$

where the additional state variables $x_3[k], \ldots, x_8[k]$ for motor 1, and in the same way $x_9[k], \ldots, x_{14}[k]$ for motor 2 are required to represent the FIR filter states of the motor dynamics from (5.10). We can estimate the motor positions $\alpha[k]$ and $\beta[k]$ solely from

[5] For our SEA this optimisation can be performed in real time, i.e., at least 50 times per second, which corresponds to the maximum control frequency of our system ($50Hz$).

these filter states because the FIR coefficients are $h_0 = h_1 = 0$:

$$\alpha[k] = \sum_{j=2}^{7} h_j u_1[k-j+1] = \sum_{j=2}^{7} h_j x_{j+1}[k] \tag{5.17}$$

$$\beta[k] = \sum_{j=2}^{7} h_j u_2[k-j+1] = \sum_{j=2}^{7} h_j x_{j+7}[k] \tag{5.18}$$

Based on the rigid body dynamics from (5.9) we can compute the acceleration from states (i.e. forward dynamics) as

$$\ddot{\theta}[k] = \frac{1}{I_z}(\tau(\alpha[k], \beta[k], \theta[k]) - D\dot{\theta}[k]). \tag{5.19}$$

Therefore "running" the dynamics here means accounting for motor dynamics by shifting the filter states, that is $x_{i+1}[k+1] = x_i[k]$ for $i = 3 \ldots 7$ and $i = 9 \ldots 13$, and then Euler-integrating the velocities and accelerations:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{x}[k] + \Delta t\, \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) \\ &= (\theta[k] + \Delta t\dot{\theta}[k], \dot{\theta}[k] + \Delta t\ddot{\theta}[k], u_1[k], x_3[k], \ldots, x_7[k], u_2[k], x_9[k], \ldots, x_{13}[k])^T. \end{aligned} \tag{5.20}$$

Alternatively, we can drop the time index k and write the dynamics in compact form as

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \left(x_2, \ddot{\theta}(\mathbf{x}), \frac{1}{\Delta t}(u_1 - x_3), \frac{1}{\Delta t}(x_3 - x_4), \ldots, \frac{1}{\Delta t}(u_2 - x_8), \frac{1}{\Delta t}(x_8 - x_9), \ldots \right)^T. \tag{5.21}$$

The gradient of $\ddot{\theta}(\mathbf{x})$ is given by the chain rule, where τ is the short notation for $\tau(\alpha, \beta, \theta)$. Note that $\theta = x_1$, $\dot{\theta} = x_2$, and α and β are calculated from $x_{3\ldots14}$:

$$\nabla_{\mathbf{x}} \ddot{\theta} = \frac{1}{I_z} \left(\frac{\partial \tau}{\partial \theta}, -D, \frac{\partial \tau}{\partial \alpha} h_2, \frac{\partial \tau}{\partial \alpha} h_3, \ldots, \frac{\partial \tau}{\partial \alpha} h_7, \frac{\partial \tau}{\partial \beta} h_2, \frac{\partial \tau}{\partial \beta} h_3, \ldots, \frac{\partial \tau}{\partial \beta} h_7 \right). \tag{5.22}$$

This only shows the second row of the Jacobian $\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})$ and for brevity we omitted the others as they are trivial. The other Jacobian $\nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})$ consists of zeros entries apart from the entry $\frac{1}{\Delta t} = 50$ at indices $(3,1)$ and $(9,2)$.

Since the dynamics of our system is non-linear and high-dimensional, we employ the ILQG method due to its ability to include constraints on the commands. Details about the ILQG algorithm can be found in Section 2.3.1 of Chapter 2.

The usual ILQG formulation is based on an analytically given cost function (deterministic) and a stochastic dynamics function. Here we use a deterministic dynamics (with the idealised analytic model) and we propose a cost function that takes stochastic information into account.

$$c(\mathbf{x}, \mathbf{u}) = w_p (x_1 - \hat{\theta})^2 + w_v x_2^2 + w_e |\mathbf{u}|^2 + w_d \left((u_1 - x_3)^2 + (u_2 - x_9)^2 \right) + w_p \sigma^2(\mathbf{u}) \tag{5.23}$$

All quantities in (5.23) (also possibly the pre-factors) are time-dependent, but we have dropped the time indices for notational simplicity. As before w_p governs the accuracy demand. In addition, a stability term w_v governs the importance of having zero velocity and w_e penalises energy consumption at the level of the springs. The weighting factor w_d penalises changes in motor commands and therefore energy consumption at the level of the servomotor. The last term includes the learned uncertainty in our equilibrium positions, which is here also scaled by w_p. This is well justified because for example for a reaching task, the arm will end up with the servo motors in a position such that the arm's equilibrium position is the desired position $\hat{\theta}$, and we have learned from data how much perturbations we can expect at any such configuration. The same holds true for slow tracking tasks, where the servos will be moved such that the equilibrium positions track the desired trajectory.

5.4 Results

In this section we present results from the optimal control model applied to the hardware described earlier in Section 5.2. We first highlight the adaptation capabilities of this framework experimentally and then show how the learned stochastic information leads to an improved control strategy over solutions obtained without stochastic information. More specifically the new model achieves higher positional accuracy by varying impedance of the arm through motor co-contraction. We study position holding, trajectory tracking and target reaching tasks.

5.4.1 Experiment 1: Adaptation towards a systematic change in the system

An advantage of the learned dynamics paradigm is that it allows to account for systematic changes without prior knowledge of the shape or source of the perturbation. To demonstrate such an adaptation scenario we setup a systematic change in the hardware by replacing the left spring, between motor 1 and the joint (i.e., between points A and C in Fig. 5.3), with one that has a lower, "unknown" spring constant. The aim is to hold a certain equilibrium position using the energy optimal position controller described in Section 5.3.2. Expectedly the prediction about the equilibrium points (i.e., $\tilde{f}(\mathbf{u})$) does not match the real changed system properties. Next, we demonstrate how the system can adapt online and increase the performance trial by trial. We specified a target tra-

jectory that is a linear interpolation of 200 steps between start position $\theta_0 = -30^\circ$ and target position $\hat{\theta} = 30^\circ$. We tracked this trajectory by recomputing the equilibrium positions, i.e., by minimising (5.15) at a rate of $50Hz$. At the same time we updated $\tilde{f}(\mathbf{u})$ during reaching. Due to the nature of local learning algorithms $\tilde{f}$ is only updated in the neighbourhood of the current trajectory and therefore shows limited generalisation. To account for this, after each trial, we additionally updated the model with 400 training data points, collected from a 20-by-20 grid of the motor's range $u_1 = u_2 = [0^\circ, 180^\circ]$. Fig. 5.8 depicts the outcome of this adaptation experiment. One can observe that the controller initially (lighter lines) fails to track the desired trajectory (red). However there is significant improvement between each trial, especially between trials 1 to 5. After about 9 trials the internal model has been updated and manages to track the desired trajectory well (up to the hardware's level of precision). A look at the equilibrium position predictions in Fig. 5.9 confirms that the the systematic shift has been successfully learned, which is visible by the asymmetric shape. Analysing the motor commands (Fig. 5.8, right) shows that the optimal controller, for all trials, chooses the motor commands with virtually no co-contraction. This is a sensible choice as it would contradict the minimum energy cost function that we have specified.

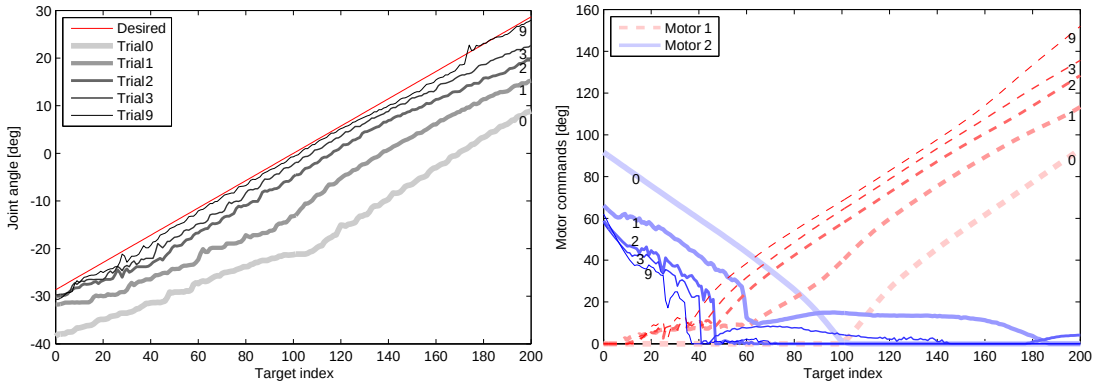


Figure 5.8: Visualisation of the adaptation process. Left: Desired (red) and observed arm positions. Right: Motor commands for the corresponding trials. Darker and thinner lines indicate later stages of learning.

5.4.2 The role of stochastic information for impedance control

Because co-contraction and energy consumption are opposing properties our controller will hardly make use of the redundant degree of freedom in the actuation. Even though minimum energy optimal control in an antagonistic system seems to be "unable to

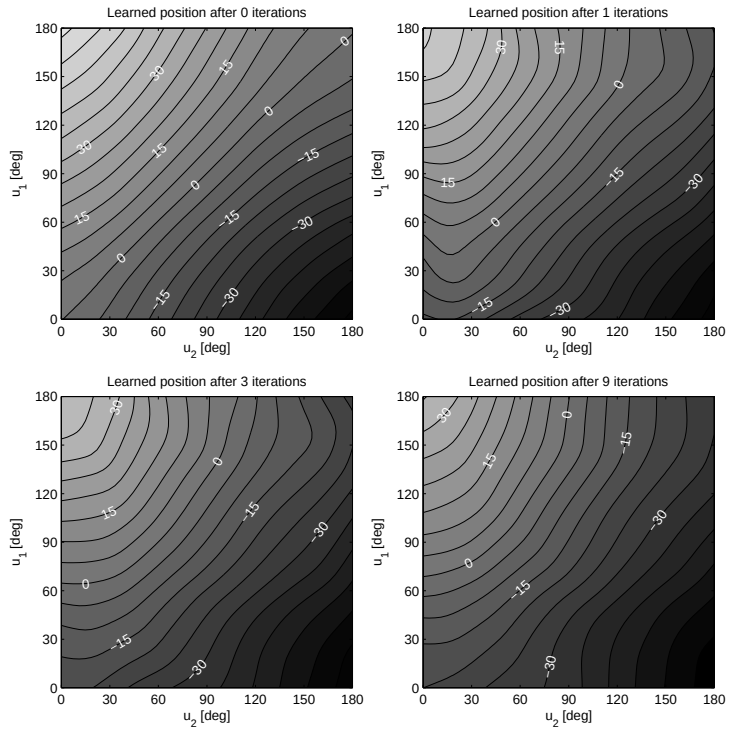


Figure 5.9: Learned position models during the adaptation process. The white numbers represent the equilibrium point positions.

co-contract" it remains our favourite choice of performance index as it also implies compliant movement and as it follows the biological motivation. "But when should the optimal controller co-contract?" If we consider the stochastic information that would arise from a task involving random perturbations we can see that the produced stochasticity holds valuable information about the stability of the system[6]. If the uncertainty can be reduced by co-contracting it will be reflected in the data, i.e., in the LWPR confidence bounds. Therefore the answer to the previous question is that, given we want to achieve high task accuracy, the controller should co-contract whenever it can reduce the expected noise/stochasticity in the system (weighted with the accuracy demand).

Suppose our system experiences some form of small random perturbations during control. In the hardware we realise such a scenario by adding a perturbation motor at the end of the arm, which mimics for example a drilling tool (panel "a" in Fig. 5.5). The perturbation on the arm is produced by alternating the servo motor positions quickly every $200ms$ from 40° to -40°. The inertia of the additional weight then produces deflections on the arm from the current equilibrium position. With these

[6]Stability here refers to the desired equilibrium position.

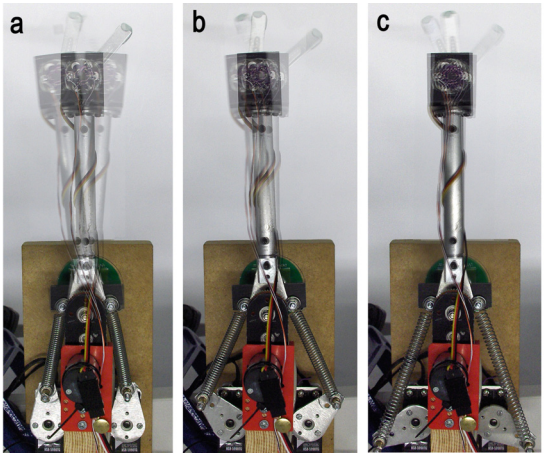


Figure 5.10: Motion traces of our SEA hardware around $\theta = 0^\circ$. The perturbation motor causes different deflections depending on the co-contraction levels: (a) $u_1 = u_2 = 0^\circ$, (b) $u_1 = u_2 = 45^\circ$, (c) $u_1 = u_2 = 120^\circ$.

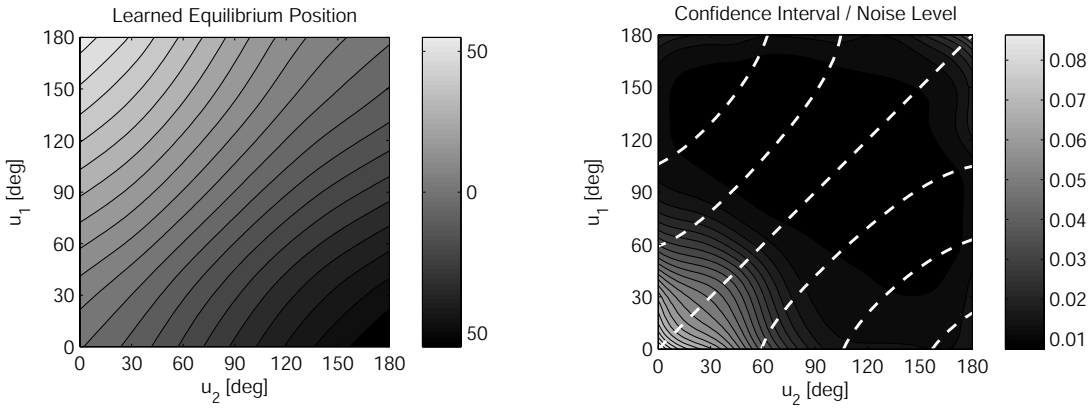


Figure 5.11: Left: Learned equilibrium position as a function of the motor positions (in degrees), with contour lines spaced at 5 degree intervals. Right: "Noise landscape", i.e., stochastic information given by the heteroscedastic confidence intervals of LWPR.

perturbations we collected new training data and updated the existing LWPR model $\tilde{f}$. The collected data reveals that the arm stabilises in regions with higher co-contraction, where the stiffness is higher. This behaviour is visualised in Fig. 5.10, which shows motion traces around $\theta = 0^\circ$ due to the perturbation motor for different co-contraction levels. This information is contained in the learned confidence bounds (Fig. 5.11) and therefore the optimal controller effectively tries to find the trade-off between accuracy and energy consumption.

5.4.3 Experiment 2: Impedance control for varying accuracy demands

Based on the learned LWPR model $\tilde{f}$ from the previous section we can demonstrate the improved control behaviour of the stochastic optimisation with emerging impedance control. We formulate a task to hold the arm at the fixed position $\hat{\theta} = 15^\circ$ and $\hat{\theta} = 0^\circ$ respectively. While minimising for the cost function in (5.15), we continuously and slowly increased the position penalty within the range $w_p = [10^{-2}, 10^5]$. The left column in Fig. 5.12 summarises the results we discuss next: At $w_p = 10^{-2}$ to approximately $w_p = 10^0$ the optimisation neglects position accuracy and minimises mainly for energy, i.e., $u_1 = u_2 = 0$. The actual joint positions, because of the perturbations, oscillate around the mean $\theta = 0^\circ$ as indicated by the shaded area. Between $w_p = 10^0$ and $w_p = 10^2$ the position constraint starts to "catch up" with the energy constraint; a shift in the mean position towards $\hat{\theta}$ can be observed. At about $w_p = 5 * 10^1$ the variance in the positions increases as the periodic perturbation seems to coincide with the resonance frequency of the system. For $w_p > 10^2$ the stochastic information is weighted sufficiently such that the optimal solution increases the co-contraction and that the accuracy improves further. If in contrast we run the same experiment while ignoring the stochastic part in the cost function, i.e., we minimize for the deterministic cost function $J = w_p(\tilde{f}(\mathbf{u}) - \hat{\theta})^2 + |\mathbf{u}|^2$ only, we can can see (Fig. 5.13) that the system does expectedly not co-contract and hardly improves performance accuracy.

5.4.4 Experiment 3: ILQG reaching task with a stochastic cost function

For certain tasks, such as quick target reaching or faster tracking of trajectories, the system dynamics based on equilibrium points $\theta = f(\mathbf{u})$ may not be sufficient as it contains no information about the velocities and accelerations of the system. Next, we assume a full forward dynamics description of our system as identified in (5.16), where the state consists of joint angles, joint velocities, and 12 motor states.

The task is to start at position $\theta_0 = 0^\circ$ and reach towards the target $\hat{\theta} = 0.3rad$ $(= 17.18^\circ)$. The reaching movement duration is fixed at 2 seconds, which corresponds to $T = 100$ discretised time steps at the hardware's operation rate of $50Hz$. This task can be formalised based on the cost function (5.23) by setting the weighting terms as follows: The time dependent position penalty is a monotonically increasing linear

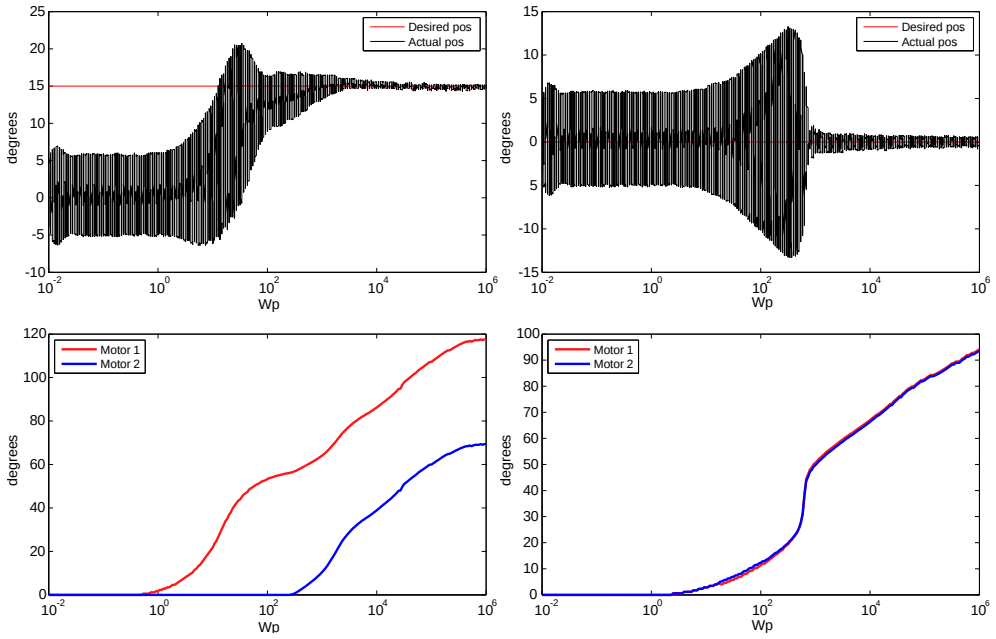


Figure 5.12: Experiment with increasing position penalty w_p for two targets. Left plot column: $\hat{\theta} = 15°$; right plot column: $\hat{\theta} = 0°$. The plots show the desired vs. measured position and corresponding motor commands as a function of the accuracy demand (pre-factor w_p). The shaded area results from the perturbation motor.

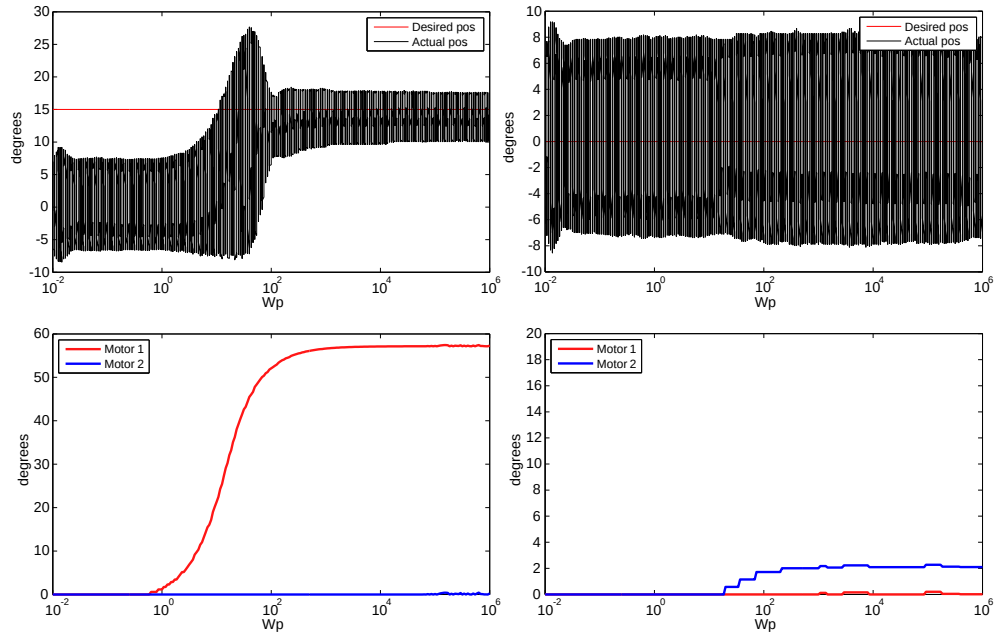


Figure 5.13: The same experiment as in Fig. 5.12 where the stochastic information was not incorporated into the optimisation.

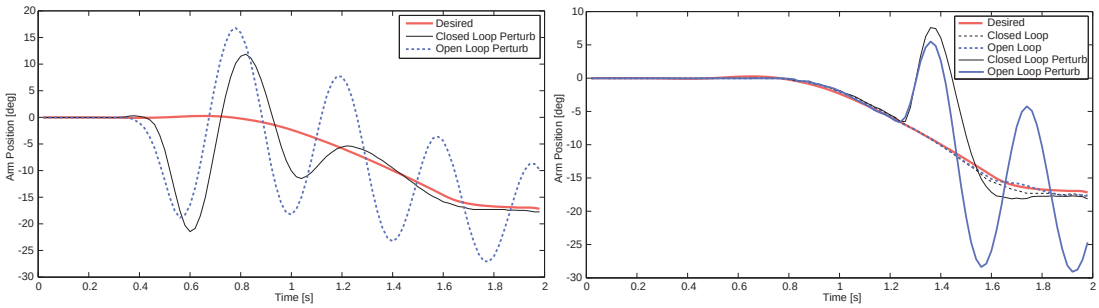


Figure 5.14: ILQG reaching task without stochastic information used in open loop and closed loop control. We perturbed the arm by hitting it once after 0.4s (left plot) and 1.2s (right plot), respectively. The dashed lines in the right plot represent unperturbed trials (open loop and closed loop).

interpolation of 100 steps, i.e., $w_p[t] = [0.1, 0.2, ..., 10]$. The penalty for zero endpoint velocity was set to $w_v[t] = 0$ for $0 < t < 80$ and $w_v[t] = 1$ for $t \geq 80$. The energy penalties are assumed constant $w_e = w_d = 1$ during the whole movement.

By using ILQG, we then compute an optimal control sequence $\bar{\mathbf{u}}$ with the corresponding desired trajectory $\bar{\mathbf{x}}$ and a feedback control law $\mathbf{L}$. Fig. 5.14 depicts the reaching performance of the ILQG trajectory, applied in *open loop* mode and in *closed loop* mode (i.e, using feedback law $\mathbf{L}$), where the robot has been perturbed by a manual push. The closed loop scheme successfully corrects the perturbation and reaches the target while the open loop controller oscillates and fails to reach the target. This experiment highlights the benefits of a closed loop optimisation which can, by incorporating the full dynamics description of the system, account for such perturbations. However the ability to correct perturbations is limited by the hardware control bandwidth (i.e., slow servo motor dynamics and $50Hz$ control board frequency). If the system also suffers from feedback or motor delays the correction ability is limited and for example accounting for vibrations or noise[7] is difficult to achieve using the feedback signals only. For such stochastic perturbations, impedance control can improve performance as it changes the mechanical properties of the system in a feed-forward manner, i.e., it reduces the effects of the perturbations in the first place.

To realise such a scenario, we defined a tracking task that starts at the zero position then moves away and back again along a sinusoidal curve during 2.5 seconds. The cost function parameters for this task are defined as follows: The time dependent position penalty is $w_p[t] = [50, 100, \ldots, 4000]$ for t=0 $< t < 80$ and $w_p[t] = 4000$ for $t \geq 80$.

[7] or any other high frequency perturbation.

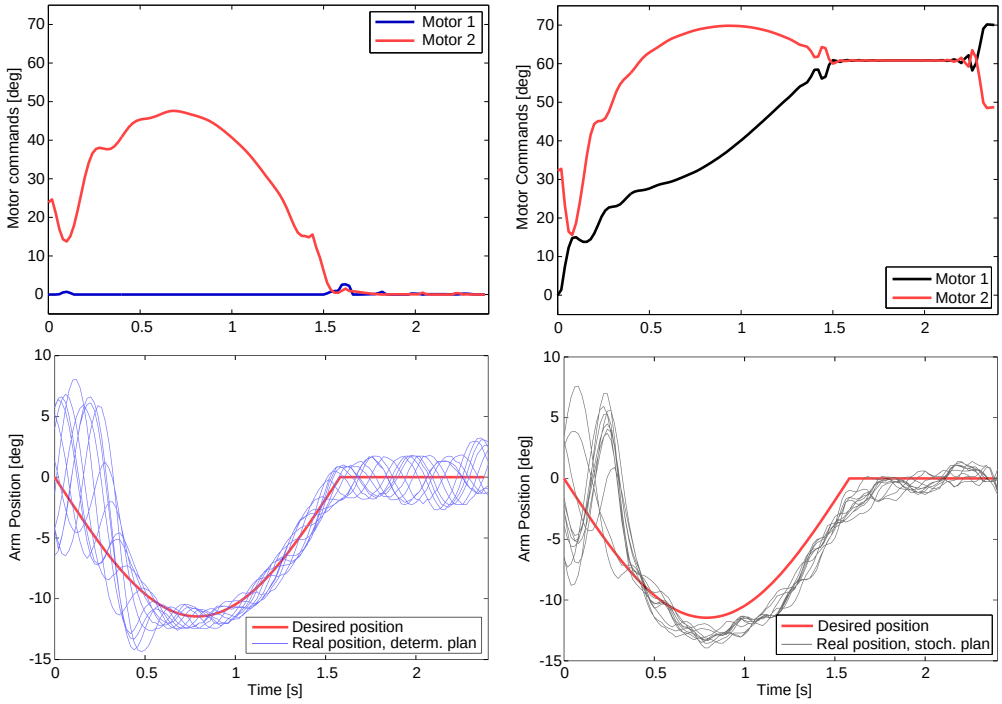


Figure 5.15: 20 trials of ILQG for a tracking task of 2.5 seconds. Left column: Deterministic optimisation exhibits no co-contraction. Right column: 20 trials of ILQG using stochastic information in the cost function. The system co-contracts as the accuracy demands increase.

The endpoint velocity term is $w_v[t] = 0$ for $0 < t < 80$ and $w_v[t] = 10$ for $t \geq 80$. The energy penalties are held constant, i.e., $w_e = w_d = 1$.

As before, we observe the benefits of using stochastic information for optimisation compared to a deterministic optimisation (not using the LWPR confidence bounds). After computing the optimal control using ILQG we ran the optimal feedback control law consecutively 20 times in each condition, i.e, with and without stochastic optimisation. Please note that the perturbation motor is switched on at all times. Fig. 5.15 summarises the results: expectedly the stochastic information in the cost function induces a co-activation for the reaching task, which shows generally better performance in terms of reduced variability of the trajectories. Evaluating the movement variability where the accuracy weight is maximal, i.e., for $t > 80$, the standard deviation of the trajectories is significantly lower with $\sigma_{stoch} = 0.55^\circ$ for the stochastic optimisation compared to the deterministic optimisation with $\sigma_{det} = 1.38^\circ$. A detailed look at the bottom right plot in Fig. 5.15 reveals a minor shift in the recorded trajectory compared to the planned one from the analytic model. We attribute this error to imprecisions

in the hardware, i.e., tiny asymmetries which are not included in the analytic model. In the case of higher co-contraction small manufacturing errors and an increased joint friction lead to deviations towards the idealised analytic model predictions. Indeed the learned dynamics model can account for these asymmetries as can be seen in Fig. 5.11, (left) along the equilibrium position $\theta = 0^\circ$, i.e., the line $u_1 = u_2$ is slightly skewed.

5.5 Discussion

In this chapter we have presented a stochastic optimal control model for antagonistically actuated systems. We proposed to learn, both, the *dynamics* as well as the *stochastic information* of the controlled system from sensorimotor feedback of the plant. This control architecture can account for a systematic change in the system properties (Experiment 1) and furthermore is able, by incorporating the heteroscedastic prediction variances into the optimisation, to compensate for stochastic perturbations that were induced to the plant. Doing so, our control model demonstrated significantly better accuracy performance than the deterministic optimisation in both, energy optimal equilibrium point control (Experiment 2) and energy optimal reaching using dynamics optimisation (Experiment 3). The improved behaviour was achieved by co-activating antagonistic motors, i.e., by using the redundant degree of freedom in the system based on the first principles of optimality. The presented results demonstrate that this is a viable optimal control strategy for real hardware systems that exhibit hard to model system properties (e.g., asymmetries, systematic changes) as well as stochastic characteristics (e.g., using a power tool) that may be unknown a priory.

An advantage of the presented control architecture is that motor co-activation (or impedance) does not need to be specified explicitly as a control variable but that it emerges from the actual learned stochasticity within the system (scaled with the specified accuracy demands of the task). Therefore co-activation (i.e., higher impedance), since it is energetically expensive, will only be applied if it actually is beneficial for the accuracy of the task.

Exploiting stochasticity in wider domains

The methodology we suggest for optimal exploitation of sensorimotor stochasticity through learning is a generic principle that goes beyond applications to impedance modulation of antagonistic systems but can be generalised to deal with any kind of

control or state dependent uncertainties. For example, if we want to control a robot arm that suffers from poor repeatability in certain joint angles or in a particular range of velocities, this would be visible in the noise landscape (given one has learned state dependent stochastic dynamics) and consequently those regions would be "avoided" by the optimal controller. In this context, the source of the stochasticity is irrelevant for the learner and therefore, it could arise from internal (i.e., noise in the motor), as well as external (i.e., power tool) sources. However, the stochastic system properties must, to a certain degree, be *stationary in time* such that the learner can acquire enough information about the noise landscape.

Biological relevance

As mentioned in the introduction biological systems are often used as a benchmark for the control of artificial systems. In this work not only the antagonistic hardware but also the actual control architecture is motivated by biological principles. Optimality approaches have been a very fruitful line of research (Todorov, 2004; Scott, 2004; Shadmehr and Krakauer, 2008) and its combination with a learning paradigm (Mitrovic et al., 2008a) is biologically well justified a priory, since the sensorimotor system can be seen as the product of an optimisation process, (i.e., evolution, development, learning, adaptation) that constantly learns to improve its behavioural performance (Li, 2006). Indeed, internal models play a key role in efficient human motor control (Davidson and Wolpert, 2005) and it has been suggested that the motor system forms an internal forward dynamics model to compensate for delays, uncertainty of sensory feedback, and environmental changes in a predictive fashion (Shadmehr and Wise, 2005; Wolpert et al., 1995; Kawato, 1999). Notably a *learned* optimal trade-off between energy consumption, accuracy and impedance has been repeatedly observed in human impedance control studies (Burdet et al., 2001; Franklin et al., 2008). More specifically the amount of impedance modulation in humans, seems to be governed by some measure of uncertainty, which could arise from internal (e.g., motor noise) or external (e.g., tools) sources (Selen et al., 2009).

In the computational model presented here these uncertainties are represented by the heteroscedastic confidence bounds of LWPR and integrated into the optimisation process via the performance index (i.e, cost function). Such an assumption is biologically plausible, since humans have the ability to learn not only the dynamics but also the stochastic characteristics of tasks, in order to optimally learn the control of a complex task (Chhabra and Jacobs, 2006; Selen et al., 2009).

Hardware limitations & scalability

This work represents an initial attempt to modulate impedance on a real antagonistic system in a principled fashion. The proposed SEA has been primarily designed to perform as a "proof of concept" of our control method on a real system. Specifically we can identify several limitations of our system that need further investigation in the future.

First, the stiffness range of the system is fairly low as spring nonlinearities are achieved by a geometric effect of changing the moment arms. There are other, mechanically sophisticated, SEA designs with large stiffness ranges, e.g., (Grebenstein and van der Smagt, 2008; van Ham et al., 2009), which also could serve as attractive implementation platforms for our algorithm. Specifically the MACCEPA design (van Ham et al., 2007) is very appealing as it is technically simple and offers a larger stiffness range; however parallels to biologically realistic implementations are less obvious in this design, as the system is not antagonistically actuated. The fact that we were able to obtain a significant increase in co-contraction from the learned stochastic information even for a hardware with very low stiffness range is promising, indicating good resolution capabilities of the localised variance measure in LWPR. Since the amount of co-contraction in OFC-LD depends on the hardware that is used, one would consequently expect that a system with larger stiffness range would necessitate less co-contraction, which could potentially become challenging for OFC-LD. It is important to note though that the co-contraction also depend on the chosen weighting parameters in the cost function. Therefore those parameters would need additional manual tuning adapted to the specific hardware and task at hand in order for OFC-LD to succeed.

Second, the relatively slow control loop ($50Hz$) causes controllability issues (i.e., slow feedback) and furthermore turned out to be sensitive to numerical integration errors within ILQG. While these numerical issues have not caused problems in an analytic dynamics formulation (Experiment 3), they turned out to be critical when we run ILQG using the full learned forward dynamics $\tilde{f}(\mathbf{x},\mathbf{u})$. Under these conditions, ILQG most of the time does not converge to a reasonable solution. A potential route of improvement could be a combination of LWPR learning with an analytic model. Instead of "ignoring" valuable knowledge about the system given in analytic form, one could focus on learning an *error model* only, i.e., aspects of the dynamics that are not described by the analytic model.

Third, the transfer of optimal controls from simulation to the real hardware has

proven to be very challenging. Currently we are computing ILQG solutions for a fixed time horizon and later applying them to the SEA. For slower movements this approach produces satisfying accuracy. In experiment 3 we have "enforced" slower and smooth movements by formulating an appropriate time-dependent cost function. However for movements with higher frequency the situation is more difficult: Errors accumulate on the hardware over the course of the trajectory, since the feedback loop for corrections is very slow. This leads to solutions that differ significantly from the pre-planned optimal solution. A potential route to resolve this problem is to use a model predictive control approach in which the optimal solutions are re-computed during control with current states of the plant as initial states. However, this approach requires computationally efficient re-computations of the optimal control law, which may be hard to obtain, especially for systems with higher dimensionality.

At last, our experiments were carried out on a low dimensional system with a single joint and two motors. Implementations on systems with higher dimensionality, however, are still very challenging as the construction of antagonistic robots is non-trivial and the availability of large degrees of freedom systems is very limited. In fact to date we are unaware of any successful large DoF implementation of a variable impedance manipulator based on SEA. Manipulators like the *Meka*[8] or *Biorob*[9] arm have fixed compliance only and the *Kuka LWR* varies impedance actively. The real challenge when building multi-joint SEA systems is to keep the total arm weight low (double amount of motors) and to come up with design that can be miniaturised in an appropriate manner.

Besides the hardware limitations high dimensional systems impose serious computational challenges on optimal control methods as well as on machine learning techniques. Here even a single joint system with two muscles required 14 states to be able to model the system dynamics appropriately. Scaling this to 7 DoF would lead to 98 states that need to be modelled. Despite these limitations we believe that the study of impedance control based on stochastic sensorimotor feedback is a promising route of research for both, robotic and biological systems.

[8] www.mekabot.com
[9] www.biorob.de

Chapter 6

A computational model of human limb impedance control

In the previous chapters we have focused on problems related to the OFC of anthropomorphic robotic systems both in simulation and on real hardware. As mentioned in Chapter 2, Section 2.2.2 optimality principles have been very successful in modelling biological movement systems. In this chapter we show how our OFC-LD framework can be employed to predict and interpret biological motor control patterns. More specifically we study impedance control in human limb reaching tasks. This is a particularly interesting control problem as it is not obvious a priori how one can combine the principles of muscle co-contraction and energy optimality in a principled fashion.

6.1 Introduction

Humans and other biological systems have excellent capabilities in performing fast and complicated control tasks in spite of large sensorimotor delays, internal noise or external perturbations. By co-activating antagonistic muscle pairs, the CNS manages to change the mechanical properties (i.e., joint impedance) of limbs in response to specific task requirements; this is commonly referred to as *impedance control* (Hogan, 1984). A significant benefit of modulating the joint impedance is that the changes apply instantaneously to the system. Impedance control has been explained as an effective strategy of the CNS to cope with kinematic variability due to neuromuscular noise and environmental disturbances. Understanding how the CNS realises impedance control is of central interest in biological motor control as well as in the control theory of artificial systems. Computational models provide a very useful tool to understand the

underlying motor functions observed in biological systems. Generally one aims to formulate models that can predict wide range of observed phenomena and that are biologically well motivated. In this chapter we investigate how impedance control can be modelled in a generally valid fashion using the optimality principles with learned dynamics developed in the earlier chapters.

We start our discussion with an intuitive example: Suppose you are holding an umbrella in a stable upright position on a rainy day. This is an effortless task, however if suddenly a seemingly random wind gust perturbs the umbrella, you will typically stiffen up your arm trying to reduce the effects of the "unpredictable" perturbation. It is well established that the *central nervous system (CNS)* manages to change the mechanical properties (i.e., joint impedance) of limbs by co-activating antagonistic muscle pairs in response to specific task requirements. This is commonly referred to as impedance control, which has been explained as an effective strategy of the nervous system to cope with kinematic variability due to neuromuscular noise and environmental disturbances. Coming back to our umbrella example: If over time you realise the wind keeps blowing from the same direction, you expectedly will become more certain about the wind's destabilising effect on your arm and you will gradually reduce the stiffness and you will possibly try to place the umbrella in a new stable position. This simple example shows intuitively how co-activation is linked to uncertainties that you may experience in your limb dynamics, and the main objective in this work is to develop a computational model that unifies the concepts of learning, uncertainty and optimality in order to understand impedance control in a principled fashion.

A large body of experimental work has investigated the motor learning processes in tasks under changing dynamics conditions (Burdet et al., 2001; Milner and Franklin, 2005; Franklin et al., 2008), revealing that subjects generously make use of impedance control to counteract destabilising external *force fields (FF)*. Indeed impedance modulation appears to be, to some extent, governed by a preservation of metabolic cost (Burdet et al., 2001) in that subjects do not just naively stiffen up their limbs but rather *learn* the optimal mechanical impedance by predictively controlling the magnitude, shape, and orientation of the endpoint stiffness in the direction of the instability. In the early stage of dynamics learning, humans tend to increase co-contraction and as learning progresses in consecutive reaching trials, a reduction in co-contraction along with a simultaneous reduction of the reaching errors made can be observed (Franklin et al., 2008). These learning effects are stronger in stable FF (i.e., velocity-dependent) compared to unstable FF (i.e., divergent), which suggests that impedance control is

connected to the learning process with internal dynamics models and that the CNS employs co-activation to increase task accuracy in early stages of learning, when the internal model is not fully formed yet (Thoroughman and Shadmehr, 1999; Wang et al., 2001).

Notably limb impedance is not only controlled during adaptation but also in tasks under stationary dynamics conditions. Studies in single and multi-joint limb reaching movements revealed that stiffness is increased with faster movements (Bennett, 1993; Suzuki et al., 2001) as well as with higher positional accuracy demands (Gribble et al., 2003; Lametti et al., 2007; Wong et al., 2009). Under such conditions higher impedance is thought to reduce the detrimental effects of neuromotor noise (Selen, 2007), which exhibits large control signal dependencies (Harris and Wolpert, 1998). Similar to our umbrella example, in the stationary case the impedance can be linked to uncertainty, which here however arises from internal sources, triggering an increase of co-activation levels.

Many proposed computational models have focused on the biomechanical aspects of impedance control (Tee et al., 2004; Burdet et al., 2006) or have provided ways to reproduce accurately observed co-activation patterns for specific experiments (Franklin et al., 2008; McIntyre et al., 1996). While such models without doubt are important for the phenomenological understanding of impedance control, they do not provide principled insights about the origins of a wider range of phenomena, i.e., they cannot predict impedance control during both, stationary and adaptation experiments. Furthermore it is not clear how impedance control can be formalised within the framework of optimal control, which has been immensely successful in the study of neural motor control. More specifically impedance control (i.e., muscle co-contraction) and energy preservation seem to be opposing properties and it has not been shown yet from a computational perspective how these properties can be unified in a single optimality framework. Referring back to Section 4.3.3 of Chapter 4 the experimental results of ILQG on antagonistic systems revealed that the minimum energy cost function leads to optimal solutions that exhibit no co-contraction whatsoever both in stationary and non-stationary conditions. Therefore OFC, while being able to reproduce kinematic pattern well, fails to reproduce co-contraction patterns observed in humans, which raises questions about the “suitability” of the minimum energy cost function that is most widely used.

Here we develop a new computational theory for impedance control which explains muscle co-activation in human arm reaching tasks as an emergent mechanism from the

first principles of optimality. Our model is formalised within the powerful theory of stochastic OFC. Unlike previous OFC formulations that require a closed analytical form of the plant dynamics model, we postulate, as discussed in earlier chapters, that this internal dynamics model is acquired as a motor learning process based on continuous sensorimotor feedback. From a computational perspective this approach offers three significant improvements over state-of-the-art OFC models relevant for neuromotor control:

1. We can model adaptation processes due to modified dynamics conditions from an optimality viewpoint, without making prior assumptions about the source or nature of the novel dynamics.

2. Dynamics learning further provides us with means to model prediction uncertainty based on experienced stochastic movement data; we provide evidence that, in conjunction with an appropriate antagonistic arm and motor variability model, impedance control emerges from a stochastic optimisation process that minimises these prediction uncertainties of the learned internal model.

3. By formalising impedance control within the theory of stochastic OFC, we overcome the fundamental inability of energy based optimisation methods to model co-contraction. Notably, in our model, co-contraction is achieved without changing the standard energy based cost function since the uncertainty information is contained in the learned internal dynamics function as a stochastic term. Therefore the trade-off between energy preservation and co-contraction is primarily governed by the learned uncertainty of the limb system and by the accuracy demands of the task at hand.

We verify our model by comparing its predictions with two classes of published impedance control experiments: Firstly, stationary reaching experiments where accuracy or velocity constraints are modulated and secondly, tasks involving adaptation towards external FF. The results from single-joint elbow motion show, as predicted by the theory, that we can replicate many well-known impedance control phenomena from the first principles of optimality, and that the proposed minimum-uncertainty approach not only describes impedance control patterns but also conceptually explains the origins of co-activation in volitional human reaching tasks.

6.2 A motor control model based on learning and optimality

Stochastic OFC has been shown to be a powerful theory for interpreting biological motor control (Todorov and Jordan, 2002; Todorov, 2004; Scott, 2004; Lockhart and Ting, 2007). For the study of impedance control, optimality principles are well motivated given the fact that humans showed energy and task optimal impedance modulation (Burdet et al., 2001). Formulating a reaching task in this framework requires a definition of a performance index (i.e., cost-function) to minimise for, typically including reaching error, end-point stability and energy expenditure. Other proposed cost functions often describe kinematic parameters only (Flash and Hogan, 1985) or dynamics parameters based on joint torques (Uno et al., 1989), both of which do not allow a study of joint impedance at the level of muscle activations.

To study impedance control it is essential to be able to define the dynamics on a muscle level. Implementations of OFC make simplifying assumptions (linear dynamics models and quadratic cost functions) due to the computational limitations OFC imposes. Linear dynamics models do not explain the full underlying dynamics of biological systems, since these typically are highly nonlinear. In order to build biologically plausible motor control models one should use system descriptions that are as accurate as possible while being computationally tractable.

In addition to the cost function, an internal model needs to be identified, which represents the (possibly stochastic) dynamics function of the controlled arm. Indeed, internal models play a key role in efficient human motor control (Davidson and Wolpert, 2005) and it has been suggested that the motor system forms an internal forward dynamics model to compensate for delays, uncertainty of sensory feedback, and environmental changes in a predictive fashion (Wolpert et al., 1995; Kawato, 1999). Following this motivation, we build our internal dynamics model based on a motor learning process from continuous sensorimotor plant feedback. Such a learned internal model offers two advantages: First, it allows to model adaptation processes by updating the internal model with newly available training data from the limbs. Second, this training data contains valuable stochastic information about the dynamics and uncertainties therein. As motivated in the introduction, the uncertainty could originate from both internal sources (e.g., motor noise) and from environmental changes during adaptation tasks. The crucial point here is that learning a stochastic internal model enables a *unified treatment* of all the different types of perturbations, the effects of which are visible

as predictive uncertainties.

By incorporating this model into the optimal control framework (Fig. 6.1), we can formulate *OFC with learned dynamics (OFC-LD)* which, besides minimising energy consumption and end point error, incorporates the prediction uncertainties into the optimisation process (Todorov, 2005). Such an assumption is appropriate since humans have the ability to learn not only the dynamics but also the stochastic characteristics of tasks, in order to optimally learn the control of a complex task (Chhabra and Jacobs, 2006; Selen et al., 2009). Algorithmically OFC-LD relies on a supervised learning method that has the capability to learn heteroscedastic (i.e., localised) variances within the state-action space of the arm (see Chapter 4, Section 4.2.2).

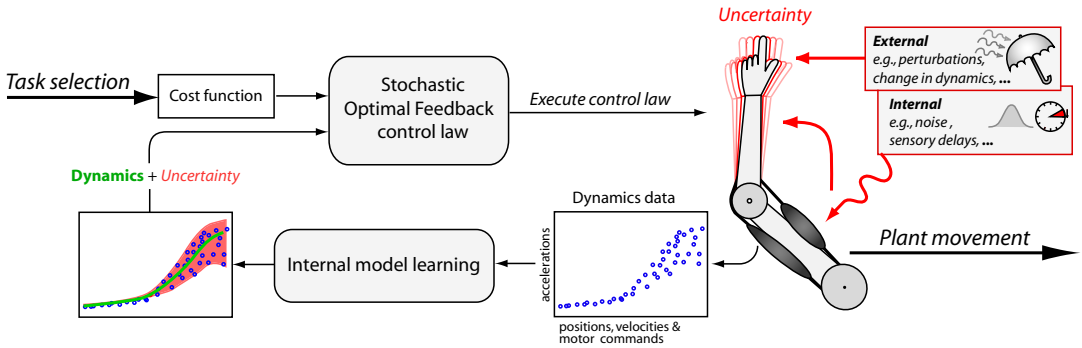


Figure 6.1: Schematic representation of our OFC-LD approach in the context of biological motor control. The optimal controller requires a cost function, which here encodes for reaching time, endpoint accuracy, endpoint velocity (i.e., stability), and energy efficiency. Further a forward dynamics function is required, which in OFC-LD is learned from plant feedback directly. This learned internal dynamics function not only allows us to model changes in the plant dynamics (i.e., adaptation) but also encodes for the uncertainty in the dynamics data. The uncertainty itself, visible as kinematic variability in the plant, can originate from different sources, which we here classify into external sources and internal sources of uncertainty. Most notably OFC-LD identifies the uncertainty directly from the dynamics data not making prior assumptions about its source or shape.

6.3 Modelling plausible kinematic variability

The human sensorimotor system exhibits highly stochastic characteristics due to various cellular and behavioural sources of variability (Faisal et al., 2008) and a complete

motor control theory must contend with the detrimental effects of *signal dependent noise (SDN)* on task performance. Generally speaking SDN in the motor system leads to kinematic variability in the arm motion and in attempts to incorporate this stochastic information into the optimisation process, earlier models assumed, what we here refer to as *standard SDN*, which monotonically increased with the control signal (Harris and Wolpert, 1998; Jones et al., 2002). Those models have been successful in reproducing important psychophysical findings (Haruno and Wolpert, 2005; Li, 2006), however in essence they simply scale the resulting *kinematic variability (KV)* with the control signal's magnitude and ignore the underlying noise-impedance characteristics of the musculoskeletal system (Osu et al., 2004; Selen et al., 2005). Consequently such methods, like all energy based methods, are only concerned with finding the lowest muscle activation possible, penalising large activations and disallowing co-contraction. Generally we define co-contraction as the *minimum of two antagonistic muscle signals* (Thoroughman and Shadmehr, 1999). Experimental evidence suggests though that the CNS "sacrifices" energetic costs by co-contracting under certain conditions to increase impedance. But how can we model plausible kinematic variability arising from SDN?

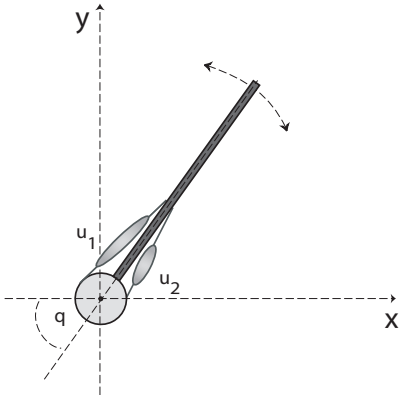
Kinematic variability in human motion originates from a number of inevitable sources of internal force fluctuations (Selen, 2007; Faisal et al., 2008). SDN (Jones et al., 2002) as well as joint impedance (Osu and Gomi, 1999) increase monotonically with the level of muscle co-activation leading to the paradoxical situation that muscles are the source of force fluctuation and at the same time the means to suppress its effect by increasing joint impedance (Osu et al., 2004; Selen et al., 2005): Since SDN propagates through the muscle dynamics and impedance of the arm leading to kinematic variability, impedance can be changed to modulate the kinematic effects of the motor noise. Consequently, even though higher impedance implies higher co-activation and thus larger SDN levels in the muscles, in humans it leads to smaller kinematic variability (Osu et al., 2004).

In order to account for this important property of human limbs, detailed muscular simulation models (Selen et al., 2005) have been proposed that showed that muscle co-contraction has a similar effect to a low-pass filter to the kinematic variability. This is achieved by a relatively complex motor unit pool model of parallel Hill-type motor units that model realistic motor variability. In this work we are primarily interested in the computational aspects of impedance control and highly complex dynamics models are not desired as they are known to impose severe computational challenges on existing optimal control methods. In order to avoid a highly complex dynamics model

we alternatively propose to increase the realism of our arm model by imposing the kinematic variability based on physiological observations, i.e., that the kinematic variability is reduced for more highly co-contracted activation patterns. This stochastic limb model is described in the next section.

6.3.1 An antagonistic limb model for impedance control

We wish to study impedance control in planar single-joint reaching movements under different task conditions such as initial or final position, different speeds and adaptation towards external forces. The single joint (point-to-point) reaching paradigm is a well accepted experimental paradigm to investigate simple human reaching behaviour (Osu et al., 2004) and the arm model presented here mimics planar rotation about the elbow joint using two elbow muscles (Fig. 6.2).



Arm parameters	
Link weight [kg]	$m = 1.59$
Link length [m]	$l = 0.35$
Center of gravity [m]	$L_g = 0.18$
Moment of inertia [kg $\cdot$ m^2]	$I = 0.0477$
Moment arms [cm]	$\mathbf{A} = [-2.5\ 2.5]^T$
Muscle parameters	
Elasticity $[N/m]$	$k = 1621.6$
Intrinsic elasticity $[N/m]$	$k_o = 810.8$
Viscosity $[N \cdot s/m]$	$b = 108.1$
Intrinsic viscosity $[N \cdot s/m]$	$b_0 = 54.1$
Rest length constant	$r = 2.182$
Muscle length at rest position [cm] ($\mathbf{q_0} = \pi/2$)	$l_0 = 2.637$

Figure 6.2: Left: Simplified human elbow model with two muscles. Right: Used arm and muscle parameters (adapted from Katayama and Kawato (1993)). Flexor and extensor muscles are modelled with identical parameters.

The nonlinear dynamics of our human elbow is based on standard equations of motion. The joint torques $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} \tag{6.1}$$

with joint angles $\mathbf{q}$, accelerations $\ddot{\mathbf{q}}$, inertia matrix $\mathbf{M}$. The joint torque produced by the antagonistic muscle pair is a function of its muscle tension $\mathbf{t}$ and of the moment arm $\mathbf{A}$, which for simplicity's sake is assumed constant. The effective joint torque from the muscle commands $\mathbf{u} \in [0,1]^2$ is given by

$$\boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = -\mathbf{A}^T \mathbf{t}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}). \tag{6.2}$$

The muscle lengths $\mathbf{l}$ depend on the joint angles $\mathbf{q}$ through the affine relationship $\mathbf{l} = \mathbf{l}_m - \mathbf{A}\mathbf{q}$ which for constant moment arms also implies $\dot{\mathbf{l}} = -\mathbf{A}\dot{\mathbf{q}}$. The constant $\mathbf{l}_m$ is the reference muscle length when the joint angle is at rest. The muscle tension follows a spring-damper model

$$\mathbf{t}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}) = \mathbf{k}(\mathbf{u})\big(\mathbf{l}_r(\mathbf{u}) - \mathbf{l}\big) - \mathbf{b}(\mathbf{u})\dot{\mathbf{l}}, \tag{6.3}$$

where $\mathbf{k}(\mathbf{u})$, $\mathbf{b}(\mathbf{u})$, and $\mathbf{l}_r(\mathbf{u})$ denote the muscle stiffness, the muscle viscosity and the muscle rest length, respectively. Each of these terms depends linearly on the muscle signal $\mathbf{u}$, as given by

$$\mathbf{k}(\mathbf{u}) = diag(\mathbf{k}_0 + k\mathbf{u}), \qquad \mathbf{b}(\mathbf{u}) = diag(\mathbf{b}_0 + b\mathbf{u}), \qquad \mathbf{l}_r(\mathbf{u}) = \mathbf{l}_0 + r\mathbf{u}. \tag{6.4}$$

The elasticity coefficient k, the viscosity coefficient b, and the constant r are given from the muscle model of Katayama and Kawato (1993). The same holds true for $\mathbf{k}_0$, $\mathbf{b}_0$ and $\mathbf{l}_0$, which are the intrinsic elasticity, viscosity and rest length for $\mathbf{u} = \mathbf{0}$, respectively.

To simulate the stochastic nature of neuromuscular signals, often models (Li, 2006) simply contaminate the neural inputs $\mathbf{u}$ with multiplicative noise, scaling the kinematic variability proportional to $\mathbf{u}$. Such signal-dependent noise cannot account for the complex interplay of neuromuscular noise, modified joint impedance and kinematic variability. We introduce stochastic information at the level of the muscle tensions by extending the muscle tension function to be

$$\mathbf{t}^{ext}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}) = \mathbf{t}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}) + \sigma(\mathbf{u})\xi. \tag{6.5}$$

The noise formulation on a muscle level (rather than on a limb level) has the advantage that it can be extended to arm models that incorporate multiple muscles pairs per actuated joint. The variability in muscle tensions depending on antagonistic muscle activations (u_1, u_2) can in a basic form be modeled as an extended SDN function:

$$\sigma(\mathbf{u}) = \sigma_{isotonic}|u_1 - u_2|^n + \sigma_{isometric}|u_1 + u_2|^m, \quad \xi \sim N(0, \mathbf{I}_2). \tag{6.6}$$

The first term (of the distribution's standard deviation) weighted with a scalar accounts for increasing variability in *isotonic* muscle contraction (i.e., contraction which induces joint angle motion), while the second term accounts for the amount of variability for co-contracted muscles (i.e., *isometric* contraction). The parameters $n, m \in R$ define the monotonic increase of the SDN, which in the literature has been reported to range from less than linear $(n, m < 1)$, linear $(n, m = 1)$ or more than linear $(n, m > 1)$. We set $n, m = 1.5$ and further make the reasonable assumption that isotonic contraction causes larger variability than pure isometric contraction $(\sigma_{isotonic} = 0.2, \sigma_{isometric} = 0.02)$. Please note the different absolute value ranges for the isotonic term $|u_1 - u_2|^n \in [0,1]$ and the isometric term $|u_1 + u_2|^m \in [0,2]$ respectively. In reality, at very high levels of co-contraction synchronisation effects may occur, which become visible as tremor of the arm (Selen, 2007). We ignore such extreme conditions in our model. The contraction variability relationship produces plausible muscle tension characteristics without introducing highly complex parameters into the arm model.

To calculate the kinematic variability, the stochastic muscle tensions can be translated into joint accelerations by formulating the forward dynamics including the variability as

$$\ddot{\mathbf{q}}^{ext} = \mathbf{M}^{-1}(\boldsymbol{\tau}^{ext}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})). \tag{6.7}$$

Using the muscle model,

$$\boldsymbol{\tau}^{ext}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = -\mathbf{A}^T \mathbf{t}^{ext}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}) = -\mathbf{A}^T \mathbf{t}(\mathbf{l}, \dot{\mathbf{l}}, \mathbf{u}) - \sigma(\mathbf{u})\mathbf{A}^T \xi \tag{6.8}$$

we get an equation of motion including a noise term

$$\ddot{\mathbf{q}}^{ext} = \mathbf{M}^{-1}(\boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) - \sigma(\mathbf{u})\mathbf{A}^T \xi). \tag{6.9}$$

Multiplying all terms leads to following extended forward dynamics equation

$$\ddot{\mathbf{q}}^{ext} = \ddot{\mathbf{q}} - \sigma(\mathbf{u})\mathbf{M}^{-1}\mathbf{A}^T \xi, \tag{6.10}$$

which is separated into a deterministic component $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = \ddot{\mathbf{q}}$ and a stochastic part $\mathbf{F}(\mathbf{u}) = \sigma(\mathbf{u})\mathbf{M}^{-1}\mathbf{A}^T$. As just shown, the extended SDN corresponds to an additional stochastic term in the joint accelerations which is directly linked to kinematic variability through integration over time. Please note that we introduced this simple but realistic noise model as a *surrogate* for a more elaborate arm muscle model, which ideally would exhibit realistic noise-impedance properties (Selen et al., 2005) all by itself.

One should also note that the stochastic component in our case is only dependent on the muscle signals $\mathbf{u}$, because the matrices $\mathbf{A}$ and $\mathbf{M}$ are independent of the arm states. However this can be easily extended for more complex arm models with multiple links or state-dependent moment arms, and in any case our learning algorithm features fully heteroscedastic variances (that is, a possibly state- and control-dependent noise model).

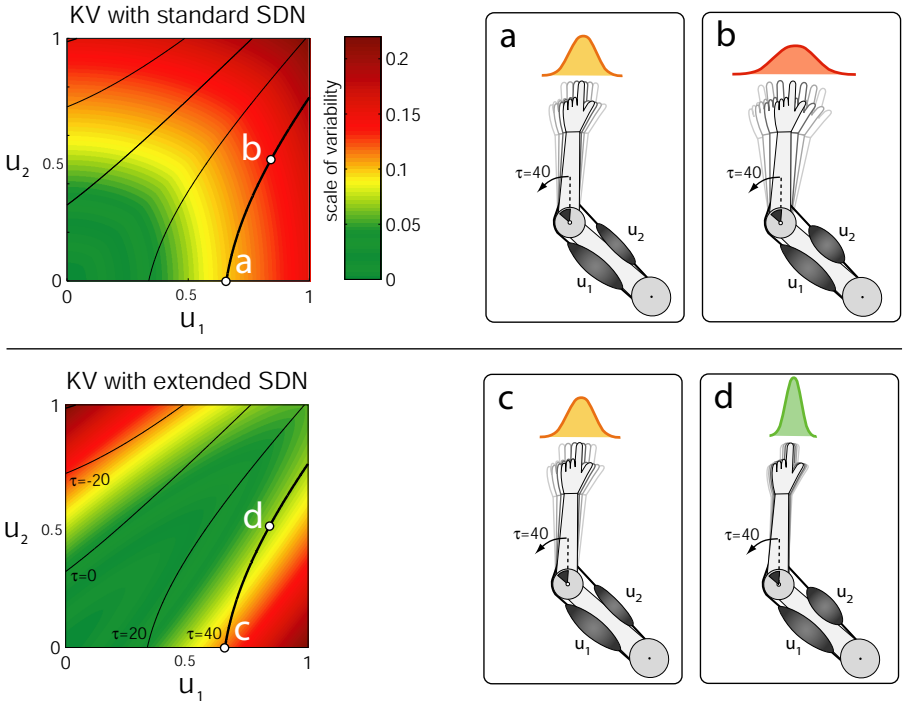


Figure 6.3: Illustration of the effects of standard and extended SDN on kinematic variability in the end-effector. Standard SDN scales proportionally to the muscle activation, whereas the extended SDN takes into account the stabilising effects of higher joint impedance when co-contracting, producing a "valley of reduced SDN" along the co-contraction line. The colors represent the noise variance as a function of muscle activations, whereas the dark lines represent activations that exert the same joint torque computed for joint angle position $\mathbf{q} = \pi/4$. (a) Only muscle $\mathbf{u}_1$ is activated, producing $\boldsymbol{\tau} = 40Nm$ joint torque with a Gaussian kinematic variability of $N\,(0, 0.1)$. (b) The same torque with higher co-contraction produces significantly higher kinematic variability of $N\,(0, 0.15)$ under standard SDN. (c) Same conditions as in (a) in the case where only muscle $\mathbf{u}_1$ is activated. In contrast to (b) the extended SDN in (d) favours co-contraction leading to smaller kinematic variability of $N\,(0, 0.05)$ and to more stable reaching.

Please note that this extended SDN models the kinematic variability that would results from an antagonistic limb system (that suffers from SDN) without introducing large complexities into the dynamics model. The assumptions made in the extended

SDN are supported by numerous experimental and computational results (Osu et al., 2004; Selen et al., 2005) and furthermore provide the computational ingredients to enable stochastic OFC to overcome the "inability" to co-activate. Most importantly, for the presented optimisation and learning framework per se it is irrelevant how the kinematic variability is modelled within the simulation (i.e, extended SDN versus highly detailed simulation model) since the learner acquires the stochastic information from plant data directly. For illustrative purposes, we present the differences between kinematic variability that arise from standard SDN (Fig. 6.3a, 6.3b) and from extended SDN (Fig. 6.3c, 6.3d) as produced by a single joint two-muscle model of the human elbow.

6.4 Uncertainty driven impedance control

Here we will briefly recapitulate the ILQG-LD framework and we show how impedance control emerges from the limb dynamics with extended SDN.

6.4.1 Finding the optimal control law

Based on the stochastic arm model, let $\mathbf{x}(t) = [\mathbf{q}(t), \dot{\mathbf{q}}(t)]^T$ denote the state of the arm model and $\mathbf{u}(t)$ the applied control signal at time t. We can express the forward dynamics in the presence of noise as

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})dt + \mathbf{F}(\mathbf{x}, \mathbf{u})d\boldsymbol{\omega}. \tag{6.11}$$

Here, $d\boldsymbol{\omega}$ is assumed to be Brownian motion noise, which is transformed by a possibly state- and control-dependent matrix $\mathbf{F}(\mathbf{x}, \mathbf{u})$. The finite horizon optimal control problem can be stated as follows: Given the initial state $\mathbf{x}_0$ at time $t = 0$, we seek a (minimal energy) control sequence $\mathbf{u}(t)$ such that the system's state is at the target $\mathbf{x}_{tar}$ at end-time $t = T$. The expected cost, given by the performance index v for such a reaching task (discretised into N steps, $T = N \cdot \Delta t$ seconds) is of the form

$$v = \left\langle w_p \,|\, \mathbf{q}_T - \mathbf{q}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_T \,|^2 + w_e \sum_{k=0}^{N} |\, \mathbf{u}(n) \,|^2 \Delta t \right\rangle. \tag{6.12}$$

The first term penalises reaches away from the target joint angle $\mathbf{q}_{tar}$, the second term forces a zero velocity at the end time T, and the third term penalises large muscle commands (i.e., minimises energy consumption) during reaching. The factors w_p , w_v,

and w_e weight the importance of each component. Typical values for a 0.5 seconds simulation are $N = 50$ steps with a simulation rate of $dt = 0.01$.

In order to find the optimal control law we employ the ILQG method because the arm dynamics $\mathbf{f}$ is highly non-linear in $\mathbf{x}$ and $\mathbf{u}$ and it does not fit into the *linear quadratic* framework (see Chapter 2). The ILQG framework is one of the computationally most efficient approximate OFC methods currently available and it supports stochastic dynamics and control boundaries, which is important to model non-negative muscle commands. ILQG iteratively approximates the nonlinear dynamics and the cost function around the nominal trajectory, and solves a locally valid LQG problem to iteratively improve the trajectory. Along with the optimal open loop parameters $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$, ILQG produces a feedback matrix $\mathbf{L}$ which serves as locally valid optimal feedback law for correcting deviations from the optimal trajectory on the plant.

It is important to note that the noise model $\mathbf{F}(\mathbf{x},\mathbf{u})$, although not visible in the aforementioned cost function *v*, has an important influence on the final solution because ILQG minimises the *expected cost* and thereby takes perturbations into account[1]. For a typical reaching-task cost function as described above, this effectively yields an additional (implicit) penalty term that propagates the final cost backwards "through" the uncertainty model. In our case, if at any time the energy cost of activating both muscles is smaller than the expected benefit of being more stable (minimising uncertainty), then ILQG will command co-contraction. This also explains why our model co-contracts stronger at the final stages of the movement (see Section 6.4.3), where noise has a rather immediate impact on the end point accuracy.

6.4.2 A learned internal model for uncertainty and adaptation

Assuming the internal dynamics model is acquired from sensorimotor feedback then we need to learn an approximation $d\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x},\mathbf{u})dt + \Phi(\mathbf{x},\mathbf{u})d\boldsymbol{\omega}$ of the stochastic plant forward dynamics $d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\boldsymbol{\omega}$. Such problems require supervised learning methods that are capable of (i) efficient non-linear regression in an online fashion (important for adaptation) and (ii) provide heteroscedastic (i.e., localised) prediction variances in order to represent the stochasticity in the dynamics. As the source of stochasticity, we refer to the kinematic variability of the system described above, which encodes for the uncertainty in the dynamics: if a certain muscle action induces large kinematic variability over trials this will reduce the certainty in those regions. Con-

[1] In Chapter 5 we have introduced the uncertainty directly into the cost function.

versely regions in the state-action space that have little variation will be more trustworthy.

We use *locally weighted projection regression (LWPR)*, which is a non-parametric incremental local learning algorithm that is known to perform very well even on high-dimensional motion data (Vijayakumar et al., 2005). Within this local learning paradigm we get access to the uncertainty in form of heteroscedastic prediction variances (see Chapter 4, Section 4.2.2). Once the learning system has been pre-trained thoroughly with data from all relevant regions and within the joint limits and muscle activation range of the arm, a stochastic *OFC with learned dynamics (OFC-LD)* problem can be formulated that "guides" the optimal solution towards a maximum prediction certainty, while still minimising the energy consumption and end point reaching error.

The LWPR learner not only provides us with stochastic information originating from internal SDN, but also delivers an uncertainty measure in cases where the dynamics of the arm changes. Notably the internal dynamics model is continuously being updated during reaching with actual data from the arm, allowing the model to account for systematic perturbations, for example due to external force fields (FF). This is an extension to previously proposed classic optimal control models that relied on perfect knowledge of the system dynamics, given in closed analytic form based on the equations of motion.

From a computational perspective, the approximative OFC methods currently seem to be the most suitable algorithms available to find OFC laws for nonlinear and potentially high dimensional systems. A limiting factor in OFC-LD is the dynamics learning using local methods, which on the one hand is an important precondition for the availability of heteroscedastic variances but on the other hand suffers from the curse of dimensionality, in that the learner has to produce a vast amount of training data to cover the whole state-action space.

6.4.3 Comparison of standard and extended SDN

In the case when the internal model is learned from a plant with stochastic characteristics similar to the extended SDN model, the prediction uncertainty reflects the limb's underlying noise-impedance characteristics, i.e., the fact that co-contraction reduces variability. The optimal control policy therefore should favour co-contraction in order to reduce the negative effects of the SDN.

In order to test the hypothesis of extended SDN, we compared two stochastic OFC-

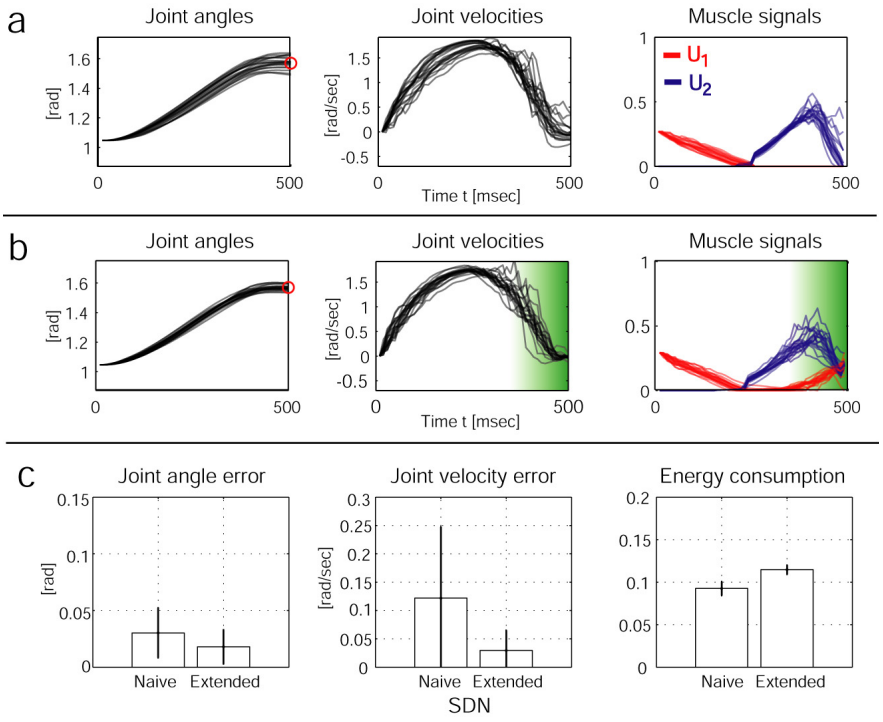


Figure 6.4: Comparison of the results from stochastic OFC using standard SDN (a) and extended SDN (b). We performed 50 OFC reaching movements (only 20 trajectories plotted) under both stochastic conditions. The shaded green area indicates the region and amount of co-contraction in the extended SDN solution. The plots in (c) quantify the results (mean +/- standard deviation). Left: average joint angle error (absolute values) at final time $T = 500msec$. Middle: Joint angle velocity (absolute values) at time T. Right: integrated muscle commands (of both muscles) over trials. The extended SDN outperforms the reaching performance of the standard SDN case at the expense of higher energy consumption.

LD solutions using internal dynamics models learned from a plant that either exhibits standard (Fig. 6.4a) or extended SDN (Fig. 6.4b). The optimal strategy found in this case is to try to avoid large commands $\mathbf{u}$ mostly at the end of the movement, where disturbances can not be corrected anymore. Notably, as is evident from Fig. 6.4a (right), there is still no co-contraction at all. In the extended noise scenario, a solution is found that minimises the negative effects of the noise by increasing co-contraction at the end of the motion (see Fig. 6.4b (right)). The results reveal that the extended SDN performs significantly better than the standard SDN in terms of end point accuracy and end point velocity (Fig. 6.4c). From minimising the uncertainty in a scenario with a neurophysiologically realistic model of kinematic variability, impedance control

naturally emerges from the optimisation, producing the characteristic tri-phasic control signals observed in human reaching (Wierzbicka et al., 1985). Next we present the model's prediction on a set of well known impedance control phenomena in human arm reaching under stationary dynamics conditions.

6.5 Results

In this section we show that the proposed OFC-LD framework exhibits viable impedance control, the results of which can be linked to well known patterns of impedance control in human arm reaching. First we will discuss two experiments in stationary dynamics, i.e., the dynamics of the arm and its environment are not changing. The third experiment will model the non-stationary case, where the plant is perturbed by an external force field and the system adapts to the changed dynamics over multiple reaching trials.

Before starting the reaching experiments we learned an accurate forward dynamics model $\tilde{\mathbf{f}}$ with data of our arm. We coarsely pre-trained an LWPR dynamics model with a data set S collected from the arm model without using the extended noise model. The data was densely and randomly sampled from the arm's operation range with $\mathbf{q} = [\frac{2}{9}\pi, \frac{7}{9}\pi]$, $\dot{\mathbf{q}} = [-2\pi, 2\pi]$, and $\mathbf{u} = [0, 1]$. The collected data set ($2.5 \cdot 10^6$ data points) was split into a 70% training set and a 30% test set. We stopped learning once the model prediction of could accurately replace the analytic model, which was checked using the *normalised mean squared error (nMSE)* of $5 \cdot 10^{-4}$ on the test data. After having acquired the noise free dynamics accurately we collected a second data set S^{noise} in analogy to S but this time the data was drawn from the arm model *including* the extended noise model. We then used S^{noise} to continue learning on our existing dynamics model $\tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u})$. The second learning round has primarily the effect of shaping the confidence bounds according to the noise in the data and the learning is stopped once the confidence bounds stop changing. One correctly could argue that such a two step learning approach is biologically not feasible because a human learning system for example never gets noise-free data. The justification of our approach is of a practical nature and simplifies the rather involved initial parameter tuning of LWPR and allows us to monitor the global learning success (via the nMSE) more reliably over the large data space. Fundamentally though, our learning method does not conflict with any stochastic OFC-LD principles that we propose.

For all experiments stochastic *ILQG with learned dynamics (ILQG-LD)* was used to calculate the optimal control sequence for reaching of duration $T = 500msec$ with

a sampling rate of $10msec$ ($dt = 0.01$). The feedback matrix L served as optimal feedback gains of the simulated antagonistic arm.

6.5.1 Experiment 1: Impedance control for higher accuracy demands

Although energetically expensive, co-contraction is used by the motor system to facilitate arm movement accuracy in single-joint (Osu et al., 2004) and multi joint reaching (Gribble et al., 2003). Experimentally, an inverse relationship between target size and co-contraction has been reported. As target size is reduced, co-contraction and joint impedance increases and trajectory variability decreases.

To model different accuracy demands in ILQG, we modulate the final cost parameter w_p and w_v in the cost function, which weights the importance of the positional endpoint accuracy and velocity compared to the energy consumption. Like this we create five different accuracy conditions: (A) $w_p = 0.5$, $w_v = 0.25$; (B) $w_p = 1$, $w_v = 0.5$; (C) $w_p = 10$, $w_v = 5$; (D) $w_p = 100$, $w_v = 50$; (E) $w_p = 500$, $w_v = 250$; The energy weight for each condition is $w_e = 1$. Next we used ILQG-LD to simulate optimal reaching starting at $\mathbf{q}_0 = \frac{\pi}{3}$ towards the target $\mathbf{q}_{target} = \frac{\pi}{2}$. Movement time was $T = 500ms$ with a sampling rate of $10ms$ ($dt = 0.01$). For each condition we performed 20 reaching trials.

As in the CNS, our model predicts the energetically more expensive strategy to facilitate arm movement accuracy. Fig. 6.5 shows the predictions of our model for five conditions ranging from low accuracy demands (A) to high accuracy demands (E). In condition (A), very low muscle signals suffice to satisfy the low accuracy demands, while in the condition (E), much higher muscle signals are required, which consequently leads to higher co-contraction levels. A similar trend of increased muscle activation has been reported experimentally (Laursen et al., 1998). From an optimal control perspective, an increase in accuracy demands means also that influence of the stochasticity in the dynamics is weighted higher, which leads to a reduction of the relative importance of the energy efficiency in the cost function.

6.5.2 Experiment 2: Impedance control for higher velocities

Next we test our model predictions in conditions where the arm peak velocities are modulated. Humans increase co-activation as well as reciprocal muscle activation

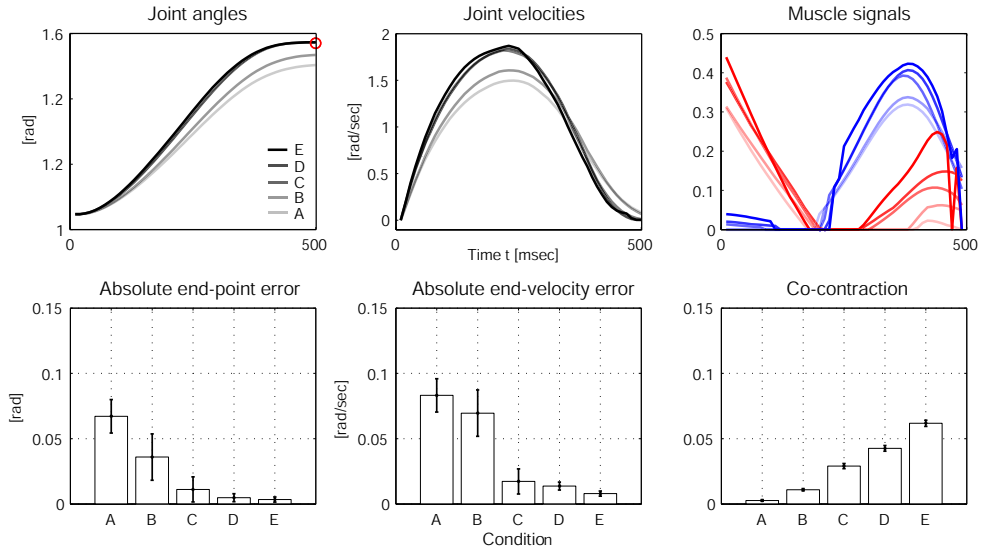


Figure 6.5: Experimental results from stochastic OFC-LD for different accuracy demands. The first row of plots shows the averaged joint angles (left), the averaged joint velocities (middle) and the averaged muscle signals (right) over 20 trials for the five conditions A, B, C, D, and E. The darkness of the lines indicates the level of accuracy; the brightest line indicates condition A, the darkest condition E. The bar plots in the second row average the reaching performance over 20 trials for each condition. Left: The absolute end-point error and the end-point variability in the trajectories decreases as accuracy demands are increased; Middle: End-point stability also increases (demonstrated by decreasing error in final velocities); Right: The averaged co-contraction integrated during 500 msec increases with higher accuracy demands, leading to the reciprocal relationship between accuracy and impedance control as observed in humans.

with maximum joint velocity and it was hypothesised that the nervous system uses a simple strategy to adjust co-contraction and limb impedance in association with movement speed (Suzuki et al., 2001; Gribble and Ostry, 1998). The causalities here are that faster motion requires higher muscle activity which in turn introduces more noise into the system, the negative effects of which can be limited with higher joint impedance. Assuming that the reaching time and accuracy demand remains constant, peak velocities can be modulated using targets with different reaching distance. Here we set the start position to $\mathbf{q}_0 = \frac{\pi}{6}$ and define three reaching targets with increasing distances: $\mathbf{q}_{near} = \frac{\pi}{3}$; $\mathbf{q}_{medium} = \frac{\pi}{2}$; $\mathbf{q}_{far} = \frac{2\pi}{3}$. The cost function parameters are $w_p = 100$, $w_v = 50$, and $w_e = 1$. We again performed 20 trials per condition using ILQG-LD.

The results in Fig. 6.6 show that the co-contraction increases for targets that are

further away and have a higher peak velocity. The reaching performance remains good for all targets, while there are minimal differences in end-point and end-velocity errors between conditions. This can be attributed to the fact that we reach for different targets, which may be harder or easier to realise for ILQG with the given cost function parameters and reaching time T.

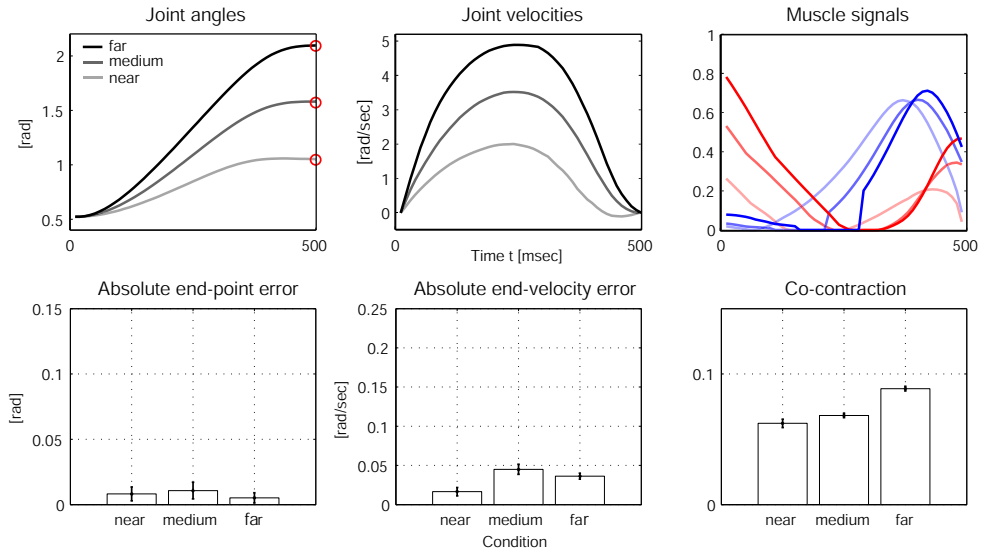


Figure 6.6: Experimental results from stochastic OFC-LD for different peak joint velocities. The first row of plots shows the averaged joint angles (left), the averaged joint velocities (middle) and the averaged muscle signals (right) over 20 trials for reaches towards the three target conditions “near”, “medium” and “far”. The darkest line indicates “far”, the brightest indicates the “near” condition. The bar plots in the second row quantify the reaching performance averaged over 20 trials for each condition. The end-point errors (left) and end-velocity errors (middle) show good performance but no significant differences between the conditions, while co-contraction during the motion as expected increases with higher velocities, due to the higher levels of muscle signals.

The presented stationary experiments exemplified how the proposed stochastic OFC-LD model can explain the emergence of impedance control. In both experiments, OFC-LD increasingly makes use of co-contraction in order to fulfill the changing task requirements by choosing “more certain” areas of the internal dynamics model. While in the first case this is directly caused by the higher accuracy demand, in the second case the necessarily larger torques would yield less accuracy without co-contraction. Typically, “M-shaped” co-contraction patterns are produced, which in our results were biased towards the end of the motion. The bias can be attributed to the nature of the

finite-horizon optimal control solution, which penalises the effects of noise more towards the end of the motion, i.e., near the target state. Notably, M-shaped co-activation patterns have been reported experimentally (Gomi and Kawato, 1996) linking the magnitude of co-activation directly to the level of reciprocal muscle activation.

6.5.3 Experiment 3: Impedance control during adaptation towards external force fields

Adaptation paradigms, typically using a robotic manipulandum, have been a very fruitful line of experimental research (Shadmehr and Mussa-Ivaldi, 1994). In such setups, subjects are first thoroughly trained under normal reaching conditions (*null field (NF)*) and then, their adaptation process to changed dynamics (e.g., novel FF) is studied in consecutive reaching trials. While we have already linked uncertainties from internal sources to impedance modulation, the force field paradigm introduces additional "external" uncertainties of often larger magnitude. As we show next, in the spirit of the umbrella example from the introduction, the notion of internal model uncertainties becomes important for impedance control during adaptation.

A particular benefit of our model is that it employs an entirely data driven (learned) internal dynamics and noise model, meaning it can model changes in the environmental conditions. In the FF *catch trial* (the first reach in the new FF condition), the arm gets strongly deflected, missing the target because the internal model $\tilde{\mathbf{f}}(\mathbf{x},\mathbf{u})$ cannot yet account for the "spurious" forces of the FF. However, using the resultant deflected trajectory as training data and updating the dynamics model online brings the arm nearer to the target with each new trial as the internal model predictions become more accurate for the new condition.

Our adaptation experiment starts with 5 trials in a NF condition, followed by 20 reaching trials in the FF condition. The reach adaptation experiments were carried out with a constant force acting on the end-effector (i.e., hand). Within all reaching trials, the ILQG-LD parameters were set to: $T = 500ms$, $w_p = 100$, $w_v = 50$, and $w_e = 1$, $\mathbf{q}_0 = \frac{\pi}{2}$, and $\mathbf{q}_{tar} = \frac{\pi}{3}$. The force-field trials arm dynamics are simulated using a *constant force field* $FF = (10,0,0)^T$ acting in positive x-direction, i.e., in direction of the reaching movement.

For each trial, we monitored the muscle activations, the co-contraction and the accuracy in the positions and velocities. Since the simulated system is stochastic and suffers from extended SDN, we repeated the adaptation experiment 20 times under

the same conditions and averaged all results. Fig. 6.7 aggregates these results. We see in the kinematic domain (left and middle plots) that the adapted optimal solution differs from the NF condition, suggesting that a re-optimisation takes place. After the force field has been learned, the activations for the extensor muscle are lower and those for the flexor muscle are higher, meaning that the optimal controller makes use of the supportive force field in positive x-direction. Indeed these results are in line with recent findings in human motor learning, where Izawa et al. (2008) presented results that suggest that such motor adaptation is not just a process of perturbation cancellation but rather a re-optimisation w.r.t. motor cost and the novel dynamics.

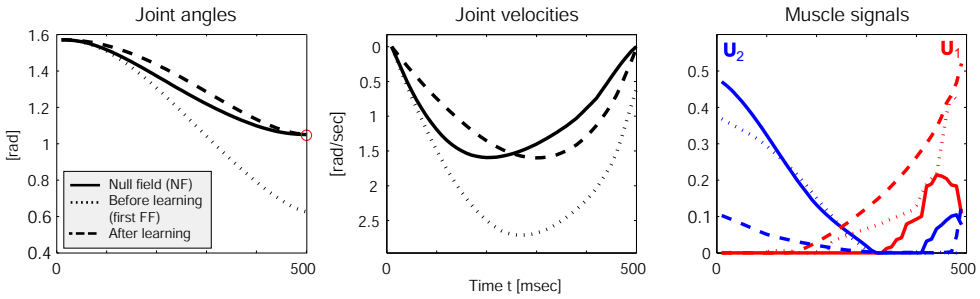


Figure 6.7: Optimal reaching movement, before, during and after adaptation. Clearly the solution is being re-optimised with the learned dynamics (including the FF).

To analyse the adaptation process in more detail, Fig. 6.8a presents the integrated muscle signals and co-contraction, the resultant absolute end-point and end-velocity errors and the prediction uncertainty of the internal model (i.e., heteroscedastic variances) during each of the performed 25 reaching trials. The prediction uncertainty was computed after each trial with the updated dynamics along the current trajectory. The first five trials in the NF condition show approximately constant muscle parameters along with good reaching performance and generally low prediction uncertainties. Even in the NF condition, the learning further reduces the already low uncertainty. In trial 6, the FF catch trial, the reaching performance drops drastically due to the novel dynamics. This also increases the prediction uncertainty since the input distribution along the current trajectory has changed and "blown up" the uncertainty in that region. Consequently the OFC-LD algorithm now has to cope with increased uncertainty along that new trajectory. These can be reduced by increasing co-contraction and therefore entering lower noise regions, which allow the algorithm to keep the uncertainty lower and still produce enough joint torque. For the next four trials, i.e. trials 7 to 10, the co-activation level stays elevated while the internal model gets updated, which is indicated

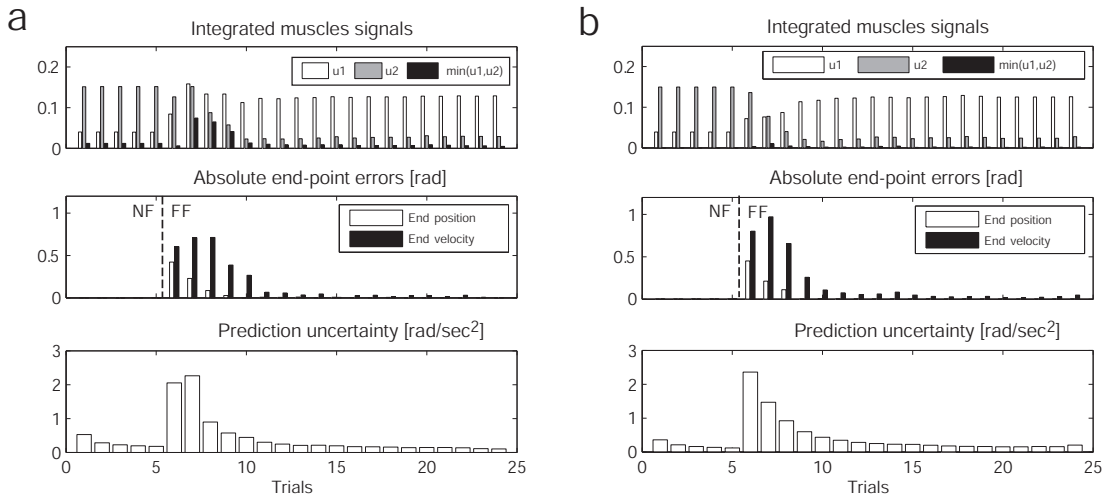


Figure 6.8: Adaptation results. (a) Accumulated statistics during 25 adaptation trials using stochastic OFC-LD. Trials 1 to 5 are performed in the NF condition. Top: Muscle activations and co-contraction integrated during $500ms$ reaches. Middle: Absolute joint errors and velocity errors at final time $T = 500ms$. Bottom: Integrated (internal model) prediction uncertainties along the current optimal trajectory, after this has been updated. (b) The same statistics for the adaptation using deterministic OFC-LD, meaning no uncertainty information is used for the optimisation. This leads to no co-contraction and therefore worse reaching performance during adaptation.

by the change in reciprocal activations and improved performance between those trials. After the 11th trial the co-contraction has reduced to roughly the normal NF level and the prediction uncertainty along the trajectory is fairly low (< 1) and keeps decreasing, which highlights the expected connection between impedance and prediction uncertainty. A further indication for the viability of our impedance control model is supported with a direct comparison to the deterministic case. We repeated the same adaptation experiment using a deterministic OFC-LD implementation, meaning the algorithm ignored the stochastic uncertainty information available for the optimisation (Fig. 6.8b). For the deterministic case, one can observe that virtually no co-contraction during adaptation is produced. This leads generally to larger errors in the early learning phase (trial 6 to 10), especially in the joint velocities. In contrast, for the stochastic algorithm, the increased joint impedance stabilises the arm better towards the effects of the FF and therefore, produces smaller errors.

The comparison of the stochastic versus deterministic adaptation example highlights the importance for the optimal controller to be able to learn the stochastic information from the motor system in the NF condition at first hand, i.e., the structure

of the kinematic variability resulting from the extended SDN. Acquiring this stochastic information structure is a prerequisite to achieve more stable reaching performance during adaptation tasks, by increasing the limb impedance.

6.6 Discussion

We presented a model for joint impedance control, which is a key strategy of the CNS to produce limb control that is stable towards internal and external fluctuations. Our model is based on the fundamental assumption that the CNS, besides optimising for energy and accuracy, minimises the expected uncertainty from its internal dynamics model predictions. Indeed this hypothesis is supported by numerous experimental findings in which the CNS sacrifices energetic costs of muscles to reach stability through higher joint impedance in uncertain conditions. We showed that, in conjunction with an appropriate antagonistic arm and SDN model, the impedance control strategy emerges from first principles as a result of an optimisation process that minimises for energy consumption and reaching error. Unlike previous OFC models, here, the actor utilises a learned dynamics model from data that are produced by the limb system directly. The learner incorporates the contained kinematic variability, here also termed noise, as prediction uncertainty which is represented algorithmically in form of heteroscedastic (i.e., localised) variances. With these ingredients, we formulated a stochastic OFC algorithm, called OFC-LD that uses the learned dynamics and the contained uncertainty information. This generic model for impedance control of antagonistic limb systems is solely based on the quality of the learned internal model and therefore, leads to the intuitive requirement that impedance will be increased in cases where the actor is uncertain about the model predictions. The simulated model predictions agree with several well-known experimental findings from human impedance control and, for the first time, does so from first principles of optimal control theory.

Even though the proposed framework here makes use of specific computational techniques for nonlinear OFC (i.e., ILQG) and heteroscedastic learning (i.e., LWPR), alternative methods could be applied. The key novelty of our computational model is that it unifies the concepts of energy-optimality, internal model learning and uncertainty to a principled model of limb impedance control.

Besides the experimental evidence, OFC-LD seems a plausible approach for modelling how the CNS realises impedance control. The formulation within optimal control theory is well motivated since impedance in humans has been shown to be tuned

optimally with respect to task accuracy and energy consumption (Burdet et al., 2001). Furthermore, our model utilises the concepts of learned internal dynamics models, which plays an important role in the formation of sensorimotor control strategies and which in practice allowed us to model adaptation and re-optimisation. Notably, the learning framework exclusively uses data for learning that are thought to be available to the nervous system through visual and proprioceptive feedback, therefore delivering a sound analogy to a human learner.

As suggested previously (Franklin et al., 2008), a key aspect for the realisation of co-activation is the introduction of an error measure into the optimisation process that "triggers" impedance control. In our model, we create a unified treatment of the various sources of kinematic variability (sensorimotor noise, external perturbations etc.) by incorporating this into a perceived error in internal model predictions. Indeed, many human motor behaviours can be explained by stochastic optimal control models that minimise the impact of motor noise (Harris and Wolpert, 1998; van Beers et al., 2004; van Beers, 2009). In the case of extended SDN, the structured stochasticity provides additional information about the system dynamics and the emergent impedance control may be a further indication of the possible constructive role of noise in the neuromotor system (Faisal et al., 2008). The methodology we suggest for optimal exploitation of sensorimotor stochasticity through learning is a generic principle that goes beyond the modelling of signal dependent sources of noise but can be generalised to deal with other kinds of control or state dependent uncertainties. An example would be uncertainties that depend on the arm position or current muscle lengths.

In the presented optimal control formulation the uncertainty of the internal model predictions are included in the dynamics formulation as a stochastic term. Alternatively one could introduce uncertainty as an additional "uncertainty term" into the cost function as presented in Chapter 5. The advantage of the current approach is that uncertainty or kinematic variability is modeled at its origin, i.e., in the dynamics of the system. Like this we not only can retain the original cost function description but also take into account the time course of the movement and therefore minimise directly for the "detrimental effects" of the uncertainty specifically to our planning time horizon as shown in the stationary experiments.

While we have suggested a computational framework to bridge the gap between optimal control and co-activation, there is still limited knowledge about the neural substrate behind the observed optimality principles in motor control (Shadmehr and Krakauer, 2008). Our model is a first attempt to formalise the origins of impedance

control in the CNS from first principles and many modifications could be considered. For instance, so far only process noise is modelled and observation noise is ignored entirely. This is a simplification of real biological systems, in which large noise in the observations is present, both from vision and proprioceptive sensors. Computationally there are methods for solving nonlinear stochastic OFC with partial observability (Todorov, 2005; Li, 2006), which could be employed for such a scenario. Experimentally, however, no connection between observation noise and impedance control has been established. While this work has focused on the origins of impedance phenomena rather than on a faithful reproduction of published patterns, the predictions of the adaptation experiments are in remarkable agreement with previous findings (Shadmehr and Mussa-Ivaldi, 1994; Izawa et al., 2008). Furthermore, to the best of our knowledge, this is the first computational model to predict impedance control for both, stationary and adaptation experiments. Most importantly, our model is able to qualitatively predict the time course of impedance modulation across trials depending on the "learnability" of the external perturbations. The presented results can be expected to scale to higher dimensional systems, since impedance control seems to originate from the antagonistic muscle structure in the joint-space domain (McIntyre et al., 1996; Gribble and Ostry, 1998; Franklin et al., 2007). It remains to be seen whether the minimum uncertainty approach has the capability to explain other important multi-joint impedance phenomena such as the end-effector stiffness that is selectively tuned towards the directions of instability (Burdet et al., 2001; Lametti et al., 2007). Nevertheless our general model of impedance control may serve as an important step towards the understanding of how the CNS modulates impedance.

Chapter 7

Conclusions

In this thesis we have investigated several aspects related to the optimal feedback control of anthropomorphic systems. The three main objectives in this thesis were (i) to implement OFC on realistic hardware systems and (ii) to develop a principled, data driven adaptation paradigm within OFC, which can (iii) exploit stochastic information in a task optimal fashion.

In Chapter 3 we discussed the issues related to the implementation of OFC on a realistic manipulator hardware with large DoF. Using ILQG with an adapted cost and dynamics function we presented results on the Barrett WAM that highlight the beneficial properties of the OFC strategy in terms of energy consumption and compliance during control. These properties are of particular importance for anthropomorphic robots designed to interact in a human centered environment. In Chapter 4 we introduced the OFC-LD framework, which allowed us to study systems that involve unknown or changing dynamics conditions by using an online supervised dynamics learning paradigm. We compared the optimal solutions of ILQG–LD with those of standard ILQG on numerous setups that exhibit kinematic and dynamic redundancies. We discussed the conceptual advantages of the OFC-LD in adaptation tasks, which are of particular interest for the modelling of biological motor control. The discussion was then extended to systems that suffer from noise and uncertainties in the dynamics. In Chapter 5 we looked into robotic systems that experience stochastic perturbations induced, for example, through the use of a power tool. We proposed to incorporate the learned stochastic dynamics information into the optimisation via an extended cost function. Implementing this stochastic OFC-LD on a newly developed antagonistic SEA, revealed that optimal impedance control, achieved by motor co-contraction, leads to improved motor performance (in terms of task accuracy) over

the deterministic optimisation. At last, in Chapter 6, we discussed stochastic OFC-LD from a biological motor control perspective. We showed that, under the assumption of a realistic (stochastic) biomechanical plant model, OFC-LD predicts a wide range of impedance control patterns observed in humans both in stationary and adaptation tasks. Most importantly, the OFC-LD model is based on biologically plausible assumptions and impedance control is not modelled explicitly but emerges from the optimisation of the defined task at hand.

This thesis has discussed several issues related to optimal control with a practical viewpoint on manipulator control. Nevertheless we have linked our results to some theoretical properties that are well known in optimal control theory. We briefly recapitulate some of them here. For example in Chapters 3 and 4, we have discussed ILQG and the *minimum intervention principle* (Todorov and Jordan, 2003) and its beneficial properties for manipulator control. Please note that the idea of minimum intervention has a long lasting history in the domain of classic control theory (Aubin, 1991) as well as in the neuro-scientific literature (Scholz and Schöner, 1999). Another example is Chapter 3 where we have taken a rather isolative view on open loop and closed loop optimal control, while other classic approaches, such as *robust control*, have not been investigated. At last, in relation to Chapters 5 and 6, it is well established from the classic optimal control literature (e.g., Holly and Hughes (1989)) that *uncertainty* in the environment or the dynamics changes the qualitative behaviour of the obtained optimal solutions. The key contribution made within OFC-LD is that that uncertainty is *learned* from data rather than known a priory and that using this stochastic information can lead to impedance control in antagonistic systems.

From an optimal control perspective ILQG currently seems the most suitable OFC method for the study of nonlinear and potentially high-dimensional systems, delivering robust solutions in a computationally efficient manner. However this method also exhibits certain drawbacks: In ILQG many open parameters must be defined, the values of which may significantly change the optimisation outcome. Some examples include the initial control commands, the Levenberg-Marquardt parameters, the convergence thresholds, reaching time and cost function weights. Furthermore due to the local iterative nature of the ILQG, for realistic systems like the Barrett WAM, it is difficult to determine how close the obtained solutions are from the global (unknown) optimal solution. One can alleviate this problem by using different initial trajectories and comparing the different optimal solutions.

The fundamental advantage of learning the dynamics is that it allows us to model

adaptation and extract stochastic properties of the plant in a principled fashion. Furthermore, as shown in Chapter 4, under certain conditions it can lead to computationally more efficient solutions within ILQG–LD. However several issues remain when working with local learning methods like LWPR. Using LWPR requires a significant amount of practical experience and manual parameter tuning. Furthermore trying to employ LWPR to learn the dynamics of the WAM hardware has revealed several practical limitations. First, in practice it is difficult to cover large spaces of the state action space, i.e., the amount of data required to learn an accurate dynamics model is very large and would require extremely long operation time on the robot. Secondly, most robotic systems are only equipped with joint position sensors, which means that joint accelerations need to be computed via numerical differentiation. This in turn introduces large sources of noise in the acceleration signals, which complicates the learning of an accurate dynamics model. Third friction and discontinuities in the dynamics pose a serious problem as LWPR averages discontinuities in an unpredictable fashion. Discontinuities in locally weighted learning have been addressed in Toussaint and Vijayakumar (2005). When developing the SEA platform used in Chapter 5, we tried to address the discontinuity issues by creating a mechanically simple system that exhibits low joint friction characteristics.

We can conclude from this thesis that it remains very challenging to achieve model based control on real robotic systems based solely on a learned dynamics model. While many approaches use learning techniques to improve tracking performance along single trajectories (e.g., Nguyen-Tuong et al. (2008a)), for techniques like ILQG–LD the situation is more difficult: It is not sufficient to learn the dynamics very accurately just in the neighbourhood of a single trajectory. For ILQG–LD to converge successfully a very large state action space needs to be learned. Therefore in practice it may be reasonable to use an accurate analytic base model (if available) and restrict the learning to an error function of the dynamics (Morimoto and Atkeson, 2003). Like this ILQG convergence is secured while, at the same time, the advantages of the dynamics learning paradigm can be exploited.

Outlook and future work

There exist a number of directions for future extensions based on work presented in this thesis.

OFC(-LD)

- Implementing a model predictive control version of ILQG similar to Tassa et al. (2007) would be particularly beneficial to improve quality of the hardware results. The challenge here certainly is to create an implementation that is computationally efficient to allow real-time control. Especially when motor dynamics are modelled the state action space becomes extensively large increasing computational complexity. To overcome computational problems one could focus on restricted areas of the state action space or use pre-computed solutions as initial trajectories for ILQG.

- In this thesis we did not perform a stability analysis for ILQG–LD and we only made empirical observations on the stability. Performing stability analysis for systems dynamics based on LWPR, which is a complicated nonlinear regression technique that depends on many parameters, can be expected to be very challenging. Furthermore there is only limited amount of previous work on stability for adaptive control using localised models (Nakanishi et al., 2005).

Biological aspects of OFC-LD

- A logical extension to the single joint experiments from Chapter 6 would be to scale the results to systems with larger DoF since many psychophysical experiments have been performed on planar arm models with multiple DoF. The aim would be to reproduce, for example, the well known results from Burdet et al. (2001) or Lametti et al. (2007), i.e., to predict directional task-dependent impedance control in Euclidean task space. The most challenging aspects here are to create an accurate arm and muscle model and to learn the extended SDN characteristics in a high dimensional space.

- Another interesting route would be to verify the extended SDN model hypothesis in psychophysical experiments. The idea would be to create conditions for human subjects, in which the SDN conditions can be modified during the experiment. For example for reaching tasks one would measure EMG in real-time and feed the SDN characteristics (in a potentially modified form) back to the manipulandum. Such manipulanda with EMG feedback have been recently proposed by Ganesh et al. (2010). A similar setup could enable us to investigate whether

new SDN conditions can be learned and if this leads to a change in muscle co-activation patterns in human subjects.

- Even though we have studied stochastic dynamics, observation noise has not been addressed in this thesis. Especially in the context of biological motor systems estimation noise is of central interest and may be addressed in the future.

Appendix A

ILQG Code in Matlab

A.1 ILQG main function

The ILQG function takes 9 values as input.

- Three function pointers: Forward dynamics function (`fnDyn`), noise model (`fnNoise`) and cost function (`fnCost`).
- Simulation step size `dt` and length of trajectory `n`.
- Initial state `x0` and initial control sequence `u0`.
- lower and upper bound on admissible controls `umin` and `umax`.

The function returns optimal control sequence `u`, corresponding state sequence `x` and optimal feedback control law `L`.

```
function [x, u, L] = iLQG(fnDyn, fnNoise, fnCost, dt, n, x0, u0, umin, umax)

% -------------- user-adjustable parameters ------------------
lambdaInit =  100;       % initial value of Levenberg-Marquardt lambda
lambdaFactor = sqrt(10); % factor for multiplying or dividing lambda
lambdaMax = 1e7;         % exit if lambda exceeds threshold lambdaMax
epsConverge = 1e-15;     % exit if relative improvement below threshold epsConverge
maxIter = 200;           % exit if number of iterations reaches threshold

% -------------- Create a nominal trajectory  -----------------
x = zeros(szX, n);          % init state sequence and cost
[x, cost] = simulate(fnDyn, fnCost, dt, x0, u, maxValue);

lambda = lambdaInit;
flgChange = 1;
```

```
% -------------- main ILQG loop -------------------------------
for iter = 1:maxIter

    %------ STEP 2: approximate dynamics and cost along new trajectory'
    if flgChange,
        [s0(n),s(:,n),S(:,:,n)] = fnCost(x(:,n), NaN, NaN);  % final cost

        for k = n-1:-1:1
            % quadratize cost
            [l0,l_x,l_xx,l_u,l_uu,l_ux] = fnCost(x(:,k), u(:,k), k);
            q0(k) = dt * l0;
            q(:,k) = dt * l_x;
            Q(:,:,k) = dt * l_xx;
            r(:,k) = dt * l_u;
            R(:,:,k) = dt * l_uu;
            P(:,:,k) = dt * l_ux;

            % linearize dynamics
            [f, f_x, f_u] = fnDyn(x(:,k), u(:,k));

            A(:,:,k) = eye(szX) + dt * f_x;
            B(:,:,k) = dt * f_u;

            % calculate control dependent noise matrix plus derivatives
            [F, F_x, F_u] = fnNoise(x(:,k), u(:,k));
            c(:,:,k) = sqrt(dt)*F;
            C(:,:,:,k) = sqrt(dt)*F_u;
        end

        flgChange = 0;
    end

    %------ STEP 3: compute optimal control law and cost-to-go
    for k = n-1:-1:1
        % a) compute shortcuts g, G, H
        g = r(:,k) + B(:,:,k)'*s(:,k+1);
        G = P(:,:,k) + B(:,:,k)'*S(:,:,k+1)*A(:,:,k);
        H = R(:,:,k) + B(:,:,k)'*S(:,:,k+1)*B(:,:,k);

        Cg  = zeros(size(g));
        CH  = zeros(size(H));
        Cs0 = zeros(size(q0(k)));
        for i=1:size(C,3)
           Cg = Cg + C(:,:,i,k)'*S(:,:,k+1)*c(:,i,k);
           CH = CH + C(:,:,i,k)'*S(:,:,k+1)*C(:,:,i,k);
           Cs0 = Cs0 + c(:,i,k)'*S(:,:,k+1)*c(:,i,k);
        end
        g = g + Cg;
        H = H + CH;

        % b) find control law
        [l(:,k), L(:,:,k)] = uOptimal(g,G,H,u(:,k),uMin,uMax,lambda);

```

```
        % c) update cost-to-go approximation
        S(:,:,k) = Q(:,:,k) + A(:,:,k)'*S(:,:,k+1)*A(:,:,k) + ...
            L(:,:,k)'*H*L(:,:,k) + L(:,:,k)'*G + G'*L(:,:,k);
        s(:,k) = q(:,k) + A(:,:,k)'*s(:,k+1) + ...
            L(:,:,k)'*H*l(:,k) + L(:,:,k)'*g + G'*l(:,k);
        s0(k) = q0(k) + s0(k+1) + Cs0 + l(:,k)'*H*l(:,k)/2 + l(:,k)'*g;
    end

    %------ STEP 4.1: new control sequence, trajectory, cost
    % simulate linearized system to compute new control
    dx = zeros(szX,1);
    for k=1:n-1
        du = l(:,k) + L(:,:,k)*dx;
        du = min(max(du+u(:,k),uMin),uMax) - u(:,k);

        dx = A(:,:,k)*dx + B(:,:,k)*du ;
        unew(:,k) = u(:,k) + du;
    end

    %------ simulate system to compute new trajectory and cost'
    [xnew, costnew] = simulate(fnDyn, fnCost, dt, x0, unew, maxValue);

    %------ STEP 4.2: Levenberg-Marquardt method'
    if costnew<cost,
        % decrease lambda (get closer to Newton method)
        lambda = lambda / lambdaFactor;

        % accept changes, flag changes
        u = unew;
        x = xnew;
        flgChange = 1;

        if iter>1 & (abs(costnew - cost)/cost < epsConverge),
            cost = costnew;
            break;            % improvement too small: EXIT
        end
        cost = costnew;

    else
        % increase lambda (get closer to gradient descent)
        lambda = lambda * lambdaFactor;

        if lambda>lambdaMax,
            break;            % lambda too large: EXIT
        end
    end
end
```

iLQG.m

A.2 Computing the optimal control law

The following function computes the ILQG control law. It computes a modified Hessian H using four steps: (1) It computes the eigenvalue decomposition $[\mathbf{V},\mathbf{D}] = eig(\mathbf{H})$, (2) It replaces all negative elements of diagonal matrix D with 0, (3) adds a positive regularisation term λ (Levenberg-Marquardt) to the diagonal of $\mathbf{D}$, (4) sets the modified inverse Hessian to $H^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^{\mathbf{T}}$.

```
1  function [l,L] = uOptimal(g, G, H, u, uMin, uMax, lambda)

3  %------ eigenvalue decomposition, modify eigenvalues
   [V,D] = eig(H);
5  d = diag(D);
   d(d<0) = 0;
7  d = d + lambda;

9  %------ inverse modified Hessian, unconstrained control law
   H1 = V*diag(1./d)*V';
11
   l = -H1*g;
13 L = -H1*G;

15 %------ enforce constraints
   l = min(max(l+u,uMin),uMax) - u;
17
   %------ modify L to reflect active constraints
19 L((l+u==uMin)|(l+u==uMax),:) = 0;
```

uOptimal.m

A.3 Cost function example

The following code shows an example cost function for a finite horizon problem. The task is to reach a position defined in the input `target`, to stop at the target and during motion minimise control effort u. For the use with ILQG, the function should be called with `t=NaN` to access the final cost and the cost function derivatives.

```
function [l, l_x, l_xx, l_u, l_uu, l_ux] = arm2Cost(x, u, t, target)

wp = 1E+4;      % terminal position cost weight
wv = 1E+3;      % terminal velocity cost weight

j = size(x,1)/2; % number of joints

%------ compute cost
if isnan(t),                        % final cost
    l = wp*sum((x(1:j)-target).^2) + wv*sum(x(j+1:2*j).^2);
else                                % running cost
    l = sum(u.^2);
end

%------ compute derivatives of cost
if nargout>1,
    l_x  = zeros(2*j,1);
    l_xx = zeros(2*j,2*j);
    l_u  = 2*u;
    l_uu = 2*eye(j);
    l_ux = zeros(j,2*j);

    if isnan(t),                            % final cost
        l_x(1:j) = 2*wp*(x(1:j)-target);
        l_x(j+1:2*j) = 2*wv*x(j+1:2*j);
        WP = repmat(wp,1,j);
        WV = repmat(wv,1,j);
        l_xx = 2*diag([WP  WV]);
    end
end
```

fn_cost.m

A.4 Simulation of the dynamics

Following code simulates the controlled system using standard Euler integration and computes the cost of the trajectory.

```
function [x, cost] = simulate ( fnDyn, fnCost, dt, x0, u )

szX = size(x0,1);       % size of state vector
szU = size(u ,1);       % size of control vector
n   = size(u,2) + 1;    % length of trajectory

%------ initialize simulation
x = zeros(szX, n);
x(:,1) = x0;
cost = 0;

%------ run simulation
for k = 1:n-1
    x(:,k+1) = x(:,k) + dt * fnDyn(x(:,k),u(:,k));

    if nargout>1
        cost = cost + dt * fnCost(x(:,k), u(:,k), k);
    end
end

%------ adjust for final cost
if nargout>1
    cost = cost + fnCost(x(:,end), NaN, NaN);
end
```

simulate.m

Appendix B

Kinematic and dynamic parameters for the Barrett WAM

B.1 Parameters for 4 DoF setup

Table B.1: *Denavit-Hartenberg (DH)* parameters of 4-DOF WAM setup. We use the DH variant proposed by Spong and Vidyasagar (1989). Units in meters and radians.

i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	0	θ_1
2	0	$\pi/2$	0	θ_2
3	0.045	$-\pi/2$	0.55	θ_3
4	-0.045	$\pi/2$	0	θ_4
Tool	0	0	0.35	

Table B.2: Joint angle limits.

Joint	**Positive joint limit** rad (deg)	**Negative joint limit** rad (deg)
1	2.6 (150)	-2.6 (-150)
2	2.0 (113)	-2.0 (-113)
3	2.8 (157)	-2.8 (-157)
4	3.1 (180)	-0.9 (-50)

Table B.3: Mass parameters and centre of mass described as (x,y,x) translations from the link frame.

Link	**Mass** [kg]	**Centre of mass** (x,y,z) [m]
1	8.3936	$(0.3506, 132.6795, 0.6286) \cdot 10^{-3}$
2	4.8487	$(-0.2230, -21.3924, 13.3754) \cdot 10^{-3}$
3	1.7251	$(-38.7565, 217.9078, 0.0252) \cdot 10^{-3}$
4	1.0912	$(11.7534, -0.1092, 135.9144) \cdot 10^{-3}$

Table B.4: Inertia matrices taken at the link's center of mass.

Link	**Inertia matrix** $(I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{yz}, I_{xz})[kg \cdot m^2]$
1	$(95157.4294,\ 92032.3524,\ 59290.5997,\ 246.1404,\ -962.6725,\ -95.0183) \cdot 10^{-6}$
2	$(29326.8098,\ 20781.5826,\ 22807.3271,\ -43.3994,\ 1348.6924,\ -129.2942) \cdot 10^{-6}$
3	$(56662.2970,\ 3158.0509,\ 56806.6024,\ -2321.6892,\ -16.6307,\ 8.2125) \cdot 10^{-6}$
4	$(18890.7885,\ 19340.5969,\ 2026.8453,\ -0.8092,\ 17.8241,\ -1721.2915) \cdot 10^{-6}$

B.2 Parameters for 7 DoF setup

Table B.5: DH-parameters of 7-DOF WAM setup. Units of meters and radians.

i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	0	θ_1
2	0	$\pi/2$	0	θ_2
3	0.045	$-\pi/2$	0.55	θ_3
4	-0.045	$\pi/2$	0	θ_4
5	0	$-\pi/2$	0.3	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.06	θ_7
Tool	0	0	0	

Table B.6: Joint limits.

Joint	**Positive joint limit** rad (deg)	**Negative joint limit** rad (deg)
1	2.6 (150)	-2.6 (-150)
2	2.0 (113)	-2.0 (-113)
3	2.8 (157)	-2.8 (-157)
4	3.1 (180)	-0.9 (-50)
5	1.3 (75)	-4.8 (-275)
6	1.6 (90)	-1.6 (-90)
7	2.2 (128)	-2.2 (-128)

Table B.7: Mass parameters and centre of mass described as (x,y,x) translations from the link frame.

Link	**Mass** [kg]	**Centre of mass** (x,y,z) [m]
1	8.3936	$(0.3506, 132.6795, 0.6286) \cdot 10^{-3}$
2	4.8487	$(-0.2230, -21.3924, 13.3754) \cdot 10^{-3}$
3	1.7251	$(-38.7565, 217.9078, 0.0252) \cdot 10^{-3}$
4	2.1727	$(11.7534, -0.1092, 135.9144) \cdot 10^{-3}$
5	0.3566	$(5.53408, 0.06822, 0.1193) \cdot 10^{-3}$
6	0.4092	$(0.0548, 28.8629, 1.4849) \cdot 10^{-3}$
7	0.0755	$(-0.0592, -16.8612, 24.1905) \cdot 10^{-3}$

Table B.8: Inertia matrices taken at the link's center of mass.

Link	**Inertia matrix**[1] $(I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{yz}, I_{xz})[kg \cdot m^2]$
1	$(95157.4294,\ 92032.3524,\ 59290.5997,\ 246.1404,\ -962.6725,\ -95.0183) \cdot 10^{-6}$
2	$(29326.8098,\ 20781.5826,\ 22807.3271,\ -43.3994,\ 1348.6924,\ -129.2942) \cdot 10^{-6}$
3	$(56662.2970,\ 3158.0509,\ 56806.6024,\ -2321.6892,\ -16.6307,\ 8.2125) \cdot 10^{-6}$
4	$(10674.91,\ 10586.59,\ 2820.36,\ 45.03,\ -110.02,\ -1355.57) \cdot 10^{-6}$
5	$(371.12,\ 194.34,\ 382.09,\ -0.08,\ -16.13,\ -0.03) \cdot 10^{-6}$
6	$(548.89,\ 238.46,\ 451.33,\ 0.19,\ -44.30,\ -0.10) \cdot 10^{-6}$
7	$(39.11,\ 38.77,\ 76.14,\ 0.19,\ 0.00,\ 0.00) \cdot 10^{-6}$

B.3 Motor-joint transformations

In the following we summarise the transformations from joint positions/torques to motor positions/torques and vice versa.

Table B.9: Arm transmission ratios.

Parameter	Value
N_1	42.0
N_2	28.25
N_3	28.25
n_3	1.68
N_4	18.0
N_5	9.7
N_6	9.7
N_7	14.93
n_6	1.0

Equation 1: Arm motor-to-joint position transformation.

$$\begin{pmatrix} J\theta_1 \\ J\theta_2 \\ J\theta_3 \\ J\theta_4 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{-1}{N_1} & 0 & 0 & 0 \\ 0 & \frac{1}{2N_2} & \frac{-1}{2N_2} & 0 \\ 0 & \frac{-n_3}{2N_2} & \frac{-n_3}{2N_2} & 0 \\ 0 & 0 & 0 & \frac{1}{N_4} \end{pmatrix}}_{=P^a_{M2J}} \begin{pmatrix} M\theta_1 \\ M\theta_2 \\ M\theta_3 \\ M\theta_4 \end{pmatrix}$$

Equation 2: Wrist motor-to-joint *position* transformation.

$$\begin{pmatrix} J\theta_5 \\ J\theta_6 \\ J\theta_7 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{2N_5} & \frac{1}{2N_5} & 0 \\ \frac{-n_6}{2N_5} & \frac{n_6}{2N_5} & 0 \\ 0 & 0 & \frac{-1}{N_7} \end{pmatrix}}_{=P^w_{M2J}} \begin{pmatrix} M\theta_5 \\ M\theta_6 \\ M\theta_7 \end{pmatrix}$$

Equation 3: Arm joint-to-motor *position* transformation.

$$\begin{pmatrix} M\theta_1 \\ M\theta_2 \\ M\theta_3 \\ M\theta_4 \end{pmatrix} = \underbrace{\begin{pmatrix} -N_1 & 0 & 0 & 0 \\ 0 & N_2 & \frac{-N_2}{n_3} & 0 \\ 0 & -N_2 & \frac{-N_2}{n_3} & 0 \\ 0 & 0 & 0 & N_4 \end{pmatrix}}_{=P^a_{J2M}} \begin{pmatrix} J\theta_1 \\ J\theta_2 \\ J\theta_3 \\ J\theta_4 \end{pmatrix}$$

Equation 4: Wrist joint-to-motor *position* transformation.

$$\begin{pmatrix} M\theta_5 \\ M\theta_6 \\ M\theta_7 \end{pmatrix} = \underbrace{\begin{pmatrix} N_5 & \frac{-N_5}{n_6} & 0 \\ N_5 & \frac{N_5}{n_6} & 0 \\ 0 & 0 & -N_7 \end{pmatrix}}_{=P^w_{J2M}} \begin{pmatrix} J\theta_5 \\ J\theta_6 \\ J\theta_7 \end{pmatrix}$$

Table B.10: Rotor inertia for each joint motor.

Parameter	**Value**
R_1	0.201
R_2	0.182
R_3	0.067
R_4	0.034
R_5	0.0033224
R_6	0.0033224
R_7	0.000466939

Equation 5: Arm motor-to-joint *torque* transformation.

$$\begin{pmatrix} J\tau_1 \\ J\tau_2 \\ J\tau_3 \\ J\tau_4 \end{pmatrix} = \underbrace{\begin{pmatrix} -N_1 & 0 & 0 & 0 \\ 0 & N_2 & -N_2 & 0 \\ 0 & \frac{-N_2}{n_3} & \frac{-N_2}{n_3} & 0 \\ 0 & 0 & 0 & N_4 \end{pmatrix}}_{=T^a_{M2J}} \begin{pmatrix} M\tau_1 \\ M\tau_2 \\ M\tau_3 \\ M\tau_4 \end{pmatrix}$$

Equation 6: Wrist motor-to-joint *torque* transformation.

$$\begin{pmatrix} J\tau_5 \\ J\tau_6 \\ J\tau_7 \end{pmatrix} = \underbrace{\begin{pmatrix} N_5 & N_5 & 0 \\ \frac{-N_5}{n_6} & \frac{N_5}{n_6} & 0 \\ 0 & 0 & -N_7 \end{pmatrix}}_{=T^w_{M2J}} \begin{pmatrix} M\tau_5 \\ M\tau_6 \\ M\tau_7 \end{pmatrix}$$

Equation 7: Arm joint-to-motor *torque* transformation.

$$\begin{pmatrix} M\tau_1 \\ M\tau_2 \\ M\tau_3 \\ M\tau_4 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{-1}{N_1} & 0 & 0 & 0 \\ 0 & \frac{1}{2N_2} & \frac{-n_3}{2N_2} & 0 \\ 0 & \frac{-1}{2N_2} & \frac{-n_3}{2N_2} & 0 \\ 0 & 0 & 0 & \frac{1}{N_4} \end{pmatrix}}_{=T^a_{J2M}} \begin{pmatrix} J\tau_1 \\ J\tau_2 \\ J\tau_3 \\ J\tau_4 \end{pmatrix}$$

Equation 8: Wrist joint-to-motor *torque* transformation.

$$\begin{pmatrix} M\tau_5 \\ M\tau_6 \\ M\tau_7 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{2N_5} & \frac{-n_6}{2N_5} & 0 \\ \frac{1}{2N_5} & \frac{n_6}{2N_5} & 0 \\ 0 & 0 & \frac{-1}{N_7} \end{pmatrix}}_{=T^w_{J2M}} \begin{pmatrix} J\tau_5 \\ J\tau_6 \\ J\tau_7 \end{pmatrix}$$

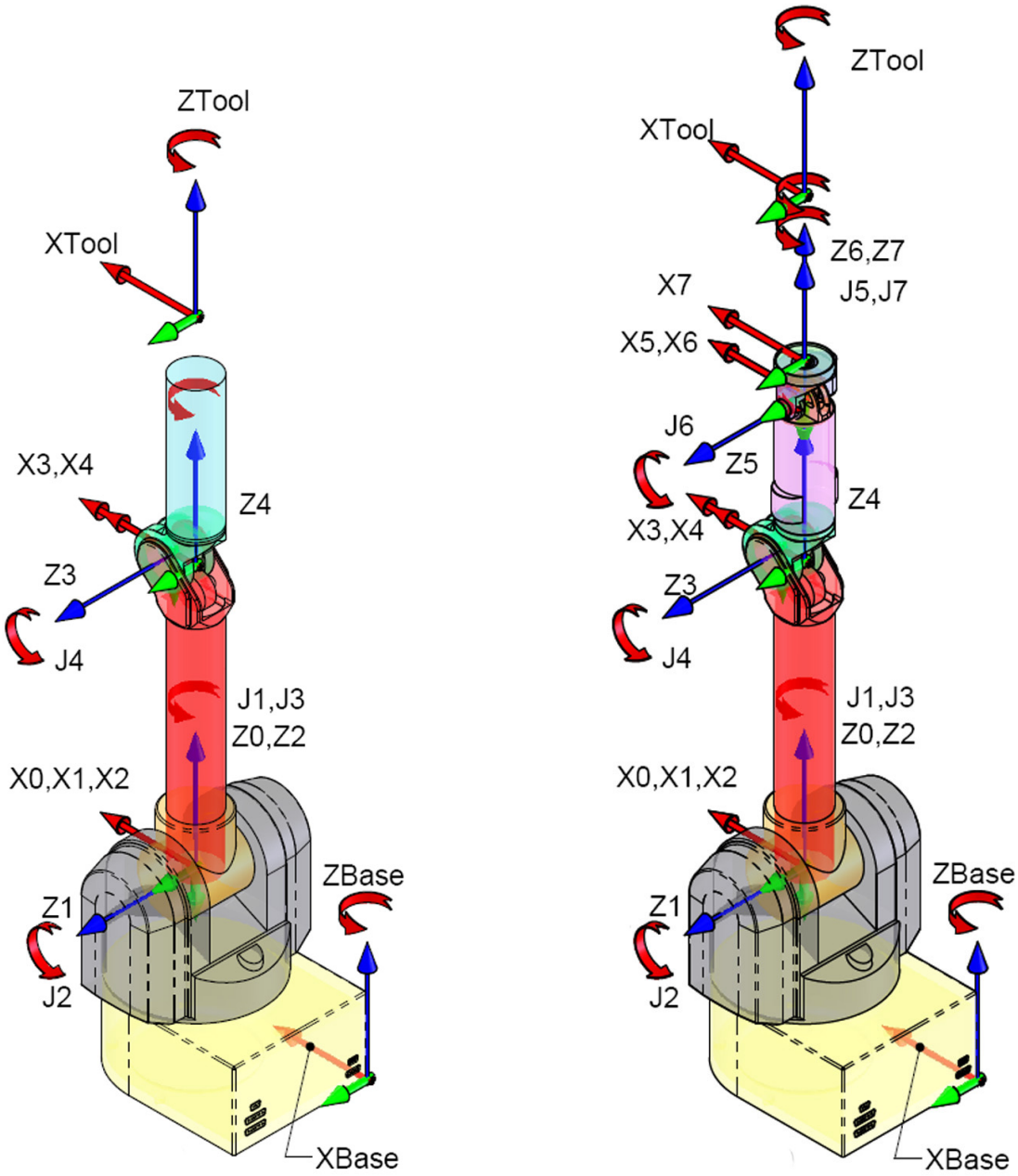
These transformations are required to map the effects of the motor inertia to the joint angle representation as used in ILQG. Hereby the rotor inertia is reflected to the joint space and added to the inertia matrix $\mathbf{M}(\mathbf{q})$ of the rigid body dynamics. This is $\mathbf{M}(\mathbf{q}) = \mathbf{M}_{links}(\mathbf{q}) + \mathbf{M}_{motor}$ where $\mathbf{M}_{links}$ is computed form the inertial parameters using a recursive Newton-Euler method within the Matlab Robotics Toolbox. The motor inertia is computed as follows (4 DoF example shown here)

$$\mathbf{M}_{motor} = P^T_{J2M} I_r P_{J2M} = T_{M2J} I_r P_{J2M}$$

where the rotor inertia is

$$I_r = \begin{pmatrix} \frac{R_1}{N_1^2} & 0 & 0 & 0 \\ 0 & \frac{R_2}{N_2^2} & 0 & 0 \\ 0 & 0 & \frac{R_3}{N_3^2} & 0 \\ 0 & 0 & 0 & \frac{R_4}{N_4^2} \end{pmatrix}.$$

Figure B.1: Illustration of the kinematic structure of the WAM with 4 DoF (left) and 7 DoF (right).



Bibliography

Abbeel, P., Quigley, M., and Ng, A. Y. (2006). Using inaccurate models in reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, volume 148, pages 1–8.

Ahn, H., Chen, Y., and Moore, L. M. (2007). Iterative learning control: brief survey and categorization 1998-2004. *IEEE Transactions on Systems, Man and Cybernetics*, 37(6):1099–1121.

An, C. H., Atkeson, C. G., and Hollerbach, J. M. (1988). *Model-based control of a robot manipulator*. MIT Press.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140.

Athans, M., editor (1971). *Special issue on the linear-quadratic-gaussian estimation and control problem*, volume 16. IEEE Transactions on Automated Control.

Atkeson, C. G. (2007). Randomly sampling actions in dynamic programming. In *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007)*, pages 185–192.

Atkeson, C. G., Moore, A. W., and Schaal, S. (1997). Locally weighted learning for control. *Artificial Intelligence Review*, 11(1-5):75–113.

Atkeson, C. G. and Schaal, S. (1997). Learning tasks from a single demonstration. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1706–1712.

Aubin, J. P. (1991). *Viability theory*. Birkhäuser, Basel.

Bellman, R. (1957). *Dynamic programming*. Princeton University Press.

Bennett, D. J. (1993). Torques generated at the human elbow joint in response to constant position errors imposed during voluntary movements. *Experimental Brain Research*, 95(3):488–498.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena Scientific, Belmont, Mass.

Bliss, G. A. (1946). *Lectures on the calculus of variations*. University of Chicago.

Bolza, O. (1909). *Vorlesungen über Variationsrechnung*. Druck und Verlag von B. G. Tuebner, Leipzig u. Berlin.

Bryson, A. E. (1996). Optimal control - 1950 to 1985. *IEEE Control Systems Magazine*, pages 26–33.

Bryson, A. E. and Ho, Y. C. (1975). *Applied optimal control*. Hemisphere/Wiley.

Burdet, E., Osu, R., Franklin, D. W., Milner, T. E., and Kawato, M. (2001). The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414:446–449.

Burdet, E., Tee, K. P., Mareels, I., Milner, T. E., Chew, C. M., Franklin, D. W., Osu, R., and Kawato, M. (2006). Stability and motor adaptation in human arm movements. *Biological Cybernetics*, 94(1):20–32.

Castellini, C., van der Smagt, P., Sandini, G., and Hirzinger, G. (2008). Surface emg for force control of mechanical hands. In *International Conference on Robotics and Automation (ICRA)*.

Chhabra, M. and Jacobs, R. A. (2006). Near-optimal human adaptive control across different noise environments. *The Journal of Neuroscience*, 26(42):10883–10887.

Choi, J. and Farrell, J. (2000). Nonlinear adaptive control using networks of piecewise linear approximators. *IEEE Transactions on Neural Networks*, 11(2):390–401.

Collins, S. H. and Kuo, A. D. (2010). Recycling energy to restore impaired ankle function during human walking. *PLoS ONE*, 5(2):e9307.

Collins, S. H. and Ruina, A. (2005). A bipedal walking robot with efficient human-like gait. In *Proceedings of the IEEE international Conference on Robotics and Automation (ICRA)*, pages 1983–1988.

Conradt, J., Tevatia, G., Vijayakumar, S., and Schaal, S. (2000). On-line learning for humanoid robot systems. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 191–198.

Corke, P. I. (1996). A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1):24–32.

Cortes, J., Martinez, S., Ostrowski, J., and McIsaac, K. A. (2001). Optimal gaits for dynamic robotic locomotion. *The International Journal of Robotics Research*, 20(9):707–728.

Csato, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668.

Daerden, F. (1999). *Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements.* Phd dissertation, Vrije Universiteit, Brussel.

Davidson, P. R. and Wolpert, D. M. (2005). Widespread access to predictive models in the motor system: a short review. *Journal of Neural Engineering*, 2:313–319.

Diedrichsen, J. (2007). Optimal task-dependent changes of bimanual feedback control and adaptation. *Current Biology*, 17(19):1675–1679.

D'Souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse kinematics. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–303.

Dumont, G. A. and Huzmezan, M. (2002). Concepts, methods and techniques in adaptive control. In *Proceedings of the American Control Conference*, volume 2, pages 1137–1150.

Dyer, P. and McReynolds, S. R. (1970). *The computation and theory of optimal control*. Academic Press, New York.

Engelbrecht, S. E. (2001). Minimum principles in motor control. *Journal of Mathematical Psychology*, 45(3):497–542.

Faisal, A. A., Selen, L. P. J., and Wolpert, D. M. (2008). Noise in the nervous system. *Nature Reviews Neuroscience*, 9:292–303.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703.

Flentge, F. (2006). Locally weighted interpolating growing neural gas. *IEEE Transactions on Neural Networks*, 17(6):1382–1393.

Franklin, D., Burdet, E., Tee, K. P., Osu, R., Chew, C. M., Milner, T. E., and Kawato, M. (2008). Cns learns stable, accurate, and efficient movements using a simple algorithm. *The Journal of Neuroscience*, 28(44):11165–11173.

Franklin, D. W., G., L., Milner, T. E., Osu, R., Burdet, E., and Kawato, M. (2007). Endpoint stiffness of the arm is directionally tuned to instability in the environment. *The Journal of Neuroscience*, 27(29):7705–7716.

Ganesh, G., Haruno, M., Kawato, M., and Burdet, E. (2010). Motor memory and local minimization of error and effort, not global optimization. *Journal of Neurophysiology*, 104(1):382–390.

Garcia, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348.

Gomi, H. and Kawato, M. (1996). Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science*, 272(5258):117–120.

Grebenstein, M. and van der Smagt, P. (2008). Antagonism for a highly anthropomorphic hand-arm system. *Advanced Robotics*, 22(1):39–55.

Gribble, P. L., Mullin, L. I., Cothros, N., and Mattar, A. (2003). Role of cocontraction in arm movement accuracy. *Journal of Neurophysiology*, 89(5):2396–2405.

Gribble, P. L. and Ostry, D. J. (1998). Independent coactivation of shoulder and elbow muscles. *Experimental Brain Research*, 123(3):355–360.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.

Haruno, M. and Wolpert, D. M. (2005). Optimal control of redundant muscles in step-tracking wrist movements. *Journal of Neurophysiology*, 94:4244–4255.

He, J., Levine, W. S., and Loeb, G. E. (1991). Feedback gains for correcting small perturbations to standing posture. In *IEEE Transactions on Automatic Control*, volume 36, pages 322–332.

Hirzinger, G., Butterfaß, J., Fischer, M., Grebenstein, M., Hähnle, M., Liu, H., Schaefer, I., Sporer, N., Schedl, M., and Koeppe, R. (2001). A new generation of lightweight robot arms and multifingered hands. In *ISER '00: Experimental Robotics VII*, pages 569–570, London, UK. Springer-Verlag.

Hogan, N. (1984). Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on Automatic Control*, 29(8):681– 690.

Hollerbach, J. M. and Suh, K. C. (1985). Redundancy resolution of manipulators through torque optimization. In *Proceedings of the IEEE 1985 International Conference on Robotics and Automation (ICRA)*, pages 1016–1021.

Holly, S. and Hughes, H. A. (1989). *Optimal control, expectations and uncertainty*. Cambridge University Press.

Hurst, J. W., Chestnutt, J., and Rizzi, A. (2004). An actuator with mechanically adjustable series compliance. Technical Report CMU-RI-TR-04-24, Robotics Institute, Carnegie Mellon University.

Izawa, J., Rane, T., Donchin, O., and Shadmehr, R. (2008). Motor adaptation as a process of reoptimization. *The Journal of Neuroscience*, 28(11):2883–2891.

Jacobson, D. H. and Mayne, D. Q. (1970). *Differential dynamic programming*. Elsevier Science Ltd., New York.

Jones, K. E., Hamilton, A. F., and Wolpert, D. M. (2002). Sources of signal-dependent noise during isometric force production. *Journal of Neurophysiology*, 88(3):1533–1544.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

Kalman, R. E. (1960a). Contributions to the theory of optimal control. In *Bo. de Soc. Math. Mexicana*.

Kalman, R. E. (1960b). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(1):35–45.

Kappen, H. (2005). Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, page P11011.

Kappen, H. (2007). An introduction to stochastic control theory, path integrals and reinforcement learning. *In 9th Granada seminar on Computational Physics: Computational and Mathematical Modeling of Cooperative Behavior in Neural Systems*, pages 149–181.

Katayama, M. and Kawato, M. (1993). Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse model. *Biological Cybernetics*, 69(5-6):353–362.

Kawato, M. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3):169–185.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727.

Kirk, D. E. (1970). *Optimal control theory: An introduction*. Prentice-Hall.

Klanke, S., Vijayakumar, S., and Schaal, S. (2008). A library for locally weighted projection regression. *Journal of Machine Learning Research*, 9:623–626.

Koditschek, D. E. and Rimon, E. (1990). Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442.

Kuo, A. (1995). An optimal control model for analyzing human postural balance. *IEEE Transactions on Biomedical Engineering*, 42:87–101.

Laffranchi, M., Tsagarakis, N. G., and Caldwell, D. G. (2010). A variable physical damping actuator (vpda) for compliant robotic joints. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Lametti, D. R., Houle, G., and Ostry, D. J. (2007). Control of movement variability and the regulation of limb impedance. *Journal of Neurophysiology*, 98:3516–3524.

Laursen, B., Jensen, B. R., and Sjogaard, G. (1998). Effect of speed and precision demands on human shoulder muscle electromyography during a repetitive task. *European Journal of Applied Physiology and Occupational Physiology*, 78(6):544–548.

Li, W. (2006). *Optimal control for biological movement systems*. Phd dissertation, University of California, San Diego.

Li, W. and Todorov, E. (2004). Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Li, W. and Todorov, E. (2007). Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):1439–1453.

Lin, T. C. and Arora, J. S. (1991). Differential dynamic programming for constrained optimal control. *Computational Mechanics*, 9(1):27–40.

Liu, D. and Todorov, E. (2007). Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *The Journal of Neuroscience*, 27(35):9354–9368.

Liu, D. and Todorov, E. (2009). Hierarchical optimal control of a 7-dof arm model. In *Proceedings of the 2nd IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 50–57.

Lockhart, D. B. and Ting, L. H. (2007). Optimal sensorimotor transformations for balance. *Nature Neuroscience*, 10:1329–1336.

Mayeda, H., Osuka, K., and Kangawa, A. (1984). A new identification method for serial manipulator arms. In *Proceedings of the 9th IFAC World Congress*, pages 2429–2434.

McIntyre, J., Mussa-Ivaldi, F. A., and Bizzi, E. (1996). The control of stable postures in the multijoint arm. *Experimental Brain Research*, 110(2):248–264.

McShane, E. J. (1939). On multipliers for lagrange problems. *American Journal of Mathematics*, 61(4):809–819.

Miall, R. and Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, 9(8):1265–1279.

Migliore, S. A., Brown, E. A., and DeWeerth, S. P. (2005). Biologically inspired joint stiffness control. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4508–4513.

Milner, T. E. and Franklin, D. W. (2005). Impedance control and internal model use during the initial stage of adaptation to novel dynamics in humans. *Journal of Physiology*, 567(2):651–664.

Mitrovic, D., Klanke, S., and Vijayakumar, S. (2008a). Adaptive optimal control for redundantly actuated arms. In *Proceedings of the 10th International Conference on Simulation of Adaptive Behaviour (SAB)*, Osaka, Japan.

Mitrovic, D., Klanke, S., and Vijayakumar, S. (2008b). Optimal control with adaptive internal dynamics models. In *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Madeira, Portugal.

Morimoto, J. and Atkeson, C. G. (2003). Minimax differential dynamic programming: An application to robust biped walking. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 1563–1570.

Müller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201.

Murray, D. M. and Yakowitz, S. J. (1984). Differential dynamic programming and newton's method for discrete optimal control problems. *Journal of Optimization Theory and Applications*, 43(3):395–414.

Nakamura, Y. (1990). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Nakamura, Y. and Hanafusa, H. (1987). Optimal redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(1):32–42.

Nakanishi, J., Farrell, J. A., and Schaal, S. (2005). Composite adaptive control with locally weighted statistical learning. *Neural Networks*, 18(1):71–90.

Narendra, K. S. and Annaswamy, A. M. (1989). *Stable adaptive systems*. Prentice Hall, Upper Saddle River, NJ, USA.

Nelson, W. L. (1983). Physical principles for economies of skilled movements. *Biological Cybernetics*, 46:135–147.

Nenchev, D. N. (1989). Redundancy resolution through local optimization: A review. *Journal of Robotic Systems*, 6(6):769 – 798.

Nguyen-Tuong, D., Peters, J., and Seeger, M. (2008a). Computed torque control with nonparametric regressions techniques. In *Proceedings of the 2008 American Control Conference (ACC 2008)*, pages 212–217.

Nguyen-Tuong, D., Peters, J., Seeger, M., and Schölkopf, B. (2008b). Learning inverse dynamics: a comparison. In Verleysen, M., editor, *16th European Symposium on Artificial Neural Networks (ESANN)*, pages 13–18.

Nguyen-Tuong, D., Seeger, M., and Peters, J. (2008c). Local gaussian process regression for real time online model learning and control. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1193–1200.

Osu, R. and Gomi, H. (1999). Multijoint muscle regulation mechanisms examined by measured human arm stiffness and emg signals. *Journal of Neurophysiology*, 81(4):1458–1468.

Osu, R., Kamimura, N., Iwasaki, H., Nakano, E., Harris, C. M., Wada, Y., and Kawato, M. (2004). Optimal impedance control for task achievement in the presence of signal-dependent noise. *Journal of Neurophysiology*, 92(2):1199–1215.

Owens, D. H. and Hätönen, J. (2005). Iterative learning control - an optimization paradigm. *Annual Reviews in Control*, 29(1):57–70.

Özkaya, N. and Nordin, M. (1991). *Fundamentals of biomechanics: equilibrium, motion, and deformation*. Van Nostrand Reinhold, New York.

Paluska, D. and Herr, H. (2006). The effect of series elasticity on actuator power and work output: Implications for robotic and prosthetic joint design. *Robotics and Autonomous Systems*, 54(8):667–673.

Pfeiffer, F. and Glocker, C. (1996). *Multibody dynamics with unilateral contacts.* Wiley-VCH.

Pontryagin, L., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. (1961). *The mathematical theory of optimal processes.* Moscow.

Pratt, G., Williamson, M. W., Dillworth, P., Pratt, J., Ulland, K., and Wright, A. (1995). Stiffness isn't everything. In *Proceedings of the 4th International Symposium on Experimental Robotics (ISER '95)*, Stanford, California.

Pratt, G. A. and Williamson, M. M. (1995). Series elastic actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 399–406.

Rasmussen, C. E. and Williams, C. (2006). *Gaussian processes for machine learning*. MIT Press.

Sahar, G. and Hollerbach, J. M. (1986). Planning of minimum-time trajectories for robot arms. *The International Journal of Robotics Research*, 5(3):90–100.

Salaün, C., Padois, V., and Sigaud, O. (2010). *From Motor Learning to Interaction Learning in Robots*, chapter Learning forward models for the operational space control of redundant robots, pages 169–192. Studies in Computational Intelligence. Springer Berlin / Heidelberg.

Schaal, S. (2002). *The handbook of brain theory and neural networks*, chapter Learning Robot Control, pages 983–987. MIT Press, Cambridge, MA.

Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084.

Scholz, J. P. and Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126:289–306.

Scott, S. H. (2004). Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5:532–546.

Scott, S. H. (2008). Inconvenient truths about neural processing in primary motor cortex. *The Journal of Physiology*, 586(5):1217–1224.

Selen, L. P. J. (2007). *Impedance modulation: A means to cope with neuromuscular noise.* Phd dissertation, Vrije Universiteit.

Selen, L. P. J., Beek, P. J., and Dieen, J. H. (2005). Can co-activation reduce kinematic variability? a simulation study. *Biological Cybernetics*, 93(5):373–381.

Selen, L. P. J., Franklin, D. W., and Wolpert, D. M. (2009). Impedance control reduces instability that arises from motor noise. *The Journal of Neuroscience*, 29(40):12606–12616.

Shadmehr, R. and Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Experimental Brain Research*, 185(3):359–381.

Shadmehr, R. and Mussa-Ivaldi, F. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neuroscience*, 14(5):3208–3224.

Shadmehr, R. and Wise, S. P. (2005). *The computational neurobiology of reaching and pointing*. MIT Press.

Spong, M. W. and Vidyasagar, M. (1989). *Robot dynamics and control*. John Wiley and Sons.

Stengel, R. F. (1994). *Optimal control and estimation*. Dover Publications, New York.

Stewart, D. E. (2000). Rigid body dynamics with friction and impact. *SIAM Review*, 42(1):3–39.

Stewart, D. E. and Anitescu, M. (2010). Optimal control of systems with discontinuous differential equations. *Numerische Mathematik*, 114(4):653–695.

Stoer, J. and Bulirsch, R. (1980). *Introduction to Numerical Analysis*. Springer-Verlag, New York.

Sussmann, H. and Willems, J. (1997). 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, pages 32–44.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA.

Suzuki, M., Douglas, M. S., Gribble, P. L., and Ostry, D. J. (2001). Relationship between cocontraction, movement kinematics and phasic muscle activity in single-joint arm movement. *Experimental Brain Research*, 140(2):171–181.

Tassa, Y., Erez, T., and Smart, W. D. (2007). Receding horizon differential dynamic programming. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20.

Tassa, Y. and Todorov, E. (2010). Stochastic complementarity for local control of continuous dynamics. *Under review, retrieved from: www.cs.washington.edu/homes/todorov/papers.htm*.

Tedrake, R. L. (2004). *Applied optimal control for dynamically stable legged locomotion*. Phd dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Tee, K. P., Burdet, E., Chew, C. M., and Milner, T. E. (2004). A model of force and impedance in human arm movements. *Biological Cybernetics*, 90(5):368–375.

Theodorou, E. A., Buchli, J., and Schaal, S. (2010a). Reinforcement learning of motor skills in high dimensions: a path integral approach. In *IEEE International Conference of Robotics and Automation (ICRA)*.

Theodorou, E. A., Tassa, Y., and Todorov, E. (2010b). Stochastic differential dynamic programming. In *Proceedings of the American Control Conference (ACC 2010)*.

Thoroughman, K. A. and Shadmehr, R. (1999). Electromyographic correlates of learning an internal model of reaching movements. *The Journal of Neuroscience*, 19(5):8573–8588.

Thrun, S. (2000). Monte carlo POMDPs. In Solla, S. A., Leen, T. K., and Müller, K. R., editors, *Advances in Neural Information Processing Systems 12*, pages 1064–1070. MIT Press.

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):1084–1108.

Todorov, E. (2006). Optimal Control Theory. In Doya, K., editor, *Bayesian Brain: Probabilistic Approaches to Neural Coding*, pages 269–298. MIT Press.

Todorov, E. (2008). General duality between optimal control and estimation. In *47th IEEE Conference on Decision and Control*.

Todorov, E. and Ghahramani, Y. (2004). Analysis of the synergies underlying complex hand manipulation. In *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4637–4640.

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235.

Todorov, E. and Jordan, M. I. (2003). A minimal intervention principle for coordinated movement. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 27–34.

Todorov, E. and Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005 American Control Conference*, volume 1, pages 300–306.

Todorov, E., Li, W., and Pan, X. (2005). From task parameters to motor synergies: A hierarchical framework for approximately optimal control of redundant manipulators. *Journal of Robotic Systems*, 22(11):691–710.

Todorov, E. and Tassa, Y. (2009). Iterative local dynamic programming. In *Proceedings of the 2nd IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 90–95.

Tonietti, G., Schiavi, R., and Bicchi, A. (2005). Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *IEEE International Conference Robotics and Automation (ICRA)*, pages 526–531.

Toussaint, M. and Vijayakumar, S. (2005). Learning discontinuities with products-of-sigmoids for switching between local models. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 904–911.

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectories in human multijoint arm movements: minimum torque-change model. *Biological Cybernetics*, 61(2):89–101.

van Beers, R. J. (2009). Motor learning is optimally tuned to the properties of motor noise. *Neuron*, 63(3):406–417.

van Beers, R. J., Haggard, P., and Wolpert, D. M. (2004). The role of execution noise in movement variability. *Journal of Neurophysiology*, 91:1050–1063.

van Ham, R., Sugar, T., Vanderborght, B., Hollander, K., and Lefeber, D. (2009). Compliant actuator designs. *IEEE Robotics and Automation Magazine*, 16(3):81 – 94.

van Ham, R., Vanderborght, B., van Damme, M., Verrelst, B., and Lefeber, D. (2007). Maccepa, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, 55(10):761 – 768.

Vanderborght, B., Van Ham, R., Lefeber, D., Sugar, T. G., and Hollander, K. W. (2009). Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators. *The International Journal of Robotics Research*, 28(1):90–103.

Vijayakumar, S., D'Souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634.

Vijayakumar, S., D'Souza, A., Shibata, T., Conradt, J., and Schaal, S. (2002). Statistical learning for humanoid robots. *Autonomous Robots*, 12(1):55–69.

Vijayakumar, S. and Wu, S. (1999). Sequential support vector classifiers and regression. In *Proceedings of the International Conference on Soft Computing*, pages 610–619.

Wang, T., Dordevic, G. S., and Shadmehr, R. (2001). Learning the dynamics of reaching movements results in the modification of arm impedance and long-latency perturbation responses. *Biological Cybernetics*, 85(6):437–448.

Wierzbicka, M. M., Wiegner, A. W., and Shahani, B. T. (1985). Role of agonist and antagonist muscles in fast arm movements in man. *Experimental Brain Research*, 63(2):331–340.

Wolf, S. and Hirzinger, G. (2008). A new variable stiffness design: Matching requirements of the next robot generation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1741–1746.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.

Wong, J., Wilson, E. T., Malfait, N., and Gribble, P. L. (2009). Limb stiffness is modulated with spatial accuracy requirements during movement in the absence of destabilizing forces. *Journal of Neurophysiology*, 101:1542–1549.

Yakowitz, S. J. (1986). The stagewise kuhn-tucker condition and differential dynamic programming. *IEEE Transactions on Automatic Control*, 31(1):25–30.

Zinn, M., Khatib, O., Roth, B., and Salisbury, J. K. (2004). Playing it safe. *IEEE Robotics and Automation Magazine*, 11(2):12–21.