

© 2012 by Anne-Sophie Gaby Marthe Puydupin.

INVERSE OPTIMIZATION OF DISCRETE-TIME SYSTEMS APPLIED TO  
HUMAN LOCOMOTION

BY

ANNE-SOPHIE GABY MARTHE PUYDUPIN

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Aerospace Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Advisor:

Professor Timothy W. Bretl

# Abstract

The problem of inverse optimization is to find the objective function that is being minimized, given knowledge of the constraints and observations of local minima. In this thesis, we consider the special case in which the objective function is a linear combination of known basis functions weighted by unknown parameters. Therefore the aim is to recover the weights governing the objective function. We propose a solution approach in this case that is based on the application of necessary conditions for optimality. We begin with a review of how these necessary conditions arise, with a particular focus on the relationship between duality theory and inverse optimization. We then proceed to describe our solution approach. Finally, we apply our approach to find a model of goal-directed human walking from experimental data with human subjects.

*To my Mother and my Grandmother.*

# Acknowledgments

This project would not have been possible without the support of many people. First of all, I would like to thank my current advisor, Professor Timothy W. Bretl, for his time, his help, his support and advice. His input has always been very valuable, and I am grateful I was given the opportunity to work him. I am also very grateful to my undergraduate advisor at the Virginia Polytechnic Institute and State University, Professor Gary Don Seidel. I had the pleasure of working with him for two years, and it is thank to him that I started working on research. I see him as my mentor and my friend, and I would like to thank him for always being here for me. I also would like to thank the professors I have encountered at the University of Illinois, in particular Professor P.R. Kumar for doing an amazing job at teaching Introduction to Optimization, and Professor Petros Voulgaris, for our conversations in the corridors of CSL. I was happy to be part of the RMS Lab for two years, and I would like to thank my labmates for their support: Navid Aghasadeghi, Abdullah Akce, Aadeel Akhtar, Aaron Becker, Miles Johnson, Dennis Matthews, Jamie Norton, Colin Das, Cem Onyuksel and Carlos Orduno. Finally, I would like to thank my friends, in particular Pritam Sukumar, Thomas Galpin, Hemant Kowshik and Kunal Srivastava for making my stay in Champaign more exciting.

# Table of Contents

|   |            |
|---|------------|
| <b>List of Figures . . . . .</b>                        | <b>vii</b> |
| <b>Chapter 1 Introduction . . . . .</b>                 | <b>1</b>   |
| 1.1 Optimization . . . . .                              | 1          |
| 1.2 Inverse Optimization . . . . .                      | 2          |
| 1.3 Clarifications and Restrictions . . . . .           | 3          |
| 1.4 A Brief History . . . . .                           | 3          |
| 1.5 Motivations . . . . .                               | 5          |
| 1.6 Objectives . . . . .                                | 6          |
| 1.7 Organization . . . . .                              | 6          |
| <b>Chapter 2 Optimization . . . . .</b>                 | <b>7</b>   |
| 2.1 Introduction . . . . .                              | 7          |
| 2.2 Unconstrained Optimization . . . . .                | 7          |
| 2.2.1 Mathematical formulation . . . . .                | 8          |
| 2.2.2 Necessary and Sufficient Conditions . . . . .     | 9          |
| 2.2.3 Unconstrained optimization algorithm . . . . .    | 10         |
| 2.3 Constrained Optimization . . . . .                  | 11         |
| 2.3.1 Mathematical formulation . . . . .                | 12         |
| 2.3.2 KKT Necessary and Sufficient Conditions . . . . . | 13         |
| 2.3.3 Constrained optimization algorithm . . . . .      | 14         |
| 2.4 Convex Optimization . . . . .                       | 16         |
| 2.4.1 Necessary and Sufficient conditions . . . . .     | 16         |
| 2.4.2 Least-squares problems . . . . .                  | 17         |
| 2.5 Duality . . . . .                                   | 18         |
| 2.5.1 Inequality constraints . . . . .                  | 18         |
| 2.5.2 Equality constraints . . . . .                    | 19         |
| 2.5.3 Weak Duality . . . . .                            | 20         |
| 2.5.4 Strong Duality . . . . .                          | 20         |
| 2.5.5 Optimality conditions . . . . .                   | 21         |
| 2.5.6 Duality and Inverse Optimization . . . . .        | 21         |
| <b>Chapter 3 Inverse Optimization . . . . .</b>         | <b>22</b>  |
| 3.1 Problem Formulation . . . . .                       | 22         |
| 3.2 Bilevel Approach . . . . .                          | 23         |
| 3.2.1 Assumptions . . . . .                             | 23         |
| 3.2.2 Formulation . . . . .                             | 23         |
| 3.2.3 Solving technique . . . . .                       | 23         |
| 3.3 New Approach . . . . .                              | 24         |
| 3.3.1 Assumptions . . . . .                             | 24         |
| 3.3.2 Formulation . . . . .                             | 24         |
| 3.3.3 Necessary conditions for optimality . . . . .     | 25         |

|                   |  |           |
|-------------------|--|-----------|
| 3.3.4             | Residual Functions . . . . .   | 25        |
| 3.3.5             | Solving techniques . . . . .   | 27        |
| 3.4               | Duality and Inverse Optimization – More details . . . . .            | 27        |
| 3.4.1             | Theory: . . . . .  | 28        |
| 3.4.2             | Our approach: . . . . .  | 29        |
| <b>Chapter 4</b>  | <b>Applications . . . . .</b>  | <b>31</b> |
| 4.1               | Application to Simulated Data . . . . .                              | 31        |
| 4.1.1             | The Unicycle Model . . . . .   | 31        |
| 4.1.2             | Derivation of the Residual Function for the Unicycle Model . . . . . | 32        |
| 4.1.3             | Results for Data Generated in Simulation . . . . .                   | 33        |
| 4.2               | Application to Experimental Data . . . . .                           | 34        |
| 4.2.1             | The Experiment . . . . .   | 35        |
| 4.2.2             | Notes on the Choice of the Model . . . . .                           | 35        |
| 4.2.3             | Recovered Trajectories . . . . .                                     | 35        |
| 4.2.4             | Results for the Experimental Data . . . . .                          | 37        |
| <b>Chapter 5</b>  | <b>Conclusions and Future Work . . . . .</b>                         | <b>39</b> |
| 5.1               | Conclusions . . . . .  | 39        |
| 5.2               | Future Work . . . . .  | 40        |
| <b>References</b> | <b>. . . . .</b>   | <b>41</b> |

# List of Figures

|     |  |    |
|-----|--|----|
| 1.1 | Illustration of an optimization problem. The system considered is a taxi driver, going from point A to point B. The red path is the minimum length path, while the green path has the minimum number of turns. These are examples of possible objective functions the driver could want to minimize. . . . .   | 1  |
| 1.2 | Illustration of an inverse optimization problem. We are given the system: taxi driver, and one observation of the system: the red path taken from point A to point B. The aim is to find what was the objective function being minimized by the system. In this case the solution would be that the driver wanted to minimize the distance between A and B, according to Fig. 1.1. . . . .                           | 2  |
| 1.3 | Illustration of the Brachistochrone problem. It was one of the first optimization problem posed, and Newton solved it in 1696. The problem was to find the trajectory which connects point a to point b so that the travel time of the particle is minimized. . . . .  | 4  |
| 2.1 | Illustration of a local minimum and a global minimum for an unconstrained problem. . . . .   | 8  |
| 2.2 | Figure representing the contours of a two-dimensional quadratic objective function $f$ and the sequence of iterations of the gradient descent method. The steps zigzag toward the solution. .  | 11 |
| 2.3 | Illustration of a local minimum and a global minimum for a constrained problem over the set $X$ . . . . .  | 12 |
| 4.1 | Example of a noiseless trajectory simulated using the unicycle model (black line), along with four different noisy measurements. Each set of measurements has been obtained by averaging 60 noisy trajectory measurements. The amount of noise varies and is characterized by the value of the standard deviation $p = 1e^{-5}, 1e^{-4}, 1e^{-3}$ and $1e^{-2}$ . . . . .  | 33 |
| 4.2 | Error between the predicted value of the parameter $c$ obtained for each of the four averaged noisy measurements, and the value used during simulation. The approach used is able to predict the correct value of $c$ ( $c = 3.7$ ), even when very noisy measurements are used (i.e. for $p = 1e^{-2}$ ). . . . .   | 34 |
| 4.3 | Recovered trajectories before and after performing gradient descent to match the end points. The blue trajectory is the observed one obtained from experimental data. The black trajectory is the predicted result obtained before performing gradient descent and the red trajectory is the predicted result after performing gradient descent. . . . .   | 36 |
| 4.4 | In this figure, observed and predicted trajectories are projected on the x-y plane. Blue curves represent observed trajectories obtained from experimental data. Red dashed curves show predicted results obtained when using only a single observation to recover the value of parameter $c$ . Black dashed curves show predicted results obtained when using 15 observations to recover the value of $c$ . . . . . | 38 |



# Chapter 1

## Introduction

We will first define the terminology used in this thesis, and illustrate the problems with a simple example. We will then glance through the history of optimization and inverse optimization, and analyse the different techniques used in literature. Finally the last section of the introduction will present the organization of the next chapters.

### 1.1 Optimization

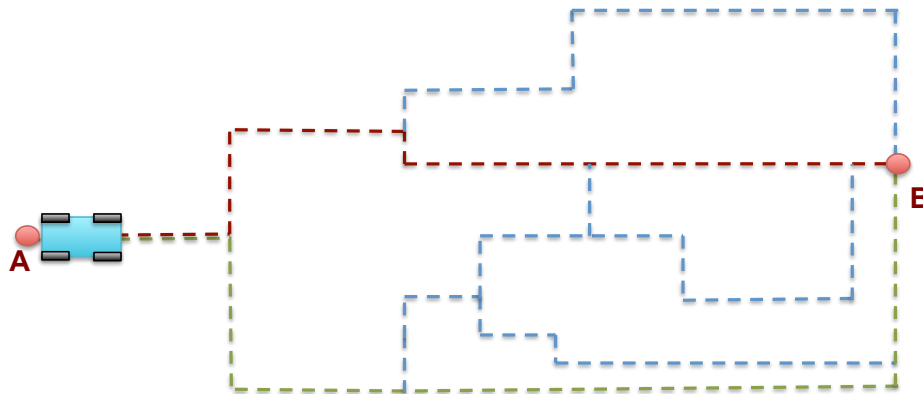


Figure 1.1: Illustration of an optimization problem. The system considered is a taxi driver, going from point A to point B. The red path is the minimum length path, while the green path has the minimum number of turns. These are examples of possible objective functions the driver could want to minimize.

Optimizing a system refers to choosing a set of parameters that makes the system the most effective, according to some criteria. In this thesis we are interested in mathematical optimization, and it consists in minimizing or maximizing an objective function, given a model of the system to be optimized. Optimization is widely used in various areas such as economics, operation research and engineering.

Given an objective function and a set of constraints defining a system, the problem of optimization is to find the set of parameters that optimizes the objective function while satisfying the constraints. A simple example is given to illustrate this problem. Consider a taxi-driver, who needs to drive from point A to point

B, as sketched in Fig. 1.1. The driver might be interested in choosing the path that is the shortest between point A and point B (red path on Fig. 1.1 ), or he might want to chose the path with the least turns (green path on Fig. 1.1 ). These are examples of objective functions that the driver could want to minimize, among others. In this case, solving the optimization problem corresponds to choosing the path that minimizes the desired objective function.

## 1.2 Inverse Optimization

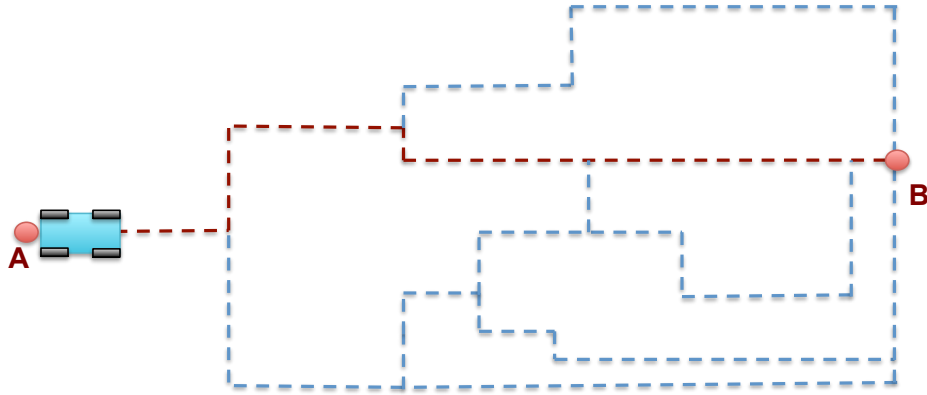


Figure 1.2: Illustration of an inverse optimization problem. We are given the system: taxi driver, and one observation of the system: the red path taken from point A to point B. The aim is to find what was the objective function being minimized by the system. In this case the solution would be that the driver wanted to minimize the distance between A and B, according to Fig. 1.1.

Inverse optimization is often used to study physical systems, for which we are given observations and we want to understand their behavior. Given observations and a set of constraints defining a system, the problem of inverse optimization is to find the objective function that is being optimized by the system, assuming the system is behaving in an optimal manner.

Considering the previous example in Fig. 1.1, here solving the inverse optimization problem consists in finding what was the driver's intent, given observations of the optimal path chosen. This is illustrated in Fig. 1.2. Therefore in this case, knowing that the driver chose the red path over the green path (represented in Fig. 1.1), the solution to the inverse optimization problem is that the driver's intent was to minimize the distance between A and B.

Inverse optimization problems are harder to solve than optimization problems due to their ill-posed nature. If a problem is ill-posed, it implies that:

- The solution may not exist.

- Several solutions may exist, i.e. the solution is not unique.
- The solution may not continuously depend on the input data.

## 1.3 Clarifications and Restrictions

Optimization is a broad field, which includes optimal control as a subfield. Optimal control problems are a specific type of optimization problems in which we are interested in solving for a control policy. Therefore the objective function might not only depend on parameters but also on control terms. In this thesis, we focus our theoretical discussion on optimization since optimal control is only a subset of optimization. The application considered is however an example of inverse optimal control, since we are interested in recovering both parameters and control laws, but we will see that the same techniques can be used.

It is also important to state the restrictions of this thesis right from the start. The problems we are considering are deterministic, discrete-time optimization/inverse optimization problems.

Discrete-time problems are problems in which the set of constraints consists of algebraic equations and the variables are drawn from a finite set. This is in contrast with continuous-time problems, in which the feasible set is usually infinite and the constraints are expressed in a continuous way either using differential equation or integration functions. Continuous-time problems are usually easier to solve than discrete-time problems if the functions are smooth, since solutions at a specific point can be derived using information given at points close-by. In discrete-time problems, there is no guarantee that the behavior from one point to another (even close-by) will be similar.

Deterministic problems are problems in which no noise is added to the system itself or to the observations of the system. In real life, it is sometimes hard to model or observe a system with precision, and this would result in uncertainties that may be quantified. Problems in which the variables may vary due to uncertainties are called stochastic problems. In contrast, deterministic problems assume perfect knowledge of the model and perfect observations of the system. The techniques used to solve stochastic optimization problems are different from deterministic optimization problems, and will not be treated in this thesis.

## 1.4 A Brief History

One of the earliest problem known in optimization is the *Brachistochrone Problem*, that was first solved by Newton in 1696. The problem consists of a particle moving along some curve under the influence of gravity, as shown in Fig. 1.3. The particle starts at rest from fixed point a and travels to fixed point b. The

optimization problem is to find the shape of the trajectory which connects point a and b so that the particle travels in minimum time. At the time, this problem and its solution lead to the beginning of Calculus of Variations. Since then different formulations of the brachistochrone problem in different spaces and fields have been proposed [2,3].

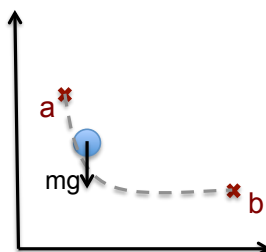


Figure 1.3: Illustration of the Brachistochrone problem. It was one of the first optimization problem posed, and Newton solved it in 1696. The problem was to find the trajectory which connects point a to point b so that the travel time of the particle is minimized.

In 1781, Monge introduces the *Transportation problem*, which is still commonly studied in economics and network theory. The transportation problem consists in finding the optimal shipping pattern between origin and destination nodes that satisfy all the requirements and minimize the total shipping cost [4, 5]. In 1806, Legendre gave the first results on least squares method for fitting a curve to the data available, but Gauss published similar results in 1821 in *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae* [6]. Least-squares problems are very commonly used for parameter estimation and data fitting, and we will discuss these problems in greater details in the next chapters. In 1847 Cauchy presents the *gradient method*, also known as the *steepest descent method*. It is an iterative method that is used to solve unconstrained optimization problems [7]. This method will also be discussed in greater details in the next chapters.

A century later, several major findings were made. In 1947 Dantzig presents the *Simplex method* for solving linear programming problems. In 1939 Karush finishes his Master thesis work on necessary conditions of optimality, but it is not until Kuhn and Tucker publish similar work in 1951 that the necessary conditions become well known to public [8]. In 1956 Pontryagin develops the maximum principle, which consists in minimizing the Hamiltonian function associated to the problem [9]. And finally in 1957 Bellman develops the optimality principle for dynamic programming using value functions. Bellman's Principle of Optimality states that "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [10].

These are some of the major discoveries and the pioneers in the history of optimization. Even though some

methods were developed more than a century ago, they are still commonly used nowadays in optimization, and most of the methods introduced here will be discussed in greater details in the next sections.

## 1.5 Motivations

The applications of optimization are very wide, ranging from economics to engineering. Many optimization problems arise in economics, for example: maximize utility, minimize costs, maximize profits, maximize welfare, minimize the loss – these are all optimization problems [11, 12]. The vocabulary used is specific to economics, but the solving techniques are the same as for other areas such as engineering. In the engineering field, problems such as: maximize efficiency, minimize energy consumption, minimize structural weight, minimize a process time (often related to minimizing cost) – are all optimization problems [13]. These are just a few examples, but it shows that studying optimization has critical impacts on a variety of fields.

Inverse optimization has also been a topic of study for more than half a century and was introduced first in [14–20]. The studies in [14, 15, 18, 20] give a formulation of the inverse problem and state the necessary conditions of optimality, while [19] looked at applications. The first papers on applications of inverse optimization were mostly about transportation network and geophysical problems. In [19], they apply inverse optimization to the capacitated plant location problem, which is a common transportation problem [21]. In [22], the author used inverse optimization to study geophysical systems, and Tarantola explained that “to solve the inverse problem is to infer the values of the model parameters from given observed values of the observable parameters”. More recent studies have applied inverse optimization techniques to a wide range of applications. For example, Ziebart et al [23] apply inverse optimization to explain why taxi drivers make specific route choices, based on GPS observations. Nielsen and Jensen [24] learn the utility of a human decision maker given inconsistent observations of behavior. Yepes et al [25] predict trajectories flown by human pilots in air traffic control. Trautman and Krause [26] focus on predicting future trajectories taken by human walkers to enable robot navigation in crowds. These are only few out of the many applications of inverse optimization.

Therefore, the motivations for studying optimization and inverse optimization are that they are widely used in different fields, and that any improvement that can be made to solve optimization or inverse optimization problems can have very large impacts. Even though the ground theory of optimization and inverse optimization is more than a century old, different solving techniques are proposed every year along with different applications, which is what makes optimization and inverse optimization so interesting.

## 1.6 Objectives

This thesis has two objectives. The first one is to introduce the background in optimization theory that is necessary for the understanding of the inverse optimization problem treated in this thesis. To do so, we will mostly be using four references that are well-known and very often cited in the domain of optimization [1, 4, 27, 28]. Each of them propose an helpful balance between theory and algorithms for numerical optimization, nonlinear programming and convex optimization.

The second objective is to introduce a new method for solving inverse optimization problems. The approach presented in this thesis differs from the other existing approaches, and uses conditions of optimality as a solving approach. We will discuss this method in details in the next chapters, and we will also use it to model human locomotion as an application.

## 1.7 Organization

The following chapter presents the background in optimization theory necessary for understanding inverse optimization. It briefly covers unconstrained, constrained optimization problems and their respective necessary conditions of optimality. It also covers the sufficient conditions of optimality for convex problems, and the theory of duality.

The next chapter focuses on the theory of inverse optimization. One section covers the existing approach, and the other section introduces a new approach for solving inverse optimization problems.

Finally, in the last chapter we apply our new solving approach to real data, in order to obtain a model for human locomotion.

The last chapter of this thesis lays down the conclusions and outlines ideas for future work.

# Chapter 2

## Optimization

### 2.1 Introduction

An optimization process is divided into two subproblems:

- Modeling: definition of objective function, variables and constraints.
- Solving: once the problem is formulated properly, use an algorithm to solve for the variables.

The problem of modeling necessitates a lot of attention, since if the model is too simplistic, the solution might not give any useful insight, and if the model is too complicated, it might not be possible to solve it. The objective function is a function that measures the performance of the system. The objective function is expressed as a function of unknown variables, that can be restricted by a set of constraints.

The solving part consists in using an algorithm to solve for the unknown variables. There exists a variety of algorithms that solves optimization problems. The choice of the algorithm depends on the type of problem being solved.

This chapter will present the optimization theory background necessary for the understanding of chapters 3 and 4. We will introduce unconstrained and constrained optimization problems, since they are the problems considered for the inverse optimization study. We will also introduce necessary conditions of optimality, as they are part of the new method for solving inverse optimization problems. Finally, we will introduce the theory of duality, since inverse optimization and duality are closely related.

### 2.2 Unconstrained Optimization

The new approach for solving inverse optimization problems that will be discussed in section 3.3 consists in simplifying a constrained optimization problem into an unconstrained problem. It is therefore necessary to first briefly introduce what unconstrained optimization is.

Unconstrained optimization problems have the following properties:

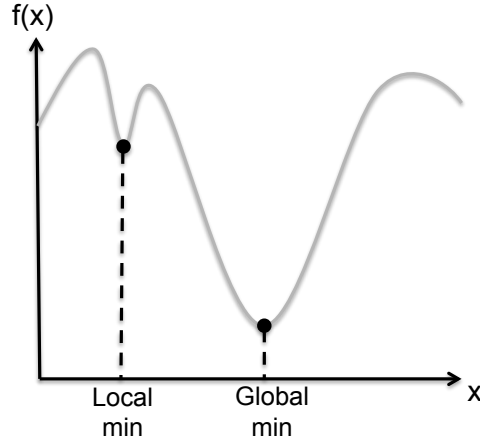


Figure 2.1: Illustration of a local minimum and a global minimum for an unconstrained problem.

- The objective function depends on unknown variables.
- These variables are not restricted.

### 2.2.1 Mathematical formulation

The general form of an unconstrained optimization problem is:

$$\min_x f(x) \quad (2.1)$$

where  $x \in \mathbb{R}^n$  and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

#### Global and local minimum:

Figure 2.1 illustrates the difference between a local minimum and a global minimum for an unconstrained problem.

We say that  $x^* \in \mathbb{R}^n$  is a *local minimum* for the above problem (2.1) if:

$$f(x^*) \leq f(x), \quad \forall x \in \mathfrak{N} \quad (2.2)$$

where  $\mathfrak{N}$  is a neighborhood of  $x^*$ .

It is a *strict local minimum* if:

$$f(x^*) < f(x), \quad \forall x \in \mathfrak{N} \quad (2.3)$$



We say that  $x^* \in \mathbb{R}^n$  is a *global minimum* for the above problem 2.1 if:

$$f(x^*) \leq f(x), \quad \forall x \in \mathbb{R}^n \quad (2.4)$$

It is a *strict global minimum* if:

$$f(x^*) < f(x), \quad \forall x \in \mathbb{R}^n \quad (2.5)$$

Global minima are usually harder to find, and most optimization algorithms solve for local minima instead. This is because to find a local solution, one needs to find a point at which the objective function is smaller than at other feasible points in its neighborhood, while to find a global solution, one needs to compare all the local solutions and find which is the best, i.e. find the one local solution that minimizes the objective function the most. For one particular type of optimization problems, convex problems, the solutions are guaranteed to be global minima, as discussed in section 2.4.

### 2.2.2 Necessary and Sufficient Conditions

There exist efficient ways to find local minima, referred to as necessary and sufficient conditions.

The objective function  $f$  is assumed to be smooth, i.e.  $f$  needs to be twice continuously differentiable.

#### First Order Necessary Conditions:

First order necessary conditions refer to properties of the gradients (first-derivative).

$$\nabla_x f(x^*) = 0 \quad (2.6)$$

The candidates  $x^*$  satisfying the first order necessary condition are referred to as a *critical points*.

#### Second Order Necessary Conditions:

The second order necessary conditions necessitate that the first order necessary conditions are satisfied. In addition to that, the following should hold true:

$$\nabla_x^2 f(x^*) \geq 0 \quad (2.7)$$

### Second Order Sufficient Conditions:

The second order sufficient conditions also require that the first order necessary conditions are satisfied. In addition,  $x^*$  should satisfy:

$$\nabla_x^2 f(x^*) > 0 \quad (2.8)$$

Therefore if the second order necessary and sufficient conditions are satisfied,  $x^*$  is a local minimum of the unconstrained optimization problem.

### 2.2.3 Unconstrained optimization algorithm

There exists different types of algorithms to solve unconstrained optimization problems: line search and trust-regions methods.

The difference between line search and trust-region methods is that line search generates a search direction, while the trust-region defines a region around the current iterate. The line search algorithms depend on the choice of the step size, while the trust-region algorithm depend on the choice of the trust-region size.

Line search algorithms include: backtracking line search, steepest descent, newton methods, golden section, Fibonacci search etc. [29,30]. Trust-region algorithms include: Cauchy point, Dogleg method, Steihaug's approach etc. [31,32]. A complete discussion of rate of convergence for each algorithm is presented in [1,4].

We discuss in more details one line search algorithm in particular: the steepest descent (also referred to as the gradient method), as it will be used in section 4.2.3. All the *line search* algorithms choose a direction  $p_k$  to update the current iterate  $x_k$  to a new iterate with a lower function value. The next iteration is obtained as follows:

$$x_{k+1} = x_k + \alpha_k p_k \quad (2.9)$$

where  $\alpha_k$  is the *step size*.

As seen in the introduction, the steepest method, also known as the gradient method, was introduced in 1847 by Cauchy. The steepest-descent algorithm chooses the search direction to be along the steepest-descent (i.e. the gradient)  $\nabla f_k$ , since it is the one along which  $f$  decreases the most. This algorithm therefore requires the calculation of the gradient  $\nabla f_k$ , and is also commonly referred to as the *gradient descent* algorithm. Note that if the calculation of the gradient is not possible, there exists derivative-free algorithms. The golden section and Fibonacci search are examples of derivative-free line search methods.

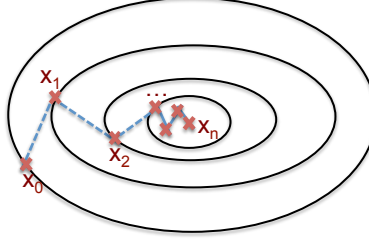


Figure 2.2: Figure representing the contours of a two-dimensional quadratic objective function  $f$  and the sequence of iterations of the gradient descent method. The steps zigzag toward the solution.

See [33] for details.

The search direction is then:

$$p_k = -\nabla f_k \quad (2.10)$$

The choice of the step size  $\alpha_k$  is delicate, as the convergence of the algorithm depends on it. If  $\alpha_k$  is too small, the algorithm will take a long time to converge, and if  $\alpha_k$  is too large, the algorithm might not converge at all. An alternative is to choose the step size at each iteration, but this is computationally expensive. The general condition on  $\alpha_k$  is that it should provide a reduction of the function  $f$ :

$$f(x_k + \alpha_k p_k) < f(x_k) \quad (2.11)$$

Several specific conditions were derived such as the *Wolfe conditions* and the *Goldstein conditions* to make sure  $\alpha_k$  produces sufficient decrease in  $f$ . These conditions will not be described in this thesis, but can be found in [1]. A typical example of convergence of this algorithm is sketched in Fig. 2.2.

## 2.3 Constrained Optimization

Constrained optimization is introduced in order to provide the reader with the necessary background to understand the discussion of inverse optimization methods in sections 3.1 and 4.1.1. Constrained optimization problems have the following properties:

- The objective function depends on unknown variables.
- These variables are subject to a set of constraints.
- The set of constraints is divided into equality constraints and inequality constraints

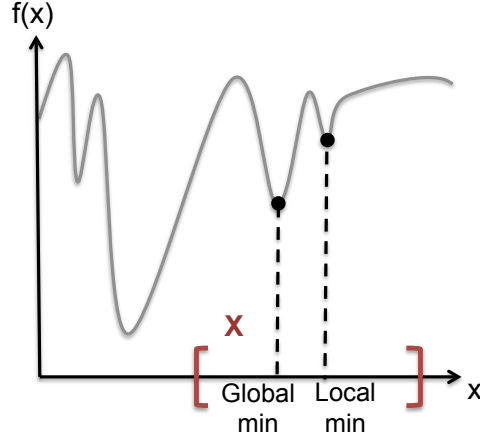


Figure 2.3: Illustration of a local minimum and a global minimum for a constrained problem over the set  $X$ .

The feasible set  $X$  is the set of points  $x$  that satisfy the constraints. If  $f$  is continuous and  $X$  is closed and bounded, then  $f$  has a minimum on  $X$ .

### 2.3.1 Mathematical formulation

The general form of constrained optimization problem is:

$$\begin{aligned} \min_x & f(x) \\ \text{s.t } & h(x) \geq 0 \\ & g(x) = 0 \end{aligned} \tag{2.12}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ .

#### Global and local minimum:

Figure 2.3 illustrates the difference between a local minimum and a global minimum for a constrained problem.

We say that  $x^* \in \mathbb{R}^n$  is a *local minimum* for problem (2.12) if:

$$f(x^*) \leq f(x), \quad \forall x \in \mathfrak{N}, \quad x \in X \tag{2.13}$$

where  $\mathfrak{N}$  is a neighborhood of  $x^* \in X$ .

It is a *strict local minimum* if:

$$f(x^*) < f(x), \quad \forall x \in \mathbb{N}, \quad x \in X \quad (2.14)$$

We say that  $x^* \in \mathbb{R}^n$  is a *global minimum* for the above problem 2.1 if:

$$f(x^*) \leq f(x), \quad \forall x \in X \quad (2.15)$$

It is a *strict global minimum* if:

$$f(x^*) < f(x), \quad \forall x \in X \quad (2.16)$$

### 2.3.2 KKT Necessary and Sufficient Conditions

Necessary and sufficient conditions for constrained optimization problems are usually referred to as Karush Kuhn Tucker (KKT) conditions [8, 34]. The KKT conditions are used as a solving technique for the new method discussed in sections 3.3 and 4.1.2.

#### Lagrange multipliers:

A Lagrange multiplier is associated with each constraints in a constrained optimization problem. The intuitive significance of Lagrange multipliers is that they indicate the sensitivity of the objective function to the associated constraint.

#### Lagrangian function:

The Lagrangian function (also called Lagrange function) associated to the system in (2.12) is defined as follows:

$$L(x, \mu, \lambda) = f(x) + \lambda g(x) + \mu h(x) \quad (2.17)$$

where  $\lambda \in \mathbb{R}^m$  are the Lagrange multipliers associated to the equality constraints  $g(x)$ , and  $\mu \in \mathbb{R}^p$  are the Lagrange multipliers associated to the inequality constraints  $h(x)$ .

#### First Order Necessary Conditions:

Just like for unconstrained optimization, the first order necessary conditions refer to properties of the first-derivative of the Lagrangian function.

$$\nabla_x L(x^*, \mu^*, \lambda^*) = 0 \quad (2.18)$$

where  $x^*$  is a local solution of (2.12) and  $(\lambda^*, \mu^*, \lambda^*)$  are Lagrange multipliers.

### Second Order Necessary Conditions:

The second order conditions refer to properties of the second-derivative of the Lagrangian function:

$$\nabla_{xx}^2 L(x^*, \mu^*, \lambda^*) \geq 0 \quad (2.19)$$

### Second Order Sufficient Conditions:

$$\nabla_{xx}^2 L(x^*, \mu^*, \lambda^*) > 0 \quad (2.20)$$

Therefore if the second order necessary and sufficient conditions are satisfied,  $x^*$  is a local minimum of the constrained optimization problem.

It is also important to note that the necessary and sufficient conditions defined above require that the inequality constraints are linearly independent, such that:

$$\nabla h_1(x^*), \nabla h_2(x^*), \dots, \nabla h_m(x^*) \text{ linearly independent}$$

where  $x^*$  is a local minimum. This is referred to as the *regularity condition*.

### Sensitivity:

As explained previously, a Lagrange multiplier is associated to each constraint. If the necessary and sufficient conditions stated above are satisfied with the solution  $(x^*, \mu^*, \lambda^*)$ , then the corresponding sensitivity is equal to  $-\lambda^*$  for equality constraints and  $-\mu^*$  for inequality constraints. The sensitivity represents the price to pay in the objective function if a constraint is violated [1, 27].

### 2.3.3 Constrained optimization algorithm

There exist several algorithms to solve constrained optimization problem. The one we will describe here is Sequential Quadratic Programming (SQP). Although not directly necessary for the understanding of chapters 3 and 4, we are including it in this thesis because it was used in the process of developing and verifying the inverse optimization algorithm proposed in section 3.3.

SQP is one of the most efficient algorithm for solving nonlinear constrained optimization problems. It can be used to solve any nonlinear constrained problem by solving quadratic subproblems. Since the algorithm requires the use of Newton steps, the convergence of this algorithm is guaranteed provided that the starting point is close enough to the solution  $x^*$

Consider a nonlinear equality constrained problems, such as:

$$\begin{aligned} & \min_x f(x) \\ \text{s.t. } & g(x) = 0 \end{aligned} \tag{2.21}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ .

SQP models the problem at the current iterate  $x_k$  by a quadratic programming subproblem, and then use the solution of this subproblem to define a new iterate  $x_{k+1}$ . The way this subproblem is modeled is by using the KKT optimality conditions we have introduced earlier.

The Lagrangian function associated with this problem is:

$$L(x, \lambda) = f(x) + \lambda g(x) \tag{2.22}$$

Here to simplify the notation, we define  $A(x)$  as the Jacobian matrix of the constraints:

$$A(x) = [\nabla g_1(x) \quad \nabla g_2(x) \quad \dots \quad \nabla g_m(x)]$$

The first-order KKT optimality conditions state that:

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) + A(x)^T \lambda \\ g(x) \end{bmatrix} = 0$$

This system of nonlinear equations can be solved using the Newton's method. At the iterate  $(x_k, \lambda_k)$  the system can be modelled using the quadratic program:

$$\begin{aligned} & \min_p \quad \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ & \text{subject to} \quad A_k p + g_k = 0 \end{aligned}$$

If  $A(x)$  is full rank and the matrix  $\nabla_{xx}^2 \mathcal{L}(x, \lambda)$  is positive definite, then there exists a **unique solution**  $(p_k, l_k)$  that can be found by solving the following system:

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ l_k \end{bmatrix} = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -g_k \end{bmatrix}$$

The next iteration  $(x_k, \lambda_k)$ , or the Newton step, is then given by:

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ l_k \end{bmatrix}$$

This is repeated until convergence of  $(x_k, \lambda_k)$  is reached. It is important to note that the choice of the initial guess  $(x_0, \lambda_0)$  is important for this problem, and that the convergence of SQP is guaranteed if the initial guess is close enough to the solution.

The SQP can be extended to work in the case where the initial problem consists of both equality and inequality constraints. The inequalities are separated in active and inactive constraint sets, and the inactive constraints are ignored. Therefore at each iteration, the quadratic subprogram is solved by taking the active set at the solution as a guess of the optimal active set.

## 2.4 Convex Optimization

Convex optimization is a special class of mathematical optimization problems. We will not be covering the theory of convex optimization in this thesis, and the reader is referred to [28] for a complete review. However, since the problem introduced in section 3.1 simplifies into an unconstrained least-squares problem (see section 3.3.3), which is convex, we will cover the notions of optimality conditions and we will discuss least-squares problems in more details.

### 2.4.1 Necessary and Sufficient conditions

For a convex optimization problem, the first order necessary conditions introduced in sections 2.2 and 2.3 become sufficient conditions. This is due to the fact that for a convex function  $f$  on a set  $X$ , we have:

$$\nabla_{xx}^2 f(x) \geq 0, \quad \forall x \in X$$

Therefore for an unconstrained convex problem, the **first order sufficient condition** is:



$$\nabla_x f(x^*) = 0 \quad (2.23)$$

And for a constrained convex problem, the **first order sufficient condition** is:

$$\nabla_x L(x^*, \mu^*, \lambda^*) = 0 \quad (2.24)$$

where  $L$  is the Lagrangian function. Moreover, the  $x^*$  satisfying these sufficient conditions is not a local minimum as discussed earlier for nonconvex problems. For convex problems, if  $x^*$  satisfies the sufficient conditions, then it is a global minimum of  $f$  on  $X$ .

### 2.4.2 Least-squares problems

We will now discuss in more details a special form of convex optimization problems. This is also in preparation for sections 3.3 and 4.1.2, where the inverse optimization problem simplifies into an unconstrained convex problem that has a least-squares form. Least-squares problems have the following properties:

- Objective function in the form of a sum of squares, i.e. quadratic function
- No constraints

The general mathematical formulation of a least-squares problem is as follows:

$$\min_x f_0(x) = \|Jx - y\|^2 = \sum_{i=1}^k (j_i^T x - y_i)^2 \quad (2.25)$$

The problem in (2.25) can be rewritten as:

$$(J^T J)x = J^T y \quad (2.26)$$

which is simply a set of linear equations. Equation 2.26 is known as the *normal equation*.

Therefore the analytical solution is found by:

$$x = (J^T J)^{-1} J^T y \quad (2.27)$$

Several algorithms can be used to solve the normal equations, the major ones being Cholesky factorization, QR factorization and Singular-Value Decomposition (SVD) [1, 16]. These algorithms are easy to implement, and their computation time is negligible.

## 2.5 Duality

This section presents the theory of duality. Inverse optimization and duality are closely related, as explained later in section 3.4. Given an initial optimization problem, referred to as *primal*, that can be nonlinear and constrained, it is possible to construct its corresponding *dual* problem. The point of doing so is that the dual problem is usually easier to solve than the primal problem [19, 20, 36].

### 2.5.1 Inequality constraints

Consider the following optimization problem, referred to as *primal*:

$$\begin{aligned} \min_x f(x) \\ h(x) \leq 0 \\ x \in X \end{aligned} \tag{2.28}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$  and  $X$  is subset of  $\mathbb{R}^n$  (no convexity restriction necessary on  $f$  and  $h$ ).

The solution to the primal problem is:

$$f^* = \inf_{x \in \Omega} f(x) \tag{2.29}$$

As previously defined, the Lagrangian function of the primal problem is:

$$L(x, \mu) = f(x) + \mu h(x) \tag{2.30}$$

where  $\mu$  is the Lagrange multiplier associated with the inequality constraints  $h$ .

Now define the following function:

$$q(\mu) = \inf_x L(x, \mu) \tag{2.31}$$

Then the corresponding *dual* problem is:

$$\max_{\mu} q(\mu) \tag{2.32}$$

$$\mu \geq 0$$

The solution to the dual problem is then:

$$q^* = \sup_{\mu} q(\mu) \tag{2.33}$$

**Remarks:**

- The Lagrangian function is linear in terms of the Lagrange multipliers  $\mu$ .
- The function  $q$  is the infimum collection of the linear function  $L(x, \mu)$ , therefore  $q$  is concave.
- The constrained set  $\mu \geq 0$  is convex.

$\Rightarrow$  The dual problem is a convex optimization problem.

### 2.5.2 Equality constraints

The same theory can be developed for the case where equality constraints are also present in the primal problem. In fact, having the equality constraint  $g(x) = 0$  is equivalent to having the following two inequality constraints  $g(x) \leq 0, -g(x) \leq 0$ .

Therefore if the primal problem is:

$$\min_x f(x) \tag{2.34}$$

$$h(x) \leq 0$$

$$g(x) = 0$$

$$x \in X$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ .

The Lagrangian function becomes:

$$L(x, \mu, \lambda) = f(x) + \lambda g(x) + (\mu^+ - \mu^-)h(x) \tag{2.35}$$

where  $\lambda$  is associated to the equality constraints  $g$ ,  $\mu^+$  to the positive inequality constraints  $h$  and  $\mu^-$  to the negative inequality constraints  $h$ .

Therefore it is possible to define a new Lagrange multiplier  $\mu = \mu^+ - \mu^-$ , with unrestricted sign. So the Lagrangian function is then:

$$L(x, \mu, \lambda) = f(x) + \lambda g(x) + \mu h(x) \quad (2.36)$$

and the dual problem becomes:

$$\begin{aligned} q(\mu, \lambda) &= \inf_x L(x, \mu, \lambda) \\ \mu &\geq 0 \end{aligned} \quad (2.37)$$

with no sign restriction on  $\lambda$ .

### 2.5.3 Weak Duality

The solution of the dual problem,  $q^*$  is the best lower bound on  $f^*$  that can be obtained from the dual function, according to the definition in (2.33). The weak duality theorem states the following:

$$f^* \geq q^* \quad (2.38)$$

It implies that if the primal function is unbounded, i.e.  $f^* = -\infty$ , then the dual problem is infeasible. Similarly, if the dual problem is unbounded, i.e.  $q^* = +\infty$ , then the primal problem is infeasible.

The difference between  $f^* - q^*$  is known as the *duality gap*.

### 2.5.4 Strong Duality

Under some conditions, the solution of the dual problem is the exact solution of the primal problem, and there is no duality gap.

According to **Slater's theorem**, for a convex optimization problem:

- if there exist a feasible  $x^*$  that is an interior point of  $X$
- if  $f(x^*) < 0$

Then we have:

$$q^* = f^* \tag{2.39}$$

Therefore to guarantee that there will be no duality gap, the primal problem needs to be convex and a feasible candidate of the primal problem needs to be an interior point.

This discussion leads to the following optimality conditions.

### 2.5.5 Optimality conditions

The following globally optimal conditions can be derived from duality theory:

- $g(x^*) \leq 0, x^* \in X \Rightarrow$  Primal Feasibility.
- $\mu^* \geq 0 \Rightarrow$  Dual Feasibility.
- $x^* = \arg \min_x L(x, \lambda^*, \mu^*) \Rightarrow$  Lagrangian Optimality.
- $\mu^* g(x^*) = 0 \Rightarrow$  Complementary Slackness.

The complementary slackness condition implies that the optimal Lagrange multiplier is equal to zero, unless the associated constraint is active at the solution  $x^*$ .

It is important to note that when the strong duality holds, it can be easier to obtain the optimal solution of the primal problem by solving the dual problem, since the dual is a convex optimization problem.

### 2.5.6 Duality and Inverse Optimization

As Bitran pointed out: *"The theory of inverse optimization is intimately involved with the theory of mathematical programming duality"* [19].

The next chapter presents inverse optimization. We will see that the new method we are introducing in section 3.3 uses a technique similar to duality to solve inverse optimization problems.

## Chapter 3

# Inverse Optimization

Considerably less attention has been given to inverse optimization as compared to optimization, in spite of its relevant applications. In this chapter, we will first clearly state the inverse optimization problem. We will then discuss one of the common approaches used to solve these problems, and finally we will introduce a new approach for solving inverse optimization problems.

### 3.1 Problem Formulation

As studied in the previous chapters, the problem of optimization is to find the set of inputs that minimize a given objective function. Given a system defined by the constraints  $g$  and an objective function  $f$  as follows:

$$\begin{aligned} \min_x \quad & f(x) = \alpha(x) + c^T \beta(x) \\ \text{subject to} \quad & g(x) = 0 \end{aligned} \tag{3.1}$$

where  $x \in R^n$  is the variable,  $f$  is the objective function composed of the basis functions  $\alpha$  and  $\beta$  such that  $\alpha(x): R^n \rightarrow R$  and  $\beta(x): R^n \rightarrow R^k$ ,  $g$  is the set of  $m$  equality constraints such that  $g(x): R^n \rightarrow R^m$ , and  $c \in R^k$  is a parameter vector. The functions  $f$  and  $g_i$  are continuously differentiable.

The corresponding optimization problem would be to solve for the state inputs  $x$  that minimize  $f$ , given knowledge of  $g$  and the objective function  $f$ , as introduced previously in section 2.3.

Inverse optimization, as its name suggests, is the opposite of optimization. Inverse optimization consists in finding the objective function that is consistent with the observations of the system, assuming it behaves optimally according to some objective function. The objective function is defined as a set of known basis functions  $\alpha$  and  $\beta$ , weighted by the unknown parameter  $c \in R^k$ . Considering the system introduced in (3.1), the inverse optimization problem consists in determining the parameter  $c$ , given observations of  $x$  and knowledge of the basis functions  $\alpha$  and  $\beta$ .

We will now study different ways of solving inverse optimization problems.

## 3.2 Bilevel Approach

The bilevel approach for solving inverse optimization problems has been introduced in [37, 38].

### 3.2.1 Assumptions

The assumptions necessary for this approach are the following: the system that is being observed is assumed to be perfect, in the sense that the system is behaving optimally, while the observations of the system might be imperfect, in the sense that noise might compromise the observations.

### 3.2.2 Formulation

Consider the system defined in (3.1), where  $f$  is the objective function composed of two basis functions  $\alpha$  and  $\beta$ ,  $g$  is the set of constraints, and  $c$  is the unknown parameter.

Given observations  $y$  of the system, that are noisy:

$$y \sim \mathcal{N}(x, \Sigma)$$

The inverse problem is to find the value of the unknown parameter  $c$  that minimizes the following:

$$\|y - x^*(c)\|^2$$

The objective function can be expressed as a sum of a series of basis functions weighted by unknown parameters. The higher the value of the parameter is, the more likely the associated basis function needs to be minimized. Therefore, the most important basis functions of the objective function correspond to the ones with the highest parameters. It is also important to note that only the relative size of the parameters is important, and this is why one of the parameter needs to be normalized to 1.

### 3.2.3 Solving technique

The bilevel approach solves an inverse optimization problem in two steps, or *levels*, as referred to in [38].

The first level consists in estimating the set of unknown parameters so that:

$$\min_c \sum_{i=1}^m \|y - x^*(x_i, u_i, c)\|^2$$

where  $m$  is the set of discrete points,  $y$  is the observation,  $x$  and  $u$  and the state and control inputs respectively, and  $x^*$  is the optimal path recovered with the current estimation of  $x$ . This is a least-squares problem, as introduced in section 2.4.2. The authors in [38] solve this problem using a derivative-free

optimization code *BOBYQA*, that stands for bound optimization by quadratic approximation, developed by Powell [39].

The second level of the bilevel method consists in solving the forward optimization problem using the parameter estimated from the first level:

$$\min_{x_i, u_i} f(x) = \sum_{j=1}^n \sum_{i=1}^m c_j \phi_j(x_i, u_i) \quad (3.2)$$

$$\text{subject to } g(x_i, u_i) = 0$$

where  $\phi$  consists of the set of  $n$  basis functions.

This is a nonlinear constrained optimization problem, as introduced in section 2.3. The authors in [38] solve this problem using a multiple shooting code *MUSCOD* developed by Bock [40].

The algorithm then goes back to the first level to find a new estimate of the parameters  $c$  given the recovered trajectory  $x^*(x_i, u_i, c)$  from the second level. It is therefore an iterative process, that requires estimating the parameters and solving an optimization problem at each iteration.

### 3.3 New Approach

The approach that we will now describe is inspired from the work done by [42], and has been published in [41].

#### 3.3.1 Assumptions

The assumptions necessary for this approach are the following: the observations of the system are assumed to be perfect, meaning that there is no noise, while the system itself might be imperfect, in the sense that it may not behave in an optimal manner.

#### 3.3.2 Formulation

For convenience, the system we are considering is displayed again:

$$\min_x f(x) = \alpha(x) + c^T \beta(x) \quad (3.3)$$

$$\text{subject to } g(x) = 0$$



where  $x \in R^n$  is the variable,  $f$  is the objective function composed of the basis functions  $\alpha$  and  $\beta$  such that  $\alpha(x): R^n \rightarrow R$  and  $\beta(x): R^n \rightarrow R^k$ ,  $g$  is the set of  $m$  equality constraints such that  $g(x): R^n \rightarrow R^m$ , and  $c \in R^k$  is an unknown parameter vector. The functions  $f$  and  $g_i$  are continuously differentiable.

The observations  $y$  are now assumed to be perfect, while the system may not behave optimally, meaning:

$$x \sim \mathcal{N}(x^*, \Sigma)$$

The inverse optimization problem associated with 3.3 is to recover the unknown parameter vector  $c$  given perfect observations of  $x$ , and having prior knowledge of  $\alpha$ ,  $\beta$  and  $g$ .

### 3.3.3 Necessary conditions for optimality

In section 2.3.2, we introduced the KKT necessary condition for optimality to solve constrained optimization problems. We will now see that we can use them here to solve the inverse optimization problem.

For a given  $c$ , assuming that  $x^*$  is a local minimum of the problem in (3.3) and is regular, there exist unique Lagrange multiplier vectors  $\lambda^* \in R^m$  such that [4], [27]:

$$\begin{aligned} \nabla_x f(x^*, c) + \sum_{i=1}^m \lambda_i^{*T} \nabla_x g_i(x^*) &= 0 \\ g_i(x^*) &= 0, \quad i = 1, \dots, m \end{aligned} \tag{3.4}$$

where  $f$  and  $g_i$  are continuously differentiable functions. We can recognize the two equations in (3.4) are the KKT necessary conditions for equality constrained optimization problems as seen in section 2.3.2. The first equation in (3.4) is the stationarity condition while the second equation ensures primal feasibility.

If the Lagrangian of problem in (3.3) is defined to be:

$$L(x, c, \lambda) = f(x, c) + \sum_{i=1}^m \lambda_i^T g_i(x) \tag{3.5}$$

then for a given  $c$ , the necessary conditions in (3.4) can be rewritten as:

$$\nabla_{(x, \lambda)} L(x^*, c, \lambda^*) = 0 \tag{3.6}$$

### 3.3.4 Residual Functions

As stated previously, the system is assumed to be only approximately optimal. Residual functions are defined in order to represent what *approximately optimal* means in a manner similar to [42]:

$$\begin{aligned}
r_{eq} &= g(x) \\
r(x, c, \lambda) &= \nabla_x f(x, c) + \sum_{i=1}^m \lambda_i \nabla_x g_i(x)
\end{aligned} \tag{3.7}$$

The necessary conditions of optimality are satisfied when the two residual functions in (3.7) are equal to zero. The method consists then in minimizing the extent to which observed decisions violate the KKT necessary conditions, i.e. minimizing the extent to which they are not equal to zero. The candidate solutions are obtained from the stationarity residual function, while the first residual function is used to check that the solutions obtained are feasible.

Using the Lagrangian defined as above, the stationarity residual function in (3.7) becomes:

$$r(x, c, \lambda) = \nabla_x L(x, c, \lambda) \tag{3.8}$$

which for problem (3.3) corresponds to:

$$r(x, c, \lambda) = \nabla_x \alpha(x) + c^T \nabla_x \beta(x) + \lambda^T \nabla_x g(x) \tag{3.9}$$

Given observations of  $x$  that are assumed to be perfect, the inverse optimization problem becomes to minimize the residual function defined in (3.9) where the unknowns are the Lagrangian multipliers  $\lambda$  and the parameter  $c$ :

$$\min_{c, \lambda} ||r(x, c, \lambda)||^2 \tag{3.10}$$

One can see that the initial constrained optimization problem in (3.3) has been modified into an unconstrained optimization problem in (3.20), with the only limitation that the objective function needs to be composed of a linear combination of known basis functions. The reader is referred to section 2.2 for the unconstrained optimization theory background.

Note also that the residual function shown in (3.9) is linear with respect to the unknown parameter  $c$  and the Lagrange multipliers  $\lambda$ . The problem therefore becomes a convex unconstrained least-squares optimization problem, which is easy to solve, as discussed in sections 2.4 and 2.4.2.

It is important to note that this method can be extended to several variations of the problem presented in (3.10). For example, one could consider the case where multiple observations of the same system are given, or one could also consider the case where the unknown parameter  $c$  changes with time. We believe our approach can be used to solve discrete-nonlinear problems that consist of a continuously differentiable cost

function and continuously differentiable constraints, but with the limitation that the unknown parameter vector needs to enter the cost function linearly.

### 3.3.5 Solving techniques

The problem in (3.10) can be recognized as a least-squares problem. As discussed in section 2.4.2, least-squares problems can be solved easily using Cholesky factorization, QR factorization or Singular-Value Decomposition (SVD) [1, 16].

The objective function in (3.10) can be rewritten in the following manner:

$$r(x, z) = \frac{1}{2} \|Jz - b\|^2 \quad (3.11)$$

where  $z$  is the vector of unknown parameters:  $z = [c \ \lambda]^T$ . Since the problem is convex, from section 2.4 we know that a global minimizer  $z^*$  of  $r(x, z)$  must satisfy  $\nabla_x r(x, z^*) = 0$ , which leads to the *normal equations*:

$$J^T J z^* = J^T b \quad (3.12)$$

Therefore, using this new approach, solving the inverse optimization problem associated with the system presented in (3.3) simplifies into solving for the unknown parameters  $z$  in the convex unconstrained least-squares problem in (3.11).

For the scope of problems we are interested in solving in this thesis, this approach is a considerable improvement as compared to the bilevel approach. Using the KKT necessary conditions of optimality to define residual functions leads to a simple unconstrained optimization problem, whereas the bilevel approach requires solving a constrained optimization problem iteratively.

We are not sure to what extent this method can be applied to, but we believe that it can be used to solve a wide range of nonlinear constrained problems, with the limitation that the unknown parameter vector needs to enter the objective function linearly.

## 3.4 Duality and Inverse Optimization – More details

As briefly stated in section 2.5, duality and inverse optimization are closely related. Now that inverse optimization has been defined and we have introduced our new solving approach, we will discuss this fact in more details.

### 3.4.1 Theory:

The notation in this part will be slightly modified, in order to match better with [19]. Remember that the primal problem  $P(d)$  is such that:

$$p^*(d) = \min_x f(x) \quad (3.13)$$

$$g(x) = d$$

$$x \in X$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^m \rightarrow \mathbb{R}$  and  $X$  is subset of  $\mathbb{R}^n$ . At this point, no convexity restrictions are necessary on  $f$  and  $g$ , the primal problem can be any nonlinear constrained optimization problem.

We know that if we define the Lagrangian function:

$$L(x, \lambda, d) = f(x) + \lambda(g(x) - d) \quad (3.14)$$

the corresponding dual problem  $Q(d)$  is to find the best lower bound to the primal problem, namely:

$$q^*(d) = \max_{\lambda} q(\lambda, d) \quad (3.15)$$

$$\lambda \in \mathbb{R}^m \quad (3.16)$$

where

$$q(\lambda, d) = \inf_x L(x, \lambda, d) \quad (3.17)$$

Note that  $P(d)$  and  $Q(d)$  refer to the primal and dual problem respectively, while  $p^*(d)$  and  $q^*(d)$  refer to the value of the objective function when  $x$  and  $\lambda$  are optimal.

The duality theorem states that if  $x^* \in X$  and  $\lambda^*$  satisfy the global optimality conditions (stated in the previous chapter) for a given vector  $d$ , then  $x^*$  is optimal in the primal problem  $P(d)$  and  $\lambda^*$  is optimal in the dual problem  $Q(d)$ . Moreover,  $p^*(d) = q^*(d) = L(x^*, \lambda^*; d)$  [19].

Therefore, one way to solve the primal problem  $P(d)$  is to select a dual vector  $\lambda$  such that the solution  $x(\lambda)$  computed from (3.19) satisfies the global optimality conditions for a given  $d$ .

This theorem implies that there is a mapping between the computation of the Lagrangian function

and the optimal solution of the primal problem  $P(d)$ . When computing the Lagrangian function in (3.19) to obtain  $x(\lambda)$ , it corresponds to an optimal solution of  $P(d)$ , and this is the underlying idea of inverse optimization. This is summarized by the following theorem :

For any  $\lambda \in \mathbb{R}^m$ , let  $x(\lambda)$  denote an optimal solution to the problem (3.19). The solution  $x(\lambda)$  is optimal in the primal problem  $P(d(\lambda))$ , where  $d(\lambda) = g(x(\lambda))$ . Moreover,  $p^*(d(\lambda)) = f(x(\lambda))$  [20].

Therefore the equation in (3.19) induces the following mapping:

$$D = \{d | p^*(d) \text{ is finite}\}$$

For a  $\lambda$ , the image of this mapping is the set:

$$D^*(\lambda) = \{d | d_i = g_i(x(\lambda)) \quad \forall i, x(\lambda) \text{ optimal in 3.19}\}$$

and clearly  $D^*(\lambda) \subseteq D$ , so we say that  $q(\lambda)$  spans  $P(\lambda)$  if  $d \in D^*(\lambda)$ , for some  $\lambda$ .

### 3.4.2 Our approach:

We will now take a look back at our approach to understand how this theory applies to us, and the reader is referred to section 3.3.3. Remember that our approach is based on the calculation of residual functions using the necessary conditions of optimality:

$$r(x, c, \lambda) = \nabla_x L(x, c, \lambda) \tag{3.18}$$

which corresponds to:

$$q(x, c, \lambda) = \inf_x L(x, c, \lambda) \tag{3.19}$$

since calculating the gradient of the Lagrangian function consists in determining the value of  $x(c, \lambda)$  that minimizes  $L$ . The equation in (3.18) induces a mapping, and the solution  $x(c, \lambda)$  is optimal in the primal problem in (3.3).

The next step in our approach is to minimize the residual functions:

$$\min_{c, \lambda} ||r(x, c, \lambda)||^2 \tag{3.20}$$

where  $r$  is convex, which corresponds to:

$$q^*(x, c, \lambda) = \max_{\lambda} q(x, c, \lambda) \quad (3.21)$$

$$\lambda \in \mathbb{R}^m \quad (3.22)$$

where  $q$  is concave.

We can now clearly see the relationship between our solving approach and the duality theory. Using residual functions in our approach is equivalent to defining a dual problem. Solving the residual functions for  $c$  and  $\lambda$  leads to finding an optimal solution  $x(c, \lambda)$  for the primal problem. Therefore one way to solve inverse optimization problems is to use the duality theory. However, as seen in section 2.5.4, it is important to satisfy the strong duality conditions, in order to ensure that the solution to the dual problem is indeed the solution to the primal problem.

# Chapter 4

## Applications

This chapter presents the application of our new inverse optimization solving approach to real data. As discussed in the introduction, inverse optimization can be used to study biological systems [37, 38]. In the scope of this thesis, inverse optimization is used to find a model of human locomotion. The first section of this chapter focuses on applying our method to data generated in simulation, while the second section focuses on applying our method to real data obtained experimentally.

### 4.1 Application to Simulated Data

In this section we apply the approach introduced in the section 3.3 to data generated in simulation in order to validate it. The model used to study human locomotion is chosen to be the unicycle model, since it is a simple nonholonomic model that allows for forward and rotational speeds while preventing translation.

#### 4.1.1 The Unicycle Model

Consider the following discretized version of the unicycle model:

$$\begin{aligned} \min_{x^i, u^i} \quad & \frac{1}{2}\tau \sum_{i=0}^{N-1} (c(u_1^i)^2 + (u_2^i)^2) \\ \text{subject to} \quad & x_1^{(i+1)} = x_1^i + \tau u_1^i \cos(x_3^i) \\ & x_2^{(i+1)} = x_2^i + \tau u_1^i \sin(x_3^i) \\ & x_3^{(i+1)} = x_3^i + \tau u_2^i \\ & x^{(0)} = x_{start} \\ & x^{(N-1)} = x_{goal} \end{aligned} \tag{4.1}$$

where  $\tau$  is the discretization rate, and  $i$  is the time step, going from 0 to  $N - 1$ . The position and the orientation are  $(x_1, x_2)$  and  $x_3$ , respectively. The two inputs are the forward speed  $u_1$  and the turning rate  $u_2$ . The start and end points are set fixed to match the experimental set-up described in section 4.2.

The unknown parameter  $c$  governs how much we penalize control effort  $u_1$  relative to control effort  $u_2$ . The problem in (4.1) is a constrained problem, as seen previously in section 2.3. The inverse optimization problem consists in recovering the value of the parameter  $c$ , using data generated in simulation. Note that the fixed end-point constraints do not appear in the derivation of residual function, and the implications are discussed in section 4.2.3.

#### 4.1.2 Derivation of the Residual Function for the Unicycle Model

Based on (3.5), the Lagrangian associated with the unicycle model in (4.1) is:

$$\begin{aligned} L(x^i, u^i, c, \lambda^{i+1}) &= \frac{1}{2}\tau \sum_{i=0}^{N-1} (c(u_1^i)^2 + (u_2^i)^2) \\ &\quad + \sum_{i=0}^{N-1} (g^i(x^i, u^i) - x^{i+1})^T \lambda^{i+1} \end{aligned} \quad (4.2)$$

where  $g$  is the set of equality constraints in (4.1).

The KKT conditions defined in (3.4) and (3.6) for the unicycle model lead to:

$$\begin{aligned} \nabla_{(x^i)} L(x^i, u^i, \lambda^i, \lambda^{i+1}) &= -\lambda^i + \nabla_{(x^i)} g^i(x^i, u^i)^T \lambda^{i+1} \\ &= 0 \end{aligned} \quad (4.3)$$

and

$$\begin{aligned} \nabla_{(u^i)} L(x^i, u^i, c, \lambda^{i+1}) &= \frac{1}{2}\tau \nabla_{(u^i)} (c(u_1^i)^2 + (u_2^i)^2) \\ &\quad + \nabla_{(u^i)} g^i(x^i, u^i) \lambda^{i+1} \\ &= 0 \end{aligned} \quad (4.4)$$

The residual function defined in (3.18) then becomes:

$$\begin{aligned} r(x^i, u^i, c, \lambda^i, \lambda^{i+1}) &= \begin{bmatrix} \nabla_{(x^i)} L(x, u, \lambda^{i+1}) \\ \nabla_{(u^i)} L(x, u, \lambda^{i+1}) \end{bmatrix} \\ &= \begin{bmatrix} -\lambda^i + \nabla_{(x^i)} g^i(x^i, u^i)^T \lambda^{i+1} \\ \frac{1}{2}\tau \nabla_{(u^i)} (c(u_1^i)^2 + (u_2^i)^2) + \nabla_{(u^i)} g^i(x^i, u^i) \lambda^{i+1} \end{bmatrix} \end{aligned} \quad (4.5)$$

and is linear as a function of the unknown parameters  $[c \ \lambda^i \ \lambda^{i+1}]^T$ . It can therefore be rewritten in a similar manner to (3.11) and solved using the algorithms for linear least-squares unconstrained problems discussed in sections 2.2 and 2.4.2. Note that the residual vector has dimensions  $(5N, 1)$ , where  $N$  is the total number of time steps, since  $x \in R^3$  and  $u \in R^2$ .



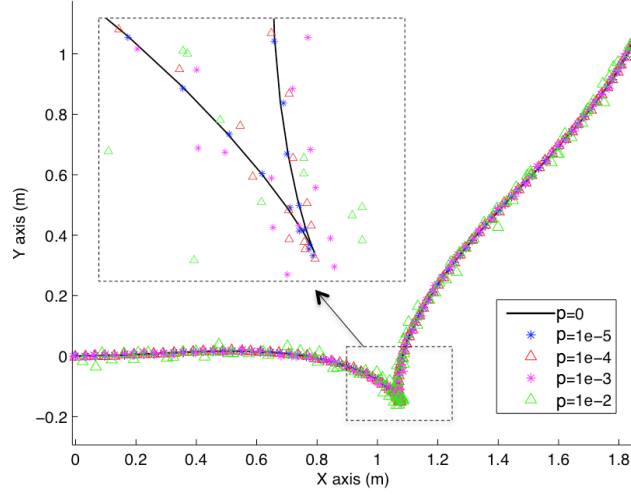


Figure 4.1: Example of a noiseless trajectory simulated using the unicycle model (black line), along with four different noisy measurements. Each set of measurements has been obtained by averaging 60 noisy trajectory measurements. The amount of noise varies and is characterized by the value of the standard deviation  $p = 1e^{-5}$ ,  $1e^{-4}$ ,  $1e^{-3}$  and  $1e^{-2}$ .

#### 4.1.3 Results for Data Generated in Simulation

Matlab 7.11.0 is used to implement the method introduced in sections 3.3 and 4.1.2. Paths are simulated using the unicycle model described in (4.1) by arbitrarily choosing a value for the parameter  $c$  ( $c = 3.7$  was chosen). Once the trajectories are generated, we can use the observations of the states  $x$  and control inputs  $u$  to solve the unconstrained least-squares optimization problem defined in (3.20) where the residual function is given by (4.5). Recall that the aim of solving the inverse optimization problem is to recover the value of the unknown parameter  $c$ . Since the trajectories are obtained from simulation, we can compare the value of  $c$  predicted using the approach introduced, to the actual value of the parameter used during the simulation.

To test the robustness of the approach, noise is added to the measurements. Gaussian white noise normally distributed with zero mean and variable standard deviation is added to the states  $x$  such that:  $x \sim \mathcal{N}(0, p)$ . The noise variable  $p$  is chosen to take the following values:  $1e^{-5}$ ,  $1e^{-4}$ ,  $1e^{-3}$  and  $1e^{-2}$ . It is important to note that since the unknown parameter is assumed to be constant (non-time-varying) in our model, we do not need to consider multiple trajectories to predict the correct value of  $c$ . Instead, we are looking at how fast the predicted value of  $c$  converges as a function of the number of data points constituting the trajectory, i.e. the time horizon of the trajectory. A total of 60 trajectory measurements are generated for each values of  $p$ , which are then averaged. In other words, for each value of noise standard deviation  $p$ , we have one corresponding averaged set of measurements. Fig. 4.1 shows a simulated noiseless trajectory, along with the four different averaged noisy measurements obtained from the four values of  $p$ .

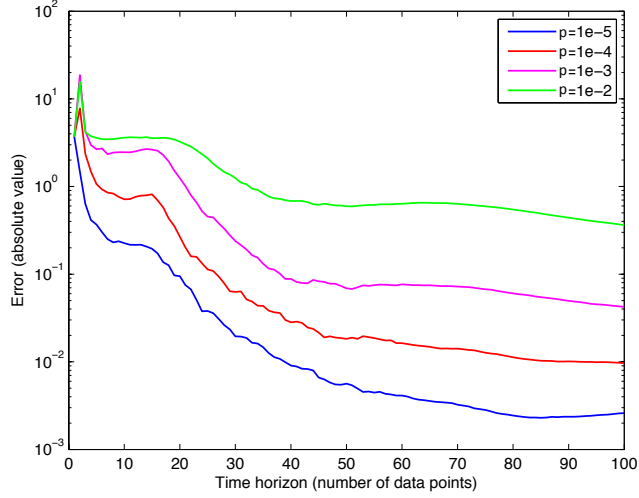


Figure 4.2: Error between the predicted value of the parameter  $c$  obtained for each of the four averaged noisy measurements, and the value used during simulation. The approach used is able to predict the correct value of  $c$  ( $c = 3.7$ ), even when very noisy measurements are used (i.e. for  $p = 1e^{-2}$ ).

Our inverse optimization algorithm is used to recover the value of  $c$  for each set of measurements. The computation time required to run the algorithm is 0.032 seconds (on a 2.5 GHz Core 2 Duo processor). The results showing the comparison between the predicted value of  $c$  and the value used during simulation are shown in Fig. 4.2.

The results in Fig. 4.2 show that even with large Gaussian white noise (i.e.  $p = 1e^{-2}$ ) the algorithm is able to predict the correct value of  $c$ . It is obvious from Fig. 4.2 that when the amount of white noise is increased, the algorithm necessitates more points from the trajectory (i.e. a longer time horizon) to correctly recover the unknown parameter. Therefore results in Fig. 4.2 show that our approach also performs well in presence of noisy measurements.

## 4.2 Application to Experimental Data

In this section we apply the approach introduced in sections 3.3 and 4.1.2 to experimental data, in order to find a model for human locomotion. The reason why we would want to model human locomotion, in the context of humanoid robot control, is to obtain a model based on observations that can be implemented to humanoid robots to generate locomotion trajectories similar to human trajectories, and to predict actions or trajectories of robots based on human observations.

### 4.2.1 The Experiment

The data used were collected for the experiment described in [37]. In summary, subjects were asked to walk in a gymnasium from a starting point to a final destination represented by a porch. The starting point was always the same, but the final position and final orientation of the porch were varying. The subjects were asked to walk from one point to another freely, without time or velocity constraints, and the trajectories were recorded using motion capture technology. An example of a subset of 6 observed trajectories is presented in Fig. 4.4 (blue lines) for one subject.

### 4.2.2 Notes on the Choice of the Model

The initial assumption in [37] was that human walking data can be modeled using a simple nonholonomic unicycle model along with an objective function that minimizes input energy. However, it was found that more complicated dynamic models could be used to better fit the experimental data. For example, [37] defines the turning input to be the derivative of the curvature. In [38], the authors not only use a more complex dynamic model, but they also define a more complex objective function that takes into account the initial tendency of the subjects to adjust the orientation of their bodies towards the target.

However, in this application example, we want to focus on validating our inverse optimization approach by showing results for the standard unicycle model and with a simple objective function defined in (4.1). As we will show in the rest of this chapter, our approach leads to promising results.

### 4.2.3 Recovered Trajectories

Once the inverse optimization problem is solved and the unknown parameter  $c$  is predicted, it is necessary to generate the trajectory obtained using the model and the recovered value of  $c$ , and compare it to the experimental data. Instead of using a shooting method to generate the recovered trajectories, a faster algorithm is employed, based on the same KKT conditions for optimality. From (4.3) and (4.4), the following relationships can be derived:

$$\begin{aligned}(\lambda^i)^T &= (\lambda^{i+1})^T J^i \\ (u^i)^T &= -(\lambda^{i+1})^T G^i\end{aligned}\tag{4.6}$$

where for the unicycle model:

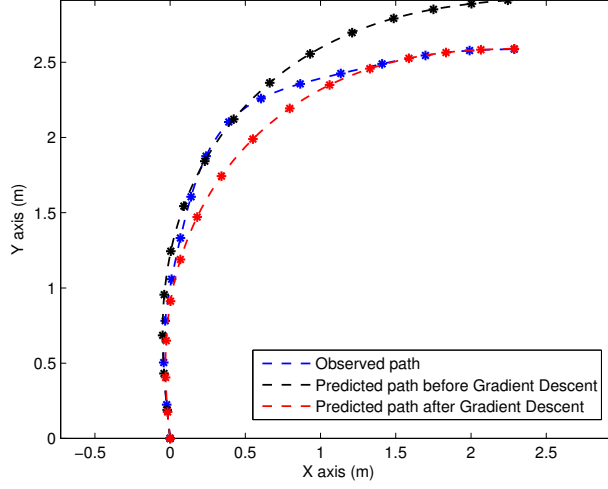


Figure 4.3: Recovered trajectories before and after performing gradient descent to match the end points. The blue trajectory is the observed one obtained from experimental data. The black trajectory is the predicted result obtained before performing gradient descent and the red trajectory is the predicted result after performing gradient descent.

$$J^i = \begin{bmatrix} 1 & 0 & -\tau u_1^i \sin x_3^i \\ 0 & 1 & \tau u_1^i \cos x_3^i \\ 0 & 0 & 1 \end{bmatrix}, \quad G^i = \begin{bmatrix} \frac{1}{c} \cos x_3^i & 0 \\ \frac{1}{c} \sin x_3^i & 0 \\ 0 & 1 \end{bmatrix} \quad (4.7)$$

which leads to:

$$\begin{aligned} \lambda^{i+1} &= (J^i)^{-T} p^i \\ (u^i)^T &= -(\lambda^i)^T (J^i)^{-1} G^i \end{aligned} \quad (4.8)$$

where we note that  $(J^i)^{-1}$  always exists. Therefore using the state equations from (4.1) and the set of equations from (4.8), it is possible to reconstruct the entire trajectory. The recovered trajectory depends only on the initial position  $x^{(0)}$ , which is given, and the initial set of Lagrange multipliers  $\lambda^{(0)}$ , which is predicted when solving the inverse optimization problem. The initial recovered trajectory obtained from  $\lambda^{(0)}$  and  $x^{(0)}$  does not necessarily satisfy the terminal constraint. This is because the end-point constraints are not taken into account in our residual function. Therefore, we use gradient descent, as introduced in section 2.2.3, to search for a  $\lambda^{(0)}$  that results in a trajectory satisfying the terminal constraint.

#### 4.2.4 Results for the Experimental Data

The unconstrained linear least-squares problem being solved is defined in (3.20) where the residual function is given by (4.5) for the unicycle model presented in (4.1). Once the least-squares problem is solved and the unknown parameter  $c$  is predicted along with the Lagrange multipliers  $\lambda$ , the set of equations in (4.8) and (4.1) are used to construct the predicted trajectory.

Figure 4.3 shows the recovered trajectory before and after gradient descent. The total computation time to solve the inverse optimization problem, construct the predicted trajectory and perform gradient descent is 0.076 seconds. One can see from Fig. 4.3 that the recovered trajectory does not perfectly fit the experimental data, and this is believed to be due to the limitations of the unicycle model.

Beyond considering only one trajectory, the experimental data consists of a set of multiple trajectories for a variety of boundary conditions. In particular, we consider a subset of the data which consists of 15 trajectories for 6 different boundary conditions (i.e. the same terminal position and orientation). Figure 4.4 shows results of recovered trajectories when using only one observed trajectory (dashed red lines) and when using multiple observed trajectories (dashed black lines). One can see that better results are obtained when using multiple observations to recover the parameter  $c$ . Note that in the case of multiple observations, the linear least-squares problem becomes:

$$\min_{c, \lambda^1, \dots, \lambda^{N_{traj}}} \sum_{i=1}^{N_{traj}} \|r(x^i, u^i, c, \lambda^i, \lambda^{i+1})\|^2 \quad (4.9)$$

where  $x^1, x^2, \dots, x^{N_{traj}}$  are  $N_{traj}$  perfectly observed trajectories and  $c$  is constant across the  $N_{traj}$  trajectories. Our results suggest that more observations used to predict  $c$  will result in a better recovered cost function, i.e. one which better predicts observed trajectories.

The total computation time to solve the inverse optimization problem using 15 observed trajectories, construct the predicted trajectory and perform gradient descent was 16 seconds. This increase in computation time is due to the fact that the size of the residual vector is no longer  $(5N, 1)$ , but  $(5N_{traj}N)$ , where  $N_{traj}$  is the number of observed trajectories. Despite the increased computation time, we believe the complexity of solving least-squares problem is less than the complexity of iterative techniques which require solving the forward optimal control problem at each iteration. A full comparison of running time and complexity is the subject of future work.

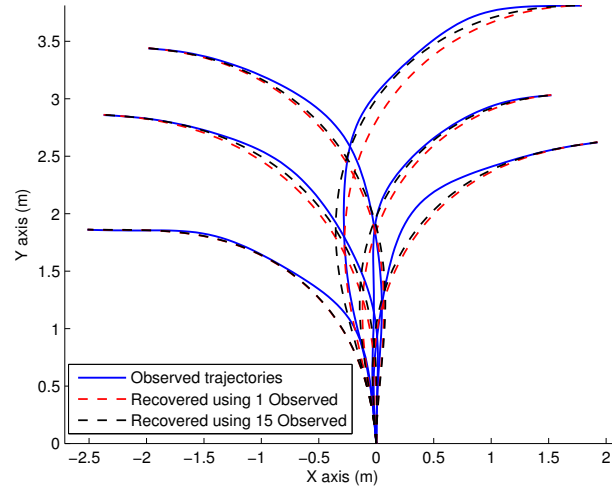


Figure 4.4: In this figure, observed and predicted trajectories are projected on the x-y plane. Blue curves represent observed trajectories obtained from experimental data. Red dashed curves show predicted results obtained when using only a single observation to recover the value of parameter  $c$ . Black dashed curves show predicted results obtained when using 15 observations to recover the value of  $c$ .

## Chapter 5

# Conclusions and Future Work

### 5.1 Conclusions

This thesis offered a study of inverse optimization and of the necessary optimization background, along with a new algorithm for solving inverse optimization problems. The first half of this thesis covered the background of optimization theory that is necessary for the understanding of the inverse optimization problem studied in the second half. In the second half of this thesis we defined clearly what an inverse optimization problem is, we gave an example of an existing solving technique, and then proposed a new solving approach. This approach was then used to find a model for human locomotion, based on experimental data obtained of human subjects walking from a fixed starting point to a target location.

The standard formulation to solve an inverse optimization problem assumes that decisions are optimal and consists in minimizing the difference between what is predicted and what is observed given a candidate objective function. The new formulation we propose assumes instead that decisions are only approximately optimal and we minimize the extent to which observed decisions violate necessary conditions of optimality. The method proposed is closely related to duality theory, and the initial nonlinear constrained optimization problem simplifies into an unconstrained linear least-squares problem, which can be solved easily and which does not necessitate iterations. This approach can be applied to discrete-time nonlinear constrained systems with the limitation that the objective function has to be a linear combination of known basis functions.

A discretized unicycle model is used to test the approach with both simulation and experimental data. The application to data generated in simulation successfully validates the approach, and shows that it also performs well in presence of noisy measurements. The approach is then applied to experimental data in order to find a model for human locomotion. Subjects were asked to walk in a gymnasium from a starting point to a final destination and their trajectories were recorded [37]. The approach gives satisfactory results when comparing the recovered trajectories to the experimental data. Furthermore, using multiple observed trajectories to recover the unknown parameter of the objective function is found to improve the quality of the recovered trajectories.

Therefore, it was shown that this new approach can be used to obtain a simple model of human locomotion, without solving the initial optimization problem iteratively, but by simply minimizing the extent to which observed trajectories violate the necessary conditions of optimality.

## 5.2 Future Work

This section suggests improvements that can be done to what has been presented in this thesis and recommendations for future work.

The details of the limitations of the new approach for solving inverse optimization problems discussed in this thesis still need to be clarified. For example, it is not clear what the conditions are for perfectly recovering the objective function, and how sensitive the approach is to model uncertainty. One suggested way to understand the limitation of this approach is to understand the conditions required so that strong duality holds. To do so, it will be important to clearly understand the relationship between inverse optimization and duality theory.

Also, more effort should be placed towards using better models for human locomotion. Studies such as [37] and [38] used more complex models, and our approach can be used to solve a more complex dynamic model along with a more complete objective function.

Another necessary improvement to this thesis is to compare our approach to existing methods, in terms of computation time and complexity. There exist more methods to solve inverse optimization problems than what has been presented in this thesis, and an in-depth comparative study is necessary to understand where the new approach presented stands among the existing methods.



# References

- [1] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer Verlag, 2006.
- [2] F. Giannoni, P. Piccione, and J. Verderesi, “An approach to the relativistic brachistochrone problem by sub-riemannian geometry,” *Journal of Mathematical Physics*, vol. 38, no. 6367, 1997.
- [3] B. Vratnagar and M. Saje, “On the analytical solution of the brachistochrone problem in a non-conservative field,” *Journal of Non-Linear Mechanics*, vol. 33, no. 3, pp. 489–505, 1998.
- [4] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*, 3rd ed., ser. International series in operations research & management science. New York: Springer, 2007, vol. 116.
- [5] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1965.
- [6] S. M. Stigler, *The History of Statistics, The Measurement of Uncertainty before 1900*, T. B. P. of Harvard University Press., Ed. Cambridge, 1986.
- [7] Y. Yuan, “Step-sizes for the gradient method,” *The Academy of Mathematics and Systems Science, Chinese Academy of Sciences*, 2006.
- [8] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” *Proceedings of 2nd Berkeley Symposium*, pp. 481–492, 1951.
- [9] V. G. Boltyanskii, R. V. Gamkrelidze, and L. S. Pontryagin, “Towards a theory of optimal processes,” *Reports Acad. Sciences USSR*, vol. 110(1), 1956.
- [10] R. E. Bellman, *Dynamic Programming*. Princeton University Press, Princeton NJ, 1957, Republished 2003.
- [11] R. Rouge and N. E. Karoui, “Pricing via utility maximization and entropy,” *Mathematical Finance*, vol. 10, no. 2, pp. 259–276, April 2000.
- [12] T. M. Whitin, “Inventory control and price theory,” *Management Science*, vol. 2, no. 1, pp. 61–68, Oct. 1955.
- [13] I. Elishakoff, R. T. Haftka, and J. Fang, “Structural design under bounded uncertainty - optimization with anti-optimization,” *Computers and Structures*, vol. 53, no. 6, pp. 1401–1405, 1994.
- [14] R. Kalman, “When is a linear control system optimal?” *ASME Transactions, Journal of Basic Engineering*, vol. 86, pp. 51–60, 1964.
- [15] M. Masak, “An inverse problem on decoupling optimal control systems,” vol. 13, no. 1, pp. 109 – 110, feb 1968.
- [16] A. Jameson and E. Kreindler, “Inverse problem of linear optimal control,” *SIAM Journal on Control*, vol. 11, no. 1, pp. 1–19, 1973. [Online]. Available: <http://link.aip.org/link/?SJC/11/1/1>

- [17] J. Casti, "On the general inverse problem of optimal control theory," *Journal of Optimization Theory and Applications*, vol. 32, no. 4, pp. 491–497, 12 1980. [Online]. Available: <http://dx.doi.org/10.1007/BF00934036>
- [18] R. W. Obermayer and F. A. Muckler, "On the inverse optimal control problem in manual control systems," *National Aeronautics and Space Administration, Contractor Report*, 1965.
- [19] G. R. Bitran, V. Chandry, D. E. Sempolinski, and J. F. Shapiro, "Inverse optimization: An application to the capacitated plant location problem," *Management Science*, vol. 27, no. 10, pp. 1120–1141, Oct. 1981.
- [20] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Weapons Systems Evaluation Division, Institute for Defense Analyses*, 1962.
- [21] R. Sridharan, "The capacitated plant location problem," *European Journal of Operational Research*, vol. 87, pp. 203–213, 1995.
- [22] A. Tarantola, *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Amsterdam, Ed. Elsevier, 1987.
- [23] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control," *AAAI Spring Symposium on Human Behavior Modeling*, 2009.
- [24] T. D. Nielsen and F. V. Jensen, "Learning a decision maker's utility function from (possibly) inconsistent behavior," *Artificial Intelligence*, vol. 160, pp. 53–78, 2004.
- [25] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of Guidance Control and Dynamics*, vol. 30, pp. 370–382, 2007.
- [26] A. Trautman, P.; Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*, pp. 797–803, 2010.
- [27] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
- [28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [29] D. Goldfarb, "Curvilinear path steplength algorithms for minimization which use directions of negative curvature," *Mathematical Programming*, vol. 18, pp. 31–40, 1980.
- [30] J. More, "Recent developments in algorithms and software for trust region methods," *Mathematical Programming*, pp. 258–287, 1983.
- [31] M. J. D. Powell, "Convergence properties of a class of minimization algorithms," *Nonlinear Programming*, vol. 2, pp. 1–27, 1975.
- [32] J. More and D. Sorensen, "On the use of directions of negative curvature in a modified newton method," *Mathematical Programming*, vol. 16, pp. 1–20, 1979.
- [33] R. P. Brent, *Algorithms for minimization without derivatives*, N. Englewood Cliffs, Ed. Prentice Hall, 1973.
- [34] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, W. Rheinboldt, Ed. Academic Press, 1982.
- [35] H. A. Taha, *Operations Research: An Introduction*, 8th ed. Prentice-Hall, 2007.
- [36] J. Shapiro and A. P. Sloan, *Mathematical Programming, Structures and Algorithms*. Wiley-Interscience, 1979.
- [37] G. Archavaleta, J. P. Laumond, H. Hicheur, and A. Berthoz, "An optimality principle governing human walking," vol. 24, no. 1, pp. 5–14, Feb. 2008.

- [38] K. Mombaur, A. Truong, and J.-P. Laumond, “From human to humanoid locomotion—an inverse optimal control approach,” *Autonomous Robots*, vol. 28, no. 3, pp. 369–383, 04 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10514-009-9170-7>
- [39] M. J. D. Powell, “The bobyqa algorithm for bound constrained optimization without derivatives,” *Centre for Mathematical Sciences, University of Cambridge, UK.*, 2009.
- [40] H. Bock and K. J. Plitt, “A multiple shooting algorithm for direct solution of optimal control problems,” *Proceedings of 9th IFAC World congress, Budapest. International Federation of Automatic Control.*, pp. 242–247, 1984.
- [41] A. S. Puydupin-Jamin, M. Johnson, and T. Bretl, “A convex approach to inverse optimal control and its application to modeling human locomotion,” *IEEE International Conference on Robotics and Automation, St Paul, MN.*, 2012.
- [42] A. Keshavarz, Y. Wang, , and S. Boyd, “Imputing a convex objective function,” *Proceedings IEEE Multi-Conference on Systems and Control*, To appear September 2011.