FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES (UB)
ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA (URV)
UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)
UNIVERSITAT ROVIRA I VIRGILI (URV)
UNIVERSITAT DE BARCELONA (UB)

MASTER THESIS

Social-aware Drone Navigation using Social Force Model

Author:
Luis Alberto GARZA ELIZONDO

Supervisors:
Dr. René ALQUÉZAR MANCHO
Dra. Anaís GARRELL ZULUETA

A thesis submitted in fulfillment of the requirements for the degree of Master in Artificial Intelligence

in the

Barcelona School of Informatics

October 7, 2016

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

Abstract

Barcelona School of Informatics

Master in Artificial Intelligence

Social-aware Drone Navigation using Social Force Model

by Luis Alberto GARZA ELIZONDO

Robot's navigation is one of the hardest challenges to deal with, because real environments imply highly dynamic objects moving in all directions. The main ideal goal is to conduct a safe navigation within the environment, avoiding obstacles and reaching the final proposed goal. Nowadays, with the last advances in technology, we are able to see robots almost everywhere, and this can lead us to think about the robot's role in the future, and where we would find them, and it is no exaggerated to say, that practically, flying and land-based robots are going to live together with people, interacting in our houses, streets and shopping centers. Moreover, we will notice their presence, gradually inserted in our human societies, every time doing more human tasks, which in the past years were unthinkable.

Therefore, if we think about robots moving or flying around us, we must consider safety, the distance the robot should take to make the human feel comfortable, and the different reactions people would have. The main goal of this work is to accompany people making use of a flying robot. The term social navigation gives us the path to follow when we talk about a social environment. Robots must be able to navigate between humans, giving sense of security to those who are walking close to them. In this work, we present a model called Social Force Model, which states that the human social interaction between persons and objects is inspired in the fluid dynamics defined by Newton's equations, and also, we introduce the extended version which complements the initial method with the human-robot interaction force.

In the robotics field, the use of tools for helping the development and the implementation part are crucial. The fast advances in technology allows the international community to have access to cheaper and more compact hardware and software than a decade ago. It is becoming more and more usual to have access to more powerful technology which helps us to run complex algorithms, and because of that, we can run bigger systems in reduced space, making robots more intelligent, more compact and more robust against failures. Our case was not an exception, in the next chapters we will present the procedure we followed to implement the approaches, supported by different simulation tools and software. Because of the nature of the problem we were facing, we made use of Robotic Operating System along with Gazebo, which help us to have a good outlook of how the code will work in real-life experiments.

In this work, both real and simulated experiments are presented, in which we expose the interaction conducted by the 3D Aerial Social Force Model, between humans, objects and in this case the AR.Drone, a flying drone property of the Instituto de Robótica e Informática Industrial. We focus on making the drone navigation more socially acceptable by the humans around; the main purpose of the drone is to accompany a person, which we will call the "main" person in this work, who is going to try to navigate side-by-side, with a behavior being dictated with some forces exerted by the environment, and also is going to try to be the more socially close acceptable possible to the remaining humans around. Also, it is presented a comparison between the 3D Aerial Social Force Model and the Artificial Potential Fields method, a well-known method and widely used in robot navigation. We present both methods and the description of the forces each one involves.

Along with these two models, there is also another important topic to introduce. As we said, the robot must be able to accompany a pedestrian in his way, and for that reason, the forecasting capacity is an important feature since the robot does not know the final destination of the human to accompany. It is essential to give it the ability to predict the human movements. In this work, we used the differential values between the past position values to know how much is changing through time. This gives us an accurate idea of how the human would behave or which direction he/she would take next.

Furthermore, we present a description of the human motion prediction model based on linear regression. The motivation behind the idea of building a Regression Model was the simplicity of the implementation, the robustness and the very accurate results of the approach. The previous main human positions are taken, in order to forecast the new position of the human, the next seconds. This is done with the main purpose of letting the drone know about the direction the human is taking, to move forward beside the human, as if the drone was accompanying him. The optimization for the linear regression model, to find the right weights for our model, was carried out by gradient descent, implementing also de RMSprop variant in order to reach convergence in a faster way. The strategy that was followed to build the prediction model is explained with detail later in this work.

The presence of social robots has grown during the past years, many researchers have contributed and many techniques are being used to give them the capacity of interacting safely and effectively with the people, and it is a hot topic which has matured a lot, but still there is many research to be investigated.

Acknowledgements

First and foremost, I want to thank God, who letting me be here and letting me live an amazing experience full of love, health and peace. To my parents thanks for always support me unconditionally, and always encourage me to be a better person. Thanks to my advisors, Dra. Anais Garrell for always supporting me and giving me the opportunity to work with her, and Rene Alquézar for all his support since I arrived to Barcelona. Thanks to Fernando Herrero for his always kindly help during the experiments carried out in this work.

Contents

List of Figures

List of Abbreviations

IRI	**I**nstitut de **R**obotica i **I**nformatica Industrial
SFM	**S**ocial **F**orce **M**odel
ESFM	**E**xtended **S**ocial **F**orce **M**odel
ASFM	**A**erial **S**ocial **F**orce **M**odel
APF	**A**rtificial **P**otential **F**ield
ROS	**R**obot **O**perating **S**ystem
ICS	**I**nevitable **C**ollision **S**tates
PfCA	**P**rediction for **C**ompAction
BCM	**B**eam **C**urvature **M**ethod
LCM	**L**ane **C**urvature **M**ethod
UAV	**U**nmanned **A**erial **V**ehicle
HRI	**H**uman **R**obot **I**nteraction
RMSE	**R**oot **M**ean **S**quare **E**rror

List of Symbols

η	Positive potential factor of the repulsive potential
k_{apf}	Positive potential factor of the attractive potential
P_{ob}	Obstacle position.
$\rho(P, P_{ob})$	Distance between the position P and the uncertainty zone.
ρ_0	Scope of the repulsive potential.
$\mathbf{e_{pi}}$	Desired direction of person *pi*
$\mathbf{v^0_{pi}}$	Desired velocity of person *pi*
$\mathbf{r_{pi}(t)}$	Denotes the actual position of person *pi* at time t
$\mathbf{r_{iR}}$	is the distance between person *i* and robot *R*
v^0_{pi}	Desired speed of person *pi*
k^0_i	Relaxation factor used in SFM
A	Strength factor of the human-human interaction force
B	Strength factor of the repulsive interaction
d	Sum of the radii between two persons
$U_{io}(\|\|\mathbf{r_i o}\|\|)$	Monotonic decreasing potential
A_{iR}	Strength factor of the human-robot interaction force
γ_{iR}	Angle formed by the desired velocity of the pedestrian *i* and the vector r_iR
d_R	Human-Robot distance parameter to adjust
λ_{iR}	Strength of the anisotropic factor
n_{iR}	Normalized vector pointing from the robot *R* to the person *i*
$\psi(\gamma_{iR}, \theta_{iR})$	Extended anisotropic factor depending on γ_{iR} and θ_{iR}
h	Height of the person
φ_{iR}	Angle formed by $\mathbf{v_{pi}}$ and $\mathbf{r_{iR}}$
$w(\varphi_{iR})$	Anisotropic factor as a function of the angle φ_{iR}
ξ	Proportional factor between 0 and 1 multiplying 2D anisotropic factor
$\mathcal{A}$	Social distance area of pedestrian being accompanied
$\mathcal{B}$	Human being accomapanied field of vision area
$\mathcal{C}$	Personal space of the human being accompanied
α	Scalar positive parameter controller of $\mathbf{f^{goal}_{R,dest}}$
β	Scalar positive parameter controller of $\mathbf{f^{goal}_{Rj}}$
γ	Scalar positive parameter controller of $\mathbf{F^{int}_{Rj}}$
δ	Scalar positive parameter controller of $\mathbf{F^{int}_{Ro}}$
$\mathbf{f^{goal}_{R,dest}}$	Robot's attraction force to the tartget's destination
$\mathbf{f^{goal}_{Rj}}$	Robot's attraction force to the pedestrian *j*
$\mathbf{F^{int}_{Rj}}$	Repulsive force created by the sum of all the pedestrians in a range of view
$\mathbf{F^{int}_{Ro}}$	Repulsive force created by the sum of all the objects in a range of view
$\mathbf{F_R}$	Total force guiding the robot
v_{safety}	Maximum velocity when one person is inside the safety zone
v_{cruise}	Maximum velocity when someone is inside its social safety zone
v_{free}	Maximum velocity when no one is inside its social safety zone
$\mathbf{w}$	Weights vector used in the regression model
$E[\nabla J(\mathbf{w})^2]_t$	Expected value of the gradient

η	Learning rate used in the optimization process of the weights
ϵ	Smoothing term which avoids the division by zero

To my parents, Luis Eduardo and Evangelina, my sisters Nena and Vicky, my brother Junior and my nephews Junior and Diego, who are always in my thoughts.

Chapter 1

Introduction

During this work, we are going to find many topics related with autonomous robotics field, but there is one in which we are going to be more devoted, navigation. The importance of this topic in robotics is crucial since it involves the security of the robots but also the environment in which the robot is navigating. There are many environments in which navigation could be dangerous, for example in social environments involving pedestrians, but also industrial environment in which not robust robot navigation could lead to unprofitable results, wasting time or money. Here, we are going to focus on social environments and robot's mobility, in which we have pedestrians walking around to their destinations, the presence of obstacles and the presence of a flying robot, also called drone. It is required to improve algorithms in order to ensure a correct and proper navigation of the drone. As peoples safety is the main priority, we have to ensure the drone will respect the personal space, will navigate carefully and will interact with the dynamic environment all the time, trying to provide a comfortable perception to the pedestrians around it.

In the literature, we can find a lot of material talking about collision avoidance techniques and robot's path planning with many variations and strategies adapted to different kind of problems, but this time we do not want only to avoid obstacles, we are also considering also a social space. We are going to implement and to improve the model called "Social Force Model", which considers the personal space of pedestrians and robots, making them part of this social environment, and allowing the robots moving as entities walking with pedestrians. This model was previously implemented and extended in [19], thus, here we present the 3D model, following the algorithm enhanced in this work.

This model states that pedestrians exert a force over other pedestrians making them change the direction they are taking. These forces are perceived more as the "motivation" of pedestrians to move in a certain direction, and this motivation also can be influenced by the presence of obstacles, but in a different way. This means that the forces depend on different functions and different parameters. Additionally, the robot exerts a force over pedestrians, making them change the direction depending on how close the drone is located.

In order to compare this method, we also implemented a standard technique widely used in robot navigation, the artificial potential field approach.

This technique provides attractive and repulsive forces to the robot, in order to make it follow a certain path to the final destination. As we said, this technique is widely used, and for that there are a lot of variations, and in this case, we inspired our model based on [54]; we took some improvements they provided and adapted them to our model. Robot's navigation is a crucial task, and we present here two methods adapted to our problem.

This work is more focused on making the drone more socially acceptable to the pedestrians around it, but also the drone has the task to accompany a human, who in this work we will call him the "main human". The drone will be with the human, trying to maintain a certain distance and respecting other pedestrian's distance, but the drone has to know the main human direction and for that reason we introduce an important topic, forecasting. One of the topics implicitly related with the robot's autonomy is the forecasting capacity. This feature, makes the robot able to predict outcomes and be prepared for them, giving it truly autonomy. Human's path prediction is also widely studied, and we can find a lot of approaches with great performance, for example, stochastic prediction, neural networks, and maximum entropy based techniques [44], [6], [55]. In this work we use a linear regression model, which helps the robot to predict the human path in a very accurate way, letting it know the direction of the human, and making it walk with him with only knowing the historical position of the main pedestrian.

Giving tasks to robots each time is easier and hold a good promise for autonomously carrying out navigation, each time with the release of more advanced and complex algorithms we can be sure that more day-by-day tasks will be carried out autonomously by robots.

1.1 Motivation

In the past years, we have seen how the presence of the robots in our life has grown, from the everyday tasks until the industrial issues, developing tasks each time more complex.

The idea of creating new robots which can navigate autonomously between people has been developed by multiple motivations, some of them to guide people, as assistants, or just as mobile information modules. The main motivation of this work is as follows, robots which are able to move with a human-like behavior in a social environment. Therefore the social robots need to fulfill extra requirements, this is, take into account other kind of factors and constraints, for instance, people comfort, when a robot is close of them. The idea of bringing robots to the social environment, creates new tasks which can serve to people, in an intelligent and autonomous way, and can be tangible.

We are not so far of the reality in which we can find robots around us, but for sure, there are a lot of features the robot must improve, to make it appropriate to navigate around people. The need of safety protocols, a clean and objective existence purpose of the robot with the assignment of

serving people, need to be evaluated in order to achieve a proper social behavior.

1.2 Objectives

In this work, we present and test a model, which integrates different techniques in order to make it robust while navigating. As we said, the main model to test is called 3D Aerial Social Social Force Model, which born from the Extended Social Force Model, and it states that the pedestrian behavior can be modeled as individual particle inspired in the fluid kinetics model, the details are presented below. The main objective is to implement the 3D Aerial Social Social Force Model, taking into account, pedestrians, obstacles and a drone, and compare the final results with another standard technique called Artificial Potential Fields.

We aim to prove the capability of the drone, to navigate safely between pedestrians using ESFM, taking into account several number of pedestrians and obstacles, supporting the hypothesis that our model provides the drone with the ability to navigate socially better than the compared approach.

We show the successful implementation of the ESFM in real-life experiments, providing an acceptable social behavior, making it able to navigate in a human-like way, accompanying the main human and keeping a certain distance along the experimentation always.

1.3 Main Contributions

The main contributions of the present work are threefold. First, we based our research on [19], in which the ESFM was implemented in a land-based mobile robot. Here, we enhance the ESFM to be used with a flying robot, adding a third dimension z with respect to [19]. This is not trivial since we had to recalculate the personal space taking into account the extra dimension, and we also needed to modify the initial algorithm to make the robot capable of accompanying the person inside a high range. We also provide a simple and robust strategy to predict the main pedestrian position, considering the past positions differentials within a fixed window. And finally, we validate our method through an extensive set of simulations and real-life experiments. At the end of the discussion, we point out the strengths and weaknesses of the approaches, giving a base for the next steps to be implemented and points to reinforce.

One article was written from this work, in which we exposed the overall model and the results from the real-life experiments. The article was sent to the International Conference on Robotics and Automation. Moreover, we are currently working on some extensions to publish our work in a journal.

In following chapters, the details of the entire work are described. In chapter 3, the hardware and software of the used tools to develop and test the algorithms are introduced. In chapter 4, we show an outlook to the bases of SFM, ESFM and APF approaches. Chapter 5 describes the method

and the different strategies we used to make the model; here we include the details of the ESFM, the linear regression implementation and gradient descent optimization. Finally, in Chapter 6, we present the set of virtual simulation and the real-life experiments.

1.3.1 Derived Publications

The derived publications of this work (* indicates equal contribution):

1. **Luis Garza-Elizondo***, A. Garrell*, M. Villamizar, F. Herrero and A. Sanfeliu. 3D Aerial Social Force Model to Accompany People. IEEE International Conference on Robotics and Automation. May-june 2017 (submitted).

2. **A. Garrell***, Luis Garza-Elizondo*, M. Villamizar, F. Herrero and A. Sanfeliu. 3D Aerial Social Force Model evaluation in crowded environments. IEEE Transactions on Robotics. (working on).

Chapter 2

State of the Art

In this chapter, we introduce the state of the art involving drones interacting with people, social navigation methods, obstacle avoidance, robot companion in urban environment, and human motion prediction. One of objectives of this chapter is to let the reader know a general idea of how this area has been evolved and to highlight the different efforts to make the robot navigation safe and to put them into the social environments.

2.1 Human-Drone Interaction

Nowadays, Human-Robot Interaction field has been widely studied during the past years, bringing us many advances in terms of perception, motion, emotion interpretation. Moreover, many of these interactions have been developed with land-based robots, static and mobile ones, and for that reason, the area of human-drone interaction is increasing and new approaches and techniques should be developed since there are other limitations and characteristics to take into account [7]. Relatively new, this area brings with it new challenges, and new advances and techniques. Some new methods introduce the interaction through the image processing of human gestures [40], using on board depth cameras in indoor environments, in which the person can control the robot through gestures. Another work with similar ideas is presented in [38], where multiple UAVs using locally on-board cameras are controlled via gestures through machine vision techniques, motivated by the idea that gestures are a natural way of communication and relatively easy to recognize through image processing. Other approach [48] also includes gestures movements to indicate instructions to the flying robot, inspired in the interaction between the human and birds in the falconeering activity.

Furthermore, other researches are motivated by the idea that adding an emotional component is part of the success of the drone in terms of acceptability, and they explore the topic of encoding emotions in drones path movements [8]. Moreover, a new platform called Assistive free-flyers (AFFs), is emerging, where small drones take part, with the intention of aid people in many different ways in indoor and outdoor environments [51], with an intention communication through the design of natural flight motions.

One of the first papers addressing the navigation of drones with people was presented in [22]. In this work, the aerial robot is used to extend human abilities. For example, to enhance the field of view of a person in

order to report accidents or anomalies. In [47] a drone-base flyer acting as a personal companion is proposed to support people in emergency situations. To enable the interaction between drones and humans, it is required to provide a communication channel, for instance, a remote, a phone, or gestural control [38]. Nevertheless, there are few feedback techniques for HDI. Some recent works, studied the head movement, and the propeller noise in order to present emotional states [43].

Studies of Human-drone interaction is becoming more and more popular in the last years, since it is inevitable the presence of flying robots in the next years. In this work, we are interested in obtaining a navigation suitable to accompany people in a comfortable manner, in order to allow flying robots to navigate with people in indoor and outdoor environments.

2.2 Robot Collision avoidance

Different techniques have been used to face this topic in robotics. It is important to say that we are more interested in aerial vehicles than in ground robots, however some of the techniques presented in this work can be applied to both kind of robots. In last years, many researchers used artificial potential field variations, as in [54] [31] [14], where modifications to the traditional method are made in order to enhance the outcome. APF is a well-know technique based on potential fields to create forces which are exerted over the robot, to make it avoid the source of these potential fields, usually obstacles and objects. This method has proved to have very good results in the avoidance task, however, there is one issue who is our concern, and we will keep talking about that along all the project, which is safety and reliability.

Others, as [18], presented a collision avoidance method with Mixed-integer linear programming (MILP) with geometric method and airspace discretization. In [11], researchers employ partial order techniques to guarantee collision avoidance between vehicles. Moreover, there are other techniques which tackle the avoidance and localization task from a probabilistic point of view [13]. In [5], authors introduce a probabilistic perspective of the Inevitable Collision States (ICS), states in which there is no feasible trajectory to avoid collision and in which they add uncertainty that prevails in real world situations. The task to avoid dynamic obstacles is faced in many forms with different results. During this project we talk not only about the avoidance task of the robots, more specifically we are interested in how people feel having a drone flying around them, for that we measure the comfort level of the persons when walking beside the drone. Other methods have appeared which tackle the path planning problem through neural networks combined with Q-learning, giving very good results in land-based robots with multiple obstacles and where global information is available all the time [12]. In [50] they tackle the collision avoidance problem with the curvature-velocity method, adding constraints derived from pyshical limitations on the robots velocities and accelerations, trying to find a point in

the space that satisfies all these constraints.

Some other methods to avoid obstacles are also inspired by living organisms, ensuring that complex control systems are not necessary at the time to react to some obstacle or to guide the moves of a body, as in [29], they define how sensory input arrays can specify motion-taus through tau-coupling. Other kind of techniques assume other agents in the environment have also a similar mechanism to avoid collisions, such in the case of [33] in which the collision avoidance problem is address using Reciprocal Velocity Obstacle, a novel technique based on the well-known velocity obstacle concept but with some modifications to fix the oscillation problem.

2.3 Social Robot Navigation

It is more latent the presence of robots in our day-by-day, maybe first as software agents, with some kind of intelligence, as we have in our cell-phones, in our fridges, microwave oven, etc. Social life will be surrounded by all kind of robots serving us in many different ways. Many approaches have appeared in order to bring this era as soon as possible. For this task there are some issues to face, as the autonomy of the robot, the intelligence, the good perception and the good interaction between the robot and their environment. Some other methods have been modeled to make the robot able to navigate in a social way, the task in which we are interested at, as in the case of [27], which specifically face the navigation issue among humans. In the last decade, human-robot interaction has become a main research field, that involves many areas, in order to bring the robot the ability to interact with the people, including for instance, learning capabilities, perception task, reasoning , manipulation and navigation ability.

Unlike the past years, we can see how robots are moving from industrial areas to daily applications, many of them being carried in our homes. Talking about robots implies to talk about a physical body (in most of the cases), and social robots mean those one who are developing tasks among people in social environments, and for that, it is necessary the navigation skill, not like the traditional one that we were used to listen or work, but a Human-aware navigation which takes into account social aspects, and new ways of measuring and control, for example the comfort, the acceptance of the robots among pedestrians, etc. There is more and more interest in this area since the past 10 years [28], in which we became to realize the presence of robots in our daily life is not that far as we thought.

Along the social issue, there are many issues to have in mind when we think about social robots, for instance, the fact that social environments are prepared just for humans and not for robots, for example stairs, ramps, visual signal and signal lights. These are also the kind of challenges the robot would face in urban environments, to adapt,to avoid and to follow the protocols the environment dictates [4]. Many efforts have been done by researchers to accomplish a safe and human-like navigation in social environments. In [32] they were capable of developing a mobile robot which

navigates in crowd environments based on interaction and local perception capabilities. Some researches include environmental perception in where information from different sensors is merged to provide the robot a more complete perception of its environment [20]. In [**Garrell2009**], a Discrete time motion model is used to represent robot's and person's motion, for people guidance in urban areas.

In [35], the authors highlight the importance of robust and reliable systems for safe navigation. Moreover in [37], it is proposed a solution to estimate the robot position in a map, when no initial guess of its initial position is given. Moreover, in [53] researchers deal with the problem of dynamic motion problem in unknown environments applying VCS-based motion planning methods.

There is one important topic to fulfill and is the navigation part in urban environments, which has been faced in many different ways. In the last years important advances has been made in robot navigation in indoor environments, where the robot has to interact with the people surrounding it, detecting objects and measuring distances within the environment, using different kind of techniques as QR codes in [30] where they highlight also the importance of cost-effective systems to fulfill the increasing demand of robots in the environment. Also [10] remarks the achievement of running the algorithms in low cost embedded systems, this time the Raspberry pi system.

Researches in urban environment navigation have highlighted three important things to pursue in this topic: the comfort, the naturalness and the sociability of the robot in urban environments [28]. Urban environments also constrain the robot motion, and subject it to a large list of social protocols all the time. The social motion of robots has to model and respect these protocols and the cultural norms in the society, and improve implicit interactions [28].

2.4 Human motion prediction

For robot navigation, prediction is one of the crucial tasks to develop, mainly because here, navigation is taken as a communication source. For that, prediction is important because both robot and human influence each other in terms of motion [28].Drone's movements can infer on people, and also the drone can understand which action the human would take depending on human's motion. Other methods take into account human movement prediction in order to avoid them, for instance in [44] they make a stochastic prediction, to estimate danger situation.

In a recent method, [6], they have opted to use neural networks architecture to cognitive navigation, which predicts possible human-robot collision, through generalizing the cognitive maps (which contain critical information for planning movements in space), to dynamic environments, this method is called Prediction-for-CompAction (PfCA). In [49], a technique is proposed based on a combination of Beam Curvature Method (BCM), Lane

Curvature Method (LCM) with prediction capabilities, which give them the possibility of avoiding moving objects in dynamic environments. Other methods as [**Garrell2009**] use particle filters to predict human movements, and use the Discrete time Motion model proposed. [52] also made use of the SFM and implement a predictor of the pedestrian trajectories which can tell other pedestrians the place and time of the next collision in order to avoid it.

As we said before, novel methods which can predict accurately how the human behaves allow an efficient navigation, and are the key to develop intelligent systems capable of being with humans in our social environment. Common approaches for predicting trajectories are using tracing filters as the Kalman filter, which assumes a normal Gaussian distribution with growing uncertainty, to predict future positions[55]. Moreover recent techniques involve the principle of maximum entropy, which yields a soft-maximum version of Markov decision processes that accounts for decision uncertainty [55]. Many techniques have been used to predict many kind of behaviors, however, in the social navigation field, motion prediction has not been that popular, many works make the call to keep doing research and to keep the interest over this area [28].

Chapter 3

Hardware and Software

3.1 Hardware

In this chapter we proceed to describe the tools we did use, during the real and synthetic experiments. In general, experimenting with real robot can be tough, especially because all the details has to be taken into account before letting the program run. For that reason, in those cases it is important to use any kind of simulator that can provide you at least a little vision of how the system might behave with the current firmware. In our case, we did use a simulator to test our navigation algorithm, before putting it into the real drone, and fortunately we first found some drawbacks which we could fix before the real tests.

In this case we use real drones and also drone simulators running with Robotic Operating System, the details are introduced below. We provide documentation and links in case the readers wanted to know deepest aspects of some particular point.

3.1.1 ONA Robot

The project was planned to be tested using the ONA robot, an unmanned aerial vehicle with a vertical take off and landing capacity. The platform can fly in autonomous way or can be controlled remotely. ONA robot has different sensors to carry on tasks related with human-robot interaction, which is one of the main objectives of the overall system to build in this project. ONA Robot is property of the *Institut de Robotica i Informatica Industrial*.

Structure and system details

The structure was made with carbon fiber with 1mm of thickness. The arms are made with a combination of carbon fiber, PVC and an epoxi matrix. The overall structure is complemented with little components printed in ABS plastic and the protection structure with expanded polypropylene.

The drone uses two lithium polymer batteries, each of them with 11.1 volts and 5300 mAh capacity. The drone is also equipped with the Pixhawk autopilot system, which gives the drone a complete autonomy, and the capacity of doing programmed missions using the GPS.

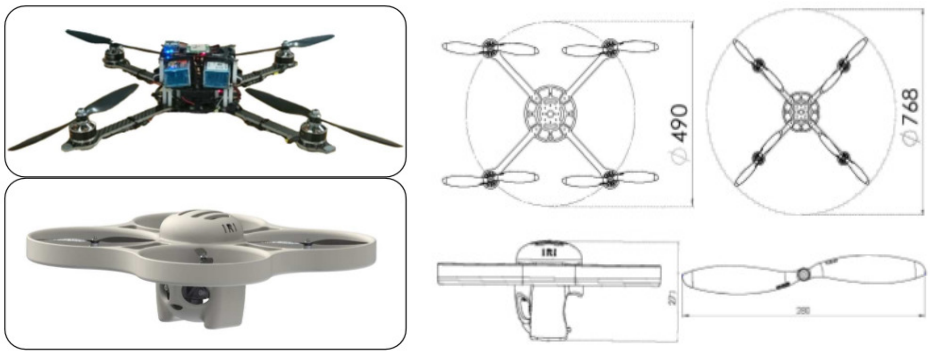


FIGURE 3.1: **ONA's Robot:** picture and Schematics of the ONA robot. ONA is property of the IRI lab, and was built for social purposes

At the end, due to the short time to do experiments, and due to some delays in the delivery of the ONA robot, we have tested the algorithm only with the AR.Drone which is going to be described in the next subsection. The two of them have very good features and are very good candidates to test our model, nevertheless, the ONA robot was built specifically to do tasks related with the human-robot interaction and social purposes.

3.1.2 AR.Drone 2.0

During the experiments, we have used the AR.Drone 2.0, a remote controlled flying quadcopter helicopter built by Parrot, a French company based in Paris. With impressive features, the quadcopter demonstrated to be a good choice to test our algorithms in a short-time setup. Some of the features included in the AR.Drone, which also can be consulted at the official webpage, are:

- Intuitive touch and tilt flight controls.
- Live video streaming
- Video recording and photo shooting
- Updated euler angles of the AR Drone
- Embedded tag detection for augmented reality games.

According to the official documentation, which reference you can find in the bibliography [42], the mechanical structure comprises four rotors attached to the four ends of a crossing to which the battery and the RF hardware are attached.

Each pair of opposite rotors is turning the same way. One pair is turning clockwise and the other anti-clockwise.

Different drone movements can be achieved varying each rotor pair speed. For example, varying left and right rotors speeds the opposite way yields roll movements, varying front and rear rotors speeds yields pitch

movements and so on. In the next picture, taken from the official manual, there is an illustration of how is yielded each kind of movement the drone can achieve.

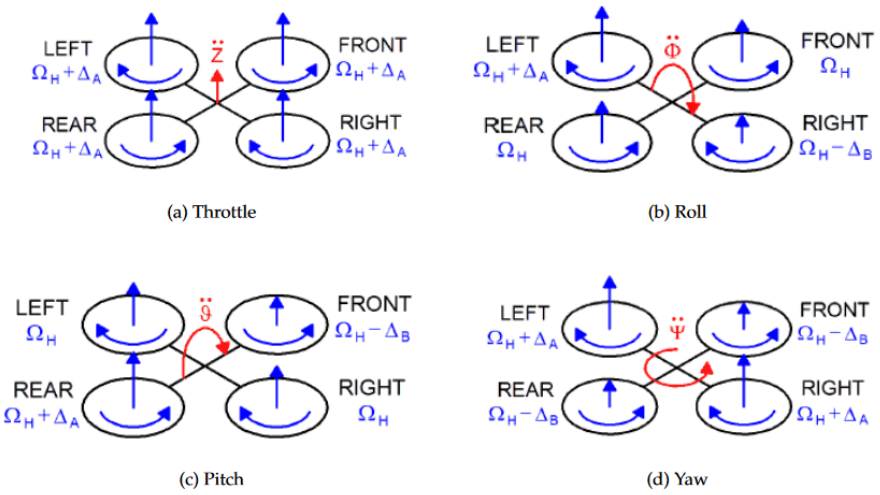


FIGURE 3.2: **Drone movements:** Picture depicting different movements the AR.Drone can do.

Maneuvers are obtained by changing the angle of the pitch, yaw and roll of the quadcopter. These terms can be better appreciated in the next figure, with the x, y and z references represented in a plane.

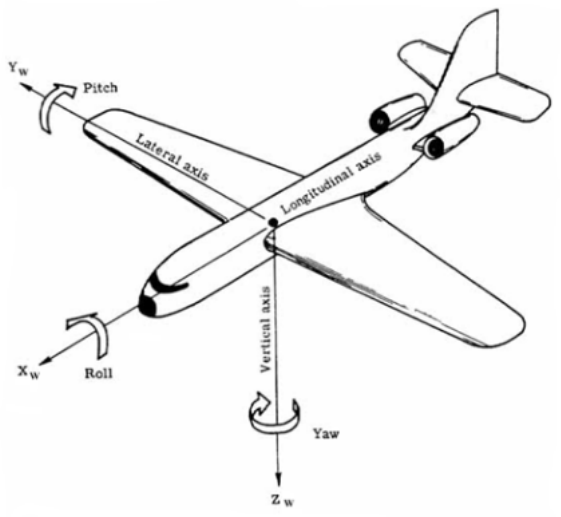


FIGURE 3.3: **Aircraft principal axes:** Pitch , Roll, and Yaw movements represented with a plane.

Engines, Batteries and Sensors

Some important features to mention about the drone are the brushless engines with three phases which are controlled by a microcontroller, which

automatically detects the engines plugged in and set up the correct engine control. The drone is also able to detect if any engine is stopped or blocked, and in this case the protection system stops all engines immediately to prevent repeated shocks.

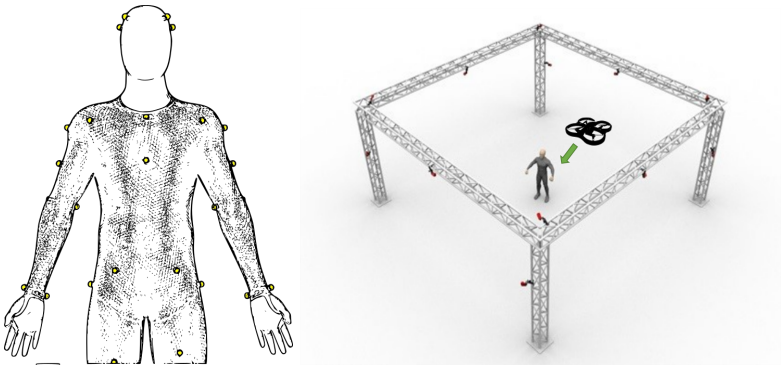
The batteries we used were 1000 mAh, 11.1 volts LiPo batteries. When the drone detects a low level of battery, it first automatically sends a warning message and then, if the level has reached a critical level, the whole system is shut down. The AR.Drone 2.0 has many motion sensors as the inertial measurement unit, the ultrasound telemeter, a 3 axis magnetometer and a pressure sensor. All of the sensors we mentioned before are fundamental for a proper control and navigation of the AR.Drone, however we omit technical details, to focus only for now on comment the main tools and some important features and a brief description of them, which could help us to have a better image of the main work tool, the AR.Drone.

Along with the drone, we made use of other important tool, called optitrack. Optitrack helps us to know the position with a high level of accuracy, using a set of special cameras, in a certain position within a delimited movement space. These indoor experiments were carried out with the aim of better observing the forces being applied on the drone by a main human. Real experiments using optitrack only take into account one human, who is the one the drone is following, and of course the drone.

3.1.3 Optitrack

Here, we explain briefly some important features of the Optitrack tool and what was its role during the experiments. It is important to know that you can find more details in the website and official documentation[1], technical aspects related with this tool, along with a wide variety of products and different configurations of this tool the user can choose depending on their needs.

Optitrack Motion Capture system is a system created by NaturalPoint Inc., company which provides different optical tracking products, which are widely used in digital animation, robotics, virtual reality and many other branches. In this case Optitrack was used to analyze an indoor environment to track the human and the drone position in a previously defined space. Knowing the position, Optitrack allow us to calculate the forces interacting between the person and the robot and to see the behavior of our model in real time. Both human and drone were using markers in the body in order to let the software detect the exact position of each of them.



(A) Optitrack markers in a human body.

(B) Drone and Human in an indoor environment monitored by the Optitrack cams.

FIGURE 3.4: **Tracking markers and Optitrack workspace:** (a) Representation of the markers that can be used in a human body, as well as other kind of objects like the drone, (b) An indoor environment with Optitrack depicting the experiments we did.

3.2 Software

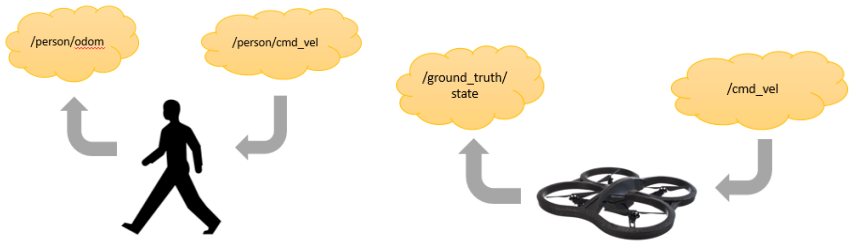
3.2.1 Robot Operating System

This project was mainly developed using the Robot Operating System, a framework widely used to build the internal software structure for robots, with a lot of features which help the developer to debug and visualize through simulations, algorithms and code made in python and/or c++. This framework provides hardware abstraction making it useful at the time of migrating the code to other kind of robots with different hardware configurations. This operating system is graph based, in which the processing of messages is done in the nodes, and these nodes are able to communicate through different kind of messages to different sensors, actuators, and also other nodes.

Basically, ROS operates with one node master which is in charge of the coordination, and other secondary nodes which are running instances (same program can run different times at the same time) of ROS programs and can be set and personalized by user for different purposes. The nodes are easily set up, and can handle situations as for example a "node for processing the output of sensor A", or maybe "node which send the current position". This kind of behavior can be seen as a blackboard communication protocol, in which everyone subscribed to the same topic, is able to see what is being published in this topic.

ROS allows us to create generic packages which can be run in any computer system with ROS installed, and reuse our already created algorithms with only adjusting a few configurations. This can help us to save a lot of time and also save money.

Along with ROS, we made use of Gazebo, which provided us the interface to simulate robots and other kind of objects. In this case we simulated a drone and pedestrians in a 3D space. As we said before, this kind of tools make easier to test code and see an approximated behavior of what we should expect in real life, due to the capacity of simulating complex behaviors in indoor and outdoor environments. During the simulation experiments we use the tum_simulator package which allows us to simulate the AR.Drone. In addition, we use an object which looks like a human, which was provided by the IRI object's library. Both objects Drone and human, were subscribed to certain topics which control the motion, and publish in others the information about the position, velocity, orientation etc.
Within the ROS environment we made use of another tool which help us to visualize in a 3D space the behavior of the robot and some other useful information, for instance axes, transformations and markers, also the forces interacting between the characters during the experiments; this graphical tool is called RViz.



(A) Humans's information flow chart.

(B) Drone's information flow chart.

FIGURE 3.5: **Rostopics of the human and the drone:** Diagram Representative of the information published by the person and the drone.

Tum Simulator

Tum simulator is a ROS package written by Hongrong Huang and Juergen Sturm at the Technical University of Munich, and basically it simulates the AR.Drone in the ROS environment using gazebo simulator.

The package allows us to simulate and also manipulate in real life experiments the drone, both options are available and is up to the user to configure and adapt the simulator depending also on the version of the AR.Drone and many other parameters. We can find the functionality diagram in the official website, which can provide us a general idea of how it works in both modalities, real life and simulated experiments [2].

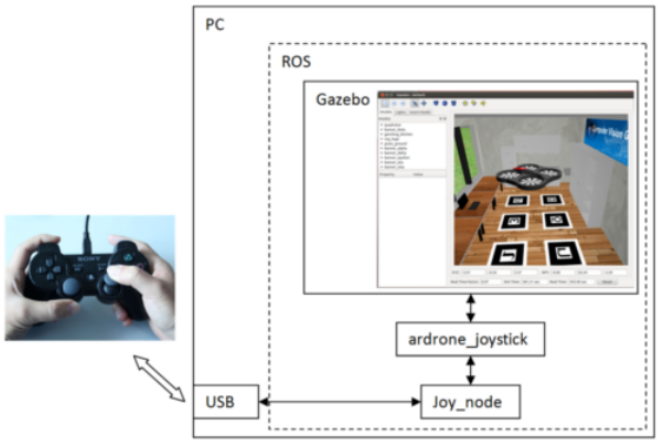


FIGURE 3.6: **tum_simulator and simulation environment:** tum_simulator configuration diagram to perform simulation using a joystick [2]

Moreover, we have the gazebo simulator in which we can use a lot of objects and obstacles, and simulate real environments, just as real life or at least the most similar possible. We made use of this simulator because the facilities in the installation process, also the experience of some of us using this simulator, and the great results we have had during different kind of experiments. Images were taken from official webpages, to depict the general idea of the flow work structure of each configuration.

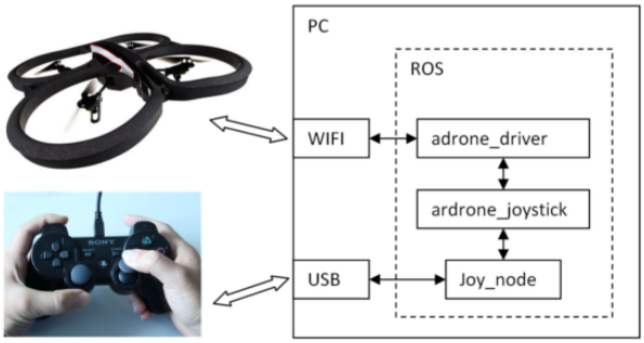


FIGURE 3.7: **tum_simulator and real-life drone's control:** real-life experiments tum_simulator configuration diagram using joystick[2].

Some images of tum simulator are in the next picture, showing the drone and a couple of buildings around the robot, the simulation environment helps us to draw very similar scenarios and experiments , which help us to save money and time, in case something could go wrong. Later we will see real-life experiments, for now we only describe in a general view the tool, and the need of having a simulator, and the advantages we have seen during the project.

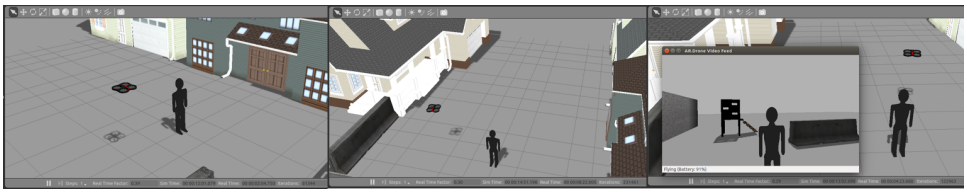


FIGURE 3.8: **tum_simulator and Gazebo:** tum_simulator from different perspectives. A scenario was built in gazebo, were a person, a drone, and some houses are shown. The last picture shows the keyboard controller window from the ardrone_tutorials.

Processing

During the project we made use of the Integrated Development Environment *Processing*, an open source computer programming language built mainly for art and visual design purposes. The IDE processing project started in 2001, by Casey Reas and Benjamin Fry, both old members of the Aesthetics and Computation Group at the MIT Media Lab. Now *Processing* is a growing open source community which has built many kind of tools and libraries to extend the initial functionality. *Processing* allows us to simulate our model in large scales, although we were not able to simulate the movements up and down of the robot in the z coordinate because we used a 2D physics library, we were able to calculate the position and velocity in this axis, and to take it into account.

With the help of Processing we created different environments with different number of static obstacles and we simulate the behavior of pedestrians walking around trying to reach a final position (goal), and a "main" human being accompanied by a drone. With these environments we ran several experiments with different configurations, in order to create files with the information, and then measure the performance of the methods being tested and which we describe later in the next chapters.

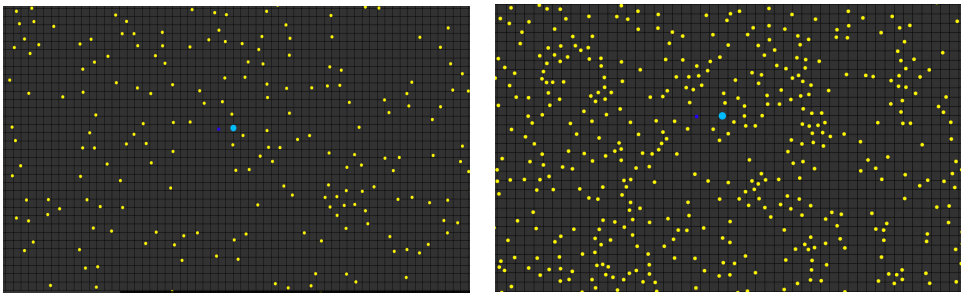


FIGURE 3.9: **Processing environment:** processing was used to test the model , and also run simulations with large number of pedestrians and obstacles.

Chapter 4

Social Navigation

4.1 Introduction

Nowadays there are many efforts to make a robot navigate safely, avoiding dynamic obstacles in crowded environments. It is not a trivial task since also involves topics as robot perception, human-robot cooperation and robot localization, working together with the navigation mechanisms, all in a synchronized way. Many techniques have been developed and integrated to carry out this difficult task.

In this work we focused on robot's navigation, which implies robot mobility through a 3D physical space, trying to avoid humans and objects, and trying to keep a certain distance from the main human in order to consider the robot as an accompanier. Thus, we implemented the extended Social Force Model which is based on a Boltzmann-like gas-kinetic model [23] also explained in [24] and [25]. It states that the movement of pedestrians can be modeled by a fluid dynamic model, and it can give a description for the collective movement of the pedestrians.

Moreover, another tested technique was the Artificial Potential Fields approach, which is widely used in the robotics field, which also involves a similar idea, repulsive forces exerted by obstacles and attractive forces trying to take it to its final destination. As this technique has been studied for many years, it has many variations to adapt the algorithm to different circumstances and face different drawbacks, for example physical constraints on velocity limitations. Here, we took the standard Artificial Potential Field with some modifications taken from [54], which help us to create an uncertainty zone around the point which is exerting a repulsive force. We adapted some parts of the APF algorithm proposed by [54] in order to take some features of this last paper, and we improve the performance of the robot's path planning.

Next, a description of each method, Social Force Model and Artificial Potential Fields Method, and the main ideas of each of them are presented.

4.2 Artificial Potential Fields

Artificial Potential fields is an approach that has been extensively used for obstacle avoidance for single mobile robots, multiple mobile robots, and moving obstacles . In the potential field method, an artificial potential field

is assigned to the area where a robot is moving [14].

Basically the APF method create artificial fields in the environment, these fields are composed by attractive fields and repulsive fields. In this case the obstacles exert a repulsive force to the robot, meanwhile the robot's final destination which we call the "goal" is applying an attractive force to the robot. The final path of the robot is calculated by the gradient of the total artificial potential.
The potential function is defined as the resultant of the all the attractive and repulsive fields, as functions of the drone's position denoted as $\mathbf{P}$:

$$\mathbf{U_{total}}(\mathbf{P}) = \mathbf{U_{att}}(\mathbf{P}) + \mathbf{U_{rep}}(\mathbf{P}) \tag{4.1}$$

The negative gradient of the potential function is defined as an artificial force which is the steepest descent direction for guiding robot to final point [31]. Therefore the attractive force applied to the robot is the negative gradient of the attractive force and the repulsive force is the negative gradient of the repulsive potential.

$$\mathbf{F}(\mathbf{P}) = -\nabla \mathbf{U}(\mathbf{P}) = -\nabla \mathbf{U_{att}}(\mathbf{P}) - \nabla \mathbf{U_{rep}}(\mathbf{P}) \tag{4.2}$$

$$\mathbf{F_{total}} = \mathbf{F_{att}} + \mathbf{F_{rep}} \tag{4.3}$$

In a normal situation, taking into account more than one obstacle, we can rewrite the total force, as next, where n is the number of obstacles.

$$\mathbf{F}(\mathbf{P}) = \mathbf{F_{att}}(\mathbf{P}) + \sum_{i=1}^{n} \mathbf{F_{rep}}(\mathbf{P}) \tag{4.4}$$

4.2.1 Attraction Force

The attractive potential exerted by the goal , is defined as :

$$\mathbf{U_{att}}(\mathbf{P}) = \frac{1}{2} k_{apf} (\mathbf{P} - \mathbf{P_g})^2 \tag{4.5}$$

In which k_{apf} is a positive coefficient, P is the robot location and P_g is the goal location. The negative gradient of the attractive potential is :

$$\mathbf{F_{att}}(\mathbf{P}) = -\nabla \mathbf{U_{att}}(\mathbf{P}) = -k_{apf}(\mathbf{P} - \mathbf{P_g}) \tag{4.6}$$

Then, an attraction force that is pointing to the final goal, and the final components of this force can be written as:

$$\begin{aligned} F_{att_x}(P) &= -k_{apf}(x - x_g) \\ F_{att_y}(P) &= -k_{apf}(y - y_g) \\ F_{att_z}(P) &= -k_{apf}(z - z_g) \end{aligned} \tag{4.7}$$

4.2.2 Repulsion Force

With the same idea, robots have to take distance from obstacles, but in this case, to calculate the repulsion force we take into account the uncertainty zone related with the robot's and object's position. Assuming this, to define a position uncertainty zone within which the true obstacle position lies [54], we consider a cylindrical area with radius r and height h, and then estimate $\rho(P, P_{ob})$ which is the distance from the robot to the obstacle's intersection uncertainty zone. The uncertainty zone is defined as:

$$\rho(P, P_{ob}) = \begin{cases} d - \frac{r}{\cos\alpha} & 0 <\| \alpha \|\leq \arctan(\frac{h}{2r}) \\ d - \frac{h}{2\sin\alpha} & \arctan(\frac{h}{2r}) <\| \alpha \|< \frac{\Pi}{2} \end{cases} \tag{4.8}$$

And is illustrated in the next figure:

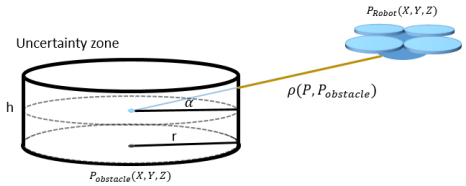


FIGURE 4.1: **Uncertainty zone:** The Uncertainty zone is represented as a cylindrical area with radius r and height h.

The repulsive potential field can be then defined as:

$$\mathbf{U_{rep}}(\mathbf{P}) = \begin{cases} 0 & \rho(P, P_{ob}) > \rho_0 \\ \frac{1}{2}\eta\left[\frac{1}{\rho(P,P_{ob})} - \frac{1}{\rho_0}\right]^2 & 0 < \rho(P, P_{ob}) \leq \rho_0 \\ \infty & \rho(P, P_{ob}) \leq 0 \end{cases} \tag{4.9}$$

Where η is a positive potential factor, P_{ob} is the closest obstacle to the robot and $\rho(P, P_{ob})$ is the distance between the robot and the intersection with the obstacle's uncertainty zone, ρ_0 is the largest impact distance factor, also called scope of the repulsive potential, and has to be greater than 0. The object will not be able to affect the robot's path, if the distance between the robot and the object is greater than ρ_0.
Again we calculate the negative gradient of the repulsive potential function to find the repulsive force.

$$\mathbf{F_{rep}}(\mathbf{P}) = -\nabla \mathbf{U_{rep}}(\mathbf{P}) \tag{4.10}$$

$$\mathbf{F_{rep}}(\mathbf{P}) = \begin{cases} 0 & \rho(P, P_{ob}) > \rho_0 \\ \eta\left[\frac{1}{\rho(P,P_{ob})} - \frac{1}{\rho_0}\right]\left[\frac{1}{\rho(P,P_{ob})^2}\right]\nabla\rho(P, P_{ob}) & 0 < \rho(P, P_{ob}) \leq \rho_0 \\ \infty & \rho(P, P_{ob}) \leq 0 \end{cases} \tag{4.11}$$

4.3 Social Force Model

Since 1950s, pedestrian behavior has been modeled in different ways, starting with an outlook focused more on the dynamics of macroscopic quantities (densities and fluxes), and treated more similar as gases or fluids. Later, researchers started to shift the attention to a more microscopic description, in which pedestrian motion is described more as an individual particle rather than a whole.

Some years ago, Dirk Helbing presented a model which suggests that the motion of pedestrians can be described by social forces, not directly exerted by the pedestrians' personal environment, but a measure for the internal motivations of the individuals to perform certain movements [26]. Different models of pedestrian behavior have been published during the last decades, with different principles and motivations, but it was in 1970 when Henderson had compared measurements of the pedestrian dynamics with Navier-Stokes equations with considerable success [26]. His work was later improved and founded mathematically by Dirk Helbing in 1990, based on a pedestrian specific gas-kinetic model [23].

We might think that human behavior in crowd situations is chaotic and without a chance to be described by any rules or equations, and certainly, in real complex situations it is, but in simpler scenarios, since every move of the human is conducted by his experience, people usually move motivated by automatic reactions previously learned, and for that the model states that it is possible to put these rules into equations of motion. We have to take into account that, although the social force represents an effect exerted by the environment on the human, the social force model defines rather a quantity which describes the "motivation" of the persons to move through a certain direction, a pedestrian's motivation to act. We can say at the end that pedestrians act as if they were subject to external forces.

4.3.1 Extended Social Force Model

Social Force model take into account pedestrian interactions and final destinations by defining a summation of existing forces which determine the final force, and therefore, people's trajectories. The model details the motion which can be expressed through a function depending on the pedestrian's relative and absolute positions and velocities [19] . In this case, we also introduce a robot in the social environment, which leads us to make use of the Extended Social-Force Model , which take into account the interaction between people and robots, and also the interaction between robots and obstacles. This model was inspired by Helbing's and Zanlungo's work[26] [52] .

Next, we introduce and define the main effects that determine the total force over a pedestrian. Formally, this approach treats each pedestrian p_i with mass m_{pi} as a particle, which new position and velocity is ruled by Newtonian mechanics. The human was always supposed to be walking in z = 0 and only wanted to reach a certain position at the height equals to 0.

$$\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix}_t + \begin{bmatrix} \Delta t^2 & 0 & 0 \\ 0 & \Delta t^2 & 0 \\ \Delta t & 0 & \Delta t^2 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{4.12}$$

where (x, y) is the position of the person, (v_x, v_y) is the velocity and (a_x, a_y) is the acceleration. The model assumes that a pedestrian p_i with mass m_{pi} tries to move at a certain desired speed v^0_{pi} in a desired direction e_{pi}, this desired direction is defined as:

$$\mathbf{e_{pi}(t)} := \frac{\mathbf{r^k_{pi}} - \mathbf{r_{pi}(t)}}{\parallel \mathbf{r^k_{pi}} - \mathbf{r_{pi}(t)} \parallel} \tag{4.13}$$

where $\mathbf{r_{pi}(t)}$ denotes the actual position of pedestrian p_i at a time t, and $\mathbf{r^k_{pi}}$ is the direction to the k_{th} goal area. If it is the case, in which a pedestrian is not disturbed, this pedestrian will walk into the desired direction $\mathbf{e_{pi}(t)}$ with a certain desired speed v^0_{pi} [26] with a desired velocity $\mathbf{v^0_{pi}}$ = v^0_{pi} * $\mathbf{e_{pi}}$, where the speed is the one that the pedestrian feels more comfortable to walk with.

Thus, the total force can thus be defined as the resultant force, taking into account all the interaction forces applied on the pedestrian, by other pedestrians and objects around him. The basic equation of motion is defined as:

$$\frac{dv_{pi}(t)}{dt} m_{pi} = \mathbf{F_i(t)} \tag{4.14}$$

This describes the force applied over a pedestrian p_i over the time. The resulting force $\mathbf{F_i}$ governs the trajectory of the pedestrian p_i, and can be described as function of the interaction forces involved, this is, the repulsive force conducted by humans and objects, and the force that drives him to the goal.

$$\mathbf{F_i} = \mathbf{f^{goal}_i} + \mathbf{F^{int}_i} \tag{4.15}$$

The interaction force $\mathbf{F_i^{int}}$ is the summation of all the repulsive forces exerted by other pedestrians, objects and robot around him, and is defined as:

$$\mathbf{F_i^{int}} = \sum_{pj \in P} \mathbf{f_{ij}^{int}} + \sum_{o \in O} \mathbf{f_{io}^{int}} + \mathbf{f_{iRj}^{int}} \tag{4.16}$$

Where P is the set of people moving in the environment and O is the set of obstacles. Next it is defined each interaction force individually, and the motivation behind this.

4.3.2 Goal force attraction

This force drives the pedestrian to the goal area. Supposing the human is always trying to reach this point, he will adapt his velocity with a certain relaxation time k_i. This force is defined as :

$$\mathbf{f_i^{goal}} = k_i(\mathbf{v_i^0} - \mathbf{v_i}) \tag{4.17}$$

4.3.3 Human-Human interaction force

The motion of the pedestrian is influenced by other pedestrians. The humans always keep a certain distance from other pedestrians depending on the crowd of the environment and the desired speed v_{pi}^0 [26]. It is normal that people feel uncomfortable with other humans who try to get closer than usual, and for that, this force act in a repulsive way over other pedestrians to keep certain distance. This idea is represented by the next function :

$$\mathbf{f_{ij}^{int}} = Ae^{(\frac{d-d_{ij}}{B})} \frac{\mathbf{r_{ij}(t)}}{d_{ij}(t)} \tag{4.18}$$

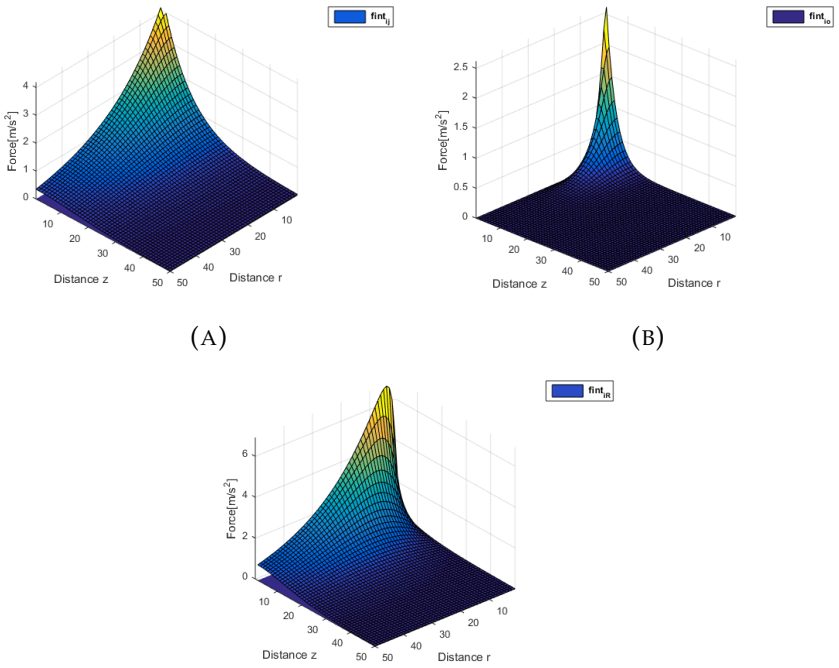
4.3.4 Human-Object interaction force

Humans also want to keep a certain distance from objects when walking, trying to follow an objects-free path to safely reach their goals. Therefore, the objects around the pedestrian exert a force diverting his path, with a repulsive and monotonic decreasing potential U_{io} [26], where the vector $\mathbf{r_{io}}$ is defined as the difference between the location of the person and the location of the obstacle o that is nearest to the pedestrian p_i. This repulsive force is defined as:

$$\mathbf{f_{io}^{int}} = -\nabla_{r_{io}} U_{io}(\parallel \mathbf{r_{io}} \parallel) \tag{4.19}$$

where $U_{io}(\parallel r_{io} \parallel)$ is defined as:

$$U_{io}(\parallel \mathbf{r_{io}} \parallel) = U_{io}^0 e^{\frac{-\|\mathbf{r_{io}}\|}{C}} \tag{4.20}$$



(A) (B)

(C)

FIGURE 4.2: **Repulsive forces :** Repulsive effect of the forces (A)$fint_{ij}$, (B)$fint_{io}$ and (C)$fint_{iR}$ as a function of distance r and distance z

4.3.5 Human-Robot interaction force

Robots also exert a repulsive force over other pedestrians p_i. In the same way, people want to keep a certain distance from other pedestrians and objects, because if they are too close from the robot, they might feel uncomfortable. One of the main contributions of this work was the extension of the study of the pedestrian's and drone's "personal" space, which in the previous work [19] was well studied taking into account a land-based robot interacting with pedestrians.

Now due to the fact we are using a drone, there is a need to extend the proposed robot-human interaction force formula, and add an extra dimension in which the force is also going to be projected. The main modifications are exposed in the anisotropic factor, where the angle Theta is added, and represent the angle formed between $\mathbf{v^0_{pi}}$ and the distance $\mathbf{r_{iR}}$ between the pedestrian *i* and the robot *R* into the *r* and *z* space.

The repulsive effect the robot causes over pedestrians is defined in [19] as:

$$\mathbf{f^{int}_{iR}} = A_{iR} e^{(\frac{d_R - d_{iR}}{B_{iR}})} \frac{\mathbf{r_{iR}(t)}}{d_{iR}(t)} (\lambda_{iR} + (1 - \lambda_{iR})(\frac{1 + cos(\varphi_{iR})}{2})) \tag{4.21}$$

where A_{iR}, B_{iR} , λ_{iR}, d_R are fixed parameters, and $w(\varphi_{iR})$ represent the anisotropic factor, which depends on φ_{iR}, an angle formed between the desired velocity of the pedestrian p_i and the vector $\mathbf{r_{iR}}$, indicating the distance from the robot to the pedestrian p_i, λ_{iR} defines the strength of the anisotropic factor and the projection $cos(\varphi_{iR})$ is calculated using n_{iR}, the normalized vector pointing from the robot to p_i and describes the direction of the force, and e_i the desired motion direction of the pedestrian p_i (which is pointing to the goal). The Extended Social Force Model works in a two dimensional space, and for that, the anisotropic factor is defined as a function depending on the angle φ_{iR} on the x an y coordinate frame. The function is defined as follow:

$$w(\varphi_{iR}) = \lambda_{iR} + (1 - \lambda_{iR})(\frac{1 + cos(\varphi_{iR})}{2}) \tag{4.22}$$

where $cos(\varphi_{iR})$ is calculated by:

$$cos(\varphi_{iR}) = -\mathbf{n_{iR}} \cdot \mathbf{e_{pi}} \tag{4.23}$$

.

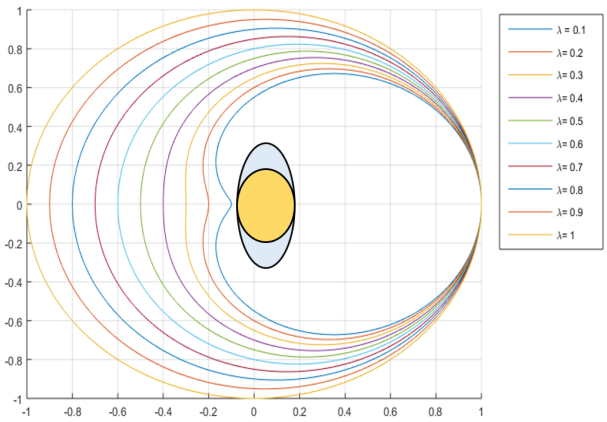


FIGURE 4.3: **Anisotropic factor :** Anisotropic factor scaling the interaction force depending on λ

In this work the anisotropic factor is then extended now with theta angle included, this is, now the angle in between r and z coordinate is introduced as θ_{iR}:

$$\psi(\varphi_{iR}, \theta_{iR}) = w(\varphi_{iR}) \cos(\theta_{iR})(h + \xi w(\varphi_{iR})\eta) \tag{4.24}$$

The formula depends now on 2 predefined variables, λ which was initially introduced in $w(\varphi_{iR})$, and the new variable ξ, and 2 angles created by the positions of the drone and the pedestrians , φ and θ , moreover η is a constant defined by us as 1.25, after performing experiments with non-trained volunteers.

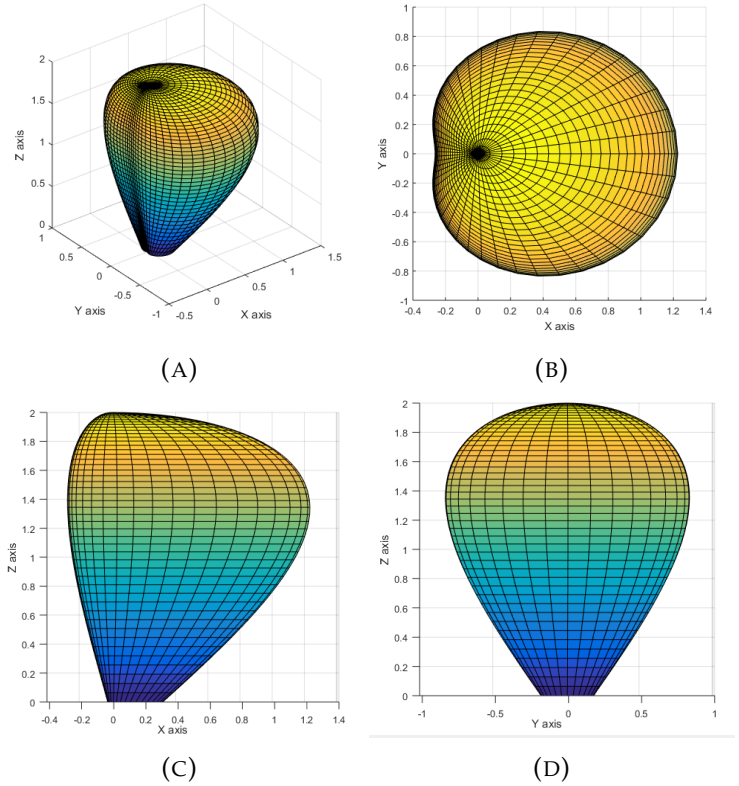


FIGURE 4.4: **Personal space of the human:** Representation of the anisotropic factor in 3D view , with λ_{iR} = 0.25 and ξ = 0.25

where ξ is a proportional factor multiplied by a constant, and it represents a portion of the original anisotropic factor which helps doing a more restrictive space of interaction. The next figure shows the effect of two different values of ξ.

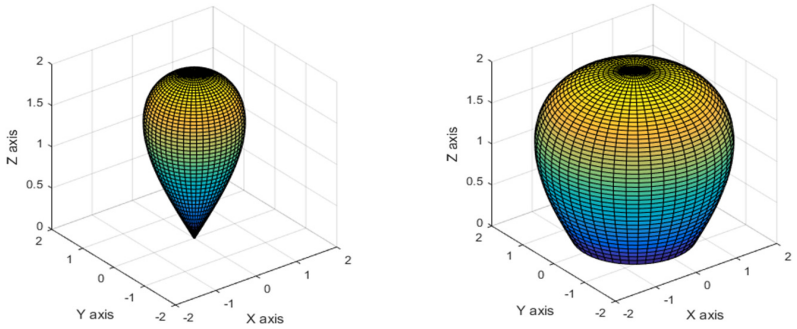


FIGURE 4.5: **Effect of ξ over the personal space:** Different values of ξ in the humans anisotropic scale factor field. *Right image:* Anisotropic scale factor space with $\lambda_{iR} = 1$ and $\xi = 0.01$ *Left image:* Anisotropic scale factor space with $\lambda_{iR} = 1$ and $\xi = 1$

The drone anisotropic factor, is different of the human's one, due to the

fact that the drone is on the air all the time, it does not has the same limitations as the human, so the anisotropic factor is defined:

$$\psi(\varphi_{Rj}, \theta_{Rj}) = w(\varphi_{Rj}) \cos(\theta_{Rj}) \tag{4.25}$$

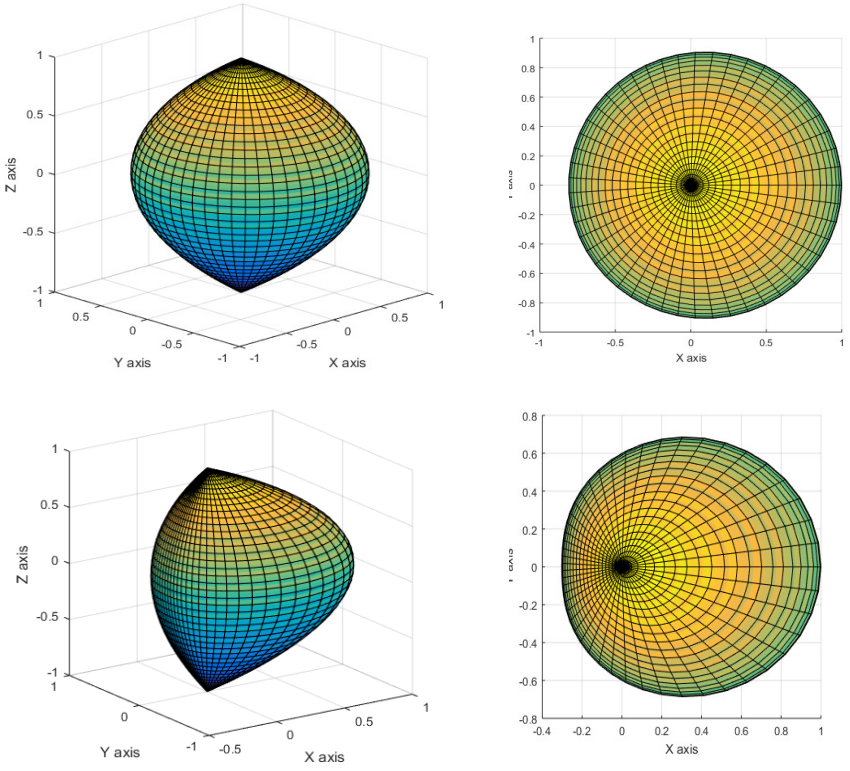


FIGURE 4.6: **Personal space of the drone:** Representation of the drone's anisotropic factor in 3D view , (a) and (b) with λ_{Rj} = 0.90 and (c) and (d) with ξ = 0.50

4.3.6 Quantitative Metrics

To evaluate the performance of the task accomplished by the flying robot, a new quantitative metric is defined. This assessment is based on "proxemics", proposed by [21] and the importance to walk side-by-side. This work considers the following taxonomy of distances between people:

- Intimate distance: the presence of another person is unmistakable, close friends or lovers (0-45cm).
- Personal distance: comfortable spacing, friends (45cm-1.22m).
- Social distance: limited involvement, non-friends interaction (1.22m-3m).
- Public distance: outside circle of involvement, public speaking (> 3m).

To define the metric used in the present work, four different areas must be defined: *(i)* Personal space $\mathcal{C}_i$ of pedestrian p_i, robot's navigation has

to be socially accepted by the person being accompanied, it is necessary that the robot does not perturb the human's vital space and walk beside the person, eq. 4.26. *(ii)* Social distance area $\mathcal{A}$, robots must be allocated in an acceptance social distance. *(iii)* The robot should be in the human's field of view as they interact during the performance of the task and must walk beside the person $\mathcal{B}$. *(iv)* Finally, there are other pedestrians in the environment p_j, the robot is not allowed to perturb pedestrians' personal space $\bigcup_{p_j} \mathcal{C}_j$.

$$\begin{aligned} \mathcal{A} &= \left\{ x \in \mathbb{R}^3 \setminus (\mathcal{B} \cup \mathcal{C}) \mid d(x, p_i) < 3 \right\} \\ \mathcal{B} &= \left\{ x \in \mathbb{R}^3 \setminus \mathcal{C} \mid d(x, p_i \pm) < 3\psi(\varphi_{p_i}, \theta_{p_i}) \right\} \\ \mathcal{C} &= \left\{ x \in \mathbb{R}^3 \mid d(x, p_i) < \psi(\varphi_{p_i}, \theta_{p_i}) \right\} \end{aligned} \tag{4.26}$$

where $\psi(\varphi_{p_i}, \theta_{p_i})$ is defined in eq. 4.24.

Moreover, the robot has beeen represented as a circle of 1 meter of diameter, with center robot's position p_r, $\mathcal{R} = \{x \in \mathbb{R}^2 \mid d(x, r) < 0.5\}$, whose area is $|\mathcal{R}| = \frac{\pi}{4}$.

Thus, we can now define the performance of the task accomplished by the robot, depending on human's position p_i and robot's position p_r.

$$\rho(r, p_i) = \int_{(\mathcal{B} \setminus \bigcup_{p_j} \mathcal{C}_j) \cap \mathcal{R}} \frac{d\mathbf{x}}{|\mathcal{R}|} + \int_{(\mathcal{A} \setminus \bigcup_{p_j} \mathcal{C}_j) \cap \mathcal{R}} \frac{d\mathbf{x}}{2|\mathcal{R}|} \in [0, 1] \tag{4.27}$$

where $\mathbf{x} \in \mathbb{R}^2$. The range of the performance function is defined between 0 and 1. If the complete area of the robot is allocated in zone $\mathcal{B}$, the performance is maximum, i.e., 1. As the robot moves far from the person and enters to zone $\mathcal{A}$, the performance decreases to 0.5. Finally, the performance in zones $\mathcal{C}_i$ is 0, as it is not allowed that the robot enters in people's personal space.

Chapter 5

3D Aerial Social Force Model and Regression Model

5.1 Introduction

In this chapter, we are going to describe the details of the implemented methods. As we said before, one of the main objectives is to make a drone able to follow a person taking into account 3 dimensions with a human-like behavior, making also a comparison with other technique called Artificial Potential Field, and complementing both methods with a regression model, which makes a prediction of the future position of the human. The prediction model was carried out by linear regression, which takes the last five positions delta values, and the weight optimization was calculated by gradient descent method, trying to minimize a cost function.

For the human-path prediction model, we decided to use gradient descent optimization technique, which navigates in the n-dimensional space in the direction of the gradient, given a cost function. RMSprop was the strategy selected to adjust the weights of the model, for its fast convergence. At the end there are simulations showing the performance of the model, as well as the description of its implementation, and also some discussion about the results we obtained, in both real and virtual simulations.

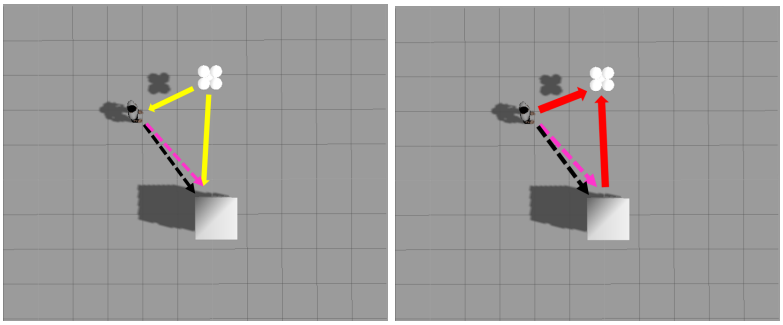
5.2 Method Description

Now, we are going to describe the relevant points of the method, it is important to define what we did and the relevant aspects of implementation's details. First we present the two models, both models work with attraction and repulsive forces, although each of them is motivated by different backgrounds, and perform different force equations. Both work with the same principle, calculating the total force (summation between the attractive and repulsive forces) and applying it to the drone or human. It is important to mention that humans follow the Social Force Model all the time, even when the drone is using artificial potential field. The experiments and the work were carried out thinking mainly in the analysis of the drone's performance, and the project final goal was stated at the beginning as the comparison between methods implemented on the drone.

In the next figure, it is shown how the attractive and repulsive forces work over the drone for both models. Yellow arrows mean the attraction force, one of them is making the robot going to the human position, and

the other one makes the robot going to the predicted position, in this case represented by a pink arrow. As you can see in the image, there is a little discrepancy between this predicted path and the real path (represented as a black arrow), this is motivated by the idea of always finding a difference between the real and the predicted path.

The software used in this case was Robotic Operating System and the Gazebo simulator, this help us to simulate the behavior of several persons interacting each other along with the drone and different objects. The pedestrians are always seeking to reach their goal, but the interaction between people, objects and drones make them to change their way.



(A) Attractive forces which conduct the robot to the goals, the main pedestrian position and the predicted position

(B) Repulsive forces exerted to the robot by the human and the obstacles

FIGURE 5.1: **Representation of the drone the human and an obstacle:** It is shown a representation of repulsive(as red arrows) and attraction forces(as yellow arrows) over the drone.

The drone will always be forecasting the position of the person, based on the historical position data. With this information, the drone makes a calculation of the past delta values and makes the forecast. In the case the main person stays in the same place, the delta value will be zero, and therefore, the forecast will be the same position.

For security and comfortability, the drone will be moving above the 2.0 meters, knowing all the time the position of the person objective, this with the help of the optitrack, which is always publishing in the proper topic the odometry of the person.

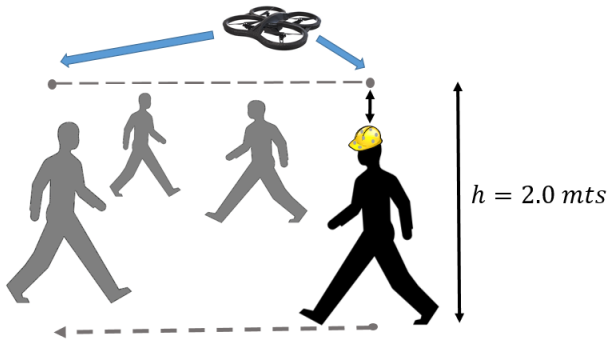


FIGURE 5.2: **Drone's force attraction:** The drone is always pursuing two goals, the position of the human being accompanied and the predicted position in future steps.

The model which governs the drone behavior is influenced by attraction and repulsion forces described in chapter 4. The Final Force $\mathbf{F}_R$ is defined by two attraction forces and two repulsion forces which keep the drone close to the human to be accompanied and also keep it far from obstacles and humans. Both attraction forces, drives the drone to a desired final position. The robot is always trying to reach these points, in this case our model has two attraction forces, one which pursues the position of the human being accompanied, and the second one that pursues the human forecasted position in the next future steps. The robot will adapt its velocity with a certain relaxation time k previously defined. Both attraction forces are defined as :

$$\mathbf{f_R^{goal}} = k(\mathbf{v_R^0} - \mathbf{v_R}) \tag{5.1}$$

The total force of the robot-human interaction force is defined as:

$$\mathbf{F_{R,j}^{int}} = \sum_{j=1}^{P} \mathbf{f_{R,j}^{int}} \tag{5.2}$$

where $\mathbf{f_{R,j}^{int}}$ is the equation previously defined in in Eq. 4.21, but now depending on $\varphi_{R,j}$ and $\theta_{R,j}$.

$$\mathbf{f_{R,j}^{int}} = A_{R,j} e^{(\frac{d_{R_p} - d_{R,j}}{B_{R,j}})} \psi(\varphi_{R,j}, \theta_{R,j}) \tag{5.3}$$

In the same way, the robot-object interaction force is a summatory of all the objects inside a certain range , where $\mathbf{f_{Ro}^{int}}$ is the equation previously defined in in Eq. 4.21, but now depending on $\varphi_{R,o}$ and $\theta_{R,o}$.

$$\mathbf{F_{R,o}^{int}} = \sum_{o=1}^{O} \mathbf{f_{R,o}^{int}} \tag{5.4}$$

and $\mathbf{f_{R,o}^{int}}$ defined as:

$$\mathbf{f_{R,o}^{int}} = A_{R,o} e^{(\frac{d_{R_o} - d_{R,o}}{B_{R,o}})} \psi(\varphi_{R,o}, \theta_{R,o}) \tag{5.5}$$

The total force which will drive the drone is also influenced by four parameters $\alpha, \beta, \gamma, \delta$, and the final force is defined as:

$$\mathbf{F_R} = \alpha \mathbf{f^{goal}_{R,dest}} + \beta \mathbf{f^{goal}_{R,j}} + \gamma \mathbf{F^{int}_{R,j}} + \delta \mathbf{F^{int}_{R,o}} \tag{5.6}$$

Constraints in drone's velocity were also taken into account, depending on how close the drone were, we defined zones inspired on how normal people interact, and as function of the proximity of the pedestrians:

$$v_R = \begin{cases} v_{safety} & \frac{d_{R,j}}{w(\varphi_{R,j})} \leq \mu_{safety} \\ v_{cruise} & \mu_{safety} < \frac{d_{R,j}}{w(\varphi_{R,j})} \leq \mu_{social} \\ v_{free} & otherwise \end{cases} \tag{5.7}$$

where, v_{safety} is the maximum velocity the drone can take when is at least one person inside the inner safety zone, v_{cruise} is the velocity when someone is inside its social safety zone and v_{free} is the maximum velocity when no one is inside its safety zone.

5.3 Human Path Prediction

One of the main contributions of this work to the previously well worked Extended Social Force Model is the prediction model we include in the algorithm, giving the drone the capacity of knowing the direction of the human just knowing the latest positions. This is, taking for example the latest 10 positions of the human, we built the model which allows us to know the next position. And not only that, as this model was built with the differentials in space, this is delta x and delta y, this allows us to build the next positions (of course with a little error), therefore forecasting the path the human would take.

We rely on linear regression models because their robustness, the model is also based on the differential values in x and y positions of the main person. In order to build the prediction model, we first create a synthetic database in which the human behaves following goals, one after other, in an infinite loop. The drone behaves as the Extended Social Force Model dictates, keeping distance from the humans and the obstacles, but also trying to keep close to the main human and to the forecasted position in every time.

This artificial environment was built taking into account 10 people, divided in two groups of five, each group with their respective sequence of goals. Each person of the first group, in which is included the "main" person, this is, the person who is being followed by the drone, will seek to reach the desired current goal. In the next picture, in the left side the first group is shown surrounded by a circle, and a smaller circle indicates who is the main human; the position and the order of the goals are provided by the numbers and the red crosses; in the right side, the second group and the sequence of their goals are shown.

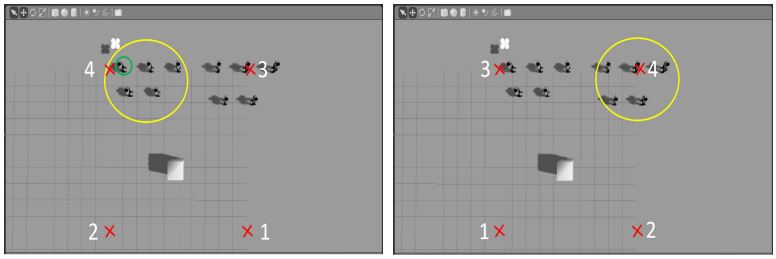


FIGURE 5.3: **Gazebo simulation:** Environment used to create synthetic data, in which the model was simulated along with other humans.

During the simulation, the position of the drone and the main human were collected, in order to build the regression model and to find the best parameters. In the next subsection it is explained in more detail how we built this model with the data collected.

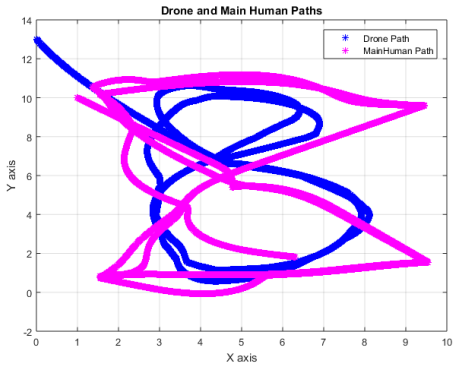


FIGURE 5.4: **Synthetic data collected through simulations:** Collected data indicating the drone path and the human path.

5.3.1 Prediction Model: Linear Regression

Simple Linear regression models imply that only one term is describing the data, this is, one input variable:

$$f(x : \mathbf{w}) = w_0 + w_1 x \tag{5.8}$$

A cost function, also called the error measure, is required and the one that we want to minimize is defined by:

$$J(w_0, w_1) = \sum_{i=1}^{N} (f(x; w_0, w_1) - y)^2 \tag{5.9}$$

where **w** = (w_0, w_1) are unknown parameters to be estimated by the data, and x, in our model, is the differential of the positions values, this is:

$$\Delta x = x_t - x_{t-1} \tag{5.10}$$

In our case, it would be useful to take into account more than one variable and when we use 2 or more values, we are building a multiple linear

regression which generalizes a simple linear regression model by allowing more than one term. Now our function will have the form:

$$f(X) = w_0 + \sum_{j=1}^{p} X_j w_j \tag{5.11}$$

To build the model, instead of using the position of the main human, we take the coordinates of the human position and build an array with the differential values between consecutive positions, and then we did estimate the parameters using gradient descent optimization, which is explained later in this chapter.

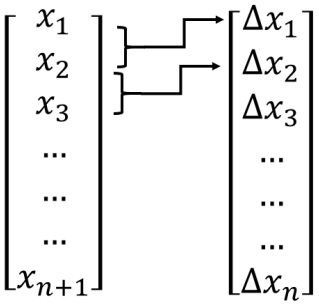


FIGURE 5.5: **Differential position values:** The Differential values help us to know the desired direction of the human, and how fast is moving to the goal.

To estimate the parameters, first we need to build the x and y matrix, recall that we have said the number of previous values to take into account were five , this mean that we are going to have 6 unknown parameters to adjust. To make the optimization of these parameters we take different sample paths of the dataset. At the end, making the model predict the next differential value help us to build the path the human would follow with great reliability. Later it can be seen how with different examples the model can predict with great accuracy the direction of the human.

$$X = \begin{bmatrix} 1 & \Delta x_{n_1,1} & \Delta x_{n_2,2} & \cdots & \Delta x_{n_p,p} \\ 1 & \Delta x_{n_2,1} & \Delta x_{n_3,2} & \cdots & \Delta x_{n_{p+1},p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \Delta x_{n_{N-p},1} & \Delta x_{n_{N-p+1},2} & \cdots & \Delta x_{n_{N-1},p} \end{bmatrix} \quad y = \begin{bmatrix} \Delta x_{p+1} \\ \Delta x_{p+2} \\ \cdots \\ \Delta x_N \end{bmatrix} \tag{5.12}$$

5.3.2 Gradient Descent Optimization Method

Now with the previously collected data, and the matrix built, we can proceed to find the unknown parameters also called weights. These weights were found using the well-known gradient descent optimization method, which with an initial random guess of the parameters, travels through the gradient of the cost function. This method works in an iterative way, in which in each iteration the weights are updated in order to be closer to the optimal minimum, changing the weights in each iteration *k*, seeking that:

$$J(w_0^{k+1}, w_1^{k+1}, ..., w_p^{k+1}) < J(w_0^k, w_1^k, ..., w_p^k) \tag{5.13}$$

until the cost function is not reduced anymore, or until the iteration cycle finishes. The gradient of the cost function is defined as follows:

$$\nabla J(\mathbf{w}) = (\frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, ..., \frac{\partial J}{\partial w_p}) \tag{5.14}$$

We have to recall that our model is taking into account 6 parameters and for that, the direction of the gradient takes 6 dimensions, the complete form of the gradient in the first two directions is shown next, and the generic form is presented after.

$$\frac{\partial J}{\partial w_p} = \frac{1}{N}\sum_{i=1}^{N}(\sum_{j=0}^{p} X_{ij}w_j - y_i)X_p \tag{5.15}$$

RMSprop strategy

In order to move through the gradient we did use the RMSprop strategy. This strategy was proposed by Geoff Hinton [3] and it states that dividing the gradient by the root of the expected value of the gradient (taking into account a short window) makes the learning work much better.

$$E[\nabla J(\mathbf{w})^2]_t = \gamma E[\nabla J(\mathbf{w})^2]_{t-1} + (1-\gamma)\nabla J(\mathbf{w})_t^2 \tag{5.16}$$

where γ means the momentum value, typically set to 0.9 or 0.95, and always < 1.

$$\Delta w_t = -\eta \cdot \nabla J(\mathbf{w}) \tag{5.17}$$

In the above equation η means the learning rate set by the user. Be careful with this parameter because it has a lot of effect in the outcome of the optimization algorithm; a good default value could be 0.001.

$$w_{t+1} = w_t + \Delta w_t \tag{5.18}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt[2]{E[\nabla J(\mathbf{w})^2]_t + \epsilon}}\nabla J(\mathbf{w}) \tag{5.19}$$

where $E[\nabla J(\mathbf{w})^2]_t$ means the decaying average over past squared gradients, and ϵ is a smoothing term that avoid the division by zero value.

The iteration cycle in which the weights are updated can be stopped when a threshold is crossed, this is, when there is no change in the minimum value found, or until the last iteration is done. In the next picture, it is shown the cost function $J(\mathbf{w})$ being decreased through iterations.

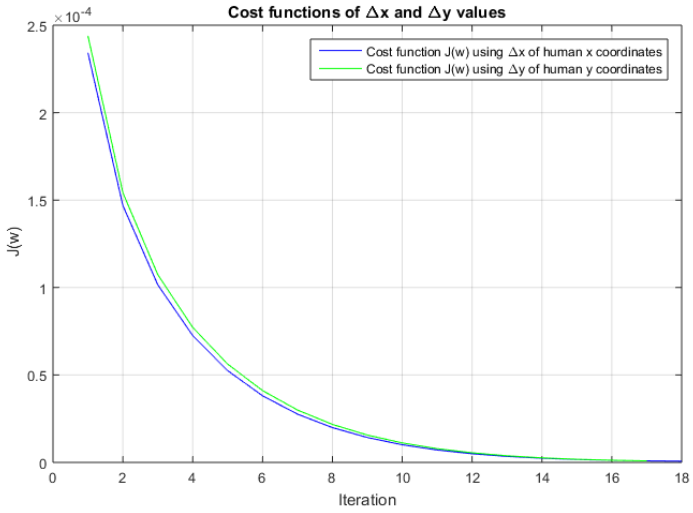


FIGURE 5.6: **Cost function through the optimization process using gradient descent:** Cost function being decreased by iterations using gradient descent method and the RMSprop strategy.

Once found the adjusted weights, we did some experiments over the dataset previously created by the simulation. We found a very good behavior of the model, no matter which data we used to test our model, in spite of we did use only one little portion of the data. Some examples are shown in the next figures, in which we took different parts of the synthetic data and we did some experiments to test the prediction capabilities of our model.

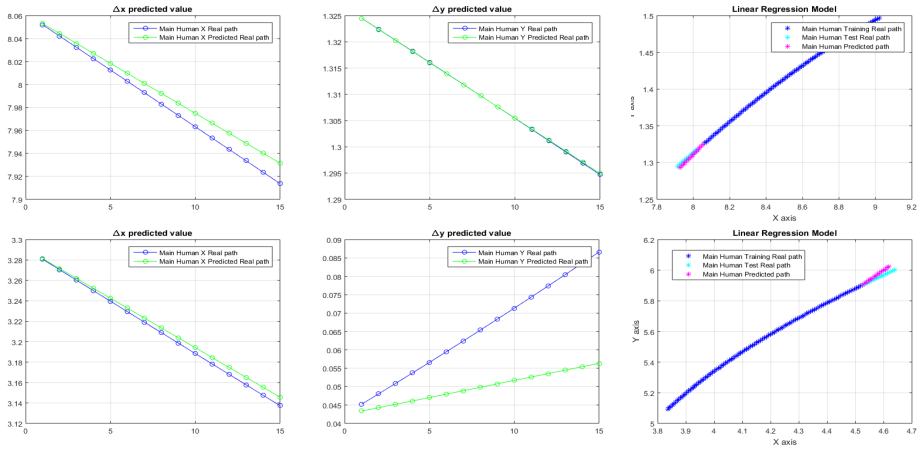


FIGURE 5.7: **Differential values and Forecasted position:** Forecasted Δx and Δy in two different cases((A) and (B) , and (D) and (E)), and the final rebuilt paths ((C) and (F)), 10 positions ahead.

Chapter 6

Experiments

The real-life experiments, as we said before, were developed using the AR-Drone 2.0 a remote controlled flying quadcopter built by Parrot. With impressive features, the quadcopter is a very good choice to test our algorithms in a short-time setup. We also made use of the Optitrack Motion Capture system, a system created by NaturalPoint Inc. In this work, Optitrack was used to analyze an indoor environment to track the human and the drone position in a previously defined space.

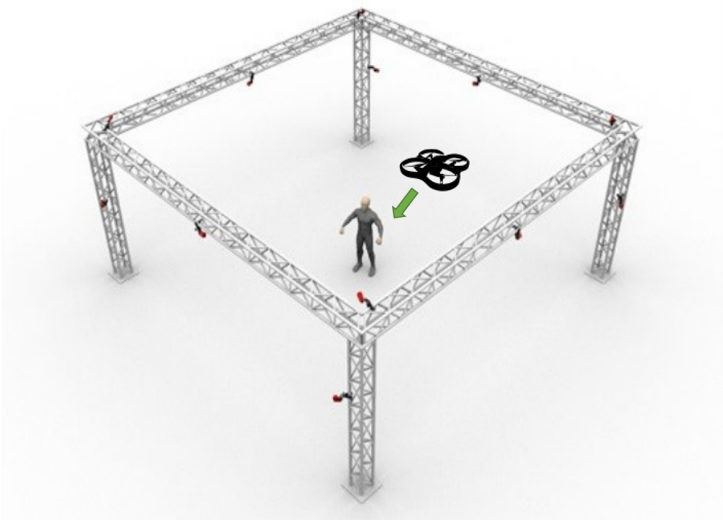


FIGURE 6.1: **Optitrack workspace:** An indoor environment representation using Optitrack depicting the experiments we performed

Optitrack allows us to calculate the forces interacting between the person and the robot and to see the behavior of our model in real time. Both human and drone were using markers in the body to let the software detect the exact position of each of them. In our experiments the working areas measures $5x5$ m, see Fig. 6.1.

6.1 Synthetic experiments

Here, we present the evaluation of the performance of the ASFM, for this reason, we have built a simulated social environment. In this section, we introduce the computation of the parameters that have been used in the model, the results of the implemented regression model to estimate person's motion and the simulations of the complete system.

Interaction	k	A	B	d	λ
Per-Per [34]	2	1.25	0.1	0.2	0.5
Per-Per [52]	4.9	10	0.34	0.16	1
Robot-Person [16]	2.3	2.66	0.79	0.4	0.59
Drone-Person	5	2.75	0.565	0.295	0.55

TABLE 6.1: **Aerial Social Force Model Parameters.** Parameters learned for Robot-drone interaction after applying the minimization process.

6.1.1 ASFM parameters

As we said, the traditional Social Force Model takes into account three different kinds of forces: the human-human, human-object, and human-robot. The first two forces have been studied in [52], whereas [16] describes the interaction between a ground robot and a person. However, the drone-person interaction parameters were not obtained in any previous work, thereby, in this section we introduce the results obtained for the parameters $\{A_{Rj}, B_{Rj}, \lambda_{Rj}, d_{Rj}\}$.

Table 6.1 shows the parameters learned after applying the minimization process, using genetic algorithms, after recording a set of database trajectories, where a drone navigates with a human.

6.1.2 Regression Model

During the synthetic experiments we simulated the behavior of several pedestrians, the drone and the main human. Then, we collected the data which help us to see an approximate real behavior of pedestrians walking to a certain path. With this, we built a model to forecast the next position few seconds ahead. The model is based on the differential values in x and y positions, so this makes it independent of the path since the slope dictates already the direction.

Different trajectories were tested with the regression model, always taking into account the latest five human's positions to make the calculation of the resultant slope, then a possible path the human would follow is built. One of the strongest points is the high accuracy when the path is not so troubled, nevertheless, it also has very good results when the path changes so drastically as in the case of the Fig. 6.2-(a). For our purpose, it is enough to know the intention of which direction the human could take, and with this, the robot can move along this direction, beside the human being accompanied.

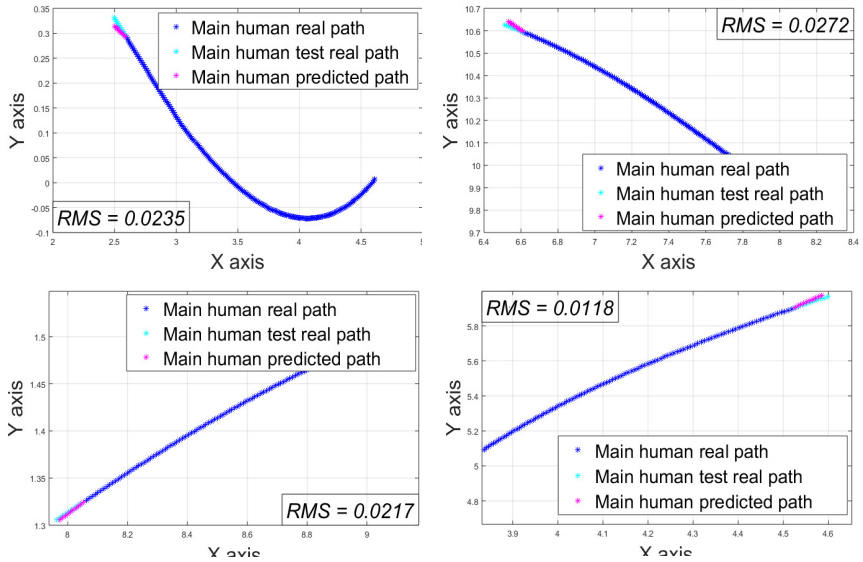


FIGURE 6.2: **Regression Model:** Different human's trajectories and the forecasted paths ten steps ahead of the last position taken into account versus the real path. The error was measure using the Root-mean-square error.

The drone's prediction capacity proved to be useful in a highly dynamic environments, in which a lot of people walk around the main human and the robot. This usually leads to a very changing behavior of the main human, nevertheless, the drone proves to adapt and to accompany the main human with very good accuracy.

6.1.3 Simulations

To this end, we have implemented a complete social environment, depicted in the first row in Fig. 6.4, which takes into account pedestrians, obstacles and drones, each element is reactive to its surrounding according to the ASFM. By doing this, we can get a dynamical environment, in which each action of the autonomous drone alters the behavior of other pedestrians in the environment.

To evaluate mathematically the correctness and the performance of the presented model, we built a simulated social environment. This simulated environment allows us to validate the performance of the method, using the metrics defined in Sec. 4.3.6, in different environments and under different density of pedestrians.

In order to give statistical consistency to our results, more than 10k experiments have been carried out, only varying the initial position of each pedestrian in the simulation. We would like to stress on the fact that the environment has a high density of persons and each person aims to a random destination.

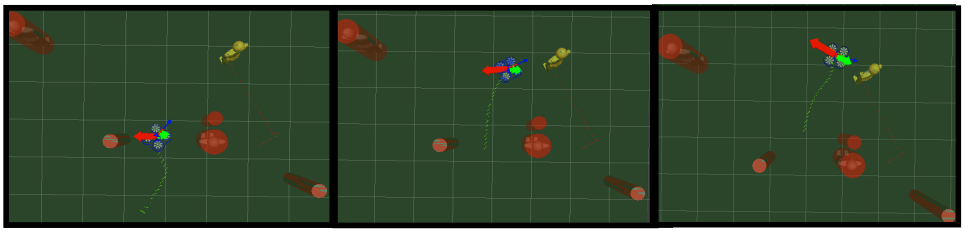


FIGURE 6.3: **Simulation experiment:** Visualization of the force interacting over the drone, using Rviz tool provided by the ROS environment.

The bottom row of Fig. 6.4 shows the overall performance of the different methods with respect to the density of pedestrians in the scene. It can be seen, that using our method ASFM, the performance highly increases. The predictive behavior using the regression model clearly enhances the performance of the task, in both scenarios.

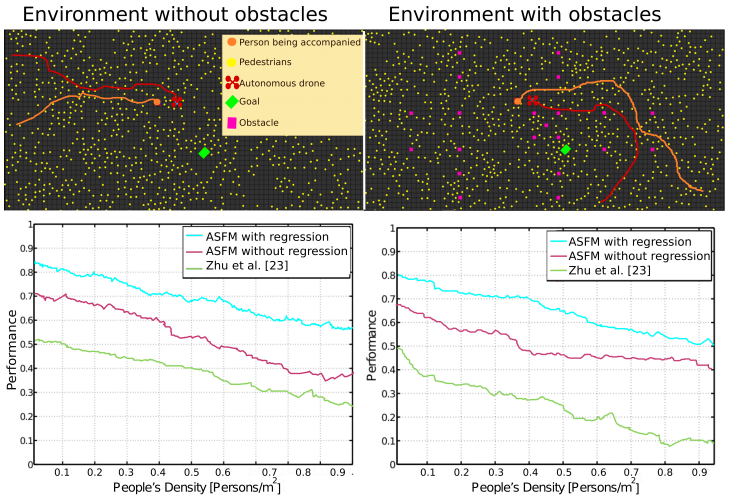


FIGURE 6.4: **Synthetic experiments:** *Top row:* Simulated environments. *Bottom row:* Performance presented previously; blue line represents the ASFM with the model regression; purple refers to ASFM without regression, and green color is the performance of the APF method presented in [**zhu2016**]. All results are function of the pedestrian density in the environment.

6.2 Real-Life experiments

In this section, we present different real-life experiments, in which we test the ASFM model, considering both, humans and obstacles. Two scenarios were tested, the first one, was an unconstrained area, while in the second one there were obstacles. Data was collected to plot how the forces dictate the behavior of the robot and how the robot compensates these forces to recover the distance.

During real-life experiments, we tested our model and its capacity to maintain the drone close to the human and making the drone accompany

the volunteer in a comfortable manner. We analyze the forces interacting in the space in real-time, in a visualization tool provided by the ROS environment, and using Optitrack, we detected the position in which the human, the objects, and the drone were located.

We noticed that all real experiments carried out with our robot and different volunteers achieved the goal, that is, the drone was able to accompany the person while navigating in a social and acceptable manner. In Fig. 6.6 three different examples are shown.

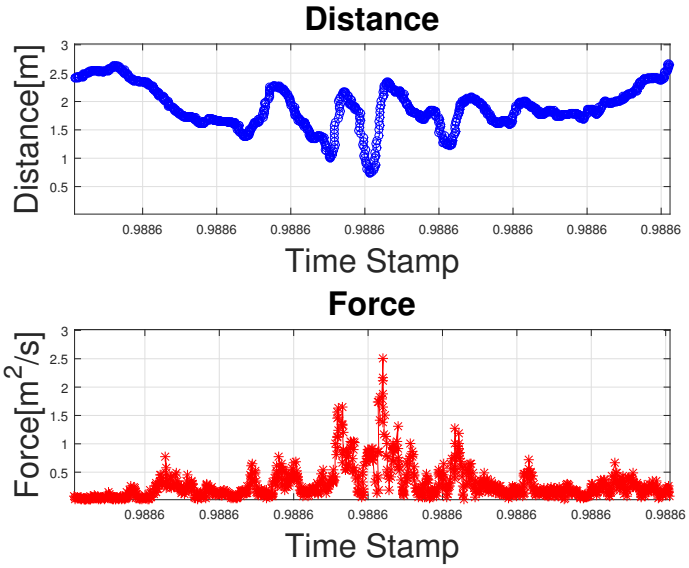
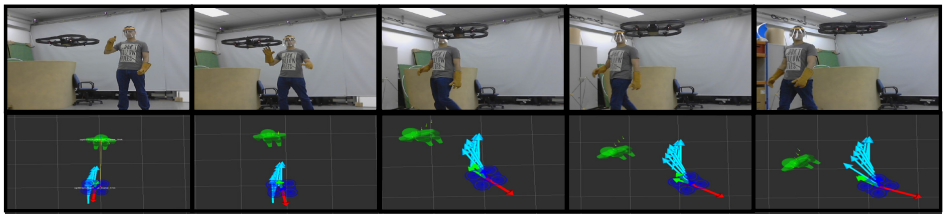


FIGURE 6.5: **Real-life experiments data:** *Top row:* Euclidean distance between the robot and the human *Bottom row:* Force magnitude .

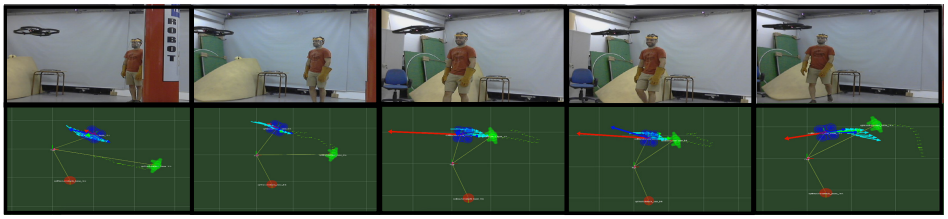
Concretely, in the Fig. 6.6a on the top, it is exposed how a human may walk around the robot in a closer way, and the figures on the bottom show how the repulsion force drawn as red arrow makes the robot move backward, respecting human's personal space.

In a different scenario, in the Fig. 6.6b on the top, it is shown how a human walks in the drone's direction as if they were to collide, and then the human moves backward quickly, in this case the repulsion force is greater when the distance is minimum as you can see in the Fig. 6.5. The total force makes the drone react to its environment making the drone navigate keeping a certain distance between it and its environment.

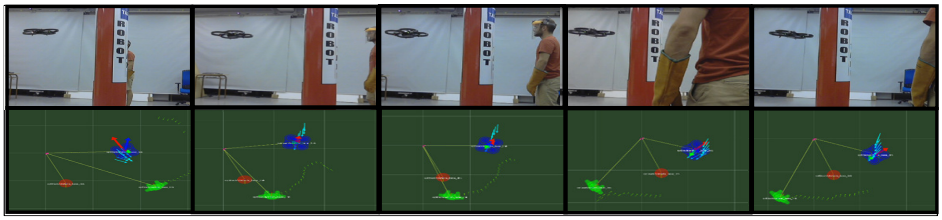
Finally, in Fig. 6.6c we use cylinders as obstacles, which blocked the way to get straight to the human, but the repulsion force that is exerting the object keeps the drone away, as long as the human moves behind the object, here again, the flying robot could avoid the obstacles and navigate in a social acceptable way.



(A) Human moving through a certain path surrounding the drone, making the drone move backward and making the correction of the orientation.



(B) Human moving forward in the direction of the drone, in the image below is shown the repulsion force depicted as a red arrow.



(C) Human moving around an obstacle, the drone stays in the same position because the sum of the forces was zero.

FIGURE 6.6: **Real-life experiments:** *Top row:* Experiments carried out by volunteers at the laboratory. *Bottom row:* Visualization of the scenario and representation of the interaction forces in real-time.

6.2.1 Discussion

Different experiments, both synthetic and real, were carried out in order to show and test the model in different situations. The novel method proved to have an excellent navigation behavior in comparison with the traditional method called Artificial Potential Fields, which, although with the same principles, led us to different results. Different volunteers in different situations tested the model, moving in different directions and varying the velocity of these movements, making the robot react different and showing that is able to maintain a safe distance without making aggressive movements which can scare the volunteer. During the real experiments the data was collected in order to do an analysis of the forces and positions and to find opportunities to enhance the overall model. The data was converted in txt file in order to make it possible to process in other software , for instance MATLAB, which was our case.

Although we expected to have different behavior in simulations and real-life experiments, surprisingly, both were similar, making easy to test different configurations and setting up them in the real drone. Drawbacks were present during the real-life experiments, but nothing that made us stop or modify the original model. One of the limitations we had was the

space of the experiments which was a 5x5 meters space.

The Rviz visualization tool helped us a lot to see the forces interacting in real-life with the drone, the humans and the objects, and to see what happened when we play with the parameters which modifies the repulsion and attraction forces. The Processing tool was useful to test the model with a large amount of pedestrians and objects, something which we could not test in real-life nor in gazebo simulator. The regression model proved to have a very good forecasting ability and a high robustness to the constant changes due to the dynamics of the environment. In chapter 8 we propose some ideas to enhance the model and to make it more robust in some aspects, also some ideas to extend the functionality.

Chapter 7

Conclusions

Up to here, we have shown the results using the model in simulations and real-life experimentation. Next, we present some conclusions of the work and future steps to extend the functionality of the model.

Nowadays, the presence of robots in the daily life is getting stronger, the need of this kind of models as the one we presented is required. We should have in mind that, unlike industrial applications, there is a social factor playing an emerging role, in which protocols, social norms and social environments dictate how robots should behave, as their main objectives are bringing safety and comfort to humans while navigate. The interest on building intelligent robots which can accompany people is increasing in the research community, for that reason, in some countries these robots will start to appear with more frequency.

The overall system has demonstrated very good results, both in synthetic and real experiments, showing the main points which we seeked at the beginning of the project. Social behavior was demonstrated, below, we will introduce the future steps of the real-life experiments on that way. The model makes the robot behave in a social way, exceeding what we expected, and although some little difficulties showed up during the setting up in real-life experimentation, there was nothing relevant which can endanger the course of the experiments.

The linear regression model based on differential values of the latest positions, although simple, had also very good results when the human did not behave in an aggressive way, this is, when the path of the human changes constantly and in different directions all the time. The experiments were developed in the AR.Drone 2.0, but we also had in mind to tested with the ONA Robot, a Drone built for social navigation purposes.

One conference article was presented from this project thesis, in which we showed the implementation of the ASFM, the synthetic and real-life experiments, also a comparison between our method and the traditional one called Artificial Potential Fields. The final results were exposed in the article (as well as in this work) demonstrating that our model had better results that the APF method. Moreover, we are currently working on an article for a journal, in which we are going to test the ASFM with untrained volunteers and make some surveys, with some questions related to how do they feel during the experiments, in order to have evidence and to test the

comfort level the persons have with a flying robot navigating in a social way.

Chapter 8

Future work

8.1 Next steps

One important topic we would like to tackle in the next future is to test the model in real-life involving non-trained volunteers, as well as more static objects. We checked our model under a large set of simulations in crowded environments using Rviz, but in real-life due to the short space, and the size of the drone, it is a difficult endeavor. Considering these new experiments, we can obtain new evidence and we can measure the comfort level of people, in order to test if the robot's behavior is socially accepted.

Moreover, we are also preparing new experiments with the ONA Robot, a drone specifically built for social tasks, and which for some delivery delays, we could not do the test. Besides, due to the model requires a large set of parameters, this implies many dimensions, and it is a difficult task to find the optimal parameters, for that reason, we found many sub-optimal values with can behave in a sub-optimal way, that is, there exist other sets of parameters that provide a social behavior as well. We are now currently working in some extensions of the work which results will help us to write and article for a journal.

8.2 Additional Features

We will consider new additional features regarding different aspects of the overall model, more specifically, the regression model and the 3D Aerial Social Force Model, which in our opinion could lead to a better behavior, making it more complex, robust and intelligent. One of the things we would like to strengthen is the mechanism the drone has to compensate the height it is following; this is, the drone follows a certain height, but sometimes the force that is leading the drone to this position is not enough to put it in the desired height, for these reasons, it is required to enhance the compensation mechanism which can help the drone to gain height or to lose height in case the drone is over the desired z position. Current updates are being developed to the model to improve this height gain and are being tested successfully.

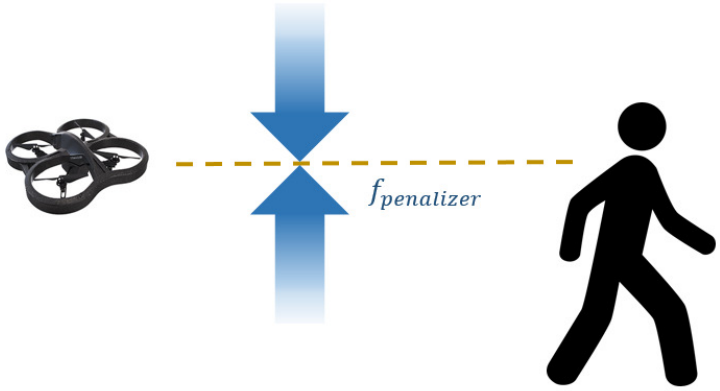


FIGURE 8.1: **Penalizing force:** An extra force to enhance how the drone follows a certain height (only in the z coordinate) is proposed.

Other issue we would like to point out is the forces and the rate ROS is updating and sending these forces to the real drone. As you can see in Fig. 8.2, the force is always changing and creating peaks which can have a negative impact over the real system and also over the real behavior. What we propose in future steps is to implement a new filter as a smoother, to avoid aggressive changes in the total force sent to the drone.

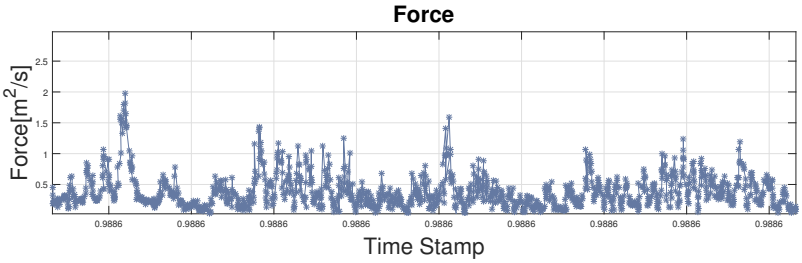


FIGURE 8.2: **Total Force peaks:** The force seems to have peaks over the time, naturally by many factors, one of them, the noise of the real system.

Finally, one of the topics to work on is to enhance the overall navigation system using another perception system. So far, we have worked with the navigation mechanism which can make the robot avoid obstacles and pedestrians, with the ability to keep close to the "main" human. One important capacity the robot should have is the capacity to recognize people and pedestrians with computer vision algorithms, which can give the robot another level of autonomy.

Nowadays, computer vision brings us great advances in recognition tasks, and the idea to add the robot the ability to recognize people and objects, could be a great improvement to what we have been working in this project.

8.3 Limitations of the current work

All the systems have limitations and this is not the exception. One of the main limitations during the project was the human and drone detections. We had to use the Optitrack system, that although it is a very accurate and reliable, we were predestined to realize the experiments in its predefined work space, also expecting to not being occupied by other researches. For now on, testing and debugging the method it is enough to count on this kind of system, but another kind of systems should be used at the time of testing it in real social environments as in the street or social centers.

Thinking so far, we can also increase the time life of the drone, this is because the limited life of the batteries which always gave us just a couple of experiments. To deal with that, we had a couple of batteries being charged while one was in use with the drone.

Bibliography

[1] URL: http://wiki.optitrack.com/index.php?title=OptiTrack_Documentation_Wiki.

[2] URL: http://wiki.ros.org/tum_simulator.

[3] URL: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[4] G. Lidoris Q. Mühlbauer F. Rohrmüller S. Sosnowski T. Xu K. Kühnlenz D. Wollherr A. Bauer K. Klasing and M. Buss. "The autonomous city explorer: Towards natural human-robot interaction in urban environments". In: *International journal of social robotics* (2009), pp. 127–140.

[5] Antoine Bautin, Luis Martinez-gomez, and Thierry Fraichard. "Inevitable Collision States : a Probabilistic Perspective". In: *IEEE International Conference on Robotics and Automation* (2010), pp. 4022–4027.

[6] Jose A Villacorta-atienza Carlos Calvo and Valeri A Makarov. "Prediction-for-CompAction : navigation in social environments using generalized cognitive maps". In: *Biological Cybernetics* (2015), pp. 307–320. ISSN: 0340-1200. DOI: 10.1007/s00422-015-0644-8. URL: http://dx.doi.org/10.1007/s00422-015-0644-8.

[7] Cauchard, Jessica R and Jane, L E and Zhai, Kevin Y and Landay, James A. "Drone & Me : An Exploration Into Natural Human - Drone Interaction". In: *2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 361–365.

[8] Cauchard, Jessica R and Zhai, Kevin Y and Spadafora, Marco and Landay, James A. "Emotion Encoding in Human-Drone Interaction". In: *ACM/IEEE International Conference on Human Robot Interaction*. ACM/IEEE. 2016, pp. 263–270.

[9] Gonçalo Charters Santos Cruz and Pedro Miguel Martins Encarnação. "Obstacle Avoidance for Unmanned Aerial Vehicles". In: *Journal of Intelligent & Robotic Systems* 65.1-4 (2011), pp. 203–217. ISSN: 0921-0296. DOI: 10.1007/s10846-011-9587-z. URL: http://link.springer.com/10.1007/s10846-011-9587-z.

[10] Alpha Daye, Suresh Gobee, and Vickneswari Durairajah. "Autonomous Tour Guide Robot using embedded system control". In: *Procedia - Procedia Computer Science* 76.Iris (2015), pp. 126–133. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.12.302. URL: http://dx.doi.org/10.1016/j.procs.2015.12.302.

[11] Desaraju, V and Ro, H C and Yang, M and Tay, E and Roth, S and Vecchio, D Del. "Partial Order Techniques for Vehicle Collision Avoidance : Application to an Autonomous Roundabout Test-bed". In: *IEEE International Conference on Robotics and Automation,2009. ICRA '09*. IEEE. 2009, pp. 263–270.

[12] Mihai Duguleana and Gheorghe Mogan. "Neural networks based reinforcement learning for mobile robots obstacle avoidance". In: *Expert Systems With Applications* 62 (2016), pp. 104–115. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2016.06.021. URL: http://dx.doi.org/10.1016/j.eswa.2016.06.021.

[13] A.G Fallis. "Probabilistic robotics". In: *Journal of Chemical Information and Modeling* 53.9 (2013), pp. 1689–1699. ISSN: 1098-6596. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[14] Fahimi Farbod. *Autonomous Robots:modeling, path planning, and control*. 2005. ISBN: 9780387095370.

[15] Friedman, Jerome and Hastie, Trevor and Tibshirani, Robert. *The Elements of Statistical Learning: Data Mining , Inference and Prediction*. 2008.

[16] G. Ferrer, A. Garrell, F. Herrero and A. Sanfeliu. "Robot social-aware navigation framework to accompany people walking side-by-side". In: *Autonomous Robots* (2016), pp. 1–19.

[17] L. A. García-Delgado et al. "Repulsive Function in Potential Field Based Control with Algorithm for Safer Avoidance". In: *Journal of Intelligent & Robotic Systems* 80.1 (2015), pp. 59–70. ISSN: 0921-0296. DOI: 10.1007/s10846-014-0157-z. URL: http://link.springer.com/10.1007/s10846-014-0157-z.

[18] Ben Gardiner et al. "Collision Avoidance Techniques for Unmanned Aerial Vehicles Technical Report# CSSE11-01". In: *Techniques* (2011), pp. 1–24. URL: http://www.eng.auburn.edu/files/acad{_}depts/csse/csse{_}technical{_}reports/csse11-01.pdf.

[19] Anaís Garrel Zulueta. "Cooperative Social Robots: Accompanying, Guiding and Interacting with People". PhD thesis. Universitat Politecnica de Catalunya, 2013.

[20] Guo, Lanshen and Wang, Yang. "Environmental Perception of Mobile Robot ". In: *IEEE International Conference on Information Acquisition* (2006), pp. 348–352.

[21] Edward Twitchell Hall. *The Hidden Dimension*. 1966.

[22] Dan He. "FlyingBuddy : Augment Human Mobility and Perceptibility". In: *International conference on Ubiquitous Computing* (2011), pp. 615–616.

[23] Dirk Helbing. *A fluid dynamic model for the movement of pedestrians*. 1992. DOI: citeulike-article-id:3945298. arXiv: 9805213 [cond-mat].

[24] Dirk Helbing. *Buch - Quantitative sociodynamics: stochastic methods and models of social interaction processes*. 2010. ISBN: 9783642115455. URL: http://books.google.de/books?hl=de{\&}lr={\&}id=9Vy1o5VjdPIC{\&}oi=fnd{\&}pg=PR5{\&}dq=helbing+quantitative+social+science{\&}ots=tE7Zp1CPmN{\&}sig=CcK3CAUlMvdpIyzQxoDgLX9uMBQ.

[25] Dirk Helbing and Peter Molnar. "Self-Organization Phenomena in Pedestrian Crowds". In: *Condensed Matter* (1998), pp. 569–577. ISSN: 0265-8135. DOI: 10.1068/b2697. arXiv: 9806152 [cond-mat]. URL: http://arxiv.org/abs/cond-mat/9806152.

[26] Dirk Helbing and Péter Molnár. *Social force model for pedestrian dynamics*. 1995. DOI: 10.1103/PhysRevE.51.4282. arXiv: 9805244 [cond-mat].

[27] B. Kluge. "Recursive agent modeling with probabilistic velocity obstacles for mobile robot navigation among humans". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1 (2003), 376–380.

[28] Thibault Kruse et al. "Human-aware robot navigation : A survey". In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743. ISSN: 0921-8890. DOI: 10.1016/j.robot.2013.05.007. URL: http://dx.doi.org/10.1016/j.robot.2013.05.007.

[29] D. N. Lee. "Guiding movements by coupling taus". In: *Ecological Psychology* 10 (1998), 221–250.

[30] Seokju Lee et al. "Autonomous Tour Guide Robot by using Ultrasonic Range Sensors and QR code Recognition in Indoor Environment". In: *EEE International Conference on Electro/Information Technology (EIT)* (2014), pp. 410–415.

[31] Guanghui Li et al. "An efficient improved artificial potential field based regression search method for robot path planning". In: *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012* (2012), pp. 1227–1232. DOI: 10.1109/ICMA.2012.6283526.

[32] Georgios Lidoris et al. "The Autonomous City Explorer (ACE) Project - Mobile Robot Navigation in Highly Populated Urban Environments". In: *IEEE international conference on Robotics and Automation* (2009), pp. 2238–2244.

[33] Ming Lin. "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation". In: *IEEE International Conference on Robotics and Automation* (2008), pp. 1928–1935.

[34] G. D. Tipaldi M. Luber J. Stork and K. O. Arras. "People tracking with human motion predictions from social forces". In: *IEEE International Conference on Robotics and Automation* (2010), pp. 464–469.

[35] Kristijan Macek, Dizan Vazquez, and Fraichard. "Safe Vehicle Navigation in Dynamic Urban Environments - A Hierarchical Approach". In: (2008).

[36] Jan R Magnus and Wendun Wang. "Weighted-average least squares prediction â^—". In: (2013), pp. 1–38. ISSN: 15324168. DOI: 10.1080/07474938.2014.977065. URL: http://www.janmagnus.nl/wips/wip30.pdf.

[37] Andreu Corominas Murtra, Josep M Mirats Tur, and Alberto Sanfeliu. "Efficient Active Global Localization for Mobile Robots Operating in Large and Cooperative Environments". In: *IEEE International Conference on Robotics and Automation* (2008), pp. 2758–2763.

[38] Nagi, Jawad and Giusti, Alessandro and Caro, Gianni A Di and Gambardella, Luca M. "Human Control of UAVs using Face Pose Estimates and Hand Gestures". In: *ACM/IEEE International Conference on Human-Robot Interaction* c (2014), pp. 252–253.

[39] Yasushi Nakauchi and Reid Simmons. "A social robot that stands in line". In: *Autonomous Robots* 12.3 (2002), pp. 313–324. ISSN: 09295593. DOI: 10.1023/A:1015273816637.

[40] Tayyab Naseer. "FollowMe : Person Following and Gesture Recognition with a Quadrocopter". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2013), pp. 624–630.

[41] Jason M. O'Kane and Jason M. O. Kane. *A gentle introduction to ROS*. 2013, p. 155. ISBN: 978-1492143239. DOI: 8-25-2014.

[42] *Parrot AR.Drone 2.0: User guide*.

[43] Silvia Julissa Roncal et al. "Daedalus : A sUAV for human-robot interaction Daedalus : A sUAV for Human-Robot Interaction". In: March (2014), pp. 2–4. DOI: 10.1145/2559636.2563709.

[44] Y. Manabe Y. Nakami T. Takamori S. Tadokoro M. Hayashi. "On motion planning of mobile robots which coexist and cooperate with human". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (1995).

[45] Weisberg Sanford. *Applied Linear Regression*. 2005. ISBN: 0-471-66379-4. URL: https://books.google.es/books?id=xd0tNdFOOjcC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

[46] Satoru Satake et al. "How to Approach Humans?-Strategies for Social Robots to Initiate Interaction-". In: *Journal of the Robotics Society of Japan* 28.3 (2010), pp. 327–337. ISSN: 0289-1824. DOI: 10.7210/jrsj.28.327.

[47] Stefan Schneegass et al. "Midair Displays : Concept and First Experiences with Free-Floating Pervasive Displays". In: *International Symposium on Pervasive Displays* (2014), pp. 1–5.

[48] Wai Shan, Florence Ng, and Ehud Sharlin. "Collocated Interaction with Flying Robots". In: *IEEE International Symposium on Robot and Human Interactive Communication* (2011).

[49] Chaoxia Shi, Yanqing Wang, and Jingyu Yang. "A local obstacle avoidance method for mobile robots in partially known environment". In: *Robotics and Autonomous Systems* 58.5 (2010), pp. 425–434. ISSN: 0921-8890. DOI: 10.1016/j.robot.2010.02.005. URL: http://dx.doi.org/10.1016/j.robot.2010.02.005.

[50] Reid Simmons. "The Curvature-Velocity Method for Local Obstacle Avoidance". In: *IEEE International Conference on Robotics and Automation* April (1996).

[51] Daniel Szafir et al. "Communication of Intent in Assistive Free Flyers". In: *ACM/IEEE international conference on Human-robot interaction* 2.1 (2014), pp. 358–365.

[52] F Zanlungo, T Ikeda, and T Kanda. "Social force model with explicit collision prediction". In: *EPL (Europhysics Letters)* 93.6 (2011), p. 68005. ISSN: 0295-5075. DOI: 10.1209/0295-5075/93/68005. URL: http://stacks.iop.org/0295-5075/93/i=6/a=68005.

[53] Xunyu Zhong, Xungao Zhong, and Xiafu Peng. "Neurocomputing Velocity-Change-Space-based dynamic motion planning for mobile robots navigation". In: *Neurocomputing* 143 (2014), pp. 153–163. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2014.06.010. URL: http://dx.doi.org/10.1016/j.neucom.2014.06.010.

[54] Lihua Zhu, Xianghong Cheng, and Fuh Gwo Yuan. "A 3D collision avoidance strategy for UAV with physical constraints". In: *Measurement: Journal of the International Measurement Confederation* 77 (2016), pp. 40–49. ISSN: 02632241. DOI: 10.1016/j.measurement.2015.09.006. URL: http://dx.doi.org/10.1016/j.measurement.2015.09.006.

[55] Ziebart, Brian D and Ratliff, Nathan and Gallagher, Garratt and Mertz, Christoph and Peterson, Kevin and Bagnell, J Andrew. "Planning-based Prediction for Pedestrians". In: *IEEE/RJS International Conference on Intelligent Robots and Systems* (2009), pp. 3931–3936.