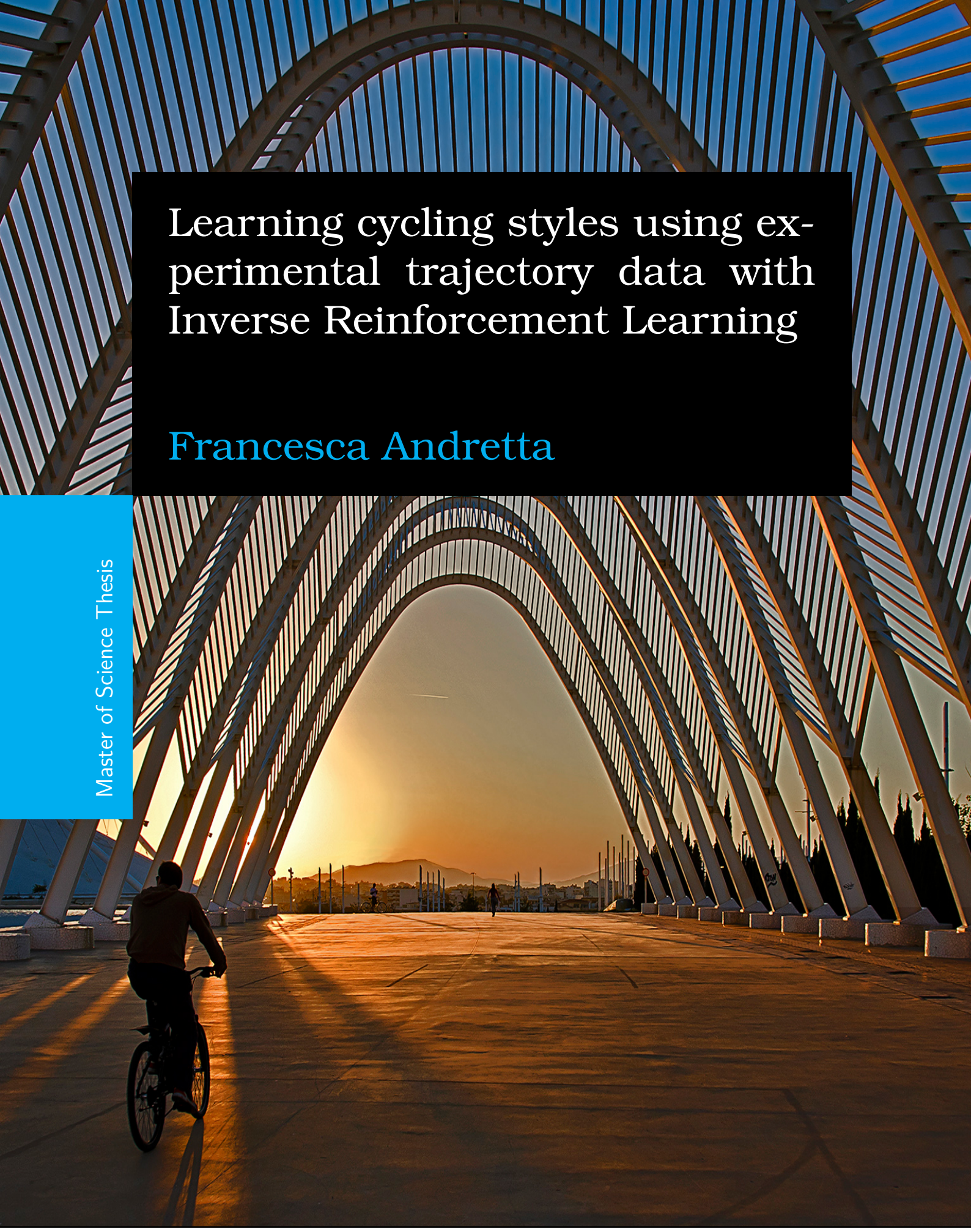
Learning cycling styles using experimental trajectory data with Inverse Reinforcement Learning

Francesca Andretta

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Learning cycling styles using experimental trajectory data with Inverse Reinforcement Learning

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Francesca Andretta

April 20, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.

Abstract

Cycling is an increasingly attractive transportation mode, thanks to its health and environmental benefits. Personalized travel assistance services can help make cycling more appealing by providing speed or route advices that can reduce travel time and increase safety while taking into account the personal preferences of cyclists. Due to its ability to learn agents' reward function, Inverse Reinforcement Learning is a suitable algorithm for learning cycling preferences from data.

This thesis aims to describe cycling styles as a set of cycling preferences encoded as a reward function composed of a weighted sum of features. The weights associated to the features composing the reward function represent the importance given to each cycling preference and express the trade-off between different goals of a cyclist. Continuous-time Inverse Reinforcement Learning extracts the weights from empirical cyclists' trajectories collected during an experiment performed in Delft. During the experiment, cyclists were asked to cycle according to three different cycling styles: cautious, normal and aggressive. Differences between weight sets extracted for each cycling styles were analyzed by means of the Kruskar-Wallis statistical test and K-Means clustering algorithm, and the averaged weights for each cycling style were used to simulate a set of test trajectories.

It is shown by simulations that the reward function identified for a specific cycling style leads to an improvement in terms of similarity to test trajectories with the same cycling style with respect to the reward functions corresponding to other cycling styles. The statistical analysis shows that the weights of cautious and aggressive cycling styles show statistical differences and define separate clusters.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors Azita Dabiri and Jason K. Moore for their assistance and support during the writing of this thesis. My friends, that became my family: Alessandro, Charalampos, Iannis, Evie, Milan, Pietro, Marco, Davide, Carlos, Cristian, Riccardo, Giovanni, Emilio, Irene and many others. My housemates Joseph, Stella, Kat, Armin; I will always feel at home with you. My friends in Italy, we never lost each other. Marco, we are always together, even when we are far away. To my family, that always supported me, whom I miss so much.

Delft, University of Technology
April 20, 2022

Francesca Andretta

Chapter 1

Introduction

In the past years, green mobility solutions have been gaining interest and the demand for alternative means of transportation like bicycles, e-bikes and scooters is raising.
Among them, bicycles are particularly valuable in terms of environmental sustainability, ease of use and popularity. Cycling is becoming increasingly appealing, and it is particularly widespread in Northen-European countries like Germany, Netherlands, and Denmark [10]. The change in mobility habits towards bicycles leads to the necessity of infrastructures and services suitable to cyclists and their traffic patterns. On this line, research on modelling and simulation of cyclists is advancing. By understanding cyclists' behaviour, mobility services can be designed in order to improve user experience. One of the recent developments in this field is given by personalized travel assistance services, e.g., personalized speed or road advice systems which provide speed, acceleration or route recommendations to cyclists in order to reduce their travel time, increase the chances of catching a green light or optimize the route for other personal preferences. The algorithms behind these services rely on a model of cycling behaviour, in terms of cycling preferences or reaction to the advice. Once the model of an individual cyclist is learned, the assistance service can be tuned in order to meet his preferences and characteristics, providing a personalized travel service.
Personalization as an engineering technique applied to travel services has been extensively researched in the driving domain, especially in the fields of advanced driver assistance (ADAS) and autonomous driving. In the former case, a driver's acceptance of the ADAS strongly depends on his individual characteristics: for example, an aggressive driver who drives in a risky way and often triggers the system will be bothered by the amount of warnings of a system that a more cautious driver would perfectly accept. Hence, an aggressive driver could ignore or disable the system. Similarly, for autonomous driving, the vehicle should provide a comfortable travel experience for the user, but the definition of comfort may vary among different persons.
Driving style refers to the habitual way of driving of an individual or a group of drivers [18]. From an engineering prospective, driving style is a complex notion involving several factors in its description. The action of driving can be defined as a list of driving patterns, such as accelerating, decelerating, overtaking etc, that are related to the external conditions (e.g. road type and weather) and on the human intentions and preferences.

This last concept can be defined as the driving style. Here, developing driving style identification techniques is interesting because it allows the prediction of driving actions of a user, given a set of external conditions. Driving style recognition has been performed on the basis of identification of specific driver preferences, usually in terms of increasing safety or minimizing fuel consumption, or specific behaviours, as unpredictable manoeuvres or aggressive interactive behaviour. Driving preferences are defined as personal inclinations towards different characteristics of the driving style such as minimizing fuel consumption, maximising safety and comfort and minimizing travel time. In the literature, different categories of driving styles have been defined based on their driving preferences. As an example, aggressive driving style has been connected to driver's preferences for high speed, abrupt acceleration and high fuel consumption, while cautious drivers tend to avoid dangerous situations and keep a stable, lower speed [6]. Thanks to the large amount of available traffic data containing valuable information, data-driven approaches have been increasingly employed and improved in this field to learn personal preferences from data and thus characterize different individuals' driving style.
For cyclists, limited literature on personalized cycling assistance is present. Dabiri et al. [7] propose an approach to give optimal acceleration advice to a cyclist with respect to his preferences in cycling, but the preferences are assumed to be manually assigned by the cyclist himself. Identification of cycling style by means of data-driven approaches has not yet been researched. Analogous to what have been developed in the automotive industry, this thesis proposes a method to learn the cycling style of each user from observed trajectory data.
Different data-driven driving style characterization techniques have been explored in the literature. Clustering algorithms grouping driving styles into a discrete number of classes as "aggressive","moderate" and "conservative" are widely used [6] [30]. Other approaches model drivers' behaviour as a pattern recognition problem, using Hidden Markov Models to characterize long-term driving intentions like lane-changing, lane-keeping, acceleration or turning [32]. Another promising method is to represent drivers preferences with a reward function that is learned from data by means of variants of the Inverse Reinforcement Learning (IRL) algorithm.
IRL is a method originated in the robotics field that has been employed to learn reward functions from human demonstrations. In the literature, several successful applications to the driving domain are present, mostly aiming to extract the driving style in order to improve the user experience for autonomous vehicles [15] [13] [31] [25] and to propose improved and personalized ADAS [11]. These research works assume that drivers optimize a set of high-level goals, like optimizing safety, comfort, energy consumption etc. From a mathematical point of view, this notion can be encoded by a reward function. Within the context of IRL for driving style identification research, reward function are usually defined as a weighted sum of a set of features, each one related to a characteristic of the driving action. These weights may vary from person to person, and the collection of the different features and their corresponding weights characterizes the driving style. Driving styles are identified by means of the IRL algorithm, which quantifies the driving preferences of a driver by identifying the weights values of his internal reward function.
IRL is gaining interest thanks to its ability of reflecting the human's decision-making process and generalization capabilities. In fact, IRL provides a description of individual drivers' long-term behaviour by characterizing the individual driver himself through his internal reward function, instead of learning his behaviour during specific manoeuvres. Moreover, this formulation describes a set of preferences for each individual driver rather than assign him to a

cluster, which gives a more detailed and personal model of the individual and is particularly advantageous in view of a possible personalized travel service.

1-1 Problem statement

IRL has shown promising results in learning driving styles and might be successfully applied to the cycling domain. The main goal of this thesis will be to investigate the possibilities of this algorithm for learning cycling styles from observed trajectory data. The research question this thesis is addressing is:

> *Given a set of fixed external conditions, can different cycling styles be learned from cyclists' trajectories using Inverse Reinforcement Learning?*

To answer the main research question, a set of sub-questions are designed. The first step is defining experiments with the goal of collecting informative data, so this sub-question arises:

> *How to design and implement an experimental setup that provides the trajectory data necessary to implement and test the proposed algorithm?*

In order to learn cycling styles, meaningful features that can capture human preferences in the form of a reward function need to be designed.

> *Which features can represent cycling styles and how can they be mathematically expressed?*

Once the reward function's shape is defined and the algorithm learns a set of weights from each trajectory, their validity should be discussed. One possible criterion is that differences in cycling preferences should imply different weight values. This leads to the following sub-question:

> *Do the identified weights related to different cycling styles show statistical differences?*

The extracted weights define reward functions that can be used to simulate cyclists' trajectories. Assuming that cyclists with the same cycling style have a similar set of preferences defining their internal reward function, the identified reward function can be used as a model that can simulate trajectories that fit the empirical ones.

> *Can the learned reward function simulate trajectories that fit the empirical ones?*

1-2 Thesis outline

The outline of this thesis is as follows. An introduction to Inverse Reinforcement Learning and its applications to traffic domain is presented in Chapter 2. Chapter 3 presents the experimental setup, including characteristics of sensors, location, participants and instructions. Chapter 4 describes the proposed approach and the features used to describe cycling styles are described in detail. The same chapter presents the employed data processing techniques. The performance of the proposed method is evaluated in Chapter 5 with a convergence analysis, a simulation study and statistical analysis of weights.
This thesis is concluded with an overall discussion and future recommendations in Chapter 6.

Chapter 2

Inverse Reinforcement Learning

2-1 Introduction

Inverse Reinforcement Learning is an approach originated by learning from demonstrations developed in the robotics field, but has also been explored in other domains such as traffic applications, cognitive science and video-games development [17] [29] [14].
While Reinforcement Learning aims to find an optimal policy, defined as the optimal sequence of actions per states for an agent given a reward, the goal of Inverse Reinforcement learning is to model the behaviour of an agent from observed demonstration by inferring its reward function.
This chapter starts with an introduction to the Inverse Reinforcement Learning algorithm and its theoretical background, and continues with a section dedicated to the versions of IRL used for traffic applications.

2-2 Environment description - Markov Decision Process

The interaction of a human agent with the environment is often formulated as a Markov Decision Process (MDP), represented by:

$$M = (S, A, P(s'|s,a), R(s,a)) \tag{2-1}$$

with S the finite set of states or environment, A the finite set of actions, P the transition probability distribution, where $P(s'|s,a)$ is the probability of an agent of being in state s' after taking action a in state s and $R(s,a)$ the reward function which encodes the utility of an agent taking action a in state s. An optional discount factor γ on rewards is absorbed into the transition probabilities. At each time t, the state $s_t \in S$ describes the environment and the human chooses an action $a_t \in A$. For each state s, a deterministic policy π is a function that outputs an action $a = \pi(s)$. A stochastic policy is defined as $\pi(s,a) = P(a_t = a|s_t = s)$, namely a probability distribution on the action to be taken in each state. A policy's expected

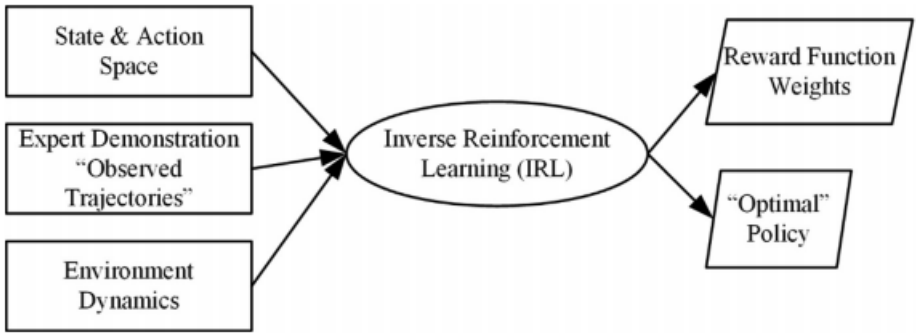


Figure 2-1: Structure of Inverse Reinforcement Learning by Alsaleh et al. [3]

reward is the expected sum of rewards that will be obtained if the policy is followed and a strategy that aims to maximize this expected reward is said to be optimal.
In a Markov Decision Process, each state only depends on the previous one and on the policy.

2-3 Background

Given a set of human trajectories $\{\tau_i\}_{i=1}^{N}$ consisting of a number of state-action pairs $\tau = ((s_0, a_0), (s_1, a_1), ..., (s_n))$, Inverse Reinforcement Learning aims to recover a reward function such that the behaviour of the agent fits the observations. A scheme representing the general high-level structure of Inverse Reinforcement Learning algorithm is shown in Figure 2-1. On the left, the inputs such as the expert demonstrations, the environment dynamics given by the transition probability and the environment description as Markov Decision Process. On the right, the expected outcomes of the algorithm is the optimal reward function weights which result in an optimal policy.
For many purposes, it is possible to assume that the reward function is linear over some features, and can be defined as the weighted sum of a number of features, described with mathematical expression involving quantities that can directly be evaluated from data. Hence, the reward function can be expressed as:

$$R(s, a) = -w^T \Phi(s, a), \tag{2-2}$$

where w is the weight factor and Φ the feature vector, with features defined as mappings from state-action space to real values which capture important properties of the observed behaviour. Diametrically, $w^T \Phi(s, a)$ can be interpreted as a cost function.
Given the feature-dependent parametrization of the reward function, a key concept in Inverse Reinforcement Learning is that the difference between empirical and expected feature values can be considered a measure of similarity between the empirical and generated trajectories. Early works [2] propose a method that recovers reward weights such that trajectories derived from this reward function and expert trajectories have similar features expectation, called Feature Matching. The core of the algorithm consists of finding a policy such that a planner based on this policy and the expert trajectories have almost equal features expectation. Feature Matching IRL does not necessarily find the correct reward weights, but recovers a policy that leads to trajectories close to the expert ones in terms of feature values.

Recovering reward weights from demonstrations imposing feature matching only is an underdefined problem. Many weight values and thus reward functions can explain the observed behaviour. For example, if a user chooses random actions, this would result in a constant reward function for which each behaviour is optimal.
Ratliff et al. [24] apply the Maximum Margin concept to IRL and formulates it as a quadratic programming problem. In this framework, the algorithm aims to find a solution better than any other solution by a margin. The margin depends on a loss function, which is chosen as the count of the number of states the planner visited but not the expert. This way, the margin is larger for policies that are very different from the demonstrated optimal policy. As Ziebart points out in his thesis [33], even if the method has the advantage of recovering a unique set of weight parameters, it still has a problem: it may be possible that the weights and consequently the reward function can't make the demonstrated behaviour optimal and better than the other possible behaviours. This issue may arise in a situation for which the demonstrated behaviour is not perfect or the reward function does not capture enough characteristics to describe human behaviour. A feature boosting approached was introduced in order to provide a more complete description of the behaviour and to address its non-optimality. This method has shown to improve the original maximum margin approach, but suffers from possible overfitting of feature definition on demonstrated behaviour.
Ziebart et al [34] introduced a probabilistic mathematical framework accounting for the non-optimality of demonstrations based on the maximum entropy principle. The observation are assumed to be the result of a human taking stochastical and near-optimal actions. The demonstrated trajectories are sampled from the distribution:

$$P(\tau_i|w) \propto e^{-w^T \Phi_{\tau_i}} \tag{2-3}$$

where the probability of choosing a sub-optimal trajectory is exponentially decreasing with respect to the trajectory's cost. The agent's reward function is the one maximizing the average log-likelihood $L(w)$ of the given N demonstrations:

$$w^* = \arg\max_w L(w) = \arg\max_w \sum_{i=1}^{N} \log(P(\tau_i|w)) \tag{2-4}$$

However, it is not possible to analytically compute w^* from this expression since the probability distribution $P(\tau|w)$ of trajectories τ given a set of weights w is intractable analytically. Hence, w^* has to be found with gradient-descent methods and hence requires the gradient of $L(w)$ [15].
The gradient of $L(w)$ can be shown to be the difference of feature expectations between the empirical trajectories and the generated ones [33]:

$$\nabla_w L(w) = \widetilde{\Phi} - E_{P(\tau|w)}[\Phi], \tag{2-5}$$

with $\widetilde{\Phi}$ the empirical feature value calculated from empirical demonstrations, and $E_{P(\tau|w)}[\Phi]$ the feature expectations, which is the expectation of feature values Φ over the the distribution of trajectories given a set of weights $P(\tau|w)$, defined previously.
Intuitively, if the expected value of a feature k is higher than the empirical one, it means that the weight corresponding to that feature should increase. This way, since the cost is minimized, the likelihood of trajectories with high value for the feature k decreases.

The main challenge in calculating 2-5 is to calculate the feature expectations $E_{P(\tau|w)}[\Phi]$, which has the form:

$$E_{P(\tau|w)}[\Phi] = \int P(\tau|w)\Phi(\tau)d\tau \tag{2-6}$$

Since it requires integrating the whole set of trajectories. As for 2-4, the probability distribution $P(\tau|w)$ of trajectories τ given a set of weights w is not computable analytically. Ziebart et al. [33] use a dynamic programming forward-backward algorithm to calculate the feature expectation, but such an approach is intractable for large state-action spaces.
Most of the literature on Inverse Reinforcement Learning is based on Ziebart's Maximum Entropy IRL method. Several sampling-based methods have been proposed to solve the problem of feature expectation calculations. Sampling-base techniques applied to IRL consist of deriving the expected feature values by sampling trajectories from a sampling probability distribution of trajectories $q(\tau)$. In Boularias et al., [4], a uniform baseline probability distribution is suggested while in Finn et al. [9] the probability distribution is refined over time [26].
Finn et al. [9] propose Guided Cost learning algorithm, a deep-learning based approach that brings two main improvements to the Inverse Reinforcement Learning framework: this algorithm is capable of learning nonlinear reward functions, suitable for high-dimensional robotics application, and handles unknown model dynamics by performing a sampling-based approximation of the gradient. Guided Cost learning algorithm [9] goes further than other sampling-based Inverse Reinforcement Learning methods by only generating trajectory samples which are useful for the estimation of the partition function, but suffers in terms of interpretation of the features.

2-4 Inverse Reinforcement Learning for traffic applications

Continuous and large scale action-spaces typical of the driving domain pose a challenge in the calculation of the expected feature values of equation 2-6 with Maximum Entropy IRL algorithm. In order to simplify the algorithm, some convenient domain-dependent assumptions can be made. Related works defined a framework for a continuous-time version of IRL that can represent the environment in traffic domain and avoid the problems associated with the discretization of a large state-action space.
One possible approach is to suppose that a human driver is an optimizer who only performs the best actions in terms of the internal reward function, so the algorithm generates the best path according to the recovered reward function. Such an approach is known as Inverse Optimal Control (IOC), and will presented in Section 2-4-2. On the other hand, a trajectory sampling IRL approach described in Section 2-4-1 supposes that a human driver can mentally construct numerous potential trajectories and choose one to execute based on the associated rewards. According to this assumption, the algorithm generates a set of feasible trajectories and the most likely one is chosen.
In both cases, a traffic-domain suited version of IRL produces trajectories, intended as position in time, instead of state-action sequences as in Maximum Entropy IRL. Since the trajectories generated by the algorithm mimic human driver trajectories, they should be smooth and dynamically feasible. In particular, acceleration and jerk should be continuous in time.
Most of the literature on IRL for automotive domain assumes that the generated trajectories

have been approximated as two-dimensional polynomial functions of time of order four or five [15],[13].

$$\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + ... + a_n t^N \\ y(t) &= b_0 + b_1 t + b_2 t^2 + ... + b_n t^N \end{aligned} \tag{2-7}$$

By parametrizing the trajectory as a function of time of a certain order, velocity and acceleration are continuous and can be computed as analytical derivatives of the trajectory function. In order to reduce computational effort, it is possible to optimize only selected points of the trajectories. Kuderer et al. [15] employ splines to represent trajectories. The trajectory is given by a set of S spline segments that define the trajectory in a time interval $[t_k, t_{k+\delta t}]$:

$$r(t) = s_k(t) \quad (t \in [t_k, t_{k+\Delta t}]), \tag{2-8}$$

where $r(t)$ is the trajectory as a function of time, k denotes an interval of the trajectory and s_k denotes the time segment defined for time in the interval $[t_k, t_{k+\nabla t}]$. Each interval shares the start and end points with the previous and following interval. These points, called control points, are a subset of the total number of points of the spline and are the only ones optimized during the algorithm.

2-4-1 Trajectory Sampling Inverse Reinforcement Learning

As presented in the introductive Section 2-4 , Trajectory Sampling IRL emerges as one of the main IRL versions used in traffic applications.
The hypothesis of this method is that since the human driving follows the constraints of traffic rules and the motion of the vehicle, the space of possible trajectories can be reduced to some small subspaces. Therefore, it is possible to assume that the human driver preplans a limited number of trajectories and then select one to follow. Hence, it is reasonable to approximate the expected feature values with the average of feature values derived from multiple generated feasible trajectories:

$$E_P(\tau|w)[\Phi] \approx \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{e^{w^T \Phi_{T_i^j}}}{\sum_{j=1}^{M} e^{w^T \Phi_{T_i^j}}} \Phi_{T_i^j} \tag{2-9}$$

Where w is the set of weights, $\Phi_{\tau_i^j}$ is the feature vector extracted from the trajectory τ_i^j, which is one of the N feasible trajectories generated given the initial position and velocity of τ^j, one of the M empirical trajectories.
Feasible trajectories are generated by representing the trajectory as a polynomial or other kind of function. Given initial position and velocity and boundary equations, the final position and velocity are sampled. By solving the boundary conditions resulting from imposing the initial and sampled final conditions, the coefficients of the polynomial are derived and therefore a trajectory is generated.
The feature values can be directly derived from the trajectory, thus a set of weights can fully characterize the reward functions associated to each generated trajectory. The algorithm optimizes the weight values by gradient-descent: given a set of weights at an iteration, the feature expectations are found according to Equation 2-9. This value is used to calculate the gradient as the difference between expected and empirical feature values, thereafter a new

weight vector is updated using this gradient. The new weight vector is used in the following iteration, and the process is repeated until convergence.
The pseudo-code of the algorithm used by [13] and [31] is given as:

1. Empirical feature vector is computed from demonstrations:
$\widetilde{\Phi} = \frac{1}{N}\sum_{i=1}^{N}\Phi_{\tau_i}$, weight vector w is randomly initialized, learning rate α a ϵ
2. For each trajectory τ^j in the set of empirical trajectories
    2.a Fix the environment start
    position, velocity and acceleration and derive a sampling space in terms
    of free parameters of this trajectory represented by a polynomial function
    to be sampled.
    2.b A set of M of trajectories is generated by sampling the free parameters.
    2.c For each trajectory in the set:
        2.c.1 Calculate the features values of the trajectory $\Phi^j_{\tau_i}$
        2.c.2 Add the feature vector to the buffer of feature vectors:
        $B^j_i \leftarrow \Phi^j_{\tau_i}$
3. While $||\widetilde{\Phi} - E_{P(\tau|w)[\Phi]}|| \geq \epsilon$ do:
    3.a Compute the sampling-based approximation of the expected feature values using
    the trajectory stored in the buffer B
    $E_P(\tau|w)[\Phi] \approx \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{e^{w^T\Phi_{T_i^j}}}{\sum_{j=1}^{M} e^{w^T\Phi_{T_i^j}}}\Phi_{T_i^j}$
    3.b Compute the gradient:
    $\nabla = \widetilde{\Phi} - E_P(\tau|w)[\Phi]$
    3.c Update the weight vector w_{i+1} with the gradient
    $w_{i+1} = w_i - \nabla\alpha$
end

In Figure 2-2, the framework of Trajectory Sampling Inverse Reinforcement Learning used by [13] and [31] for driving style identification is presented. Their assumption is that a driver focuses on high-level goals that produce high-level decisions (e.g. lane-changing/acceleration, lane-keeping/deceleration). Before taking a decision, a driver generates different trajectories in his mind (on the top of the figure) and evaluates them according to his internal reward function (at the center of the figure). Each potential trajectory has a probability exponentially depending on the reward function, and the driver selects trajectories according to this probability distribution (right part of the figure).

2-4-2 Inverse Optimal Control

Several interpretation of the term "Inverse Optimal Control" can be found in the literature. In some papers, IRL and Inverse Optimal Control are equivalent while in others, Feature Matching Inverse Reinforcement Learning is called Inverse Optimal Control [33]. Classical Inverse Optimal Control is described as the solution of optimal control problems: Inverse Optimal Control consists of finding the reward function given a stabilizing control law.
In this section, the focus will be on the approach referred to as modern Inverse Optimal Control, which derives from the IRL framework [1], and can be denoted as Optimal IRL. Inverse Optimal Control is based on the assumption that demonstrations are optimal with respect to the reward function and that the feature expectation vector $E_{P(\tau|w)[\Phi]}$ can be

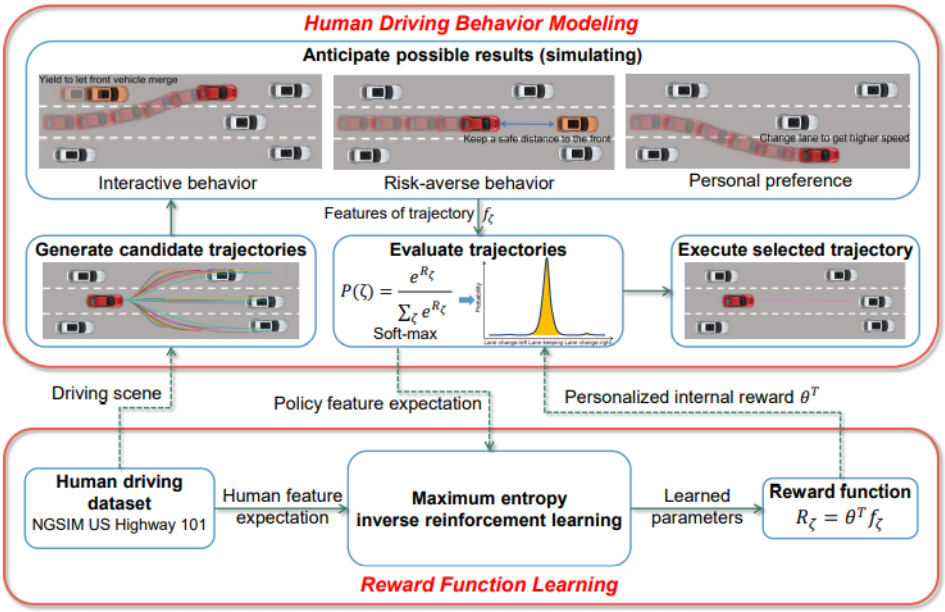


Figure 2-2: Inverse Reinforcement Learning for driver behaviour modeling by Huang et al. [13]

approximated as the feature values vector of the most likely trajectory.

$$E_{P(\tau|w)[\Phi]} \approx \Phi(\arg\max_{\tau} P(\tau|w)) \approx \frac{1}{N}\sum_{j=1}^{N} \Phi_{\tau^j} \tag{2-10}$$

Where Φ_{τ_j} are the feature values of the optimal trajectory associated to each one of the N empirical trajectories. The most likely trajectory's feature vector is calculated as the average of the feature vectors calculated from the trajectories generated by optimizing the parameters following the assumption that demonstrations are not assumed to be drawn from a probability distribution, but they are generated by actually minimizing a cost function. This looks like a rough approximation, but it has been shown to perform well in driving applications.
As for Trajectory Sampling IRL, Inverse Optimal Control algorithm optimizes the weight values by gradient-descent, the only difference is given by the expression of the feature expectation and thus the calculation of the gradient. At each iteration a set of weights is given, which fully defines a reward functions associated to a trajectory since the feature values are a function of the trajectory only. For each empirical trajectory, the initial condition are fixed. The trajectory maximizing the reward function associated to the weight vector at that iteration is found and the feature values of this optimal trajectory are calculated. The process of finding the optimal trajectory given initial conditions, constraint and reward function and the subsequent derivation of the feature values is repeated for all the empirical trajectories at each iteration. After all the optimal trajectories given the reward function at that iteration have been found, the feature expectations are found according to Equation 2-10 by averaging the feature values of the optimal trajectories associated to the different empirical trajectories starting states. This value is used to calculate the gradient as the difference between expected and empirical feature values, and a new weight vector is updated using this gradient. The new

weight vector is used in the following iteration, and the process is repeated until convergence. The steps of the algorithm are summarized as follows:

1. The empirical feature vector is computed from demonstrations:
$\widetilde{\Phi} = \frac{1}{N}\sum_{i=1}^{N}\Phi_{\tau_i}$,
the weight vector w is randomly initialized, learning rate α and stop threshold ϵ are given
2.While $||\widetilde{\Phi} - E_{P(\tau|w)[\Phi]}|| \geq \epsilon$ do:
    2.a For each trajectory τ^j in the set of empirical trajectories
        2.a.1 Fix the environment start position, velocity and acceleration and find the optimal trajectory with respect to the reward function $R_i = -w_i^T\Phi_{\tau_i}$:
        $\tau_i^{j*} = \text{argmin } (-R(\tau, w_i))$
        given C road, velocity and acceleration constraints.
        2.a.2 Derive the feature values $\Phi_{\tau_i^{j*}}$ from the optimal trajectory τ_i^{j*}
    2.b Compute the expected feature values by evaluating the feature function for all the optimized trajectories:
    $E_P(\tau|w)[\Phi] \approx \Phi_{ML}^w = \frac{1}{N}\sum_{j=1}^{N}\Phi_{\tau_i^{j*}}$
    2.c Compute the gradient:
    $\nabla = \widetilde{\Phi} - E_P(\tau|w)[\Phi]$
    2.d Update the weight vector w_{i+1} with the gradient ∇:
    $w_{i+1} = w_i - \nabla\alpha$
end

2-5 Conclusions

Research on driving style identification can be used as a useful starting point for answering the research questions of this thesis. IRL has shown to be promising in learning driving style, and could be suitable for identifying cycling style as well.
It can be argued that external factors have a smaller impact on cycling behaviour with respect to driving behaviour. In particular, cyclists are usually not strictly constrained into lanes, and they are not required to comply to speed limits. As a consequence, the behaviour of cyclists depends on a stronger way to internal factors, such as the personal cycling preferences, thus it is very heterogeneous [22]. IRL demonstrated to be effective in modelling human preferences with the notion of a reward function. In traffic applications, it has the advantage of giving a general model of the road user's internal reward function motivating his decisions rather than describing his behaviour during specific manoeuvres or considering specific parameters. In particular, the last strategy is the one used in clustering approaches, which consider a set of features of the driving or cycling action (e.g. speed, abrupt acceleration, fuel consumption) and then cluster them according to some notion of distance between their values, but it is agnostic of the model behind the decisions of performing determinate cycling or driving actions which lead to these values.
For these reasons, IRL is considered to be a good method to identify cycling preferences, that define the cycling style of an individual. In the next chapters, the development of an experimental study and the details of the proposed approach will be discussed.

Chapter 3

Experiments

3-1 Introduction

In the absence of available public datasets for cyclist trajectories, part of this thesis project was devoted to an experimental study, involving the development of the experimental setup and performing the experiments. The first part includes the description of the experiment (scenario, participants, instructions) and data collection (sensors). After performing the experiments, data were processed in order to be suitable inputs for the cyclist preference extraction algorithm as it will be described in Section 4-3. The type of data necessary to study individual cycling behaviour is trajectories, namely cyclist position in time. From trajectories, it is possible to derive velocity, acceleration and distance from road edges.
Gavriilidou et al. [10] present a methodology to set-up a large-scale cycling experiment. Three main data collection approaches can be used to gather trajectory data:

- Observation of real-life cycling: guarantees absence of bias and influence on the behaviour of cyclists.
- Cyclist simulator: The validity and performance of cyclists virtual reality simulators is unknown. The data gathered with a valid simulator are independent on external conditions, which makes it a potentially useful tool for future research.
- Controlled experiment: This approach guarantees controllability on the participants and the scenario design. When performed in an artificial environment (e.g. a gym), it is possible to control weather conditions. The main problem that may arise is the learning effect, namely the impact of familiarity with the experiment on participants behaviour. Other factors as time and fatigue contribute to modifications in participants behaviour as well. This disadvantage is stronger in experiments that require participants to behave naturally, as they were observed during their every-day life activities. Another drawback is that usually participants are selected with a certain bias, so the results may not be representative of all population. A way to counterbalance this effect is to choose participants from different age groups, gender and other experiment-dependent characteristics.

3-2 Proposed experimental setup

The experiment was designed in order to provide informative data in terms of different cycling preferences. The participants were asked to cycle according to a set of instructions on their cycling style to force variability of shown cycling preferences.
In the next section, a detailed explanation of the experimental plan, including a description of the chosen scenario, participants and instructions will be given. The following section presents the data collection plan, with the choice and characteristics of used sensors.

3-3 Experimental plan

3-3-1 Scenario

The chosen location was Shoemakerstraat, in Delft. The path is shown in Figure 3-1, with numbered intersections and a blue dot for the starting point right below the Kruithuisweg causeway. The chosen path includes five intersections, two long straight segments and four turns.
The route involves intersections, where cyclist behavior is informative in terms of tendency to accelerate or decelerate abruptly in order to stop or avoid the traffic light, straight segments where cyclists can show how fast they are willing to ride and at turns, where the distance with respect to the borders of the road and the difference with respect to the desired velocity are particularly important factors. The path is wide enough to allow overtaking in a safe way.
The traffic lights of the first, second, third and fourth intersections are adaptive. The fifth one can be manually triggered: if the button is pressed, it goes green. Most of the traffic lights cannot be controlled, and being adaptive there is no way of knowing the exact timing. The variable traffic light has been considered as variability factor intrinsic in the non-artificial experimental setup. Most of the times the adaptive traffic lights were red, so the participants performed the experiments under similar traffic lights configuration. However, the trajectories are grouped together according to the cycling style or the kind of road segment, such as straight, curve (approaching a stop, starting from a stop, in the middle of the path), so having the participants performing the experiment under the same traffic light conditions is not a necessary requirement.
The route is marked in the Figure 3-1: the participants start cycling from the starting point (blue dot), if necessary stop at the first intersection, continue straight until the second intersection, ride through the S-shaped segment until the third intersection, cross the road, cycle through the second long straight segment, crosses the road at the fourth intersection, and go back to the initial point. In order to facilitate participants' route following, a set of arrows indicating the direction were positioned in the path.

3-3-2 Instructions

In the context of this preference-learning algorithm, the collected data should show differences in cycling preferences. Without this requirement, the experimental data could show little differences in terms of cycling preference and thus it would not be possible to evaluate the

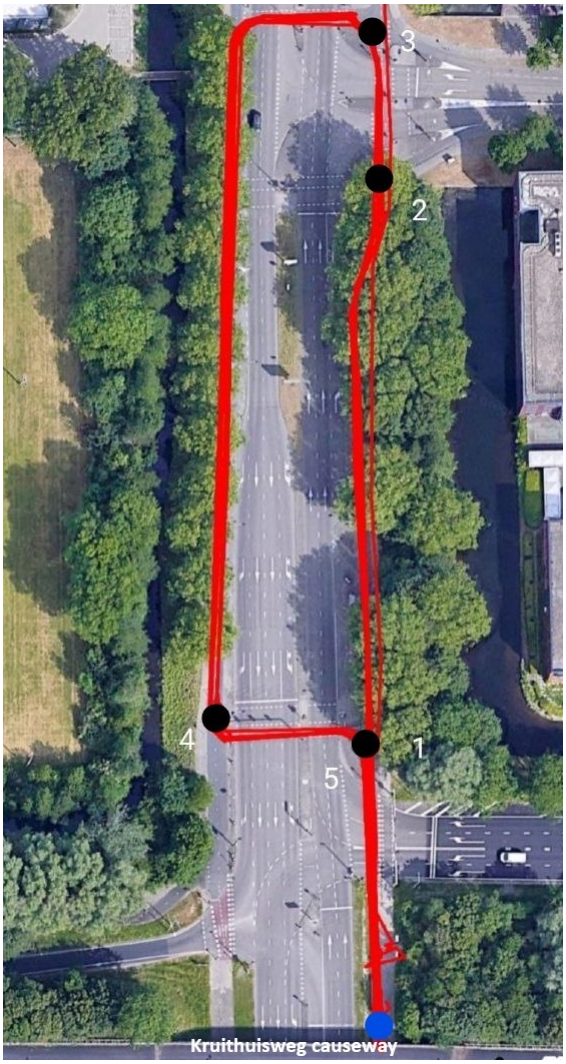


Figure 3-1: Location and route chosen for the experiment. The blue dot is the starting point, the black dots represent the intersections and the number shows their order in the route. The red lines show an example of the collected trajectories of one participant.

performance of the algorithm.
In order for the experiment to follow this requirement, participants were asked to ride the same route with three different cycling styles aiming to show different cycling preferences:

- Style 1: The rider is asked to keep maximum distance from road edges, obstacles, and other cyclists, never overtake and avoid abrupt acceleration.
- Style 2: The cyclist is asked to ride as he would normally do, ideally trying to maintain the desired speed and overtake only if the cyclist ahead is slower.
- Style 3: The cyclist is instructed to minimize his travel time through the course without breaking the law or endangering anyone. He should overtake if given the chance.

3-3-3 Participants

Ten healthy 23-28 years old male participants were recruited among TU Delft students. The participants have the same gender and are in the same age group because the main goal of this project is to distinguish three groups of cycling styles depending on the instructions, rather than the cycling style of the individual. The data has been anonymized to protect the participants' identity, and each subject has been given a unique identifying number. Participants were asked to bring their own bicycle, and were given an helmet for safety reasons. The experiment was approved by the Human Research Ethics Committee (HREC) of Delft University of Technology.

3-3-4 Time and duration

Two participants per time ride along the path, along with other road-users. Each participant does six laps, two for each cycling style, for a total of 2.5 km. Each lap approximately takes five minutes, including the waiting time at the stops. In order to avoid physical fatigue, two five-minutes breaks were scheduled, for a total duration of the experiment of 40 minutes per couple of participants. To minimize the risks arising from the request of cycling according to Style 3, all the experiments were performed during non-peak hours, when the presence of other road users is limited. In any case, participants were explicitly instructed to not endanger themselves or others. Moreover, they were asked to cycle according to their safety margins and to respect traffic rules, traffic lights and other road users.

3-4 Data collection

The sensors used during the experiment are differential Global Position Systems (GPS) in Real-Time Kinematic positioning (RTK) mode. This technology can reach centimeter accuracy by processing the measurements in differential mode.
The setup was provided by TU Delft GRS Lab, and consists in a low-cost u-blox ZED-F9P (U-blox, Thalwil, Switzerland) receiver which connects to the TU Delft GNSS station which provides real-time correction of the position data, as shown in Figures 3-2, 3-4. The position is returned in ellipsoidal coordinates (latitude and longitude) and needs processing in order

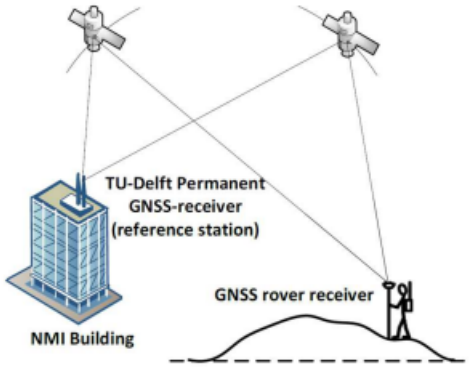


Figure 3-2: Setup for differential Global Positioning System (GPS), which contains a reference station, and a rover Global Navigation Satellite System (GNSS) receiver used in the field for surveying [5]

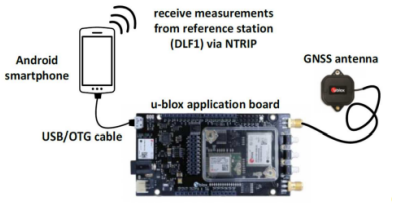


Figure 3-3: The low-cost u-blox GNSS receiver, used as a rover, offers RTK position solutions

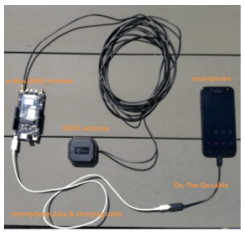


Figure 3-4: Connection of u-blox receiver and smartphone used by the participants

to be used, as will be explained in Section 4-1.
Each participant wore a backpack equipped with the differential GPS setup shown in Figure 3-4, with the antenna on the top, connected to the receiver and to a smartphone where the data file is logged.

Chapter 4

Proposed approach

4-1 Data post-processing

The position data of the trajectories collected during the experiments by means of the differential GPS sensors are in geographical coordinates form, as shown in Figure 4-1. They were transformed into cartesian coordinates by means of the function *latlon2local* of MATLAB [20], which projects the geographic coordinates in the ETRS89 standard to the local cartesian coordinates, as it is shown in figure 4-2. From the obtained data, the trajectories were considered from the starting point of the path. The initial parts of the trajectories before starting the experiment were discarded and each trajectory was divided into segments:

- Straight segments: long straight segments on the right and left of the path, where cyclists can overtake and accelerate, if asked to cycle according to style 3.
- Approaching stop segments and turns: in these road segments, cyclists decelerate to maintain stability or stop.

An example of a trajectory divided into these two category of segments is shown in Figure 4-3 In the results of Chapter 5, these two segments will be analyzed separately since difference in cyclist behaviour during straight segments and turns (e.g. acceleration profile) depends on the physical difference between the road segments rather than on a difference in cycling preferences. Hence, a relative difference between cycling styles among the same segment group trajectory can be analyzed. From the collected trajectories, 115 straight segments and 147 turn segments were extracted. In the literature on IRL for traffic applications, 90 to 300 short trajectories have been used, so the amount of gathered data is in line with the related research works [16] [27] [19] [13][31].
Each segment trajectory was smoothed with Savitzky-Golay filter with order 3 and window length one third of the length of the segment, following the literature on cycling style identification from position trajectory data [3] [21]. Then, the velocity was calculated by numerical derivation as:

$$v_x(t) = \frac{x(t+\delta t) - x(t)}{\delta t}, v_y(t) = \frac{y(t+\delta t) - y(t)}{\delta t} \tag{4-1}$$

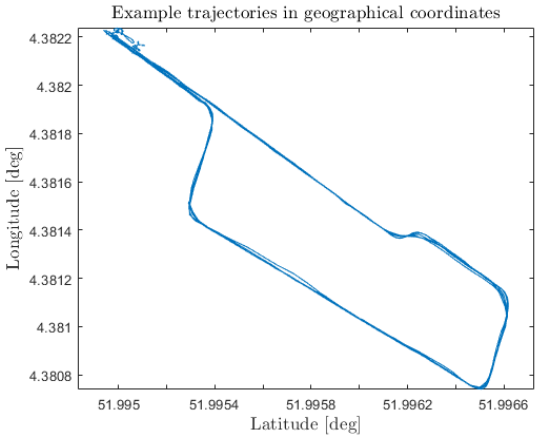


Figure 4-1: All the trajectories of one participant in geographical coordinates

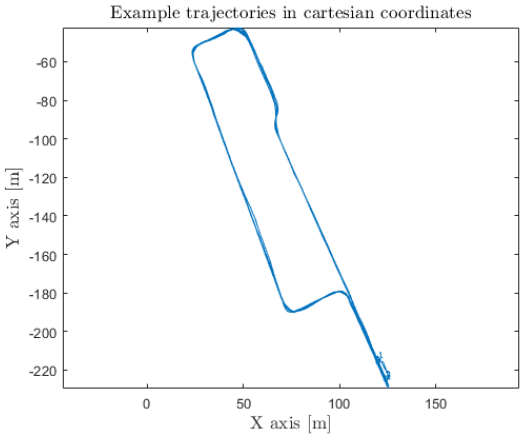


Figure 4-2: All the trajectories of one participant in cartesian coordinates

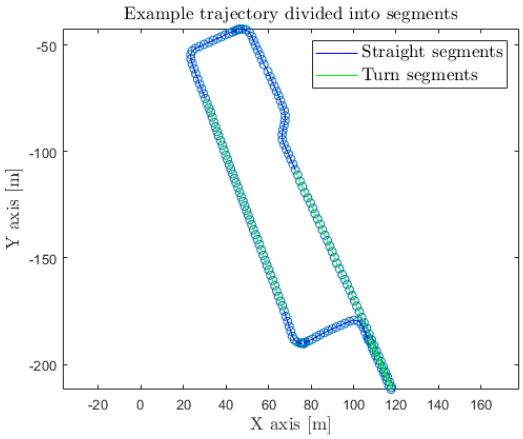
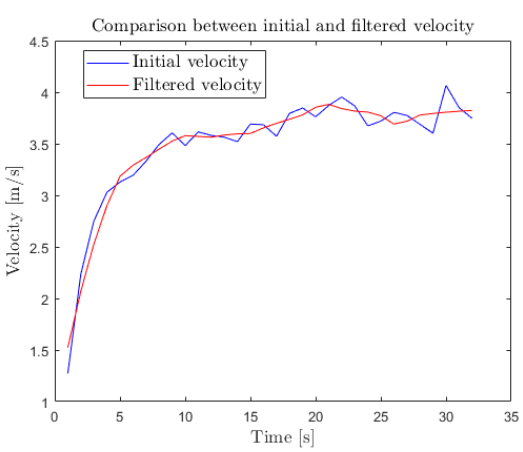
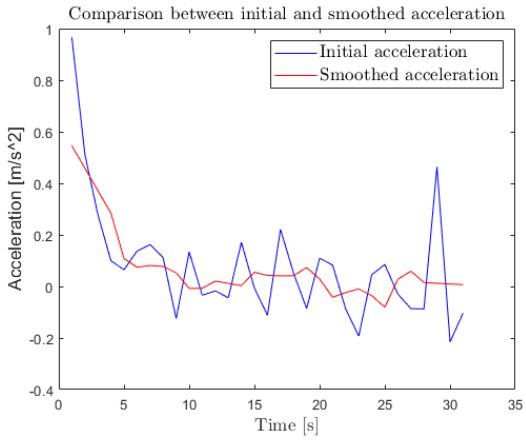


Figure 4-3: Example of division a single trajectory into segments: straight (green), turns (blue)



(a) Comparison between initial and filtered velocity



(b) Comparison between acceleration resulting from initial and filtered velocity

Figure 4-4: Comparison between initial and Savitzky-Golay filtered velocity and acceleration

with $\delta t = 1$s given by the sample rate of the sensors, 1 Hz. Then, since the sampling rate is relatively high compared to other experimental data on cyclists' trajectories [10], and this could lead to possible high error on the acceleration calculation, v_x and v_y were smoothed with Savitzky-Golay filter with order 3 and window length on third of the length of the segment, and v calculated as $v = \sqrt{v_x^2 + v_y^2}$ and the acceleration calculated as the numerical derivative of v as $a(t) = \frac{v(t+\delta t) - v(t)}{\delta t}$. The Figures 4-4a and 4-4b show the comparison between initial and smoothed velocity and the acceleration profile resulted by derivation of the initial and smoothed velocity respectively.
In order to calculate the distance of the cyclist from the lane edges, accurate position data of the line in the middle of the cycling path is needed. This data was collected by slowly walking through the middle of the cycling path with a GPS-equipped backpack. The position of road edges was collected in the same way. The resulting position data was defined as a polygon, since this format is useful for the algorithm to detect violations of the constraint defined by the road edges.

4-2 Feature definition

Once the empirical data have been collected and processed, the learning algorithm was developed. The first step is to define the reward function describing the cycling styles to be identified from data.
Choosing the features of the reward function is a domain-dependent problem, and has been handled in different ways in the literature. Considering the reward shape definition and consequent feature structure, defining reward function as the linear sum of weighted features can give insights on the real-life meaning, but can result in an inaccurate approximation of the real reward function behind humans decision. On the other hand, using a function approximator like neural network would improve the accuracy, but it is more difficult to connect it with human features.
The choice for the proposed algorithm is a model with a cost function defined as a weighted

sum of features, that are easily interpreted and thus suitable for characterizing individual cyclist's preferences.
The literature on Inverse Reinforcement Learning uses the concept of reward function instead of cost function, since it is easier to intuitively explain human preferences. By using a cost function instead, the whole framework remains consistent: instead of $\arg\max(R)$, $\arg\min(J)$ is used, with J as the cost function where $J = -R$.
In the field of Inverse Reinforcement Learning for traffic applications, the objective function expressions reported in the literature usually include terms related to travel efficiency, comfort and interaction with other vehicles. These terms have been defined quite consistently across the literature: the first one is a function of the difference between the velocity and the desired velocity, the comfort factor depends on the acceleration and the last one is usually related to some distance or speed difference with respect to the surrounding vehicles.
The mathematical expression of the feature functions are commonly:

- Quadratic function, e.g. $\Phi(f) = f^2$, in order to define a quadratic-programming problem
- Exponential function, e.g. $\Phi(f) = e^f$, usually used for dangerous states, such as approaching stops or other vehicles. This way, the feature value increases exponential when the agent gets closer to these states.
- Absolute value $\Phi(f) = |f|$

In this work, the feature functions have an absolute value form, which is a function structure widely used in the literature for describing reward or cost functions. Moreover, the feature functions are normalized with respect to each maximum allowed value. This ensures that the features are on the same range, and that the gradient descent optimization moves smoothly towards the minimum by updating the weights at a similar rate. Since features are normalized, the set of weights derived for each trajectory is a consistent information that can be used for comparison and statistical analysis.
Given each trajectory τ, from which the velocity $v(t)$, acceleration $a(t)$ are derived according to 4-1, the features proposed in this research are listed as:

- **Acceleration**. In the research works for driving style characterization, this term has been connected to comfortable driving. For cyclists, it is representative of the willingness of a cyclist to accelerate to arrive to the destination faster, to avoid a red light or to pass another cyclist.
  $\Phi_{\mathrm{a_{pos}}}(\tau, a_{\max}) = \frac{1}{t_{\max}} \sum_{t=0}^{t_f} \frac{|a(t)|}{a_{\max}}$ for $a(t) > 0$
- **Negative acceleration**. This term is connected to the behaviour of cyclist in situations where their velocity is smaller than the one of the cyclist ahead or when a cyclist is approaching a traffic light and can accelerate trying to avoid the red light or decelerate and then stop. It is also relevant when describing the cyclists' to the behaviour in turns.
  $\Phi_{\mathrm{a_{neg}}}(\tau, a_{\min}) = \frac{1}{t_{\max}} \sum_{t=0}^{t_f} \frac{|a(t)|}{a_{\min}}$ for $a(t) < 0$
- **Minimizing travel time**. Minimizing travel time is one of the common goals of many cyclists. However, given the same external conditions as rush, weather etc. some cyclist could give a higher value to this criterion with respect to others.
  $\Phi_t(\tau, t_{\max}) = \frac{t_f}{t_{\max}}$

- **Cycling in lower speed than the desired speed** Cycling with a speed different than the desired one can be frustrating. However, there can be persons that are comfortable with cycling at a slightly lower speed than the desired one, possibly in order to save energy.
  $\Phi_{v_{\text{lower}}}(\tau, v_{\text{des}}) = \frac{1}{t_{\max}} \sum_{t=0}^{t_f} \frac{|v(t)-v_{\text{des}}|}{v_{\text{des}}}$ for $v(t) < v_{\text{des}}$
- **Cycling in higher speed than the desired speed** As for the previous point, other persons could prefer to cycle with a velocity higher than desired one. At the same way, cycling at higher speed than the desired one could be more frustrating for some cyclists than others.
  $\Phi_{v_{\text{higher}}}(\tau, v_{\max}, v_{\text{des}}) = \frac{1}{t_{\max}} \sum_{t=0}^{t_f} \frac{|v(t)-v_{\text{des}}|}{v_{\max}-v_{\text{des}}}$ for $v(t) > v_{\text{des}}$
- **Distance from middle of the cycling path**. This feature encodes the concept of safety margins with respect to the road edges. The safest choice would be to cycle at the center of the lane, especially in curves.
  $\Phi_d(\tau, d_{\max}) = \frac{1}{t_{\max}} \sum_{t=0}^{t_f} \frac{|d(t)|}{d_{\max}}$ where $d(t)$ is the distance from middle of the cycling path at time t.

Having the features above, the objective function for each cyclist's trajectory τ is defined as follows:

$$J(\tau, C, v_{\text{des}}, w) = w_{\text{a}_{\text{pos}}}\Phi_{\text{a}_{\text{pos}}}(\tau, a_{\max}) + w_{\text{a}_{\text{neg}}}\Phi_{\text{a}_{\text{neg}}}(\tau, a_{\min}) + w_t\Phi_t(\tau, t_{\max}) + w_{v_{\text{higher}}}\Phi_{v_{\text{higher}}}(\tau, v_{\max}, v_{\text{des}}) + w_{v_{\text{lower}}}\Phi_{v_{\text{lower}}}(\tau, v_{\text{des}}) + w_d\Phi_d(\tau, d_{\max})$$

With acceleration a, vecity v, final time t_{f} and distance from middle of the lane d derived from the trajectory τ. C is the set of constraints given by the road constrains, namely the limits given by the lane and consequently the maximum distance $d_{\max}$the velocity constraints given by maximum velocity $v_{\max}$, acceleration constraints as the maximum acceleration $a_{\max}$ and the maximum travel time $t_{\max}$. These constraints do not depend on the cyclist, but on the segment type. The desired velocity is calculated for each cyclist as the mean velocity during the trajectories where the cyclist was asked to cycle normally and maintain his desired speed. For each segment type, $v_{\text{des}} = \frac{\sum_{t=1}^{T} v(t)}{T}$, with T duration. The set of weights, one for each feature, is the output of the algorithm proposed in Section 4-3-5.
All the features presented in this section are normalized, so their magnitude is less or equal to one. In order to reasonably trust the empirical feature values that are crucial in this method, the uncertainty on these quantities should be less than 10% of their maximum value. This estimation is rough because it is employed to verify that the theoretical range of accuracy fits the accuracy provided by the data collection equipment.
In the following part of the section, the uncertainty on each feature is calculated from its mathematical expression and estimations for the velocity, acceleration and maximum distance values. Then, assumed that the uncertainty should be at maximum 10% of the maximum feature value, the required accuracy on position, velocity and acceleration data is derived and the results compared with the accuracy of the employed sensors. Given 1 Hz sampling rate of the GPS and duration of the recorded path $t = t_{\max}$, the maximum number of collected data points is $n_{\max} = fs \cdot t_{\max} = t_{\max}$. In the feature calculation, $t_{\max}$ is canceled because every feature is divided by $t_{\max}$ as in the features' mathematical expressions indicated in 4-2, so it

will not be part of the uncertainty expression.

- $\mathbf{\Phi_{a_{pos}}}$
  With estimated maximum acceleration $a_{\max} = 2\frac{m}{s^2}$, the maximum uncertainty for this feature $\delta(\Phi_{a_{pos}})$ is calculated through the law of propagation of uncertainty [12]:
  $\delta(\Phi_{a_{pos}}) = \sqrt{\left(fs \cdot \frac{\delta^2(a)}{a^2_{\max}}\right)}$.
  Given the requirement of having $\delta(F_{a+}) \leq 0.5$, it results $\delta(a) \leq 0.2\frac{m}{s^2}$

- $\mathbf{\Phi_{a_{neg}}}$
  At the same way, the minimum negative acceleration is given by $|a_{\min}| = 2\frac{m}{s^2}$, so the same result follows the calculations.

- $\mathbf{\Phi_{v_{pos}}}$ and $\mathbf{\Phi_{v_{neg}}}$
  Since the desired cycling velocity is not a precise value but more of a range in which the cyclists is comfortable riding, it will be assumed to be without uncertainty for the scope of this section. As a consequence, the uncertainty of the velocity features follows:
  $\delta(\Phi_{v}) = \sqrt{\left(fs \cdot (\frac{\delta^2(v)}{(v_{\max} - v_{des})^2})\right)}$
  With estimated maximum velocity of $v_{\max} = 12\frac{m}{s}$ and estimated desired velocity $v_{des} = 4$, the uncertainty for the feature F_{v-} is more conservative and the uncertainty on the velocity results: $\delta(v) \leq 0.2\frac{m}{s}$

- $\mathbf{\Phi_{d}}$
  Both the center of the cycling path position and the cyclist's position suffer of uncertainty, so the uncertainty of the distance d is: $\delta d = 2\delta(s)$ Given a maximum distance $d_{\max} = 2m$, the uncertainty on position needs to be:
  $\delta(\Phi_{d}) = \sqrt{\left(fs \cdot \frac{\delta^2(d)}{d^2_{\max}}\right)}$
  Resulting in $\delta(s) \leq 0.2m$

Overall, the required accuracy on position is 0.1 m, which is consistent with the required accuracy of velocity and acceleration, which are obtained through differentiation of the position. The sensors employed for data collection can reach this accuracy on position, so the choice of sensors is consistent with the required accuracy on feature values.

4-3 Proposed algorithm

The main research questions arising during this project is if it is possible to characterize cyclist's style using Inverse Reinforcement Learning. In Alsaleh et al. [3] [21], Maximum Entropy Inverse Reinforcement Learning algorithm was applied to cyclist behaviour during specific short manoeuvres, namely overtaking and following. As a consequence, the state-action space is limited to the possible action and states during these cycling situations and

being restricted, it is possible to use the Maximum Entropy version of Inverse Reinforcement Learning. When longer paths are considered, the computational effort of this algorithm becomes too high. As a consequence, continuous-time versions of Inverse Reinforcement Learning as Inverse Optimal Control and Trajectory Sampling Inverse Reinforcement Learning were considered for this project, in line with the literature on Inverse Reinforcement Learning for traffic applications.
Another goal of this thesis project is to design an experimental setup in order to collect empirical trajectories. In order to evaluate the performance of a preference identification algorithm, there should be a noticeable preference difference in the gathered data. This issue represents a big challenge since collecting a big amount of data from different cyclists is time-consuming, requires participants availability, tools to gather data (e.g. sensors, cameras etc.). Moreover, there are not available public datasets for cyclists trajectories. For this reason, an experimental setup in which cyclists are instructed on which preferences to show was developed. This way, variability on cycling preferences is forced during the experiment and the algorithm is tested on its capabilty of learning the preferences related to the instructions given to participants. The details of the experiments were presented in Chapter 3.
From a conceptual perspective, this changes the premises of time-continuous Inverse Reinforcement Learning algorithm. The assumption is that trajectories are not sampled from a probability distribution, but the trajectory of each cyclist is optimal with respect to the reward function, which should be consistent with the instructed cycling preferences.
This makes the proposed approach close to Inverse Optimal Control, with the modification of using one demonstration and extracting the weights corresponding to the behaviour shown in this demonstration, since it is assumed to be representative of the cycling preferences.
A high-level representation of the algorithm is shown in the Figure 4-5. As can be seen in the figure, in step 1, the empirical feature vector is calculated (bottom-left of the figure) for each demonstrated cycling trajectory. In step 2, given the initial position and velocity and the constraints, the trajectory that is optimal with respect to the reward function at iteration i is calculated and the corresponding feature values are derived in step 3 (top of the figure). In step 4, the difference between empirical and optimal feature values gives the gradient used for weights update (bottom-right of the figure), then the optimization process restart from step 1. The process is repeated iteratively until convergence or trigger of the stop conditions.
In the remainder of this subsection, deeper analysis of the proposed algorithm, including learning rate, optimization vector, end conditions and constraints choice are presented. The pseudo-code is reported in subsection 4-3-5.

4-3-1 Learning rate

Another difference with respect to the standard Inverse Optimal Control algorithm is the use of learning rate schedule method, a common technique in optimization problems. The learning rate is initialized and used to calculate the new weight vector at the first iteration. If the weights derived with this learning rate lead to an optimal trajectory with higher error with respect to the error of the optimal trajectory at the previous iteration, it means that the optimization process is escaping from the minimum because the learning rate is too high. In this case, the learning rate is decreased by multiplying it for a decay factor $0 < \gamma < 1$. If the error is less than the previous iteration, the learning rate and weight values are kept and the optimization process continues. The considered error is the norm of the difference between

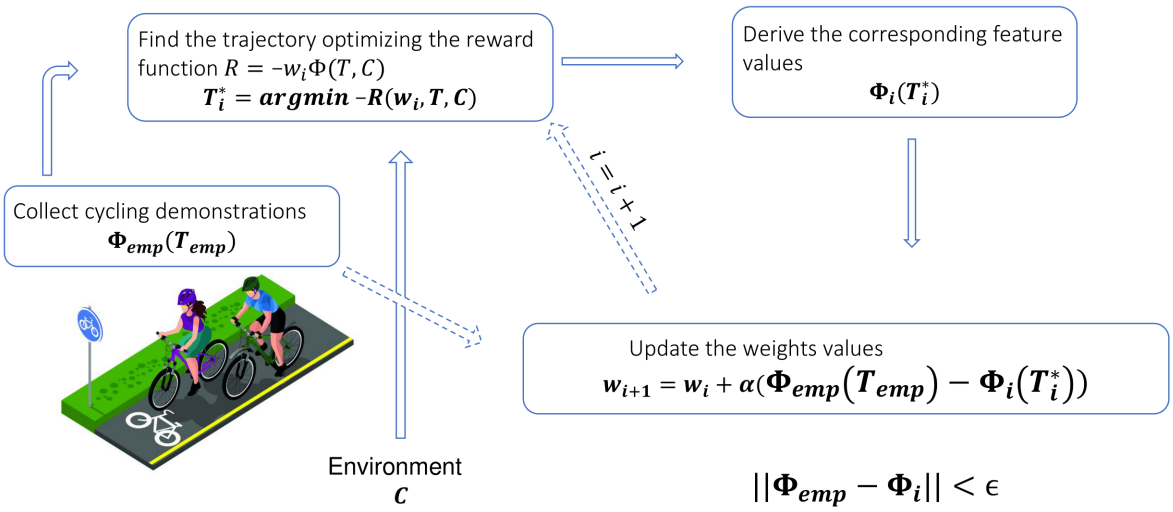


Figure 4-5: Proposed Inverse Reinforcement Learning approach for cyclist preferences extraction. On the left, the empirical cycling trajectories are collected and the corresponding empirical features derived. At each iteration, the trajectory optimizing the reward function is determined (upper-left corner) and the corresponding feature values calculated (upper-right corner). The difference between empirical and generated feature values is used as a gradient to update the weights values defining the reward function to be optimized at the next iteration.

simulated and empirical features. In detail, the process at iteration i:
1. Compute the error between generated features and empirical features: $e_i = ||\widetilde{\Phi} - \Phi_i(\tau_i^*)||$
2. If $e_i \geq e_{i-1}$
   a. The optimization is escaping from the minimum because the learning rate is too big: $\alpha = \alpha\gamma$
   b. Update the weight value with the new learning rate and go back to the beginning of iteration i:
   $w_i = w_{i-1} - \nabla\alpha$
   else:
   a. The weight vector can be updated with the previous learning rate:
   $w_{i+1} = w_i - \nabla\alpha$
   b. The algorithm can continue with iteration $i + 1$

4-3-2 End conditions

In order to stop the optimization process, a set of end-conditions are checked. These end conditions are:

- The error between generate features and empirical features is less than a threshold: in this case, the generated trajectory is similar enough to the empirical one and the optimization can stop.
  COND_1: $e_i \leq \epsilon_e$
- The learning rate is less than a threshold: the learning rate is so small that the updated weight values are almost equal to previous ones.
  COND_2: $\alpha \leq \epsilon_\alpha$
- The generated trajectories do not lead to a decrease of the error for a consecutive number of times: this means that there is no improvement by updating the weight value so probably the optimization stopped in a local minimum.
  COND_3: $n_e \geq N_e$

4-3-3 Constraints

Regarding the constraints, nonlinear constraints act as upper and lower bounds for velocity and acceleration, since a cyclist can only reach a maximum speed and acceleration. An upper and lower boundary on the final time $t_{final} = t_k$ ensure that the arrival time is in a reasonable range considering the length of the path. All the position points should lie inside the limits of the roads, which is given as a closed polygon.
The constraints C read:
$0 \leq v(t) \leq v_{\max}$
$a_{\min} \leq a(t) \leq a_{\max}$
$t_{\min} \leq t_{final} \leq t_{\max}$
$x(t), y(t) \in P$
Where P is the polygon defined by the road edges.

4-3-4 Optimization vector

In order to reduce computational time, the spline-segmentation method proposed by Kuderer et al [15] was employed. The trajectory is defined as a two-dimentional fourth-order piece-wise spline.
A trajectory with total time duration t_f is divided into N_{tot} segments, each one described by:

- the time interval $[t_k, t_{k+\delta t}]$, with k denoting each interval, Δt the duration of each time interval and $N_{tot}\Delta t = t_f$

- a polynomial function of the trajectory on x-axis over time:
  $x(t) = s_x^k(t)$, $t \in [t_k, t_{k+\Delta t}]$
  with:
  $s_k^x(t) = a_0^k + a_1^k t + a_2^k t^2 + a_3^k t^3 + a_4^k t^4$,
  $x(t)$ the position on the x-axis over time and $s_k^x(t)$ the fourth-order spline describing the position x-axis over the time interval $[t_k, t_{k+\nabla t}]$.

- a polynomial function of the trajectory on y-axis over time:
  $y(t) = s_y^k(t)$, $t \in [t_k, t_{k+\Delta t}]$
  with:
  $s_k^y(t) = b_0^k + b_1^k t + b_2^k t^2 + b_3^k t^3 + b_4^k t^4$,
  $y(t)$ the position on the y-axis over time and $s_k^y(t)$ the fourth-order spline describing the position y-axis over the time interval $[t_k, t_{k+\nabla t}]$.

The knots of the spline are the points where the segments are joined. Since the segments share the last point with the first point of the next segment, the control points of a segment k are expressed as the first point of the segment only, and the last point of the last segment:

$$c_k = [s_x^k(1), s_y^k(1), t_k] \cup [x(t_f), y(t_f), t_f] \qquad k \in N_{tot} \tag{4-2}$$

The optimization process optimizes these control points only, which are a subset of the total points of the spline and are defined by their position and time. For each dimension, a set of optimal control points is optimized. Then, the continuous function of position over time for both axis is derived as a piece-wise fourth-order spline $x(t)$ and $y(t)$ respectively. The union of $x(t)$ and $y(t)$ gives the two-dimensional trajectory $\tau(t) = (x(t), y(t))$. The velocity for x and y axis is calculated as the analytical derivative of the functions of position over time $x(t)$ and $y(t)$, resulting in $\dot{x}(t)$ and $\dot{y}(t)$ respectively, which are evaluated in the points given by the time vector t_k. The velocity is then given by $v(t) = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}$. The acceleration is obtained in a similar way, by finding the analytical second derivative of the functions $x(t)$ and $y(t)$, resulting in $\ddot{x}(t)$ and $\ddot{y}(t)$, evaluated in the time points given by t_k, and then calculating $a(t) = \sqrt{\ddot{x}(t)^2 + \ddot{y}(t)^2}$. Given the trajectory and its derivatives, the feature values are calculated.
The final optimization vector is then defined as the the set of control points for each segment

of the whole trajectory, denoted as:

$$v = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_k \\ \dots \\ c_{N_{tot}} \end{bmatrix} = [x_c, y_c, t_c]$$

with c_k a vector containing the x-position, y-position and time for each control point, and x_c the vector of the positions of all the control points on the x-axis, y_c the vector of the positions of all the control points on the y-axis and t_c the time vector corresponding to the control points defined by x_c and y_c.
At each iteration, given a set of weight values w the optimization problem is given by:

$$\begin{aligned} \min_{\tau} \ & -R(\tau) = \min_{\tau} w\Phi(\tau) = \min_{v} w\Phi(\tau(v)) = \min_{x_c, y_c, t_c} w\Phi(\tau(x_c, y_c, t_c)) \\ \text{s.t.} \quad & 0 \leq t_c \leq t_{\max} \\ & g(\tau) \leq 0 \end{aligned} \tag{4-3}$$

with $g(\tau)$ nonlinear constraint on the trajectory, velocity and acceleration:

$$\tau(x_c, y_c, t_c) \in P \tag{4-4}$$
$$0 \leq \dot{\tau}(x_c, y_c, t_c) \leq v_{\max} \tag{4-5}$$
$$a_{\min} \leq \ddot{\tau}(x_c, y_c, t_c) \leq a_{\max} \tag{4-6}$$

4-3-5 Pseudo-code

The pseudo-code of the proposed algorithm follows:

For each empirical trajectory τ_j^s of person j and cycling style s

a. The empirical feature vector is computed from the demonstration $\widetilde{\Phi}$
b. The weight vector w is initialized to ones, the parameters initial learning rate α_0, number of spline control points N, decay γ factor are given, stop conditions set to FALSE
c. While **stop conditions** are all **FALSE** do:
   c.1. The initial and final state of τ are fixed, and the optimization vector is optimized with respect to the reward function $\boldsymbol{R_i = -w_i^T \Phi_i}$, given the constraints.
   The resulting optimized control points generate an optimal trajectory: $\boldsymbol{\tau_i^* = \textbf{argmin } (w_i^T \Phi_i)}$
   c.2. Compute the generated feature values by evaluating the feature function for the optimized trajectory: $\boldsymbol{\Phi_i(\tau_i^*)}$
   c.3. Compute the error between generated features and empirical features: $\boldsymbol{e_i = ||\widetilde{\Phi} - \Phi_i(\tau_i^*)||}$

c.4. Compute the gradient:
$\boldsymbol{\nabla = \widetilde{\Phi} - \Phi_i(\tau_i^*)}$
c.5. if $\boldsymbol{e_i \geq e_{i-1}}$
c.5.Optimization escaping from the minimum because the learning rate is too big:
$\boldsymbol{\alpha = \alpha\gamma}$
c.5.b. Update the weight value with the new learning rate:
$\boldsymbol{w_i = w_{i-1} - \nabla\alpha}$
c.5.c. Go back to c.1
c.6. else:
c.6.1. The weight vector can be updated with the previous learning rate:
$\boldsymbol{w_{i+1} = w_i - \nabla\alpha}$
c.6.2. : The algorithm can continue with the following iteration:
$\boldsymbol{i = i + 1}$
c.7.: Check stop conditions, if one is **TRUE** break and save the weights of the last iteration
d.: Save the extracted weights in the weight buffer:
$\boldsymbol{B \leftarrow w_j^s \leftarrow w}$
end

Chapter 5

Results

The first part of the chapter will focus on a convergence study of the optimization algorithm. In Section 5-1, the convergence of the learned cycling style towards the empirical trajectories is examined. Then, the weights extracted from the trajectories will be analyzed in terms of difference between cycling styles. Section 5-3-1 will explore the statistical differences between group of weights related to different cycling styles. A series of statistical tests will be performed in order to analyze the connection between weight distributions and cycling styles.
As anticipated in Section 4-1, the trajectory data related to straight road segments and turns will be considered separately during the optimization process and the subsequent statistical analysis of the weights. This follows from the fact that the behaviour of cyclists during turns is constrained by stability and safety reasons: in particular, cyclists decelerate during turns, while they do not do so in a straight segment. This choice builds on the literature on IRL for driving style recognition: in [11] path segmentation is used and in [25], straight and turning road segments are distinguished.

5-1 Convergence study

Given a single demonstrated trajectory, for each iteration the initial position (x_0, y_0), the velocity v_0 and final position (x_f, y_f) are fixed. These conditions are boundary conditions of a spline trajectory τ defined as in equation 2-7, where the remaining spline control points (x_i, y_i, t_i) are the optimization variables with respect to the learned reward function $R = -w\Phi(\tau)$. At each iteration, the learned weights result in a reward function that generates a trajectory which gets closer and closer to the empirical one. The error term is the norm between features resulting from the optimized trajectory and empirical ones. Since the features are considered a powerful way to encode the characteristics of the trajectory, having a decreasing feature error results in an increased similarity of trajectories. An example of spline trajectories generated at each iteration converging towards the empirical trajectory is shown in Figure 5-1. The circle markers are the control points of the splines, which are the optimization variables optimized at each iteration, from which the trajectory is derived

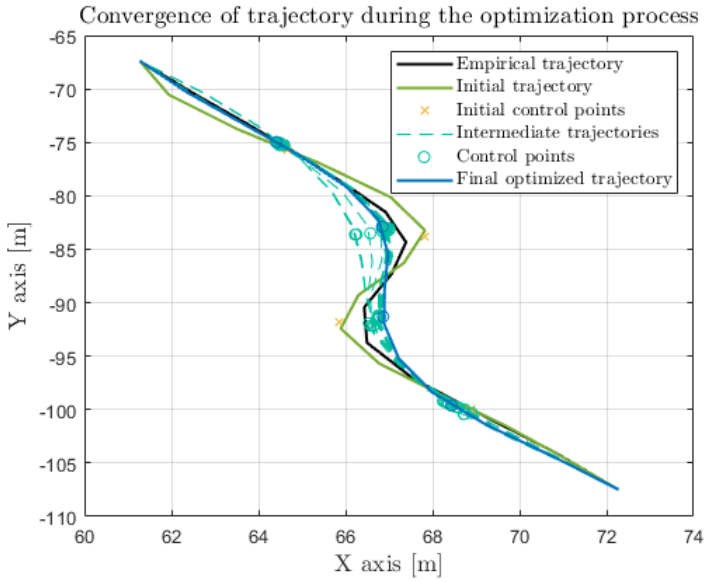
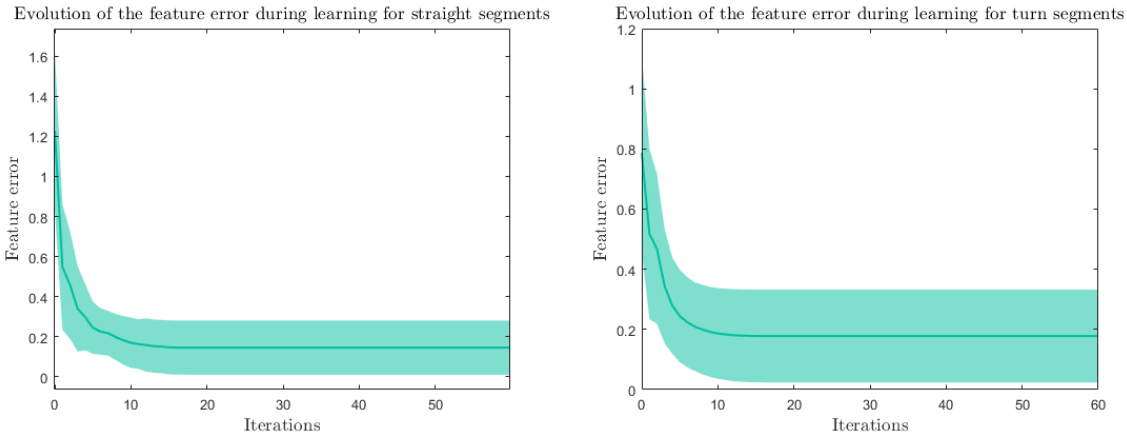


Figure 5-1: Example of spline trajectories generated by the reward functions whose weights are optimized during the learning process. The markers represent the control points of the spline trajectories. They are the optimization variables.

according to Section 4-3-4.
The evolution of the feature error averaged for all trajectories during learning is shown in Figure 5-2, 5-2b for turns and 5-2a for straight segments. The shaded bands represent the standard deviation of the feature error evolution for all the trajectories, and the solid line represents the mean feature error at each iteration. It can be noticed that the learning process rapidly converges towards a steady small error value, on average 0.1779 for turns and 0.1463 for straight segments, usually after 20 iterations. The mean reduction from the initial feature error to the final one is 87.23% for straight segments and 74.55 % for turns. This result is consistent with the literature on IRL in traffic domain, especially for continuous-time approximations [15]. The limitation of convergence using those methods could be due to the assumption of optimality of demonstrations, which also the proposed algorithm makes, or to approximation errors caused by trajectory sampling in trajectory-sampling based methods. For each trajectory, the convergence of the algorithm is evaluated in terms of the evolution of the norm of the difference between the empirical feature values and the expected feature values during learning. A trajectory is intended as a function position in time, and the velocity is considered an important information to be analyzed. Indeed, the position itself is only related to the safety margins with regard to distance from the middle of the cycling road, while the velocity function gives information about the tendency to keep the desired velocity and it is related to the total travel time, another feature of the proposed reward function.
Figures 5-3 and 5-4 refer to the best and the worst straight trajectories in terms of feature error. The same applies to turns in figures 5-5 and 5-6. For all these plots, on the left, the convergence of the generated velocity profiles towards the empirical velocity is shown. Here, the dashed line show the velocity resulting from intermediate reward functions during the learning process. On the right, the 2-dimensional plot of the trajectory is reported. It can be noticed that for trajectories showing poor convergence, the algorithm rapidly converges to a

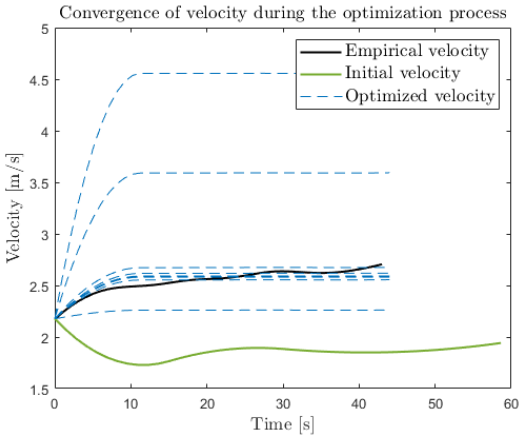


(a) Evolution of feature error for straight trajectories **(b)** Evolution of feature error for turn trajectories

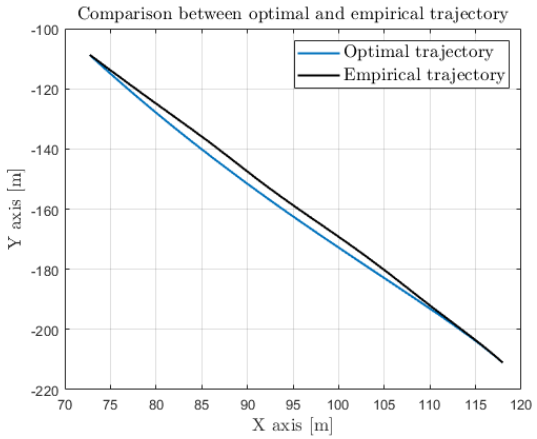
Figure 5-2: Evolution of feature error during learning

sub-optimal reward and cannot further improve the result. In both cases, the main difference is that the velocity profiles are different with respect to the empirical ones. The learning rate schedule proposed in 4-3-1 takes care of possible high initial learning rates, so the problem probably arises because the proposed reward function structure or spline parametrization of the trajectory cannot describe accurately the behaviour shown in these demonstrations. For straight segments, the algorithm tends to generate trajectories that reach a steady velocity such that the features error is minimized. In the case of the worst straight trajectories, the shape of the empirical velocity profile differs significantly with respect to a speed profile converging to a constant value, and the algorithm cannot simulate a trajectory which reproduces a similar velocity profile. In the worst turn trajectory, the cyclists abruptly decelerates to stop and then abruptly accelerates again: this behaviour may be too complex to be represented by a linear reward function and a simplified spline trajectory model. Regarding the best results, it can be seen that the algorithm almost perfectly imitates the empirical trajectories. For most of the demonstrations, the proposed algorithm can learn a reward function that generates a smooth trajectory that imitates the cyclist behaviour reasonably well.

In order to have a high-level picture of the cycling styles learned by IRL, a comparison of averaged empirical and optimized speed profiles for different cycling styles is shown in Figures 5-7,5-8,5-9 and 5-10. The shaded bands represent the standard deviation of the speed profiles of all the trajectories in that segment, and the solid line represents the mean velocity. To help visualization of the different speed profiles, different road segments have been considered separately. The first Figures 5-7 present the empirical and optimized speed profiles of turns starting right after a stop, while the second set of Figures 5-8 shows the speed profiles of curves in the middle of the path, not approaching a stop segment nor starting from a stop. The third set of Figures 5-9 shows the speed profiles of turn segments approaching a stop. Finally, the last set of Figures 5-10 show the speed profile of empirical and generated trajectories during straight segments. The plot of the velocity profiles for the three cycling styles' trajectories show relevant differences among the three. In straight segments, aggressive cyclists ride with a higher speed accelerating during the path compared to normal cyclists who tend to keep their desired trajectory throughout the road segment. The pattern is similar for the simulated trajectories' speed profiles, implying that the algorithm could learn high-level speed profile

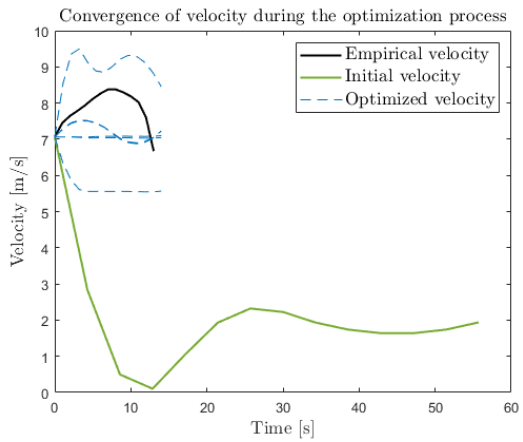


(a) Convergence of velocity

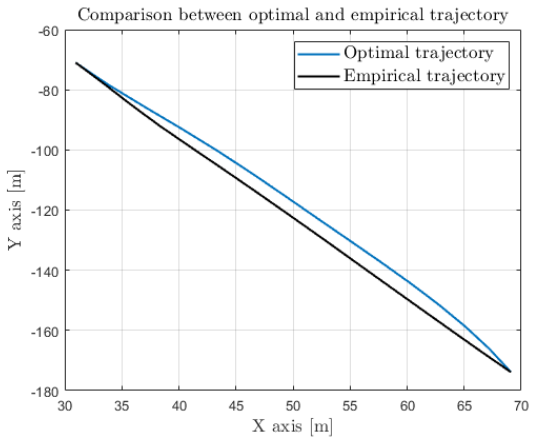


(b) Comparison between the generated and empirical straight trajectories

Figure 5-3: Best generated straight trajectory

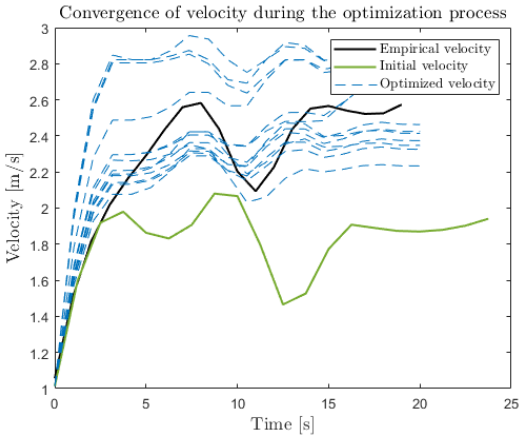


(a) Convergence of velocity

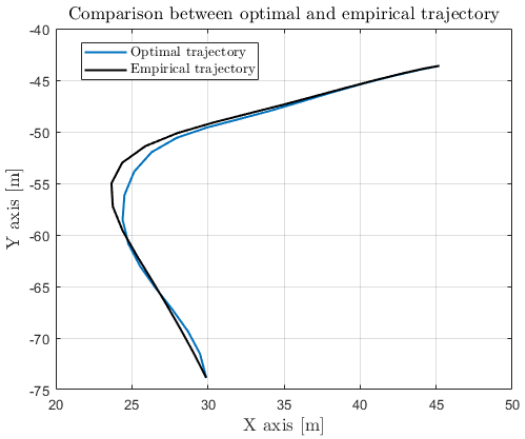


(b) Comparison between the generated and empirical straight trajectories

Figure 5-4: Worst generated straight trajectory

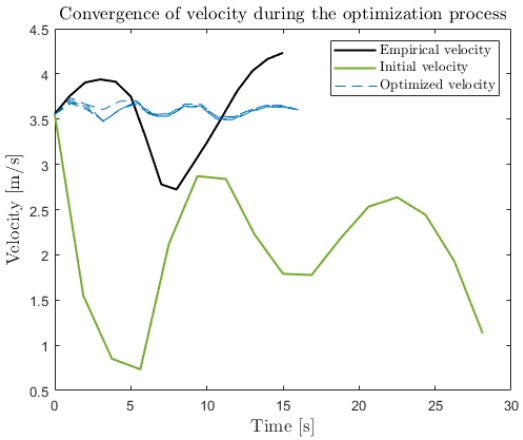


(a) Convergence of velocity

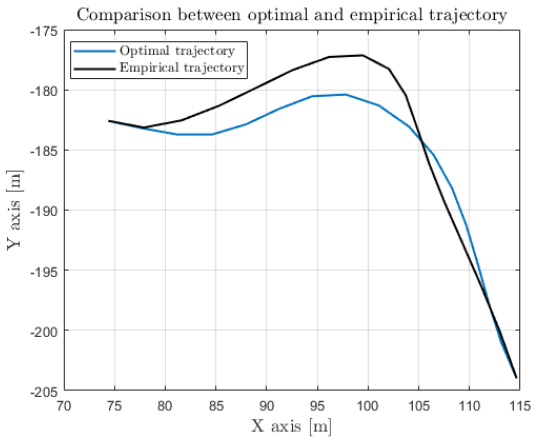


(b) Comparison between the generated and empirical turn trajectories

Figure 5-5: Best generated turn trajectory

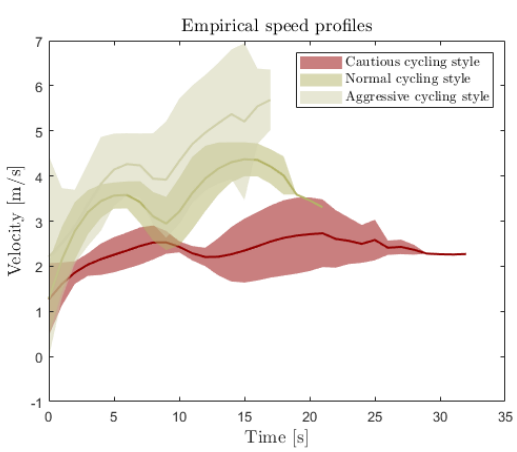


(a) Convergence of velocity

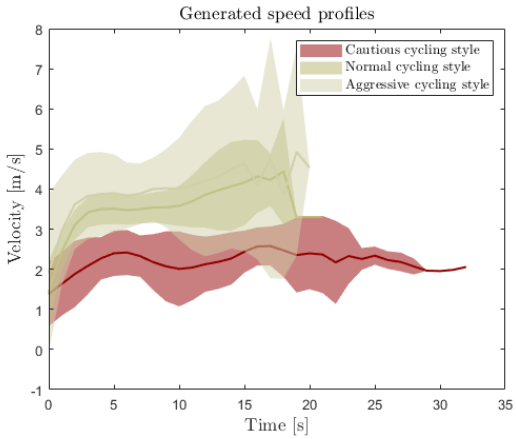


(b) Comparison between the generated and empirical turn trajectories

Figure 5-6: Worst generated turn trajectory



(a) Empirical speed profiles of curves starting from a stop



(b) Optimized speed profiles of curves starting from a stop

Figure 5-7: Comparison between empirical and optimized speed profiles for turn segments starting from a stop

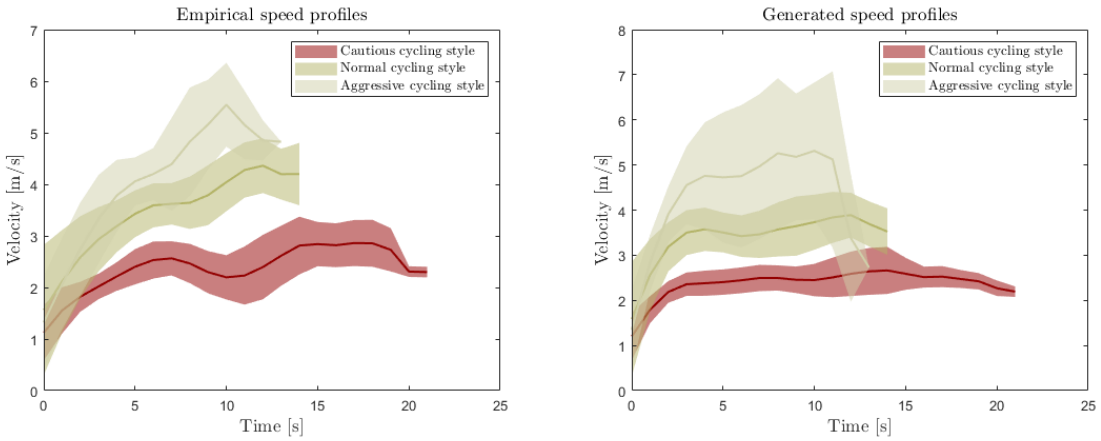
differences between cycling styles. However, the final increase of velocity during straight segments for aggressive cyclists is not reproduced by the generated speed profile. Different turning road segments were taken into consideration: a standard turn (no stops before or right after it), a turn immediately after a long straight segment and a turn immediately after the traffic light. As it can be noticed from the images, the speed profile of these three considered types of curves is different: for a turn right after a stop, aggressive cyclists tend to rapidly accelerate to a high but stable velocity, while normal cyclists slowly accelerate until the desired speed and cautious riders maintain a low, constant speed. At the same way, approaching a stop, aggressive riders abruptly decelerate while normal and cautious ones keep a stable, lower velocity. These characteristics are reproduced by the speed profiles generated by the optimal trajectories. However, it can be noticed that the speed profiles of normal cycling styles and aggressive cycling styles overlap significantly.

5-2 Simulation study

In this section, an evaluation of the learned cycling style is performed. The human likeness (HL) is a metric for measuring the closeness of a model to human demonstrations commonly used in the literature on driving style identification [13]. For trajectories, higher human likeness means better model accuracy, and can be described in terms of the mean error between empirical and generated trajectory. The lower the error, more accurate is the reward function. The HL error is expressed as:

$$HL = \frac{||\tau_{\text{emp}} - \tau_{gen}||}{n_t}, \tag{5-1}$$

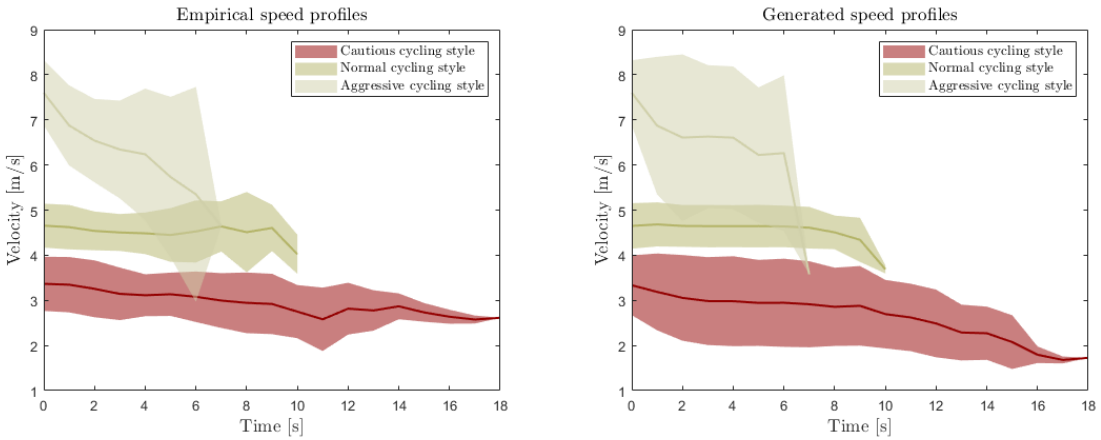
where τ_{emp} denotes the empirical trajectory and τ_{gen} denotes the generated trajectory, both sampled at the same timestamps and divided by the total number of timestamps n_t.
The trajectory dataset was split according to the cycling styles and then into training and



(a) Empirical speed profiles for the S-shaped turn in the middle of the path

(b) Optimized speed profiles for the S-shaped turn in the middle of the path

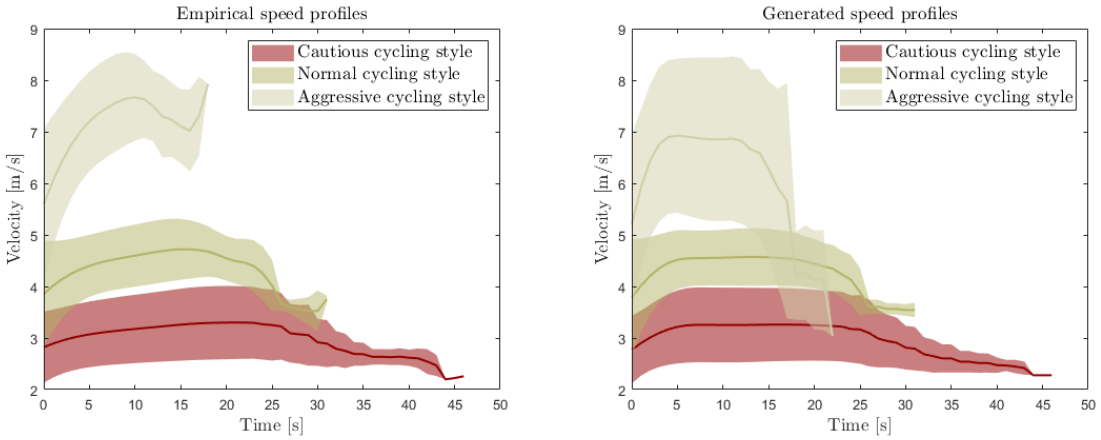
Figure 5-8: Comparison between empirical and optimized speed profiles for the S-shaped turn in the middle of the path



(a) Empirical speed profiles of curves approaching a stop

(b) Optimized speed profiles of curves approaching a stop

Figure 5-9: Comparison between empirical and optimized speed profiles for turn segments approaching a stop



(a) Empirical speed profiles of straight segments **(b)** Optimized speed profiles of straight segments

Figure 5-10: Comparison between empirical and optimized speed profiles of straight segments

test sets, resulting in a training and a test trajectories set per cycling style. The optimal weights for each trajectory were grouped into training and test sets for each cycling style, consistently with the trajectory sets. In other words, for a specific trajectory in one of the training trajectory set, the extracted weights are in the corresponding training weights set.
Then, the average weights for each weights training set are calculated and denoted as w_{train_1} for cautious cycling style, w_{train_2} for normal cycling style and w_{train_3} for aggressive cycling style. For each test trajectory, fixed the initial position and velocity and final position, each of the aforementioned weights are used to generate a trajectory.
The empirical test trajectory and the simulated ones are compared in terms of human likeness error HL. During the simulation study, the human likeness of trajectories generated by weights of the same cycling style of the test trajectory is compared to the ones generated by the other weights. Given a test trajectory τ_{test_k} with cycling style k (where $k = 1$ cautious, $k = 2$ normal and $k = 3$ aggressive), w_{train_1}, w_{train_2} and w_{train_3} are used to generate trajectories τ_{sim_1}, τ_{sim_2} and τ_{sim_3} respectively.
The similarity between the generated trajectories and τ_{test_k} is inversely proportional to the human likeness error given in equation 5-1. The human likeness error between the test trajectory τ_{test_k} and the trajectory generated by the corresponding cycling style training weights set are denoted as e_{k_1}, e_{k_2} and e_{k_3}. If the algorithm learned how to generate trajectories with cycling style k, it should generate trajectories that are similar to the test ones using weights trained on trajectories with cycling style k. As a consequence, given a test trajectory, the error resulting from a trajectory generated by the weights of the training set corresponding to the cycling style of the test trajectory should be lower than the other two.
The errors resulting from the simulation study on the test trajectories is then averaged according to the cycling style of the training set and the cycling style of the test set, resulting in an average error between the empirical test trajectories and the trajectories generated with each training set. The tables 5-1 and 5-2 reports the mean errors for each training and test set for straight segments and turns respectively. The trajectories are re-sampled every 0.1 seconds.

The results of the simulation analysis show that using the reward function trained on trajecto-

Table 5-1: Average trajectory error [m] between test straight trajectories and trajectories simulated with different cycling styles' reward functions

Rraining cycling style	Test - Cautious	Test - Normal	Test - Aggressive
w_{train_1}	0.4023	0.2748	1.3050
w_{train_2}	0.8407	0.0693	1.2108
w_{train_3}	1.1541	0.5729	0.7963

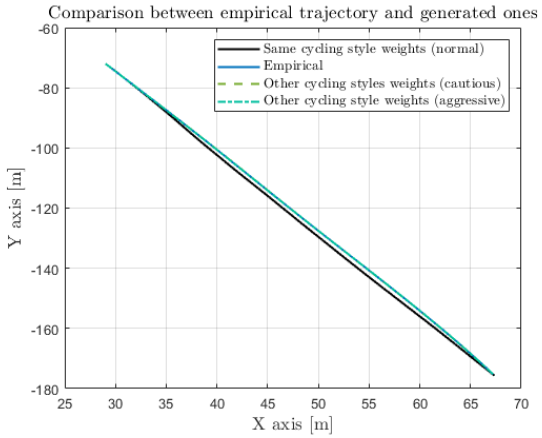
Table 5-2: Average trajectory error [m] between test turn trajectories and trajectories simulated with different cycling styles' reward functions

training cycling style	Test - Cautious	Test - Normal	Test - Aggressive
w_{train_1}	0.4963	0.6129	0.6753
w_{train_2}	0.5486	0.1819	0.2864
w_{train_3}	0.5590	0.2060	0.2609

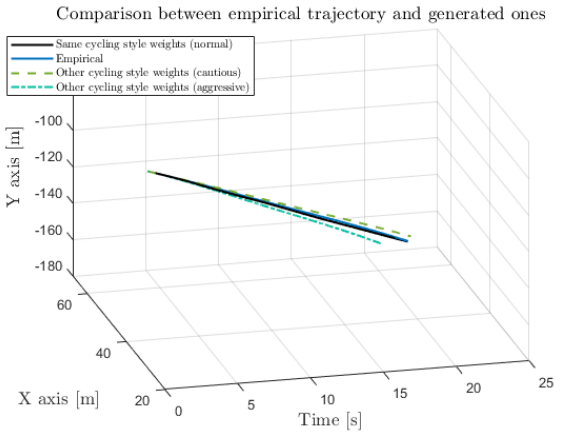
ries of the same cycling style results in a lower HL error, for both straight and turn segments. The improvement with respect to trajectories generated with other cycling styles' weights is particularly noticeable when cycling styles with clear differences as considered, as aggressive and cautious. This is reasonable since the resulting reward function reflects the difference of the demonstrated cycling behaviour, which is very different for cautious and aggressive cyclists. For straight segments, the improvement is particularly noticeable with respect to all the cycling styles. For turns, normal and aggressive cycling styles' weights generate trajectories with similar errors, while the difference is stronger when cautious cycling style weights are considered. For straight segments, the average improvement is 58.86 %, while for turns the average improvement is 28.85%, but reduces to 10.31% when only normal and aggressive cycling styles are considered. Figures 5-11,5-12 and 5-13,5-14 show a comparison between the test trajectory and the trajectories generated by the reward function trained on trajectories with the same cycling style as the test one and the ones trained on a different cycling style trajectory set, leading to the best and worst results in terms of HL error for straight segments and turns respectively. On the right, a 3-dimensional plot of the trajectory as a function of time is reported, with time on the x-axis, x-position on the y-axis and y-position on the z-axis. The plots show that even if the improvement is not noticeable, the reward functions trained on the same cycling style set still produce a smooth trajectory, similar to the test one.

5-3 Weight analysis

This section investigates the distributions of weights extracted from trajectories run with different cycling styles. For each trajectory, a set of six weights is extracted, one for each feature. The whole set of all the trajectories can be grouped according to two factors: the participant who run the trajectories and the cycling style instructions related to a specific trajectory. Assuming that the participants behave consistently with respect to instructed cycling preferences, this section will focus on the weight distributions resulting from the trajectories grouped according to the cycling style.

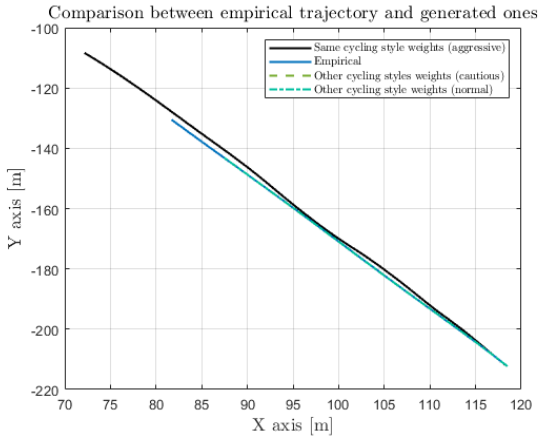


(a) Comparison of test and simulated trajectories

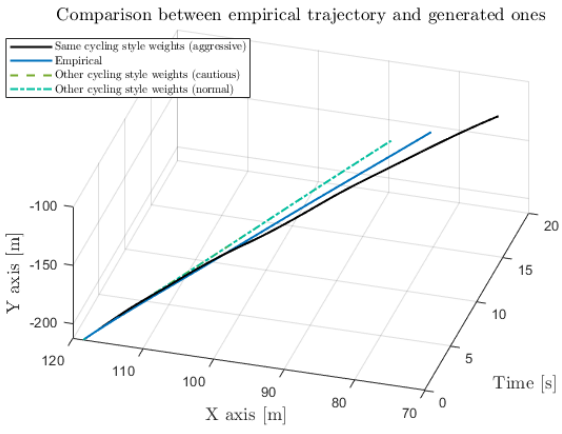


(b) Comparison of test and simulated trajectories as function of time

Figure 5-11: Simulation leading to the best results for the straight trajectory generated by the weights with same cycling style as the test one

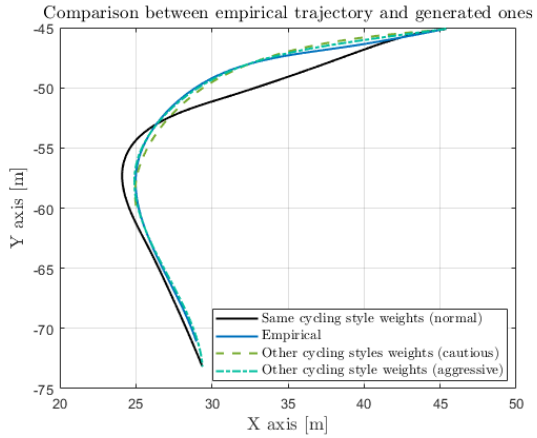


(a) Comparison of test and simulated trajectories

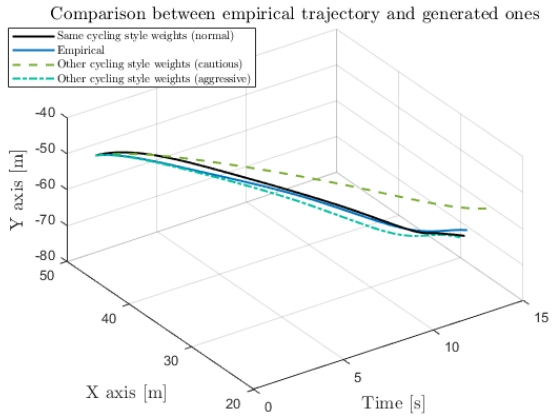


(b) Comparison of test and simulated trajectories as function of time

Figure 5-12: Simulation leading to the worst results for the straight trajectory generated by the weights with same cycling style as the test one

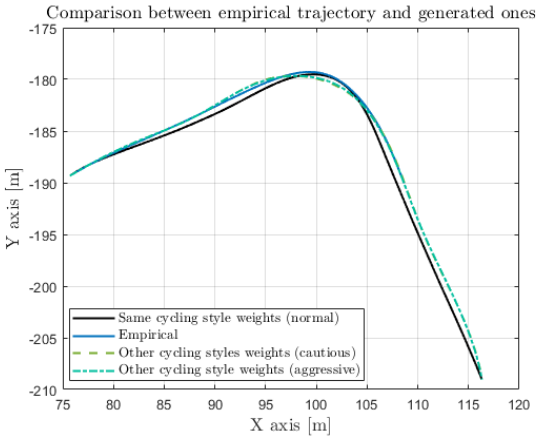


(a) Comparison of test and simulated trajectories

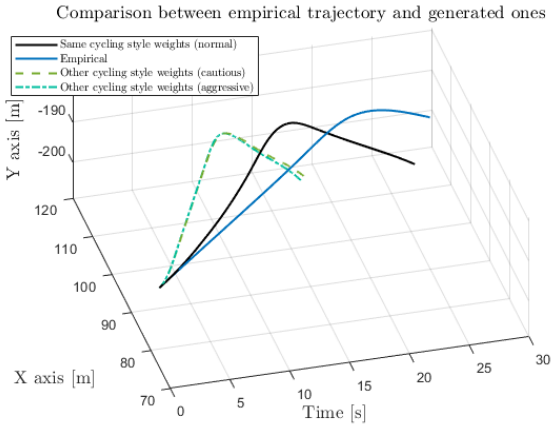


(b) Comparison of test and simulated trajectories as function of time

Figure 5-13: Simulation leading to the best results for the turn trajectory generated by the weights with same cycling style as the test one



(a) Comparison of test and simulated trajectories



(b) Comparison of test and simulated trajectories as function of time

Figure 5-14: Simulation leading to the worst results for the turn trajectory generated by the weights with same cycling style as the test one

Table 5-3: P-values resulting from Shapiro-Wilk test on normality for different features' weights of each cycling style extracted from straight trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
Cautious cycling style	$1.72\ 10^{-6}$	1.7410^{-8}	0.0036	$3.70\ 10^{-11}$	0.00012	0.00014
Normal cycling style	$1.07\ 10^{-6}$	$2.17\ 10^{-8}$	$6.37\ 10^{-9}$	$1.09\ 10^{-7}$	$1.39\ 10^{-6}$	$6.75\ 10^{-6}$
Aggressive cycling style	0.0009	0.054	0.192	0.00011	$1.59\ 10^{-10}$	0.00042

5-3-1 Statistical analysis

The distributions of the weight related to each feature for each cycling style will be compared. In this section, each feature is considered separately in order to verify the statistical difference between weights for each feature and consequently understand which features are more important in the description of cycling styles.
Statistical procedures determine the response of a variable in different population groups. For each feature, the distribution of weights for each cycling style can be formalized as a population group, and the factor whose effect is to be analyzed is the cycling style. Each group is defined as: w_k^j, with $k \in \{\text{Style 1, Style 2, Style 3}\}$ and $j \in F$, with F the set of the features $F = \{$Positive acceleration, Negative acceleration, Speed higher than desired one, Speed lower than the desired one, Time, Distance from the middle of the cycling path$\}$.
One of the most common statistical tests is ANOVA. ANOVA is a method that analyzes the difference between means of two or more population groups and it is based on the law of the total variance, which divides observed variance in a given variable into components due to different causes of variation: the variation between groups, defined as the variation of group means from the overall mean, and the variation within group, which is the variation of the values of each group from the estimate of their group mean. Then, ANOVA compares these two components: if the ratio between between-group and within group variability is higher than a threshold determined using the F-test, the group means are statistically different from each other. For each feature j, the standard ANOVA models the data (the weights) as a linear model:

$$y_{nk} = \mu_k + \epsilon_{nk} \tag{5-2}$$

where y_{jk} is an observation, with n observation number and k population group (in our case {Style 1, Style 2, Style 3}), μ_k the mean of the group and ϵ_{nk} the error associated to the y_{nk} data value. The main assumptions needed to represent the data with the linear model of equation 5-2 are the independence and normality of observed data and homogeneity of variance. The normality assumption were tested on the data by means of the Shapiro-Wilk test, reported to be one of the most powerful normality tests [23], with null hypothesis that the data comes from a normal distribution with an unknown mean and variance. The test rejects the null hypothesis at the 5% significance level, when the p-value is less than 0.05. The table with the statistics for weights for different features for straight and turn segments is reported in tables 5-3-1 and 5-4. The null hypothesis was rejected for almost every group of feature weights, so it can be concluded that the data do not follow a normal distribution. As a consequence, a non-parametric version of ANOVA, the Kruskar-Wallis test, was used on the extracted weight distributions.
The Kruskar-Wallis test is based on an analysis of variance using the ranks of data values rather than data themselves. The data are ranked from 1 for the lowest value of the data for a group to N for the highest value. Converting data to rank is necessary to avoid the need

Table 5-4: P-values resulting from Shapiro-Wilk test on normality for different features' weights of each cycling style extracted from turn trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
Cautious cycling style	0.0010	0.00023	$9.98\ 10^{-7}$	$5.65\ 10^{-11}$	$2.57\ 10^{-6}$	0.0019
Normal cycling style	0.00062	0.00096	$5.49\ 10^{-7}$	$1.04\ 10^{-06}$	0.00023	0.0126
Aggressive cycling style	0.0078	$5.4\ 10^{-7}$	$5.51\ 10^{-8}$	0.0019	$3.32\ 10^{-5}$	0.00017

Table 5-5: P-value resulting from Kruskar-Wallis test for different features' weights extracted from turn trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$7.71\ 10^{-9}$	0.0008	$6.39\ 10^{-22}$	$1.59\ 10^{-15}$	$3.01\ 10^{-15}$	$6.47\ 10^{-5}$
Effect size	0.2456	0.0850	0.6639	0.4593	0.4505	0.1201

of assumptions on the distribution of data by transforming the data into a uniform distribution of ranks regardless of the underlying distribution. The procedure still requires a set of assumptions to be satisfied: independent sets of data, sufficient sample size, continuous or ordinal data. The null hypothesis is an hypothesis which considers a difference in data (in this case, weights values) be due to chance rather than the effect of an external factor (cycling style). Hence, the null hypothesis of this test is that the mean ranks of the three distributions related to the three cycling styles are equal. The test determines if the null hypothesis is rejected or not based on a procedure similar to ANOVA's, but considering mean of ranks instead of mean of data values.
The assumptions of the Kruskar-Wallis test are satisfied by the data: the sets of weights are independent, each group has at least 30 values and the data is continuous. An additional assumption on similarity of distribution shape can be considered. The similarity of distributions was qualitatively investigated by inspecting histogram representation of data and quantitatively checked by means of the MATLAB function `fitmethis` [8], which finds the distribution that best fits data among all distributions available in MATLAB. The results of both the methods did not show similarity of distribution shape among groups. However, this is not a necessary assumption for the Kruskar-Wallis test. When this assumption is not respected, the results need to be interpreted in terms of mean ranks difference, which intuitively represent the tendency of a certain group to show higher or lower valued weights with respect to the others, instead of medians difference.
The p-value and effect size given by the Kruskar-Wallis test are shown in tables 5-5 and 5-6 for turns and straight segments respectively. The effect size is calculated as the etasquared η^2 coefficient, as in [28]:

$$\eta^2 = \frac{H - k - 1}{n - k} \tag{5-3}$$

With H the statistics resulting from the Kruskar-Wallis test, k the number of groups and n the total number of observations. The resulting value is comprised between 0 and 1, with a value <0.2 indicating a weak effect size.

The p-values resulting from the test show that the null hypothesis is always rejected, so the

Table 5-6: P-value resulting from Kruskar-Wallis test for different features' weights extracted from straight trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$2.21\ 10^{-18}$	0.00100	$2.11\ 10^{-17}$	$8.72\ 10^{-21}$	$3.66\ 10^{-16}$	$2.55\ 10^{-5}$
Effect size	0.6956	0.1017	0.6553	0.7928	0.6060	0.1680

three groups do not come from the same distribution. However, the effect size is weak for features *distance from the middle of the cycling path* and *negative acceleration*. The results of the Kruskar-Wallis test do not provide information on which group means are different. The information about which pairs of means are significantly different is important in order to evaluate how each cycling style is different from the others. Given significant Kruskar-Wallis tests, the pairwise comparison is performed by means of the post-hoc multicomparison test with Dunn-Sidák's approach, which provides a correction factor to take into account the errors arising from multiple comparisons.

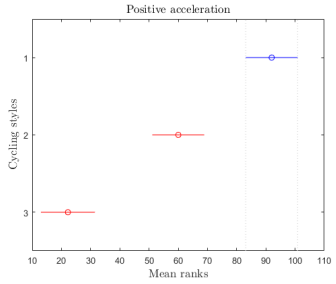


Figure 5-15: Feature *positive acceleration*

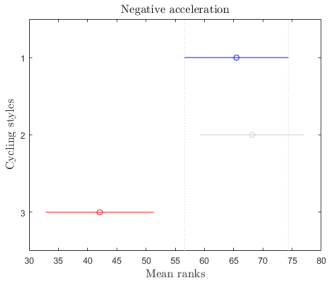


Figure 5-16: Feature *negative acceleration*

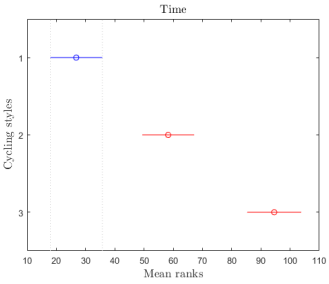


Figure 5-17: Feature *travel time*

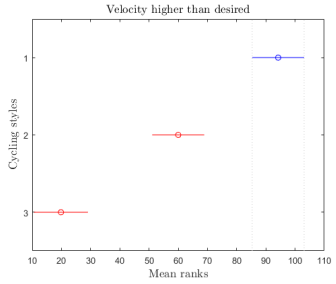


Figure 5-18: Feature *velocity higher than desired*

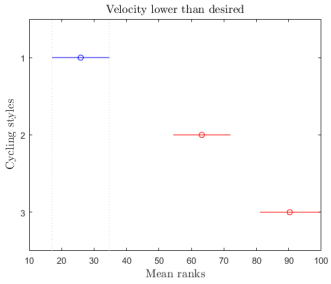


Figure 5-19: Feature *velocity lower than desired*

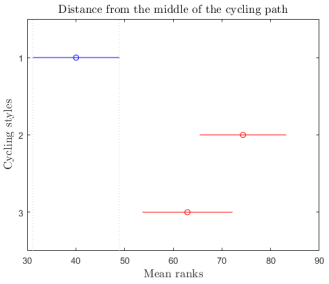


Figure 5-20: Feature *distance from the middle of the cycling path*

Figure 5-21: Multi comparison Dunn-Sidák test for weights of different features for straight segments

The results of the test are shown in Figures 5-21 and 5-28. The x-axis represents the mean

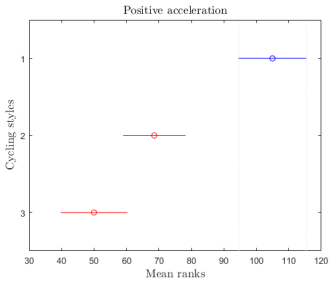


Figure 5-22: Feature *positive acceleration*

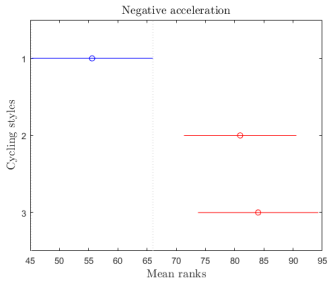


Figure 5-23: Feature *negative acceleration*

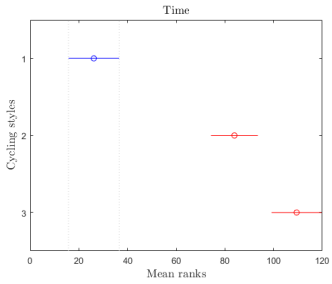


Figure 5-24: Feature *travel time*

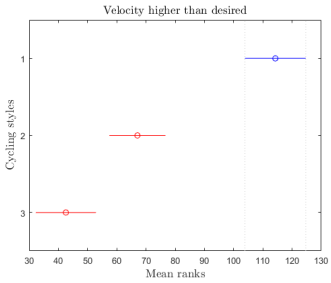


Figure 5-25: Feature *velocity higher than desired*

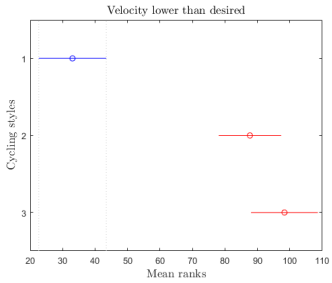


Figure 5-26: Feature *velocity lower than desired*

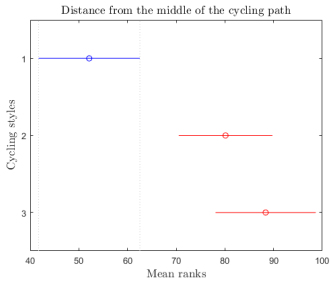


Figure 5-27: Feature *distance from the middle of the cycling path*

Figure 5-28: Multi comparison Dunn-Sidák test for weights of different features for turn segments

ranks for each feature, while the y-axis the cycling style of the analyzed weights distribution: starting from the bottom cautious, normal and aggressive. The intervals do not represent the variance, but are test intervals: if they are separated, two groups are statistically different. It is noticeable from the figures that the first (bottom - cautious cycling style) and last (up - aggressive cycling style) intervals do not overlap for most of the features. This implies that performed tests show a clear statistical difference between the cautious cycling style and aggressive cycling style for all weights.

In particular, the test shows that the mean ranks of the feature *positive acceleration* tend to be higher for cautious cycling style groups, meaning that this group tends to minimize positive accelerations more than others. At the same way, the mean ranks of the weights to the feature *travel time* result to be increasing from the more cautious to the more aggressive cycling style, which is intuitively connected to a higher importance given to minimizing travel time for more aggressive cyclists. The mean ranks of weights relative to features *velocity higher than desired* show a similar pattern: cautious cycling style weights tend to have higher ranks than the others, since cautious cyclists tend to avoid cycling at higher speed than the desired one. On the other hand, for feature *velocity lower than desired* the opposite holds: aggressive and normal cycling style groups have higher mean ranks than the cautious one, since the importance given to minimizing velocity lower than the desired ones is higher for these cyclists. For feature *distance from the middle of the cycling path*, the Kruskar-Wallis test shows that the weights of cautious cycling style have lower mean rank with respect to normal and aggressive styles, implying that for this group it is less important to minimize the

Table 5-7: P-values and effect sizes resulting from the comparison of different features' weights for cautious and normal cycling style extracted from straight trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$7.65\ 10^{-5}$	0.9784	0.00010	$1.95\ 10^{-5}$	2.20^{-6}	1.86^{-5}
effect size	0.2883	0.0032	0.3309	0.3612	0.3570	0.1354

distance from the middle of the cycling path, maybe trying to maximize the margins with respect to the other lane. However, the results do not show a clear difference between normal and aggressive cycling styles in this matter. At the same way, the weights of the feature *negative acceleration* does not show differences between normal and aggressive cycling styles. For each comparison, the effect size was derived by means of the Mann-Whitney U-test, which is the version of Kruskar-Wallis test used to compare only two groups. The effect size is calculated as the r^2 coefficient, as suggested in [28]:

$$r^2 = \frac{Z^2}{n} \tag{5-4}$$

with Z the value resulting from the Mann-Whitney U-test, n the total number of observations. The resulting value is comprised between 0 and 1, with a value <0.2 indicating a weak effect size. The p-values resulting from the Dunn-Sidák-corrected pairwise comparisons and effect sizes are reported in tables 5-7,5-9,5-8 for straight segments and 5-10,5-12,5-11 for turn segments. For each segment, the different combinations between cycling styles are considered, analyzing the difference between cautious-normal, normal-aggressive and aggressive-cautious separately. A low p-value (< 0.05) indicates that the statistical difference between groups is relevant. From the p-values reported in the tables it results that the difference between cycling and aggressive cycling style is confirmed for all the features in both of the segment types, while this does not hold for the cautious and normal cycling style weights of the feature *negative acceleration* in straight trajectories and aggressive and normal cycling style weights of the feature *distance from the middle of the cycling path* in straight trajectories. ´Table 5-12 shows that there is no significant difference between weights of normal and aggressive cycling styles for turns since the p-value is higher than 0.05 and thus the null hypothesis is not rejected. On the other hand, for straight segments the test shows that statistically significant difference is present also when normal and aggressive cycling styles are compared. From the effect size values, it follows that the features *travel time* and *velocity higher than desired* have the highest effect size, meaning that the relationship between the cycling styles and weight values is particularly strong for these features. On the other hand, *negative acceleration* and *distance from the middle of the cycling path* show a weak effect size for most of the considered comparisons.

5-3-2 Clustering analysis

The proposed algorithm's capability of extracting parameters that represent the three cycling styles presented in Section 3-3-2 is evaluated by performing a clustering analysis on the

Table 5-8: P-values and effect sizes resulting from multiple comparison tests of different features' weights for cautious and aggressive cycling style extracted from straight trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	0	0.0074	0	0	0	0.0093
effect size	0.4536	0.0924	0.3634	0.4754	0.3886	0.1048

Table 5-9: P-values and effect sizes resulting from multiple comparison tests of different features' weights for normal and aggressive cycling style extracted from straight trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$3.12\ 10^{-6}$	0.0022	$8.14\ 10^{-6}$	$6.38\ 10^{-7}$	0.00129	0.364
effect size	0.3692	0.0813	0.4055	0.4550	0.2343	0.0350

Table 5-10: P-values and effect sizes resulting from multiple comparison tests of different features' weights for cautious and normal cycling style extracted from turn trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$5.03\ 10^{-5}$	0.0085	$2.08\ 10^{-11}$	$9.73\ 10^{-8}$	2.63^{-10}	0.00331
effect size	0.1629	0.0720	0.4635	0.2534	0.3469	0.0705

Table 5-11: P-values and effect sizes resulting from multiple comparison tests of different features' weights for cautious and aggressive cycling style extracted from turn trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	$8.07\ 10^{-9}$	0.0012	0	$1.33\ 10^{-15}$	$6.79\ 10^{-14}$	7.60^{-5}
effect size	0.1888	0.0700	0.4416	0.3768	0.3214	0.1222

Table 5-12: P-values and effect sizes resulting from multiple comparison tests of different features' weights for normal and aggressive cycling style extracted from turn trajectories

Features	Positive acceleration	Negative acceleration	Travel time	Velocity higher than desired	Velocity lower than desired	Distance from the middle of the cycling path
p-value	0.1783	0.8765	0.0052	0.0120	0.4017	0.6148
effect size	0.0405	0.0060	0.1384	0.0809	0.0290	0.0076

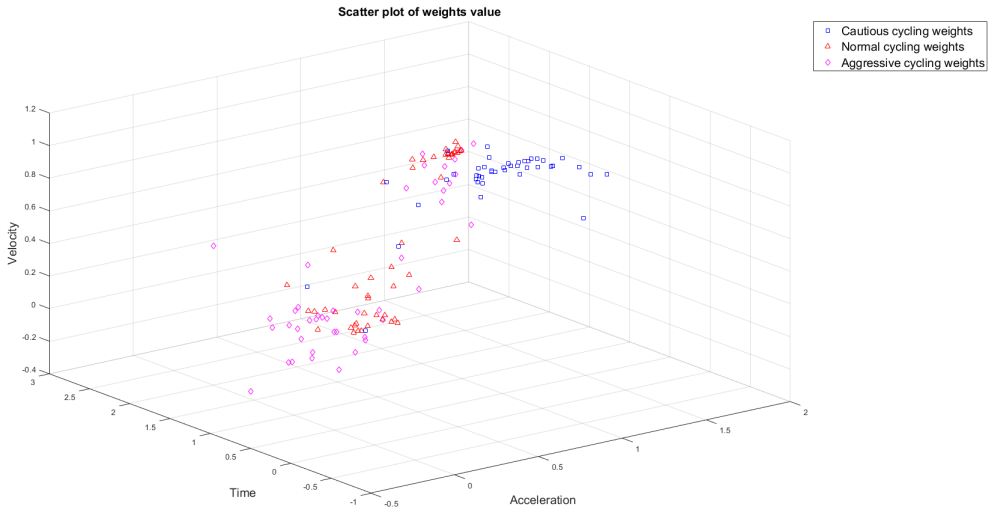


Figure 5-29: Scatter plot of the values *positive acceleration*, *velocity higher than desired* and *travel time* features for turns

weights. Intuitively, groups of weights related to different cycling style should be split into three different clusters. Unsupervised clustering algorithms can be used in the feature space defined by the features described in Section 4-2.
K-means Clustering was chosen as clustering approach. It is part of the class of methods called partitional clustering, which aim to group the data points into a pre-specified number of clusters K, which is suitable for the scope of this analysis, since the number of clusters is the known number of cycling styles. The main advantage of this approach is that it is easy to implement and computationally efficient. A key-concept in clustering is the (dis)similarity measure, since points in the same cluster are closer than points in different clusters, and it is needed to estimate the distance between them. The most common measure is the Euclidean distance, but others are possible, as the Random Forest predictor in Chen et al. [6].
In general, clustering has been used with data-points given by quantities that can directly be connected to the driving action. In this case, this technique is employed to evaluate the learning algorithm's results in terms of connection of the weights distribution to different cycling styles.
In order to visualize the clusters, a set of three features was selected to plot the weights values in 3 dimensions. Then, K-means clustering was applied on the feature space defined by all the features, and the results evaluated by means of a confusion matrix compared the original labels (the cycling styles of the empirical trajectories) and the ones assigned by the clustering algorithm.
First, K-Means clustering was performed on weight values of the features *positive acceleration*, *velocity higher than desired* and *travel time*. The features were chosen in order to consider three different kind of information on the cycling style: acceleration, velocity and time. The choice between positive and negative acceleration and velocity is motivated by the effect sizes and test intervals shown in Figures 5-21 and 5-28. For both straight and turn segments, the feature *velocity higher than desired* has higher effect size than the feature *velocity lower than desired*, and the same applies to the feature *positive acceleration*. Moreover, the feature *neg-*

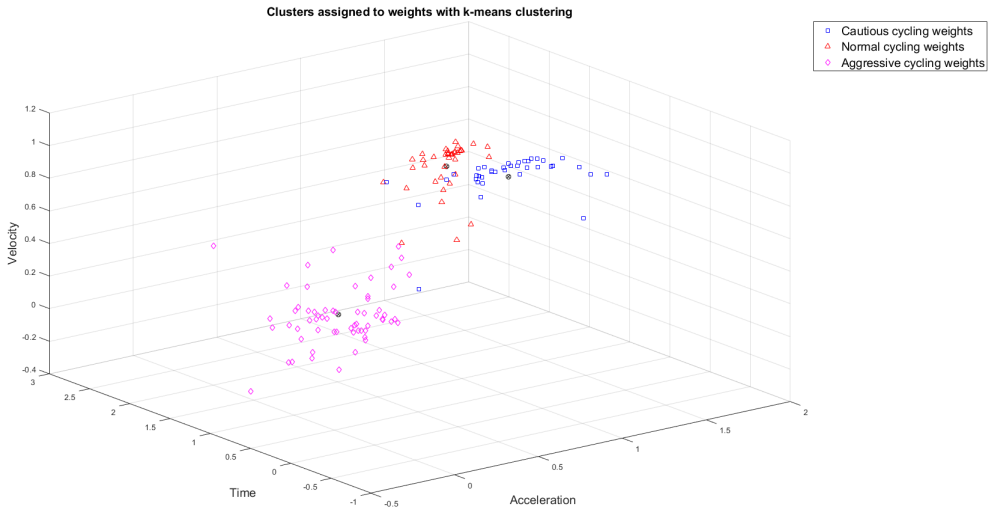


Figure 5-30: Clusters of the values *positive acceleration*, *velocity higher than desired* and *travel time* features for turns

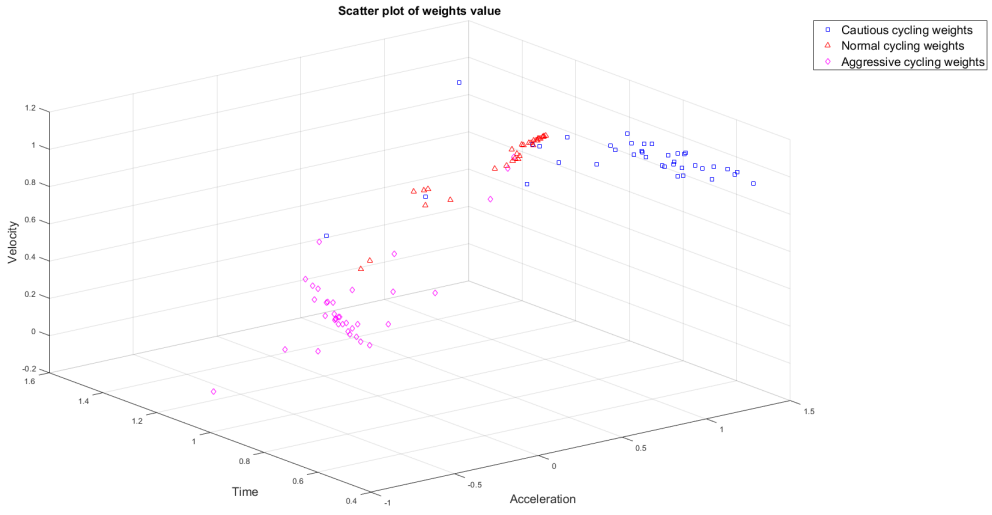


Figure 5-31: Scatter plot of the values *positive acceleration*, *velocity higher than desired* and *travel time* features for straight segments

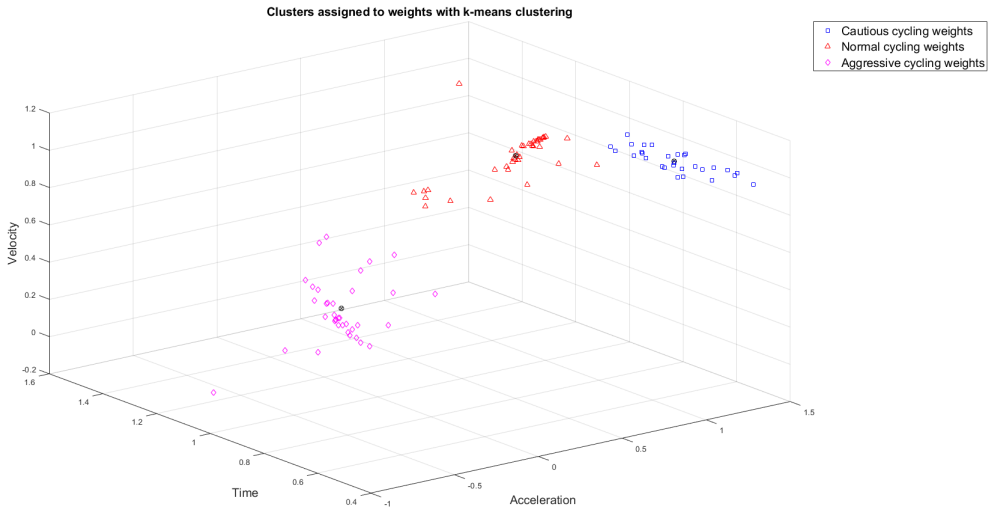


Figure 5-32: Clusters of the values *positive acceleration*, *velocity higher than desired* and *travel time* features for straight segments

ative acceleration and *distance from the middle of the cycling path* did not show significant difference between cycling styles, so they were not chosen as relevant features for the first clustering analysis. From a visual analysis of the clusters, it is clear that the aggressive and cautious styles' clusters are distinguishable for both turns and straight segments. The scatter plot in Figure 5-29 shows that the normal cycling style weights do not form a clearly distinguishable cluster, confirming the results of the statistical test. However, K-means clustering algorithm can distinguish three clusters, one of them being the union of a part of normal cycling style weights and aggressive ones. This may be due to different behaviour of participants, for some of them cycling in a normal way could be similar to others participants' aggressive cycling style.
Then, K-Means clustering was performed on the weight values of all the features. For each set of weights, the algorithm returns one cluster index and the centroid location of the cluster. In order to evaluate the results by connecting the clusters found with K-Means to the cycling styles, each index was linked to the cluster defined by weight values with the closest centroid with respect to the calculated one. Figure 5-33 presents the confusion matrix comparing the assigned and original clusters for weights derived from straight segments, in Figure 5-34 the same applies to turns. The confusion matrices confirm the insight given by the K-Means clustering algorithm applied to a subset of weights corresponding to three features: for straight segments, the derived weights show statistical differences between cycling styles, and thus can be clustered into three different groups. For turns, the cautious cycling style group is a well-defined cluster, while normal and aggressive cycling style weights cannot be distinguished effectively.

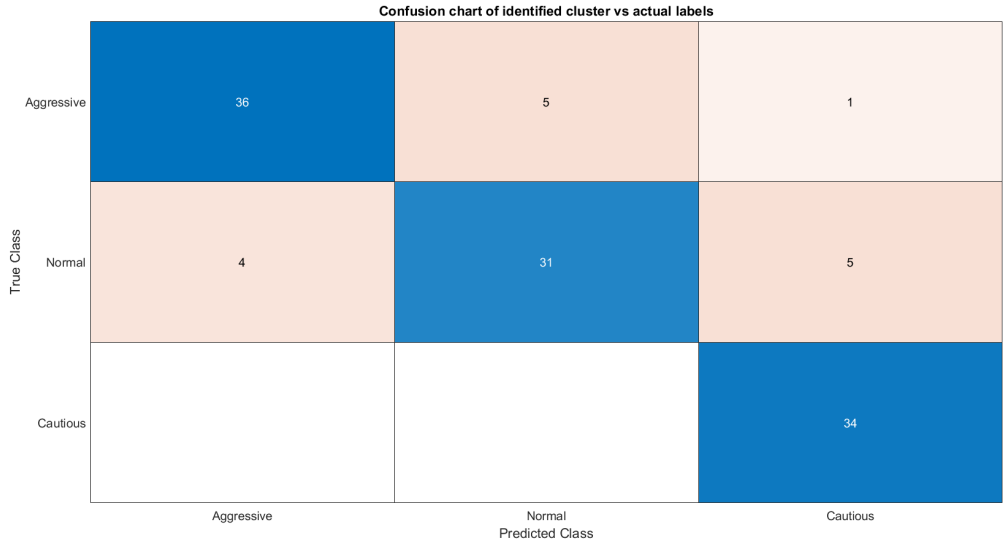


Figure 5-33: Confusion matrix for the results of K-Means clustering algorithm applied to weights derived from straight segments

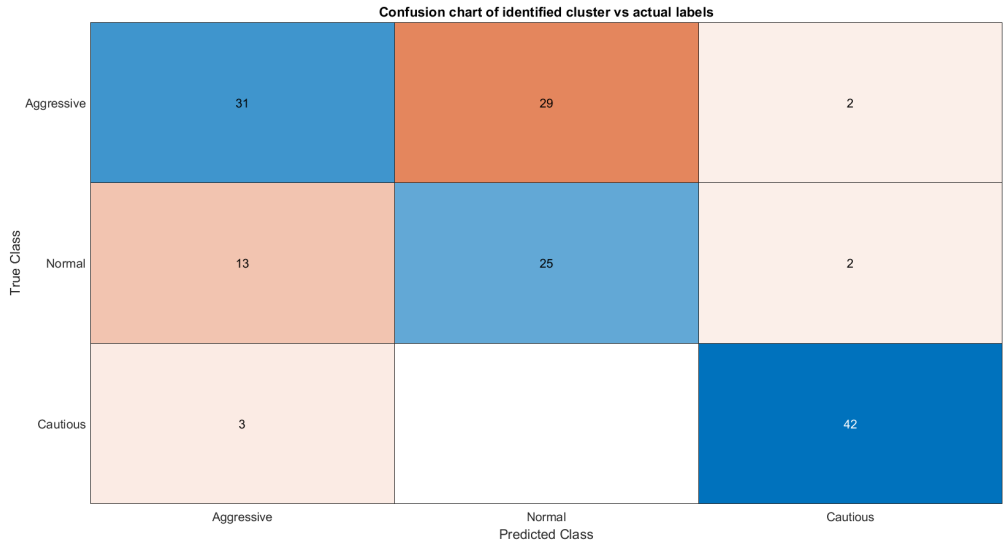


Figure 5-34: Confusion matrix for the results of K-Means clustering algorithm applied to weights derived from turn segments

5-4 Discussion

The experimental setup described in Chapter 3 provides cycling trajectory data of three different cycling style: aggressive, normal and cautious. In this chapter, the capability of the IRL algorithm described in Chapter 4-3 of learning these different cycling styles is evaluated. The cycling style imitation is analyzed in terms of convergence towards human demonstration. After that, a simulation study compares the cycling style resulting from the learned reward functions against empirical cyclists' test trajectories. Then, a statistical analysis on the group of weights related to different cycling styles is performed, to test the implication of different cycling behaviours to the difference in weight distribution.
The convergence of the learning algorithm is measured by the feature error over optimization iterations. The error is calculated as the norm between simulated and empirical features, and it is calculated at every iteration. For both turns and straight segments, the error rapidly converges to a final asymptotic value, after 20 iterations. Smaller error means an improved reward function that leads to better generated demonstrations. The limitations of the convergence can be due to the assumption of a linear reward function with hand-crafted features and time-invariant weights, which may not fully represent human cycling behaviour. However, the algorithm converges to reward functions that can generate trajectories similar to the empirical ones, and the errors improve during learning.
For each demonstration, the empirical and optimized trajectories, the extracted weights and the cycling style are saved, defining three groups of trajectories and weights corresponding to aggressive, normal and cautious cycling.
First, the empirical and optimized trajectories of each cycling style group are averaged and split into different road segment in order to visually appreciate the shape of the speed profiles. The similarity between the speed profiles of the averaged optimized and demonstrated trajectories suggests that the algorithm learns the high-level characteristics of different profiles. In particular, the travel time, the velocity range of the learned cautious and normal cycling styles are consistent with the empirical averages. Regarding aggressive cycling, the generated trajectories can capture the abrupt changes in velocity and the importance given to reducing travel time, resulting in fast trajectories, but some characteristics of the speed profile, as increasing the velocity at the end of straight segments, are not learned by the algorithm. Moreover, normal and aggressive cycling styles speed profile are similar in curves.
In the simulation study, the cycling style of the aforementioned reward functions is compared with empirical test trajectories. The test set consists of three groups of empirical trajectory, each one related to a set of cycling preferences. The trajectory error between the trajectories generated with the three rewards and the empirical one is used as a human likeness evaluation metric, and the average error is calculated for the three groups of trajectories in the test set. The results show that the reward function learned for a specific style leads to a strong improvement in terms of similarity to test trajectories with the same cycling style with respect to other cycling styles. When the normal cycling style trajectory test set is considered, the improvement is still present, but it is reduced. A possible explanation is that normal cycling style shares some characteristics with cautious cycling, e.g. the tendency to maintain the desired speed, and aggressive cycling, as reducing travel time. Moreover, the data related to this category have a higher variability: this can result from the fact that experimental instructions for aggressive and cautious cycling are very clear and easily interpreted, while normal cycling is participant-dependent, so the difference in trajectory data result from a

difference in the participants' notion of normal cycling. Since the proposed algorithm is able to learn the differences in these preferences, the averaged reward is more sparse.
Since the identified weights are the main output of the algorithm, an analysis of the distribution of the three groups of weights was performed. Each cycling style defines a set of weights, but some features may be more informative than other for distinguishing the styles. In order to test the statistical difference between the distributions of weights for different cycling style, the non-parametric Kruskar-Wallis test is employed. Each feature is considered separately to understand which features are linked to distribution of weights with higher difference between cycling style. Then, cycling styles are pairwise compared. The test confirms the statistical difference between weights distribution of different cycling styles with relevant effect sizes for all the features except *negative acceleration* and *distance from the middle of the cycling path.* This could mean that the negative acceleration and distance from the middle of the cycling path are not relevant in the characterization of the proposed cycling styles: in particular, a cyclist negative acceleration is usually segment-specific: for example, in turns or before stops, the cyclists are asked to decelerate in order to maintain stability, while accelerating depends more heavily on cycling intentions. Distance from the middle of the cycling path is a feature that is connected to safety margins of cyclist. During the performed experiments, very few road users were present in order to guarantee a safe environment for participants. As a consequence, cyclists had the freedom to ride in the whole cycling path. When higher levels of interaction with other cyclists are present, a safety feature defined by the distance from the middle of the cycling path and distance from other cyclists would be more relevant for characterizing the cycling style, as reported in the literature. Another possible explanation could derive by the fact that for each road segment, an average of 20% of the trajectories look horizontally shifted, even though the speed profile is reasonable. Probably the differential GPS could not resolve the ambiguities of some position points and estimated the next position based on the previous ones, so the information on the distance from the road edges may not be completely reliable and thus the corresponding weights.
Positive acceleration, *Velocity higher than desired* and *Travel time* are the feature for which the different cycling styles' weights distribution show a higher distance in the Kruskar-Wallis test. They have been selected as input features for a scatter plot and a clustering K-means algorithm. For straight segments, the difference between the three cluster is clearly visible. For turns, the scatter plot of the weights shown in Figure 5-29 show a clear distinction between the cautious cycling style cluster and the other two, while it is not possible to clearly distinguish aggressive and normal cycling. This outcome confirms the Kruskar-Wallis test test for curves reported in 5-28, where no clear statistical difference between normal and aggressive cycling style weight distributions could be appreciated. A possible reason to explain the difference with respect to straight segments is that all the straight road segments shared the same characteristics, while there is more variability for curves. Curves starting after a straight segment, after a stop or approaching a stop were joined in the same turn trajectory dataset. Each of these road segments has different characteristics that can influence the cyclist behaviour beyond his cycling style. For this reason, the influence of cycling style may be reduced and the corresponding weights could show less statistical difference. The outcomes of the weight analysis for normal and aggressive cycling style weights can be connected to the results of the simulation study, which indicates that normal cycling is not well defined as an individual cycling style, but rather shares characteristics of aggressive cycling, depending on the road user. A concept that could help representing an intermediate cycling style is moderate cycling, which is cycling without being too aggressive or cautious. The clustering

algorithm shows that a group of weights forms a cluster that is distinguishable by the other two. The algorithm has been tested on the capability to learn the instructed cycling styles, but the real cycling style of the demonstrated trajectories could be different with respect to the label given by the experimental instruction. For this reason, the weights for trajectories labelled differently could be similar because their corresponding cycling style is actually similar.

Chapter 6

Conclusions and recommendation

Cycling as a transportation mode not only has an impact on urban challenges such as reducing traffic congestion and air pollution, but also improves the lifestyle of individuals providing an easy and regular physical activity. Designing infrastructures and services that make cycling more appealing can improve living conditions in urban areas and promote a healthier lifestyle. In particular, personalized travel services has been gaining interest in the past years, as personalized speed or road advices that can reduce travel time and increase safety while taking into account the personal preferences of cyclists. After learning individual cyclist preferences, travel services can be tuned in order to meet the user demands and needs, increasing the acceptance of the system and compliance to the advices.
This thesis proposed a version of Inverse Reinforcement Learning to characterize cycling style by learning cyclist preferences from experimental data. This method models the preferences of the cyclist with a reward function he optimizes while cycling. The reward function is composed of a weighted sum of features representing relevant characteristics of the cycling action. The learning algorithm learns the weights corresponding to each cycling demonstration, which express the trade-off between different goals, such as keeping desired speed, reducing acceleration and minimizing travel time. Given the continuous nature of the traffic domain, a continuous version of IRL was used, in line with the literature of driving style identification with IRL. In this chapter, the answers to the questions introduced in Chapter 1 are discussed and recommendation for future research are given.

6-1 Conclusions

The goal of this thesis is defined as:

> *Given a set of fixed external conditions, can different cycling styles be learned from cyclists' trajectories using Inverse Reinforcement Learning?*

The following sub-questions were designed to answer the main research question of this thesis project, and will be discussed:

How to design and implement an experimental setup that provides with the trajectory data necessary to implement and test the proposed algorithm?
The main requirement for the design of the experimental setup is that the gathered data should be informative enough to be used to investigate the difference in cycling preferences of participants. In order to do that, an experimental setup where participants were asked to cycle according to different cycling styles was designed. The experimental instructions explicitly states the cycling preferences the participants should show, and each set of preferences is assumed to be connected to a cycling style. In particular, avoiding intense acceleration and maximizing safety margins are associated to a cautious cycling style while minimizing travel time, abrupt accelerations are associated to a more aggressive cycling style.
The collected data show differences in cycling style which can be qualitatively appreciated in Figures 5-7,5-8,5-9 and 5-10. In particular, the speed profiles of the three cycling styles show qualitative shape difference, which gives information on the tendency to accelerate or keep the desired velocity, and the travel time duration. Moreover, performing the experiment in a real-world environment made it possible to capture naturalistic cyclist behaviour in terms of interaction with the environment. The experiment, however, was conducted in off-peak hours in order to guarantee a safe experience to the participants. As a consequence, a limited number of interactions with other road users happened. Moreover, since the experiment was performed with sensor-equipped bikes, only the data of the participants were available so the information related to interaction with other users, such as distance during overtaking or following, could not be derived from the gathered data. Considering these factors, the experimental setup is considered to be apt for the scope of learning cycling preferences from trajectory data, but it would benefit from improvements on gathering the data related to interaction between cyclists.

Which features can represent cycling style and how can they be mathematically expressed?
Do the identified parameters related to different cycling styles show statistical differences?

The features proposed to represent cycling style give a good description of the cyclist's inner reward function. They take into account different characteristics of the cycling action, including acceleration, velocity, travel time and safety. Since cyclists' behaviour is constrained by the physical characteristics of a bike, jerk was not taken into account in this description. As mentioned in Chapter 5, the difference in cycling styles resulted in a clear statistical difference in weight distributions for cautious and aggressive cycling. Regarding normal cycling, the weights did not show clear statistical differences between normal and aggressive cycling for turns in most of the features. This may be due to a limited difference between normal and aggressive cycling during curvy road segments, since most

of the cyclists tend to keep their stable velocity during turns, while the cautious ones decelerate further. The difference between the characteristics of different curvy road segments could have also played a role, and another possible reason is that the participants' interpretation to the normal cycling style instructions may be different from one person to the other, resulting in a less defined distribution of weights. The results confirmed that higher the real difference between cycling preferences, the higher the statistical difference between weights. As a consequence, the algorithm could learn the importance given to different features by cyclists riding according to different cycling styles.

Can the learned reward function simulate trajectories that fit the empirical ones?

In the simulation study, the capability of the reward functions averaged with respect to each cycling style's weight to predict trajectories that reflect the original cycling style is evaluated. The generated trajectories are compared with test trajectories run with different cycling styles. The results show that the learnt reward functions can effectively generate trajectories which show different cycling preferences. In particular, the improvement in the error between trajectories generated by the reward function trained on trajectories run with the same cycling style and the other reward functions trained on the other cycling styles is particularly strong for cautious and aggressive cycling styles, which is intuitively reasonable and it is in line with the results on weight distributions difference. In fact, the feature-based representation of human behaviour leads to good generalization capabilities, since the model learns the importance given to a set of general features rather than the trajectory itself.

It can be concluded that the proposed experimental setup provided informative data to apply the proposed data-driven method for learning cycling preferences. During learning, the algorithm was able to converge close to the vicinity of the empirical trajectories. However, some limitations need to be acknowledged: two assumption of the algorithm, namely the optimality of demonstrations and a linear time-invariant reward function model may not hold in every situation. Human behaviour is complex and the demonstrations are not always optimal with respect to the reward function, thus the algorithm cannot address sub-optimal and irrational decision. Moreover, the hand-crafted features may not give a complete representation of cycling style. The weights resulting from the learning processes show statistical differences when aggressive and cautious cycling styles are considered, while the difference is minor for normal cycling.

6-2 Future work

During this thesis project, several fields for future research and improvement were found:

Experimental setup

The current experimental setup provided informative data. However, it does not take into

account interaction between cyclists and some information, such as the distance between cyclists and other road users, are lost. Moreover, it was not possible to learn a unique individual cyclist preferences, because it would require a big amount of data for each person. In order to gain new insight about different cycling styles in interactive situations, a new experimental setup should be developed in order to take into account the interaction between individuals.

Scenario

More complex scenarios should be considered, for example queue forming, complex intersections. Adding these scenarios would provide information on complex cycling patterns, which can be useful in learning the cycling style in interactive environments. Moreover, cyclist's reaction to traffic light could be investigated as part of a personalized model of each individual cyclist.

Feature definition

New features should be designed in order to take into account interaction with other cyclists and other informative characteristics of cycling style in a naturalistic environment. These features may include distance between cyclists, velocity difference, features related to specific manoeuvres as overtaking and following. A comprehensive reward function could describe complex behaviours, by taking into account different characteristics of the cycling action.

Inverse Reinforcement Learning algorithm

Modelling the reward function as a linear sum of features has shown to be effective in several driving and cycling style identification. However, more complicated behaviours could be addressed by defining a more complex parametrization of the reward function, such as using neural networks to map raw states to reward values. Manual feature definition relies on domain knowledge and human experience, and may not always the be the more accurate description of the cycling style. Deep neural networks have shown to be a promising method for extracting informative features directly from data, and they could be employed in this field.

Bibliography

[1] Nematollah Ab Azar, Aref Shahmansoorian, and Mohsen Davoudi. From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, 50:119–138, 2020.

[2] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.

[3] Rushdi Alsaleh and Tarek Sayed. Modeling pedestrian-cyclist interactions in shared space using inverse reinforcement learning. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2020.

[4] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 182–189, Fort Lauderdale, FL, USA, 2011. PMLR.

[5] H. van der Marel C. Tiberius and H. Dun. Introduction to geophysics and remote sensing. Assignment on high-precision GPS precision, 2021.

[6] Kuan-Ting Chen and Winnie Chen. Driving style clustering using naturalistic driving data. *Transportation Research Record: Journal of the Transportation Research Board*, 2019.

[7] A. Dabiri, A. Hegyi, and S. Hoogendoorn. Optimized speed trajectories for cyclists, based on personal preferences and traffic light information-a stochastic dynamic programming approach. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–17, 2020.

[8] Francisco de Castro. http://www.mathworks.com/matlabcentral/fileexchange/40167-fitmethis, 2022.

[9] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. *Proceedings of the 33Rd International Conference on Machine Learning*, 2016.

[10] Alexandra Gavriilidou, Maria J. Wierbos, Winnie Daamen, Yufei Yuan, Victor L. Knoop, and Serge P. Hoogendoorn. Large-scale bicycle flow experiment: Setup and implementation. *Transportation Research Record*, 2019.

[11] C. Gote, M. Flad, and S. Hohmann. Driver characterization driver specific trajectory planning: an inverse optimal control approach. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014.

[12] P. C. Gregory. *Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with 'Mathematica' Support.* 2005.

[13] Zhiyu Huang, Jingda Wu, and Chen Lv. Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning. In *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[14] Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 2019.

[15] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[16] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Robotics: Science and Systems*, 2012.

[17] Seong Lee and Zoran Popovic. Learning behavior styles with inverse reinforcement learning. *ACM Trans. Graph.*, 2010.

[18] Na Lin, Changfu Zong, Masayoshi Tomizuka, Pan Song, Zexing Zhang, and Gang Li. An overview on study of identification of driver behavior characteristics for automotive control. *Mathematical Problems in Engineering*, 2014, 2014.

[19] Francisco Martinez-Gil, Miguel Lozano, Ignacio García-Fernández, Pau Romero, Dolors Serra, and Rafael Sebastián. Using inverse reinforcement learning with real trajectories to get more trustworthy pedestrian simulations. *Mathematics*, 2020.

[20] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 9.11.0.1769968 (R2021b)*, 2021.

[21] Hossameldin Mohammed, Tarek Sayed, and Alexander Bigazzi. Toward agent-based microsimulation of cyclist following behavior: Estimation of reward function parameters using inverse reinforcement learning. In *98th Annual Meeting of the Transportation Research Board*, 2019.

[22] Hossameldin Mohammed, Tarek Sayed, and Alexander Bigazzi. Microscopic modeling of cyclists on off-street paths: a stochastic imitation learning approach. *Transportmetrica A: Transport Science*, 2021.

[23] Nornadiah Mohd Razali and Bee Yap. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *J. Stat. Model. Analytics*, 2011.

[24] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum margin planning. In *Proceedings of the 23rd International Conference on Machine Learning*, page 729–736. Association for Computing Machinery, 2006.

[25] Sascha Rosbach, Vinit James, Simon Großjohann, Silviu Homoceanu, and Stefan Roth. Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving. *CoRR*, 2019.

[26] Russ Salakhutdinov. Deep reinforcement learning and control. http://www.cs.cmu.edu/~rsalakhu/10703/Lectures/Lecture_inverse_RL.pdf.

[27] Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018.

[28] Maciej Tomczak and Ewa Tomczak. The need to report effect size estimates revisited. an overview of some recommended measures of effect size. 2014.

[29] Aaron Tucker, Adam Gleave, and Stuart Russell. Inverse reinforcement learning for video games. *CoRR*, 2018.

[30] Jianping Wu, Yiman Du, Geqi Qi, and Ming Xu. Leveraging longitudinal driving behaviour data with data mining techniques for driving style analysis. *IET Intelligent Transport Systems*, 2015.

[31] Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters*, 2020.

[32] Bing Zhu, Yuande Jiang, Jian Zhao, Rui He, Ning Bian, and Weiwen Deng. Typical-driving-style-oriented personalized adaptive cruise control design based on human driving data. *Transportation Research Part C: Emerging Technologies*, 2019.

[33] Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, 2010.

[34] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*, 2008.