Reactive Flock Traffic Navigation: The Design of an Emergent Behavior Based on Social Potential Fields

T H E S I S

Master of Science in Intelligent Systems

Instituto Tecnológico y de Estudios Superiores de Monterrey

By

Edén Alejandro Alanís Reyes

May 2009

Reactive Flock Traffic Navigation: The Design of an Emergent Behavior Based on Social Potential Fields

THESIS

Master of Science in

Intelligent Systems

Instituto Tecnológico y de Estudios Superiores de Monterrey

By

Edén Alejandro Alanís Reyes

May 2009

Instituto Tecnológico y de Estudios Superiores de Monterrey

Gradúate Programs in Mechatronics and Information Technologies

The committee members, hereby, certify that have read the Master Thesis presented by Edén Alejandro Alanís Reyes and that it is fully adequate in scope and quality as a partial fulfillment of requirements for the degree of Master of Science in:

Intelligent Systems

Thesis Committee:

PhD. Ramón F. Breña Pinero
Thesis Advisor

PhD. Gildardo Sánchez Ante
Committee Member

PhD. Carlos Astengo Noguez
Committee Member

PhD. Joaquín Acevedo Mascarúa
Director of Research and Gradúate Programs. School of Engineering.

May de 2009

Reactive Flock Traffic Navigation: The Design of an Emergent Behavior Based on Social Potential Fields

THESIS

Master of Science in

Intelligent Systems

Instituto Tecnológico y de Estudios Superiores de Monterrey

By

Edén Alejandro Alanís Reyes

May 2009

To my beloved family, for their endless love, support and patience.

Acknowledgments

I would like to express my sincere gratitude to my Thesis Committee, integrated by Dr. Ramón Brena Pinero, Dr. Gildardo Sánchez Ante and Dr. Carlos Astengo Noguez, for their adivisory during the development of this research work. Also, I want to thank Luis Miguel García Velázquez for his constant assistance. Thank you very much for your patience and support you provided me to improve this work.

I want to thank my father for being always an inspiration for me to pursue wisdom, and for being the source of my strength and encouragement. Besides, I want to thank my mother for her day-and-night prayers and blessings, without which it would have been impossible for me to face and go through this callenging step in my life.

I want to thank my brothers, for being such good friends of mine, and for all their support and company; thank you very much for being always there for me. Special thanks to Gustavo Reyes Venegas and Roberto Reyes Venegas, for staying in touch with me all this time I have been out of home.

Last but not least, I want to thank all my friends in Morelia: Coy, Xaime, Rubén, Karla, Miriam, Paco, Lucy, Gloria, Mayra and all the rest, for showing me so much love when I left home. I really missed you guys all this time.

Besides, in my wonderful stay in Monterrey I met people whose friendship I am sure will last forever; thank you all for making me feel as if I never left home.

Edén Alejandro Alanís Reyes

Instituto Tecnológico y de Estudios Superiores de Monterrey
May 2009

Reactive Flock Traffic Navigation: The Design of an Emergent Behavior Based on Social Potential Fields

Edén Alejandro Alanís Reyes, M. Sc.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2009

Thesis advisor: PhD. Ramón F. Brena Pinero

Urban traffic management is a recurrent problem in big cities nowadays, due to different issues like the increasing number of cars that enter the city roads, making the demand greater than the roadway's capacity. Therefore, the need to control urban traffic arises with high priority, in order to decrease side effects like high fuel consumption resulting in high vehicle emissions.

Currently, traffic is mainly controlled by traffic lights, static signs and electronic boards, which provide important information about traffic flow, accidents or other related data. Besides, in some places, there have been introduced more advanced schemes like intelligent traffic signal coordination programs; however, this approach brings only 1% of delay reduction [25], which does not represent an efficient solution. Additionally, there are several treatments that are aimed to gain more benefits from the infrastructure and often lead to very high delay reductions, but some of them (the overpass, for instance) are very expensive.

In recent years, some other high-level approaches have been proposed, making use of Multiagent Systems (MAS) techniques, which have proven to be very efficient and promissing regarding their evaluation results. One of such innovative mechanisms is the "Flock Traffic Navigation based on Negotiation" method (FTN) [4]. Inspired by nature, this method proposes a mechanism for vehicles to gather up into groups, in order to get a speed bonus that enables them to travel faster, thus reducing congestion levels.

The present research work explores a reactive approach to the FTN model, which aimes to overcome specific issues detected on this mechanism, as well as to serve as an alternative solution which will enable an interesting comparative analysis between a

deliberative solution against a *reactive* one. Furthermore, this might lead to the design and development of a *hybrid* urban traffic management method.

The proposed solution, called Reactive Flock Traffic Navigation (rFTN, for short), describes an emergent behavior, which is embedded in vehicles and designed to make them navigate under the FTN paradigm. In order to achieve this, several mechanisms have to be performed by each car, within the rFTN calculations. Each one of this workings are explained.

This document also presents experimental data, which is aimed to both *explore* and *validate* the performance of the proposed solution model. In this phase, the rFTN was compared against traditional and advanced methods for the traffic management problem.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Traffic is defined as the movement of people, goods or vehicles from one location to another. This movement takes place along a specific facility or pathway that can be called a guideway. As people's transportation requirements change, traffic flow is affected and, typically, increased. Thus, one of the main challenges in traffic control is to accommodate the traffic in a safe and efficient way. Efficiency can be thought of as a measure of movement levels relative to the objective for a particular transportation system and the finances required for its operation, while safety is concerned with the management of traffic to reduce or eliminate accidents, which is an important issue, as human lives are closely related to the different types of transportation [5].

Nowadays, people always face traffic congestions in big urban areas. Currently, traffic is controlled through traffic-lights, static signs and, in some places, electronic boards that inform drivers about relevant information like traffic flow, accidents or weather-related situations. Traffic congestion describes a condition in which vehicle speeds are reduced below normal, increasing drive times, and causing vehicle queuing. It occurs only when the demand is greater than the roadway's capacity [12, 29]. The US Federal Highway Administration (FHWA) defines traffic congestion as the level at which transportation system performance is no longer acceptable due to traffic interference.

Congestion levels have risen in cities of all sizes since 1982, indicating that even the smaller areas are not able to keep pace with the rising demand. Furthermore, there are several statistics that point to worsening congestion levels, as it extends to more time of the day, more roads, affects more of the travel and creates more extra travel time than in the past [9]. According to the US Federal Highway Administration [1], congestion levels have risen to levels experienced by the next largest population group every 10 years in 2001; that is, cities between 500,000 and one million people experienced the congestion of cities between one and three million in 1992.

Congestion occurs during longer portions of the day and delay more travelers and goods than ever before: today, the average weekday trip takes almost 40 percent longer in the peak-period, than the same trip in the middle of the day; on the other hand, in 1982, it was only 13 percent longer. The 2007 Urban Mobility Report [25] estimates congestion costs in about $78 billions, if current fuel prices are used, and states that the average annual delay for every person using motorized travel in the peak periods in the 437 U.S. urban areas studied climbed from 14 hours in 1982 to 38 hours in 2005".

On the other hand, it has been shown that reducing total congestion saves time and fuel, and leads to decreased vehicle emissions [1]; thus, it can be inferred that the benefits of solving this problem are related to financial, social and environmental dimensions, which is why it is an important issue to deal with.

Recently, some novel agent-based methods, focused on alleviating traffic congestions, have been proposed, taking advantage of emerging technologies like the Global Positioning System (GPS). One of such methods is the "Flock Traffic Navigation Based on Negotiation" [4], developed by Carlos Astengo Noguez and Ramón Brena Pinero, in which they propose that vehicles could navigate automatically in groups called "flocks", inspired in the way birds, animals and fish organize themselves effortlessly traveling in flocks or heards that move in a very coordinated way. This method makes use of coordination mechanisms between vehicles for them to group into flocks, in order to handle traffic navigation along intersections, in a safer way than other related methods.

In this research work it is being proposed to gather vehicles into flocks, based on the Artificial Potential Fields (APF) principle [14], in which they will sense attraction/repulsion forces, that are going to be induced by other vehicles as will be described later, along the paths they take during their trip, with the assumption that cars are controlled by agents. An emergent behavior, which enables vehicles to gather up into flocks, have been designed and developed upon this APF approach and is embedded in each vehicle agent. This model is called *Reactive Flock Traffic Navigation* (rFTN, for short).

It is important to outline that, in this flock navigation model, the greater the flock size is, the greater the chances it has to go through an intersection without decelerating or stopping. Hence, traffic congestion levels and waiting times at intersections are expected to decrease.

Regarding the city in which this model is going to be implemented, the following assumptions are taken into account [4]:

- Cars are controlled by intelligent autonomous agents.
- Every car is able to know the position of other vehicles at any given time.
- Vehicles are able to know the destination of other vehicles by sharing this information with them.
- Vehicles know the speed limit within the city.
- Vehicles have a way to measure the distance between them.
- Vehicles have a way to compute the distance to the intersections.

- The city blocks are always of a squared shape.
- Time and space are discretes.

A simulation environment can be a useful tool when it comes to the analysis of the performance of a given method, or algorithm, like the one proposed in this research work. Besides, it enables us to explore different variations of the algorithm's settings, which leads us to conduct a rich experimentation to analyze their effects in the variables that are going to be measured. In recent years, urban traffic simulators have been developed, in order to show the way we can obtain better results regarding different traffic variables [28, 3]. Therefore, this thesis proposes and experiments with a traffic navigation approach that produces a reduction in vehicles' waiting time and overall trip time, with the aid of a simulation.

1.1 Problem Statement

Traffic congestion has been a recurrent issue throughout the history, ever since ancient civilizations. It has been handled in a broad variety of ways, according to the specific needs of the particular scenario in which it appears. In the first century B.C. Julius Caesar, current emperor of Rome, banned all wheeled traffic during daylight; later on, Emperor Hadrian limited the number of vehicles entering Rome, by the first century A.C. [5].

The first traffic lights were implanted in London, just outside the British Houses of Parliament, on december 10th, 1868. However, it has not been reported any other innovation on this matter ever since, aside from the very expensive multilevel intersections, which not always can be afforded, or the installation of such traffic lights, which is inefficient in the sense that it requires cars to stop even when there is no traffic at all and they could go through the intersection.

In recent dates, emerging location-based technologies have taken advantage of the Global Positioning System (GPS), and the mobile communications; this motivates the efforts of developing futuristic traffic handling, that could be feasible using these new technologies as a tool to support and ease their implementation.

One of the models already proposed is related to agent-based automated vehicle coordination mechanisms [9]; however, each vehicle in this model negotiates *individually* with the intersection which it is about to go through, in order to make a reservation of the time slots of the intersection during which they may pass. This leads to traffic merging and may result in dangerous situations, as individual vehicles are being alternated. Moreover, external factors like mechanical failures or blown tires might be a cause of accidents, which is a major issue if we realize that human lives are being exposed to a certain risk.

Thus, another approach on this matter is dealing with vehicles as *groups*, called "flocks" [4]; this model is far safer, because it abandons the idea of merging and alternating individual vehicles at intersections. In order to create the vehicle flocks, coordination mechanisms from Multiagent systems are used; *agents* are considered autonomous computer systems located in a given environment and capable of interacting with other agents by way of cooperation or negotiation, among others.

However, this model presents an issue in scenarios of a high volume of traffic; as negotiations between a great amount of cars have to be done, the processing time for them is delayed in approximately 1 second, for an environment of 100 agents. This might not sound that critical, but if we suppose a given vehicle travels at 100 kilometers per hour in the scenario, this processing delay will result in agents receiving negotiation messages 27.7 meters later, which might be too late to resume negotiations, and even coordination activities, accurately.

My research work deals with a novel way of creating these vehicle flocks, inspired in the Artificial Potential Fields (APF) principle, in which vehicles will sense an attraction force induced by their destination, as well as attraction/repulsion forces induced by other vehicles complying with some *requirements* specified in this navigation model; it is important to outline that the main attraction force that any vehicle senses is the one towards its destination, and that not every vehicle in the scenario induces an attraction force to another one, as they have to comply with some features that will be specified later on this document. This behavior will enable the vehicles to gather up into flocks as they move towards their destination.

Potential function approaches to navigation enables us to express multiple constraints and objectives in an elegant way within navigation problems. Besides, they can be easily extended to fit a Multi-agent scenario, in order to develop a so-called *emergent behavior* [23], like the flocking traffic model proposed in the FTN. So, with the potential function approach we can develop a simple *reactive* model that makes the vehicles navigate in a very similar way as the FTN, letting aside its negotiation mechanism, but making use of less resources and computational time to be processed.

1.2 Motivation

Urban traffic congestions is a problem that people face everyday in big cities. According to The 2007 Urban Mobility Report [25], in 2005 traffic congestions caused 4.2 billion hours of travel delay and 2.9 billion gallons of wasted fuel, which represents an increase of 2 million hours and also 2 million gallons from 2004, to a total cost of more than $78 billion USD, which is a problem regarding with the fact that "urban areas are not adding enough capacity, improving operations or managing demand well enough to keep congestions from growing larger".

Furthermore, if we take into account that according to the Federal Highway Administration [1] reducing total congestion saves time and fuel, and leads to decreased vehicle emissions, we can realize that the urban traffic problem affects three different areas: financial, social and environmental. Thus, efforts aiming to solve this issue can lead to benefits on three major dimensions of a population.

Developing solution models for alleviating traffic congestions is not an easy task, as it is a very fast growing problem and the best options on this matter are quite expensive. For instance, overpasses represent the most optimal alternative to enable fluid traffic at intersections, because it never requires cars to slow down or stop when they are passing through, just like if there wasn't an intersection at all; however, it is only worth the cost when built in highly congested intersections.

Therefore, some other treatments were designed to take advantage of existing infrastructure, such as the traffic signal coordination programs that try to reduce waiting times at intesections. These kind of techniques barely decrease the overall trip delays, but they provide a more regular traffic flow nonetheless. Moreover, with the aid of Artificial Intelligence and emerging technologies, some other traffic handling schemes have arisen recently, and have prooved to outperform the current strategies.

Previous research work has been conducted on this matter, like the Stone and Dresner's model [9], which provides a novel way to manage traffic at intersections, or the Flock Traffic Navigation model [4], which organizes traffic in a rational way via negotiations and coordination mechanisms from the Multiagent Systems theory. Since these and other models have proven to be able to decrease congestion levels, they have opened a new research area that is broadening the solutions portfolio for this problem, taking advantage of the emergent technologies.

Hence, the main motivation of this research work is to decrease urban traffic variables, such as average waiting times, by developing a navigation model capable of organizing vehicles in an efficient way, while demanding a low computational effort to be processed. Besides, the analysis of the results drawn from this model and its comparison against other traffic methods will make this research area richer in concepts and techniques, which might open new research lines.

1.3 Objectives

The general objective of the present Master Thesis is to design and develop a model capable of gathering up vehicles into flocks by making use of the Social Potential Fields paradigm, which is based on the Artificial Potential Field (APF) technique.

Since my research work is related to the development of a new approach of flock creation in the Flock Traffic Navigation model, the objectives include:

- To determine if it is possible to flock cars using an APF approach.

- To find out how this approach could change (or modify) the Flock Navigation algorithm.
- To measure performance parameters and urban traffic variables, in order to be compared against other related methods.
- The developed model should have the following features:
  - To have a simplified implementation.
  - To require a low computational effort to be processed.
  - To decrease overall trip-times, and other urban traffic variables which will be later discussed.
  - To maintain a safe scenario for vehicle-navigation, by avoiding collisions between them.

Furthermore, the following issues regarding the Social Potential Fields and APF's have to be solved:

- To research about APF techniques, in order to determine which definition of potential field is the most suitable for the solution model.
- To determine the way vehicles are going to be attracted by other cars, at the beginning of their trajectory.
- To state the way vehicles are going to split up from their flocks, at the end of their trajectory, in order to accurrately reach their desination.
- To ensure that vehicles do not collide during their trip, by determining the apropriate restrictions within the APF model.
- To modulate the maximum acceleration and speed of the vehicles, in order to keep a realistic and safe environment.
- To design an APF-based intersection management mechanism.
- To design an APF-based mechanism to make vehicles mantain a flocking formation, based on the available lanes.
- To make vehicles look for alternative paths when congested streets have been detected, based on an APF-based repulsion force.

1.4 Hypothesis

There are several methods for path planning, but, according to Lee and Kardaras [18], the most practical of them are achieved by using potential field approaches, because the path is determined locally by some forces that are defined so that the agent moves towards an objective position; the other relative approaches need a global analysis of the scenario and in some of them a preprocessing phase is also needed.

The hypothesis of this research is that the Social Potential Fields paradigm (SPF) can lead to a proper vehicle flock creation in the Flock Traffic Navigation algorithm, resulting in a less congested and hence more fluid traffic in cities. Therefeore, this Thesis is guided by the following research questions:

- Is there an efficient way of creating vehicle flocks through SPF methods?
- What is the best flock members selection criteria?
- How can we make vehicles mantain a flock formation?
- How can the concepts of maximum acceleration and speed be accurately implanted in the SPF model?
- Do congestion levels, and related traffic parameters, decrease by implementing this model?
- What is this model's performance, compared to the negotiation-based method proposed?
- Is this model suitable to create a *hybrid* model along with the FTN?

1.4.1 Justification

In previous research projects [15, 26, 6, 20] the APF method has been applied successfully in solving a broad variety of path planning problems, making this approach one of the most popular and effective reactive methods. The excellent results and the low computational effort needed to compute APF-based algorithms is the underlying feature of the research work done on this matter; besides, the APF model enables the development of a scalable algorithm, since navigation or path planning restrictions can be added, removed or modified easily at any time. Taking these facts into consideration, the research conducted in this Master Thesis can be justified, as the underlying technique which will be used is aligned with the Thesis objectives, hence, strengthening the idea of the proposed hypothesis to be correct.

1.5 Contribution

This research work provides major information about the way in which it is possible to develop a flocking emergent behavior, in order to organize urban traffic and, thus, alleviate congestion levels and decrease vehicles' waiting times and navigation times. Based on the research questions, this work contributes particularly on the better understanding of:

- The way in which an SPF model can be used to direct traffic in an efficient way, which will enable us to organize vehicles' navigation in an ordered manner. And, if we take into consideration specific features, we can take advantage of this ability to *direct* traffic, in order to create groups or *flocks* of vehicles.
- The low computational effort needed to process the rFTN model, since the SPF approach is known to be a simplistic, though highly effective, path planning model. Since the proposed model will be compared against other traffic approaches, this research work also provides an analysis of different traffic variables measured on different traffic navigation models.
- The way in which a given vehicle decides which other cars to gather up with. This is a major feature of both the FTN and the rFTN models, due to the flocking nature of those navigation paradigms.
- The innovative way in which the rFTN uses the results of the SPF model, in order to establish a maximum acceleration and speed for vehicles. This is done to keep a realistic and safe urban navigation environment.
- The way in which vehicles mantain a flock formation.
- The way vehicles avoid congested streets, if any, based on the SPF paradigm.
- The way vehicles avoid collisions, to keep a safe navigation.
- The measures of traffic variables, which are computed to show that the rFTN is able to decrease congestion levels and other related urban traffic parameters.
- The advantages that can be considered to be implanted into the FTN model, in order to enhance its performance. This can lead to the development of a robust hybrid algorithm, combining the strenghts of both flocking models.

1.6 Document structure

The general organization of this Master Thesis document is presented in the following:

Chapter 2 presents a collection of interrelated concepts, relevant to my thesis, involving general Path Planning concepts and methods, the Artificial Potential Field approach, which is the foundation of the urban traffic navigation method proposed in this research work, and simulation concepts that will be useful in the description of the experimentation platform. The proposed solution model is presented in Chapter 3, which provides the reader with an in-depth description of the Reactive Flock Traffic Navigation model and its workings, which are aligned to this research objectives, along with the methodology under which the experimentation was conducted. Then, Chapter 4 introduces the experimental platform, which was developed to support the conduction of experiments and their analysis. Later, Chapter 5 describes the preliminar experiments performed over specific features of the solution model, in order to tune them; besides, it explains the experimental setup, test cases and the results. Afterwards, Chapter 6 presents the final experimentation over the complete solution model, and the results obtained during this phase. Finally, Chapter 7 states the conclusions drawn form this work and, also, it describes the suggested future work.

Chapter 2

Theoretical framework

In this Chapter it is presented the theoretical background underlying this Master Thesis. Topics included in this Chapter are necesary in order to understand the whole developed model, its implications and workings. First, it is provided a brief review of relevant Traffic Theory concepts, traditional solution methods for alleviating traffic congestions and general path planning methods. Later, it is introduced the Artificial Potential Field approach and related concepts that will contribute to the full comprehension of this paradigm. Then, this Chapter provides a description of Group Behaviors, in order to undestand the concept of emergent behaviors created in multi-agent environments with the aid of the so-called Social Potential Fields. Besides, a brief description of research works related to this Thesis is presented. Finally, general simulation concepts are included, along with a brief introduction to the experimental platform which was used.

2.1 Traffic Theory

Traffic streams can be cathegorized based on their operational perfomance. Traffic streams that operate free of traffic control policies or devices (i.e. traffic-lights, traffic signals, etc.) are known as *uninterrupted flow* and it is only influenced by the characteristics of the pathway and the interactions of the vehicles in the stream. On the other hand, traffic streams which operation is influenced by signals or any other traffic control devices are classified as *interrupted flow* [12].

Moreover, traffic flow theory involves the development of mathematical relationships among the principal elements of the streams. These traffic streams can be characterized by a number of operational perfomance measures, typically divided in *macroscopic* and *microscopic* measures.

Macroscopic measures. These measures describe the traffic stream as a whole. The macroscopic measures relevant to the aim of this research work are presented next.

1. **Traffic flow** q is the number of vehicles passing some designated spatial point during a time interval. The equation is as follows:

$$q = \frac{n}{t} \tag{2.1}$$

   Where q is the traffic flow of vehicles per unit time, n is the number of vehicles passing at a given point and t is the duration of the time interval.

2. **Density** p is the number of vehicles traveling over a unit length of highway at a given instant in time. Its equation is the following:

$$p = \frac{n}{l} \tag{2.2}$$

   Where p is the traffic density in vehicles per unit distance, n is the number of vehicles occupying some length of the roadway at some specified time and l is the length of the roadway.

3. **Speed** u is distance that a given vehicle travels during a unit of time, as depicted in the following equation:

$$u = \frac{d}{t} \tag{2.3}$$

   Where u is the speed in distance per unit of time, d is the traveled distance and t is the time period.

Microscopic measures. They describe characteristics regarding individual pairs of vehicles, within traffic streams. A microscopic concept of traffic streams, which is relevant to this research work is described below.

1. **Average traffic speed: time-mean speed** $\overline{u}_t$ the arithmetic mean of the vehicles speed as observed at a given point along the roadway. Its equation is the following:

$$\overline{u}_t = \frac{\sum_{i=1}^{n} u_i}{n} \tag{2.4}$$

   Where $\overline{u}_t$ is the mean spead in unit distance per unit time, u_i is the speed of the i-th vehicle at a designated point and n is the number of measured vehicles spot speeds.

2.2 Traditional Solution Methods for Traffic Congestion

The US Federal Highway Administration (FHWA) [1] defines traffic congestion as the level at which transportation system performance is no longer acceptable due to traffic interference. Congestions are determined by geographic features, weather, collisions, vehicles breakdowns and traffic flow policies (i.e. traffic lights, traffic signs, etc); in a congested system the following four components interact [19]:

- **Duration:** amount of time that the congestion affects the travel system.
- **Extent:** number of people or vehicles affected by congestion, and geographic distribution of congestion.
- **Intensity:** severity of congestion.
- **Reliability:** variation of the other three elements.

The search for a solution regarding this issue is not easy, since "this problem has grown too rapidly and is too complex for only one technology or service to be the solution", according to the 2005 Urban Mobility Report, in which it is also stated that major improvements in this matter can take 10 to 15 years and that any smaller efforts may not meet all the needs. So, they recommend a balanced approach: "begin to plan and design major capacity increasing projects, plans or policy changes while immediately relieving critical bottlenecks or chokepoints, and aggressively pursuing operations improvements and demand management options that are available". Important elements of this approach are listed below [25]:

- **More capacity:** In order to serve new developments, new streets and urban freeways will be needed.
- **Greater efficiency:** The more efficiently roads and public transportation are operated, the more productivity we will get from the existing system at a relatively low cost. This can be the result of educating travelers about their options or providing a more diverse set of travel and development options.
- **Manage the demand:** Modifying the way travelers use the transportation network, in order to accommodate more demand: public transportation, carpools, and traveling in off-peak hours.
- **Development patterns:** This is concerned with techniques which aim to change the way that commercial, office and residential developments occur, in a way that they can sustain the urban quality of life and gaining an increment of economic development without the typical increment of mobility decline.
- **Realistic expectations:** Large urban areas will be congested, and so will be locations near key activity centers even in smaller areas.

Additionally, the 2005 Urban Mobility Report includes the effect of four treatments designed to gain more benefits from the existing infrastructure, in order to alleviate traffic congestion. Those techniques are: *freeway entrance ramp metering*, *freeway incident management programs*, *arterial street access management programs* and *traffic signal coordination programs*. All of them provide a more regular traffic flow, and are described below [25]:

Freeway entrance ramp metering. Entrance ramp meters regulate the flow of traffic on freeway entrance ramps by the use of traffic signals. This means thah they are designed to create more space between entering vehicles so those vehicles do not collide or disrupt the mainlane traffic flow: signals allow one vehicle to enter the freeway at some interval, which can be set to two to five seconds, for instance. ***Effect:*** 5% of delay reduction.

Freeway incident management programs. Set of operations that aim to remove crashed and disabled vehicles from the lanes and, hence, reduce secondary crashes. They work in conjunction with surveillance cameras, cell phone incident call-in programs and other elements to remove these disruptions, decrease delay and fuel consumption and improve the reliability of the system. ***Effect:*** 7% of delay reduction.

Arterial street access management programs. Efforts which are directed to reduce the potential collision and conflict points, typically by turn restrictions, acceleration and deceleration lanes, among others. Such programs are a combination of design standards, public sector regulations and private sector development actions. ***Effect:*** 3.5% of delay reduction.

Traffic signal coordination programs. Traffic signaling is coordinated in a way that reduces the waiting times of travelers and trip delays. Traffic signal timing can be a significant source of delay on the major street system. Much of this delay is the result of managing the flow of intersecting traffic, but some of the delay can be reduced if the traffic arrives at the intersection when the signal is green instead of red. ***Effect:*** 1% of delay reduction.

As we can see, all the previous solutions actually contribute in congestion reduction, but in a very small percentage, being the smallest one the Traffic signal coordination programs; however, the largest amount of delay reduction, provided by the Freeway incident management programs, is still not that much greater (only 7%). This is a great motivation for the research and development of new traffic handling approaches, such as the novel algorithms described in Section 2.6.

2.3 Path Planning Methods

Path Planning is a major undertaking in robotics; it is the ability of a robot, or agent, to plan its own motions, in order to perform a trajectory from one initial point to a goal point, through a specific environment in which obstacles might exist. Some examples of path planning applications in real life are [11]:

1. **Planetary Exploration:** Autonomous robots, such as the Mars rover and the Sojourner, are used to explore other planets.

2. **Personal Transport Vehicles:** They provide pedestrians with a means of transportation in places where the pollution and noise of automobiles is undesirable. One example is the CyCab 355.

3. **Museum Tour Guides:** A robot named RHINO served as a fully autonomous tour-guide at the Deutsches Museum Bonn. It was capable to lead visitors frome one exhibit to the next by computing a paath using a stored map of the museum.

There exists several methods for solving a path planning problem, and their application depends on the task that the agent is to perform. The remainder of this section provides a brief description of some frequently used path-planning methods, namely: *Roadmaps, Cell Descomposition* and *Probabilistic Roadmaps*.

It should be remarked that only the basic version of these methods is taken into account in this document, in order to provide a general description of them. All of these methods, in their simplest approach, need a *preprocessing* phase, which makes them unfit for real-time path planning applications, such as the problem this research work is trying to solve.

Nevertheless, high-level versions of these methods have been developed, as the one conducted in [27], describing a new approach on Probabilistic Road Maps which uses "the two input query configurations as seeds to explore as little space as possible"; this, among other important features, results in a drastic reduction of planning times, making this model a far better undertaking capable of handling real-time problems.

However, it should be made clear that this research work is aimed to the design of an emergent behavior for groups of reactive agents; within the design of such behavior, it is used a path-planning paradigm based on artificial potential fields (described later on this Chapter), in order to provide agents with motion capabilities, but it is not focused in serving as a path-planner in a broad sense.

2.3.1 Roadmaps

The roadmap paradigm in path planning consists of capturing the connectivity of the agent's environment in a network of one-dimensional curves, called the *roadmap*.

Once a roadmap has been computed, it is used as a set of standardized paths. Then, path planning within this approach consists on connecting the initial and goal positions to points in the roadmap, and searching for a path between those points [16].

Several methods based on this general idea of path planning have been proposed; some of them are listed in the following:

1. **Visibility Graph:** the standard visibility graph is defined in a two-dimensional polygonal configuration space. The nodes v_i of the visibility graph include the start location, goal location and all the vertices of the obstacles in the workspace. The graph edges e_{ij} are straight-line segments that connect two line-of-sight nodes v_i and v_j.

2. **Voronoi Diagram:** the generalized Voronoi Diagram (GVD) is the set of points where the distance to the two closest obstacles is the same. Path planning is achieved by moving away from the start point until reaching the GVD, then along the double equidistant GVD to the vicinity of the goal and, finally, from the GVD to the goal. The advantage of this diagram is that it yields free paths which tend to maximize the clearance between the mobile object and the obstacles in the scenario.

2.3.2 Probabilistic Roadmaps

The Probabilistic Roadmap (PRM) approach divides planning into two phases: the *learning phase*, during which a roadmap in the workspace is built; and the *query phase*, during which user-defined query positions are connected with the precomputed roadmap. This roadmap is composed of a set of nodes located within the environment, which form a graph; the nodes of that graph are free positions in the workspace, while the edges correspond to free paths computed by a local planner. In order to find a path between a given start point and a goal point, the path planning process is reduced to connecting those points to nodes in the roadmap graph, and searching for a path between those points. This process is formally stated in Algorithm 2.1.

2.3.3 Exact Cell Decomposition

These structures represent the scneario by the union of simple regions called *cells*. The shared boundaries of cells often have a physical meaning such as a change i nthe closes obstacle or a change in line of sight to surrounding obstacles; two cells are *adjacent* if they share a common boundary. An *adjacency graph*, as its name suggests, encodes the adjacency relationships of the cells, where a node corresponds to a specific cell and an edge connects nodes of adjacent cells [11]. The process of computing *exact cell decomposition* is provided by Algorithm 2.2.

Algorithm 2.1 Proabilistic Roadmap

Step 1. Throw N independent random points in the free space of the scenario and connect any two of them that can be connected by a free straight line. The result is a roadmap G, which might have more than one connected components.
Step 2. Specify a and b, the start and the goal points, respectively.
Step 3. Connect a to one of the closest roadmap nodes in G. Once a connection is obtained, b is tried for connection to the same component of G to which point a is connected.
Step 4. If A and B are the nodes with which a and b are connected, respectively, a search on G can construct a path between A and B.
Step 5. If a or b could not be connected to two nodes of G, report failure.

Algorithm 2.2 Exact Cell Decomposition

Step 1. Partition the scenario into disjoint cells.
Step 2. Obtain a non-directed adjacency graph G: two nodes in G are connected if and only if the corresponding cells are adjacent.
Step 3. Specify a and b, the start and the goal points, respectively.
Step 4. Determine A and B, the cells that contain the start and goal, respectively.
Step 5. Search for a path from node A to node B within G.

2.4 Artificial Potential Fields Theory

The Artificial Potential Fields (APF) technique was first introduced by Khatib in [14], with the objective of achieving real-time obstacle avoidance for manipulators and mobile robots. A *manipulator* is a stationary robot, capable of affecting its environment by making use of its *end-effector*, which acts as the hand of a robotic arm, for instance. The whole philosophy of the APF method describes that the manipulator moves within a field of forces and that "the position to be reached is an attractive pole for the end-effector, and obstacles are repulsive surfaces for the manipulator parts".

The APF theory states that for any goal-directed robot in a scenario that contains stationary or dynamically moving obstacles, an APF map can be formulated and computed, taking into account an attractive force located at the robot's goal position and repulsive forces induced by the obstacles in the environment. This potential field can be expressed as follows:

$$U_{art}(x) = U_{goal}(x) + U_{obs}(x) \tag{2.5}$$

Where $U_{art}(x)$, $U_{goal}(X)$, and $U_{obs}(X)$ represent the resulting APF, the attractive potential from the goal, and the repulsive potential from the obstacles, respectively. Furthermore, x denotes a set of independent parameters, called operational coordinates, that describe the position and orientation of the robot. A possible expression of attractive potential would be:

$$U_{goal}(x) = -\frac{1}{2}k_p(x - x_{goal})^2 \tag{2.6}$$

Where k_p is a positive gain.

An example of repulsive potential is given as follows:

$$U_{obs}(x) = \begin{cases} \frac{1}{2}\eta(\frac{1}{D(x)} - \frac{1}{l_0})^2 & \text{if } D(x) \leq l_0 \\ 0 & \text{if } D(x) > l_0 \end{cases} \tag{2.7}$$

Where η is a constant and l_0 is a distance threshold, beyond which no repulsive force will be received by the robot. $D(x)$ is the distance to the closest obstacle.

Generally speaking, U_{obs} is chosen such that U_{art} is a non-negative continuous and differentiable function that tends to infinity when x approaches the surface of an obstacle and tends to zero when x approaches the goal position, x_{goal}. Given equation (2.5), the force resulting from the APF at x, can therefore be derived:

$$\overrightarrow{F}_{art} = - \bigtriangledown [U_{art}(x)] \tag{2.8}$$

The above expression tells us that applying artificial potential field $U_{art}(x)$ to a robot can be realized by using $\overrightarrow{F}_{art}$ as a command vector to control the robot in its operation space. In doing so, the joint forces corresponding to $\overrightarrow{F}_{art}$ must be obtained using the Jacobian matrix. Under such a control, the robot will be able to avoid obstacles as the repulsive force in the potential field *pushes* it away into the valleys of the field. At the same time, it can move toward a goal location as the attractive force in the potential field pulls it in the direction of a global zero-potential pole.

According to equation (2.8), equation (2.6) gives us:

$$\bigtriangledown U_{goal}(x) = k_p(x - x_{goal}) \tag{2.9}$$

Which is a vector based at x, points away from x_{goal}, and has a magnitude proportional to the distance from x to x_{goal}. So, the farther away x is from x_{goal}, the bigger the magnitude of the vector. In other words, when the robot is far away from the goal, the robot quickly approaches it, and when the robot is close to the goal, it slowly approaches it. This feature is useful for mobile robots because it reduces "overshoot" of the goal.

In the same way, equation (2.8) leads equation (2.7) to:

$$\bigtriangledown U_{obs}(x) = \begin{cases} \eta(\frac{1}{l_0} - \frac{1}{D^2(x)}) \bigtriangledown D(x) & \text{if } D(x) \leq l_0 \\ 0 & \text{if } D(x) > l_0 \end{cases} \tag{2.10}$$

Where l_0 allows the robot to ignore obstacles sufficiently far away from it and the η can be viewed as a gain on the repulsive gradient. These scalars are usually determined by trial and error.

It is important to remark that the definition of the APF that one might use for a given scenario is not necessarily attached to the shape of the potential function given in equation (2.6), as we can define our own APF, having the shape which best fits the path-planning behavior we want the mobile objects to have.

2.4.1 Gradient Descent

As we saw in the previous section, an APT function defines a gradient of forces through which the agent is attracted by is destination and pushecd away from obstacles. In order for the agent to reach its destination, it must follow a gradient-descent path towards the position of its goal; so, we present in this section the Gradient Descent algorithm which is a very well-known approach to optimization problems. The rest of the information presented on this section is based on [11].

The idea is simple: starting at the initial position, take a small step in the direction opposite the gradient. This gives a new position, and the process is repeated until the gradient is zero. More formally, we can define a gradient descent algorithm like:

Algorithm 2.3 Gradient Descent

Input: A means to compute the gradient $\bigtriangledown U(p(i))$ at a point p

Output: A sequence of points $\{p(0), p(1), ..., p(i)\}$

$p(0) = p_{start}$

$i = 0$

while $\bigtriangledown U(p(i)) \neq 0$ **do**

  $p(i+1) = p(i) + \alpha(i) \bigtriangledown U(p(i))$

  $i = i + 1$

end while

In algorithm 2.3, the notation $p(i)$ is used to denote the value of a position p at the ith iteration and the final path consists of the sequence of iterates $\{p(0), p(1), ..., p(i)\}$. The value of the scalar $\alpha(i)$ determines the step size at the i iteration. It is important that $\alpha(i)$ be small enough that the robot is not allowed to "jump into" obstacles, while being large enough that the algorithm does not require excessive computation time. In motion planning problems, the choice for $\alpha(i)$ is often made on an *ad hoc* or empirical basis, perhaps based on the distance to the nearest obstacle or to the goal. Finally, it is unlikely that we will ever exactly satisfy the condition $\bigtriangledown U(p(i)) = 0$. For this reason, this condition is often replaced with the more forgiving condition $\| \bigtriangledown U(p(i)) < \epsilon \|$, in which ϵ is chosen to be sufficiently small, based on the task requirements.

This Gradient Descent algorithm will make the mobile agent navigate towards a direction in which the potential field attraction force is maximally increased, and, since the source of this attraction force is represented by the desination of the mobile agent, it will reach that position at the end of the path.

2.4.2 Potential-Guided Path Planning

The APF technique enables us to develop path planners for both single and multi-agent scenarios; by taking advantage of its attraction/repulsion paradigm, we can ensure that a given mobile agent reaches its destination, without colliding against any obstacles in its path. In this section we present a path planning approach based on APFs, which can be consulted further in [16].

There are some simple potential-guided path planning techniques. In principle, these techniques do not assume any specific potential function. Hence, they are applicable with the potential function defined in the previous sections, and with other potential functions as well.

In its original conception, the potential field approach to motion generation consists of regarding the agent or robot in the scenario as a unit mass particle moving under the influence of the force field $\overrightarrow{F} = -\overrightarrow{\bigtriangledown} U$.

This way of using the potential function is applicable for generating paths on-line. It is well-suited when the obstacles are not know in advance, but sensed during motion execution. If a prior model of the obstacles is available, the same method can be used to plan a path by simulating the motion of the particle. However, in this case, there exist simpler and more efficient path planning techniques using potential field.

One of these techniques generates a path in a "depth-first" fashion, without backtracking. Like on-line generation, it may be very fast in favorable cases, but it may also get stuck at local minima of the potential function. Another technique operates in a "best-first" mode. It deals with local minima by "filling" them up. The third technique consists of optimizing a functional constructed by integrating the potential along a complete path between the initial and the goal configurations.

2.5 Group Behaviors

The simulation of emergent group behaviors, such as flocking, has been widely used in creating realistic animations for groups of virtual agents such as birds, or *boids*, as they were named by Craig W. Reynolds in [23]. These animations are done by designing and implementing simple motion rules on each element of the group, which are later translated into a so-called *emergent behavior* of the whole group of agents.

This is an early application of artificial formation behavior, which proposes a simple egocentric behavioral model for flocking which is embedded in each member of the group; the contribution of Reynolds' research work is the generation of successful overall group behavior, by means of every individual agent sensing and reacting to their local environment and close neighbors. In spite of the fact that this idea was first conceived for computer animation, it has been extended and used to control flocks of robots, which usually show homogeneous behavior.

Reynolds's research work showed that flocking is a dramatic example of emergent behavior, in which global behavior arises from the interaction of simple local rules, which tell the agent it should move with its neighbors. In order of decreasing precedence, such rules are:

1. **Collision Avoidance:** avoid collisions with nearby flockmates.
2. **Velocity Matching:** attempt to match velocity with nearby flockmates.
3. **Flock Centering:** attempt to stay close to nearby flockmates.

Collision avoidance and *velocity matching* are complementary. They ensure that the members of the flock are free to fly to the interior of the group without running into one another. Collision avoidance directs the boid away from an imminent impact, while velocity matching ensures that the separations between boids remain approximately the same at every moment, with respect to ongoing geometric flight. This is, with these rules a flock of birds can fly establishing a minimum required separation between them, and being able to maintain it.

Furthermore, *flock centering* makes a boid want to be near the center of the flock. Because each boid has a localized perception of the world, "center of the flock" actually means the center of the nearby flockmates. Flock centering causes the boid to fly in a direction that moves it closer to the centroid of the nearby boids. If a boid is deep inside a flock, the population density in its neighborhood is roughly homogeneous; the boid density is approximately the same in all directions. In this case, the centroid of the neighborhood boids is approximately at the center of the neighborhood, so the flock centering urge is small. But if a boid is on the boundary of the flock, its neighboring boids are on one side. The centroid of the neighborhood boids is displaced from the center of the neighborhood toward the body of the flock. Here the flock centering urge is stronger and the flight path will be deflected somewhat toward the local flock center.

These simplistic rules have proved efficient in building realistic simulations of flocks, which behavior is very similar to that present in real flocks of birds.

2.5.1 Social Potential Fields

Potential Fields methods have been shown to be powerful in solving difficult path planning problems; within a multi-agent scenario, it is frequently needed that the agents develop some sorth of coordination and communication mechanisms, which are usually achieved by exchanging messages between them, through a communication protocol. However, when this scenario deals with hundreds or thousands of agents, to exchange messages and to process all of them is very time-demanding and results in communication delays and coordination inaccuracies.

So, the *Social Potential Fields* approach [22, 2] aims to determine control laws within a distributed-control framework. In this approach, each agent in the scenario senses the resultant potential field from all other components (i.e. agents, obstacles, or objectives), or at least the neighboring components, and acts under the resultant force. Once these force laws are defined, force calculations can be carried out by individual agents in a distributed manner and, thus, the control is completely distributed.

Using these force laws, the resulting system displays “social” behaviors such as clustering, guarding, escorting, patrolling and so on; this is where the term *social potential fields* comes from. Furthermore, interaction of forces between groups of robots can be achieved by extending this approach and thus processing a resultant potential field from all other *groups* of agents.

This resultant potential field is called a *global control force*, because it coordinates the agents and determines the distribution and direction of them within a system. Besides, in this model are also taken into account the *local control forces*, which are those enabling the agents to avoid collision with obstacles or to approach a given object or goal position. Finally, when the combined force is computed by a given agent, there are several ways that its motion can be controlled by that force; for example, it can gain an acceleration proportional to that force, or it can move in the direction of it, either for a length proportional to the magnitude or for a fixed length, as in Algorithm 2.3 for Gradient Descent.

Additionally, in order to design these potential force laws to achieve given behaviors, it has been proposed the following hierarchical methodology:

1. **Required Behavior:** specify the various groups of agents and their interactions.

2. **Desing the Potential Laws:** applying this hierarchical methodology for designing potential force laws, taking into account:

   (a) ***Intra-group social force laws*** among individuals of each given agent group.

   (b) ***Inter-group social force laws*** between individuals of distinct groups.

2.6 Related Work

Recently, new location-based technologies have taken advantage of the Global Positioning System (GPS) and mobile or wireless communications; this has motivated new research efforts with the underlying objective of developing novel ways of handling traffic, that can be implemented in real scenarios with the support of these emerging technologies, as well as Artificial Intelligence techiniques.

For instance, Kurt Dresner and Peter Stone [9], at the University of Texas at Austin, made use of wireless communications, in order to develop an intersection control mechanism based in a reservation protocol; in this model, a given vehicle negotiates individually with an intersection, for it to make a reservation of the time slots that it will need to pass through.

In a similar way, the use of the GPS and cellular communications technologies allowed Carlos Astengo and Ramón Brena [4] to envision and develop the *Flock Traffic Navigation Based on Negotiation* model, in which vehicles navigate in groups called "flocks", for which the use of coordination mechanisms between them is necessary.

Both models apply Multiagent Systems (MAS) theory as their underlying development schema. MAS are systems composed of multiple intelligent agents, characterized by being autonomous within the environment they are in, and able to interact with other agents via negotiation, coordination, cooperation, etc. The two of them are going to be described further on this section.

2.6.1 Multiagent Traffic Management: A Reservation- Based Intersection Control Mechanism

The information of this section is based on the research work of Kurt Dresner and Peter Stone [9], which is aimed to increase the efficiency related to moving cars through an intersection with minimal infrastructure, noting that vehicles are driven by a central computer.

In this model, it is assumed that intersections have a wireless communication system, through which they can send messages to the vehicles and receive messages from them as well; for this communication to happen, the authors of this model developed a specific protocol, which tells the cars whether or not they are allowed to pass through it, just the way it happens nowadays with red and green lights. However, agents preserve their autonomy, since they are allowed to drive the way they decide.

The intersections are divided into a $n \times n$ grid of reservation slots, where n is the granularity of the reservation system. As it is expected, each one of the slots can be reserved only by one car at any time step. Cars travel at the speed limit, and reserve these slots by sending messages containing several parameters, such as: the time they will arrive to the intersection, their velocity, the direction they will face when reaching the intersection, among others.

This research work models a single intersection, and using the parameters described above, it simulates the trip of the vehicle across it, taking into account which slots will be occupied by it at each time step, considering a few time steps before and

	Reservation			Light	
Lanes	**Gran.**	**Avg**	**Max**	**Avg**	**Max**
1	1	0.016	0.912	5.847	15.526
2	2	0.017	0.925	5.488	15.536
3	3	0.023	1.435	5.482	15.506
4	4	0.019	1.590	5.351	15.536
5	5	0.031	1.902	5.439	15.506
6	6	0.025	1.926	5.378	15.517

Table 2.1: Average and Maximum delays for the simulation of Reservation and Traffic Light systems

after for sefety reasons. When any of these slots is reserved, the system rejects the vehicle's reservation request and so, it has to decelerate and try again in the next time step. On the other hand, if the vehicle realizes that it can't attend the reservation, it gets it cancelled and the process begins again.

In order to test the model, a simulator was developed, presenting an area with dimension $400m \times 400m$, in which lanes are 3.5 m wide and each car dimension is 2 m wide by 4 m long. This simulator runs having the following two main parameters:

- Number of lanes in each of the four directions of the intersection.
- The probability or likelihood to generate a new vehicle in each of the directions at every time step.

Within this simulator, the reservation-based model performance was compared against both the overpass and the traffic light intersection control policies. At this point, it is important to mention that, in the reservation-based model, the ability to turn is not required, and so the overpass is an optimal solution, since cars never have to decelarate at the intersection; the only delays in the overpass are produced by cars traveling in the same direction, but, since in this model vehicles travel at the speed limit, they are never required to slow down due to this matter and results in a delay of 0. Thus, the overpass is taken as the lower bound for the simulation perfomance.

In order to compare the reservation-based model against the traffic light policy, both approaches were tested in different scenarios, varying the number of lanes up through a 6×6 intersection, as well as the granularities of the reservation system; results from those experiments are presented in Table 2.1, which presents the Average and Maximum delays for a simulation of 1,000,000 steps for both models, having a traffic light system with period of 20 seconds and a car-generating probability of 0.001. We can see that the reservation model clearly outperformed the traffic light system in all cases.

Furthermore, Dresner and Stone have proposed four main improvements for this initial approach in [8]. Such improvements are listed in the following:

1. It augments the proposed intersection control mechanism to allow for more flexible vehicle control, including turning from any lane and accelerating while in the intersection.

2. It introduces a detailed protocol by which vehicles and intersection managers can communicate and coordinate their actions.

3. It describes a driver agent that makes good use of this protocol.

4. It demonstrates how this augmented system, by using the proposed protocol, can still outperform traffic control mechanisms such as traffic light and stop signs.

The objective of these enhancements is to transform the initial reservation-based model into a more realistic and implementable system.

2.6.2 Flock Traffic Navigation Based on Negotiation

All information presented on this topic is based on the research work of Astengo and Brena [4], called "Flock Traffic Navigation Based on Negotiation" (FTN, for short), which proposes a new way of handling traffic through groups of vehicles called flocks, inspired by the way birds, animals and fish coordinate themeselves to gather up and travel with less effort within flocks, herds or schools.

The reservation-based model explained previously results in traffic merging at intersections; in high density traffic, this represents a dangerous scenario, since individual cars are being alternated at intersections and external factors such as mechanical failures might lead to accidents, even if the model performs as accurate as possible. On the other hand, FTN is a traffic handling model in which intersections deal, not with individual vehicles, but with groups of them, making the process far safer than the reservation-based model. The advantage of traveling together is that, the greater the group is, the greater the chances it has to pass through an intersection; besides, they get a "social bonus" which allows the group to increase their speed.

In order to enable the creation of flocks, a coordination mechanism is performed between navigating vehicles. At the very beginning, flocks have just two members and then additional ones join them; any "two partner candidates search for a meeting point to travel together until they reach a certain point from which they no longer bennefit from doing it", and so they split to continue traveling on their own. Figure 2.1 presents the so-called "bone structure", which depicts the appearance of this kind of trip.

In order for the vehicles to evaluate the convenience of traveling together, they compute the time it will take to get to its destination if they travel individually, considering their own initial and ending points. Afterwards, they search for partners within

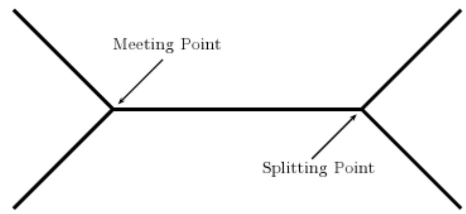


Figure 2.1: Bone Structure.

a specific region, and all agents in it respond by sharing their information; this information is used to build a bone structure for each pair of vehicles, from which they can compute the new trip time, considering the social bonus.

Then, each agent compares its individual time against the time for traveling with a given partner; in the case that the individual time is less than the joint time, there is not going to be any relation between those agents. Otherwise, the complete travel time is going to be computed, consisting on the individual time to get to the meeting point, the joint travel time considering the bonus, and the individual time to get to the destinatioin, after the splitting point.

When several candidates are found, the vehicle now has to find out which one provides the minimum travel time. If two given vehicles A_i and A_j represent the best option for one another, then we found a Nash equilibrium strategy (there is no better option for neither of the two vehicles) and they must travel together; if only A_i represents the best option for A_j, and the converse is not true, A_i is placed in a "Pareto set" and waits for A_j to find a partner with Nash equilibrium, in which case A_i will have to recalculate its travel time considering the bone structure of those two vehicles. Finally, if it finds the new bone structure of those vehicles not better than the individual time, it searches for the second best partner; if it doesn't find any partners it must travel alone without the social bonus.

The overall experimental results of this model are presented in Table 2.2; each experiment was replicated one hundred times, and vehicles were generated uniformly in a 5×5 block area, and targets in a 20×20 city sector. We can see that the computational time to process this model with 100 agents (i.e. vehicles) is nearly one second, and increases with a greater amount of them. If we suppose vehicles travel at 100 kilometers per hour, the previous processing delay will result in agents receiving negotiation messages aproximately 27.7 meters later, if the amount of traffic is equal to 100 agents, and much more if this amount is greater. This might be too late to resume negotiations and coordination activities in an accurrate way.

Number of Agents	Saved Time (ST)	Variance of ST	Computational Time (CT)	Variance of the CT	Maximum ST	Minimum ST
2	3.17	34.7688	0.0149	0.0004	20	0
5	12.29	44.1423	0.0146	0.0003	30	0
10	24.175	78.3251	0.0222	0.0004	51	6
25	42.15	141.1389	0.0769	0.0003	80	19
50	64.705	252.1217	0.2489	0.0005	119	37
75	85.41	319.613	0.5258	0.0006	138	48.5
100	100.915	398.374	0.9116	0.0005	144.5	44.5
250	204.13	1320.3	5.4622	0.00001	308.5	130.5
500	366.575	2172.4	21.7039	0.00001	468.5	264

Table 2.2: FTN total time savings

2.7 Simulation Concepts

Simulation is defined as "the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output" [17].

As for traffic simulations, there are two types of models [7]: continuous space models and discrete time approximations based on differential equations (*macrosimulation*) or discrete space and time representations based on decentralized autonomous systems (*microsimulation*).

Traffic macrosimulation involves the simulation of general aspects of the system, like average car velocities, car density, car flow, etcetera. In order to accomplish this, mathematical models that describe each one of those variables are used ([10] and [21]). The disadvantage of macrosimulation is that it assumes that cars are similar (i.e. homogeneous), ignoring fine details like modeling of individual features of them.

Microsimulation, as mentioned in [10] and [21], models every element in a separated way, allowing individual elements to interact with others. For instance, elements in urban traffic simulation could be vehicles themselves. Every car has a number of specific parameters like length, width or maximum allowed velocity. In this way, the urban traffic can be seen as a collective behaviour generated by cars. Agent-based simulation is a type of microsimulation, because an agent can be seen as the so-called "element" mentioned in [10] and [21] or the decentralized autonomous system mentioned in [7].

Agent-based simulation is defined as a simulation made up of agents, objects, or entities that behave autonomously. These agents are aware of (and interact with) their local environment through simple internal rules for decision-making, movement and, action. Agent-based simulation has been proposed for many situations involving a large number of heterogeneous individuals, such as vehicles and pedestrians in traffic, people in crowds, artificial characters in computer games, agents in financial markets, and human and machines on battlefields. The aggregate behaviour of the simulated system is the result of the dense interaction of the relatively simple behaviours of the individual simulated agents [24].

Now, in order to develop a multi-agent simulation like the one that this research works needs as an experimental framework, we can use one of the several multi-agent platforms that are available. The reason is that a multi-agent platform has already implemented many of the core functionalities needed by agents, instead of building the entire system from scratch. Furtermoe, most multi-agent plaforms can be easily integrated with other libraries, thus speeding up the implementation phase.

For this reaserch work, we have selected **NetLogo**, which is a cross-platform multi-agent programmable modeling environment for simulating both natural and social phenomena. It was authored by Uri Wilensky in 1999 and is in continuous development at the Center for Connected Learning and Computer-Based Modeling. It is particularly well suited for modeling complex systems developing over time, making possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many of them.

A list of its most important features is provided in the following [30]:

- **System:** NetLogo is cross-platform, as it runs in Windows, Mac or Linux. Runs are exactly reproducible cross-platform.
- **Language:** It is fully programable and provides a simple language structure, which is a *Logo* dialect extended to support agents; it offers a large vocabulary of built-in language primitives.
- **Environment:** The created model can be viewed in either 2D and 3D. It provides a *command center* for on-the-fly interaction, as well as an *interface builder* with buttons, sliders, switches, choosers, etcetera. Furthermore, NetLogo offers a powerful and flexible plotting system to analyze the desired data, and *agent monitors* for inspecting and controlling agents. Besides, the *BehaviorSpace* tool can be used to collect data from multiple runs of a model and the developer is able to export and import functions (export data, save and restore state of model, make a movie, etcetera).
- **Web:** Models can be saved as applets to be embedded in web pages.

2.8 Summary

This Chapter presents the main theoretical concepts, which are closely related to this research work. The first section describes some Traffic Theory information, along with specific parameters that are of interest for the proposed solution model; then, the most common traditional solutions for traffic congestions are presented. Later, related path planning approaches were introduces, along with the actual technique that was used. It was important to complete this theoretical explanation with the description of the *group behaviors* concept, derived from the SPF approach, which this research work uses as the underlying design method. Besides, the description of related work was also important to present previous efforts undertaken in this area. Finally, relvant simulation concepts are provided, along with a general description of the selected simulation platform.

Next Chapter provides an in-depth description of the Traffic Simulator, which was developed to sever as an experimental platform for this research work.

Chapter 3

Solution Model

The scenario being modeled in this research work deals with the traffic problem within large cities, and the congestions that often follows as a result. So, this can be described as a multiagent scenario, in which vehicles are considered intelligent agents that will act as stated by the *group behavior* that has been designed, for them to organize themselves as flocks of vehicles that travel to close destinations, taking the FTN as the underlying paradigm of urgan navigation.

The design of this group behavior is based on attracting and repulsion points that a given agent will sense along his path to its destination, and that eventually will affect such a path. The most important of these attracting points is the destination of a given vehicle; this is, during his trip, no matter what other attracting or repulsion points can be present at a given time in the path of a vehicle, the destination remains as the strongest attraction that the vehicle senses, and so it will try to reach it at every time. This prevents vehicles from traveling from one site to another just following their flockmates; rather, it lets vehicles gather up into groups while navigating towards their destination.

The other attracting points that a given agent senses are other vehicles that are heading towards a place near its destination. This will make agents move into flocks or groups of vehicles, which is consistent with the general objective of my thesis work, and eventually will help them get to its destination with the lowest trip delay, as they do not stop at the intersections, if the size of the flock they are travelling in is sufficiently large, as it will be described further in this document.

Furthermore, each vehicle induces a repulsion force to other vehicles when they get closer to them than a certain distance; the objective here is to prevent vehicle collisions within the proposed navigation model. Besides, this repulsion forces will be useful when redirecting traffic around congested streets, in which case, vehicles within a given congested street will induce a repulsion to other approaching vehicles, thus, congestion-worsening is prevented.

Acceleration and velocity modulation is taken into account, enabling vehicles to speed-up to meet their flockmates or to slow-down to avoid collisions, based upon the attraction/repulsion paradigm of the rFTN. To achieve this, the rFTN proposes a novel approach which is more convenient than other APF-based models.

3.1 Overall Description

The rFTN is a traffic management approach, in which vehicles will be attracted or repulsed by other elements in their environment, as it will be explained further, with the objective of gathering vehicles into flocks and, hence, exploiting the advantages of the FTN model, such as decreasing waiting times and preventing congestions. However, the rFTN is meant to take less computational time to be processed, than the FTN, given its reactive nature. In this section, it is provided an overall description of the rFTN model, while discussion of its technical details is delayed for the following sections.

First, let us analyze a single-agent scenario of this model. Figure 3.1, presents a diagram of such a scenario, in which one given agent is heading towards his destination, labeled T_1, by way of an attractive force which is called by the literature an Artificial Potential Field (APF, for short), as I described previously in section 2.4 of this document. Waves in that figure represent the *equipotential contours* of the APF, along which the attraction force is the same.

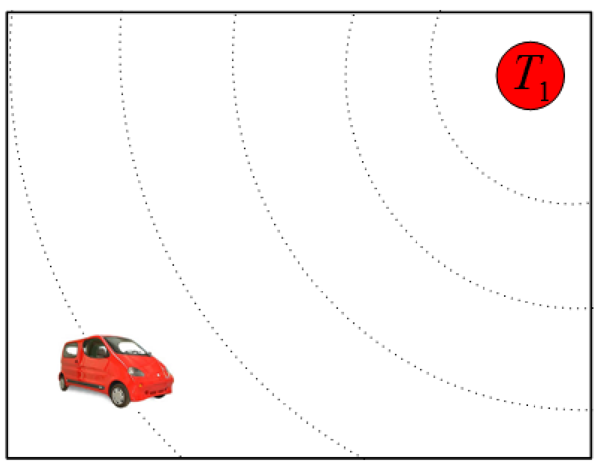


Figure 3.1: Single-agent scenario. Waves represent equipotential contours of the Potential Field.

This model is very straightforward, but if we introduce to the scenario the streets and blocks, it becomes more challenging. In Figure 3.2, we can see that, as for the rFTN, the potential-field-based path planning process will have to take into account that the city blocks act like restrictions of the environment (i.e. vehicles cannot go through them). So, every time a vehicle reaches an intersection, it will compute the resulting attraction force and then it will update its heading towards the neighbor intersection located in the direction stated by resulting force vector. That is, a given vehicle computes the attaction/repulsion forces at any time, which determine if the vehicle should speed-up or slow-down, but it will change its heading only when arriving to an intersection; this will prevent vehicles from taking *U-turns* at the middle of the streets, which is not a proper way of driving. Furthermore, for this entire research work the city is composed of *two-way* streets, of only one lane per way.

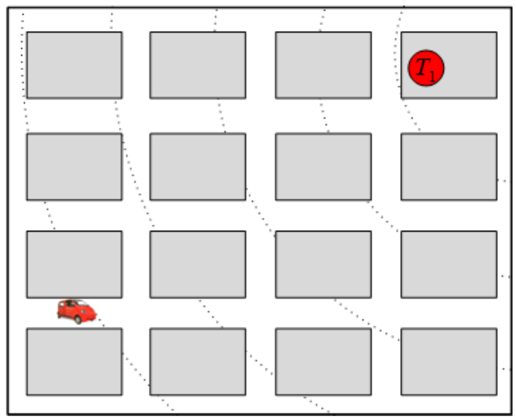


Figure 3.2: Single-agent scenario with blocks.

Now, let us analyze a multiagent scenario of this model, like the one which is depicted in Figure 3.3. Every agent in the rFTN will be attracted by other vehicles, only if they are heading towards a destination which is *vectorially* near to the destination of the given agent. According to the FTN terminology, vehicles that comply with the previous description will be called *flockmates* within the rFTN.

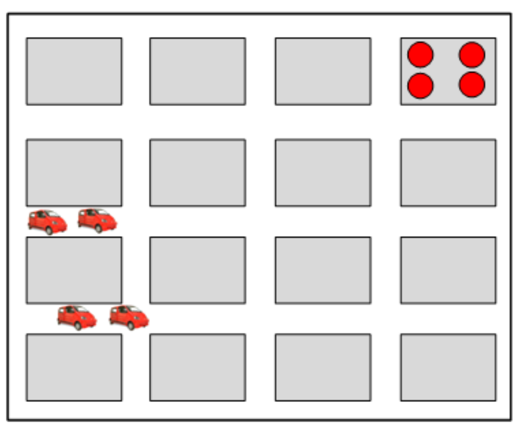


Figure 3.3: Multiagent scenario showing a group of vehicles and their targets.

The direction that vehicles will follow in the presence of flockmates is going to be determined by the resulting sum of the attraction forces induced **both** by their individual destination and the rest of the group. If no proper flockmates are detected, they will follow their path towards their destination by their own, which means that it is not convenient to gather up with other vehicles, as flocking will not bring an advantage (i.e. there is no reason to choose a vehicle heading North as a flockmate, when I am heading South); rather, it will only make vehicles move away from their destinations.

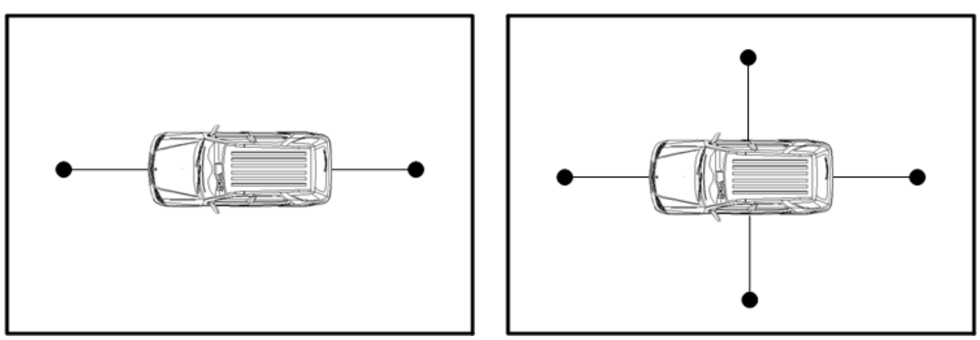
Once vehicles are gathered up, we have to make them maintain their *flocking formation* along the streets; the design of this formation in the rFTN meets the following two requirements:

1. To make vehicles stay with their group, hence, serving as an adherence rule to the flock.

2. To make vehicles navigate on the street lanes in an ordered way, avoiding inappropriate formations: a vehicle located at the middle of two lanes would be chaotic, for instance.

Without this *flocking formation* paradigm, the rFTN's attraction/repulsion nature could make the gathering of flockmates result in a bunch of vehicles crowding the streets without any order. On the other hand, the so-called Bone Structure proposed in the FTN model (see Figure 2.1) is robustly maintained.

The way the rFTN approaches this matter is by way of establishing specific *virtual attachment points*, to which flockmates will sense an attraction that will make them stay near that position. This attachment points are located at the front and at the rear of vehicles when they are navigating within a one-lane street, and an additional pair of them can be added at both sides of vehicles for this formation paradigm to be extended to two or more lanes. The distance a from the vehicle to this points can be conveniently set, so that vehicles do not get too close to each other. These one-lane and multiple-lane attachment points are depicted in Figure 3.4(a) and Figure 3.4(b), respectively. This paradigm is similar to that proposed in [2].



(a) Attachment points for one lane. (b) Attachment points for multiple lanes.

Figure 3.4: Attachment points.

Furthermore, in order to avoid collisions, the rFTN model establishes a *repulsion zone* that surrounds each vehicle. The repulsion capability of this zone starts at the boundaries of the vehicle and ends at a given distance d; within this repulsion zone, it is defined a *safety zone*, which acts from the boundaries of the vehicle to a certain distance s, as depicted in Figure 3.5. Beyond d no repulsion is induced, while vehicles entering this zone will sense an increasing repulsion until they get to the *safety zone*, within which its magnitude will be ∞. A given vehicle computes this repulsion force taking into account only other vehicles driving on lanes with the same way, and do not take into consideration those navigating the other way.

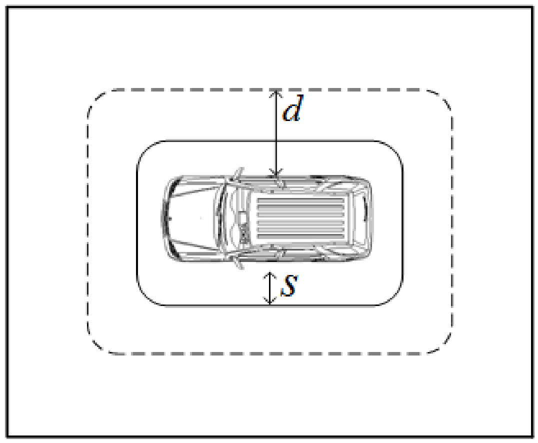


Figure 3.5: Upper-view of a vehicle. The dashed contour delimits the repulsion zone of vehicles, while the solid contour delimits the safety zone.

Now, another important feature of this model is that it measures the density of vehicles within each of the streets, so that the path planning mechanism of the vehicles can be aware of that and be redirected to an alternative (and less congested) street, as we can see in Figure 3.6. This is, the congested streets will produce a repulsion force to the given agent or agents that are planning a path to their destinations. This issue is not taken into account within the FTN model, and it is important to deal with, since, otherwise, the mechanism of flocking vehicles could itself lead to congested streets or worsening already-congested streets.

Finally, rFTN intruduces an efficient management approach at intersections. The objective here is to let larger flocks go through the intersections with almost no need to stop, just as if there was an overpass located at the intersection; the motivation of doing this is that it will decrease the trip delays.

3.2 Model Design

Having the description presented in section 3.1 as a background, now this section provides the technical details by which the rFTN features are achieved. The hierarchical methodology stated in [22] is followed, in order to provide a detailed and systematic

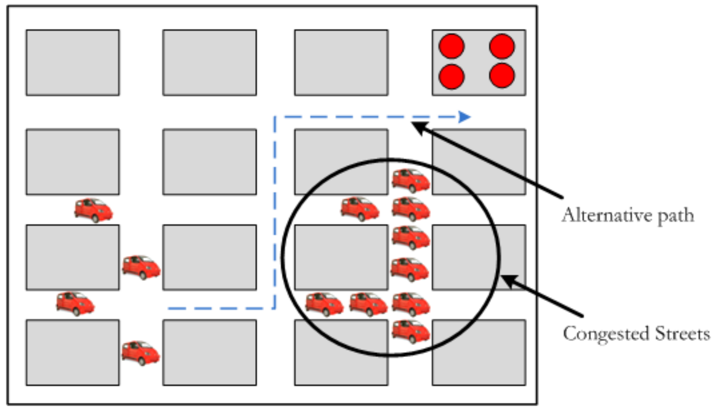


Figure 3.6: Redirecting traffic to alternative paths, when there are congested streets.

description of the potential force laws within the rFTN model. For more information about this methodology please refer to section 2.5.1.

First, it is stated in a formal way the *required behavior* of this model, which has been commented several times throughout this document. Then, introduce both the *intra-group* and the *inter-group* social force laws are introduced, upon which the navigation mechanisms are based.

Intra-group forces within the rFTN are those which let vehicles perform the following features:

1. Flockmates selection.
2. Reactive creation of the Bone Structure.
3. Navigate towards destination.
4. Flocking formation.
5. Collision avoidance.

As for the inter-group forces, the rFTN features that they enable are:

1. Intersection management.
2. Congestion avoidance.

3.2.1 Required behavior

Agents in the scenario will follow a path towards their individual destination, which will be directed by a potential force located in that position; besides, vehicles will gather up into flocks with other agents heading towards a place vectorially near their individual destination, by way of another potential force induced by such vehicles; their navigation within a flock will comply with specific formation restrictions, as well as collision avoidance.

Furthermore, flocks will be able to detect congestions and to avoid them by sensing a repulsion force. Finally, flocks will have to reactively coordinate their pass through intersections, taking into account the size of each flock, in order to compute a repulsion force.

3.2.2 Intra-group Social Force Laws

This section presents an in-depth description regarding the rFTN features that are computed among flockmates, called intra-group social force laws.

First, vehicles will perform ***flockmates selection*** by computing an attraction force induced by those vehicles within a defined *vision range*, which is actually like a radar of neighbor cars; all vehicles inside this vision range are then *flockmate candidates* which are tested to know the magnitude of attraction that they are going to induce. Candidates that do not induce any attraction are not members of the flock, while candiates with a high attraction magnitude are the best flockmates and, consequently, are the ones to which the potential force will direct vehicles computing this test. The previous process is a *reactive analogy* of the negotiation mechanism performed in the pure FTN to create flocks of vehicles. This attraction force is given the following potential field:

$$U_{flockmates}(x) = \sum_i \frac{1}{2} k_p (x - x_i)^2 \tag{3.1}$$

If we compute the proper derivatives, equation (3.1) results in the following force law:

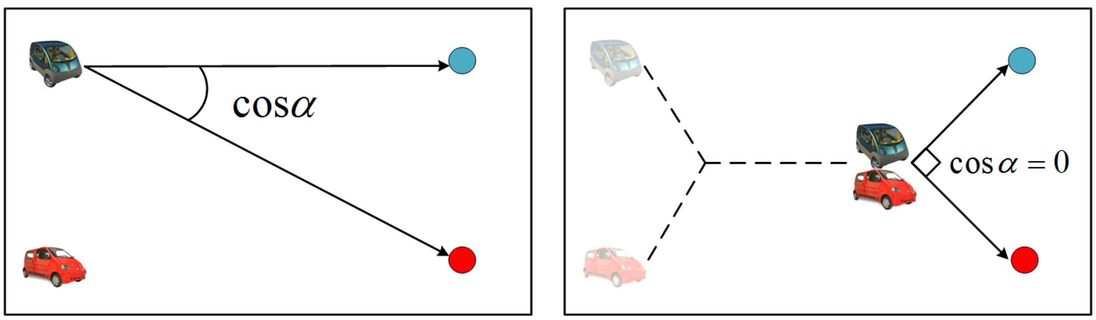
$$F_{flockmates}(x) = -\sum_i k_p (x - x_i) \tag{3.2}$$

Where k_p is a positive position gain, x is the current vehicle's position and x_i is the position of flockmate i. The summation is over the force induced by all flockmates located within the vision range. The scalar k_p is determined by the following *test*: assume agent a is computing the force induced by its flockmate candidate b and let x_a be the position of agent a, t_b be the position of the destination of agent b and let t_a be the position of the target of agent a; then k_p is determined by the *cosinus* of the angle between the vectors $t_b - x_a$ and $t_a - x_a$, as it is shown in Figure 3.7(a).

It should be remarked that the *cosinus* function varies from -1 to 1, but **this *test* only takes into account agents whose destinations are** 0 **to** 90 **degrees away**, and so we can see that the position gain assigned to each candidate's force varies from

0 to 1, if the given candidate is heading 90 or 0 degrees away from my destination heading, respectively. Notice that the latter case represents an agent heading the same direction, and so it has to be assigned a grater gain (i.e. it represents the best flockmate), in which case the attraction force will direct the agents towards each other until they reach a reactively created *Meeting Point*. On the other hand, the former case represents an agent which is heading to a place far away from my destination, and so, it is not capable of affecting my path. Vehicles heading more than 90 degrees away **are not considered elegible as flockmates**, since their desination is located completely in the opposite direction, and so they will not be taken into account for this process.

Besides, it should be noticed that the k_p factor makes the force induced by flockmates *fade away* in a convenient way, as it will get *weakened* as the vehicles approaches their destination, since the angle between those vectors is increased, which is graphically shown in Figure 3.7.



(a) Testing candidates. This results in the Meeting Point.

(b) Splitting Point.

Figure 3.7: Reactive Creation of the Bone Structure.

On the other hand, vehicles should be able to ***navigate towards their destination***, either with their flockmates (if any), or on their own, when no flockmates were found. This, obviously, is a very important feature within the rFTN or other related navigation models. The potential equation that enables this feature is defined as:

$$U_{destination} = k_t \frac{1}{r_t} \tag{3.3}$$

Once again, the scalar k_t represents the strength of the field and r_t is the euclidean distance between the agent and its destination or target. Let x denote the position of the agent and let x_t denote the position of its target t. The distance r_t is then given by $r_t = |x_t - x|$ (the norm of the vector $x_t - x$).

Since $U_{destination}$ is defined in terms of r_t, and r_t is defined in terms of the vector x, we can use the chain rule in order to compute the resulting force, as follows:

$$F_{destination} = -\frac{d\mathrm{U}_{destination}}{d\mathbf{x}} = -\frac{d\mathrm{U}_{destination}}{d\mathrm{r}_t} \cdot \frac{d\mathrm{r}_t}{d\mathbf{x}} \tag{3.4}$$

We insert the appropriate derivates to obtain:

$$F_{destination} = -k_t \frac{1}{r_t^2} \cdot \frac{\mathbf{r}_t}{r_t} \tag{3.5}$$

Where $\mathbf{r}_t$ is the vector $\mathbf{r}_t = x_t - x$, between the positions x (of the agent) and x_t (of the target).

As we can see, equation (3.3) describes a pontential which gets stronger as the vehicle approaches its destination (i.e. the Euclidean distance is reduced). When the vehicle is far away from its target the force induced by it is weak and the vehicle is more likely to deviate its path towards a greater gradient like the one induced by other vehicles, stated in equation (3.1). Later, when the vehicle gets closer to its destination, the potential induced by equation (3.3) gets stronger and, since the attraction force induced by equation (3.1) fades away, the vehicle updates its heading towards the destination, which represents the strongest attraction force at that point.

In this way, the potential forces that are sensed by a given vehicle are "deformed", thus enabling to reactively build the so-called "Bone Structure" within the FTN model, depicted in Figure 2.1, without the need of exchanging messages between agents, and thus, describing a model lighter to process.

Now, the rFTN proposes that vehicles within a group should maintain a specific formation, so that they can keep an ordered navigation on the streets. Hence, it is introduced the *attachment points* paradigm, which are specific attraction sites conveniently located around the vehicles (see Figure 3.4), for flockmates to get attached to them. Additionally, this ***flocking formation*** paradigm contributes to the adherence of flockmates to their group, thus, maintaining the Bone Structure all the way from the Meeting Point to the Splitting Point.

The attraction force that these attachment points induce is given by the potential field definition given in equation (3.6), which derives in the force law stated in equation (3.7).

$$U_{ap}(x) = \frac{1}{2} k_{ap} (x - x_{ap})^2 \tag{3.6}$$

$$F_{ap}(x) = -k_{ap}(x - x_{ap}) \tag{3.7}$$

Where k_{ap} is a fixed positive position gain, x is the vehicle's position and x_i is the position of the nearest attachment point of all flockmates.

To keep a realistic and safe navigation model, ***collision avoidance*** is taken into account by inducing a repulsion force within a *repulsion zone* and also a *safety zone*, that surround each vehicle as explained in the previous section and depicted in Figure 3.5. If the Euclidean distance between two given vehicles is r, the repulsion potential that enables this features is given by:

$$U_r(x) = \begin{cases} 0 & \text{if } r > d \\ -\frac{1}{2}k_r(x - x_v)^2 & \text{if } s < r \leq d \\ \infty & \text{if } r \leq s \end{cases} \tag{3.8}$$

Where x is the current position of the vehicle computing this feature, x_v is the position of the approaching vehicle and d and s are the distance limits of the *repulsion* and *safety* zones, respectively. Equation (3.8) derives in the following force law:

$$F_r(x) = \begin{cases} 0 & \text{if } r > d \\ k_r(x - x_v) & \text{if } s < r \leq d \\ \infty & \text{if } r \leq s \end{cases} \tag{3.9}$$

It can be seen that the repulsion force increases linearly from the boundaries of the *respulsion zone* all the way to the boundaries of the *safety zone*, inside which the force is ∞.

Now, in order to avoid problems regarding purely reactive navigational models, such as local minima or cyclic behavior, the rFTN makes use of a ***noise*** force law as stated in [2]. This is a unitary force, as stated in equation (3.11), derived from the potential equation (3.10):

$$U_n(x) = -k_n x \tag{3.10}$$

$$F_n(x) = k_n \tag{3.11}$$

The direction of this force vector is given randomly betwen 0 and 2π and is computed every specific amount of time. For the rFTN it is calculated only at intersections, when vehicles are allowed to change its heading.

3.2.3 Inter-group Social Force Laws

The inter-group force laws are those which are computed between different group of vehicles. These forces enable flocks to coordinate themselves during their navigation.

One of such forces is computed when flocks arrive at intersections, for vehicles to coordinate their pass through them in a safe and efficient way. When the flock enters an *intersection safety zone*, which surrounds the intersection at a distance m, as illustrated in Figure 3.8. Distance m should be conveniently set to allow vehicles to decelerate or even stop if necessary.

The repulsion potential established for this ***intersection management*** objective has the objective of making vehicles decelerate to avoid collisions and to let larger

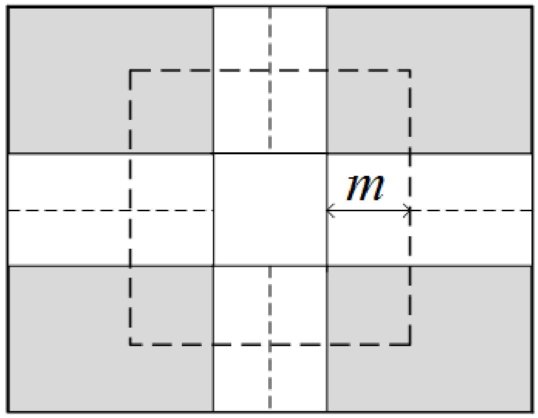


Figure 3.8: Intersection Safety Zone, delimited by the dashed contour.

flocks pass through the intersections first, as the repulsion force they induce is greater. This potential is defined by equation (3.12).

$$U_{intersection}(x) = -\sum_i \frac{1}{2} k_{int} (x - x_i)^2 \tag{3.12}$$

Once again, by computing the proper derivatives, equation (3.12) results in the following force law:

$$F_{intersection}(x) = \sum_i k_{int} (x - x_i) \tag{3.13}$$

Where k_{int} is a positive gain, x is the vehicle's position and x_i is the position of vehicle i of the other flock. Thus, the summation is over the force induced by all of the vehicles which belong to another flock approaching the same intersection.

When flocks enter the *intersection safety zone* they already know the direction they are going to take as their new heading when they pass the intersection, and so this repulsion force will only act when those new headings lead to collisions. For instance, suppose that one flock is approaching an intersection from North to South and it is going to keep heading South beyond the intersection, while a second flock is approaching the same intersection from South to North and its heading beyond the intersection is still North; since their paths do not collide, this repulsion force will not be activated. On the other hand, if the second flock's new heading was West, their paths would collide and so this respulsion force at the intersection would be activated. This example is depicted if Figure 3.9.

On the other hand, if there are already congested streets, the flocking nature of both the rFTN and the FTN could contribute to worsening congestion levels, and so it is important to detect and avoid them. Within the rFTN model this is achieved by computing a repulsion force parameterized by p, the *traffic density* factor stated in equation (2.2), which will be rewritten here for convenience:

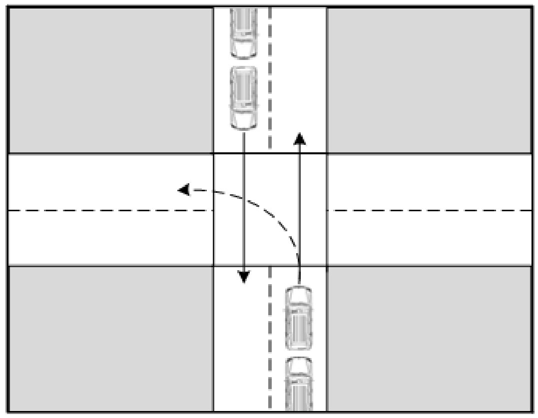


Figure 3.9: Example of colliding and non-colliding paths at intersections. The dashed arrow shows a colliding path which will activate the repulsion force.

$$p = \frac{n}{l}$$

Where p is the traffic density in vehicles per unit distance, n is the number of vehicles occupying some length of the roadway at some specified time and l is the length of the roadway.

The potential equation by which this ***congestion avoidance*** feature is achieved is defined as:

$$U_{ca}(x) = \begin{cases} 0 & \text{if } p = 0 \\ -\frac{1}{2}k_{ca}(x - x_a)^2 & \text{if } p > 0 \end{cases} \tag{3.14}$$

Once again, x is the current position of the vehicle, x_a is the position of the nearest vehicle within the congested street and p is the traffic density factor. It should be noticed that when $p = 0$, there is no traffic at all, and so no repulsion will be induced; otherwise, factor k_{ca} in the repulsion potential is given by:

$$k_{ca} = \left(\frac{\overline{u}_t}{m_{speed}} - 1 \right)^2$$

Where $\overline{u}_t$ is the *average traffic speed*, see equation (2.4), of vehicles within the street which is being analyzed, and m_{speed} is the maximum allowed speed in the city. When $\overline{u}_t = 0$ it means that all vehicles are stopped and so term k_{ca} would result in the maximum value, which is 1, while, on the other hand, if the average speed is equal to the maximum speed, k_{ca} would be 0 and no repulsion will be induced. This behavior can be depicted in Figure 3.10.

Equation (3.14) derives in the following force law:

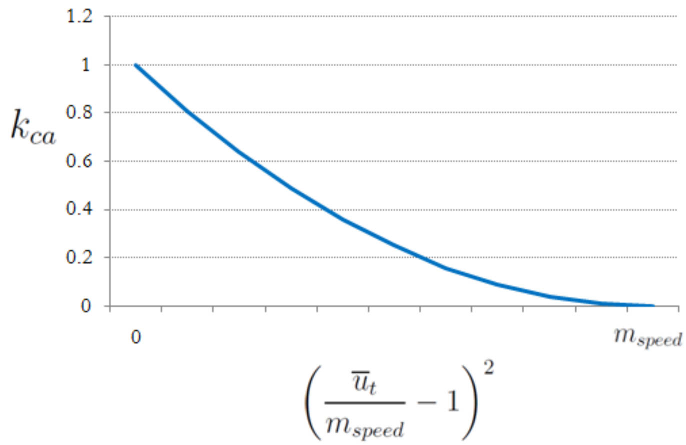


Figure 3.10: Factor k_{ca}.

$$F_{ca}(x) = \begin{cases} 0 & \text{if } p = 0 \\ k_{ca}(x - x_a) & \text{if } p > 0 \end{cases} \tag{3.15}$$

Vehicles compute this force law at intersections when they are able to change their headings; every street that may be reached through a given intersection is analyzed as explained previously.

3.3 Acceleration and Velocity Modulation

Now, once we compute the resulting force at a given moment of time, by adding up all of the force laws explained before, we can use the resulting force vector either as the acceleration of the vehicle or as the direction that it must follow, according to the APF theory. In this research work, it is preferred to use it as the acceleration, in order to meet the following two objectives:

1. Flockmates have to be able to reach each other. This can be achieved by modulating the acceleration and, consequently, the velocity of vehicles, in order to let them speed up or slow down to meet flockmates located ahead or behind them, respectively.

2. To keep a realistic scenario, in which vehicles may travel at different speeds.

However, since the sum of the force laws may result in a very large value, the force vector can not be taken directly as the acceleration or it might lead to dangerous and even unrealistic scenarios. Thus, the rFTN takes into account an *acceleration modulating* process, that can be described as follows. According to [13], let **F** be the resultant force vector at time t, $\mathbf{v}$ represents the velocity of the vehicle at some time t, m denotes its mass, and $\Delta\mathbf{v}$ the velocity change from time t to $t + \Delta t$, described by:

$$\Delta\mathbf{v} = \left(\frac{\mathbf{F} - \mathbf{v}}{m}\right)\Delta t \tag{3.16}$$

For simplicity, we will consider objects of unitary mass; that is, $m = 1$. Then, the rFTN makes $\Delta\mathbf{v}$ comply with $-a_{max} \leq \Delta\mathbf{v} \leq a_{max}$ by using its magnitude to parameterize the following equation:

$$a = a_{max}\tanh(|\Delta\mathbf{v}|) \tag{3.17}$$

Figure 3.11, shows the plot for equation (3.17). We can see that if parameter $\Delta\mathbf{v}$ is too large or too small at time t, the acceleration is bounded to a_{max} and $-a_{max}$, respectively, and so it remains in a realistic value range, making the navigation safe for vehicles. This modulation is computed at every time step of the simulation.

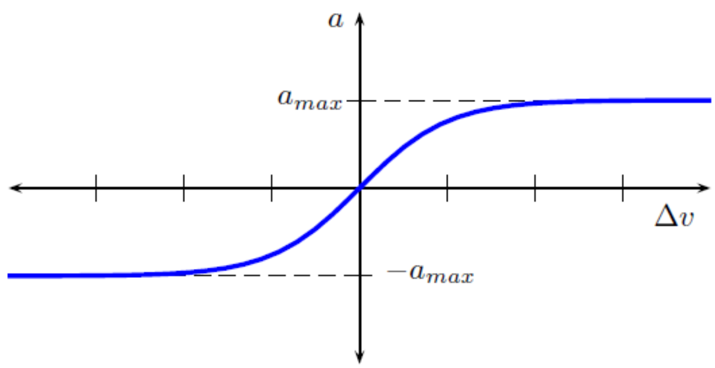


Figure 3.11: Acceleration Modulation.

Finally, the velocity is computed in the following way:

$$\mathbf{v} = \mathbf{v} + a \tag{3.18}$$

Still we have to make sure that this velocity complies at any given moment of time with $-v_{max} \leq \mathbf{v} \leq v_{max}$, and, in order to achieve this *velocity modulation* the rFTN makes use of the following function:

$$v = \frac{v_{max}}{1 + e^{-\mathbf{v}/2}} \tag{3.19}$$

For the rFTN navigational model, the minimum velocity will be set to 0 and the maximum velocity v_{max} will be determined by the city traffic restrictions. Overall behavior of equation (3.19) can be analyzed in the plot provided by Figure 3.12.

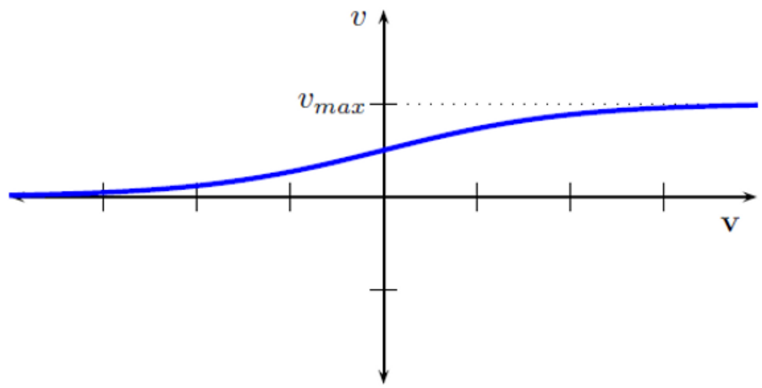


Figure 3.12: Velocity Modulation.

These processes are different from those perfomed in [13], in which the authors only get the acceleration and velocity parameters “clipped” to comply with the maximum/minimum restrictions of their model. It is claimed that the processes explained in this section are innovative and a more convenient way to deal with this problem.

3.4 Summary

This Chapter presented an in-depth description about the solution model. First, it is provided an overall description of the set of features designed for this navigation method, which represent its underlying workings. Later, technical details about the solution model are given, in order to describe the actual calculations perform within the algorithm. The next Chapter introduces a simulation tool developed to serve as an experimental platform for this research work.

Chapter 4

Traffic Simulator

This Chapter introduces a Traffic Simulator tool, that was developed to support the experimental analysis within this research work.

4.1 The Simulation Tool

As stated in Section 2.7, the NetLogo plaftform is a multi-agent programable modeling environment which is useful to simulate both natural and social phenomena. Complex systems can be modeled as well, in which both the individual behavior of agents and the emergent behavior which results from the interaction of several individuals can be analyzed and explored.

Because of these features, NetLogo is the platform that was selected to design and develop the experimental framework which enabled us to gather up the required data to measure different performance-parameters of the rFTN, and to compare them against other navigational methods that were also implemented, namely: *Traffic-Light-based* navigation, and *Chaotic* navigation; please refer to section 6.1 for a detailed description of these approaches.

This experimental framework consists in a Traffic Simulator, in which the user can set the values of different parameters of both the navigational model and the simulation itself. A list of those parameters and their description is provided in the following:

1. ***Size of the Grid:*** this parameter sets the amount of blocks and streets that will be displayed in the simulation. Values range from a 1×1, to a 9×9 city-blocks grid.

2. ***Number of Cars:*** the user can specify the number of cars that will be displayed during the simulation; cars and their destinations are generated in a random position on the streets.

3. ***Speed Limit:*** this parameter is used to determine the maximum allowed speed within the city; vehicles will not exceed this speed limit restriction under any circumstances, for the scenario to keep a realistic and safe behavior. The speed

of vehicles is modulated as explained in section 3.3, for vehicles to comply with this restriction.

4. ***Maximum Acceleration:*** to establish the a_{max} and $-a_{max}$ factors, which will be used as explained in the previous section.

5. ***Position Gains:*** for each Force Law previously described, there is a parameter that can be adjusted by the user to set the value for the position gain that each of them requires. Values range from 0.1 to 2.0.

6. ***Vision Range:*** this parameter establishes the radius of the "radar" which lets vehicles compute the rFTN calculations with the vehicles within this range. It can be set to cover the whole scenario, in which case the calculations will take into account every other vehicle in the city.

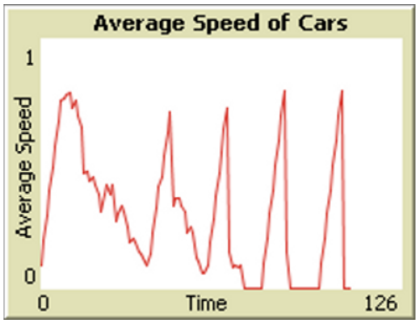
7. ***Navigational Model:*** it lets the user select the navigational model for the simulation. Options are: *rFTN*, *Traffic-Light-based*, and *Chaotic*; the latter two models will be described further later on this document.

Besides, user can observe and analyze the parameters during the simulation with the aid of the provided plots, which present on-line information about the average speed of cars, average wait time of cars, the average computational time to compute the rFTN's calculations and an additional factor called dispersion. Figure 4.1 presents sample plots for the previously mentioned parameters. All of these parameters will be described in the subsequent chapters.

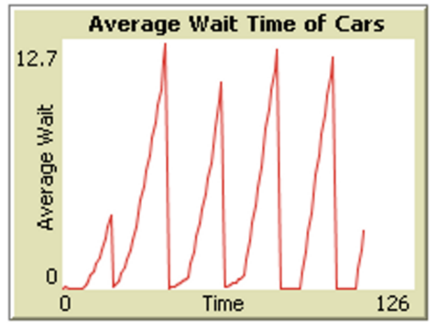
Finally, the user can observe the navigation of vehicles within the simulation itself, presented in an area which shows the city blocks in gray color, and the streets in white. Vehicles can go everywhere in the city block, since all of the streets in the simulation are two-way streets. Figure 4.2(a) and Figure 4.2(b) present a 2D and 3D representation of the Traffic Simulator, respectively.

4.2 Summary

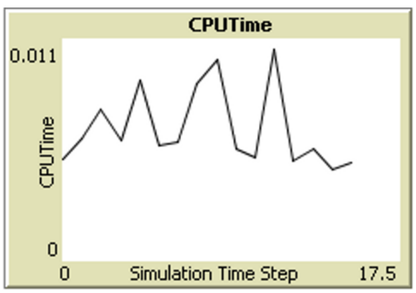
This Chapter provided a description of the a Traffic Simulator, that was developed using the NetLogo platform. This simulation tool served as an experimental platform, which enabled the conduction of experiments and the measurement of different parameters within them. Next Chapter presents and explains the preliminar experimentation conducted as part of this research; the drawn results are explained.



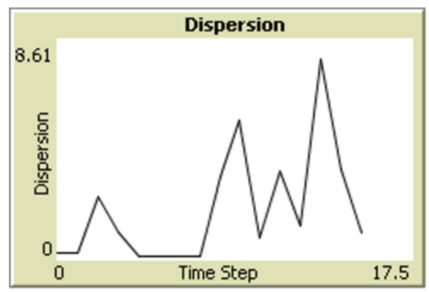
(a) Sample Plot for the Average Speed of cars.



(b) Sample Plot for the Average Wait Time of cars.

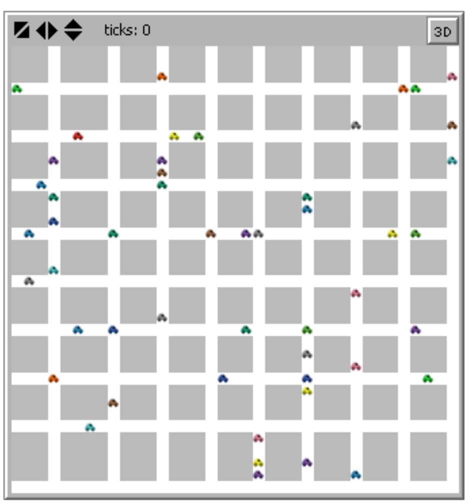


(c) Sample Plot for the Computational Time.

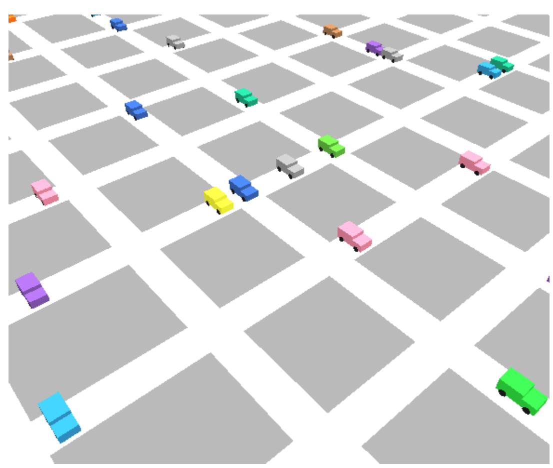


(d) Sample Plot for the Dispersion Factor.

Figure 4.1: Sample Plots for different parameters of the model.



(a) Traffic Simulator in 2D.



(b) Traffic Simulator in 3D.

Figure 4.2: Traffic Simulator.

Chapter 5

Preliminary Experimentation

The objective of the preliminar experimentation presented in this chapter is to find an appropriate configuration to tune critical parameters of the proposed solution model, such as the intersection safety zone and the variables included in the forces that make the *flocking* feature possible. This experimentation is done in order to obtain *exploratory data* about the rFTN's performance with different configurations of those parameters and to select those which provide the designed behavior.

A general description of the experiments is given in section 5.1, while the actual experimentation setup is presented in section 5.2, which provides with the traffic-related parameters that are going to be measured as *key outputs*, that will be analyzed to select the configuration of the parameters that leads to the desired performance of the solution model. The results drawn from this experiments are presented and discussed in section 5.3. Finally, conclusions of this preliminar experimentation are given in section 5.4.

5.1 Description of the Experiments

These preliminar experiments are intended to determine the most appropiate configuration for the rFTN's parameters that enable critical features of the model. Different values are going to be tested, in order to select those which provide the desired performance of the solution model, based on its design and expected behavior.

The rFTN's features that are the most important to be tuned are listed below:

- The **reactive creation of the bone structure**, since this feature enables vehicles to gather up with others, in order to create flocks, which is clearly a major objective within this research work. Recall that the flocking process within the rFTN involves two other features or processes, namely: the *flockmates selection* and the the attraction force that makes vehicles *navigate towards their destination*; this way, the only parameter that is going to be tuned within this experiments is the k_t factor, related to the magnitude of the attraction induced by the destination, since the k_p variable, that describes the magnitude of attraction related to the flockmates, is always computed as explained in section 3.2.2.

- The **intersection management** feature, since it is of major importance to determine the configuration of its parameters that will enable vehicles to go through intersections with the lowest possible delay, while providing a safe environment for them by preventing collisions. One of the parameters that are going to be tuned for this feature is the *intersection safety zone*, in order to make sure that its length m is long enough to let vehicles decelerate on time before they get to the intersection and, hence, to avoid collisions. Besides, it is important to determine the appropriate value for the k_{int} parameter, which is related to the repulsion that the intersection induces, making sure that this force is strong enough to make cars decelerate as much as needed.

- The **collision avoidance** feature, that also enables a safe scenario by preventing collisions between vehicles traveling on lanes of the same way. The length d of the *repulsion zone* and the k_r factor are going to be tuned to ensure that no collisions are going to happen; as for the length s of the *safety zone*, it will always be set to be the half of the repulsion zone's length.

It should be noticed that not all of the rFTN's features will be enabled, only those of interest for this preliminar experimentation, leaving the test of the complete set of features for the final experimentation to be presented later on this document.

The simulation itself was run for a fixed amount of $5,000$ time steps, for each of the experiments presented next in this chapter. Vehicles and their targets were generated uniformly within a 9×9 block area, with a probability of 0.1 in each time step. The speed limit was set to 1.0 units of length per time step and the maximum acceleration value was 0.4 cells per simulation steps squared. The so-called *vision range* of vehicles was set to make them compute the flockmate-selection test with every other car in the world.

5.2 Experimentation Setup

This section describes a list of different measurements that are of interest for this preliminar phase of experimentation; such measurements will provide with important information about the performance of the model under different configurations of its parameters, in order to select the specific values that leads the model to aquire the desired performance.

The measured parameters and their descriptions are presented in the following:

1. **Cohesion Index (CI)**.-
   Since flocking is about gathering elements or objects into groups, the average distance among them decreases, and so the cohesion of those elements within the scenario raises, and viceversa.

   Thus, we can measure the *closeness* of elements as in equation (5.1).

   $$X = \frac{1}{d^2 + \epsilon} \tag{5.1}$$

   Where d is the distance between two given elements, and ϵ is a factor that is set to be sufficiently small to avoid divisions by zero without affecting the original equation. The value that has been chosen for this factor is $\epsilon = 0.001$.

   Furthermore, we can use this variable to compute a cohesion index as in equation (5.2).

   $$D = \frac{\sigma^2}{\mu}, \tag{5.2}$$

   $$\sigma^2(x) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2$$

   Where σ^2 is the variance, n is the number of elements being measured, and μ is the mean of them.

   This parameter was designed to serve as evidence of the fact that vehicles were actually gathering into flocks; however, it does not provide further information about the created flocks themselves, like the number of groups that were created, the size of the flocks, etcetera.

2. **Average Completion Time (ACT)**.-
   This parameter tells us the average of the amount of time it took vehicles to reach their destinations. The overall trip time of each vehicle is computed and, at the end of the simulation, all of them are averaged.

   Since the experiments are carried out via a simluation, the completion time units will be expressed in terms of *simulation steps* and so, this parameter will actually tell the amount of steps that it took for all vehicles in the simulation to get to their destinations.

3. **Number of Collisions (NC)**.-
   As its name already suggests, this parameter will count the number of collisions between vehicles that happened within the simulation time. This will help to

analyze the ability of the proposed solution model to keep a safe navigation environment for urban traffic.

The rFTN's features that are going to be tuned (described above) were progressively added up throughout this experimentation phase. That is, in the first experiments only the *flocking* feature was enabled, in order to isolate it and, hence, be able to analyze it without external "noise". Once the convenient values for flocking were selected, they were used to run once again the experiments, but this time enabling the *intersection management* feature as well. A final set of experiments were conducted at the end, enabling the *collision avoidance* feature and using the selected values for the parameters of both of the previously tuned features.

5.3 Preliminary Results

This section presents the results drawn from this experimentation phase. Refer to the previous section for a description of the way these experiments were conducted, or the way in which the results presented here were computed.

Flocking is the first rFTN feature to be tuned by experimentation. Having this mechanism working appropriately is essential to establish the traffic-management paradigm upon which this research work was conducted; besides, this feature will serve as a navigation principle, over which the rest of the features can be added up.

Table 5.1 shows the results drawn from the experiments that were conducted to tune the k_t factor related to the attraction induced by the destination. We can see that different values were tested, all of them complying with being greater than 1, which is the greatest value that can be achieved by the k_p factor regarding the attraction force induced by flockmates (see section 3.2.2). It should be noticed that, if k_t was set to be less than 1, the attraction by flockmates would be stronger and vehicles will tend to follow its group more than their destinations, which would probably prevent cars from reaching them at all; that is why all tested values comply with that restriction.

k_t	**Cohesion Index (CI)**	**Average Completion Time (ACT)**	**CI/ACT factor**
1.1	233.8173	50.4581	4.63
1.2	187.1972	45.0823	4.15
1.3	159.4297	30.7619	5.18
1.5	72.1395	28.9782	2.48
1.7	45.1836	27.0638	1.67

Table 5.1: Cohesion Index and Average Completion Time results for different k_t values.

Furthermore, two key outputs were measured: the Cohesion Index (CI) and the Average Completion Time (ACT). From the analysis of both parameters we can select the most convenient value to tune the feature in question, as we need the model to present a reasonably high CI (to prove that vehicles are actually gathering up), but making sure that this flocking mechanism do not deviate vehicles' paths too much, in which case it will take too long for them to get to their destinations, thus presenting too large ACT values.

Taking this into consideration, we are looking for a k_t value that provides the gratest *CI/ACT factor*, which is a measure of the proportion between the provided CI and the computed ACT. Hence, Table 5.1 shows that $k_t = 1.3$ complies with this desired condition; moreover, we can see that it presents an ACT very similar to the lowest drawn value (which is 27.0638) and also a high CI. Thus, it will be used to tune this force law. Other alternatives such as $k_t = 1.2$ or $k_t = 1.5$ introduce very large ACT and CI values, respectively, making them less convenient for the desired behavior of the solution model, which is shown in the "CI/ACT factor" column.

Now that we have tuned the flocking mechanism, vehicles have a navigation approach that can be enhanced by introducing additional features. As mentioned before, it is of major importance to keep a safe and realistic scenario within this solution model; thus, it is imperative to conduct experiments directed to determining the proper configuration of the *intersection management* process taken into account within the rFTN, with the objective of completely preventing collisions.

It should be recalled that the designed intersection management requires two parameters, namely: m, the length of the so-called *intersection safety zone* and k_{int}, the magnitude of the repulsion induced by the intersection. Refer to section 3.2.3 for more details about this process.

Another set of experiments were run to achieve this objective, by making use of the already tuned *flocking* process. Several values for both parameters m and k_{int} were tested, as shown in Table 5.2. Each cell of the table represents a different configuration of both parameters, and were used to perform different experiments; collisions **at intersections** were counted for each of them and the final sum is provided.

	m		
k_{int}	**1**	**2**	**3**
1.1	896	72	17
1.3	775	46	8
1.5	732	34	0
1.7	719	17	0
1.9	703	9	0

Table 5.2: Number of collisions for different k_{int} and m values.

We can see that the greatest amount of collisions happened when the length m of the *intersection safety zone* is too small ($m = 1$). Besides, the results show that, for this case, even if the magnitude k_{int} of repulsion is increased, the number of collisions is still too large.

On the other hand, when $m = 2$ the number of collisions decreased drastically if compared to those counted when $m = 1$, but even a single collision is not affordable in this research work. Furthermore, in column $m = 3$ we can see that for $k_{int} = 1.5$ to $k_{int} = 1.9$ no collisions happened and all those parameters are convenient in terms of making the navigation safety for vehicles; however, it will be selected the value $k_{int} = 1.5$, as it is enough to ensure collision avoidance, and greater values could only introduce unnecessary trip-time delays.

So far, we have tuned the *flocking* and the *intersection management* processes, in a progressive way. Nevertheless, in order for the solution model to completely prevent accidents during the entire trip of vehicles (and not only at intersections), we have to once again determine the appropriate values to make the *collision avoidance* feature work in the safest possible manner.

This feature requires three parameters to be set: d, the length of the *repulsion zone*, s the length of the *safety zone* and factor k_r which determines the magnitude of the induced repulsion.

In a very similar way as with the intersection management configuration, several values for parameters d and k_r were tested, in order to know the most convenient combination. Recall that, as for factor s, it will always be set to be half the length of d.

In each of the conducted experiments, the number of collisions were counted. Table 5.3 presents the total amount of collisions that happened for each test scenario. It is shown that when $d = 0.75$ and $s = 0.375$, vehicles are most likely to collide between them, as the repulsion zone is too small and they are not able to slow down in time, no matter the magnitude of the induced repulsion.

k_r	$d = 0.75$, $s = 0.375$	$d = 1$, $s = 0.5$	$d = 1.5$, $s = 0.75$
1.5	581	56	12
1.7	574	32	5
1.9	563	24	0
2.1	548	13	0
2.3	536	4	0

Table 5.3: Number of collisions for different k_r, d and s values.

On the other hand, when $d = 1$ and $s = 0.5$ the amount of collisions is not that large, but still does not represent a safe scenario for urban traffic. It is only until $d = 1.5$ and $s = 0.75$ when there are no collisions detected for $k_r = 1.9$ to $k_r = 2.3$; yet, it is once again enough to select $k_r = 1.9$, in order to avoid using a repulsion force unnecessarily stronger.

5.4 Summary and Conclusions

The preliminar experimentation presented in this Chapter is aimed to provide the proposed solution model with *exploratory data* about the appropriate configuration of critical mechanisms, such as *flocking*, *intersection management* and *collision avoidance*.

Having this objective as a background, the rFTN navigation model was tested progressively by adding up the three processes mentioned above. This processes involved the following rFTN features described in section 3.2:

1. Flockmate selection.
2. Reactive creation of the bone structure.
3. Navigation towards destination.
4. Collision avoidance.
5. Intersection management.

The complete set of feeatures is going to be tested in the final experimentation conducted for this research, and the results drawn from it are reported in Chapter 6.

The results presented in this preliminary experimentation determined the most convenient parameters to tune the solution model as designed. Thus, it can be concluded that the rFTN's behavior as a navigation model is the desired one, and that further experiments can be conducted to test its performance regarding now its ability to manage traffic; furthermore, a full comparison between this model and the FTN has been enabled.

The experimentation was conducted via a traffic simulation, which had the following characteristics:

1. Each simulation was run for a fixed time of $5,000$ steps.
2. The traffic simulator provided with a 9×9 block area.
3. Vehicles and their targets were generated uniformly with a probability of 0.1
4. Cars take into account all the rest of the vehicles to perform the flockmate-selection process.
5. Speed limit was set to 1.0 units of length per time step.
6. Maximum acceleration value was 0.4 cells per simulation steps squared.

There were defined some performance parameters that were measured during the simulation, as *key outputs* that were used to select the most appropriate values for critical variables of the previously mentioned rFTN features. Those key outputs are summarized in the following list:

1. Cohesion Index (CI).

2. Average Completion Time (ACT).

3. Number of Collisions (NC).

The performed experiments to tune the *flocking* process took into account the CI and ACT. The objective of this analysis was to determine a value for the k_t factor to make the scenario stay as close as possible to the largest CI, while ensuring that vehicles do not deviate their path too much, in order to prevent the ACT from being too high. As for the *intersection management* and *collision avoidance* the output measured was the number of collisions that were detected during each experiment; it is clear to see that in this case the best configuration of parameters is the one that leads to a collision-free navigation for vehicles.

In this way, the tuned parameters and their values are presented in Table 5.4. These values can be used to conduct the final experimentation phase, as it has been proven that they make the rFTN achieve the designed behavior and performance.

Parameter Name	**Value**
k_t	1.3
k_{int}	1.5
m	3 cells
k_r	1.9
d	1.5 cells
s	0.75 cells

Table 5.4: Selected values to tune the rFTN model.

Chapter 6

Final Experimentation and Results

This Chapter presents the final experimentation phase, which was conducted for the solution model of this research work to be compared against other two basic, low-level urban traffic methods and also other advanced approaches. These experiments are intended to show that the rFTN outperforms representative traditional navigation models we can see these days, hence, proving to be an effective and innovative solution for the urban traffic management problem. Besides it will show that the rFTN's performance is as good as the advanced approaches mentioned here, and sometimes even better. As explained before, Chapter 5 presented *exploratory data* about the solution model's perfomance under different configurations for their critical parameters; now, this Chapter is aimed to provide with *validatory data*, which were drawn by testing and comparing the rFTN against other navigation methods, taking into account different traffic-related parameters. Once again, for this experimentation phase an experiment is defined as the result of separately testing the three methods under identical setup.

First, section 6.1 provides an in-depth description of the final experiments that were conducted for this research work. Then, section 6.2 provides the selected setup for this final experimentation, specifying the measured traffic parameters which are of most interest in this *validatory* phase, as well as the values for the force laws's variables in the rFTN. The analysis of the results drawn from this experiments is provided in section 6.3, which will show a comparison of the rFTN's performance against both traditional and high-level methods. Finally, section 6.4 presents the conclusions for this final experimentation phase.

6.1 Description of the Experiments

This final experimentation phase is aimed to prove that the rFTN is a more efficient model to deal with urban traffic management, than two traditional methods that were selected for this matter, namely: *Traffic light based navigation* and *Chaotic navigation*. These two approaches are currently present in most cities, and the navigation policies that they take into account are well known. Furthermore, it will be proved that it can even outperform high-level methods, like the FTN.

As previously mentioned in Chapter 5, in this final phase of experimentation all of the features designed for the rFTN will be enabled, in order to compare the complete solution model's performance and its ability to handle traffic. For convenience, a general description of the rFTN's features will be provided in the following, in despite of the fact that they were thoroughly described in Chapter 3, along with a formal definition for the other navigation methods mentioned above:

1. **rFTN:** this experimentation will test the rFTN's fitness as a solution for handling urban traffic, and so, the complete set of features described in section 3.2 are enabled.

   Recall that the list of features is divided into two groups:

   (a) **Intra-group forces:** which include the *Flockmates selection, Reactive creation of the Bone Structure, Navigate towards destination, Flocking formation*, and *Collision avoidance* processes.

   (b) **Inter-group forces:** which considers the *Intersection management* and *Congestion avoidance* features.

   Furthermore, modulation of the acceleration and velocity of vehicles will also be taken into account and will be performed as explained in section 3.3.

2. **Traffic-Light based navigation:** this is the traditional model in which vehicles stop at every intersection when the traffic light is red, and they may go through the intersection if the light is green.

   In the developed simulator, traffic lights are present in all of the intersections in the city and vehicles accelerate to reach the maximum allowed velocity whenever possible. On the other hand, cars decelerate when they get close to each other, in order to avoid collisions.

3. **Chaotic navigation:** in this model there are no traffic lights at intersections. Moreover, there does not exist any intersection management policies; vehicles in this navigation approach try to go through them whenever possible. That is why it is called *chaotic*.

   Vehicles decelerate before reaching an intersection, in order to avoid collisions and to keep a safe environment. Besides, in a similar way as in the *Traffic-Light* model, cars navigate at the maximum velocity of the city and decelerate if they approach another vehicle, for collision avoidance.

4. **Flock Traffic Navigation Based on Negotiation (FTN):** see section 2.6.2 for a brief explanation of this method, or refer to [4], for an in-depth description of it.

The number of vehicles in the experiments was set to 50, 100, 200, 300 and 400. The rest of the simulation-related parameters remained the same as in the preliminar experimentation presented in Chapter 5; a summary of them and their values is presened in the following:

- The city was represented by a 9×9 block area.
- Vehicles were generated uniformly along with their targets.
- The *vision range* of cars was also set to cover the whole world, in order to test the computational time required to compute the processes with the largest work load.

Besides, the speed limit was set to 1.0 units of length per time step and the maximum acceleration value was 0.7 cells per simulation steps squared. As for the traffic-light model, the light period was set to 20 time steps of the simulation.

In this set of tests, every experiment was run three times, in order to test the rFTN, the Traffic-light based and the Chaotic navigation in each one of the runs. As one may expect, this enabled a fair comparison of the traffic navigation approaches described previously, since they shared identical simulation setup and were tested over the exact same scenario (i.e. coordinates of vehicles and their targets remained the same each run). As for the FTN, the experimentation results presented in [4] are going to be used.

6.2 Experimentation Setup

This section provides a list of the parameters which were selected to test the ability of the models mentioned above to handle urban traffic. Once again, it is important to measure the computational time required for the **complete** rFTN to be processed, in order to prove that the solution model is able to be work in real-time scenarios, such as traffic management.

Not all of the parameters listed in section 5.2 are of interest for this final experimentation, only those traffic-related ones presented in the following:

1. **Average Speed (AS)**.-
   This parameter is measured by computing the mean of all vehicles's speed in the scenario at every moment of time during the simulation.

   This calculation is performed by using the equation (2.4) from the traffic theory concepts.

2. **Average Wait Time (AWT)**.-
   This parameter tells us the average units of time that vehicles remain stopped, during their trip to their destinations. It is measured every time a given vehicle's velocity is 0, and stops when that vehicle begins to accelerate.

   This parameter is measured at every moment during the simulation.

3. **Average Completion Time (ACT)**.-
   This parameter was explained before in section 5.2, but it will be described again here for convenience: the ACT tells us the average of the amount of time it took all of the vehicles to reach their destinations. The overall trip time of each vehicle is computed and, at the end of the simulation, all of them are averaged; it is expressed in terms of *simulation steps*.

4. **Average Computational Time (ACPUT)**.-
   This variable shows the average of the amount of CPU time, in seconds, that it takes for a given vehicle to compute the flocking calculations. It is measured at every time step during the simulation, and will provide information about the computational effort required for the rFTN to be computed.

   It should be recalled that one of the major objectives of this research work is to design and develop an algorithm that requires the minimum computational time to be processed and, so, the results for this parameter are expected to be low. Therefore, this measurement is extremely important, since it will lead us to find out if the rFTN meets this objective.

5. **Saved Trip-Time (STT)**.-
   This factor represents the aritmetic difference of the Average Completion Time (ACT) between the rFTN and one of the other traffic navigation approaches. Formally:

   $$S = C_{other} - C_{rFTN}$$

   Where S is the Completion Time Delay, C_{other} is the ACT presented by the other method against which the rFTN is being compared and C_{rFTN} is the ACT value drawn when vehicles are navigating as in the rFTN model.

   Now, this parameter is intended to show the amount of trip time which the solution model is able to save, while being compared against other approaches.

As for the rFTN itself, there are several parameters that have to be set to tune the model, regarding variables included in its force laws and other safety zones both around vehicles and at intersections. These values were appropriately set, based on the experimentation conducted in Chapter 5 for the critical features, such as flocking, collision avoidance and intersection management. Variables for the *flocking formation* (k_{ap}) and *congestion avoidance* (k_{ca}), where conveniently selected, as well as the *noise* parameter (k_n). Table 6.1 presents the selected values for the force laws's variables.

Parameter name	**Value**
k_p	computed as explained in section 3.2.2
k_t	1.3
k_{ap}	1.5
k_r	1.9
k_n	0.2
k_{int}	1.5
k_{ca}	1.2

Table 6.1: Selected values to tune the rFTN's force laws for final experimentation.

The so-called *safety zones* taken into account by the solution model's processes are present in two different locations:

1. **Around Vehicles:** which is composed by both the **Repulsion Zone** d and the **Safety Zone** s.

2. **Around Intersections:** composed by the **Intersection Safety Zone** m.

Values for these dimension variables (d, s and m) are provided next in Table 6.2. For more specific information about them, refer to section 3.2 or to section 5.3 to know the way that these values were selected, based on experimentation.

Parameter name	**Value**
Vehicle Repulsion Zone d	1.5 cells
Vehicle Safety Zone s	0.75 cells
Intersection Safety Zone m	3 cells

Table 6.2: Selected values to tune the rFTN's safety zones.

6.3 Results Anlysis

Now, it will be presented the results drawn from the final experimentation phase, where the rFTN is compared against both traditional methods and advanced approaches such as the FTN. First, section 6.3.1 provides a comparative analysis between the Traffic-light based model and the Chaotic navigation, in order to know which of them is a more efficient traffic method. Later, section 6.3.2 provides a comparison between the rFTN and Traffic-Light based navigation, while section 6.3.3 compares the

Number of Agents	Traffic-Light based navigation	Chaotic navigation
50	0.418	0.877
100	0.395	0.775
200	0.323	0.517
300	0.289	0.352
400	0.246	0.156

Table 6.3: Final Results for the Average Speed (AS) parameter, concerning the Traffic-lights and Chaotic navigation.

solution model results against those obtained by the Chaotic model. Then, an analysis of rFTN and FTN's results is conducted in section 6.3.4.

6.3.1 Comparing Traffic-Lights against Chaotic navigation

Traffic-light based and Chaotic navigation are traffic models that were selected to be compared against the rFTN's performance, in order to prove that the proposed solution model is a better approach to the urban traffic problem, than the traditional ones. However, it is also of interest for this phase of experimenation to know which of these traditional approaches is better than the other, thus representing a more challenging competition for the rFTN.

Table 6.3 provide the results regarding the **Average Speed (AS)** parameter, drawn from the conducted experiments. We can see that, consistently, the Chaotic navigation presented the higher values for this paramenter, for all scenarios but the last one consisting in 400 agents. Moreover, we can see that in both models the AS tends to decrease as the number of agents in the scenario increases; these data is evidence for the fact that both approaches lead to congestions that are worsened as more vehicles are introduced in the city.

Now, the **Average Completion Time (ACT)** results are shown in Table 6.4. For scenarios with a relatively low amount of agents (50 to 100), the Chaotic navigation performed better than the Traffic-lights, which is consistent with the nature of both models, since the chaotic navigation does not requires vehicles to stop at intersections if there are not other vehicles approaching it (which is more likely to happen in scenarios with fewer agents) and so their trip-times are lower.

On the other hand, in crowded scenarios the chaotic navigation is most likely to lead to congestions, because the amount of vehicles that have to go through intersections is greater and the coordination between them is more difficult, while the ordered navigation provided by traffic-lights, in a way, guarantee that all vehicles will go through intersections, as it assigns some amount of green-light time to all streets. Hence, for these crowded scenarios (200 to 400 agents) traffic-lights performed better.

Number of Agents	Traffic-Light based navigation	Chaotic navigation
50	115.972	113.813
100	163.273	161.128
200	220.375	233.380
300	279.732	284.442
400	339.185	396.041

Table 6.4: Final Results for the Average Completion Time (ACT) parameter, concerning the Traffic-lights and Chaotic navigation.

Number of Agents	Traffic-Light based navigation	Chaotic navigation
50	23.2920	22.4160
100	63.1694	62.1911
200	85.0973	89.6881
300	102.8412	107.9545
400	146.4592	152.6523

Table 6.5: Final Results for the Average Wait Time (AWT) parameter, concerning the Traffic-lights and Chaotic navigation.

Finally, Table 6.5 presents the results for the **Average Wait Time (AWT)** parameter, in which we can see that, consistently with the ACT results, the Chaotic navigation presented lower values for the 50 and 100-agent scenarios, while, on the other hand, the traffic-lights performed better for the rest of them.

We can conclude that the traffic-light based model represents a better approach to the urban traffic management problem, as it presented better performance than the chaotic navigation, regarding the ACT and AWT parameters, which provide with important traffic-related data.

6.3.2 Comparing rFTN against Traffic-Light based navigation

Traffic-light based navigation is a current well-known approach, and so it is essential to compare its results against the rFTN's.

Table 6.6 presents the results of these methods, regarding the **AS parameter**. In this table we can see that vehicles in navigating as in the *Traffic-light* model tend to decrease their speed (in the ***average***), when the number of agents in the scenario increases. This is because, in despite of the fact that some vehicles are moving at, perhaps, the maximum speed limit, the rest of them are stopped at an intersection; this fact affects the average speed of the complete set of cars (i.e. as a whole) within the scenario at any given moment of time.

Number of Agents	Traffic-Light based navigation	rFTN navigation
50	0.418	0.902
100	0.395	0.904
200	0.323	0.909
300	0.289	0.910
400	0.246	0.915

Table 6.6: Final Results for the Average Speed (AS) parameter, concerning the Traffic-light-based navigation and the rFTN.

Number of Agents	Traffic-Light based navigation	rFTN navigation
50	115.972	43.240
100	163.273	54.348
200	220.375	70.831
300	279.732	82.864
400	339.185	93.873

Table 6.7: Final Results for the Average Completion Time (ACT) parameter, concerning the Traffic-light-based navigation and the rFTN.

On the other hand, we can see that, for all of the test scenarios, the AS values obtained whith the rFTN remain almost equal to the maximum allowed speed value, no matter the amount of agents in the simulation. In fact, as the number of agents increases, so do the AS values; this behavior is coherent with the design of the solution model because, as one may recall, the acceleration is modulated in terms of the magnitude of the force vector, which is computed taking into account the flockmates that each vehicle has "selected", and it is more likely for the set of flockmates to be greater in scenarios with larger amount of agents. Consequently, the acceleration value remains in the maximum, hence enabling vehicles to navigate at the maximum allowed speed.

Now, the **ACT values** obtained in this final experimenation are presented in Table 6.7, in which we can see that, consistently, the rFTN's results are lower than those obtained with the traffic-light method. Moreover, the values of the latter approach grow in a somewhat exponential way, while those of the former grow almost linearly. These results prove that the rFTN leads to lower overall trip times, than those which can be obtained with a traditional traffic-light navigation.

Furthermore, Table 6.8 shows the **STT results** drawn from the previous ACT values, which represent the amount of trip-time saved by using the rFTN model, over the traffic-light approach. These results show that the magnitude of the solution model's savings are always more than one ten of simulation steps and it increases as the number

Number of Agents	Saved Trip-Time (STT)
50	16.295
100	10.185
200	12.543
300	31.289
400	61.470

Table 6.8: Final Results for the Saved Trip-Time (STT) parameter, concerning the Traffic-light-based navigation and the rFTN.

Number of Agents	Traffic-Light based navigation	rFTN navigation
50	2.2920	0.0009
100	2.4022	0.0062
200	3.2464	0.0025
300	16.3881	0.0067
400	25.4412	0.0089

Table 6.9: Final Results for the Average Wait Time (AWT) parameter, concerning the Traffic-light-based navigation and the rFTN.

of agents in the scenario is greater, which is clearly an advantage of the solution model, because it will lead to grater trip-time savings in the presence of high urban traffic volumes.

Concerning the **Average Wait Time (AWT)** values presented in Table 6.9 we can see that, once again, the rFTN clearly outperforms the traffic-light model, by more than 3 orders of magnitude. Moreover, the AWT values for the latter method have a greater growth rate than that presented by the rFTN; that is, as the number of agents in the world increases, the traffic-light's AWT values grow by a larger amount each time. On the other hand, the results presented by the rFTN remains low and almost the same for all of the different test scenarios, which proves that the solution model represents a very effective traffic-handling approach.

6.3.3 Comparing rFTN against Chaotic navigation

When there are no traffic-lights in a city, it leads to a scenario in which vehicles, at intersections, just try to go through them whenever they can, without any intersection-management policy to follow. This can be thought of as a *chaotic* navigation approach. Since this is another traditional well-know approach, it is important to test and compare it against the rFTN. It should be remarked that the rFTN's results are the same presented in section 6.3.2.

Number of Agents	Chaotic navigation	rFTN navigation
50	0.877	0.902
100	0.775	0.904
200	0.517	0.909
300	0.352	0.910
400	0.156	0.915

Table 6.10: Final Results for the Average Speed (AS) parameter, concerning the Chaotic navigation and the rFTN.

Number of Agents	Chaotic navigation	rFTN navigation
50	113.813	43.240
100	161.128	54.348
200	233.38	70.831
300	284.442	82.864
400	396.041	93.873

Table 6.11: Final Results for the Average Completion Time (ACT) parameter, concerning the Chaotic navigation and the rFTN.

Table 6.10 presents the results of both the *Chaotic* navigation and the rFTN, regarding the **AS parameter**. These data show that the former approach makes vehicles decrease their ***average*** speed, as a greater amount of agents is present in the scenario. This an expected behavior, since this model does not guarantee that vehicles can go at the maximum speed, as they have to stop frequently at intersections for an **undefined** amount of time.

Once again, the rFTN presented results that are almost the same for all the test scenarios. As explained before, in scenarios containing a greater amount of agents, it is more likely for a given vehicle to select a greater amount of flockmates as well and, since the acceleration and, hence, velocity modulation is computed in terms of the summation of force laws over the flockmates, the average speed of cars tends to increase, as shown in Table 6.10.

The **ACT values** drawn from this final set of experiments are shown in Table 6.11. We can see that the rFTN is able to make vehicles end their trip within a lower amount of simulation time steps than the Chaotic approach. Moreover, in crowded scenarios, like the 400-agent scenario of this table of results, the ACT values for the solution model are almost one third of those drawn from the chaotic navigation. That is, the amount of overall trip time required for the latter approach is almost three times the amount needed for the rFTN model.

Now, Table 6.12 provides the **STT results** computed with the ACT values pre-

Number of Agents	Saved Trip-Time (STT)
50	70.573
100	106.780
200	162.549
300	201.578
400	302.168

Table 6.12: Final Results for the Saved Trip-Time (STT) parameter, concerning the Chaotic navigation and the rFTN.

Number of Agents	Chaotic navigation	rFTN navigation
50	22.416	0.0009
100	62.1911	0.0062
200	89.6881	0.0025
300	107.9545	0.0067
400	152.6523	0.0089

Table 6.13: Final Results for the Average Wait Time (AWT) parameter, concerning the Chaotic navigation and the rFTN.

sented previously in Table 6.11. These data show the rFTN's savings regarding the average trip-time needed for vehicles to get to their desinations, which tend to increase as the amount of agents in the scenario is greater. Moreover, these values are even greater than those drawn from the comparison between the rFTN and the Traffic-light based navigation, which means that the existence of a intersection management policy, even a low-level one such as traffic-lights, can lead to decreasing overall trip-times; this is because policies, in a way, guarantee that vehicles will make it through the intersection, and on the other hand, a chaotic navigation cannot.

Besides, we can see that in crowded scenarios the STT values tend to increase, hence, proving that ability of the solution model to handle high volumes of traffic in an efficient way.

Finally, concerning the **Average Wait Time (AWT)**, data provided in Table 6.13 show that the rFTN navigation method drastically outperforms the chaotic navigation, by more than 3 orders of magnitude. Once again, AWT values grow by a greater rate than those obtained with the rFTN, which means that it will lead to greater wait times in scenarios containing large amounts of agents; on the contrary, the rFTN is capable of keeping these AWT values very low, no matter the amount of vehicles in the city, as its values grow by a very low rate.

6.3.4 Comparing rFTN against FTN

Being the FTN an advanced method for handling traffic, it is necessary to extend the set of experiments, in order to formally compare the rFTN's results against those presented in [4] for the FTN approach (see Table 2.2). This will also provide with the proper data to conduct an analysis of both models' performance and to make conclusions.

Hence, the set of test scenarios will now include 2, 5, 10, 25, 50, 100, 250 and 500 vehicles, in order to match the number of agents which were selected to run the FTN experiments, while the rest of the parameters mentioned previously in this Chapter, both for the simulation and the solution model, remain the same.

As Table 2.2 shows, two important parameters were measured in the FTN's experimentation phase: *Saved Time* and *Computational Time*, which are renamed here as **Saved Trip-Time (STT)** and **Average Computational Time (ACPUT)**, respectively.

Moreover, the *saving times* for the FTN were measured by comparing the trip-time that takes for vehicles to get to their destinations without flocking with any others, against the trip-time obtained by navigating as in the FTN.

In a similar way, the rFTN's saving times were drawn from the comparison of this model against the *Chaotic navigation*, as the latter approach describes a *non-flocking* behavior that does not take into account any traffic-management policy; this way we will succeed in matching the strategy used to compute the FTN's experimentation measurements.

Nevertheless, it should be remarked that these *saving times* were measured in a different way for each of these models, and does not necessarily provide exact same information. Moreover, the FTN experiments were run over a 50×50 city block area, while the rFTN experiments used a 9×9 city block; thus, the FTN workload was greater and it will affect the ACPUT parameter, since this measure in the FTN model depends on the number of flocks.

However, results drawn from this experimentation serve as an initial comparison between both navigation approaches.

Table 6.14 provides the results of both models, concerning the **Saved Trip-Time (STT)** parameter, along with their variances, which are always almost the same for both models. We can see that for 1 to 10 agents these values are almost the same for both models, being the FTN results higher for the 2-agent scenario and slightly lower in the 5 and 10-agent ones. However, as for the STT values themselves, both models appear to perform poorly for this range of agents, since there is not much vehicles can do about gathering into groups, in order to get the flocking benefits that these models are designed to provide.

Furhtermore, for 5 to 50 agents the rFTN presented slightly higher saving times, than those obtained by the FTN model, but in the range from 75 and 500 agents the latter approach proved to be capable of bringing the greater amount of time savings, being the 100 scenario the only one in which the former navigation method performed better.

	FTN		rFTN	
Number of Agents	**Saved Trip-Time (STT)**	**Variance of STT**	**Saved Trip-Time (SST)**	**Variance of STT**
2	3.17	34.7688	1.325	36.762
5	12.29	44.1423	16.712	47.235
10	24.175	78.3251	24.252	72.281
25	42.15	141.1389	46.450	122.174
50	64.705	252.1217	70.573	259.237
75	85.41	319.613	77.625	324.128
100	100.915	398.374	106.780	448.127
250	204.13	1320.300	181.635	1451.983
500	366.575	2172.400	333.258	2242.971

Table 6.14: FTN and rFTN's saving times.

This is probably due to the *speeding bonus* that the FTN assigns to the flocks, based on their size; this way, the larger the flock, the faster they are allowed to travel across the city, hence enabling vehicles to get to their desinations in less amount of time, reducing their overall trip-time; clearly, scenarios containing a greater amount of agents are more likely to create larger flocks with greater speeding bonuses. On the other hand, the rFTN takes into account a *speed limit*, and so vehicles are required to modulate their velocity to meet this restriction (see section 3.3); even if they could, vehicles are never allowed to break this speed limit and, therefore, they are not able to achieve the trip-time reduction in the exact magnitudes obtained by the FTN.

However, the STT values drawn from the rFTN remain very close to the FTN's, which means that the solution model presented in this research work is capable of providing almost the same time savings (often even better), with the additional, and very critical advantage of keeping a safe and also realistic scenario, in which vehicles do not profit from *speeding*, but from their ability to go through intersections faster, given their flock size. Having said that, it can be claimed that the intersection management mechanism designed for the rFTN is proving to be a highly efficient solution to this problem.

Figure 6.1 shows a plot for the STT values, in order to graphically analyse their behavior as the number of agents increases. As stated above, we can see that for 2 to 50 agents both models performed almost the same, being the rFTN time savings slightly higher. This plot also also shows that for the final test scenarios, composed of 250 and 500 vehicles, the FTN performed better, but still the rFTN results stayed very close.

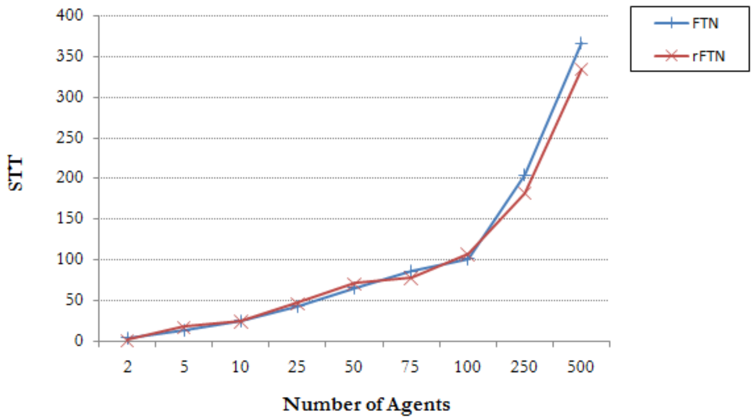


Figure 6.1: Plot for the Saving Trip-Times results.

Now, Table 6.15 presents the results drawn from both models, reagarding the **Average Computational Time (ACPUT)** that each one need to perform its activities. It is important to mention that the FTN experimentation was conducted using a Centrino 1.10 GHz microprocessor, and the Matlab 7.0 platform, running on a Windows XP operating system [4]. As for the rFTN, it was used a Pentium Dual-Core 1.73 GHz microprocessor, and it was selected NetLogo as the experimentation platform, running on Microsoft Windows Vista.

We can see in Table 6.15 that the FTN requires a somewhat low ACPUT when tested within scenarios containing less than 100 agents; however for the range of 100 to 500 agents it needs more than one second to run, which can be seen as a hazardous procedure, because of the velocity involved. One vehicle at 100 km/hr might travel 27.7 meters before it receives the negotiation messages, which may be too late to resume the algorithm's processes accurately, and will result in coordination problems. Moreover, these results show that the FTN's fitness as a solution to the real-time traffic management problem is clearly low.

On the other hand, the rFTN requires always a very low computational time to be processsed with its complete set of features, even when tested in crowded scenarios like the 100-agent to 500-agent scenarios taken into account in these experiments. In more specific terms, the solution model always needs an ACPUT in the order of a fraction of a second: *milliseconds* for most of the scenarios (25 to 100 agents) and slightly greater for the 250 and 500-agent scenarios.

Therefore, it can be claimed that these results prove that the rFTN model is able to handle real-time problems, like the urban traffic scenario, in which it is essential for the underlying traffic-management mechanism to respond almost immediately, or delays may lead to disasters in a real-life implementation.

Moreover, Figure 6.2 provides a plot for the ACPUT values presented above, in which we can see that the rFTN drastically outperforms the FTN model, in terms of

	FTN model		rFTN	
Number of Agents	**Avg. CPU Time (ACPUT)**	**Variance of ACPUT**	**Avg. CPU Time (ACPUT)**	**Variance of ACPUT**
2	0.0149	0.0004	0.00061	0.0002
5	0.0146	0.0003	0.00071	0.0001
10	0.0222	0.0004	0.00089	0.0004
25	0.0769	0.0003	0.00138	0.0002
50	0.2489	0.0005	0.00268	0.0003
75	0.5258	0.0006	0.00396	0.0004
100	0.9116	0.0005	0.00430	0.0003
250	5.4622	0.00001	0.01372	0.0000087
500	21.7039	0.00001	0.01830	0.0000092

Table 6.15: FTN and rFTN's CPU Time.

computational time. Besides, the figure shows that the ACPUT results for the latter approach increase in an exponential way, while the former presents a more linear grow rate.

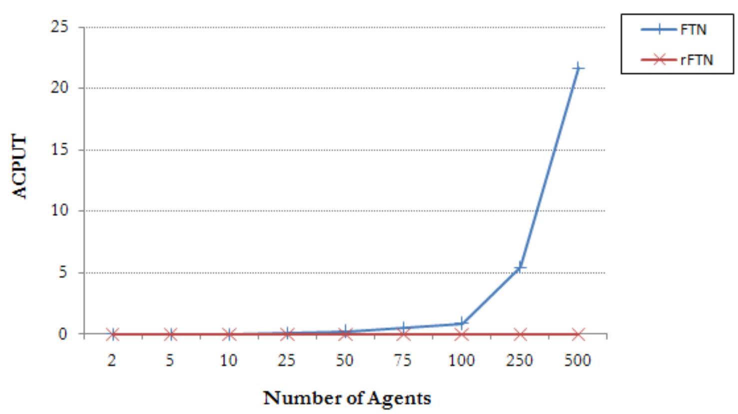


Figure 6.2: Average Computational Time.

As stated before in this document, one of the main objectives of this research work is to design and develop a reactive analogy to the FTN algorithm, which is intended to be "lighter" to process in terms of computational effort. This is why it was of major relevance to compute the amount of CPU time which was required for the solution model to run the final set of experiments and, now, it can be claimed that the design of the rFTN clearly meets this objective.

Besides, the very low values obtained by the rFTN accounts in a positive way for the scalability of the features that this model includes, as more restrictions and/or processes can be added in the form of force-laws, without affecting the overall computational effort required to process the algorithm.

Number of Agents	Average Speed (AS)	Avg Completion Time (ACT)	Avg Wait Time (AWT)
50	0.902	43.240	0.0009
100	0.904	54.348	0.0062
200	0.909	70.831	0.0025
300	0.910	82.864	0.0067
400	0.915	93.873	0.0089

Table 6.16: Summary of the rFTN final results.

6.4 Summary and Conclusions

This Chapter presented the final experimentation of this research work, which was conducted to provide a comparative framework of the solution model's performance against both traditional and advanced traffic management methods, with the objective of providing *validatory data*. This evaluation enables us to actually make conclussions about its fitness as a solution to the urban traffic problem.

The following related-methods were taken into account for these final set of tests: *Traffic-light based navigation*, *Chaotic navigation* and the *FTN* model; the rFTN was compared individually against each one of them. For the first two traditional approaches, the test scenarios included 50, 100, 200, 300 and 400 vehicles, while for the comparison between the rFTN and the FTN the test scenarios were extended to 2, 5, 10, 25, 50, 75, 100, 250 and 500 agents, in order to match the experimental framework used to test the FTN in [4].

The parameters which were of most interest in this validatory phase are the following:

1. Average Speed (AS).
2. Average Wait Time (AWT).
3. Average Completion Time (ACT).
4. Average Computational Time (ACPUT).
5. Saved Trip-Time (STT).

Table 6.16, Table 6.17 and Table 6.18 present a summary of the experiment results for the rFTN, Traffic light and Chaotic navigation models, respectively.

Based on the information provided by the presented tables, we can conclude that the solution model outperforms both traffic lights and a chaotic navigation:

- In the "wait time" column, rFTN is better than the other two methods by more than 3 orders of magnitude;

Number of Agents	Average Speed (AS)	Avg Completion Time (ACT)	Avg Wait Time (AWT)
50	0.418	115.972	23.2920
100	0.395	163.273	63.1694
200	0.323	220.375	85.0973
300	0.289	279.732	102.8412
400	0.246	339.185	146.4592

Table 6.17: Summary of the Traffic-light model final results.

Number of Agents	Average Speed (AS)	Avg Completion Time (ACT)	Avg Wait Time (AWT)
50	0.877	113.813	22.416
100	0.775	161.128	62.1911
200	0.517	233.38	89.6881
300	0.352	284.442	107.9545
400	0.156	396.041	152.6523

Table 6.18: Summary of the Chaotic navigation final results.

- Concerning the average speed, we see that not only rFTN is better than the other two (i.e. the average speed for the rFTN model is almost equal to the maximum allowed speed within the city, while the traffic-light-based model and the chaotic model presented a lower average speed), but also that it gets improving when increasing the number of vehicles, which is the opposite of what happens to traffic lights and a chaotic scheme. This improvement in the rFTN method can be attributed to the fact that with more cars in the scene, flocks are easier to form;

- In completion time we also see an advantage of rFTN over the other two, as well as an increased advantage as the number of vehicles gets higher.

The great advantage of rFTN over traffic lights and chaotic traffic as the number of vehicles increases is an argument for the scalability of this proposed method.

Now, as for advanced models such as the FTN, the solution model has proven to be as effective in terms of the measured traffic-related parameters. Besides, this experimentation phase has provided evidence for the drastically low computational effort that the rFTN demands, which is clearly an advantage over the FTN, and accounts for the model's fitness within real-time aplications, such as traffic handling. Table 6.19 provides a summary of the comparison between both approaches.

	Saving Times		ACPUT	
Agents	**FTN**	**rFTN**	**FTN**	**rFTN**
2	3.17	1.325	0.0149	0.00061
5	12.29	16.712	0.0146	0.00071
10	24.175	24.252	0.0222	0.00089
25	42.15	46.450	0.0769	0.00138
50	64.705	70.573	0.2489	0.00268
75	85.41	77.625	0.5258	0.00396
100	100.915	106.780	0.9116	0.00430
250	204.13	181.635	5.4622	0.01372
500	366.575	333.258	21.7039	0.01830

Table 6.19: Summary of the comparison between FTN and rFTN.

Chapter 7

Conclusions

This Chapter presents the conclusions and contributions that can be drawn from this project, as well as several suggestions to extend the effort undertaken for this research work.

The present document explains an emergent behavior, designed to serve as an urban traffic management method, having the *flocking* approach introduced in [4], as the underlying navigation paradigm, in which vehicles gather up into groups, in order to speed-up their trip to their destinations across the city. The rFTN achieves this this behavior by way of an *implicit coordination* mechanism, embedded in vehicles, as explained in Chapter 3.

The conducted experiments provided full evidence of the high performance of the solution model, concerning both traffic parameters and computational effort required to run the algorithm. Along the experiments several test cases were explored, varying the number of agents within the simulation to know the way it affected the proposed traffic-handling model, and to compare its results against those obtained by different related approaches, both traditional and advanced.

7.1 Conclusions

The present research work focused on the development of a reactive approach to the Flock Traffic Navigation (FTN) method, which still was able to provide the scenario with the benefits specified by that advanced traffic-handling model. It was of major relevance for the designed algorithm to show a very low demand of computational effort, since the FTN's performance on that matter proved to be an opportunity area. This way, this project was aimed to build a traffic-management mechanism, which could be competitive when compared to the large set of solutions to this problem, including the FTN.

The rFTN was designed upon the *Social Potential Fields* approach, which establishes a set of control laws within a distributed-control scenario; such forces are intended to determine a specific behavior for the agents who are going to compute them. In this case, the most important one was for vehicles to create *flocks* in the way described by the FTN, and so the design of the solution model's force laws was oriented to achieve that goal.

Moreover, we can say that the rFTN is the result of several behaviors, or features, designed for this model to perform as expected, such as the following:

1. Flockmates selection.
2. Reactive creation of the Bone Structure.
3. Navigate towards destination.
4. Flocking formation.
5. Collision avoidance.
6. Intersection management.
7. Congestion avoidance.

Two phases of experimentation were designed with different objectives: the Preliminar Experimentation presented in Chapter 5 was aimed to provide *exploratory data*, in order to know the proper values for critical variables of the rFTN to make it perform as expected, while the Final Experimentation provided in Chapter 6 was focused on obtaining *validatory data* about the model's performance when compared against some other solutions, namely: Traffic-Light based navigation, Chaotic navigation and the FTN itself.

From the latter experimentation phase, we can conclude that the drawn results are very favorable for the rFTN as an urban traffic management method. The solution model outperformed both of the selected traditional approaches; the results shown a significant improvement in every traffic-related parameter that was measured during experiments, sometimes by more of three orders of magnitude. Furthermore, the rFTN presented very high saving times, even in the presence of high volumes of traffic, which is a very positive feature of the solution model, since it is in crowded scenarios when the major traffic problems arise, as can be seen by analyzing the results for the traffic-lights and chaotic navigation. Besides, this is consistent with the model's expected behavior, since it was designed to bring benefits to larger flocks, and crowded scenarios bring more possibilities for vehicles to select appropriate flockmates to gather up with.

Even when tested against high-level methods, the rFTN remained as a competitive mechanism, since its results reached the standards of the FTN algorithm, which is very important, as it is a direct reference for the undertaking of this research work. Besides, in some test cases the rFTN slightly outperformed the FTN, and stayed very close to it in the rest of them. Hence, it can be claimed that the solution model is capable of providing almost the same results as the FTN, with the additional characteristic of keeping a safe and realistic scenario by taking into account a limit for the speed at which vehicles are allowed to travel, which is something that is not determined in the FTN.

Moreover, the computational time demanded by the rFTN is shown to be very low, while being compared to that required by the FTN. It should be recalled that the experiments were setup to make vehicles analyze and test all of the rest, in order to select their flockmates (which represents the largest possible work load for the algorithm) and still the greatest amount of time it took for the rFTN to perform its activities was in the order of the centiseconds. As for this parameter, the solution model drastically outperformed the FTN, meeting one of the main objectives established for this reasearch and proving that it is capable of handling real-time scenarios.

7.2 Contributions

This research work contributes to the enrichment of the studies that have been conducted for the urban traffic problem. The measurements drawn from this project also accounts for the extension of the available knowledge for this matter. Besides, the comparative framework provided by the experiments contributes for the understanding of the way the traditional methods can be outperformed by the use of emergent technologies that enable us to develop high-level traffic management approaches. In more specific terms, the contributions of this research work are:

- ***Scientific contributions:***

    1. The development of an innovative and high-level urban traffic-management mechanism, having the FTN as the underlying paradigm, but created upon a completely different approach: a **reactive** approach, instead of a **deliberative** one. This also contributes directly to enabling the analysis of the advantages and disadvantages of both approaches. Besides, it extends the information and knowledge for the urban traffic projects being undertaken within the *Context Intelligence Research Chair* of ITESM, Monterrey Campus.
    2. The development of an intersection management approach, based on the same social potential fields paradigm that was used to design the rest of the rFTN features. An intersection management approach is not included in the FTN, as it assumes the existence of one in the city.
    3. The design of proper flocking-formations, in order to increase the adherence to the flock, and to make vehicles respect the lane restrictions while navigating across the city.

4. An explicit mechanism of collision avoidance, that helps mantain a safe scenario for vehicle navigation.
5. The reactive way of creating the so-called bone structure and the convenient and efficient way of selecting flockmates, which might be taken into consideration for designing a hybrid algorithm between the FTN and rFTN, since the former model's high demand of computational time lies largely in the performance of this process.
6. The solution model includes the process of redirecting traffic away from already-congested streets, which is a feature that is not performed by the FTN.
7. An innovative *acceleration* and *velocity* modulation, computed upon the force vector which results from the summation of the rFTN's force laws. These two processes are different from those performed in several research works, like the one conducted in [13], in which both the acceleration and velocity parameters are just "clipped" to comply with the maximum/minimum values.

- ***Practical contributions:***

1. The development of a Traffic Simulator, which served as an experimental platform for this solution model. It might be scaled to include other traffic approaches, in order to extend the comparative framework and to conduct more experiments to measure their performances.
2. The implementation of two traditional traffic navigation models on the Traffic Simulator, which were added to the comparative framework of this project, namely: the *Traffic-light based navigation* and the *Chaotic navigation*.
3. To provide an analysis of the effects of the rFTN's flocking mechanisms (preliminar experimentation).
4. To provide data about the final results, drawn from the implementation of the solution model and its comparison against the two traditional methods mentioned lines above, as well as the high-level FTN model itself.

7.3 Future Work

Some ideas about the way that the conducted work may be extended are listed in the following:

- To design and test some other potential functions, in order to explore if it is possible to enhance the rFTN's features.

- To develop a complete and in-depth analysis of the advantages and disadvantages of the rFTN and FTN, in order to develop a hybrid approach which can profit from the best features of both models.

- The intersection management mechanism developed within this model assumes that vehicles are able to know the exact position of both the intersections and the rest of the cars in the scenario, at any given moment of time. However, in a real-life scenario, it could be difficult to come up with this information in an accurate and fast way; being this feature very critical in terms of navigation safety, it could be taken into consideration that this model can be extended by the design and the development of an alternative intersection management approach, using different techniques like Fuzzy Logic or Bayesian Networks, which may enable vehicles to perform inferences and forecasts based on less data that the one required in the developed approach.

- In order to extend the experimental testbed, the Traffic Simulator can be scaled-up to include two or more lanes in the city, since, as for now, it only takes into account one-lane streets. This extension will enable the design and development of mechanisms to switch between the *flocking-formations* mentioned in this research, as required by the varying number of lanes in the streets they are currently navigating on, which will result in an interesting flock-coordination mechanism. Furthermore, this issue will contribute to push the model to an even more realistic implementation.

- The blocks in a real city are not always of a squared shape, and so the Traffic Simulator can be extended further to take this issue into consideration to make more realistic experiments, in which case the navigation approach described here might be redesigned.

- Since the presented solution has proven to be a very scalable model, it could be possible to extend its features to include more elements, such as pedestrians or vehicles with a higher priority like ambulances, police patrols, etcetera.

- The final experiments can also be extended to compare the rFTN against several other urban traffic management approaches. The analysis that can be drawn from this evaluation might lead to the detection of opportunity areas, if any, in which case, new research lines can be open to enhance the proposed solution model and to make it even more competitive.

Bibliography

[1] Federal Highway Administration. Traffic congestion and reliability: Linking solutions to problems. Technical report, U.S. Department of Transportation, 2003.

[2] Tucker Balch and Maria Hybinette. Social potentials for scalable multi-robot formations. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 73–80, San Francisco, CA, April 2000.

[3] Michael Balmer, Nurhan Cetin, Kai Nagel, and Bryan K. Raney. Towards truly agent-based traffic and mobility simulations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS' 04)*, pages 60–67, New York, NY, USA, August 2004.

[4] Ramón Brena-Pinero and Carlos Astengo-Noguez. Flock traffic navigation based on negotiation. In *3rd International Conference on Autonomous Robots and Agents (ICARA 2006)*, pages 159–164, 2006.

[5] Encyclopædia Britannica. Traffic control. *Encyclopædia Britannica Online*, 2008.

[6] Adha I. Cahyadi, Chang Ya-Chun, and Yoshio Yamamoto. Stable mobile robots teleoperation via potential field method. In *Proceedings of the 2008 IEEE/ASME International Conference onAdvanced Intelligenct Mechatronics*, pages 347–352, July 2-5 2008.

[7] Christophe Claramunt and Bin Jiang. A qualitative model for the simulation of traffic behaviours in a multi-lane environment. In *Journal of Geographical Sciences*, pages 11:29–42, 2002.

[8] Kurt Dresner and Peter Stone. Multiagent traffic management: An improved intersection control mechanism. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477, Utrecht, The Netherlands, July 2005.

[9] Kurt Dresner and Peter Stone. Multiagent traffic management: An improved intersection control mechanism. In *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477, July 2005.

[10] Patrick A. M. Ehlert and Leon J. M. Rothkrantz. Microscopic traffic simulation with reactive driving agents. In *Proceedings of the 2001 IEEE Intelligent Transportation Systems*, pages 860–865, 2001.

[11] Howie Choset et. al. *Principles of Robot Motion. Theory, Algorithm, and Implementations*. The MIT Press, 2005.

[12] Nicholas J. Garber and Lester A. Hoel. *Traffic and Highway Engineering*. Brooks/Cole Publishing Company, 1999.

[13] Andrew Howard, Maja J. Mataric, and Gaurav S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, pages 299–308, Fukuoka, Japan, June 2002.

[14] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. IEEE Int. Conf. Robotics Automat*, pages 500 – 505, March 1985.

[15] Y. Kitamura, T. Tanaka, F. Dishino, and M. Yachida. 3-d planning in a dynamic environment using a doctree and an artificial potential field. In *Proceedings IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 474–479, 1995.

[16] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[17] Jin Joo Lee and Paul A. Fishwick. Simulation-based real-time decision making for route planning. In *Proceedings of the 27th conference on Winter simulation*, pages 1087–1095. ACM Press, 1995.

[18] Sukhan Lee and George Kardaras. Elastic string based global path planning using neural networks. In *Proceedings IEEE International Simposium on Computational Intelligence in Robotics and Automation.*, pages 108 – 114, 1997.

[19] T. Lomax, S. Turner, and G. Shunk. *Quantifying Congestion*. National Academy Press, 1997.

[20] Marta C. Mora and Josep Tornero. Path planning and trajectory generation using multi-rate predictive artificial potential fields. In *International Conference on Intelligent Robots and Systems*, 2008.

[21] Praveen Paruchuri, Alok Reddy Pullalarevu, and Kamalakar Karlapalem. Multi agent simulation of unorganized traffic. In *Proceedings of the first international*

joint conference on Autonomous agents and multiagent systems: part 1, pages 176–183. ACM Press, 2002.

[22] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robot. Auton. Syst.*, 27:171 – 194, 1999.

[23] Craig Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proceedings of ACM SIGGRAPH '87.*, pages 25 – 34, 1987.

[24] Susan M. Sanchez and Thomas W. Lucas. Exploring the world of agent-based simulations: simple models, complex analyses. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, pages 116–126. Winter Simulation Conference, 2002.

[25] David Schrank and Tim Lomax. Annual urban mobility report. Technical report, Texas Transportation Institute, 2007.

[26] Chaojian Shi, Mingming Zhang, and Jing Peng. Harmonic potential field method for autonomous ship navigation. In *7th. International Conference on ITS*, pages 1–6, 2007.

[27] Gildardo Sánchez and Jean-Claude Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *International Symposium on Robotics Research*, 2001.

[28] Alexander Th. van den Bosch, Maarten R. Menken, Martijn van Breukelen, and Ronald T. van Katwijk. A test bed for multi-agent systems and road traffic management. In *Proceedings of the 15th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC' 03)*, pages 43–50, Nijmegen, The Netherlands, October 2003.

[29] Tzai-Der Wang and Colin Fyfe. Traffic jams: An evolutionary investigation. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, pages 381–384, 2004.

[30] Uri Wilensky. Netlogo. `http://ccl.northwestern.edu/netlogo`, 1999.