TÉCNICO LISBOA

Human machine interface to manually drive rhombic like vehicles such as transport casks in ITER

Pedro Filipe da Silva Lopes

Thesis to obtain the Master of Science Degree in

Engenharia Electrotécnica e de Computadores

Supervisor(s): Doctor Alberto Manuel Martinho Vale
Professor Rodrigo Martins de Matos Ventura

Examination Committee

Chairperson: Professor João Fernando Cardoso Silva Sequeira
Supervisor: Doctor Alberto Manuel Martinho Vale
Member of the Committee: Professor Fernando Gómez Bravo

November 2015

Aos meus pais

À Vânia

Acknowledgments

First would like to thank my supervisor and ”boss”, Doctor Alberto Vale, for all the support, guidance, availability, and especially for caring with the success of his student. I also thank to Professor Rodrigo Ventura, for his concern and helpful comments.

I would like to thank Doctor Bruno Gonçalves and IPFN as a whole for all the material and logistic support, and for giving me the opportunity to partake in a scientific research project.

A big thanks to all my friends for all the good times that we passed in and out of classes, even the guys that majored in Energy.

I would like to thank my family especially my mother and father, for raising me like they did and for all the love and support they gave me during my life. Lastly I would like to thank my S.O., my girlfriend Vânia Cardoso, that was by my side along this tempestuous journey of taking a Masters Degree at IST, thanks for all the love and patience.

Resumo

O futuro da geração de energia passa pela fusão nuclear que é um processo sustentável e seguro de produção de energia, no entanto liberta alguma radiação, activando os materiais próximos. O transporte destes materiais tem que ser feito com o auxilio de manipuladores e veículos autónomos. É o que irá acontecer no ITER, o transporte das cargas vai ser efectuado remotamente por um sistema automático de transporte de carga, o CTS.

O CTS é um veiculo de grandes dimensões e com configuração que o torna omni-direccional. Em operações nominais trabalha em modo automático, contudo, surgirão certamente situações em que vai ser necessário a condução do veiculo remotamente.

Esta tese insere-se na necessidade de conduzir o veiculo manualmente da forma mais eficiente e segura, não comprometendo a integridade do veiculo, carga e do edifício em opera. Num ambiente industrial usual este veículo poderia ser conduzido com o operador nas proximidades do veiculo, contudo no ITER isto não é possível e o operador tem que controlar o veiculo remotamente num local dedicado para esta tarefa. Esta tese compara duas maneiras de conduzir este tipo de veículos: Focando-se nas rodas, que é a mais comum, ou focando-se no centro do veiculo, tendo como objectivo provar a viabilidade da segunda.

No decorrer desta Tese foram desenvolvidos dois dispositivos para conduzir o veiculo, um foca-se nas suas rodas e usa um gamepad, o outro foca-se no centro do veiculo e usa um joystick acompanhado por um disco rotacional construído de raiz para este efeito utilizando um encoder, montado num suporte impresso em 3D e ligado a um micro-controlador.

Para avaliar a viabilidade de cada dispositivo foram feitos um total de 200 testes feitos por 12 pessoas que nunca tiveram contacto com o veiculo ou os dispositivos. Cada utilizador conduziu o veiculo com os dois dispositivos, a dinâmica e sistema de visualização do veiculo foi simulada em Matlab.

Palavras-chave: Fusão nuclear, ITER, Comando manual, Veiculo Autónomo, Centro de rotação instantâneo, Impressão 3D, Teste de usabilidade.

Abstract

The future of energy generation goes through nuclear fusion which is a sustainable and safe method to produce energy, unfortunately releases some radiation, which activates nearby materials. Handling of these materials must be done by manipulators and autonomous vehicles. This will happen at ITER, the cargo transfer operations will be done remotely by the Cask Transfer System, CTS.

CTS is a large vehicle with a configuration that makes it omni-directional. In nominal operations it is autonomous, however, certainly will occur situations where the vehicle must be driven remotely by an operator in manual mode.

This thesis arises from the need to have an efficient and safe way to manually drive the vehicle, that does not compromises the physical integrity of the vehicle, cargo and building. In a common industrial workplace these types of vehicles are usually driven with an operator near the vehicle, however in ITER because of the radiation it must be remotely operated.

This thesis compares two methods to drive the vehicle, one focused on the wheels, which is more common, and other which focuses on the vehicle center, with the objective of discover which is best.

Along this thesis, two devices were developed to drive the vehicle, one that focuses on the vehicle wheels and uses a gamepad, and other that focus on the vehicle center with a joystick plus a rotational disc that was custom built and has an encoder mounted on a 3D printed support which is connected to a micro-controller.

To evaluate each device a total of 200 tests were taken by 12 people that never had any contact with the devices neither the vehicle. Each user drove the vehicle with both devices, the dynamics and visualization system was simulated with Matlab.

Keywords: Nuclear Fusion, ITER, Manual driving, Autonomous vehicle, Instantaneous center of rotation, 3D printing, Usability tests

Contents

List of Tables

List of Figures

Nomenclature

$\dot{\psi}$	Vehicle angular velocity.
$\dot{\psi}_F$	Front wheel angular velocity.
$\dot{\psi}_R$	Rear wheel angular velocity.
ψ	Vehicle heading.
θ	Orientation of the center of the vehicle velocity vector.
θ_F	Orientation of the front wheel velocity vector.
θ_R	Orientation of the rear wheel velocity vector.
C	Vehicle center.
Fw	Vehicle front wheel.
M	Longitudinal length of the vehicle.
M_F	Distance between the vehicle center and front wheel.
M_R	Distance between the vehicle center and rear wheel.
R_C	Radius of the circumference between the vehicle center and the ICR
R_F	Radius of the circumference between the front wheel and the ICR
R_R	Radius of the circumference between the rear wheel and the ICR
Rw	Vehicle rear wheel.
v	Value of the center of the vehicle velocity vector.
v_F	Value of the front wheel velocity vector.
v_R	Value of the rear wheel velocity vector.
ICR	Point of the instantaneous center of rotation

Glossary

CPRHS	Cask and Plug Remote Handling System, the system in charge of the remote handling of activated cargo at ITER.
CTS	Cask Transfer System, a mobile robot with a rhombic-like configuration.
Energy Fw	Variable used to measure the stability of the front wheel.
Energy Rw	Variable used to measure the stability of the rear wheel.
GPad v.1	Notation used for Gamepad v.1 in graphs and tables.
GPad v.2	Notation used for Gamepad v.2 in graphs and tables.
Gamepad v.1	A device to drive the vehicle focusing in its wheels, was developed at IPFN in the past.
Gamepad v.2	A device to drive the vehicle focusing in its wheels,it is the upgraded version of Gamepad v.1, it was developed for this thesis.
HCB	Hot Cell Building, the maintenance and refurbishment building of ITER.
IPFN	Instituto de Plasmas e Fusão Nuclear, the research group involved with the ITER project in Portugal.
ITER	Iternational Thermonuclear Experimental Reactor, first energy efficient fusion reacto, currently under development.
JRD	Joystick and Rotational Disc, a deviced developed in this thesis used to drive a rhombic vehicle at its center.
RH	Remote handling.

Rhombic-like vehicle	A type of vehicle with two steerable and drivable wheels across its longitudinal axis.
TB	Tokamak Building, where the fusion reactor is housed in ITER.
TES	Trajectory Evaluator and Simulator, the simulator for rhombic-like vehicles that was used in this thesis, developed at IPFN.
Usability test	The technic used to test each device by being used by other people .
VV	Vacuum Vessel, the inside of the tokamak reactor, where the plasma is contained.

Chapter 1

Introduction

The scope of this thesis is to develop an efficient device to remotely drive land vehicles with rhombic-like configuration. A vehicle such as that has two wheels across its longitudinal axis, both drivable and steerable [1], making it an omni-directional vehicle. This thesis will address the manual driving mode of a specific rhombic vehicle, the Cask Transfer System (CTS) which will operate in the International Thermonuclear Experimental Reactor (ITER) in Cadarache, France. During nominal tasks it transports activated cargo and parts required for the reactor maintenance, these components need to be changed periodically due to wear and tear caused by reactor operations.

The environmental conditions where the vehicle typically operates are very harsh due to high levels of radiation. Because of this, no operators are allowed inside the building or near the vehicle in order to drive it.

The vehicle is supposed to have three operating modes: automatic mode, where the vehicle moves from one place to another completely autonomously with a pre-established route/mission controlled by software; semi-automatic, where the velocity of the vehicle is controlled manually and the orientation is given by software; and the situation addressed in this thesis, the manual operating mode where the vehicle velocity and orientation is manually controlled by an operator, this task is dangerous because the vehicle transports activated cargo which can be damaged. This thesis aims to develop a good and safe method to manually drive these types of vehicles.

1.1 Nuclear fusion

1.1.1 What is nuclear fusion?

Nuclear fusion can be seen as the most basic form of energy in the universe. It is what happens on all the stars of the universe [2]. This should mean that in order to do nuclear fusion here on earth a synthetic sun should be built. Hopefully that is not the case, the fusion reaction process can be achieved due to advancements in physics and engineering. Special types o nuclear reactors can achieve nuclear fusion either by magnetic, inertial confinement, or by an hybrid reaction [3].

Research on nuclear fusion started during the 1950's in Harwell, UK where the Zero Energy Toroidal

Assembly (ZETA) worked from 1954 to 1958, however the major step occurred in 1968 where two Russian scientists, Igor Tamm and Andrei Sakharov announced the results of a magnetic confinement device called tokamak. Today, the tokamak is the dominant experimental technique to study fusion [4].

A tokamak is a magnetic confinement type of reactor, meaning that it uses a magnetic field to confine the hot plasma, since no solid material could withstand the extremely high temperature of the plasma.

Temperature is a measure of the average kinetic energy of particles, so by heating a material it will gain energy. At extreme temperatures, the electrons are separated from the nuclei and a gas becomes a plasma which is very hot and electrically charged [5]. By heating the material until it reaches a certain temperature, the energy of accidental collisions within the plasma is high enough to make the particles fuse together.

Although different isotopes of light elements can be paired to achieve fusion, the deuterium-tritium (D-T) reaction has been identified as the most efficient for fusion devices. Deuterium can be distilled from all forms of water, which means its is widely available considering that 71 percent of the earth surface is covered by oceans. Tritium is a fast-decaying radioelement of hydrogen which can only be produced during the fusion reaction when in contact with lithium [6].

The deuterium-tritium fusion during its reaction, produces one helium, one neutron and energy. In order to fuse both elements, the energy necessary is 0.1 MeV, which corresponds to 1.2 billion Kelvin. The helium nucleus carries an electric charge which will respond to the magnetic fields of the tokamak and remain confined within the plasma, however the remaining neutrons will not since they are not electrically charged and will be absorbed by the surrounding walls of the tokamak and transfer their energy as heat.

1.1.2 ITER

The ITER project was born in 1985 at the Geneva Superpower Summit. The initial members were the former Soviet Union, the USA, the European Union and Japan, later, the Republic of Korea and the People's Republic of China in 2003, and India in 2005 [7]. Currently the International Thermonuclear Experimental Reactor, ITER is being built in Cadarache France.

ITER is a joint international research and development project that aims to demonstrate the scientific and technical feasibility of fusion power, by building the world's largest experimental tokamak fusion reactor.

The goal at ITER is to build the first energy sustainable fusion reactor, outputting at least ten times more energy than what is needed to start the reaction, it was designed to produce 500 megawatts of power while needing only 50 megawatts to operate.

Currently the record for higher energy release is held by The Joint European Torus (JET) in Culham, U.K, that generates 70% of the energy intake. Other reactors that hold some sort of record are: The Tore Supra tokamak that is part of the Cadarache nuclear research centre, which holds the record for the longest plasma duration time, six minutes and 30 seconds; the Japanese JT-60 achieved the highest value of fusion triple product, which is density, temperature and confinement time of the plasma. On

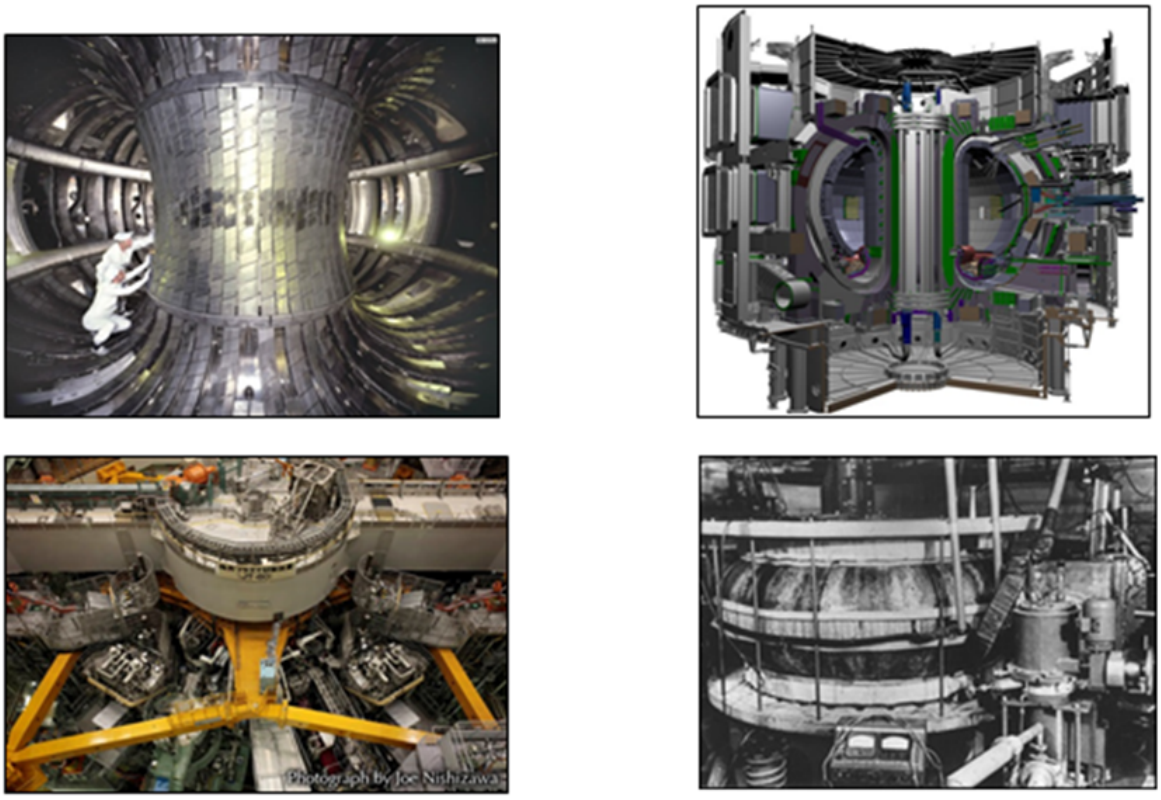

Figure 1.1: Tokamak reactors

Figure 1.1 are some examples of tokamak reactors, on top left the inside of JET reactor torus, on top right the prototype of ITER tokamak reactor, on the bottom left the inside of JT-60 building, on the bottom right the first tokamak reactor.

1.2 ITER remote handling activities

1.2.1 Remote handling systems

At ITER the components that suffer from wear and tear must be inspected and maintained. Because of high levels of radioactivity caused by Deuterium-Tritium pulses the materials in the proximity of the plasma become activated and need to be remote handled with the help of manipulators and autonomous vehicles. The complexity associated with these operations is very high and proves to be one of the project major challenges.

Regarding the transport of the activated cargo, there are two main buildings in which the vehicles will operate, the tokamak building (TB) and the hot cell building (HCB). The first is where the tokamak reactor is installed, and in its cluttered halls is where an autonomous vehicle will operate moving cargo from the Vacuum Vessel (VV) to the HCB, which houses the maintenance and refurbishment operations of reactor parts and the vehicle itself. These two buildings can be seen in Figure 1.2, on the left image the TB and on the right the HCB.

The main components that need to be remotely handled can be seen in Figure 1.3, on the left

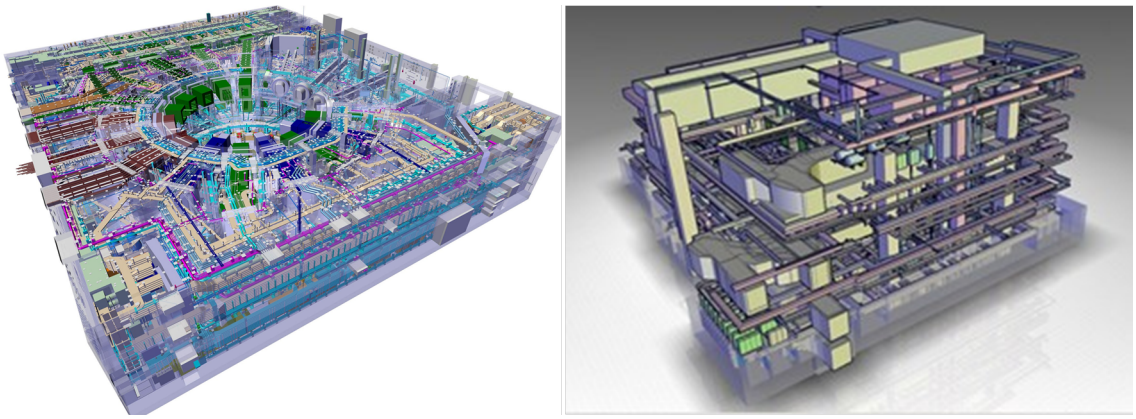

Figure 1.2: Tokamak and hot cell buildings

the port plugs, on right the divertor cassettes and on the center the blanket modules. These need to be transported from the TB where they are taken out from the VV, which is the torus of the tokamak, via a special manipulator and are transported to the HCB by a rhombic vehicle for maintenance and refurbishment [8].

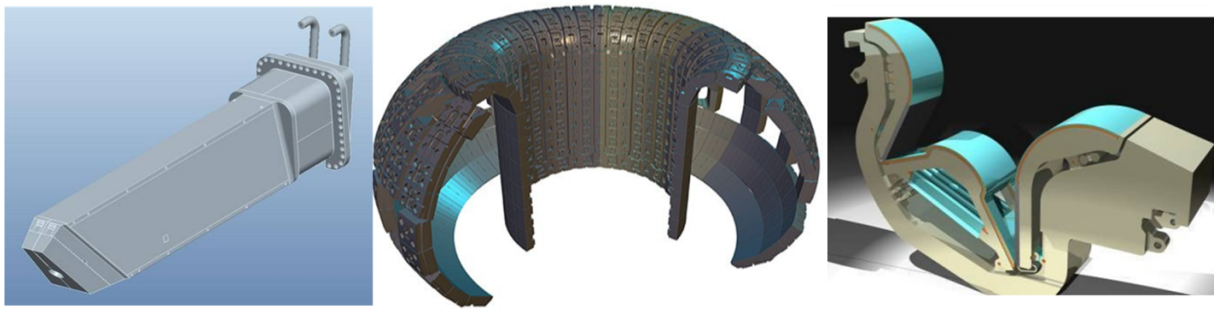

Figure 1.3: Port plug, divertor and blanket modules

1.2.2 Rhombic-like vehicles

The system in charge of transporting the activated cargo inside ITER is the Cask and Plug Remote Handling System (CPRHS) [9]. The system and its different parts can be seen in Figure 1.4.

The CPRHS needs to transport different types of containers, depending on its mission the cargo size and weight may change. It will operate in three different floors of the TB and in the HCB. Access between floors is made via a lift that links three levels of TB with the HCB, levels B1, L1 and L2, the CPRHS on a mission can be seen in Figure 1.5.

The CPRHS can be decomposed in three different parts, Figure 1.4:

- **The cask envelope** - Container that enclosures the in-vessel components of the remote handle tools to be transported, this is the component that will transport the activated cargo.
- **The Cask Transfer System** (CTS) - The CTS is the autonomous vehicle that will transport the cask envelope from the TB to the HCB. It has a rhombic-like configuration with two wheels along its longitudinal axis, both capable of steering and driving.

- **The pallet** - This component serves as interface between the CTS and the cask envelope. The pallet is equipped with a platform to support the cask load and help the docking procedures.

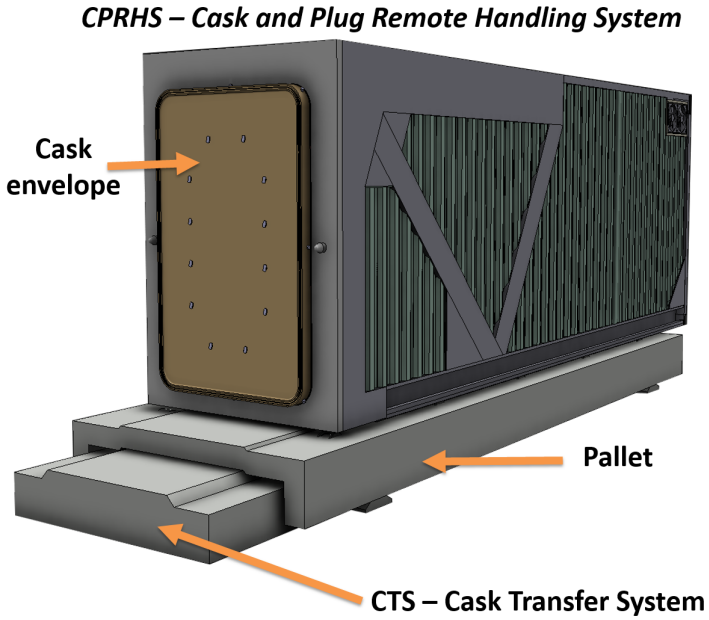

Figure 1.4: CPRHS components

Because of TB's layout complexity, the CTS needs to be an omni-directional vehicle, with high maneuverability and flexibility

Its maneuverability allows for very good driving capabilities, specially when the vehicle is being controlled by software in automatic mode. Having to manually control two wheels at the same time can be tiring and cumbersome to a human operator, so the manual operation mode of the vehicle needs to be simple and intuitive.

1.2.3 CTS operation modes

The CTS is expected to have three operation modes, automatic, semi-automatic and manual mode.

- **Automatic mode:** The vehicle is completely autonomous and follows a set of instructions, which boil down to following a trajectory. The CTS is supposed to operate in this mode when executing nominal operations, which consist in repetitive remote handling operations, excluding periodic maintenances.

- **Semi-automatic mode:** The vehicle orientation is controlled automatically and wheel velocities are manually controlled by an operator, meaning that the vehicle follows a pre-computed path and the operator commands the velocity of each wheel. This mode is to be used when the vehicle needs to recover from a minor navigation failure and needs to resume nominal operations, or if the risk of collision with a building wall is high.

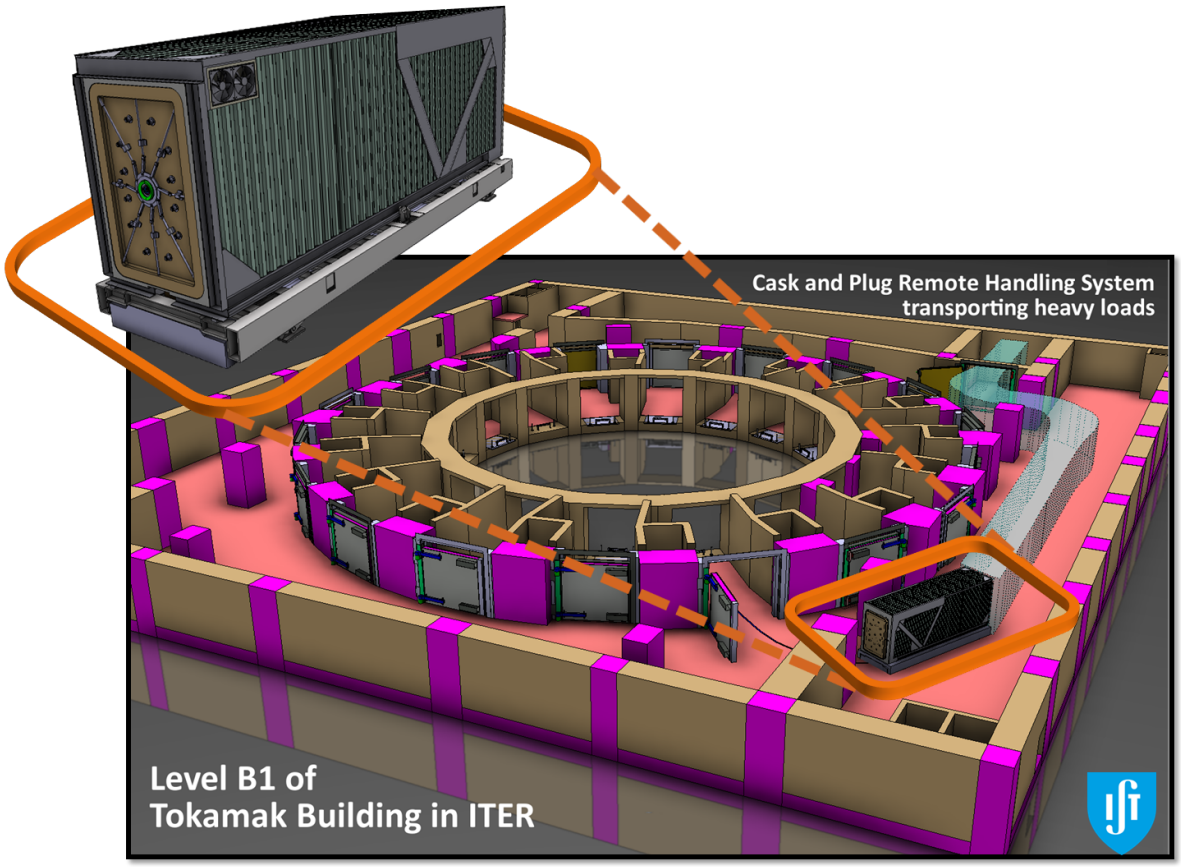

Figure 1.5: The ex-vessel CPRHS and a typical RH mission

- **Manual mode:** The orientation and velocity of each wheel is manually controlled by an operator. A total of four variables need to be managed, which is not trivial and could be dangerous to the vehicle and reactor. Hence, this mode is expected to be only used in cases of a major navigation failure and an operator needs to retrieve the vehicle to a safe place, or be used in rescue operations.

In all modes, the CTS requires a localization system. In the case of the manual operation mode where the reference frame is the world, the system needs to know what is the vehicle position and heading. Information on the whereabouts of the vehicle can only be given by sensors installed on the scenario and on the vehicle, like cameras or lasers. Cameras give the live visual feed of the building, and laser range finders (LIDAR) installed on the building walls allow the reconstruction of the scenario, and with proper software the vehicle orientation can be found by methods shown in [10].

1.3 Manual driving mode

1.3.1 Robots in Industry

ITER despite being an experimental facility with scientific and research goals can be considered an industrial workplace. Most of the robots in industry are used in handling, welding and assembly applications. The definition of industrial robot is the following: ”An industrial robot is an automatically controlled,

reprogrammable, multipurpose manipulator programmable in three or more axes." [11]. In this definition, a vehicle used to autonomously transport cargo is not considered an industrial robot.

In the case of industrial robot manipulators, the manual controls are sent via a hand-held device called teach pendant which is mainly used to program the robot list of instructions [12]. The teach pendant is the interface between the human and the machine, and it is equipped with a dead man's switch, which is a device that assures safety by stopping the robot if something unexpected happens to the operator [13].

In the case of autonomous vehicles, they are not programmed via a teach pendant like the serial manipulators, the system controlling them is also supervising them. The manual driving mode is widely used in opposition of automatic modes when the vehicle in question needs to transport heavy cargo and does not perform repetitive tasks, an example of this can be seen in Figure 1.6 on the left image, where an A-VT air cushion transporter is transferring Vega booster into ESA launch assembly.

Figure 1.6: Vega booster being transferred and Kuka's omnimove controller

Unlike serial manipulators where there are many axes to control, most of land vehicles used in industry can be simplified to control one or two points, its wheels. Because of this joysticks or buttons can be used to drive the vehicle. In common situations it is possible to have the operator near the vehicle driving it, which is impossible in ITER.

Kuka is one of the world's leading suppliers of intelligent automation solutions and one of its autonomous vehicles is the Omnimove [14], it is a vehicle to transport heavy loads. It has mecanum wheels which makes it an holonomic vehicle. It is driven using a 3-axis joystick, the first two axis of the joystick are used control the xy position of the vehicle and the third axis controls its heading, with this device the driving operation is simplified to control only the center of the vehicle and its heading. This device can be seen in Figure 1.6, on the right side.

Solving is a company specialized in the design and manufacture of heavy load handling systems, and one of its transporters has a different approach to the driving method. It uses two 2-axis joysticks to control each wheel. The vehicle in question has two steerable and drivable wheels across its longitudinal axis, like the CTS. The device and vehicle design are shown in Figure 1.7. The device has multiple

functionalities that allow for a better driving experience.

Figure 1.7: Solving's AGV on the left and manual controller on the right

1.3.2 Devices and technologies

Following is a list of possible technologies that can be used as part of a device to manually drive a vehicle.

- **Keyboard:** Keyboard technology can be dated to the 1700s with the first typewriters, however first consumer keyboards were only available in 1970s. Nowadays they are accessible to anyone, and every computer uses one. The keyboard is a digital input device, with interrupt or pressure buttons, and lacks the sensibility that a analog device possess. To change the velocity or orientation values, the operator would have to keep pressing a button or press it multiple times. Nonetheless it has a large number of input possibilities, the standard keyboard as 104 keys, with this high amount of possible inputs it should be able to support any type o driving method.

- **2-axis Joysticks:** A 2-axis Joystick can be an analog or digital input device with two axis, that outputs a vector in the direction in which the Joystick is pointing. The analog joystick uses potentiometers while digital rely on single electrical connections. It is a mature technology, it was patented in 1926 by the United States Naval Research Laboratory [15]. It has many applications as a controller device, it is the principal control device inside an aircraft cockpit, and nowadays every gaming console has at least one built in their controllers. The analog version of device, has the sensibility needed to control the CTS. As a single device it can only send signals to manually control one of the vehicle wheels, however when paired with another, like the case of a gamepad it can be used to manually control both wheels or the vehicle's center.

- **3-axis Joystick:** The 3-axis Joystick is a 2-axis joystick with a rotation axis on top. To control the vehicle linear velocity the 2-axis joystick would be used and the third axis would control the vehicle heading. This device is widely used around industry, mainly in surveillance applications. This device can be used to control the vehicle only at its center.

- **Touch-screen:** Touch screens are used in industry for some time, mainly on displaying interfaces, the first transparent touch screen was developed by CERN in the early 1970s.This technology

Figure 1.8: Different technologies for control devices

can be better used for a high level control of a vehicle fleet, where a path is drawn in the touch interface with the finger and the vehicles are commanded to follow it. However this is not a true manual mode, the system still needs to compute a trajectory and chose what commands need to be sent to the vehicle, an example of this is shown in [16], where multiple robots are controlled using a large touch interface in a supervisor mode. Other alternative is to simulate a joystick or directional pad and use the touchscreen as the input device, this was used in [17], where a mobile land vehicle is being controlled with a PDA, its touch-screen is emulating a directional pad, with up-down, left-right commands, this method lacks the sensibility and feedback needed to drive a rhombic-like vehicle.

- **Motion sensing and Virtual Reality:** Motion sensing and virtual reality are together. To assure the immersion of virtual reality at least motion sensing technology must be installed on the head mounted display (HMD). This type of technology would allow the best viewing system, the operator is able to ”be” the vehicle. Currently, virtual reality is being used more often, since Oculus VR company decided to build a cheap and efficient HMD which is largely used in video-games. This type of technology is still not used largely in industry. The use of an Oculus Rift to control a mobile robot can be seen in [18] and [19], however one know issue with HMD technology is the fatigue that causes on the operator, displaying symptoms of simulator sickness [20].

In Figure 1.8 on top is a computer keyboard, on the middle row from left to right: a Microsoft Sidewinder 2-axis Joystick, a NGS Maverick gamepad that includes two 2-axis Joysticks and a 3-axis Joystick. On the bottom row: on left a touch screen and on the right an Oculus Rift.

1.3.3 Viewing system

In situations where the operator is unable to be near the vehicle in order to drive it, like in ITER, a viewing system is needed. Instead of the typical first person-view, that everyone experiences while driving a car, the proposed viewing system is a third person top-side view of the scenario.

The viewing system must rely on a virtual reality reconstruction of the scenario, which can be made with 3D generated graphics or via video camera feed. Either way the vehicle's orientation and position is needed, and one possible way to know them, is to install laser range finders (LIDAR) in the building walls, like what was done in [10], where a rhombic-like vehicle location is obtained using LIDAR sensors.

The system cannot rely on the video cameras since they may not cover the entire building and probably have low resolutions in order to withstand radiation. Since the vehicle is omni-directional, the operator needs to be aware of the vehicle surroundings. A top-side view should be better than a first-person view in this particular case.

In Figure 1.9 it is shown the top-side view of part of the TB gallery of ITER.

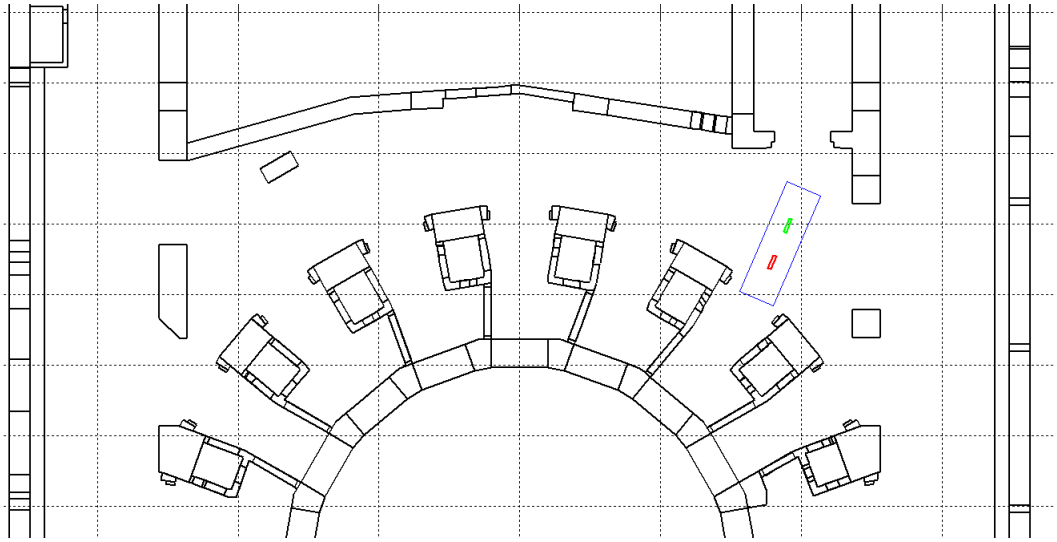

Figure 1.9: Top view of the tokamak building in the simulation software

1.4 Problem statement

This thesis follows the work presented in [21], where the first steps in studying the manual driving mode of rhombic-like vehicles were taken. In that work the vehicle was being driven by sending commands directly to each wheel with a gamepad The gamepad's two 2-axis analog joysticks send commands to change the direction of the rear and front wheels, and the left and right trigger buttons send commands to increase and decrease the wheel velocities, this setup made the vehicle drivable, however driving with this device seems too complex.

One problem that an operator driving the vehicle focused in each wheel faces, is that he becomes confused and switches what joystick is commanding which wheel. One way to avoid this, is to lock the

direction of one wheel and focus on the other, this workaround transforms a vehicle with rhombic-like configuration into a vehicle that can be modeled by a bicycle model.

In Figure 1.10 is shown the comparison in the driving a car, on the left side, vs a rhombic-like vehicle, on the right side, using steering wheels and pedals as devices. The steering wheel changes the wheel orientations and the pedals increase or decrease the velocity of the vehicle. A user will need to manage two steering wheels and two sets of pedals in order to drive a rhombic-like vehicle. This can be seen on the right side of the Figure 1.10.

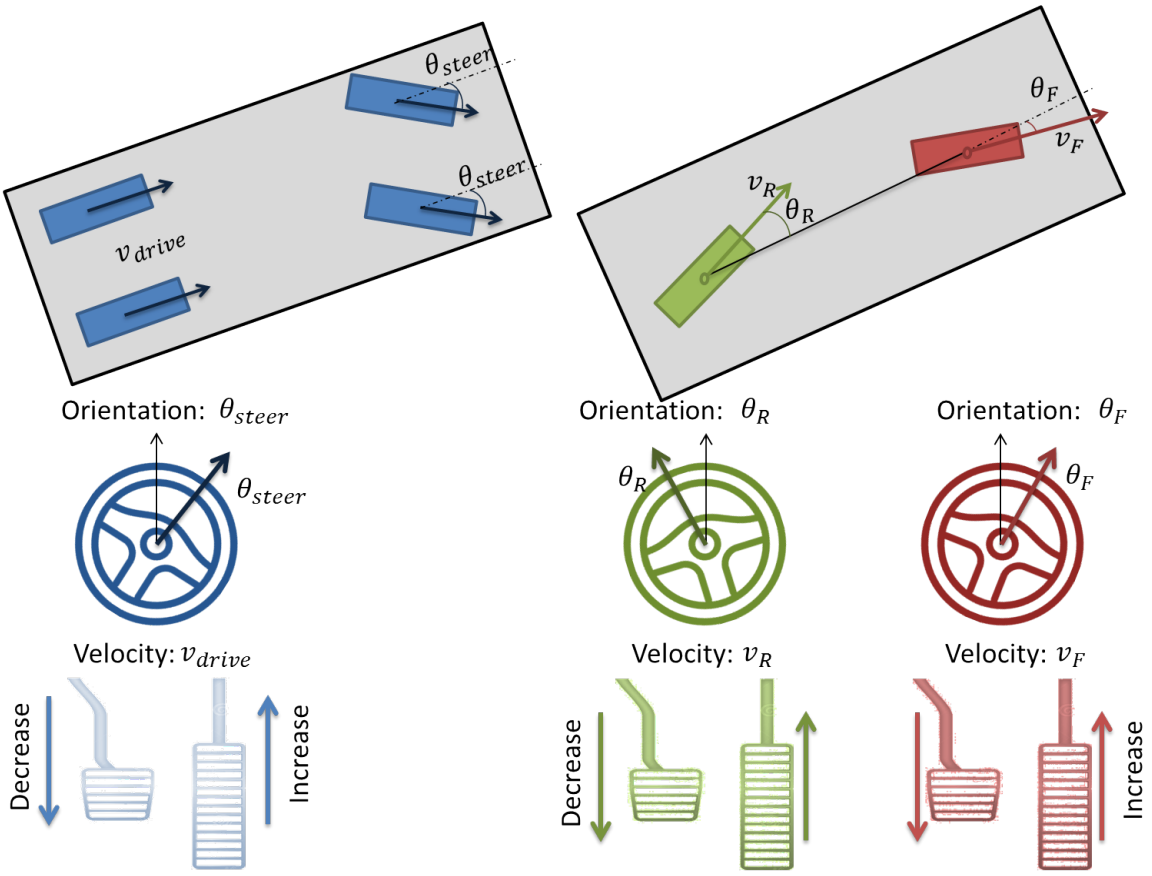

Figure 1.10: Comparison between a car and a rhombic like vehicle

The proposed solution is to make the focus of control a point at the center of the vehicle, by doing this, instead of sending commands to the wheels, the vehicle is being driven on its center, and the user is oblivious to the variables of the two wheels. At its center the operator only needs to manage three variables the center velocity value and orientation, and the vehicle heading, at its wheels the operator needs to manage four variables each wheel orientation and velocity.

By driving the vehicle at its center it is expected to make the vehicle easier to drive. However, this limits the vehicle motion space, so this driving method needs to be evaluated in order to check if it is suited to be used with a rhombic-like vehicle such as the CTS.

1.5 Thesis structure

The thesis is structured as follows, in Chapter 2 the kinematic model of the vehicle is obtained using the instantaneous center of rotation. It is explained the difference between driving the vehicle at its wheel versus at its center. Are shown two concepts for a devices to drive the vehicle, one device to command each wheel independently and other to drive the vehicle by sending commands to its center, it is also shown a possible model for a control stand for this device. Also are shown three methods to send commands to the vehicle center that were evaluated via a simulation.

In Chapter 3 is explained what was developed during this thesis, the steps took to implement the concepts of Chapter 2 transforming them into real devices. It is introduced the simulator that is used for the interface and vehicle dynamics. It is explained how the devices were built, hardware and software. Lastly a support for the encoder was 3D printed and is shown in the last section along with additional features.

In Chapter 4 are shown the results obtained from testing the devices of Chapter 3 using the Alonzo Kelly method. It is introduced how the usability tests will take place. Two different tests were evaluated, the first with an experienced user, the author of this thesis which drove the vehicle with three devices in three different scenarios, and the second with a group of 12 people without any experience driving rhombic-like vehicles which evaluated the performance of two devices in one scenario. This Chapter is where the best method to drive the vehicle is found.

In Chapter 5 is presented the conclusions obtained from all the work developed during this thesis, as well future work that can be done to achieve more accurate results.

Chapter 2

Conceptual Design

In this Chapter it is shown the vehicle model and its kinematics that were obtained using the Instantaneous Center of Rotation. The concept behind driving the vehicle focusing in each wheel independently or driving focusing in its center, the pros and cons of each method. This chapter is the backbone for all that was developed along this thesis.

2.1 Vehicle kinematic model

2.1.1 Vehicle model

The rhombic-like vehicle that is the focus target in this thesis is the CTS. Its simplified model is shown on Figure 2.1 on the left image, which only takes into account the vehicle 2D configuration on an xy plane.

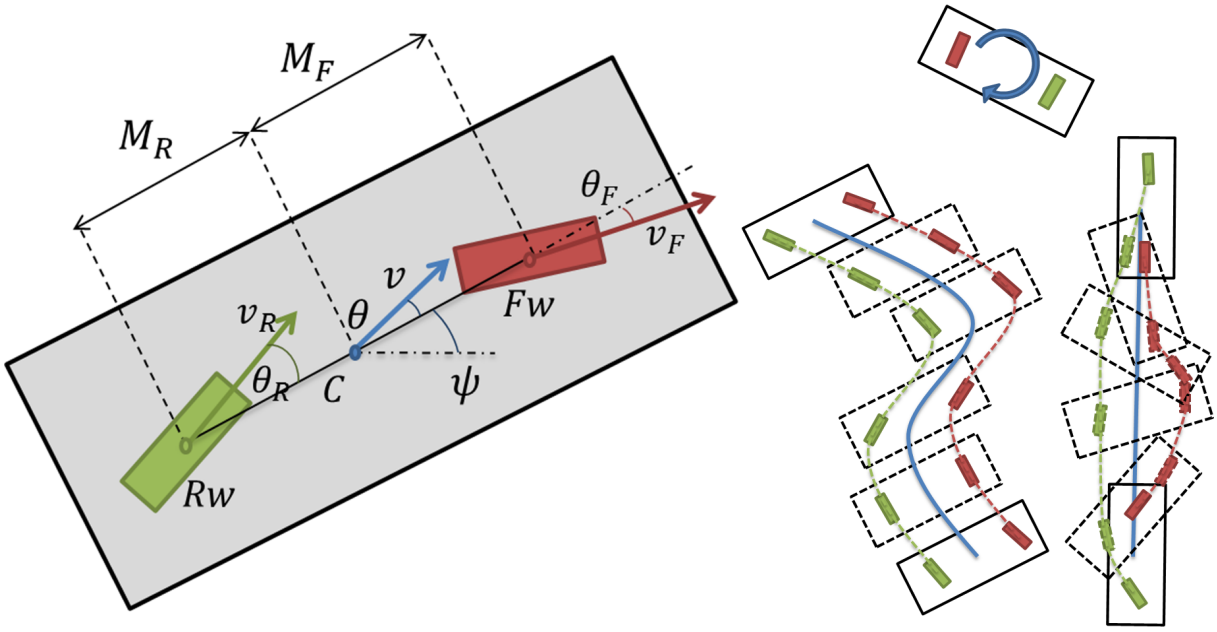

Figure 2.1: Vehicle model and motion capabilities.

1. **Vehicle's body variables**

- Fw - Front wheel
- Rw - Rear wheel
- C - Center of the vehicle
- M_F - Distance between the C and Fw.
- M_R - Distance between the C and Rw.

2. **Vehicle's wheel variables**

   - v_F - Front wheel velocity
   - v_R - Rear wheel velocity
   - θ_F - Front wheel orientation
   - θ_R - Rear wheel orientation

3. **Vehicle's center variables**

   - v_C - Center velocity value
   - θ - Center orientation
   - ψ - Vehicle heading

A rhombic-like vehicle is controlled by changing θ_F, θ_R, v_F and v_R. The vehicle is omni-directional because both wheels are steerable and drivable, the vehicle can perform crab-like movements, rotate in the same place and a combination of the two, which can be seen in the right image of Figure 2.1. The wheels can have any orientation and velocity, this can put the vehicle in situations that can cause wheel slippage or extra strain in the motor. An example of this is when θ_F and θ_R have a difference of 180 degrees and v_F and v_R have the same values.

2.1.2 Instantaneous center of rotation (ICR)

Previous studies on the kinematics of rhombic-like vehicles were done in [22] and the resulting system of equations can be seen in (2.1), where the vehicle's center position is given by the vector $[x_G, y_G]$, and $[\dot{x_G}, \dot{y_G}]$ are the first derivatives of the vehicle's center position.

$$\begin{bmatrix} \dot{x_G} \\ \dot{y_G} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{M_R.\cos(\psi+\theta_F)}{M} \\ \frac{M_R.\sin(\psi+\theta_F)}{M} \\ \frac{\sin(\theta_F)}{M} \end{bmatrix} .v_F + \begin{bmatrix} \frac{M_F.\cos(\psi+\theta_R)}{M} \\ \frac{M_F.\sin(\psi+\theta_R)}{M} \\ -\frac{\sin(\theta_F)}{M} \end{bmatrix} .v_R \tag{2.1}$$

A different system was developed in this thesis using the Instantaneous Center of Rotation (ICR). This change of approach will result in a different set of equations for the vehicle kinematic model.

ICR is a model where any moving particle is, at all times, describing a circumference [23]. Which is tangent to the velocity vector of the particle, its center is somewhere on a straight line orthogonal to the velocity vector that contains the particle.

If a body only has one particle it is impossible to know where the ICR is located, and therefore the length of the circumference radius is unknown. However, by applying the same method to two or more points of a rigid body, the ICR position and radius can be found. There are two particular situations in this method: One, if the body is moving on a straight line, the ICR is placed at infinity and the circumference radius i s infinite. Two, when the vehicle is rotating in the same place, the ICR is placed at the center of the vehicle and the circumference radius is in between those points.

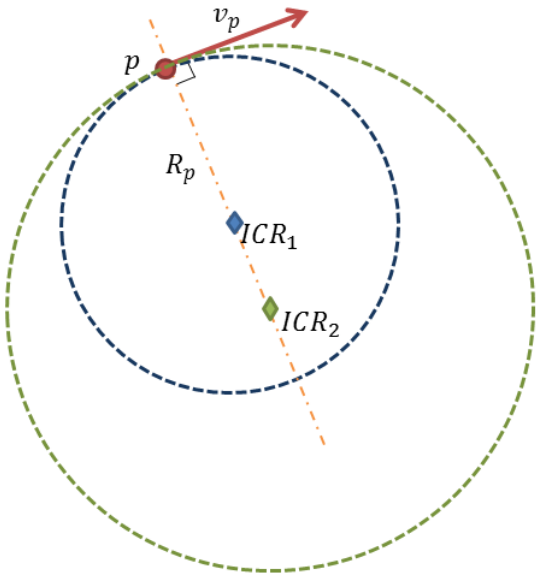

Figure 2.2: Instantaneous center of rotation, single particle.

In Figure 2.2 the particle p with velocity v_p can describe an infinite number of circumferences all of them along R_p which is a line orthogonal to v_p and containing p.

2.1.3 ICR - Applied to rhombic-like vehicles

Rhombic-like vehicles have four degrees of freedom (4 DoF), front wheel velocity and orientation v_F and θ_F, rear wheel velocity and orientation v_R and θ_R.

Some assumptions were made that constrain the vehicle kinematic model. It is assumed that the vehicle wheels do not suffer from slippage, this means that the velocity vectors for both wheels are aligned with the wheel directions. The vectors v_F and v_R make,respectively, an angle θ_F and θ_R with the vehicle longitudinal axis. This is a valid assumption given that the vehicle moves at low speeds and because of this, the lateral forces generated by the tires are small [24].

Other assumption is that the vehicle body is rigid, meaning that the distance between both wheels is always the same. The wheels do not drift away from the vehicle body as it is moving. While rotating, all the vehicle particles have the same angular velocity. If the vehicle had different angular velocities along itself, it either means the vehicle is disintegrating or the wheels are slipping.

On Figure 2.3 the vehicle model shows the ICR variables that are necessary to obtain the vehicle kinematics. Some applications of this method were used to control four wheeled steerable vehicles in [25] and [26].

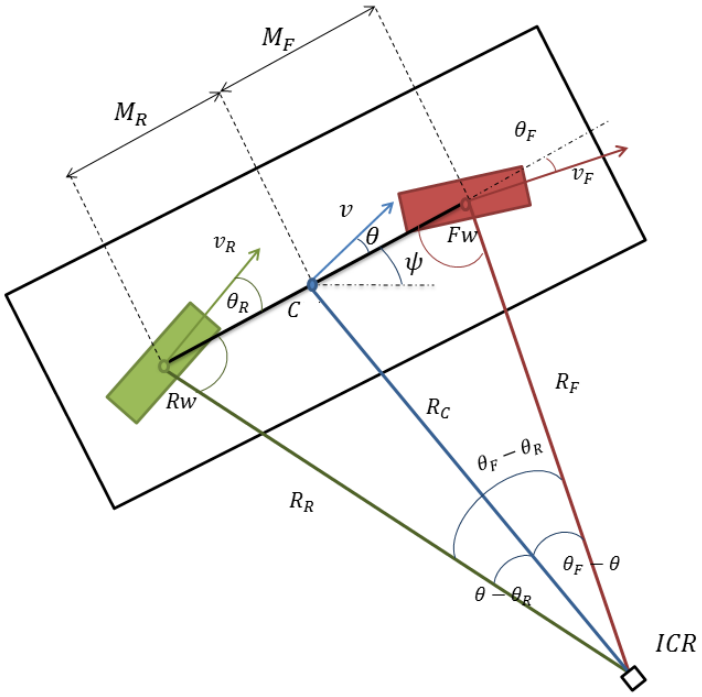

Figure 2.3: System model.

ICR notation

- ICR - Instantaneous center of rotation, a point with 2 dimensions (x, y)
- R_R - Distance between ICR and Rw
- R_F - Distance between ICR and Fw
- R_C - Distance between ICR and C

Angular radius

While moving in a circular trajectory, the velocity vector is tangent to the circumference of the moving body. One of the first steps is to encounter the circumference radius. Three radii were calculated, the radius of the circumference for the rear wheel, R_R, for the front wheel, R_F, and for the vehicle center, R_C. These lines are orthogonal to the respective velocity vectors.

Looking at Figure 2.3 the lines R_R and R_F intersect, this is where the vehicle is at that instant describing a circular movement, the ICR. Following are the steps taken to calculate the angular radii.

To obtain the radius R_R and R_F, the sine law was applied to the triangle $< Rw, Fw, ICR >$ resulting in:

$$\frac{\sin(\theta_F - \theta_R)}{M_R + M_F} = \frac{\cos(\theta_F)}{R_R} = \frac{\cos(\theta_R)}{R_F} \tag{2.2}$$

From equation (2.2), it can now be found the equations for R_R and R_F

$$\boxed{R_R = \frac{\cos(\theta_F).(M_R + M_F)}{\sin(\theta_F - \theta_R)}} \tag{2.3}$$

$$\boxed{R_F = \frac{\cos(\theta_R).(M_R + M_F)}{\sin(\theta_F - \theta_R)}} \tag{2.4}$$

To find the equation for R_C instead if the triangle $< Rw, Fw, ICR >$, two other triangles were considered. These are $< Rw, C, ICR >$ and $< Fw, C, ICR >$, by applying the sine law to each triangle, the equation for R_C is found.

Applying the sine law to $< Fw, C, ICR >$ results in:

$$\frac{\sin(\theta_F - \theta)}{M_F} = \frac{\sin(\frac{\pi}{2} - \theta_F)}{R_C} \tag{2.5}$$

and to $< Rw, C, ICR >$:

$$\frac{\sin(\theta - \theta_R)}{M_R} = \frac{\sin(\frac{\pi}{2} + \theta_R)}{R_C} \tag{2.6}$$

Multiplying both sides of equation (2.5) by $\frac{M_R}{\cos\theta_R}$ and both sides of equation (2.6) by $\frac{M_F}{\cos\theta_F}$ results in:

$$\tan\theta_F.\cos\theta - \sin\theta = \frac{M_F}{R_C} \tag{2.7}$$

$$\sin\theta - \tan\theta_R.\cos\theta = \frac{M_R}{R_C} \tag{2.8}$$

Finally adding equation (2.7) with equation (2.8), the equation for R_C is obtained.

$$\boxed{R_C = \frac{(M_F + M_R)}{\cos\theta(\tan\theta_F - \tan\theta_R)}} \tag{2.9}$$

Angular velocities

As stated in Section 2.1.2 the angular velocity for each wheel and the vehicle center is the same. In the general case of uniform circular motion is given by:

$$\dot{\psi} = \frac{v}{R_C} \tag{2.10}$$

Equation (2.10) also applies to each wheel, with equations (2.4), (2.3) and the velocities v_F and v_R, the following can be obtained:

$$\boxed{\dot{\psi}_R = \frac{v_R.\sin(\theta_F - \theta_R)}{(M_F + M_R).\cos\theta_F}} \tag{2.11}$$

$$\boxed{\dot{\psi}_F = \frac{v_F.\sin(\theta_F - \theta_R)}{(M_F + M_R).\cos\theta_R}} \tag{2.12}$$

$$\boxed{\dot{\psi} = \frac{v.\cos\theta.(\tan\theta_F - \tan\theta_R)}{(M_F + M_R)}} \tag{2.13}$$

As stated in section 2.1.2 there are some assumptions that restrict each wheel movement space, since they are physically connected, both wheels and its center have the same angular velocity, and the

wheels can not drift away from the vehicle body, so:

$$\dot{\psi}_R = \dot{\psi}_F = \dot{\psi} \tag{2.14}$$

Because of these restrictions there are three equations for the angular velocity of the vehicle frame $\dot{\psi}$.

Linear velocity vector: Orientation (θ) and value (v)

In order to control the vehicle at its center, the linear velocity vector is also needed, it is composed by its value v and angle θ.

As stated in section (2.1.2) this vector is not subject to forces that would cause the vehicle to slip, so the angle θ is obtained directly from geometry, it depends only on the orientation of the two wheels.

The equations that allow the discovery of R_C, equations (2.7) and (2.8) when multiplied respectively, by M_R and M_F, and subtracted, allow to discover the equation for θ

$$\boxed{\theta = \arctan(\frac{M_R.\tan\theta_F + M_F.\tan\theta_R}{M_F + M_R})} \tag{2.15}$$

It can be seen in equation (2.15), that θ only depends of the orientation of the wheels. The velocity vector value can be obtained due to the restrictions shown in equation (2.14), that states that the angular velocities are the same. By rearrangement of equation (2.13) with equation (2.12), and equation (2.13) with equation (2.12), two equivalent equations for the velocity value appear.

$$v = \frac{v_R}{\cos\theta_F}\frac{\sin(\theta_F - \theta_R)}{\cos\theta.(\tan\theta_F - \tan\theta_R)} \tag{2.16}$$

$$v = \frac{v_F}{\cos\theta_R}\frac{\sin(\theta_F - \theta_R)}{\cos\theta.(\tan\theta_F - \tan\theta_R)} \tag{2.17}$$

Which can be simplified into:

$$v = \frac{v_F.\cos\theta_F}{\cos\theta} \tag{2.18}$$

$$v = \frac{v_R.\cos\theta_R}{\cos\theta} \tag{2.19}$$

The duality of the previous two equations is shown if equation (2.14) is applied to each one.

Replacing θ in equations (2.18) and (2.19) by equation (2.15):

$$\boxed{v = v_R.\cos\theta_R\sqrt{\Big(\frac{M_R.\tan\theta_F + M_F.\tan\theta_R}{M_F + M_R}\Big)^2 + 1}} \tag{2.20}$$

$$\boxed{v = v_F.\cos\theta_F\sqrt{\Big(\frac{M_R.\tan\theta_F + M_F.\tan\theta_R}{M_F + M_R}\Big)^2 + 1}} \tag{2.21}$$

In conclusion the equations obtained for the control variables, have two options that resulted from the

duality found in v and $\dot{\psi}$, which does not appear on θ because it does not depend of the wheel velocities.

Following are the equations that are used to build the kinematic model:

Angular velocity duality $\dot{\psi}$:

$$\boxed{\dot{\psi} = \frac{v_R}{\cos\theta_F}\frac{\sin(\theta_F - \theta_R)}{M_R + M_F}} \tag{2.22}$$

$$\boxed{\dot{\psi} = \frac{v_F}{\cos\theta_R}\frac{\sin(\theta_F - \theta_R)}{M_R + M_F}} \tag{2.23}$$

Linear velocity value duality v:

$$\boxed{v = \frac{v_F.\cos\theta_F}{\cos\theta}} \tag{2.24}$$

$$\boxed{v = \frac{v_R.\cos\theta_R}{\cos\theta}} \tag{2.25}$$

Linear velocity orientation θ

$$\boxed{\theta = \arctan(\frac{M_R.\tan\theta_F + M_F.\tan\theta_R}{M_F + M_R})} \tag{2.26}$$

Another way to represent the vehicle kinematic equations is the following:

$$\dot{x} = v.\cos(\psi + \theta) \tag{2.27}$$

$$\dot{y} = v.\sin(\psi + \theta) \tag{2.28}$$

$$\dot{\psi} = \frac{v.\cos\theta.(\tan\theta_F - \tan\theta_R)}{M_R + M_F} \tag{2.29}$$

Given the vehicle kinematic configuration, there are some wheel configurations that can make the vehicle side-slip. These cases happen when the vehicle wheels make 180 degrees between each other, and when the wheels are orthogonal, the solutions shown in this thesis allow these configurations. In the first case the vehicle shall remain in the same place without changing its heading, the velocity vectors cancel each other. In the second case the vehicle moves as the lateral steering force that the orthogonal wheel will exert is not considered.

In this thesis the vehicle wheels are not constrained in their rotation, which means they can rotate endlessly without limit. Each wheel has two possible combinations. A vector with an angle between 0 to 90 degrees and a positive value for the velocity value, or the angle flipped 180 degrees with a negative velocity value. When the vehicle wheel is constrained to have the maximum angle between -90 to 90 degrees and the target angle exceeds this value, the wheel needs to rotate 180 degrees and the velocity value needs to change signal. To do this the vehicle velocities can not be restricted to be either positive or negative.

If the velocity values need to be always positive the vehicle wheels need to rotate 360 degrees in order to keep the vehicle omni-directional.

2.2 Vehicle center VS Independent wheel driving

The main difference between driving the vehicle at is center or at its wheels are the number of variables being managed by the operator, outlined in Figure 2.4. While driving the vehicle focusing in each wheel the operator needs to take into account at least 4 variables, the rear and front wheel orientations and velocities, and has two control points, the center of each wheel. Driving at the vehicle center, the operator needs to manage three variables, the velocity and orientation of the vehicle center and its heading, in one control point, the center of the vehicle. In both situations the operator must also be aware of the volume occupied by the vehicle body and its position relative to the scenario.

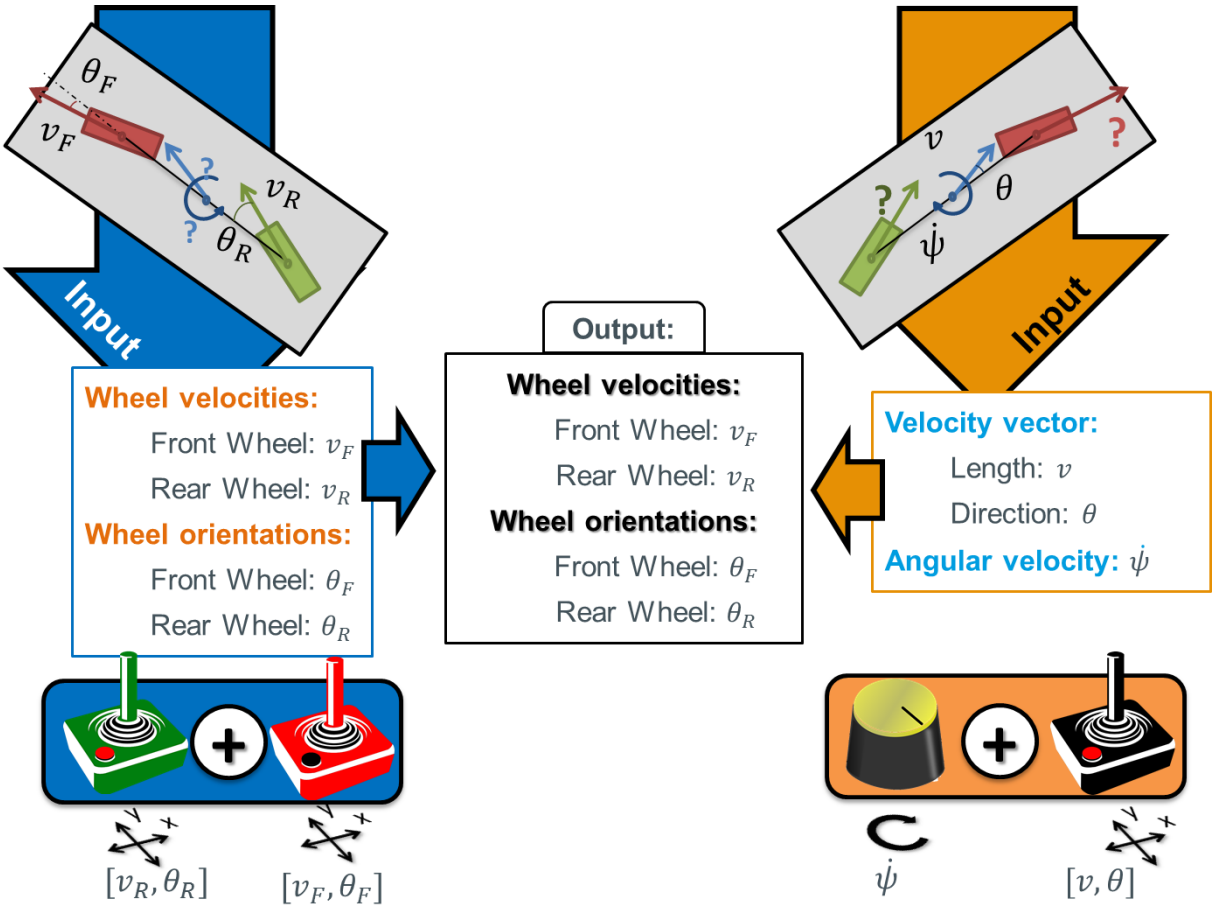

Figure 2.4: Differences between control methods

There are advantages and disadvantages in both methods, the main advantage that driving the vehicle at each wheel has, is that the number of inputs is the same as the outputs. Meaning that the vehicle motion capabilities remain intact. By driving the vehicle at its center, it is needed to transform three inputs into four outputs. By driving the vehicle at its center the operator has less variables to manage, and more important, can be more focused on the vehicle itself. This should make the vehicle easier to drive, in the operator perspective. Ultimately the ease of driving the vehicle focusing in its center can outweigh the movement capabilities of independent wheel driving.

2.3 Independent wheel driving device

The vehicle has two control points, the front and the rear wheels, both driving and steerable. So the direct approach is to control the vehicle directly at each wheel. This type of control happens when a car is driven. The pedals change the velocity and the steering wheel changes the front wheel orientations. Using the same approach with a rhombic-like vehicle means that in order to control it, a set of two pedals plus two steering wheels are needed, instead of this strange mechanism, two joysticks will be used in the form of a gamepad.

This device was chosen because it uses a mature technology and was available to be used at the start of this thesis.

With this device the velocity value is always positive and the wheels are free to rotate endlessly.

2.3.1 Gamepad

Using a gamepad is the solution proposed by the author of [21]. That method of driving the vehicle imposed limitations to the vehicle motion freedom. It limits the wheel orientations from 0 to 180 degrees and the difference between the wheel angles can not be greater that 90 degrees. The vehicle continues to be omni-directional, however the restrictions need to be overcome by the operator driving the vehicle. The velocity is also being incrementally increased and decreased, also the variables do not return to zero by default, this can cause collisions when driving the vehicle.

This approach limits the operator when driving the vehicle, so a new approach for the independent wheel driving device was thought, where it is removed the angle restrictions allowing the vehicle to rotate its wheels freely in any orientation, and the method to increase and decrease the velocities also changes. Both velocity orientations and values are being commanded only by the joysticks.

2.4 Vehicle center driving device

The goal is to drive the vehicle as shown in the right image of Figure 2.1, being able to move the vehicle from one place to another keeping its heading (crab-like motion) as shown on the left side of the image, rotating in the same place only changing its heading, shown on the center of the image, and lastly, the combination of the two, changing its heading while describing a trajectory with its center, as depicted in the right side of the image.

With this device, the velocity that is sent to drive the vehicle has always a positive value and has 360 degrees of freedom. Like with the independent wheel driving device, the wheel velocities are supposed to be positive and can rotate endlessly.

2.4.1 Joystick

The goal with the analog joystick is to send commands to the vehicle center velocity vector. Its value and orientation, which alone, allows for crab like movements where the vehicle heading does not change.

Because it is an analog joystick it has the required sensitivity to drive a vehicle.

This device was chosen because of its maturity and industrial applications in the remote control of heavy transporters and other types of machines, and because it was available to work with it at the start of the thesis.

2.4.2 Rotational disc

The rotational Disc is basically a turning device, these are used everywhere from the microwave timer, to volume control, their function is to set a given parameter, in the particular case of this thesis is to change the angular velocity of the center of the vehicle, therefore commanding its heading.

Figure 2.5: DJ jog wheel

Conceptually this rotational disc should look like a DJ jog wheel that can be seen in Figure 2.5 without return to zero position feature.

This device was chosen because it could be made, and the electronic components to build it were available to work with.

2.4.3 3D Concept model for a control desk

A 3D model for a control desk was developed using Solidworks. The design was inspired by a company that is specialized in control room furniture, Lund Halsey [27], one of their control rooms can be seen in Figure 2.6.

The devices that compose the system are seen in Figure 2.7 and are divided in three categories. Displaying elements denoted by the green balloons, core elements denoted by the blue balloons, and functional elements denoted by the green balloons.

The displaying elements are used to display information, either graphical or written. The core elements are essential to drive the vehicle. Finally the functional elements are optional or ad-ons.

Following are the components that take part in the the device assembly:

Figure 2.6: A Lund Halsey control room furniture installation

Figure 2.7: Concept control stand I

1. **Integrated panel PC** - The panel PC is used to run the software that will control the vehicle. It is where the manual and automatic control algorithms are computed. Its monitor is supposed to be the focus point while driving the vehicle, there will be displayed the reconstructed top view of the environment either by video camera feed or by virtual reality.

2. **2-axis analog Joystick** - The Joystick is used to send commands for the vehicle center velocity. Because it has two axis and they are analog it is possible to obtain a value and an angle, which will

constitute the vehicle center velocity vector, given by its value v and its orientation θ. The Joystick should also be equipped with a deadman switch to prevent unexpected operator errors.

3. **Rotational Disc** - The rotational disc is an endless rotation device. It is used to command the angular velocity of the vehicle center. Therefore controlling its heading. This device and the joystick are the driving devices.

4. **Vertical display** - This display monitor mainly shows information regarding the orientation status of the vehicle and its wheels, it shows the vehicle heading and the direction of the movement as well as the visual representation of the cask heading and wheel orientations.

5. **Large Horizontal Display** - This large monitor is where all the information regarding navigation and localization system will be displayed, and other relevant information regarding the vehicle or reactor status. It will also be used as a supervising device because of all the information it displays.

6. **Emergency STOP button** - This allows the operator to stop all the activities of the navigation system. Note that stopping the vehicle suddenly can cause damage to the vehicle or the surrounding area or introduce error in the vehicle localization, so the button should only be pressed in case of emergency.

7. **Computer mouse** - This is a regular computer mouse, is plugged to the panel PC and is used to interact with the graphical user interface.

8. **Computer keyboard** - This is a regular computer keyboard, used as an alpha-numerical input in the computer.

9. **Telephone** - A telephone to be used in internal communications.

In Figure 2.8 it is given more emphasis to the driving devices, the Joystick and the rotational disc, as well as the viewing system.

2.5 Methods to control the vehicle at its center

The vehicle has two wheels, both drivable and steerable, and has four input variables, v_F, v_R, θ_F and θ_R which result in three output variables v, θ and $\dot{\psi}$. The goal, in order to drive the vehicle at its center, is to invert the system and use its outputs as inputs, however this task is non trivial due to the non-linearity of the system. Following are shown three methods that aim to solve this problem.

2.5.1 Motion decomposition in simple movements

The motion of the vehicle can be split in three types of motion, pure translation, pure rotation and a mixture of both. In the case of a pure translation, the vehicle simply moves from one point to another without changing its heading. In a pure rotation the vehicle rotates around itself while keeping its center in the same position. In the last case, the vehicle is changing its heading while changing its position.

Figure 2.8: Concept control stand II

In the first situation the commands sent to both wheels are directly taken from the Joystick. In this situation the vehicle does not change its heading, so both wheels have the same direction and are equal to θ, the same happens with the wheel velocities, both are equal and have the same value as v.

In the second situation, both wheels are orthogonal to the vehicle heading and the angular velocity, $\dot{\psi}$, is taken from the the rotational device.

The proposed solution for the third type of movement is an averaged sum of the previous two components. The commands sent to the wheels by pure translation are $\theta_{TF}, \theta_{TR}, v_{TF}, v_{TR}$, and by pure rotation, $\theta_{RF}, \theta_{RR}, v_{RF}, v_{RR}$ so the following is obtained:

$$\theta_{TF} = \theta_{TR} = \theta \tag{2.30}$$

$$v_{TF} = v_{TF} = v \tag{2.31}$$

$$\theta_{RF} = \theta_{RR} = \frac{\pi}{2} \tag{2.32}$$

$$v_{RF} = -v_{RR} = \dot{\psi} \tag{2.33}$$

Finally the sum of the contribution of both components :

$$\boxed{\theta_F = \frac{a.\theta_{TF} + b.\theta_{RF}}{a+b}} \tag{2.34}$$

$$\boxed{\theta_R = \frac{a.\theta_{TR} + b.\theta_{RR}}{a+b}} \tag{2.35}$$

$$\boxed{v_F = \frac{a.v_{TF} + b.v_{RF}}{a+b}} \tag{2.36}$$

$$\boxed{v_R = \frac{a.v_{TR} + b.v_{RR}}{a+b}} \tag{2.37}$$

The author of this thesis proposes to use a and b which are constants to tune the weight of each component. The necessity of these constants lie in the need of tuning what component weights more in order to achieve a better driving device.

2.5.2 Inverse kinematics using an approximation of the system

Other solution to solve the control problem is to approximate the system using a first order Taylor series [28], and inverting the resulting system using the Pseudo-Inverse [29].

The goal is to get v_F, v_R, θ_F and θ_R from the input variables v, θ, and $\dot{\psi}$.

The Taylor series is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point. A first order approximation was used, only the first derivatives were computed.

This approach results in an iterative algorithm that at each new position, computes the vehicle wheels variables taking into account the previous values.

Where the function $g(t) = f(u(t))$ is a system of functions that defines the model and the first order Taylor approximation of the system is given by equation (2.38).

$$g \approx f(u_0) + \frac{\partial f(u_0)}{\partial u}(u - u_0) \tag{2.38}$$

The Algorithm

The system will be modeled as a function where $g(t) = f(u(t))$ and the goal is to obtain $u(t) = f^{-1}(g(t))$ and at each motion update, the following algorithm is computed in order to obtain the values for v_F, v_R, θ_F and θ_R:

1. $f(u(t)) = \begin{bmatrix} v(t) \\ \dot{\psi}(t) \\ \theta(t) \end{bmatrix}$
2. The system is approximated using the Taylor series expansion was used:

   $g \approx f(u_0) + \frac{\partial f(u_0)}{\partial u}(u - u_0)$
3. In order to get the new values for u it is needed to compute the pseudo-inverse of the system matrix:

   $u = u_0 + Pinv(\frac{\partial f(u_0)}{\partial u}(g - f(u_0))$
4. Making $J = \frac{\partial f(u_0)}{\partial u}$ the system matrix, the pseudo-inverse (Pinv) is given by:

$$Pinv(J) = (J^T J)^{-1} J^T$$

5. The variable that is desired to be manually controlled by the operator is the vehicle's heading, the system does this by taking into account its time derivative, so the following approximation is needed:

   $$\psi(k) = T\dot{\psi(k)} + \psi(k-1)$$

   where T is the sampling time.

6. At last the following is obtained:

$$\begin{bmatrix} v_F(k) \\ v_R(k) \\ \theta_F(k) \\ \theta_R(k) \end{bmatrix} = \begin{bmatrix} v_F(0) \\ v_R(0) \\ \theta_F(0) \\ \theta_R(0) \end{bmatrix} + Pinv.J. \begin{bmatrix} v_F(0) \\ v_R(0) \\ \theta_F(0) \\ \theta_R(0) \end{bmatrix} . \begin{bmatrix} \dot{\psi}(k) - \dot{\psi}(0) \\ v(k) - v(0) \\ \theta(k) - \theta(0) \end{bmatrix}$$

These steps will be calculated periodically (T), giving an approximation to the vehicle wheel velocities and orientations.

$$J = \begin{bmatrix} \frac{\partial v}{\partial v_F} & \frac{\partial v}{\partial v_R} & \frac{\partial v}{\partial \theta_F} & \frac{\partial v}{\partial \theta_R} \\ \frac{\partial \dot{\psi}}{\partial v_F} & \frac{\partial \dot{\psi}}{\partial v_R} & \frac{\partial \dot{\psi}}{\partial \theta_F} & \frac{\partial \dot{\psi}}{\partial \theta_R} \\ \frac{\partial \theta}{\partial v_F} & \frac{\partial \theta}{\partial v_R} & \frac{\partial \theta}{\partial \theta_F} & \frac{\partial \theta}{\partial \theta_R} \end{bmatrix} \tag{2.39}$$

To simplify the equations let Z be equal to:

$$Z = \left(\frac{M_R . \tan\theta_F + M_F . \tan\theta_R}{M_R + M_F} \right)^2 + 1 \tag{2.40}$$

The equations that compose the Jacobian matrix, seen in equation (2.39) are the following:

The θ derivatives are the following:

$$\frac{\partial \theta}{\partial v_F} = 0 \tag{2.41}$$

$$\frac{\partial \theta}{\partial v_R} = 0 \tag{2.42}$$

$$\frac{\partial \theta}{\partial \theta_F} = \frac{M_R . \sec^2\theta_F}{(M_F + M_R).Z} \tag{2.43}$$

$$\frac{\partial \theta}{\partial \theta_R} = \frac{M_F . \sec^2\theta_R}{(M_F + M_R).Z} \tag{2.44}$$

With $\dot{\psi}$ equal to equation (2.22) its derivatives are:

$$\frac{\partial \dot{\psi}}{\partial \theta_F} = \frac{v_R}{M_R + M_F} . \sec^2\theta_F . \cos\theta_R \tag{2.45}$$

$$\frac{\partial \dot{\psi}}{\partial \theta_R} = -\frac{v_R}{M_R + M_F} \frac{\cos(\theta_F - \theta_R)}{\cos\theta_F} \tag{2.46}$$

$$\frac{\partial \dot{\psi}}{\partial v_F} = 0 \tag{2.47}$$

$$\frac{\partial \dot{\psi}}{\partial v_R} = \frac{1}{M_R + M_F} \frac{\sin(\theta_F - \theta_R)}{\cos \theta_F} \tag{2.48}$$

With $\dot{\psi}$ equal to equation (2.23) its derivatives are:

$$\frac{\partial \dot{\psi}}{\partial \theta_F} = \frac{v_F}{M_R + M_F} \frac{\cos(\theta_F - \theta_R)}{\cos \theta_R} \tag{2.49}$$

$$\frac{\partial \dot{\psi}}{\partial \theta_R} = -\frac{v_F}{M_R + M_F} . \sec^2 \theta_R . \cos \theta_F \tag{2.50}$$

$$\frac{\partial \dot{\psi}}{\partial v_F} = \frac{1}{M_R + M_F} \frac{\sin(\theta_F - \theta_R)}{\cos \theta_R} \tag{2.51}$$

$$\frac{\partial \dot{\psi}}{\partial v_R} = 0 \tag{2.52}$$

With v equal to equation (2.25) its derivatives are:

$$\frac{\partial v}{\partial \theta_F} = \frac{v_R . M_R . \sec^2 \theta_F . \cos \theta_R . (M_F . \tan(\theta_R + M_R . \tan \theta_F))}{(M_R + M_F)^2 . \sqrt{Z}} \tag{2.53}$$

$$\frac{\partial v}{\partial \theta_R} = v_R \Big(\frac{M_F . \sec \theta_R . (M_F . \tan \theta_R + M_R . \tan \theta_F)}{(M_R + M_F)^2 . \sqrt{Z}} - \sin \theta_R . \sqrt{Z} \Big) \tag{2.54}$$

$$\frac{\partial v}{\partial v_F} = 0 \tag{2.55}$$

$$\frac{\partial v}{\partial v_R} = \cos \theta_R \sqrt{Z} \tag{2.56}$$

With v equal to equation (2.24), its derivatives are:

$$\frac{\partial v}{\partial \theta_F} = v_F . \Big(\frac{M_F . \sec \theta_F . (M_F . \tan \theta_R + M_R . \tan \theta_F)}{(M_R + M_F)^2 . \sqrt{Z}} - \sin \theta_R . \sqrt{Z} \Big) \tag{2.57}$$

$$\frac{\partial v}{\partial \theta_R} = \frac{v_F . M_R \sec^2 \theta_R . \cos \theta_F . (M_F . \tan(\theta_R + M_R . \tan \theta_F))}{(M_R + M_F)^2 . \sqrt{Z}} \tag{2.58}$$

$$\frac{\partial v}{\partial v_F} = \cos \theta_F . \sqrt{Z} \tag{2.59}$$

$$\frac{\partial v}{\partial v_R} = 0 \tag{2.60}$$

Because of equation (2.14) the duality is also present in the derivatives of $\dot{\psi}$ and v this results in two sets of four derivatives for each variable, in both sets there is a derivative that is equal to zero. The elements that will compose the Jacobian matrix are chosen in such a way that all elements of the matrix are non-zero so. Making:

$$f(v_F, \theta_F) = v_F . \cos \theta_F \tag{2.61}$$

and

$$f(v_R, \theta_R) = v_R . \cos \theta_R \tag{2.62}$$

It is known, due to vehicle integrity restrictions that $f(v_F, \theta_F) = f(v_R, \theta_R)$, which means that :

$$\frac{\partial v}{\partial \theta_F} = \frac{\partial v}{\partial \theta_F} \tag{2.63}$$

and:

$$\frac{\partial \dot{\psi}}{\partial \theta_F} = \frac{\partial \dot{\psi}}{\partial \theta_F} \tag{2.64}$$

2.5.3 Alonzo Kelly forward rate kinematics

A method developed by Alonzo Kelly to obtain the actuated inverse solution for the system was also studied, which is presented in [30]. This method is a vector algebra formulation for a vehicle kinematics and also relies on the instantaneous center of rotation.

The model that was adopted for a vehicle with a rhombic-like configuration was the generalized bicycle method, the resultant system can be seen in equation (2.65) using as model the one presented in Figure 2.9

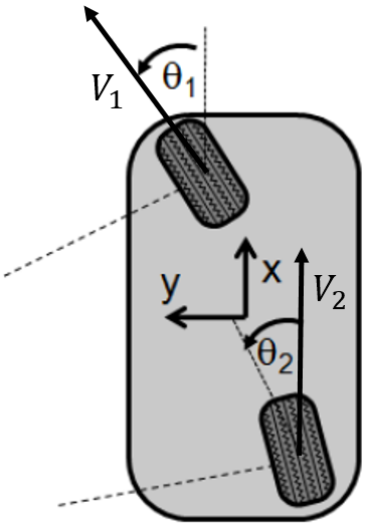

Figure 2.9: Model used by Alonzo Kelly in his formulation

$$\begin{bmatrix} v_{1x} \\ v_{1y} \\ v_{2x} \\ v_{2y} \end{bmatrix} = \begin{bmatrix} V_x - \omega.y_1 \\ V_y + \omega.x_1 \\ V_x - \omega.y_2 \\ V_y + \omega.x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -y_1 \\ 0 & 1 & x_1 \\ 1 & 0 & -y_2 \\ 0 & 1 & x_2 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \tag{2.65}$$

In equation (2.65) (x_1, y_1) and (x_2, y_2) are the positions of both wheels, looking at Figure 2.9 wheel 1 is the top and wheel 2 is at the bottom, ω is the angular velocity of the vehicle, V_x and V_y are the components of the linear velocity vector.

When this method is applied to the rhombic-like vehicle being studied the following is obtained:

With the vehicle velocity vector equal to:

$$v = \begin{bmatrix} v_x \\ \\ v_y \end{bmatrix} \tag{2.66}$$

Where v_x is the vector horizontal projection given by:

$$v_x = v\cos(\theta) \tag{2.67}$$

And v_y is the vector vertical projection given by:

$$v_y = v\sin(\theta) \tag{2.68}$$

With these two results the velocity vectors that will be sent to each wheel are found, for the front wheel:

$$\boxed{v_F = \begin{bmatrix} v_x & v_y - \omega.M_F \end{bmatrix}} \tag{2.69}$$

And for the rear wheel:

$$\boxed{v_R = \begin{bmatrix} v_x & v_y + \omega.M_R \end{bmatrix}} \tag{2.70}$$

With a vector for each wheel the velocity value can be found by taking the norm of each vector and the orientation of the wheels is given by:

$$\boxed{\theta_F = \tan^{-1}\left(\frac{v_y - \omega.M_F}{v_x}\right)} \tag{2.71}$$

$$\boxed{\theta_R = \tan^{-1}\left(\frac{v_y + \omega.M_R}{v_x}\right)} \tag{2.72}$$

2.6 Comparison between methods to control the vehicle at its center

Following is shown a comparison between the methods presented in last Section. By splitting the motion in pure translation and rotation (see Section 2.5.1), the pseudo-inverse method (see Section 2.5.2) and the method developed by Alonzo Kelly (see Section 2.5.3). For simplicity and from this point on these methods are identified as Split, Pseudo-inverse and A. Kelly respectively.

The goal is to evaluate the response of each method to a set of fixed inputs and obtain their response. The method with better response/results will be used in the final interface.

2.6.1 Four sets of data to simulate an operator driving

The following results show the behaviour of each method when subjected to various sets of input variables. The linear velocity value v and orientation θ, and the angular velocity $\dot{\psi}$. Following is the description of each data set used as input variables:

- **First set:**
  - v: Logarithmically rising until it saturates in 0.2 m/s.
  - θ: Linear increment from 20 to 40 deg.
  - $\dot{\psi}$: Constant value of 0 rad/s.
- **Second set:**
  - v: Logarithmically rising until it saturates in 0.2 m/s.
  - θ: Constant value of 20 deg.
  - $\dot{\psi}$: Linear increment from 0 to 0.1 rad/s
- **Third set:**
  - v: Logarithmically rising until it saturates in 0.2 m/s.
  - θ: Constant value of 20 deg.
  - $\dot{\psi}$: Constant value of 0.1 rad/s.
- **Fourth set:**
  - v: Logarithmically rising until it saturates in 0.2 m/s.
  - θ: Linear increment from 20 to 80 deg.
  - $\dot{\psi}$: Constant value of 0.1 rad/s.

Each set of variables has one hundred samples, which regarding simulation time correspond to a duration of two seconds. The first set represents a crab-like movements where only the linear velocity orientation and value are changing. The second set is a combination of crab-like motion with heading changes, the third and fourth sets are used to compare how the methods behave when the vehicle is changing its heading ψ and the linear velocity orientation θ is fixed (third set), against when both θ and ψ are changing (forth set).

The three methods were tested using Matlab 2012b running the respective system for each method.

2.6.2 Results for the first set

This set is used mainly to evaluate the behaviour of each method when performing crab like movements, where only the linear velocity vector changes, variables θ and v. In this set of values, the angular velocity is zero, so the vehicle's heading ψ remains unchanged and the wheels should have the same orientation has the variation of θ.

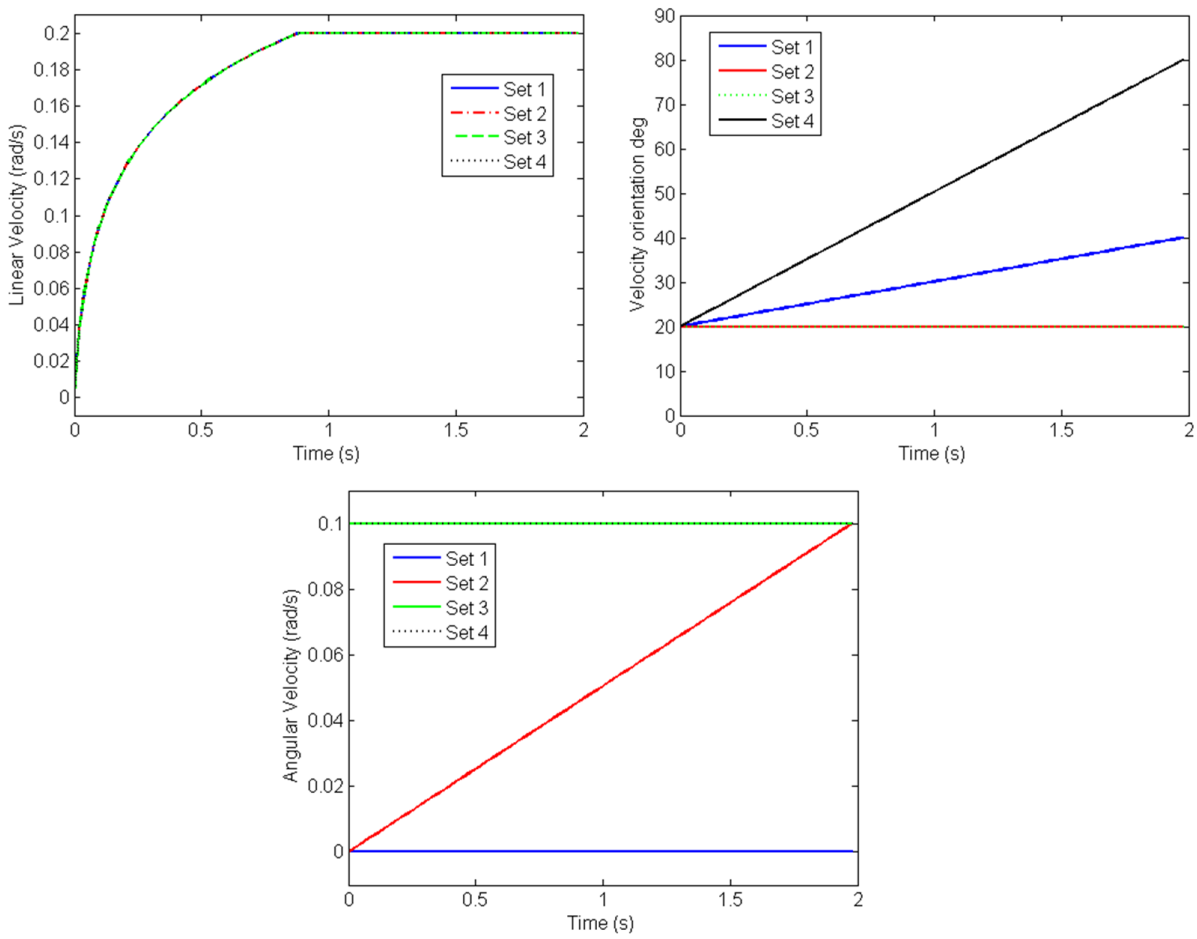

Figure 2.10: Input variables for all sets

In Figure 2.11 on the top are shown the results for front and rear wheels' velocity value v_F and v_R. In this and on the following Figures, left images are always relevant to the front wheel and the right referent to the rear wheel. It can be seen that the velocities for A. Kelly and Split methods are coincident, whereas the Pseudo Inverse's results present a velocity value that is slightly lower than the values obtained with the other two methods. In all situations the velocity has an expected evolution.

The previous assumption regarding the wheels' orientation can be seen in the bottom images in Figure 2.11 which shows the front and rear wheels orientation evolution, all three graphs are superimposed. And is seen that both wheels increase linearly from 0 to 20 degrees.

This means that all three methods behave well when applied to crab-like movements where only the linear velocity vector is being changed.

2.6.3 Results for the second set

In Figure 2.12 is shown the wheel velocities progression in the top images. Since this time the angular velocity starts at zero and is incremented linearly along the time, none of the methods show spikes or superelevations at the beginning, all three methods have a smooth evolution. In this set regarding the rear and front wheel velocities, the Pseudo-inverse and A. Kelly methods show closer values.

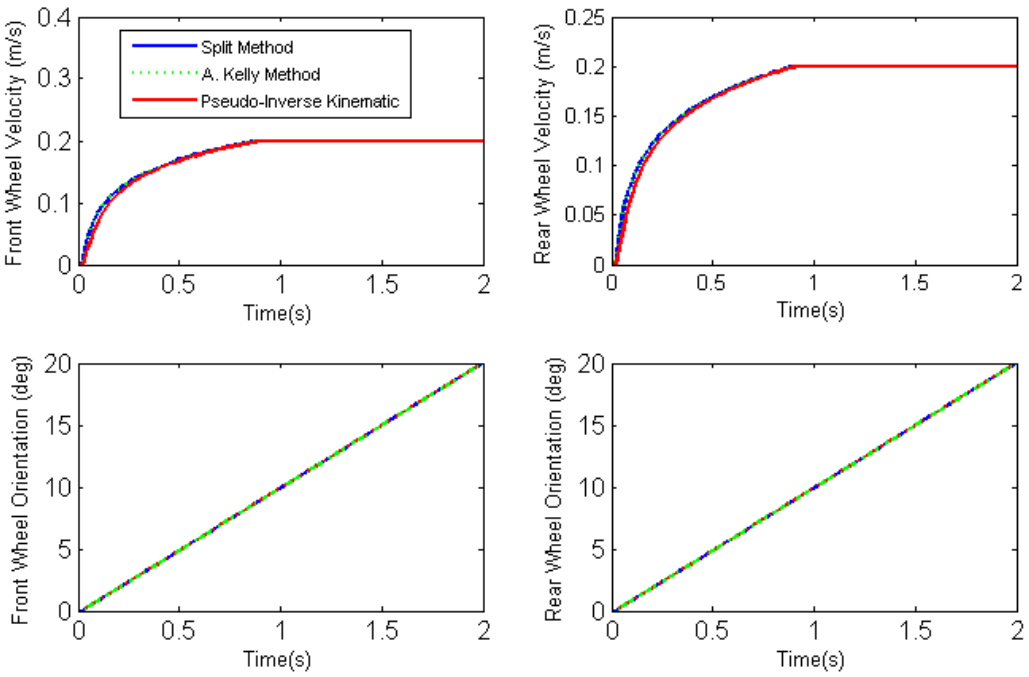

Figure 2.11: Wheel velocity and orientation evolution with the first set

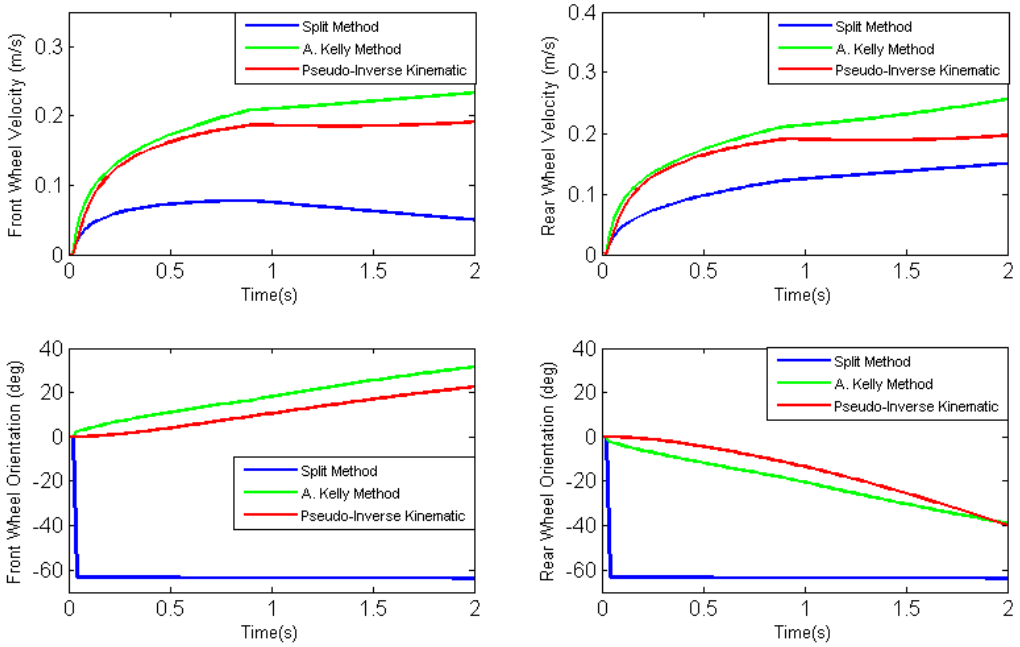

Figure 2.12: Wheel velocity and orientation evolution with the second set

In Figure 2.12 at the bottom is shown the wheel orientations of the vehicle. Split method changes orientations abruptly and quickly achieves its goal orientation, which means the heading change is mostly done by changes in the velocity values. The other two methods produce similar curves with A. Kelly being slightly faster.

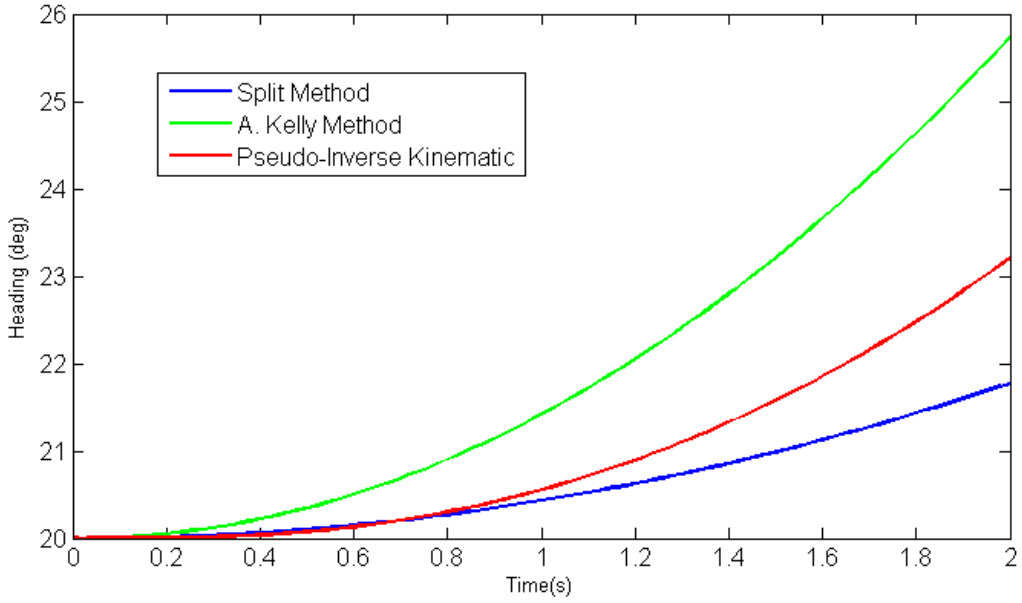

Figure 2.13: Vehicle heading evolution with the second set

Because the angular velocity is being incremented, all methods show that the vehicle heading has a second derivative that can be seen in Figure 2.13.

2.6.4 Results for the third set

In Figure 2.14 on top is shown the velocity evolution for the vehicle wheels. The curve shapes that were produced show that the Pseudo-inverse method is closer to A. Kelly. Split method has a little superelevation at the start, whereas the other two methods are smoother, and do not have that behaviour.

In Figure 2.14 on bottom is shown the orientation evolution of the vehicle wheels. It can be seen that the Pseudo-inverse and A. Kelly methods converge to the same values. Regarding the wheel orientations both A. Kelly and the Split methods show spikes on the orientation values in the test beginning, Split method is the fastest followed by A. Kelly.

It is expected that the vehicle heading is supposed to change linearly, in Figure 2.15 is shown its evolution for the three methods. Like in the wheel orientations case, the vehicle heading values with the Pseudo-inverse method are closer to the A. Kelly than the Split method. As can be seen the heading on all methods is linearly increasing, in the Split and A. Kelly methods it happens instantly, whereas in the Pseudo-inverse graph shows a transient behaviour until time around t=0.3 s, after that it behaves linearly as the other two methods.

2.6.5 Results for the fourth set

The Fourth set is similar to the third, however the linear velocity orientation θ is being incremented from 20 to 80 degrees, which should have no influence on the vehicle heading comparing to the third set. It is expected that the heading results obtained in this set should be the equal to the third set.

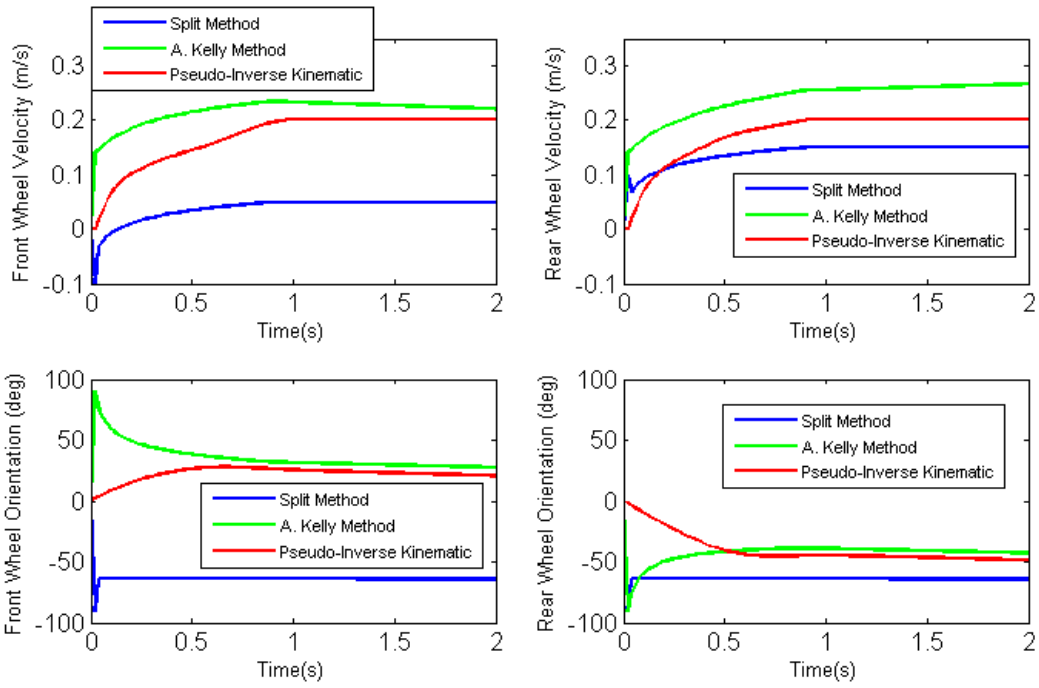

Figure 2.14: Wheel velocity and orientation evolution with the third set

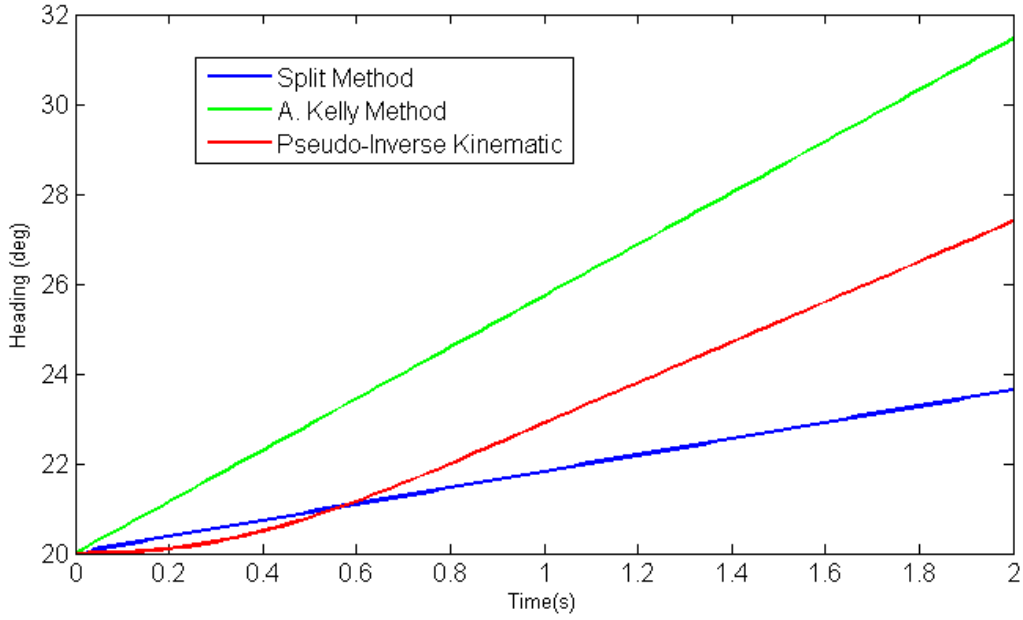

Figure 2.15: Vehicle heading evolution with the third set

By looking at Figure 2.16 A. Kelly and Pseudo-inverse have the same type of behaviour. The assumption that vehicle's heading should be equal to the results obtained with the third set is not true for the Pseudo-inverse and split methods, as can be seen in the bottom part Figure 2.16 and more detailed in Figure 2.18 where it is plotted the difference between heading values for each method in the third and fourth sets.

It can be seen in Figure 2.18 that in the end there is a difference of approximately four degrees

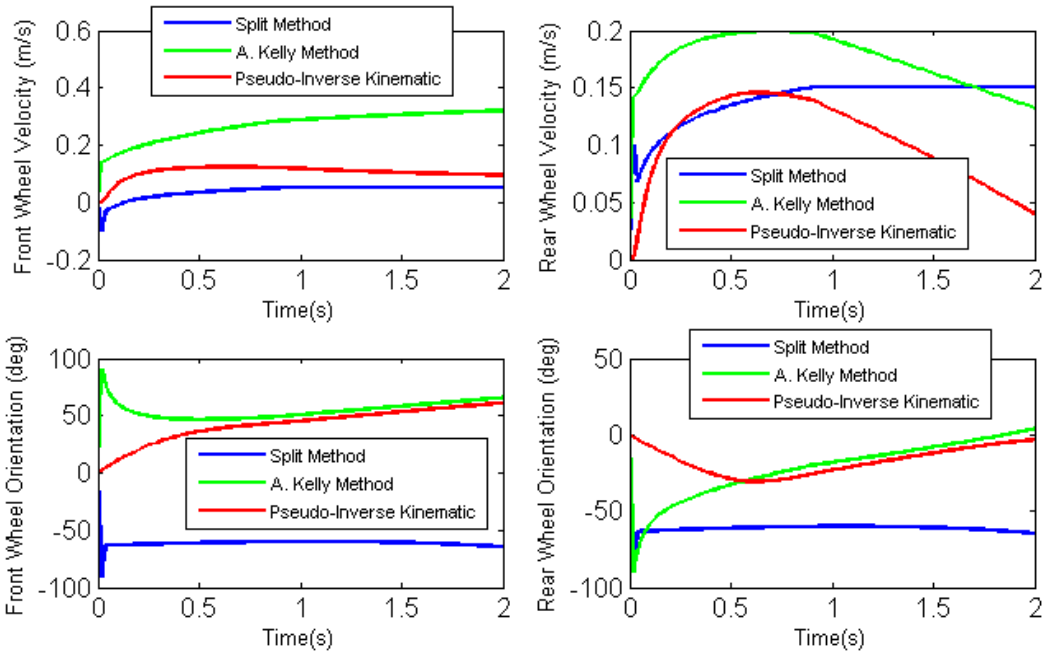

Figure 2.16: Wheel velocity and orientation evolution with the fourth set

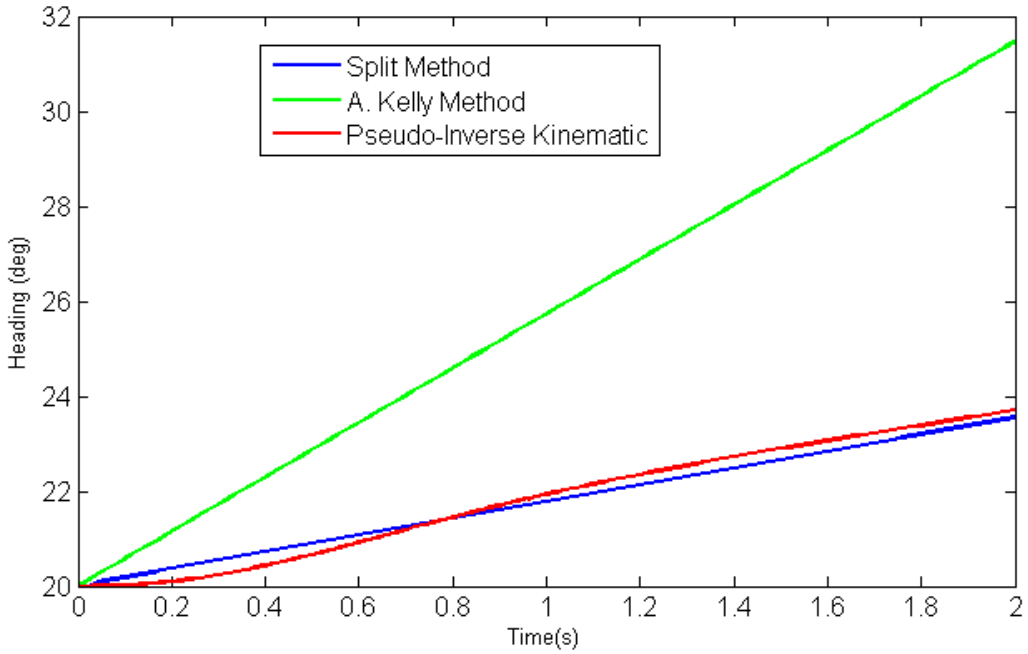

Figure 2.17: Vehicle heading evolution with the fourth set

between the results with third and fourth sets using the Pseudo-Inverse method. Using the A. Kelly method there are no differences. Using the Split method shows minimal differences. This means that changing the velocity vector orientation does not affect A. Kelly method, which is the only where heading is the same as in set 3.

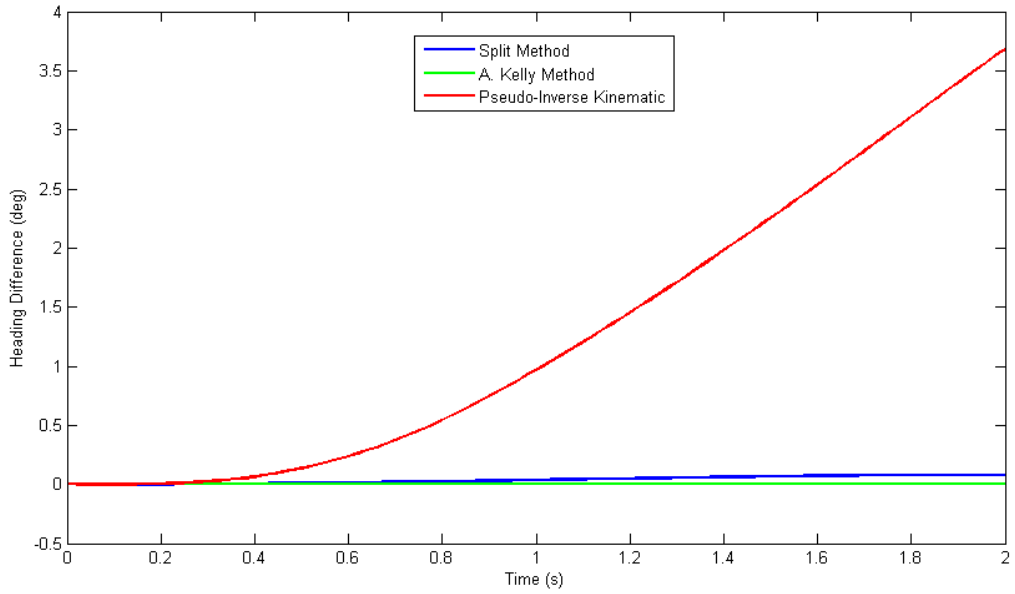

Figure 2.18: Heading difference between the third and fourth sets

2.6.6 Conclusions

With this type of study, it is not possible to find which is the best method to drive the vehicle, because part of that study needs to be tested in the devices with people. By selecting the method that has shown better results especially in the third and fourth tests, the Author of this thesis expects the chosen method to be the best while using the Joystick and Rotational Disc as driving device.

Resuming, the Split method has the faster response time, however the transitions are abrupt, which can cause stress in the vehicle motors. Usually fast transitions are destructive in the long term to mechanical devices

The Pseudo-inverse method has the slower response time with the smoother transitions which can be better perceived by the operator, unfortunately shows deviation in the heading values when the linear velocity orientation changes which is not the desired behaviour.

Lastly A. Kelly method, also has spikes and harsh transitions at the beginning of each set when $\dot{\psi}$ suddenly changes from 0 to a constant value, however when driving the vehicle, its angular velocity is not supposed to change abruptly form 0 to a value.This method shows no changes on the heading values when the linear velocity orientation is changed, which is the expected behaviour.

The most suited are the Pseudo-inverse and A. Kelly methods because of the way they evolve throughout time. Their evolution seems more ”natural”, in opposition with the Split method results that show abrupt changes especially on the wheel orientations. For the usability tests will be used the A. Kelly method, because overall it obtained better results, the heading difference shown in Figure 2.18 is the reason why A. Kelly method was chosen.

Chapter 3

Implementation

In this Chapter is shown the process of design and development of the physical devices, the simulator that was used for the interface and vehicle dynamics, the process of developing the rotational disc, what hardware and software was used.

The human machine interface has two main parts, the simulator that was used as visual interface and dynamic simulator, and the devices. A device to drive the vehicle focusing in its wheels and another focusing in its wheels, a gamepad is used for the first, and a joystick plus a custom built rotational disc for the second.

3.1 Trajectory Evaluator and Simulator (TES)

The Trajectory Evaluator and Simulator (TES) is a software that was developed at Instituto de Plasmas e Fusão Nuclear (IPFN) at Técnico Lisboa. Its main purpose is to generate optimal trajectories for a rhombic-like vehicle. They can have two types, free roaming or line guidance, in the first, each wheel can follow a different trajectory and in the second case both wheels follow the same trajectory. Besides this it also generates a report on the most risky points of collision and the swept volume of the vehicle along the mission. In Figure 3.1 it is shown the TES v2.0 interface showing a CTS trajectory from the lift to Port 11.

The software that was developed in this thesis was built to work as module for TES, since it lacks a good method to manually drive the vehicle.

The manual driving mode that was part of TES allowed the vehicle to be driven via keyboard or Gamepad. The keyboard as it is implemented is not suited to drive rhombic-like vehicles. To send commands to each wheel the operator needs to keep pressing buttons to increment the velocity and orientation of the wheels, this method compromises the vehicle and building safety. The gamepad has the same philosophy as the keyboard.

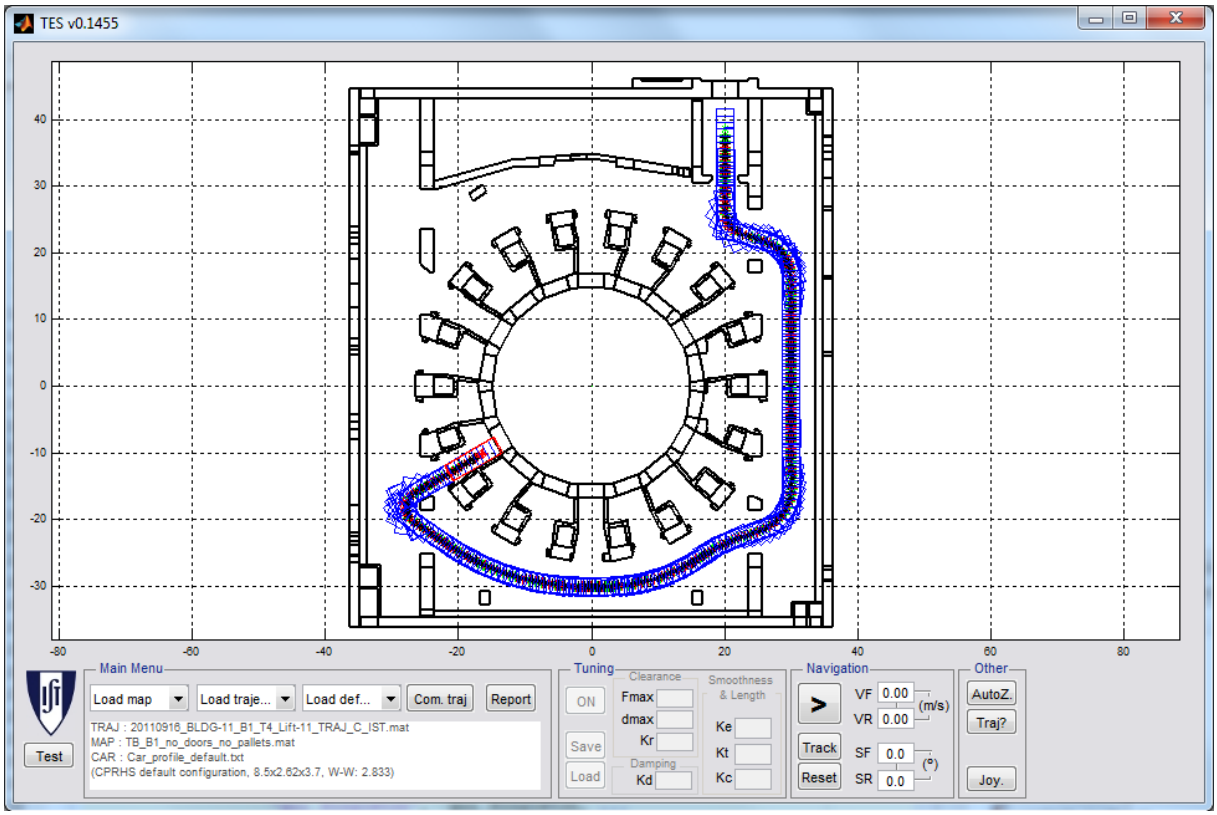

Figure 3.1: Trajectory evaluator and simulator

3.2 Independent wheel driving device

To drive the vehicle focusing on each wheel was used a NGS Maverick gamepad, that was plugged directly in the computer without need of any extra data acquisition. A Matlab function called ”vrjoystick” handles objects of the joystick type, this function returns the state and position of each button and joystick. It is being called within a Matlab timer once every 30 ms, after the gamepad mode button has been selected.

Two different methods were tested, one developed previously that was implemented on TES, lets call it Gamepad v.1 and other developed during the execution of this thesis, Gamepad v.2, their differences will be explained next.

Gamepad v.1 has two 2-axis joysticks which change the wheels orientation and four digital buttons to increase and decrease the wheels velocity, besides this has a reset button that puts the wheels parallel with the vehicle heading and a stop button. It was developed to prevent situations where wheel slippage could happen, meaning that the orientations that each wheel can have is limited, it is impossible to spin the wheels more than 180 degrees, and place the wheels in a position in which the velocity vectors are orthogonal. Other important fact is that the velocity is set to a given value, it does not return to zero, meaning that when not actuated the vehicle remains with the same velocity, which can be dangerous. If something happens to the user while driving it is impossible to prevent a collision. The operator must press a button to make the vehicle stop, the cross button of the gamepad that can be seen in Figure 3.2.

Gamepad v.2 also uses two 2-axis Joysticks to command the vehicle wheels. The position of each

joystick returns a vector that can be defined as the velocity vector. The two resultant velocity vectors will then be sent to each wheel, the orientation of the vectors are the wheel orientations and their lengths are its value. This method removes the limitations on the wheel orientations and velocities, driving is now non-restricted, also, the velocities values considered is always positive

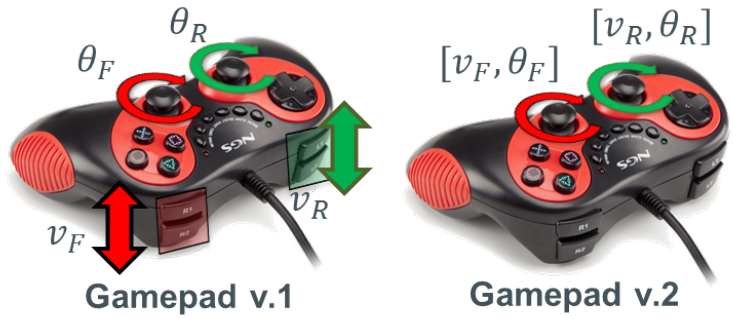

Figure 3.2: Differences between both gamepads

Because the joysticks always return to a zero position, this means that it has a built-in safety feature. If the user leaves the joystick, the velocities sent to each wheel are zero and the vehicle will stop.

Another feature that was developed for Gamepad v.2 is the ability to drive the vehicle as a car, imposing the orientation of one of the wheels in the vehicle to be the same as its heading. This feature should be useful when driving the vehicle across long trajectories without many curves.

Each joystick returns an (x, y) vector of its position, the value and the orientation. With this, the linear velocity vector can be calculated. And two variables are defined, x_{Joy} and y_{Joy} which have values on the interval $[-1, 1]$, these will define the vector v_{Joy}, with its orientation given by equation (3.1):

$$\theta_{Joy} = \tan^{-1}(\frac{y_{Joy}}{x_{Joy}}) \tag{3.1}$$

The length of the vector is obtained using equation ((3.2)

$$v = v_{max}\sqrt{x_{joy}^2 + y_{joy}^2} \tag{3.2}$$

Where v_{max} is the maximum velocity that the vehicle can have, in this case was set to 0.4 m/s by default.

In Figure 3.2 is shown the differences between both devices. The joysticks in Gamepad v.1 are used only to command the wheel directions θ_F and θ_R and the four trigger buttons increase or decrease of the velocity v_F and v_R. Whereas with Gamepad v.2 both the velocities and orientations $[v_F, \theta_F]$ and $[v_R, \theta_R]$ are manually controlled with the joysticks.

In Figure 3.3 on the right is the graphical interface with TES on a map developed for tutorial purposes, and on the right is a user driving with Gamepad v.2, notice the left index finger pressing the button to force the vehicle behave like a car, the vehicle is being driven only with the right thumb.

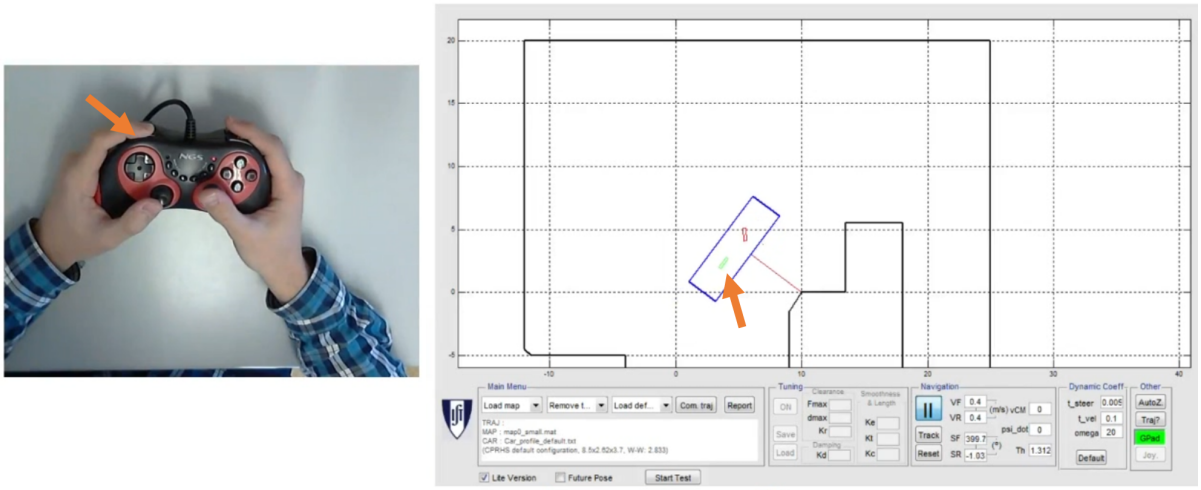

Figure 3.3: Gamepad v.2 - Car-like driving feature

3.3 Center of the vehicle driving device

In order to develop a device to drive the vehicle focusing at its a center, a Microsoft Sidewinder 2-axis joystick and a Lika I65-P incremental encoder were used, the joystick controls the vehicle's center velocity vector and the encoder controls its heading by changing the angular velocity.

The joystick was plugged directly into the computer and the encoder was connected to an Arduino UNO board which was communicating via serial port with Matlab. The script to make the acquisition of data from the encoder was developed and deployed in the Arduino to ease the computational load on Matlab, and can be seen in Appendix A. The first iteration of the rotational disc, instead of an encoder used a rotational potentiometer, it was abandoned because of rotation limitations that potentiometers have.

On Figure 3.4 on the left image are shown the devices that compose the device, from left to right are the Arduino programming board, the Lika incremental encoder and Microsoft's Joystick the core parts of this device, the image on the right shows the previous setup with the potentiometer instead.

Figure 3.4: Setup used to drive the vehicle at its center.

3.3.1 Joystick

The communication between the joystick and the computer running TES is via a built-in function of Matlab, which has the sole purpose of handling joystick devices, which is ”vrjoystick.m”. This function creates an object of the joystick type, and has all the necessary variables, like the joystick xy position and what buttons are being pressed. This is running for as long as the ”enable joystick” button is toggled in the UI and is repeating every 20 ms within a Matlab timer.

The velocity vector was obtained with the same method as the one explained in equation (3.2).

Besides crab like motion, in which the Joystick imposes the same orientation and velocity to both wheels, another mode was implemented that enables the vehicle to be driven like a car. In this configuration the vehicle has the same kinematic model as a bicycle where one of the vehicle wheels is locked and has the same angle as the vehicle heading. This behavior can be seen in Figure 3.5. Where, by pressing button 2 of the Joystick the vehicle rear wheel becomes locked in position and the vehicle starts moving like a car.

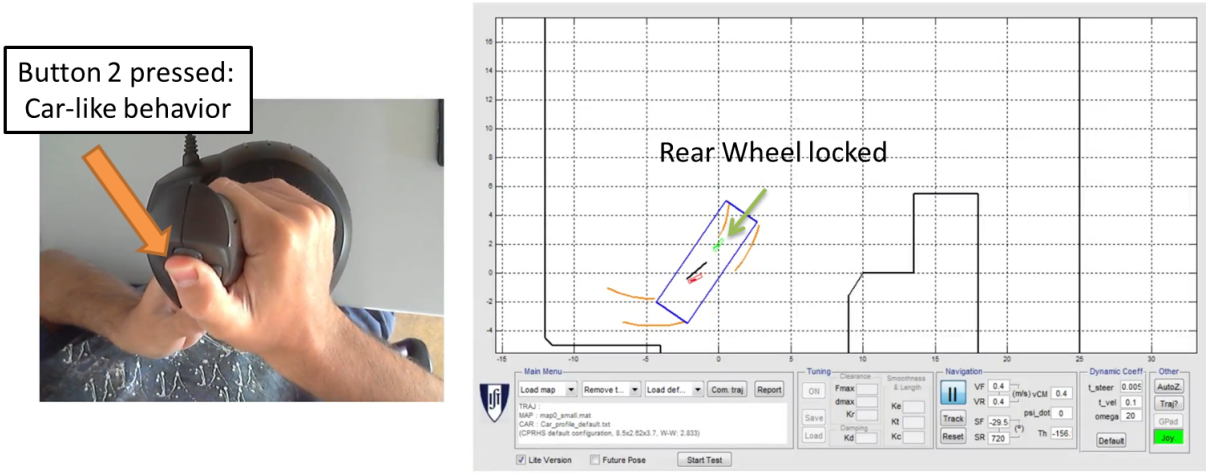

Figure 3.5: Car-like behavior with Joystick

3.3.2 Encoder

The encoder is a Lika I65-P incremental encoder and its response is given by two pulse trains. When rotating clockwise or counter clockwise the phase of the pulse trains shifts, and by analyzing that pattern, a method to acquire the angular velocity can be developed. The data acquisition is done in an Arduino board, the encoder was connected to it using the connections shown in Figure 3.6, the Arduino was then connected to a computer via the USB port.

The encoderPIN A and B send square pulse signals that change depending on the direction of rotation of the encoder knob, and encoderPIN 0 is not used however it signals a complete revolution.

To see the waveforms generated by the encoder it was connected to an oscilloscope, the Tektronix TDS 2014, these can be seen in in Figure 3.7.

The waveform in blue corresponds to encoderPIN B and the waveform in yellow to encoderPINA.

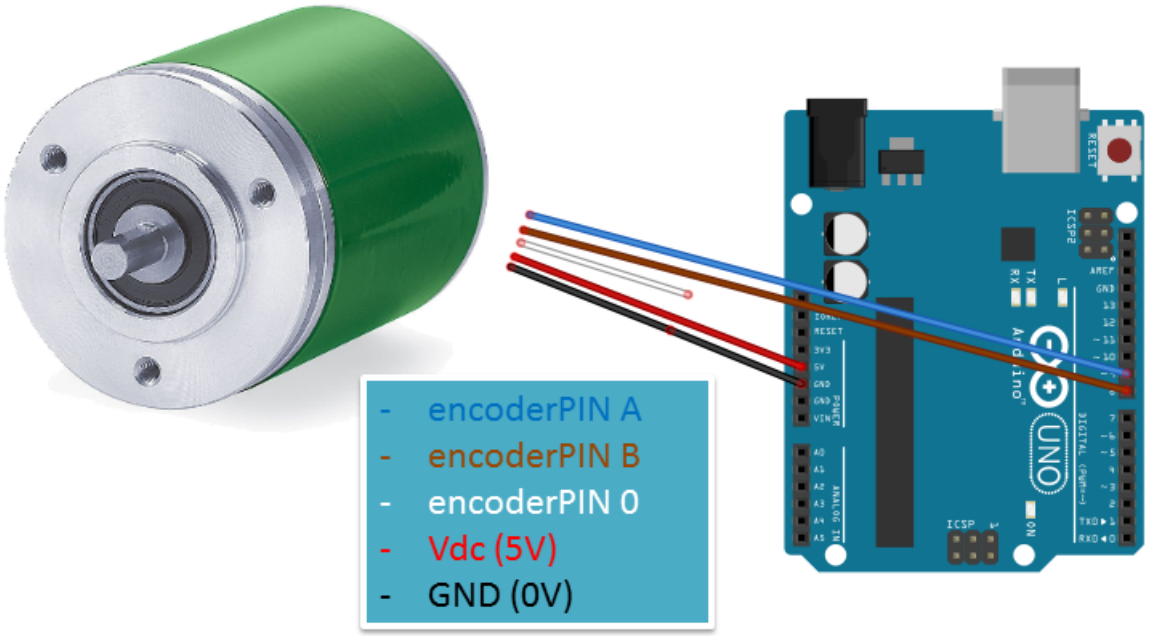

Figure 3.6: Connection diagram between the encoder and the Arduino.

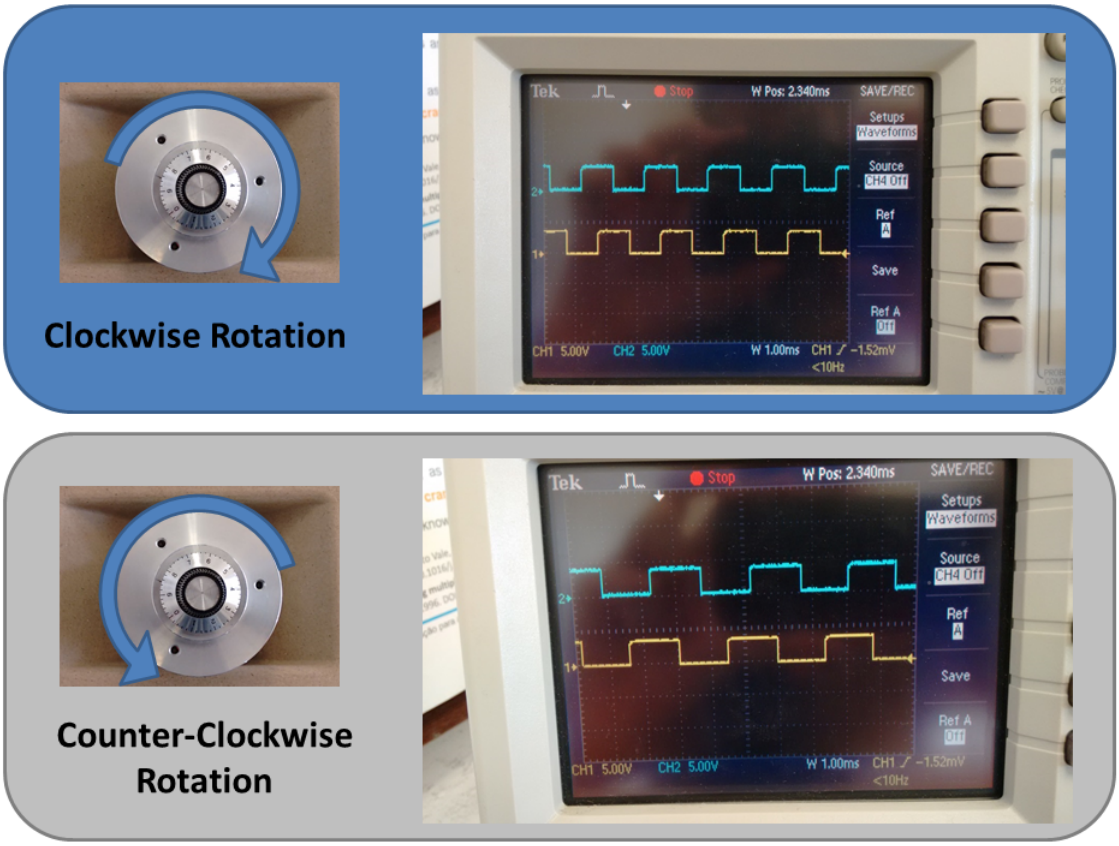

Figure 3.7: Waveforms obtained from the encoder.

When the encoder is being turned in a clockwise motion, encoderPINB square pulse goes to HIGH first, when rotating in a counter-clockwise motion, encoderPINA square pulse goes to HIGH first.

3.3.3 Arduino programming board

In order to ease the load on the computer, a scrip was written and implemented on the Arduino to get the angular velocity from the encoder.

The baudrate was set to 115200 because of the encoder resolution, which has a maximum resolution of 10000 PPR (pulses per revolution), with this resolution and lower baudrates the Arduino was not being fast enough to read the pulses and writing them to the serial port.

However there is a trade-off, because fast baudrates are prone to send invalid data. In Matlab a check is made to confirm if the value read is valid, and in between readings, flushes the serial buffer. The input buffer is also limited to 32 bits in order to keep the buffer size low, it has 32 bits because its the size of the float variable sent by the Arduino corresponding to the angular velocity of the encoder rotation.

The script developed consists in a cumulative variable that increments or decrements, depending on the direction of rotation.

The first step is to detect if the encoder is turning, and what is the turning direction, this is obtained by detecting the rising edges from the encoder pulse trains.

It starts by detecting the rising edge of encoder PIN A, when it happens if encoder PIN B is LOW the counter is decremented, else is incremented. In this part, the pulses are being counted which correspond to the amount rotation.

The amount of time that passes between successful pulse readings is being stored and will be used to calculate the velocity.

When increasing the velocity from 0 to a given value the equation to model it is shown in equation (3.3)

$$v_{enc}(k) = v_{enc}(k-1) + \frac{v_{acq}(k)}{1 + \frac{1}{\tau . T_e}} \tag{3.3}$$

Where v_{enc} is the variable that is storing the encoder velocity, v_{acq} is the instant velocity read by Arduino, τ is a constant value and T_e is the elapsed time between readings.

The velocity when the encoder knob does not move decreases to 0 exponentially, and is shown in equation (3.4).

$$v_{enc}(k) = v_{enc}(k-1)e^{-\alpha T_e} \tag{3.4}$$

Where α is a constant value to tune the decaying, this method makes the velocity decrease smoother.

In each cycle this value is stored in an array and after a fixed number of cycles, the array is averaged giving an estimation of the angular velocity. Averaging the values removes oscillation, which results in smoother movements.

In Figure 3.8 is a simplified fluxogram of the script that makes the data acquisition of encoder values and converts it in a float value to be used as angular velocity. Bellow is the description of the variables used.

- **startTimer and finishTimer:** Float variables that are used to measure time lapse between data

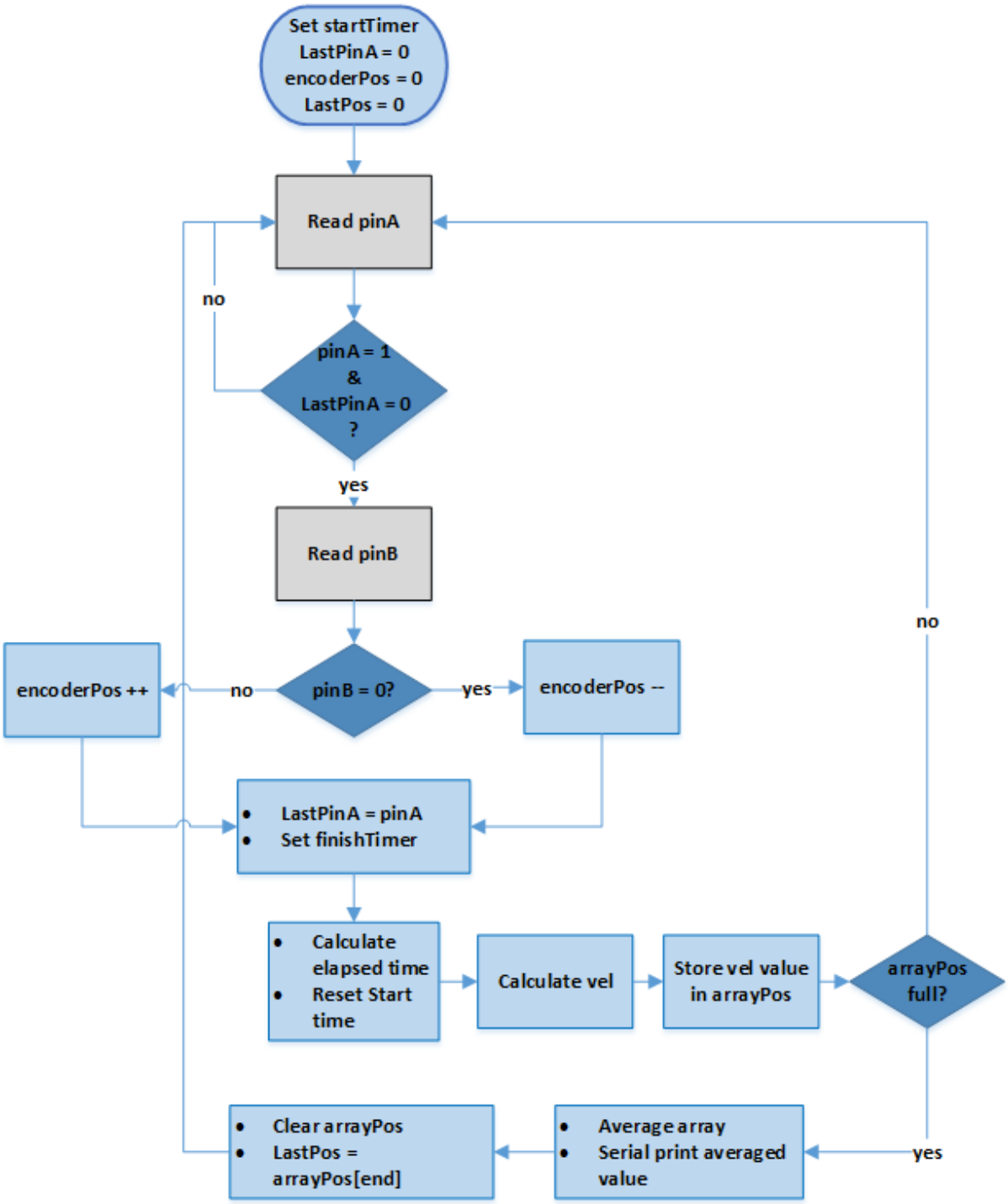

Figure 3.8: Encoder data acquisition flowchart.

acquisition.

- **pinA and pinB:** Boolean variables that store the respective pulse signal values.
- **encoderPos:** Integer variable that is incremented or decremented according to the direction of rotation, stores number of pulses.
- **LastPinA:** Boolean variable to store the last value of pinA.

- **LastPos:** Float variable to store the last velocity value.
- **arrayPos:** array of floats that store multiple velocity values.

The scrip developed can be seen in Appendix A

3.3.4 Final device

The device is a combination of the previous topics with an interface developed with Matlab, using TES as the graphical interface, and with an encoder and joystick as input devices, on Figure 3.9 is shown the result. On the right is what is being seen on the computer running TES, the scenario that is being used is half of the ITER's Tokamak building, and on the left is an operator driving the CTS/CPRHS with both hands, one on the joystick and the other on the encoder.

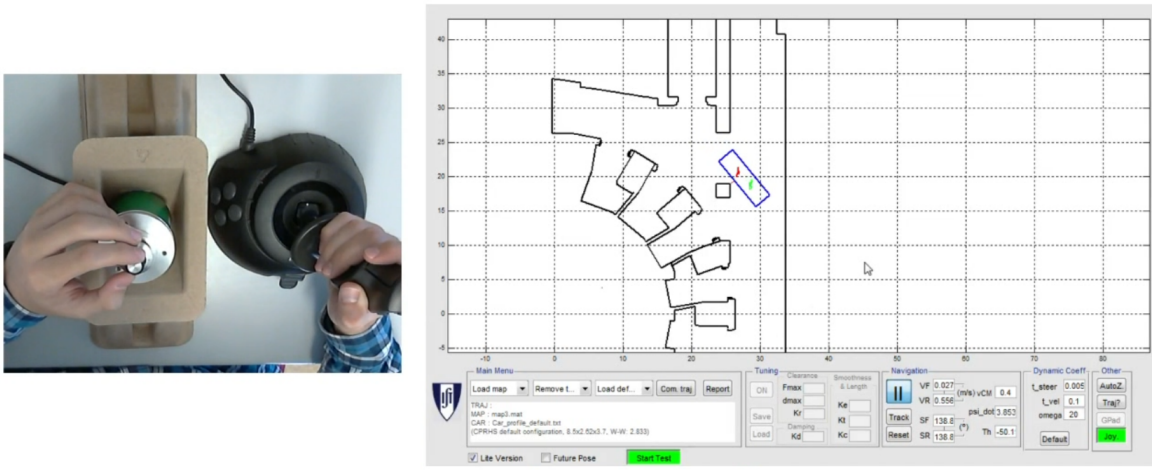

Figure 3.9: Device to drive the vehicle at its center

3.4 Complementary content

3.4.1 3D printed encoder support

The proposed design for a control desk shown in Section 2.4.3 was not possible to develop during this thesis because it needed many components that could not be acquired. However a support for the encoder was designed with CAD and produced in a 3D printer, available at IPFN labs. The printer used was a Ultimaker 2, and the software used to design each part of the support was Solidworks. The 3D virtual part was later converted to STL format and imported to Cura, which is the software that is needed to send the project to the 3D printer. In Cura is it specified how the part will be printed choosing the printing parameters such as, the printing speed, the part resolution which depends on how thick is the plastic being extruded, etc. In Figure 3.10 on the left is the representation of disc on Cura and on the right is the final product after it was printed.

The support is composed of three different parts: A disc that latches to the encoder rotating shaft, and the supporting base that needed to be split in two pieces in order to fit in the printing area of the 3d

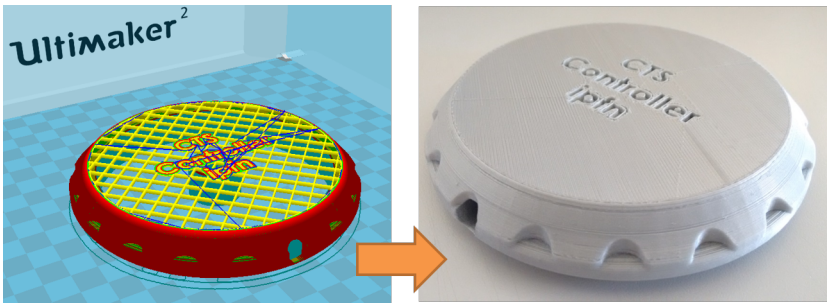

Figure 3.10: Cura and resulting 3D print

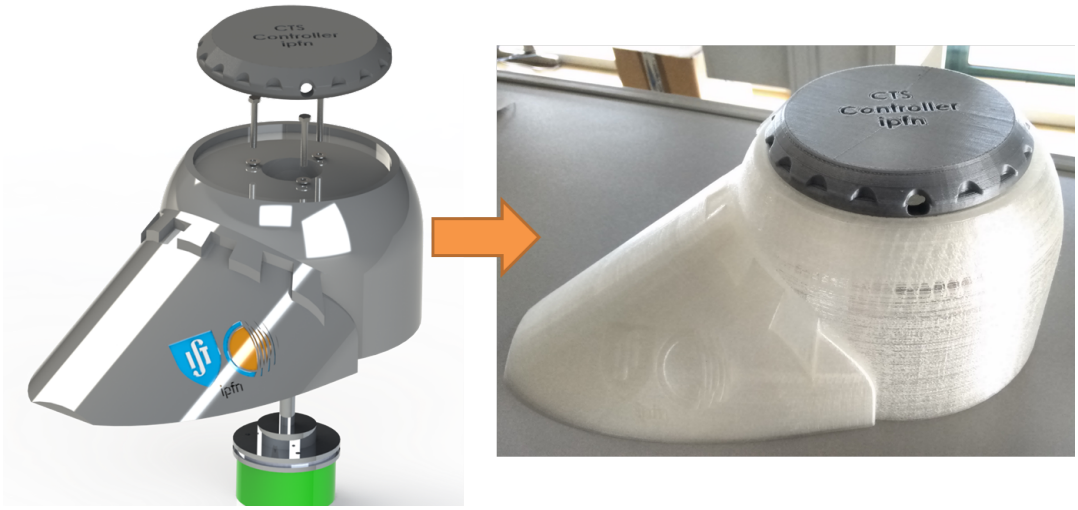

Figure 3.11: Encoder support from cad model and 3D print

printer. The different parts that compose the device can be seen in Figure 3.11 on left, where is shown an exploded view of the device, on the bottom in green is the 3D model of the encoder, on the right part of the Figure can be seen the final 3D printed support.

3.4.2 Safety curves to aid driving

In order to prevent collisions and ease the driving task, a little add-on was implemented in the interface that draws four curves, one for each of the CTS corner. The curves represent the future positions of the vehicle corners, this extra information is expected to help the driving operation. This is accomplished by simulating the vehicle dynamics N time steps in the future.

In Figure 3.5 the orange curves represent the visual representation of the vehicle vertexes keeping the same driving in the same way.

3.4.3 Interface with a rhombic-like vehicle prototype

A small prototype of the CPRHS (scale 1:25) was built in LEGO and developed at IPFN [31] in a previous project that was not part of this thesis. The prototype has a rhombic-like configuration with wheels capable of turning endlessly. It uses a the Mindstorm NXT as the controller, and communicates with Matlab through a software, also developed at IPFN, RobDAC. The NXT controller is used as motor

drive and as mean of communication between the prototype and the computer running RobDAC, the computation is all done on the computer side. The prototype can be seen in Figure 3.12.

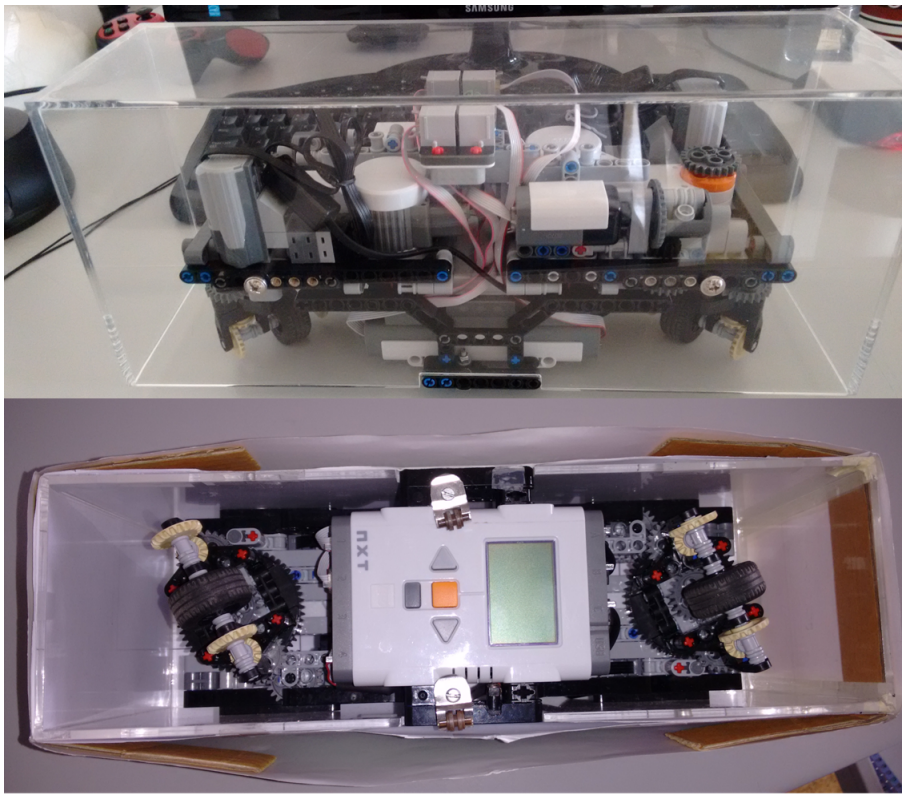

Figure 3.12: CPRHS prototype, seen from bellow

The communications between the computer/Matlab and the prototype are made via bluetooth. A serial object is created in the Matlab and is controlled almost the same as the simulated CTS in TES. Like TES, RobDAC had the same manual control, the Gamepad v.1. The implementation of the software developed to drive the vehicle at its center and gamepad v.2 only needed some compatibilities figured out. The controller needed the vehicle's heading information, which is impossible to acquire with the desired accuracy without external sensors, so the values being sent to the vehicle are in its own frame instead of the world frame, in this case the heading feedback is not needed and the vehicle can still be driven. Driving the prototype in its own frame shows a different behavior than the one experienced when driving the simulated vehicle.

Chapter 4

Experimental Results

This Chapter shows the results of driving a rhombic-like vehicle with Gamepad v.1, Gamepad v.2 and JRD.

In order to evaluate the devices a set of metrics was chosen, number of collisions, safety distance, energy, distance to a trajectory, the length of the generated trajectories, the velocities and orientations of the vehicle wheels and its heading. The tests are taken in three different scenarios, each with a different map and goal.

The first set of tests were taken by an experienced user, the author of this thesis, where all three devices were tested in three different scenarios.

In the second set of tests 12 people without experience with the devices or the vehicle, drove with Gamepad v.2 and JRD, a total of 200 tests were done. Because Gamepad v.1 shown undesired results with the experienced user, it was not used for these tests. The scenario used in these tests is always the same due to its complexity, half of the tokamak building of ITER.

4.1 Devices and Test Scenarios

In this Section are explained the three devices that will be used to drive the vehicle and shown the four scenarios where the tests will take place, these are presented in Figure 4.1.

- Gamepad v.1: Device to drive the vehicle focusing at its wheels, it was the driving method used in TES v2.0 and is shown in Section 3.2, each joystick commands each wheel orientation and the four trigger buttons increase and decrease the velocity of each wheel.

- Gamepad v.2: Device to drive the vehicle focusing at its wheels, it is an improved version of Gamepad v.1. To drive the vehicle the user needs less buttons than the previous version, and the wheel orientations are not limited.

- JRD (Joystick and Rotational Disc): Device to drive the vehicle focusing at its center, it was shown in Section 3.3, it merges a Joystick that commands the center velocity vector of the vehicle and

an encoder to control its heading, the method developed by Alonzo Kelly to transform center of vehicle variables into wheel variables is the one chosen to be used in the tests.

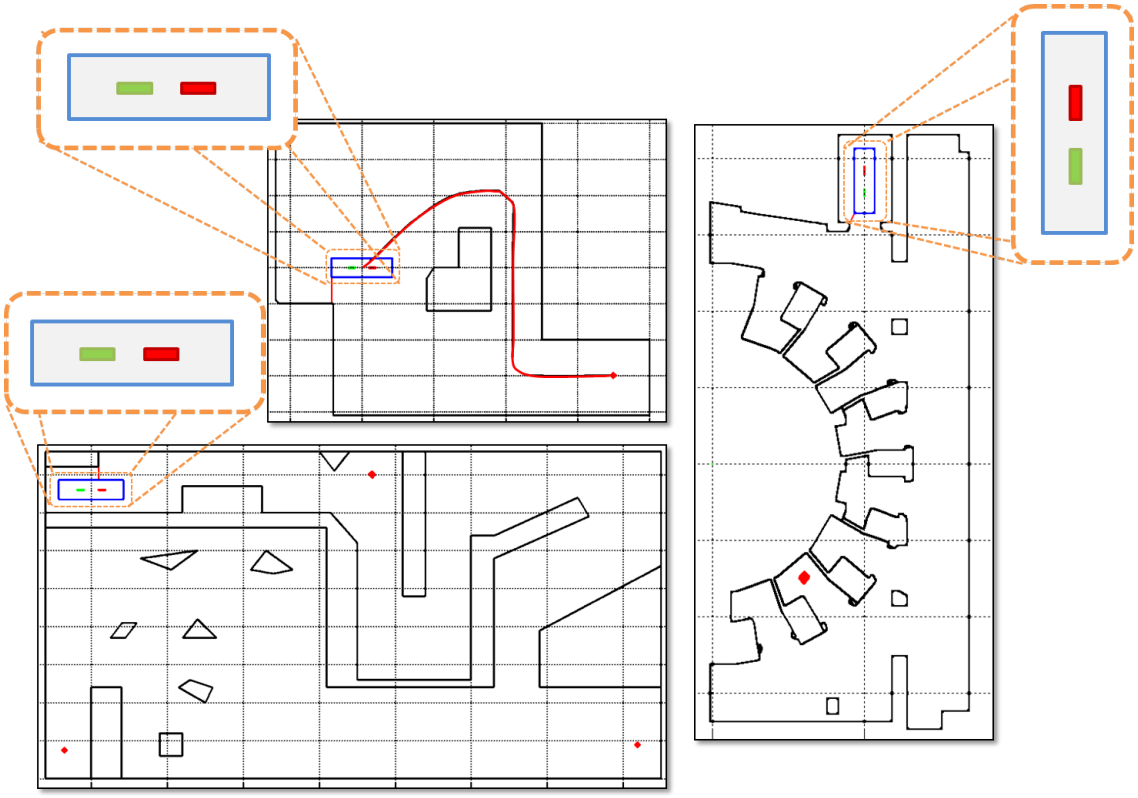

Figure 4.1: Different Maps used in each trial

- Scenario 0: It is used as a tutorial area, in it the operator can try all the functionalities of each device, without worrying with a goal to achieve or avoid collisions with the walls, the map used is shown in Figure 4.1 in the top left image.
- Scenario 1: It takes place in the same map has the first scenario, but this time the operator needs to follow a trajectory, the goal in this scenario is to drive the vehicle center as close to the trajectory as possible, the trajectory is the red curve shown in Figure 4.1 in the top image
- Scenario 2: It was developed with the goal of a global performance evaluation for each device, it has places that can be better navigated with crab-like movements, others where the free roaming shines. In order to complete this trial the user must reach all three checkpoints marked with red diamonds. This trial is impossible to complete without the omni-directional capabilities of the vehicle, the map for this scenario is shown in the bottom image of 4.1.
- Scenario 3: It is half of Tokamak building lower level, where the CTS is expected to operate. The goal with it is to simulate an operation of the CTS using the manual mode instead of the automatic mode. Going from the start point inside the elevator to one docking port on the bottom of the reactor, it is shown in Figure 4.1 on the left. The vehicle is placed at the starting position and the goal port is marked with a red diamond.

4.2 Metrics Chosen

In order to evaluate the performance of each device, some metrics were chosen, and are the following:

- **Number of collisions** - This metric counts how many times a user crashed with a wall of the scenario, in order to obtain a better score the number of collisions should be low. It is not expected to have the vehicle clashing with the scenario, however due to the velocity that the vehicle can achieve in the simulator, it is expected that some collisions will occur.
- **Safety distance:** - This metric evaluates the danger of collision of the vehicle with a wall. In each cycle the simulator is evaluating the shorter distance between the vehicle and the nearest wall. This distance is measured in meters. This value should be as high as possible, meaning the vehicle is far from a collision.
- **Energy:** - Energy in the context of this thesis is the product of the linear velocity with the angular velocity summed along time, and it is calculated for the rear and front wheel.
- **Time Duration:** - The time that takes for a user to complete a trial. The duration of each trial is measured in seconds.
- **Distance to the path:** - A scenario 2 has a pre-computed path that the user needs to follow, this metric evaluates the distance between the vehicle's center and that trajectory. Measured in meters. The values needs to be small, meaning that the user is successfully following the trajectory.
- **Path Length** - The length of the generated path from driving the vehicle in each trial, it is measured in meters.
- **Velocity and wheel orientations:** - The different wheel velocities and orientations across each trial will be recorded and are related to the driving stability, motor strain and driving patterns.
- **Vehicle Heading:** - The changes in the vehicle heading along each trajectory, the obtained values will translate in the level of oscillation the vehicle is subject while moving and to detect driving patterns.

The most important metrics are the safety distance and the number of collisions, undesired results in them can endanger the vehicle safety. Energy relates with the motor strain, a trajectory with many oscillations generates more energy than a smooth trajectory, it is also used to identify driving patterns. Wheel velocities, orientations and vehicle heading are used to identify driving patterns and oscillatory behavior. The trajectory length and time duration are not so important, they depend on how the user tackles the trial and how safe he likes to drive.

In order to obtain meaningful values from these metrics the mean and standard deviation were calculated, the first is represented by a colored curve, and the second by a gray shaded area.

The data that was acquired during the tests needed to be matched and this was done by aligning the trajectory length of each test. In a set of results in the same scenario the trajectories shapes and lengths

are almost the same. The data was matched to the trajectory that has less elements in its array, to do this additional arrays were created that store the cumulative sum of the trajectories length. These new arrays were then multiplied by the ratio of the array being matched with the array that has the smallest trajectory. Figure 4.2 has a diagram of the process.

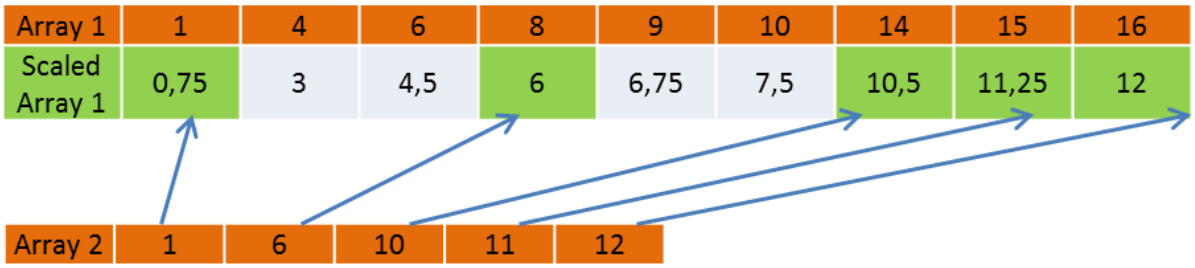

Figure 4.2: Method to match arrays

4.3 Experienced user

After developing the interface, the devices and scenarios, the author of this thesis is considered an experienced user because of the amount of time he spent developing the devices and driving the vehicle with them. All the devices and scenarios will be tested and evaluated by him and the results will be used to tune the simulator. The results obtained in these tests are needed to clarify how the test group evaluation should be done. A total of of 50 tests were recorded and evaluated, 20 on scenarios 2 and 4 and 10 on scenario 3.

4.3.1 Scenario 1 - Trajectory following performance

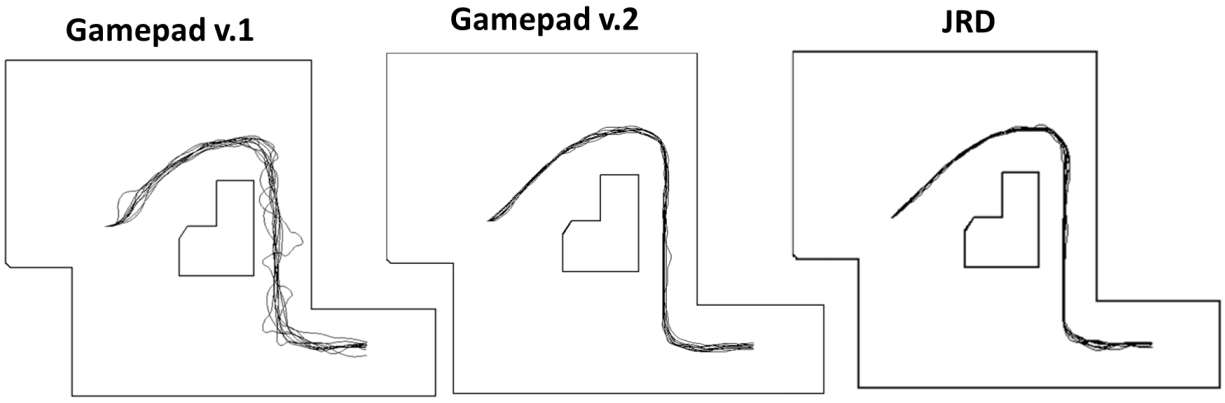

Figure 4.3: Obtained trajectories on scenario 1 with an experienced user

Figure 4.3 has the paths that resulted from using each controller in scenario 1 where the map is simple and the goal is to follow a trajectory. Gamepad v.1 has less tests than the other two because after a couple of them, its low performance was noticed.

By looking at the generated trajectories by Gamepad v.1 they have many oscillations when compared with the other two devices. Gamepad v.2 and JRD produce smooth trajectories without any major oscillations. Due to the low level of complexity, the evaluated metric is the distance from the vehicle center to the trajectory.

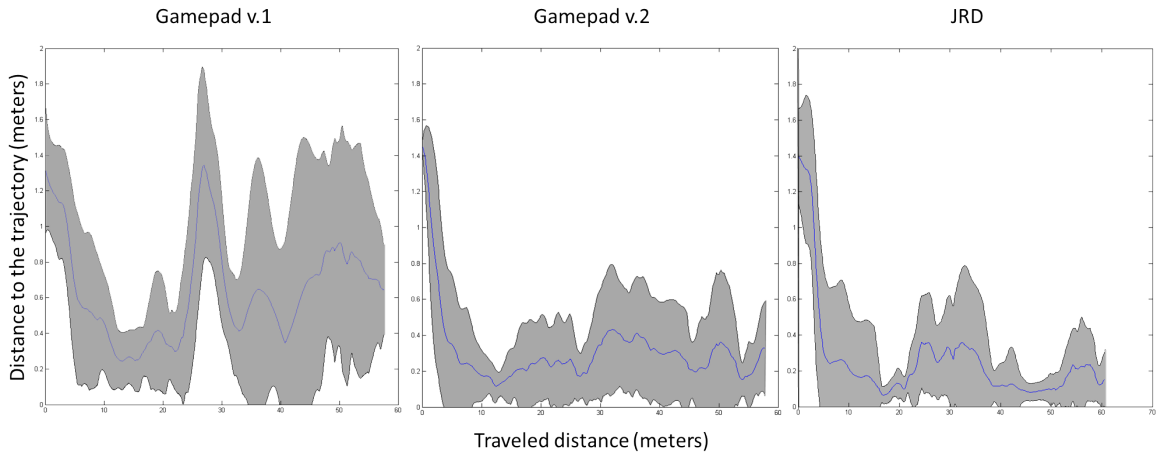

Figure 4.4: Distance to the trajectory in scenario 1

In Figure 4.4 is shown the results for the distance between the vehicle center and the path, the graph does not start at zero because the vehicle center was not placed on top of the path at the start of the test. The controller with the worst performance is Gamepad v.1 almost everywhere has higher values. The shaded area is especially big on the last half of the tests, which means that the user was no able to produce similar trajectories between trials. The other two devices produce similar graphs, however JRD has the smaller shaded area for the last half of the test, the user was able to produce similar trajectories.

Evaluating the trajectory following capabilities, both Gamepad v.2 and JRD are good choices, with JRD being slightly better.

4.3.2 Scenario 2 - Safety Distance and Collision Performance

In Figure 4.5 in the top left image the map is divided in seven zones depending on how easy is to drive in them, and the trajectories resulted from driving the vehicle with all three devices. There are cases of wall intersections with Gamepad v.1 because it was impossible to drive the vehicle from the start to the end without any collisions. It can be seen that Gamepad v.2 and JRD have better results, however in zone 4 Gamepad v.2 has more oscillations than JRD, since it excels at movements that do not require the vehicle to change its heading, like in zones 1, 3 and 5.

The safety distance can be seen in Figure 4.6, the top left image are the results for Gamepad v.1, the top right image for Gamepad v.2 and at the bottom for JRD. The worst device is Gamepad v.1, many times the average values drop bellow the safety threshold which is represented by the red line. The other two devices have few places where the safety distance is near the safety threshold , and the results for JRD show a smoother curve. It is also important to note that in order to complete this scenario with Gamepad v.1, the vehicle crashed a total of 49 times whereas with the other two no crashes occurred.

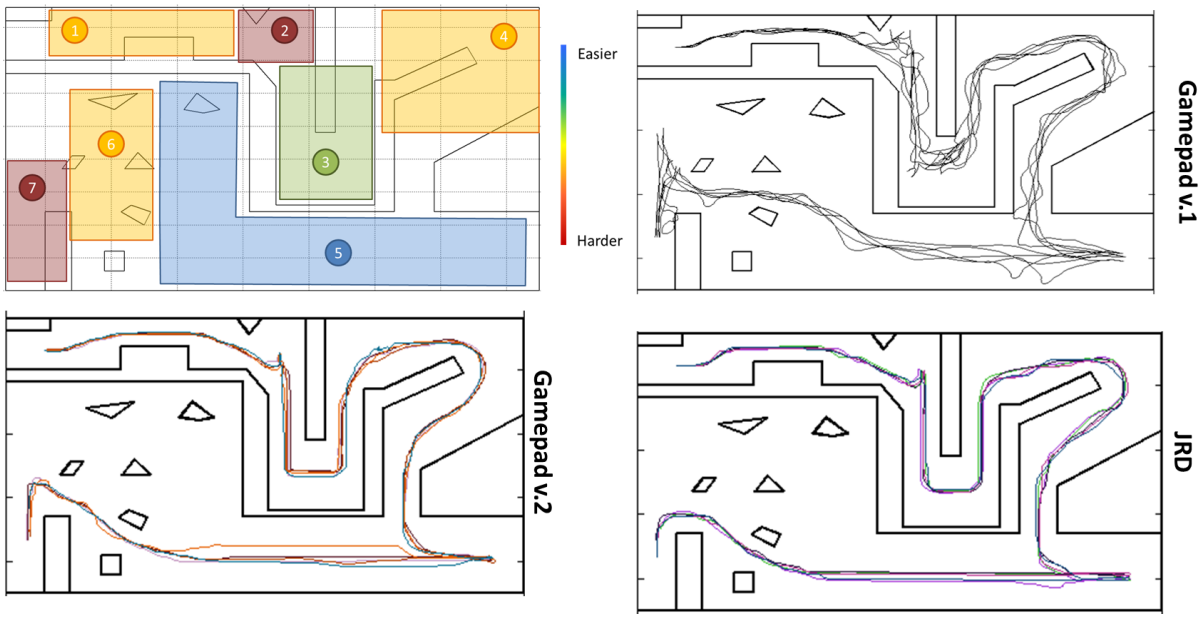

Figure 4.5: Generated trajectories and complexity zones of scenario 2

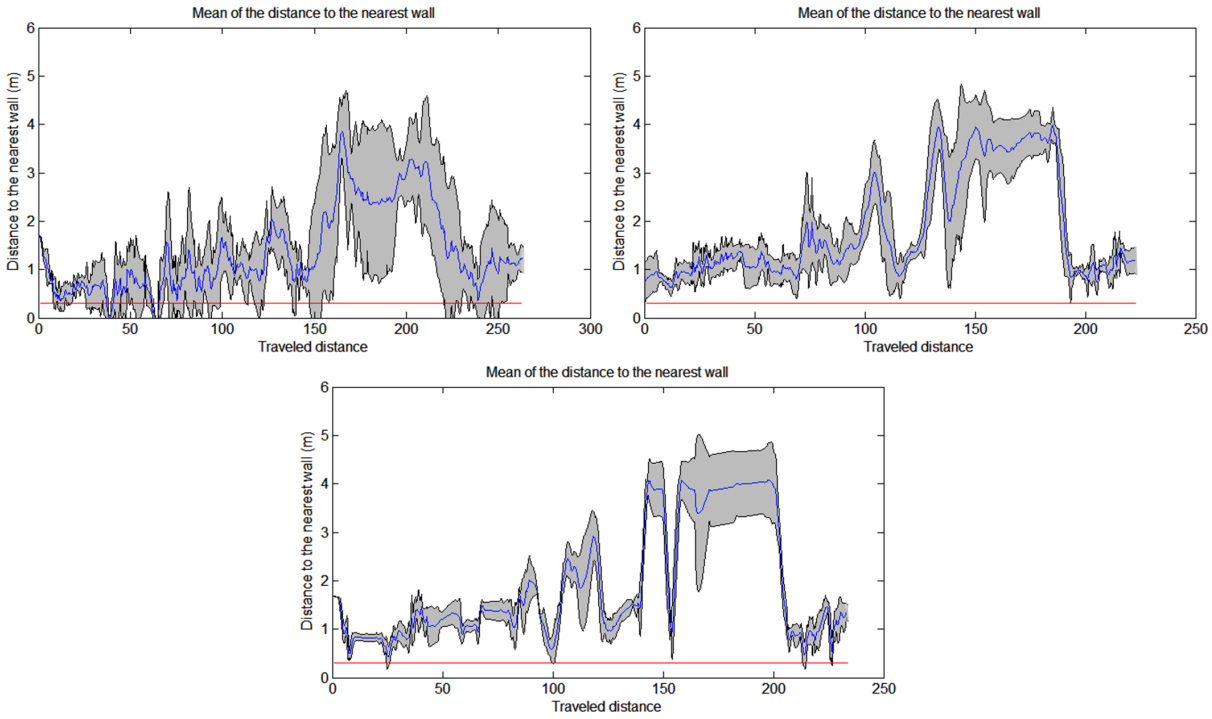

Figure 4.6: Distance to the nearest obstacle in scenario 2

The values for the energy spent in each wheel for Gamepad v.1 will not be shown because the device was driven at slower speeds and the method of how angles and velocities are sent to each wheel is not the same as the other two devices.

In Figure 4.7 is shown the time durations, it can be seen that there is a huge difference between the durations of Gamepad v.1 and the durations of the other two devices, JRD and Gamepad v.2 have similar values of time duration.

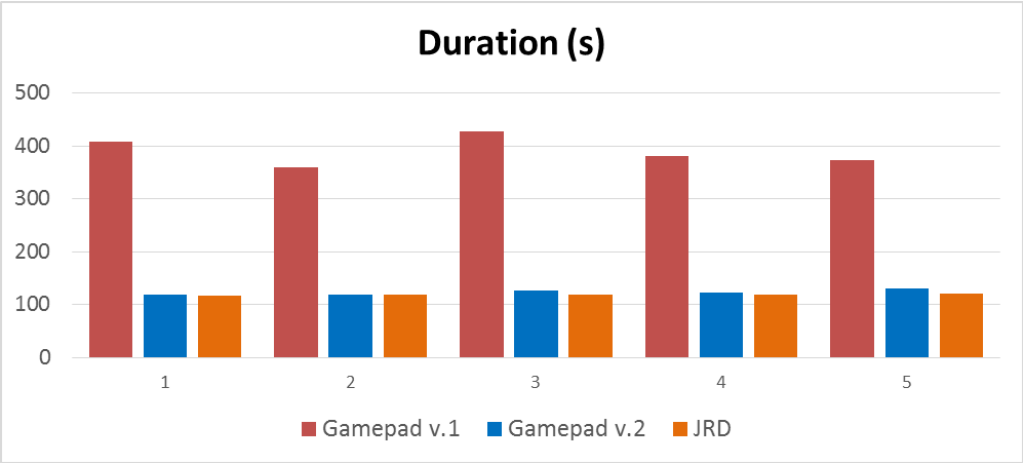

Figure 4.7: Time duration of the trials made in scenario 2

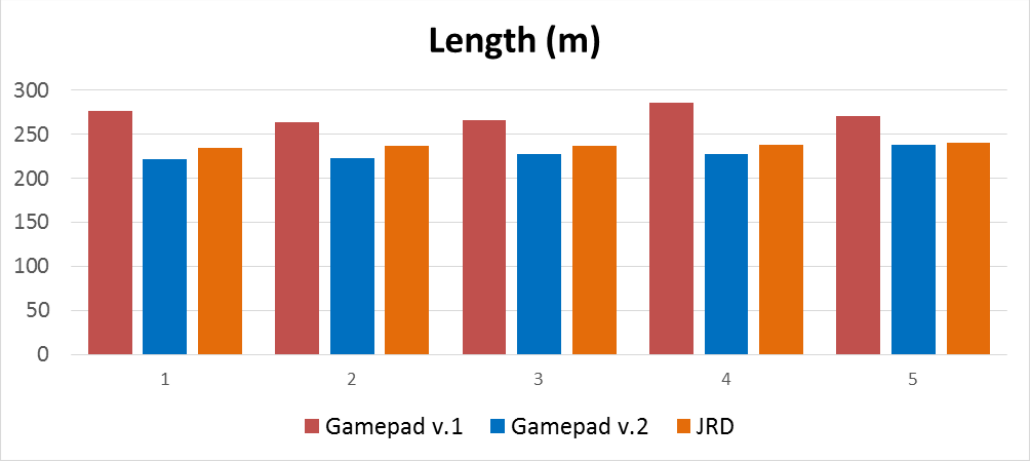

Figure 4.8: Length of the trajectory of the trials made in scenario 2

In Figure 4.8 is shown the trajectory lengths, which have similar values for all devices, however Gamepad v.1 produced the higher values, this is due the corrections that had to be made across the trial to compensate the mistakes done while driving. This information can be crossed with the trajectories produced by Gamepad v.1 in Figure 4.5 where it can be seen that the trajectory had to be corrected many times, hence the oscillations.

In Figure 4.9 is shown the energy spent by the front and rear wheels across all trials, the energy is higher with Gamepad v.2 than JRD because the vehicle oscillates more when is being driven with Gamepad v.2.

4.3.3 Scenario 3 - Overall performance

Following are the results obtained from controlling the vehicle on scenario 3. In Figure 4.10 on the right side the map is divided in five different zones according to their complexity, the other three graphs plot all the trajectories for each device, which are the xy positions of the vehicle across all tests using the three different controllers.

In the case of Gamepad v.1, it is noticeable that the trajectory on zone 1 is perfect, on all tests the

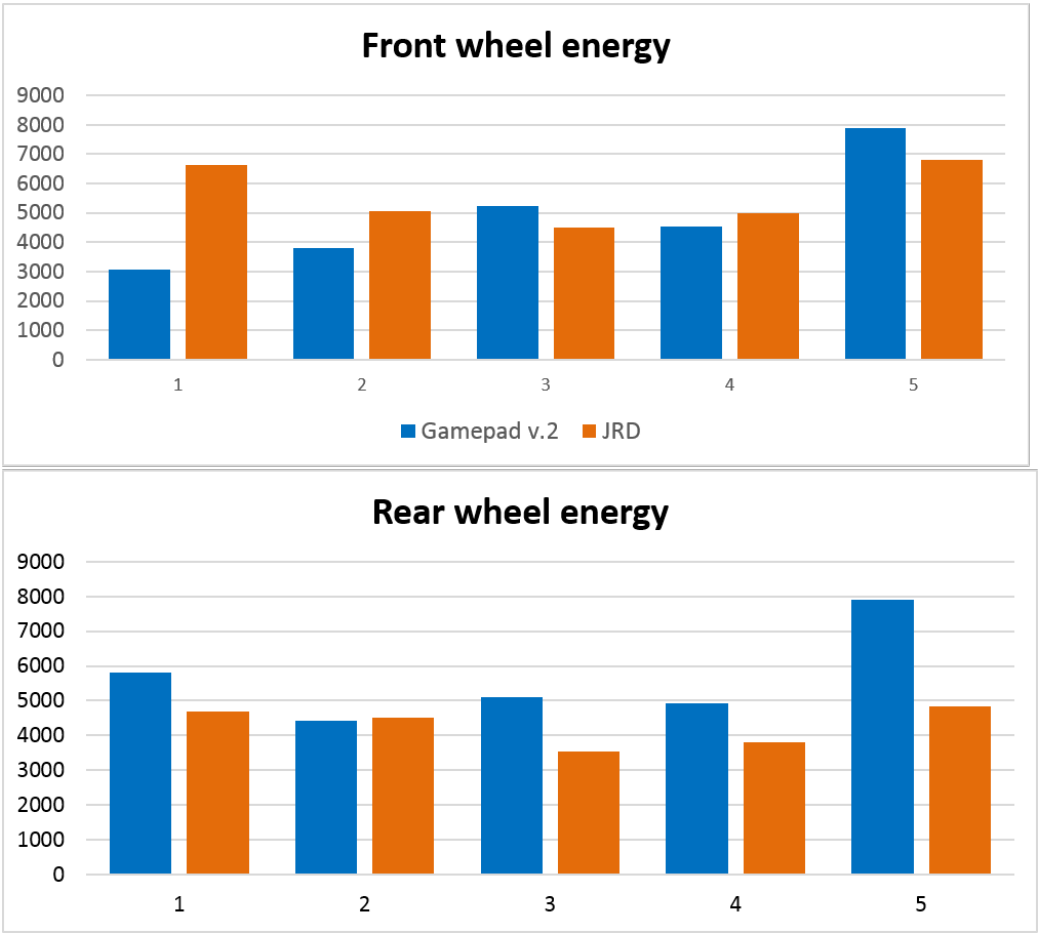

Figure 4.9: Energy spent in trials made in scenario 2

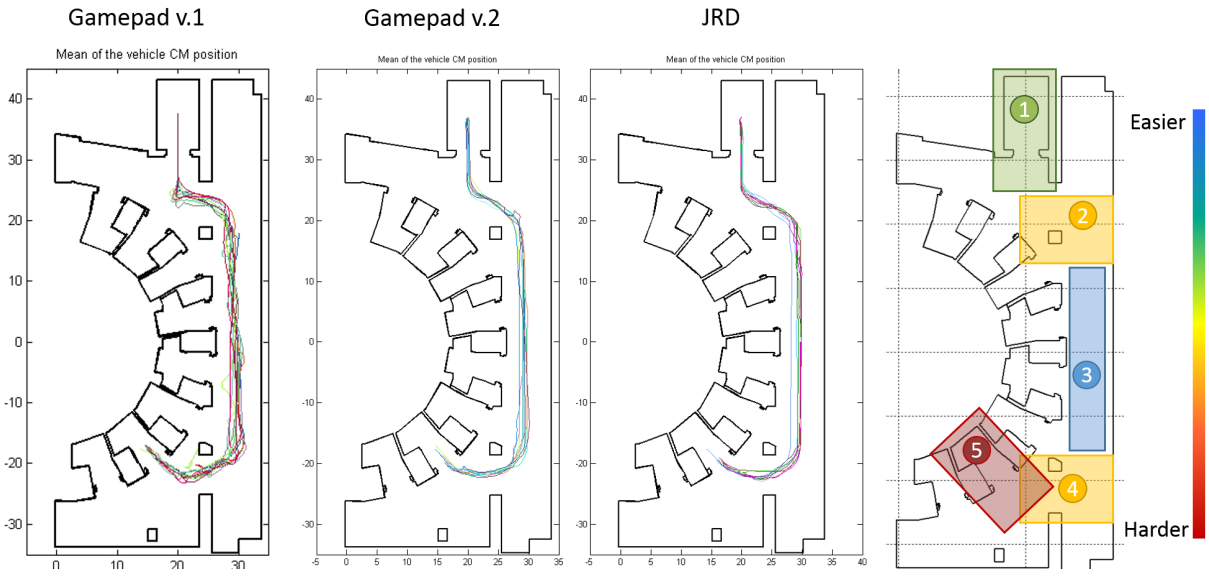

Figure 4.10: Generated trajectories and complexity zones in scenario 3

vehicle went straight down, however the problems start to appear as soon as it is needed to turn, in zone 2, the generated trajectory starts to oscillate, making the transition from zone 2 to zone 3 rather unstable. Even on zone 3 where the path is easier, there are different patterns on the generated trajectories. The

oscillating behavior aggravates when reaches zone 4, where it is needed to contour the pillar. Lastly when zone 5 is reached, the oscillating behavior continues. When looking at the second and third images from the left that correspond to results obtained with Gamepad v.2 and JRD respectively, the first thing to look at, is how smooth the trajectories are on zones 3, 4 and 5, when compared with the other device. Looking at the trajectories generated from analyzing the data both Gamepad v.2 as JRD are suited.

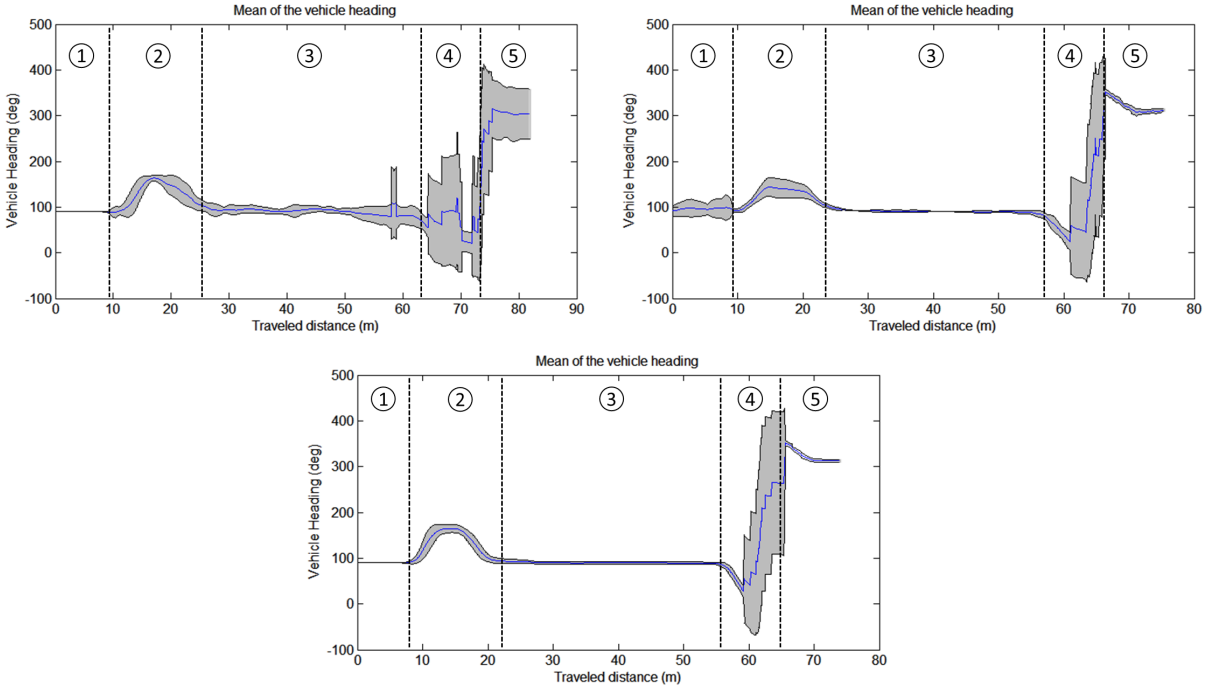

Figure 4.11: Average heading across all tests on scenario 3 with an experienced user

In Figure 4.11 is shown the vehicle heading changes, each section relates to the corresponding zones in Figure 4.10. This can be related with motor strain, in order to change the vehicle heading the motors need to be actuated.

In the case of Gamepad v.1, exiting the lift there are no heading changes as expected, however when making the curve to enter the docking port, zones 4 and 5, the generated curve has many oscillations and the shaded area is also larger than the other two devices. With Gamepad v.2 and JRD the shaded area is small and the curves have similar shapes, meaning that the user was able to reproduce the similar trajectories, the evolution of the heading is also much smoother with these two devices. Looking at the vehicle heading Gamepad v.2 and JRD are suited to drive the vehicle.

In Figure 4.12 is shown the average distance to the nearest obstacle, each section relates to the corresponding zone in Figure 4.10. When using Gamepad v.1 the shaded area is larger than when using the other two devices, also the vehicle tends to have lower safety margins especially at the end of test. It is important to mention that in order to complete the test with Gamepad v.1 in this scenario, a total of 30 collisions occurred. The only zone where the shaded area goes bellow the safety threshold with Gamepad v.2 and JRD is in the last part of the trajectory, on zone 5.

In Figure 4.13 is shown a trajectory that was generated in TES on the left, and on the right is the

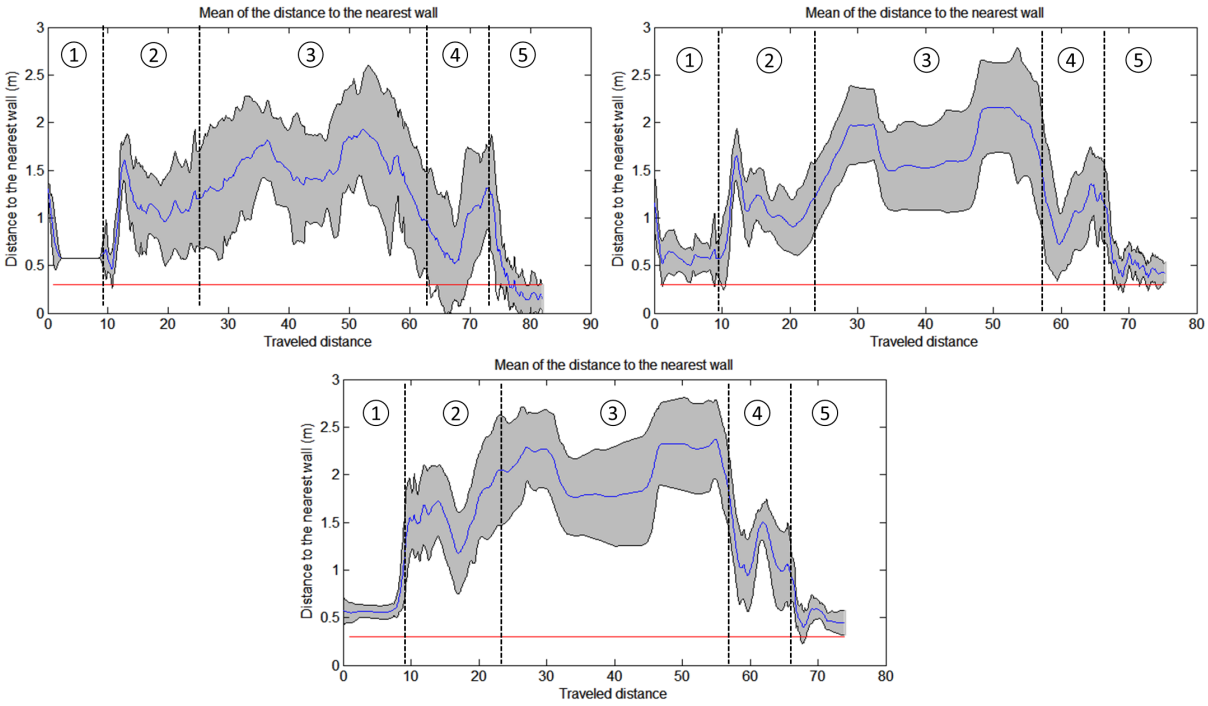

Figure 4.12: Average minimum distance to the nearest obstacle across all tests on scenario 3 with an experienced user

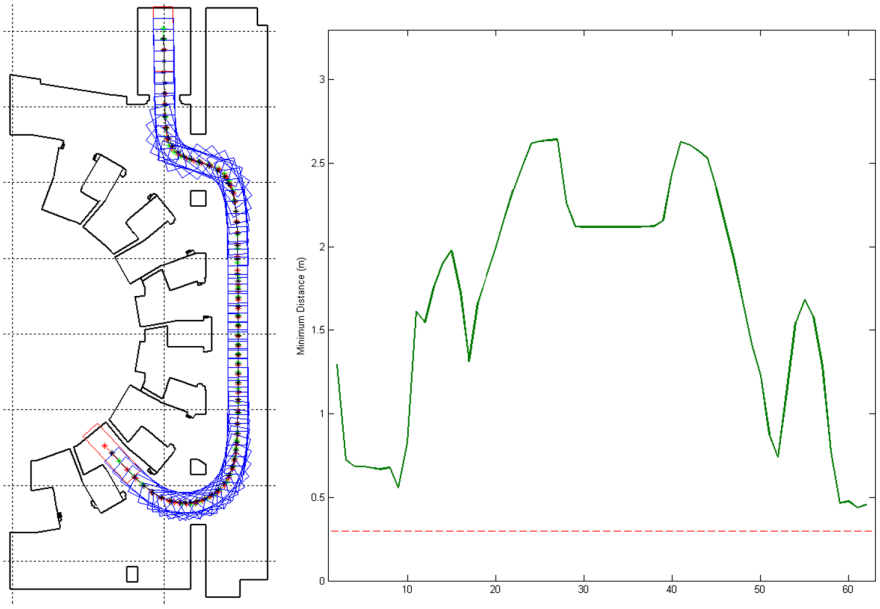

Figure 4.13: Minimum distance to the nearest obstacle, automatic trajectory

minimum distance to the nearest wall, and the dotted red line is the safety threshold. Comparing the shape of the resulting graph with the three graphs in Figure 4.12, it resembles the results obtained from Gamepad v.2 and JRD.

Like the results in scenario 2, the device whose tests were longer is the Gamepad v.1 and this results can be seen in Figure 4.14, because the velocities that were used with this device are lower than the other two since it was impossible to drive the vehicle at high speeds.

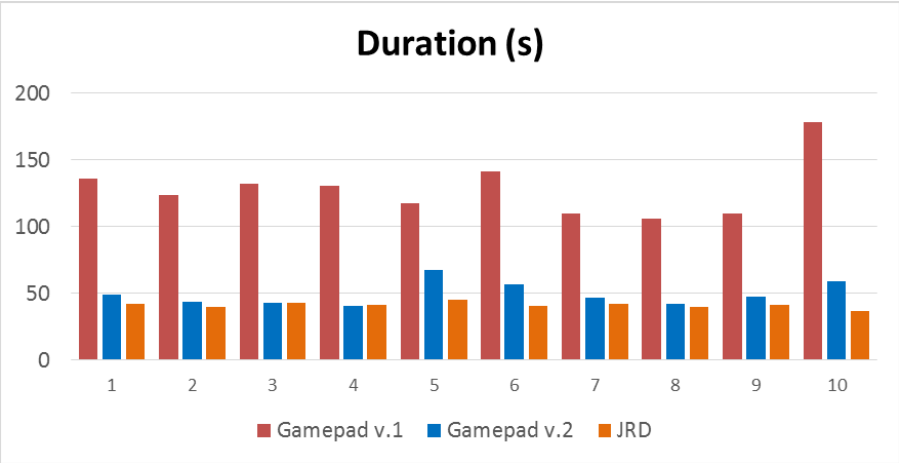

Figure 4.14: Time duration of the trials made in scenario 3 by an experienced user

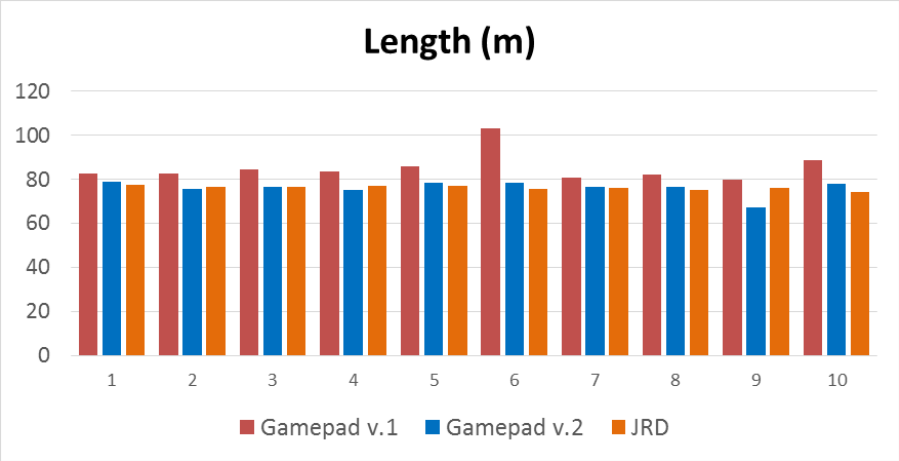

Figure 4.15: Length of the trajectory of the trials made in scenario 3 by an experienced user

In Figure 4.15 is shown the trajectory lengths for all trials, which have similar values for all devices with Gamepad v.1 producing the higher bars, this is due to corrections that the operator has to due in order to achieve the goal point.

In this scenario it can be seen that with Gamepad v.2 one wheel has higher energy values than the other, this can be seen in Figure 4.16 in this case the rear wheel, which is the wheel that was the main focus of attention when driving is this scenario, looking at Figure 4.1 the rear wheel is the green wheel and is the first wheel to leave the starting area. Besides this the other wheel was also kept at a constant orientation during most of the test, at the exception of zones 4 and 5 where in order to dock the user needs to use both wheels and the vehicle omni-directional capabilities. With JRD the energies are distributed between both wheels more evenly, this is because the driving device, not the user, manages both wheels.

4.3.4 Notes for the group tests

The tests with the experienced user were useful because they served to structure the tests for the test groups, and it was seen that Gamepad v.1 is the device with the worst performance, more oscillations,

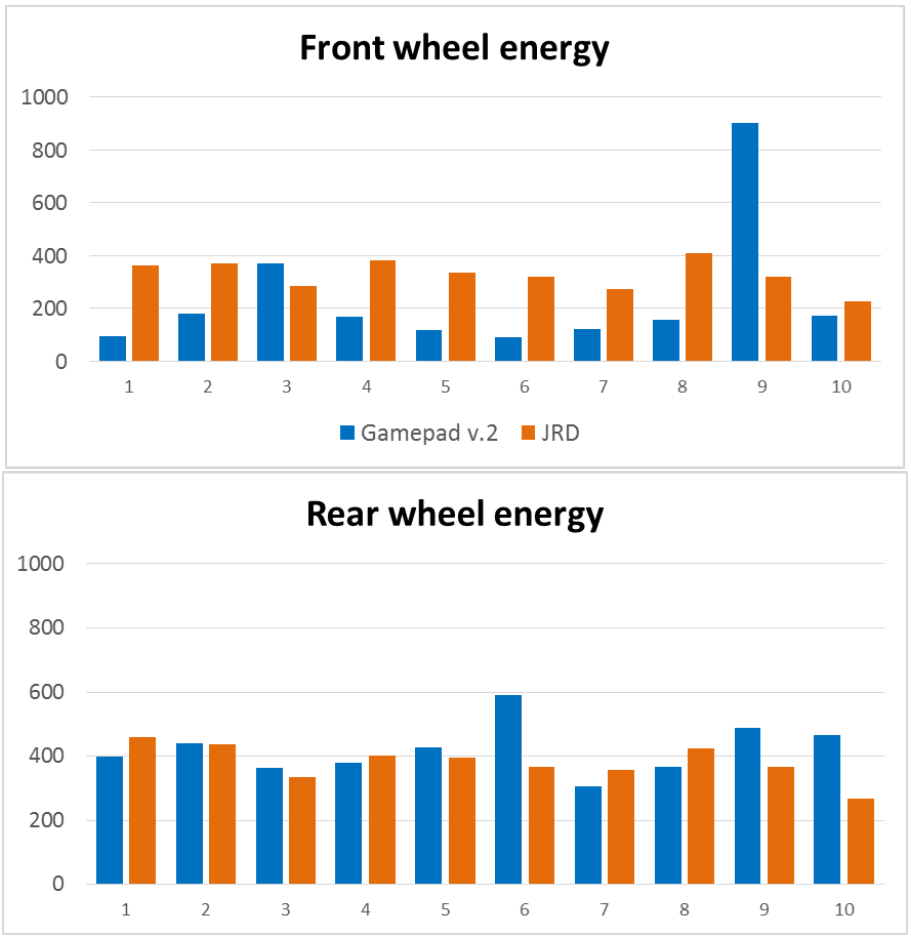

Figure 4.16: Energy spent in trials made in scenario 3 by an experienced user

large amount of collisions, produces longer trajectories and takes more time to reach the end.

Comparing the results from the three scenarios, scenario 1 can be completed without using the vehicle omni-directional capabilities so it will not be used in the group tests, scenarios 2 and 3 have a high degree of complexity, but scenario 2 takes a lot of time to complete and has large areas where the users are free to navigate making it difficult to identify driving patterns. Scenario 3 will be chosen to be tested by the user groups because has a complex layout, driving patterns can be identified and is also half of ITER's tokamak building which is where the CTS will operate.

4.4 Trials with the test groups

To evaluate the performance of each device, two test groups were created with people with no previous experience with TES or rhombic-like vehicles. The people that took part in the groups have an age range from 11 to 50 years old, with an average of 26 years. All of them have experience with computers and 58% had experience with console games or gamepads, a total of 12 people were involved in these usability tests and were separated in two groups. A total of 200 trials were performed, each user had to complete the trial five times, which means going from the start position to the end goal without collisions, with a total of 120 successful tests.

The two devices being evaluated are the Gamepad v.2 and JRD, users from Group 1 start by taking the test with Gamepad v.2 and after five successful trials switch to JRD, and the opposite for users of Group 2.

Before driving in scenario 3, all users test each device in scenario 0 where they were informed of each device features, the function of each button and are allowed to drive the vehicle freely. After the user has become acquainted with the device at hand he switches device and repeats the same process. After the user has become fully acquainted with each device, the scenario changes and the evaluation process takes place. The scenario that is used for evaluation purposes is half of the lower floor of the tokamak building.

4.4.1 Collision evaluation

In order to keep the vehicle and its cargo safe collisions should not be allowed. This metric is also one of the most simple, a high number of collisions is not wanted, so the most suited device should be the one with less collisions.

Table 4.1: Number of collisions across the test group trials

	Gamepad v.2						JRD					
# Trial	**1**	**2**	**3**	**4**	**5**	**Total**	**1**	**2**	**3**	**4**	**5**	**Total**
User 1	10	0	1	0	0	**11**	0	0	1	0	0	**1**
User 2	0	1	1	1	0	**3**	0	0	0	0	0	**0**
User 3	14	6	1	0	0	**21**	0	0	4	0	0	**4**
User 4	2	1	1	1	0	**5**	0	0	0	0	0	**0**
User 5	1	0	0	0	1	**2**	0	1	0	1	0	**2**
User 6	1	0	0	0	0	**1**	0	0	0	0	0	**0**
Total Group 1	**28**	**8**	**4**	**2**	**1**	**43**	**0**	**1**	**5**	**1**	**0**	**7**
User 7	6	0	1	1	0	**8**	2	2	0	0	2	**6**
User 8	2	0	0	1	0	**3**	0	0	0	0	0	**0**
User 9	3	0	0	0	0	**3**	0	0	0	0	0	**0**
User 10	0	0	0	0	0	**0**	0	0	1	0	0	**1**
User 11	1	0	0	0	0	**1**	0	0	1	0	0	**1**
User 12	1	0	3	1	0	**5**	1	1	0	0	0	**2**
Total Group 2	**13**	**0**	**4**	**3**	**0**	**20**	**3**	**3**	**2**	**0**	**2**	**10**
Total Collisions	**41**	**8**	**8**	**5**	**1**	**63**	**3**	**4**	**7**	**1**	**2**	**17**

In the table 4.1 is shown the number of collisions across all trials with all users. It can be seen that the device with most collisions overall is the Gamepad v.2 with a total of 63 collisions, whereas JRD had 17 collisions. It can also be seen that most collisions with the Gamepad v.2 happen in Group 1. This group started the tests with the Gamepad v.2, the first contact with the testing scenario was with this device which explains the high number of collisions. The same behavior also happens in Group 2, where the number of collisions with the JRD is higher, however the number of collisions is not as big as Group 1 with Gamepad v.2. The collisions of Group 1 with the Gamepad v.2 are approximately 68.3 $\%$ of the total collisions with that device, and the collisions of Group 2 with JRD approximately 58.8$\%$ with that device.

Overall, with Gamepad v.2 most collisions happen in the first trials. After some successful trials, the

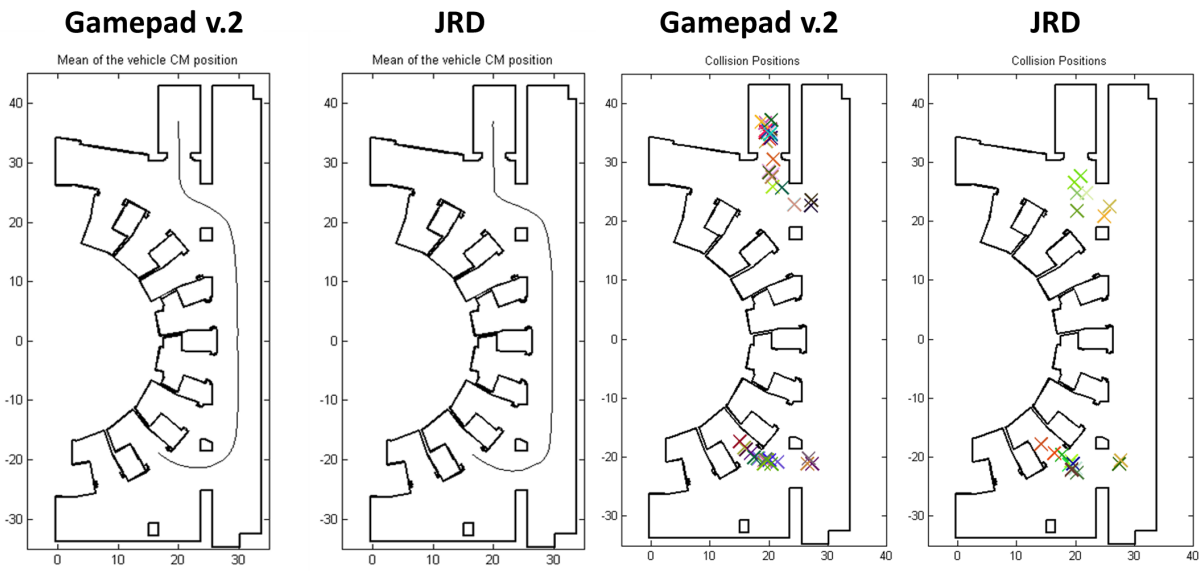

Figure 4.17: Average generated trajectory and position of the collision points

users learn how to navigate the map and the number of collisions highly decrease. Another assumption is that after a couple of successful trials, the user memorizes the trajectory and drives mainly by muscular memory. With JRD, the number of collisions are evenly distributed, especially if User 3 is taken out as an outlier. Taking into account the number of collisions, JRD is more intuitive and user friendly than Gamepad v.2.

In Figure 4.17 the right image shows the positions where collisions happened. Looking at both images it can be seen that they happen almost in the same zones, with the exception of the high number of collisions leaving the starting area when controlling with Gamepad v.2. The location where the vehicle starts is narrow and in order to leave it the user needs to go straight down, and for some users this was not an easy task. In Figure 4.17 on the left is shown the average trajectory produced by the vehicle center across all trials, both trajectories have identical shape. All the generated trajectories can be seen in Appendix B.1 on Figure B.1.

4.4.2 Safety Distance

Evaluating the safety distance is hard, because it changes depending on the type of user. Some users want to avoid collisions at maximum and drive far from the walls, whereas other users are confident that they can drive without collisions and drive close to the walls to achieve faster times, although this was not an intended behavior.

In Figure 4.18 is shown the average distance to the nearest wall for Gamepad v.2 on top and JRD on the bottom. The red line is the minimum safety distance of 30 centimeters and it is only crossed by the shaded area, in the last part of the trial when the user is parking the vehicle in a very narrow space.

The differentiating factor between the two approaches is the oscillations, the top curve has more oscillations than the bottom curve. This is because while controlling the vehicle with Gamepad v.2 the vehicle tends to oscillate more than when controlling with JRD, this behavior can only be confirmed when

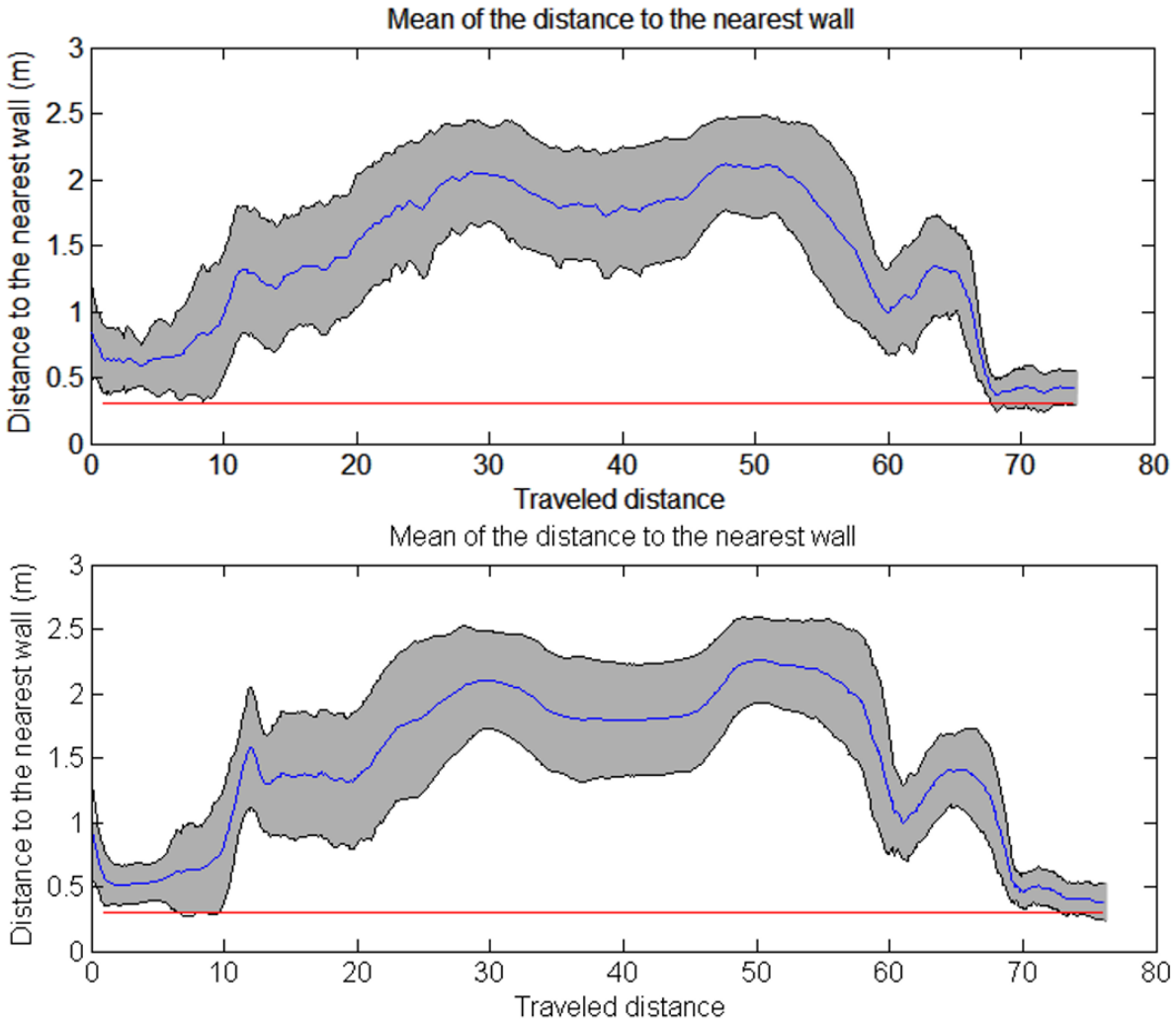

Figure 4.18: Average of all minimum distances to the nearest wall

looking at the vehicle heading and wheel orientations. All individual results can be seen in Appendix B.2.

4.4.3 Wheel orientations and vehicle heading

Some users while driving the vehicle with Gamepad v.2 chose to drive focusing only on one of its wheels locking the front wheel which is the red wheel from Figure 4.1. This makes the vehicle behave as a car. The wheel orientations for Gamepad v.2 can be seen in Figure 4.19, where the bottom image shows rear wheel results and the top front wheel results. The different driving behaviors cause the front wheel to have a larger shaded area, the individual results for each user are shown in Appendix B.3, and can be seen that there are different patterns in the wheel orientations.

The wheel orientation values with JRD can be seen in Figure 4.20, where both wheels have similar behaviors, with smaller shaded areas. This means that controlling the vehicle at its center leads to less motor strain and the vehicle behaves similarly with different people.

In Figure 4.21 is shown the average of the vehicle heading for all tests. It can be seen that the shape of both images is the same, however in the case of Gamepad v.2, the top image, there are more

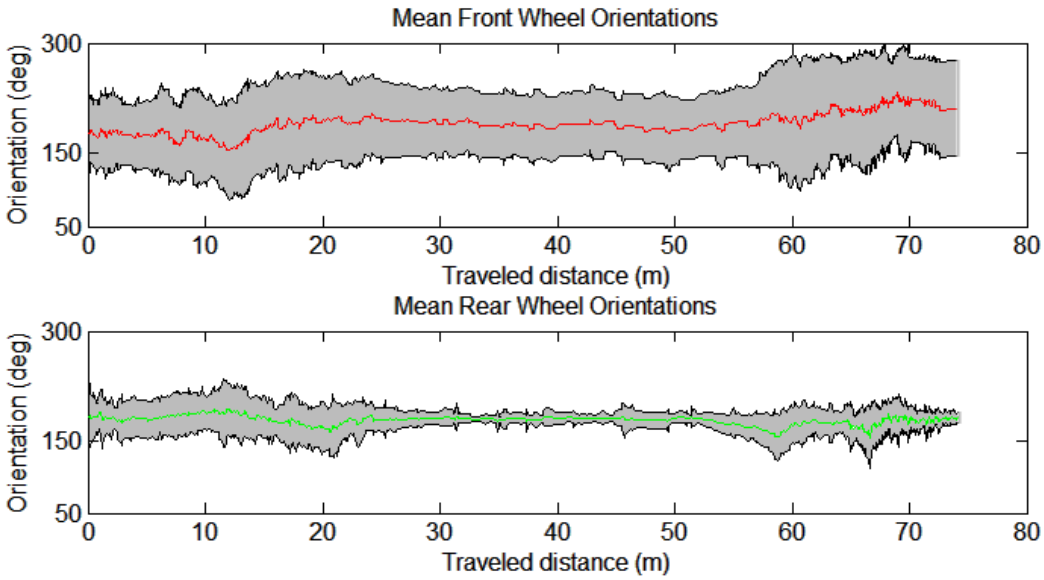

Figure 4.19: Average of the wheel orientations with Gamepad v.2

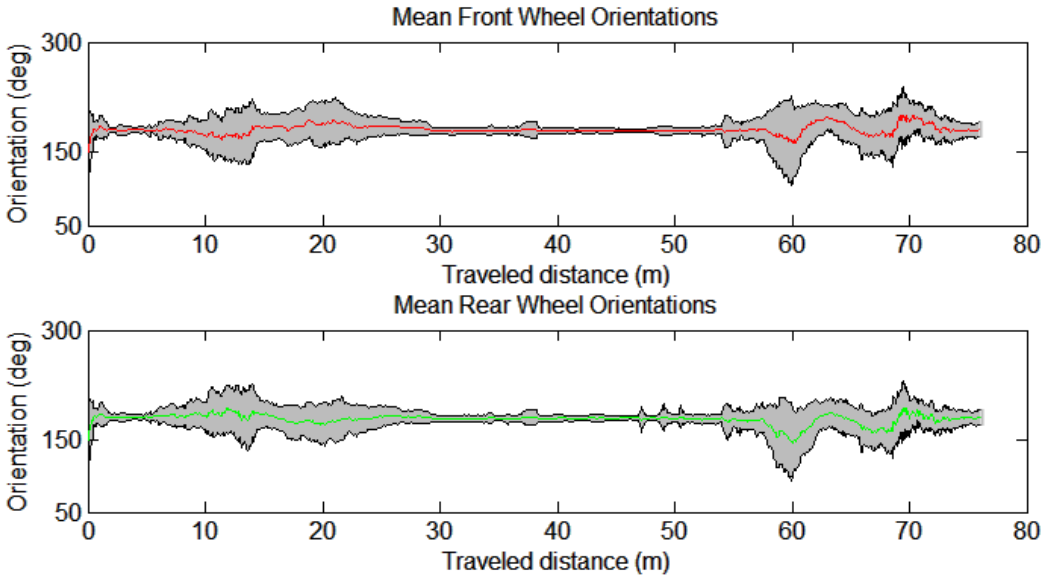

Figure 4.20: Average of the wheel orientations with JRD

oscillations in the heading values. To see the individual results for each user redirect to Appendix B.4 for the heading results and Appendix B.3 for the wheel orientations.

4.4.4 Wheel velocities

Wheel velocities are used to qualify the user behavior while driving the vehicle and motor strain. In Figure 4.22 is shown the average of the wheel velocities for all trials with Gamepad v.2, in the bottom image is shown the rear wheel values and on the top is shown the front wheel values. The first aspect that can be seen is the difference between the front and rear wheel graphs, front wheel velocity has a much larger shaded area, this is mainly due to the different driving behaviors of the users, which can

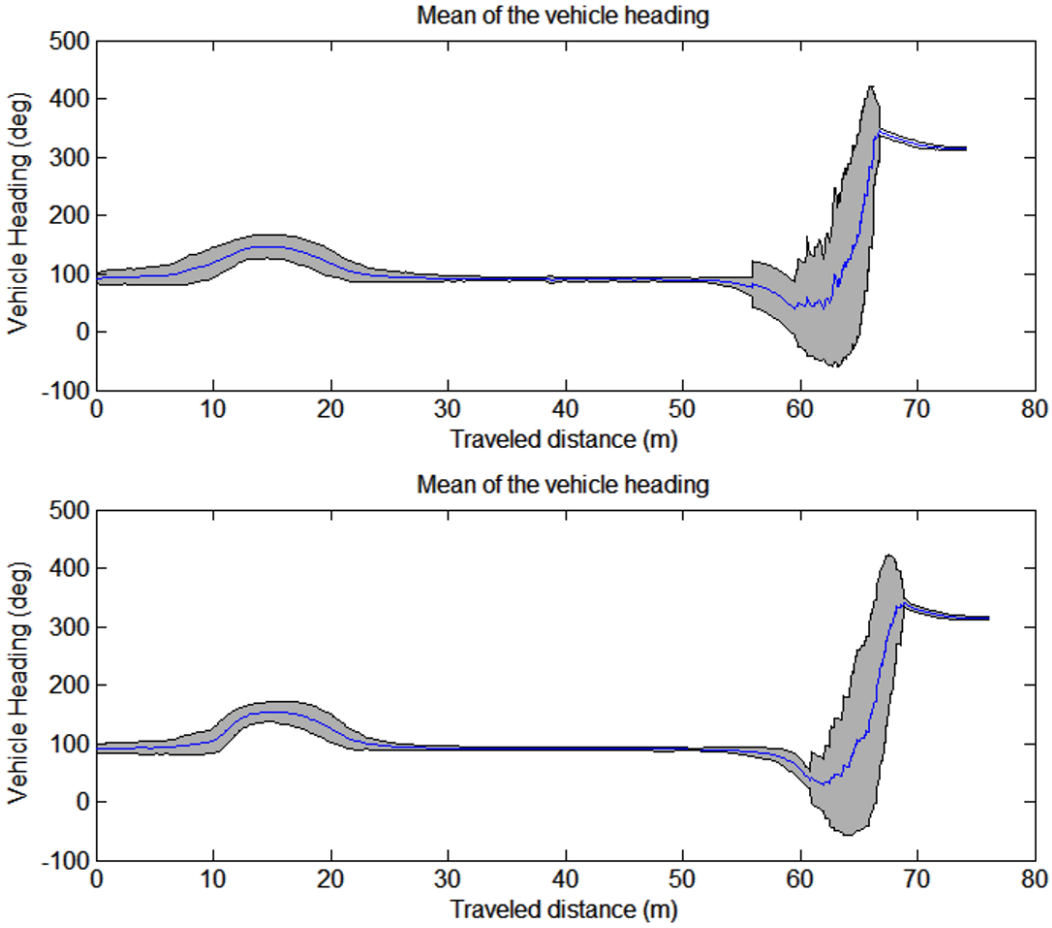

Figure 4.21: Average of the vehicle heading

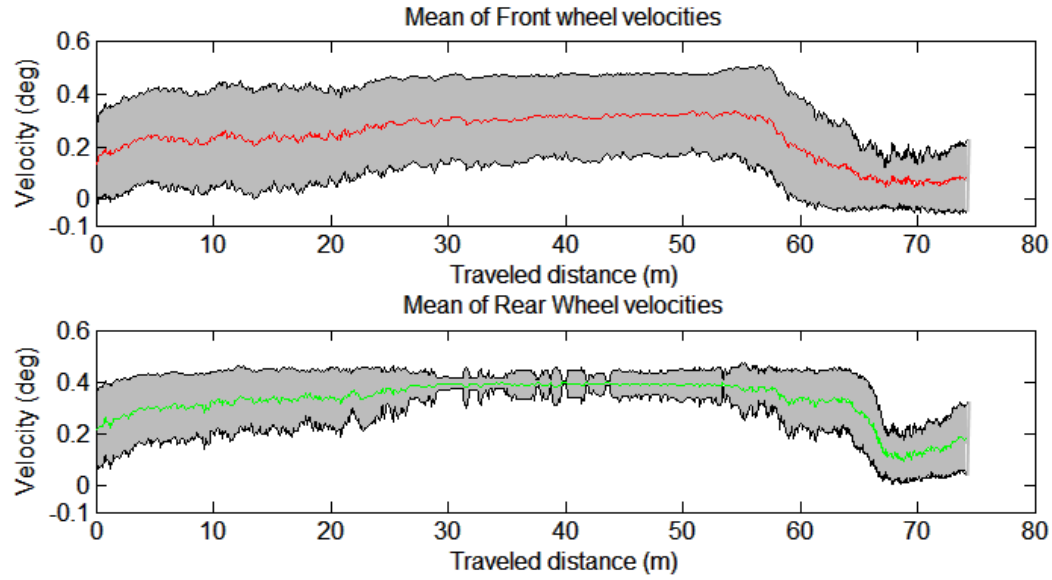

Figure 4.22: Average of the wheel velocities with Gamepad v.2

all be seen in Appendix B.5, some used the option to drive the vehicle as a car, whereas others simply kept the front wheel unattended with its velocity equal to zero and finally there are those that drive the vehicle acting on both wheels. In both cases it is noticeable the reduced velocity when arriving near the

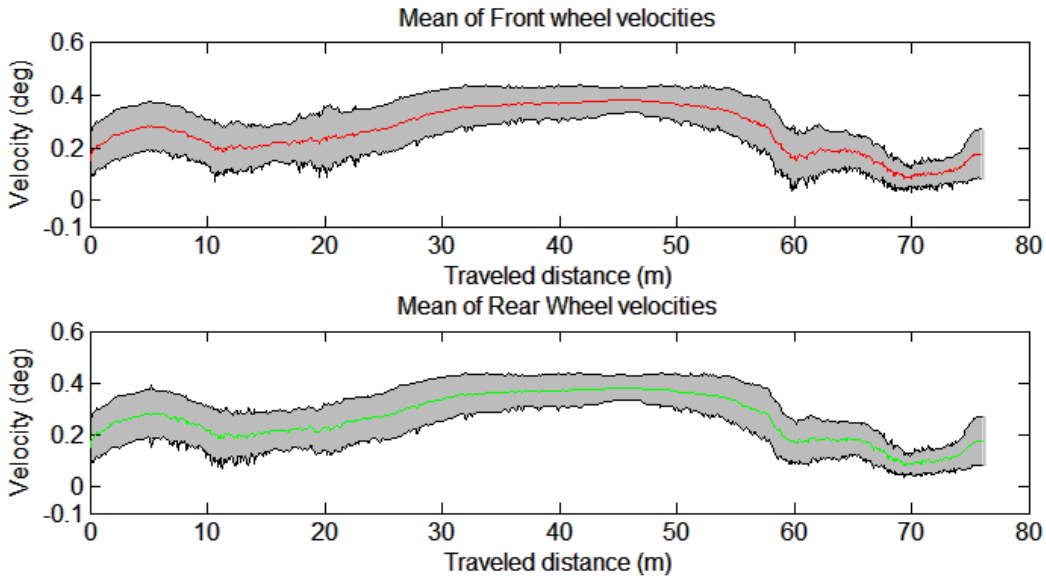

Figure 4.23: Average of the wheel velocities with JRD

docking port.

In Figure 4.20 is shown the average o the wheel velocities for JRD, in which can be seen that the velocities are almost identical for both wheels, which is expected since the driving device is managing both wheels at the same time. It has smoother transitions than Gamepad v.2, which is synonym of less motor strain, and lastly, the shaded area is narrower especially on the results for the front wheel, which has a large shaded area for Gamepad v.2.

4.4.5 Duration, energy and trajectory length

In Figure 4.24 is shown the average values of duration, length and energies for all users for each trial, looking at the duration of each trial it is noticeable that Gamepad v.2 durations get lower as the number of trials increase, the duration values for JRD are not that consistent, the first trial takes more time to complete than the others, but after that, no pattern is shown. The length is almost the same for Gamepad v.2 and JRD, in order to reach the end goal there is only one path, and the majority of the users followed it. The result for the energies is where the two devices diverge, in JRD case both wheels have similar values, and with Gamepad v.2 the rear wheel energy is remarkably higher than the front wheel. Driving the CTS like a car is the reason for this, since the front wheel is the wheel fixed in place and the rear wheel steers.

In Figure 4.25 is shown the average values of duration, length and energies for each user in all trials, in these can be seen the advantage that being experienced with gamepads can have when driving with Gamepad v.2, users 2, 4, 5, 6, 7, 11 and 12 are familiar with applications where gamepads are needed, and looking at their energy scores with the exception of 5 and 7 all have a good duration and energy results. User 5 has relative short trials but the energy spent on the rear wheel is extremely high, user 7 has the longest trials and the energies on both wheels are low, this user is very cautious and drives

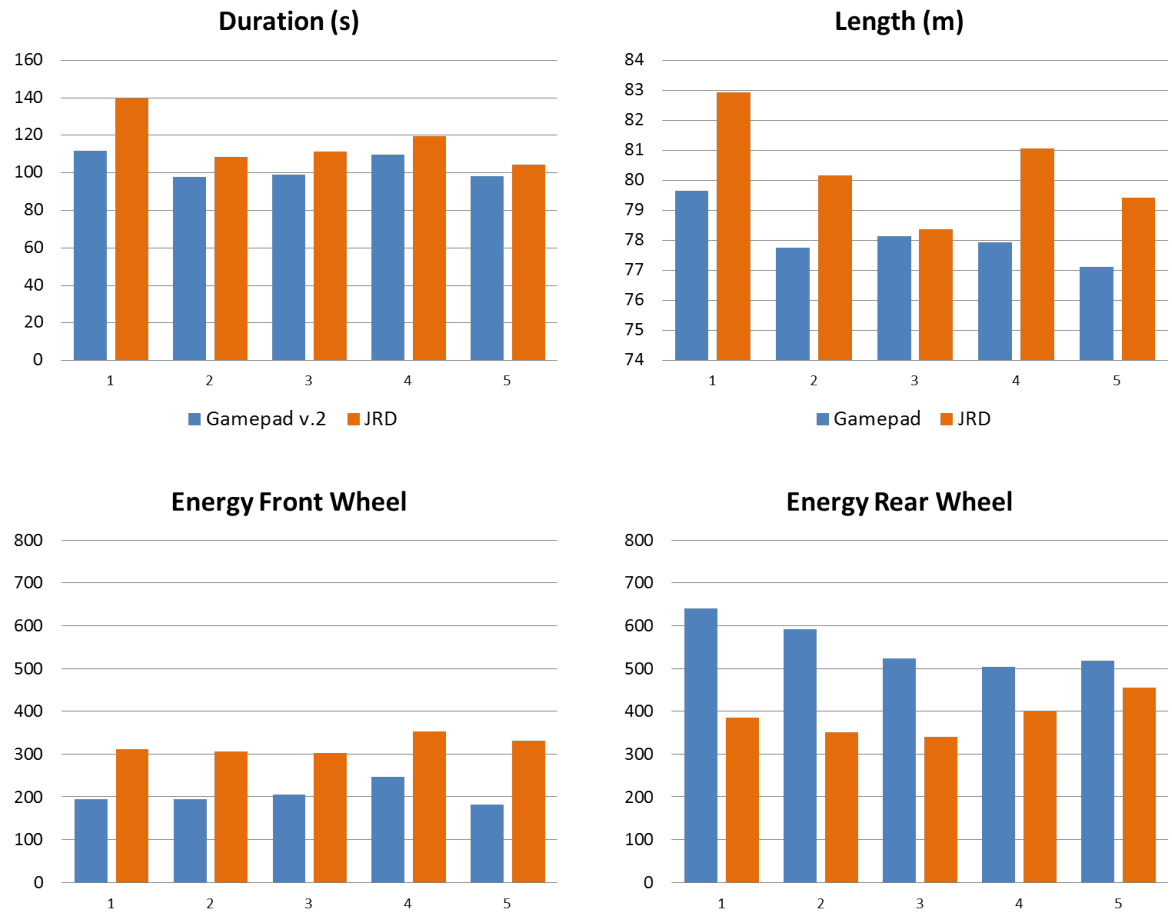

Figure 4.24: Duration, energy and length for the five trials

at very low speeds. The trajectory lengths are coherent with the duration and do not differ too much between users.

Table 4.2: Average values of duration, energy and trajectory length

	Duration (s)	Energy FW	Energy RW	Length (m)
Gamepad v.2	100,29	204,45	555,29	78,11
JRD	116,61	320,82	385,76	80,39

Table 4.2 summarizes the results of Figure 4.24.

The table with all the results and the individual graphs for each user can be seen in Appendix B.6.

4.5 User feedback

All users from the test groups agreed that driving the vehicle at its center with JRD was easier and more intuitive than driving each wheel with Gamepad v.2 and the total number of collisions with Gamepad v.2 corroborates this opinion. However 5 users said that with time and practice Gamepad v.2 should be better than JRD given the freedom that it provides.

The majority of the users focused mainly on the rear wheel of the vehicle when driving with Gamepad v.2, driving the vehicle mainly as a car or simply let the front wheel stay still. Only two users had

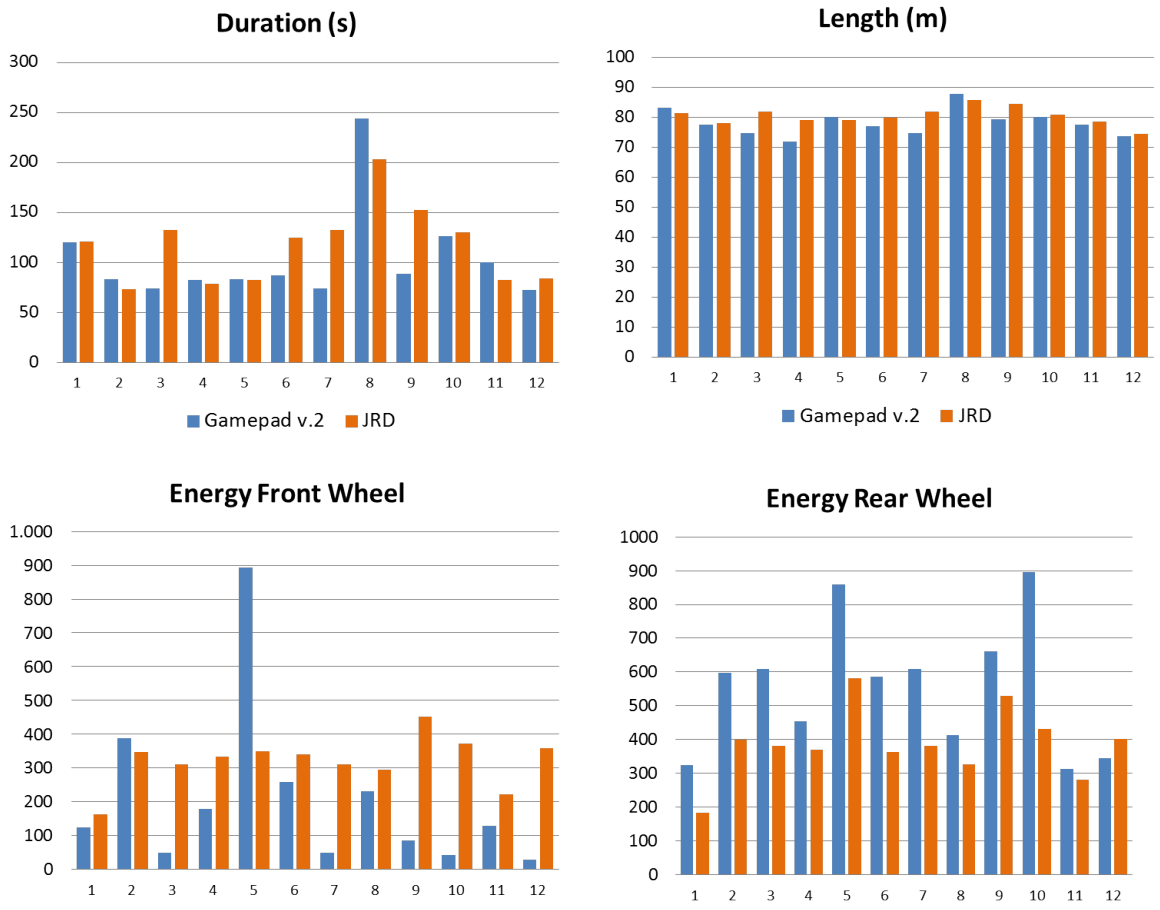

Figure 4.25: Duration, energy and length for the twelve users

successful trials controlling each wheel independently. This method to drive the vehicle was never used with JRD.

Some users expressed the need of having the velocity tune near the encoder instead of the built-in linear potentiometer on the bottom part of the Joystick, other suggested to use the buttons at the top of the Joystick's shaft to tune it.

Two users complained about the length of the Joystick and that it should be shorter. These users gripped the joystick at the basis instead of having the hand centered in the Joystick's shaft, for them it was not ergonomic.

Some users had difficulties understanding that the vehicle was being controlled taking into account the world frame and not the vehicle frame, this happened with the two younger subjects of the test groups and was also a problem with a student from Ciência Viva.

Two users from Group 1, insisted in driving the vehicle with Gamepad v.2 commanding both wheels at the same time, and after a large number of collisions they had to abandon this and were forced drive the vehicle like a car in order to finish the test.

4.6 Ciência Viva

During the duration this thesis, IPFN received the Ciência Viva program, where a group of young students, ages 14 to 17 years old, come to an university research center to perceive what is being done in a physics research lab and help them decide in what course to enroll.

One of the days was related to robotics and remote handling, so the students were asked to drive the vehicle in TES with the developed devices. All of them had experience with gamepads via console gaming, and were not familiar with joysticks neither rotational discs.

The tests done by them were not as structured as the trials given to the test groups, and the version that they tested was changed to have background music, sounds and a scoreboard to keep them interested. In Figure 4.26 can be seen a user driving the vehicle in the tutorial scenario, in the bottom right side of the projected image the scoreboard can be seen.

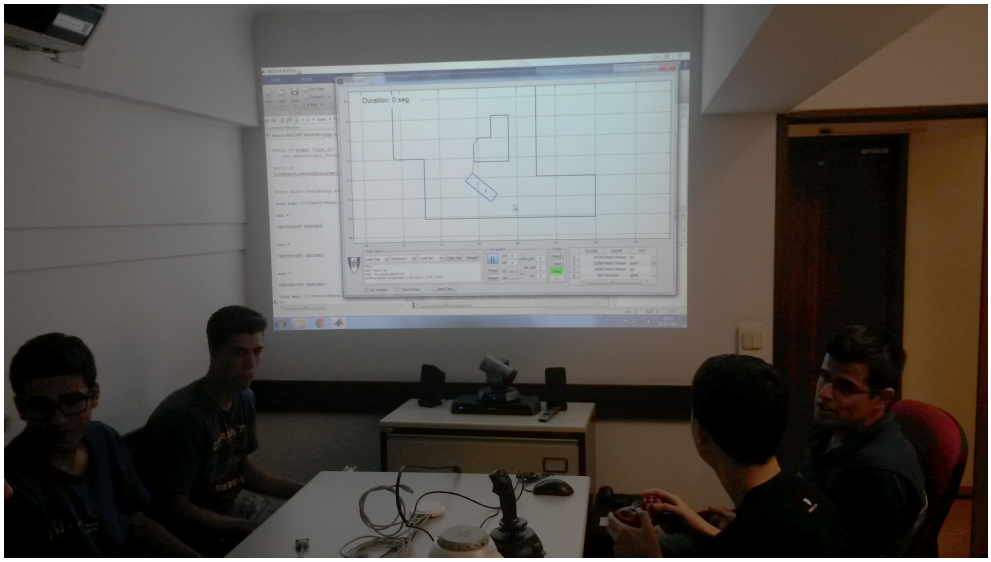

Figure 4.26: Tests with Ciência Viva students

The feedback they gave agreed with the feedback from the test groups, the vehicle is easier to drive with JRD for beginners, however due to their experience with video game consoles they also stated that with some experience they could achieve better ”scores” with Gamepad v.2.

4.7 Conclusion

Better usability of a device leads to less mistakes, and in narrow scenarios, there are no space for mistakes. Making a wrong turn can end up in a collision, damaging the vehicle and the environment. A common mistake while driving the vehicle at its wheels is switching the front for the rear wheel, this tends to happen especially when the vehicle is not complying with the gamepad's joystick position, in Figure 4.27 shows that situation. Assume that the vehicle has its heading as pictured in the top of the Figure and the desired movement is to place the front wheel down and the rear wheel right, however there are two cases in which the vehicle can be, with the same heading as the top image or turned 180

degrees, in the first situation using the gamepad inputs shown in the Figure 4.27 the output movement is indeed the desired one, however in the second situation, using the same inputs will lead in the opposite of the intended movement. This type of confusion will cause problems while driving and is the major flaw when driving the vehicle focusing on the wheels independently.

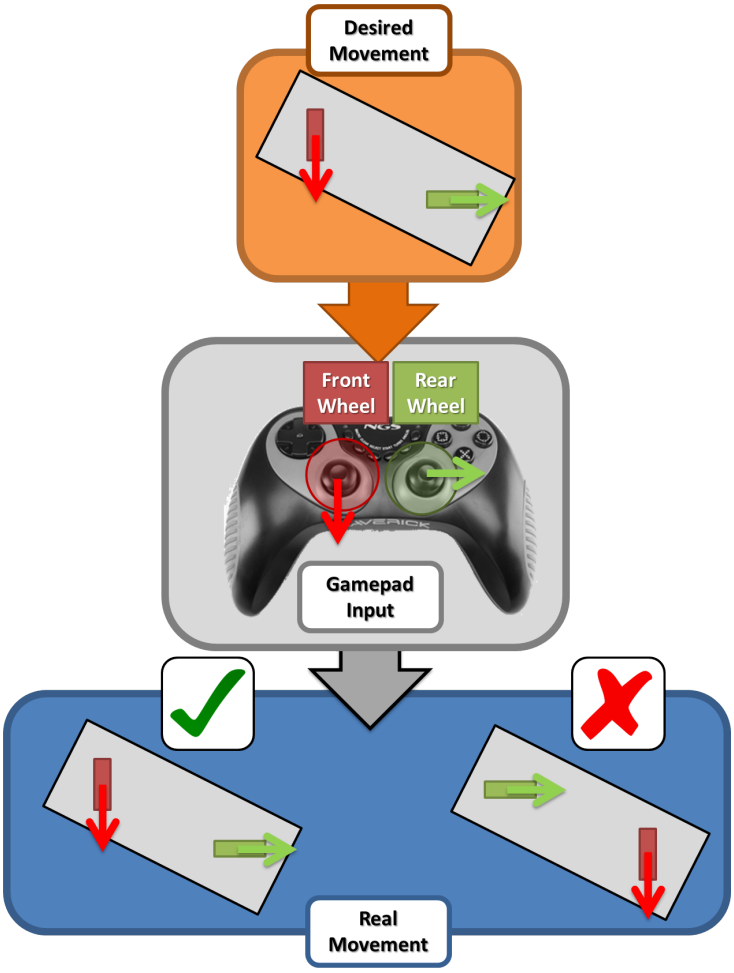

Figure 4.27: Problems associated with independent wheel control

This mistake was avoided by the by locking one of the vehicle wheels and driving the vehicle like a car, this is a solution that limits the movement freedom of the vehicle like was stated before. When the device being used is the JRD this problem ceases to exist, instead of two wheels the operator is focused in the vehicle center.

The tests with an experienced user revealed that Gamepad v.1 should not be used to drive rhombic-like vehicle in narrow environments like ITER tokamak building or the map used in scenario 2, the problem is not with the technology used in the device, but how it was implemented. The velocity and orientation commands are set to a value and do not return to zero unless they are manually set to zero, this leads to collisions and undesired motion.

The results obtained by the experienced user with Gamepad v.2 and JRD are very close, both devices are suited to drive the vehicle in narrow scenarios. The vehicle when driven with Gamepad v.2 exhibit more oscillations in the trajectories when compared with JRD especially in straight paths. As an

experienced user with both devices the author of this thesis still has problems driving the vehicle with Gamepad v.2 when needs to manage both wheels or needs to drive the vehicle in a new scenario.

The results obtained with the test groups revealed that for first time users with no previous experience JRD achieves better results than Gamepad v.2, less collisions overall, less energy spent with both wheels used uniformly. The feedback given by the users was very important because it can be used to improve the devices and point flaws in the way the tests were structured.

With the tests performed there are still a couple of questions that need to be answered: What happens in the long run? Will Gamepad v.2 surpass JRD with a certain amount of training? The freedom that Gamepad v.2 allows is really needed? To answer these question, five trials per user and a scenario with only one goal is not enough.

To wrap up, the tests concluded that JRD has a faster learning curve and is simpler to use than Gamepad v.2, generates slightly smoother trajectories and produces less strain in the vehicle motors.

Chapter 5

Conclusions and Future work

The thesis goal was to develop a better way to manually drive a rhombic-like vehicle, this means that the interface needed to be user friendly and to provide a safe way to drive the vehicle. Two methods are being evaluated, driving the vehicle being focused in its wheel or in its center. The initial hypothesis was that by driving the vehicle focused in its center would be easier than in its wheels. A device to drive the vehicle focusing in its wheel was already developed and available at the start of this thesis, the Gamepad v.1 but driving a rhombic-like vehicle with it was not adequate due to the high number of collisions the vehicle has while driving with it. The problems were not the device technology but how it was implemented, so a new version was made, Gamepad v.2, in which the way the vehicle is driven changed. Less buttons are used and the joystick's full capabilities are used, both orientations and velocities are sent via the joysticks. A device to focus on the vehicle center uses a 2-axis joystick and a rotational disc (JRD). Joystick commands the center velocity (value and orientation), while the rotational disc commands the vehicle angular velocity. It needed to be custom built, and relies on an encoder to command the vehicle angular velocity.

When using both gamepad devices to drive the vehicle, the user sends commands to each wheel, both gamepad's joysticks are assigned to a different wheel. With JRD the user sends inputs to the vehicle's center via the joystick and encoder, the system then transforms them in front and rear wheel inputs. The joystick allows for crab-like movements (without rotation), and the vehicle heading is commanded by the encoder, this method limits the motion freedom since it is not possible to drive each wheel independently, however, the vehicle continues to be omni-directional.

To convert the vehicle's center inputs into front and rear wheel variables, three methods were evaluated, a simple method that split the motion in pure translation and rotation and an averaged sum of bot, a method that relies on an inversion of a first degree Taylor approximation of the kinematic system, and lastly a method developed by Alonzo Kelly, a forward rate kinematic system. The test to chose the most suited method was done by simulating their response to a set of inputs emulating an operator. The method with best results was the A. Kelly method, returning smoother curves, without abrupt changes and deviations from the expected heading behavior.

The reason to have an 3D printed support for the encoder is to offer the people from test groups an

ergonomic device. The previous iteration had a cardboard support for the encoder with a potentiometer cap for turning. S disc to turn the encoder shaft and a support for its body were designed in Solidworks (CAD) and exported to Cura (CAM) in order to be 3D printed. The support increases encoder stability and with the disc, more precise commands could be sent to change the vehicle heading. Lastly, it gives a finishing touch to the device.

To evaluate both methods all three devices were considered, Gamepad v.1, Gamepad v.2 and JRD. Tests were needed in order to evaluate the ability of each device to drive the vehicle safely, the tests consisted in following a trajectory, traveling from one point to another while avoiding collisions, keeping a maximum safety distance to the walls and smooth trajectory. In the first stage, the tests were conducted by an experienced user, the author of this thesis. Which concluded that the worst device to drive the vehicle is Gamepad v.1, it is the device with more collisions, less stability and deviates from the goal objectives. Between Gamepad v.2 and JRD, the score is even, no collisions, similar test duration and length, JRD produces slightly smoother trajectories, both devices are good candidates to drive the vehicle. Because of these results, Gamepad v.1 will not be used or evaluated in test groups.

The usability tests groups consisted in a total of twelve individuals that tested each device, they were split in two groups of six. The first group started the test with Gamepad v.2 and the other with JRD, a total of 200 trials were done, 120 successful (without collisions). The first group had more collisions than the second because they had to learn how to navigate the map with Gamepad v.2, which can be hard for inexperienced users. The vast majority of collisions happened while driving with Gamepad v.2. Many users with Gamepad v.2 needed to use the feature to drive the vehicle like a car or simply let the front wheel unattended, reducing the control variables and making the vehicle easier to drive. With JRD no user needed to use the same feature. By driving the vehicle like a car the energy spent in the rear wheel is much higher than in the front wheel, which will cause extra stress in that wheel motors. The results proved that it is easier and safer to drive the vehicle with JRD than Gamepad v.2, meaning that driving the vehicle at its center is also easier and safer than at each wheel. The users feedback also reinforced this statement, as the vast majority agreed that it was easier to drive with JRD, however they also stated that Gamepad v.2 can achieve better results if used by an experienced user.

During the test period, students from Ciência Viva also tested the devices, which lead to the creation of a funnier version of the simulator, with background music, action sounds and a scoreboard. Their feedback was the same as the test groups, ”easier to drive with JRD, but with time and experience maybe Gamepad v.2 would be better”. Keep in mind these students are experienced with gamepads. The scoreboard had a negative impact on the tests because users shows interest in achieving the higher score, which lead to careless driving and high speeds, which resulted in more collisions.

Opinions from outside people also helped to point what could be better with the device and interface, for instance most people when driving the vehicle with JRD instead of continuously turning the disc, did little increments, this way of changing the vehicle heading conflicted with the data acquisition script in Arduino that was developed with continuous motion in mind. If the user is turning the disc with little touches sometimes the script does not recognize it as movement and the angular velocity remains zero, the acquisition script should be improved to support this type of movements. Other common suggestion

was, to have the maximum velocity controlled at the joystick buttons instead of the bottom potentiometer, or place it close to the rotational disc. Other users complained about the joystick's shaft length and suggested a reduction, or use a shorter joystick. Regarding possible improvements to test procedures, instead of a free trajectory from point A to B. The goal of following an optimal trajectory generated in TES could be better because it would be easier to compare the results with an optimal solution, and then study what could be improved in the way people drive the vehicle.

Lastly, with all the previous information, the conclusion can be summarized in the following: Gamepad v.1 is obsolete and JRD is better than Gamepad v.2 to drive rhombic-like vehicles. Hence driving the vehicle focused at its center is better than directly at each wheel. However the real device should have both methods to drive the vehicle, at its center and at its wheels with the option of commuting between the two modes.

Regarding future work, more tests can be made in order to obtain more data and reinforce or discredit JRD as the best method to drive the vehicle.

A test where the user needs to follow a path in a complex scenario without deviating from it. The trajectory outcome is expected to have more oscillations than without a path to follow, as the user is expected to make small adjustments to his driving while following the path. From the obtained data in this thesis, Gamepad v.2 is expected to have worse performance in such test.

A test in scenario 3 where instead of the same position for the end goal, in each trial its position is changed. This should take away the muscle memory factor and make the user think in new approaches to reach the goal, this test should be taken by people with a certain amount of experience with each device, to prevent beginner mistakes like clashing at the trial beginning. Because of the motion limitations JRD is expected to have worse performance, with tests taking more time to complete.

Besides tests on the developed interface, the devices can be improved with the user suggestions redesign the encoder support to have the max velocity control near the disc, or add more features to help driving. The interface can be improved by rendering a 3D view of the environment besides the 2D top side view. This can lead to another type research, what is the best viewing system to drive the vehicle, third or first person views. The goal is to test each device in first person view, analyze the results and decide which is most suited with this viewing system. Or with the 3D environment built, drive the vehicle in a virtual reality scenario with an HMD.

In order to simulate a real operation, build a vehicle prototype with the same rhombic-like configuration and equip it with cameras and one or two laser range finders, and use it to simulate a remote handling operation. The vehicle and the operator are placed in separate rooms and the information of vehicle location is given by sensors placed on the vehicle and/or environment.

Bibliography

[1] Alberto Vale and Isabel Ribeiro. Motion planning of large scale vehicles for remote material transportation. In *Motion and Operation Planning of Robotic Systems*, pages 249–292. Springer, 2015.

[2] Directorate-General for Research. Fusion research: An energy option for europe's future, 2007.

[3] Terry Kammash. Fusion reactor physics: principles and technology. *IEEE Transactions in Nuclear Science*, 1975.

[4] European Comission. Fusion: A brief history. `http://ec.europa.eu/research/energy/euratom/index_en.cfm?pg=fusion&section=history`, 2013.

[5] T Hamacher and AM Bradshaw. Fusion as a future power source: recent achievements and prospects. In *18th World Energy Congress, Buenos Aires, World Energy Council, London*, 2001.

[6] D Mandal, MRK Shenoi, and SK Ghosh. Synthesis & fabrication of lithium-titanate pebbles for iter breeding blanket by solid state reaction & spherodization. *Fusion Engineering and Design*, 85(5):819–823, 2010.

[7] Alfredo Pironti and Michael Walker. Control of tokamak plasmas: introduction to a special section. *Control Systems, IEEE*, 25(5):24–29, 2005.

[8] Isabel Ribeiro, Carlo Damiani, Alessandro Tesini, Satoshi Kakudate, Mikko Siuko, and Carlo Neri. The remote handling systems for iter. *Fusion Engineering and Design*, 86(6):471–477, 2011.

[9] C González Gutiérrez, C Damiani, M Irving, J-P Friconneau, A Tesini, I Ribeiro, and A Vale. Iter transfer cask system: status of design, issues and future developments. *Fusion Engineering and Design*, 85(10):2295–2299, 2010.

[10] J Ferreira, A Vale, and R Ventura. Optimizing range finder sensor network coverage in indoor environment. *IFAC Intelligent Autonomous Vehicles (IAV 2010)*, 2010.

[11] Frank L Lewis, Chaouki T Abdallah, and Darren M Dawson. *Control of robot manipulators*, volume 236. Macmillan New York, 1993.

[12] Daniel N Lapedes. *McGraw-Hill dictionary of scientific and technical terms*. McGraw-Hill Book Co., New York, NY, 1978.

[13] V Daniel Hunt. *Industrial robotics handbook*. Industrial Press Inc., 1983.

[14] Stuart Shepherd and Alois Buchstab. Kuka robots on-site. In *Robotic Fabrication in Architecture, Art and Design 2014*, pages 373–380. Springer, 2014.

[15] Naval Research Laboratory. A timeline of nrl's autonomous systems research, 2011.

[16] Jun Kato, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. Multi-touch interface for controlling multiple mobile robots. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3443–3448. ACM, 2009.

[17] Terrence Fong, Charles Thorpe, and Betty Glass. Pdadriver: A handheld system for remote driving. In *IEEE international conference on advanced robotics*, number LSRO2-CONF-2003-004, 2003.

[18] Daniel Bug. Oculus rift control fo a mobile robot. Master's thesis, KTH, School of Computer Science and Communication (CSC), 2014.

[19] Tomas Kot and Petr Novák. Utilization of the oculus rift hmd in mobile robot teleoperation. In *Applied Mechanics and Materials*, volume 555, pages 199–208. Trans Tech Publ, 2014.

[20] Jason D Moss and Eric R Muth. Characteristics of head-mounted displays and their effects on simulator sickness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(3):308–319, 2011.

[21] Joao Bivar and Alberto Vale. Behavior of digital and analog controller devices for manual driving of rhombic like vehicles. *2011 19th Mediterranean Conference on Control & Automation (MED)*, Jun 2011.

[22] Alberto Vale, Daniel Fonte, Filipe Valente, and Isabel Ribeiro. Trajectory optimization for autonomous mobile robots in iter. *Robotics and Autonomous Systems*, 62(6):871–888, 2014.

[23] Elwood Russell Johnston. *Vector mechanics for engineers: statics and dynamics*, volume 1. Tata McGraw-Hill Education, 2009.

[24] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] Danwei Wang and Feng Qi. Trajectory planning for a four-wheel-steering vehicle. In *ICRA*, pages 3320–3325, 2001.

[26] KN Spentzas, I Alkhazali, and M Demic. Kinematics of four-wheel-steering vehicles. *Forschung im Ingenieurwesen*, 66(5):211–216, 2001.

[27] Lund halsey website. `http://www.lundhalsey.com/`.

[28] Paul Dienes. *The Taylor series. An introduction to the theory of functions of a complex variable*. Oxford, 1931.

[29] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.

[30] Alonzo Kelly. A vector algebra formulation of kinematics of wheeled mobile robots. Technical report, Carnegie Mellon University, 2010.

[31] Alberto Vale, Ricardo Oliveira, and Rodrigo Ventura. Driving and steering wheel system for omni-directional vehicles, 2014.

Appendix A

Arduino script for data acquisition from the encoder

Following is the script written to acquire the angular velocity from the encoder.

```
[fontsize=\small]

#include <Average.h>
#include <math.h>

#define alfa 0.0000001
#define vel_tune 0.000001
#define tau 0.00000000001
#define n_samples 150

 unsigned long start, finish, elapsed;
 float vel, vel_mov, av_vel;
 float array_vel[n_samples];
 int last_pos, cnt;
 int encoder0PinA = 8;
 int encoder0PinB = 9;
 int encoder0Pos = 0;
 int encoder0PinALast = LOW;
 int n = LOW;
 boolean mov, positive;

 void setup() {
   pinMode (encoder0PinA,INPUT);
   pinMode (encoder0PinB,INPUT);
   Serial.begin (115200);
   start = micros();
   finish = 0;
   last_pos = 0;
   vel = 0;
   cnt = 0;
 }

 void loop() {


   n = digitalRead(encoder0PinA);
   if ((encoder0PinALast == LOW) && (n == HIGH)) {
     if (digitalRead(encoder0PinB) == LOW) {
       encoder0Pos--;
       mov = true;
       positive = false;
     } else {
       encoder0Pos++;
       mov = true;
       positive = true;
     }
```

```
   if(encoder0Pos>32767||encoder0Pos<-32767){
   encoder0Pos = 0;
    }
   }
   encoder0PinALast = n;



finish = micros();
elapsed = finish-start;

if(abs(elapsed<1000)){
  elapsed = 1000;
}


if(mov==true){
elapsed = abs(finish-start);
start = micros();
if(positive){
vel_mov = -(encoder0Pos-last_pos)/(elapsed);}
vel = vel+(1/(1+(1/(elapsed*tau))))*vel_mov;
} else{
  vel_mov = (encoder0Pos-last_pos)/(elapsed);
  vel = vel-(1/(1+(1/(elapsed*tau))))*vel_mov;
```

```
} else{
 vel = vel/exp(alfa*abs(elapsed));
 if (abs(vel) < 0.001){
  vel = 0;
 }
 //Serial.print("LOWSPEED: ");
 //Serial.println(vel);
}
mov = false;
last_pos = encoder0Pos;

array_vel[cnt] = vel;
cnt++;
if (cnt==n_samples){
   av_vel = mean(array_vel,n_samples);
   delay(1);
   cnt = 0;

   Serial.println(av_vel);

 }
```

Appendix B

Tests groups results

B.1 Vehicle position

In Figure B.1 are all the generated trajectories that resulted from the test groups. Gamepad v.2 on the left and JRD on the right.

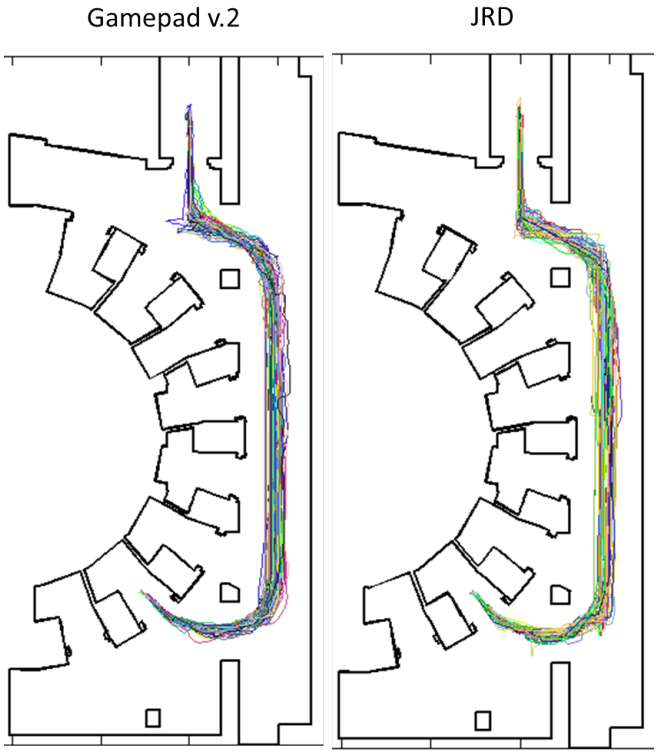

Figure B.1: All generated trajectories from the test groups

B.2 Individual results - Safety distance

In Figures B.2 to B.13 are shown the average distance to the nearest object for all users. On the left are the results for Gamepad v.2 and on the right for JRD. The blue curve are the average values and the

black one is the standard deviation.

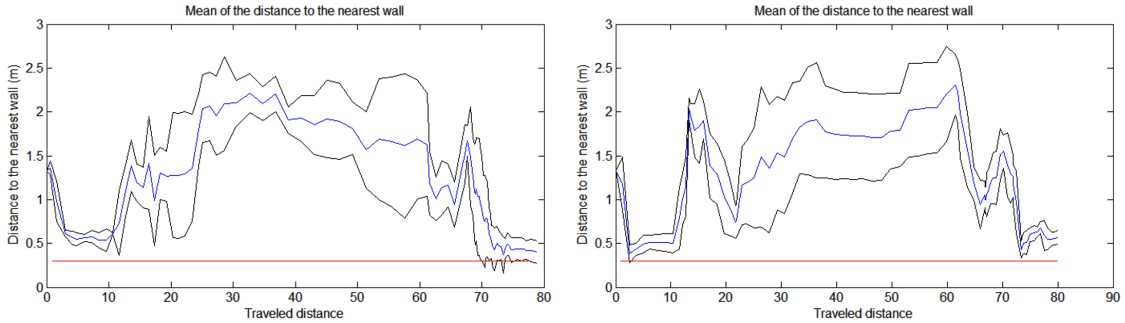

Figure B.2: Distance to the nearest wall of user 1

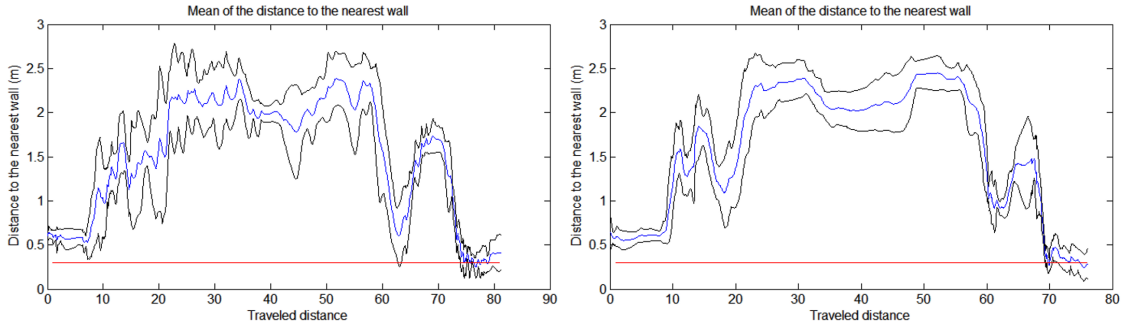

Figure B.3: Distance to the nearest wall of user 2

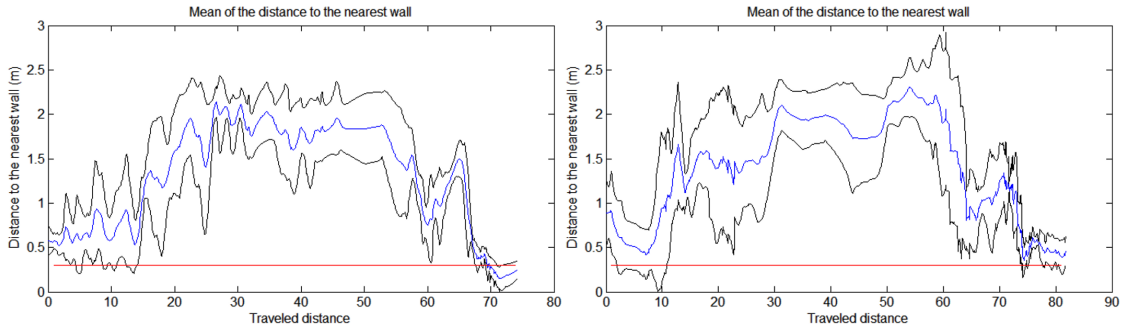

Figure B.4: Distance to the nearest wall of user 3

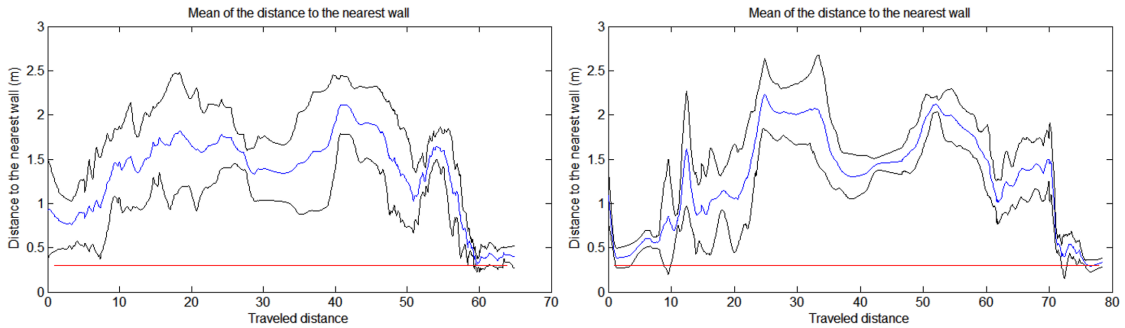

Figure B.5: Distance to the nearest wall of user 4

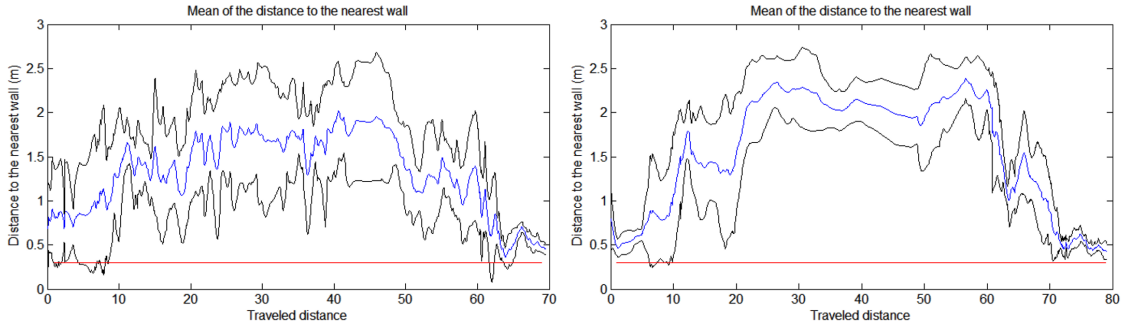

Figure B.6: Distance to the nearest wall of user 5

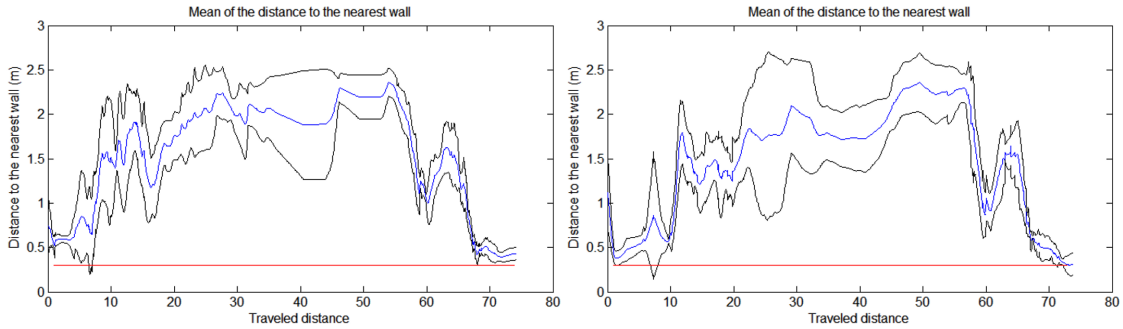

Figure B.7: Distance to the nearest wall of user 6

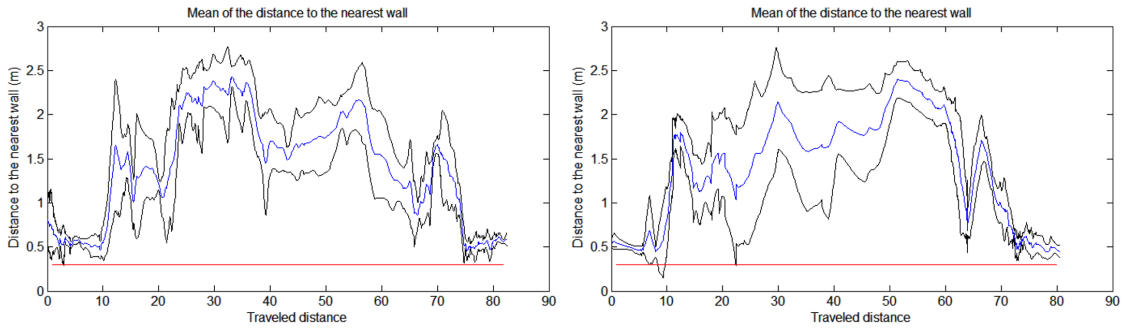

Figure B.8: Distance to the nearest wall of user 7

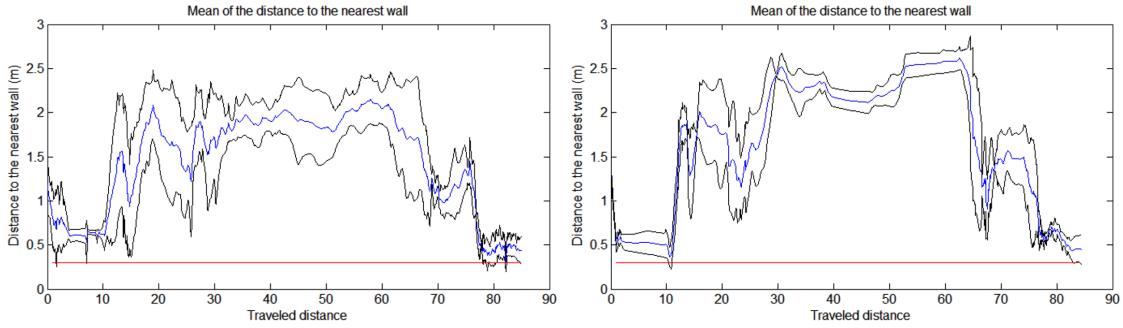

Figure B.9: Distance to the nearest wall of user 8

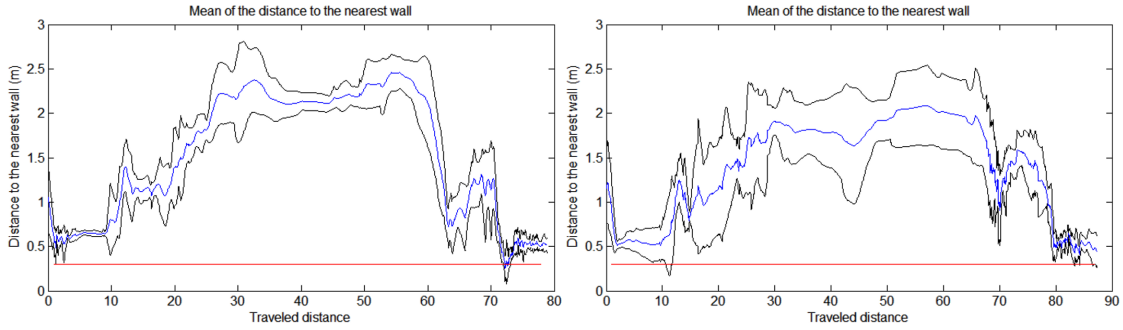

Figure B.10: Distance to the nearest wall of user 9

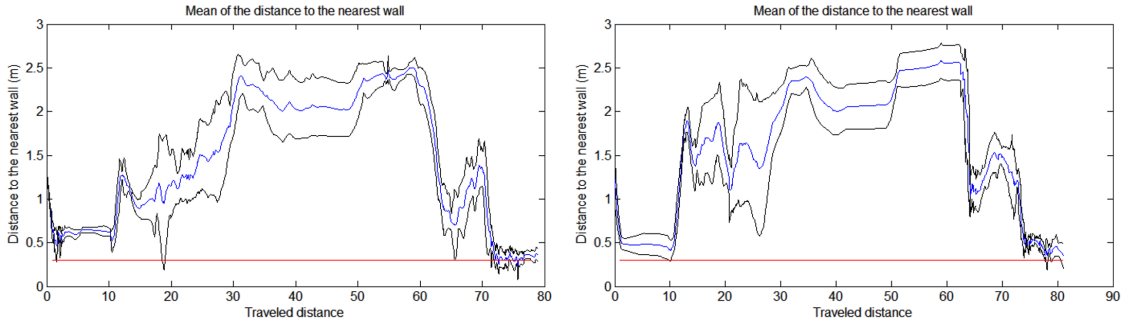

Figure B.11: Distance to the nearest wall of user 10

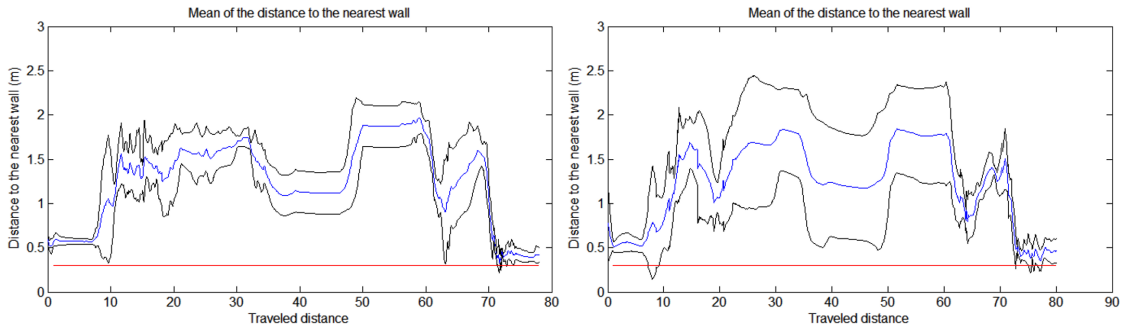

Figure B.12: Distance to the nearest wall of user 11

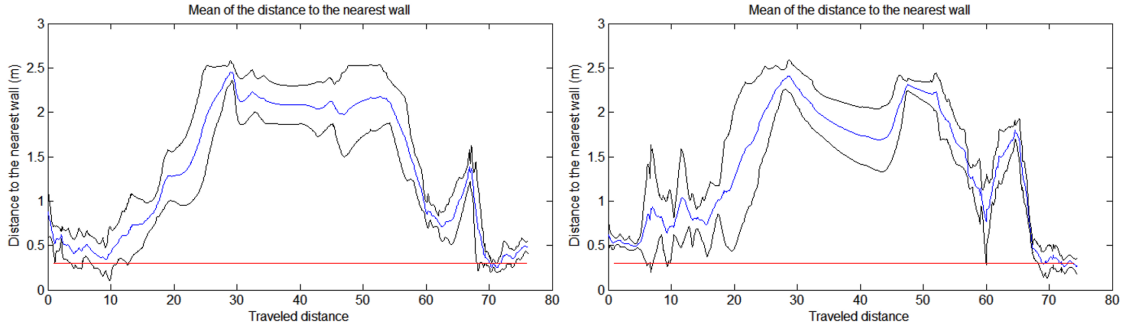

Figure B.13: Distance to the nearest wall of user 12

B.3 Wheel orientations

From Figures B.14 to B.25 are shown the average wheel orientations for all users. On the left are the results for Gamepad v.2 and on the right for JRD. The blue curve are the average values and the black one is the standard deviation.

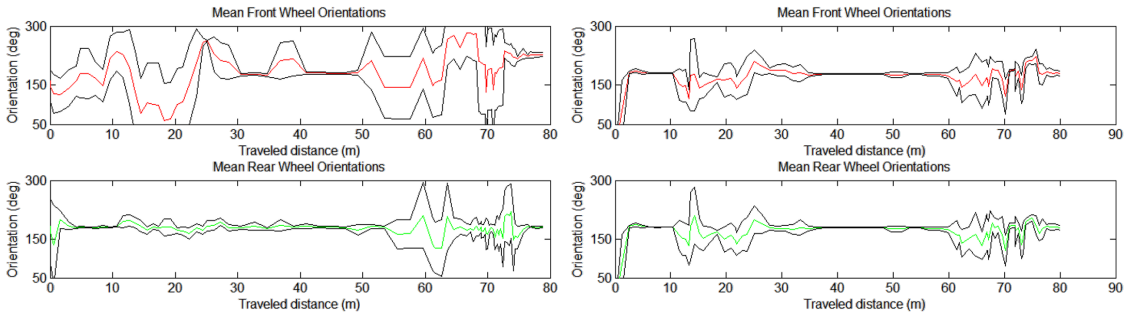

Figure B.14: Wheel orientations of user 1

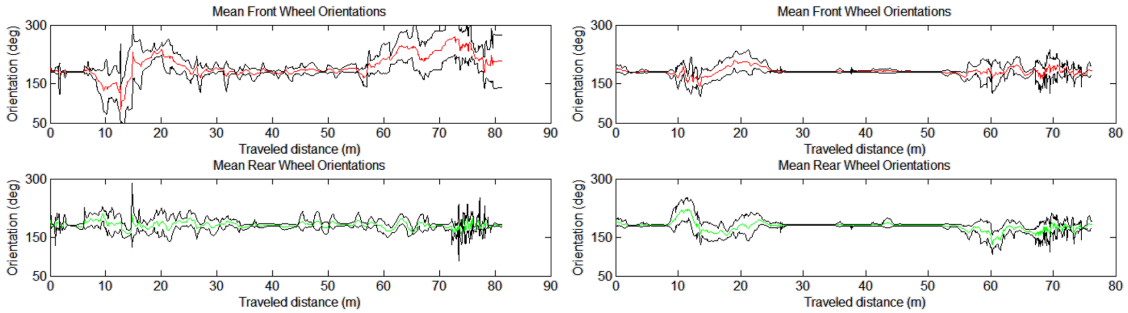

Figure B.15: Wheel orientations of user 2

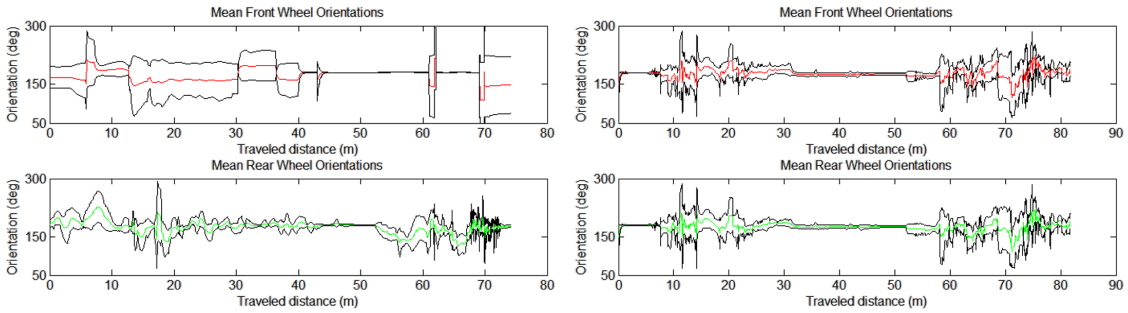

Figure B.16: Wheel orientations of user 3

B.4 Vehicle heading

From Figures B.26 to B.37 are shown the average vehicle heading for all users. On the left are the results for Gamepad v.2 and on the right for JRD. The blue curve are the average values and the black one is the standard deviation.

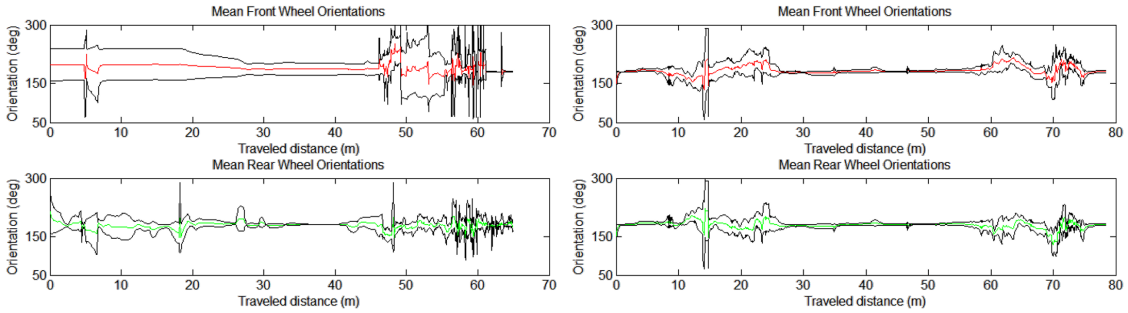

Figure B.17: Wheel orientations of user 4

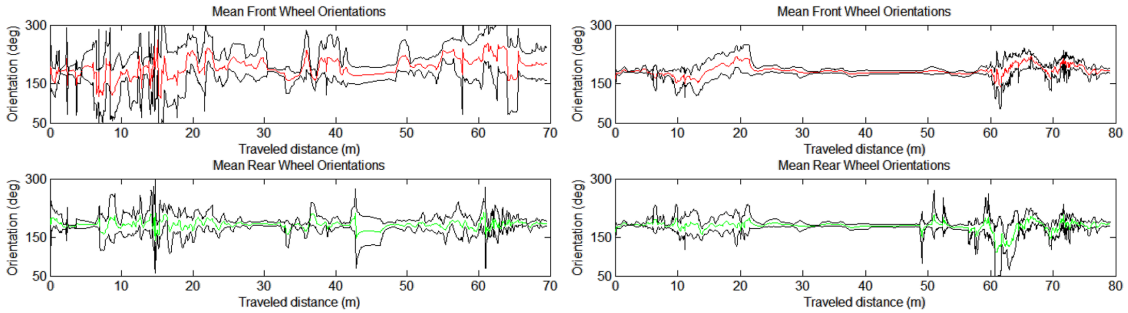

Figure B.18: Wheel orientations of user 5

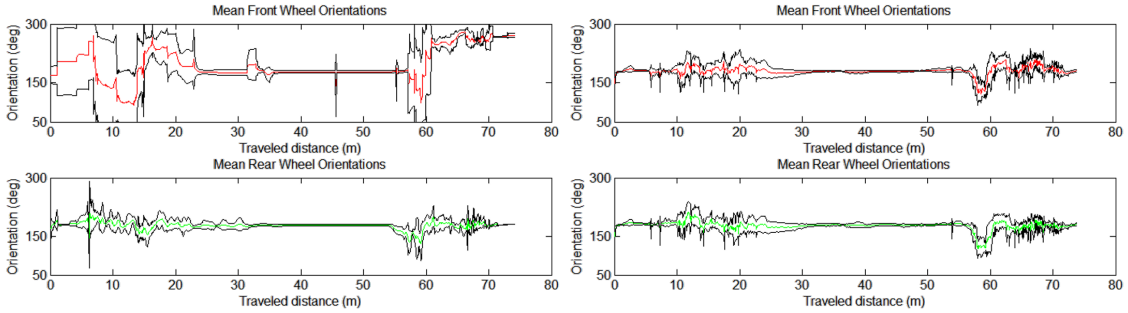

Figure B.19: Wheel orientations of user 6

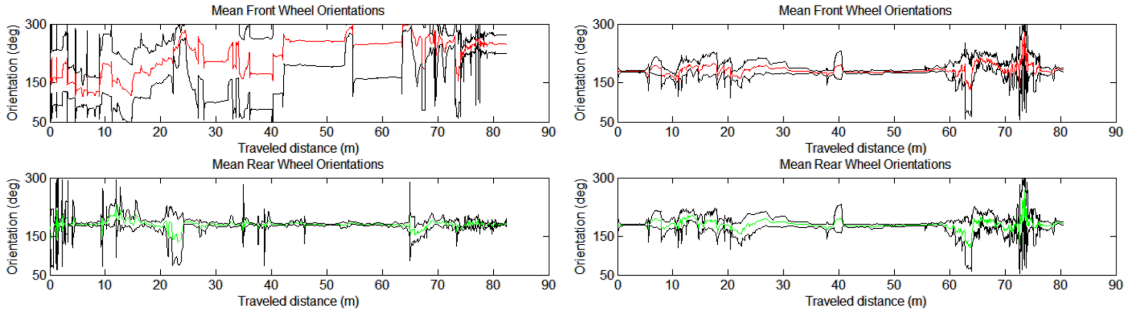

Figure B.20: Wheel orientations of user 7

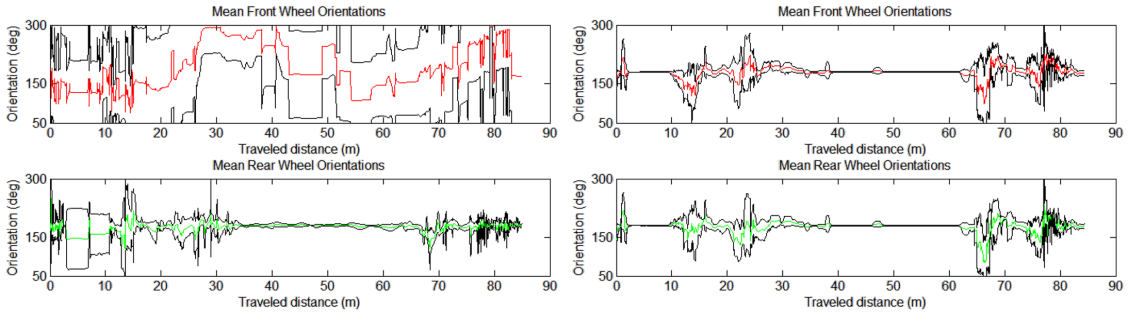

Figure B.21: Wheel orientations of user 8

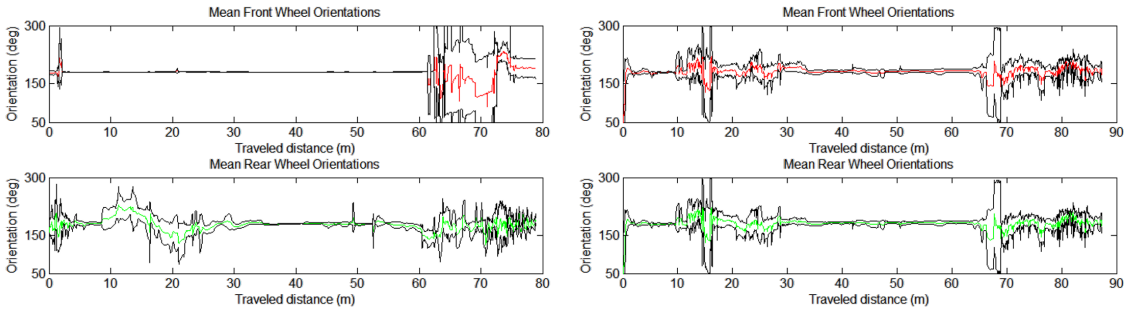

Figure B.22: Wheel orientations of user 9

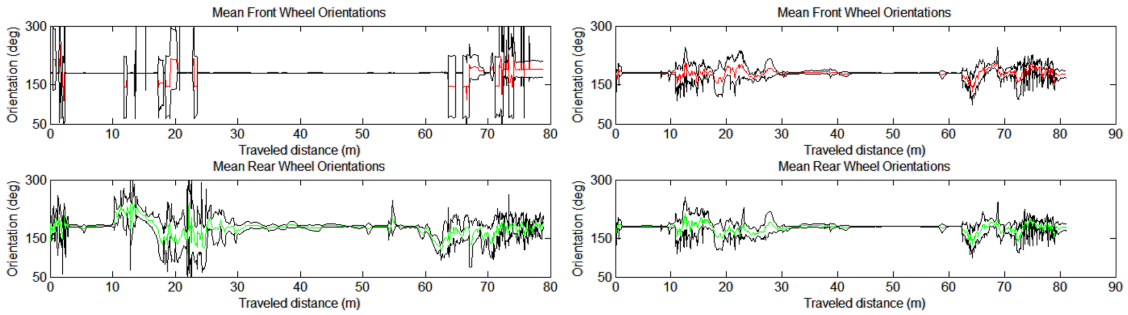

Figure B.23: Wheel orientations of user 10

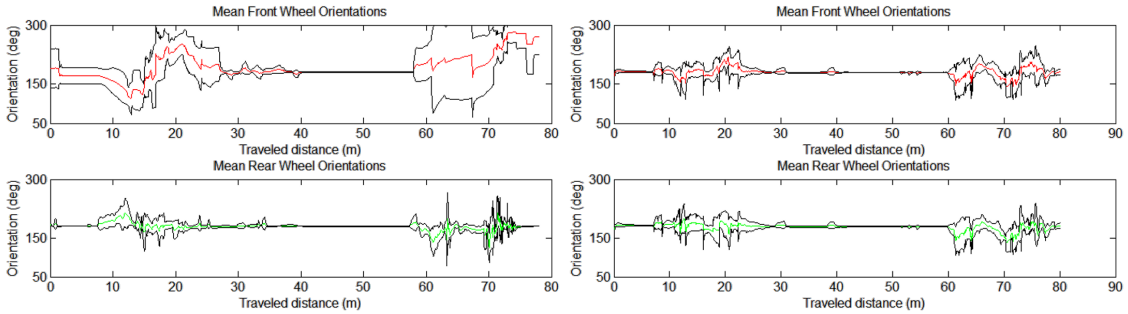

Figure B.24: Wheel orientations of user 11

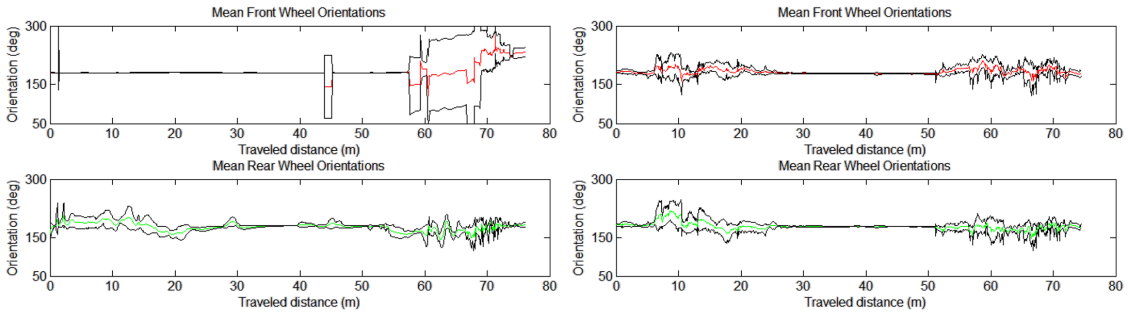

Figure B.25: Wheel orientations of user 12

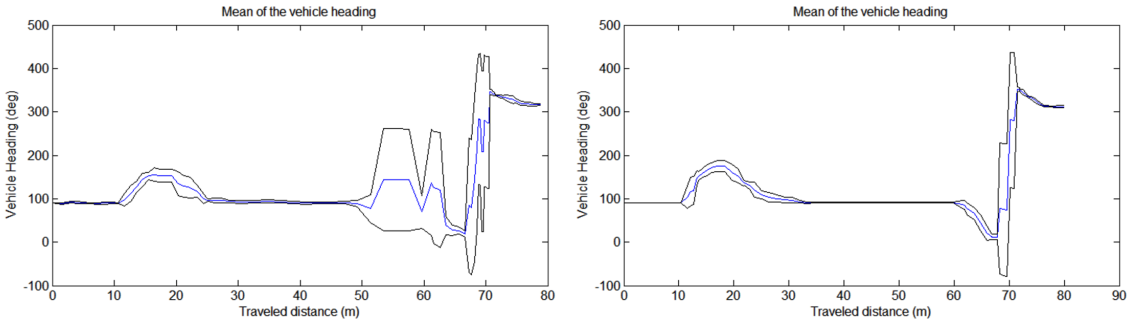

Figure B.26: Vehicle Heading for user 1

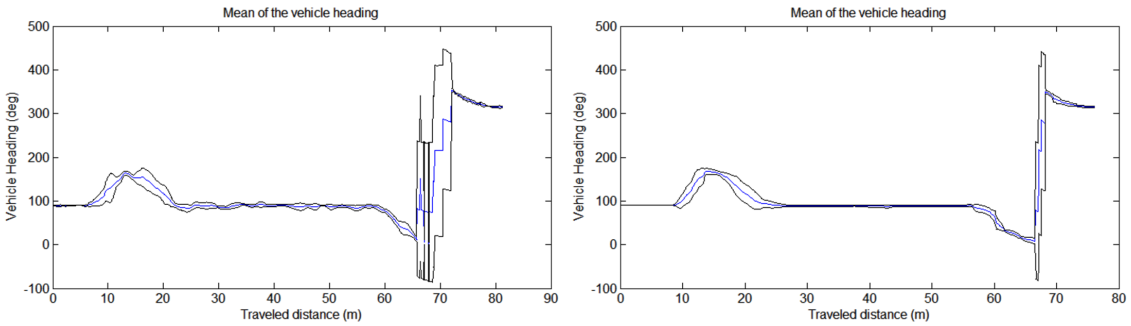

Figure B.27: Vehicle Heading for user 2

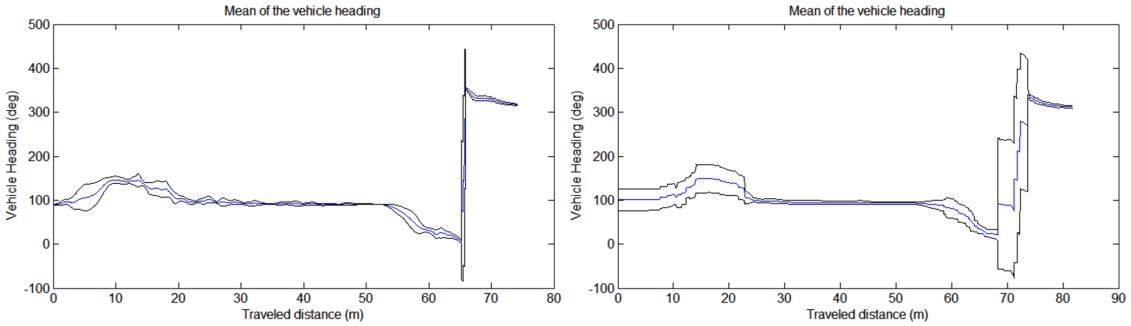

Figure B.28: Vehicle Heading for user 3

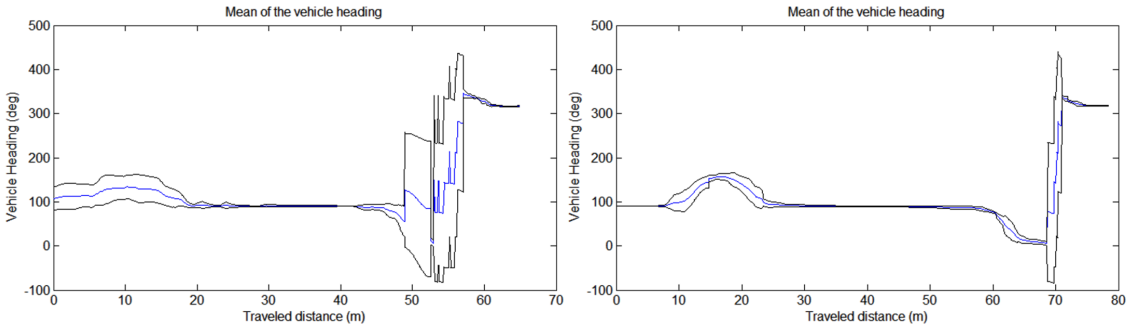

Figure B.29: Vehicle Heading for user 4

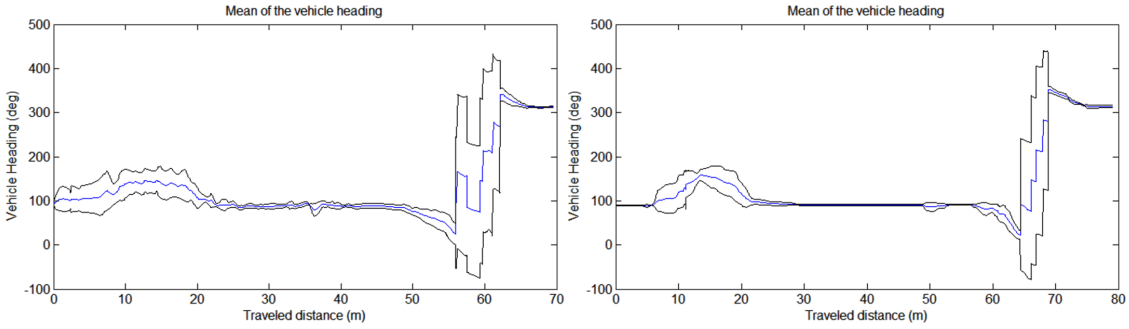

Figure B.30: Vehicle Heading for user 5

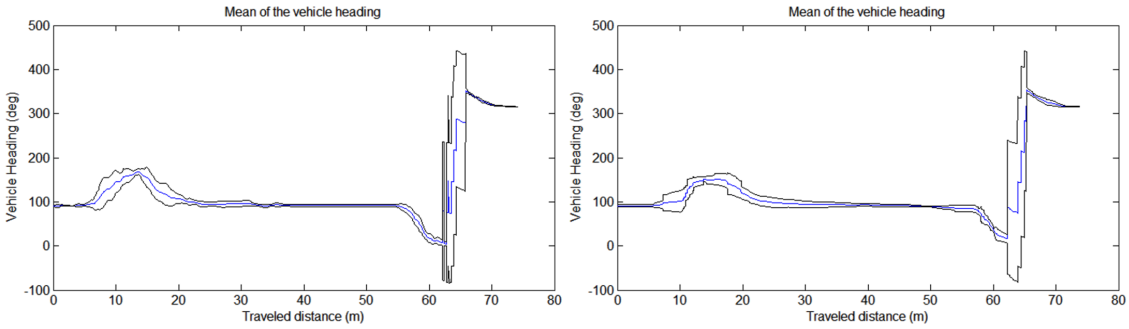

Figure B.31: Vehicle Heading for user 6

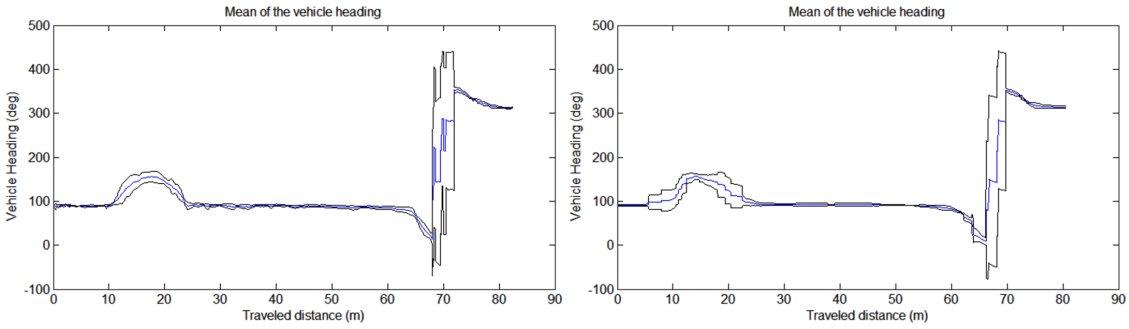

Figure B.32: Vehicle Heading for user 7

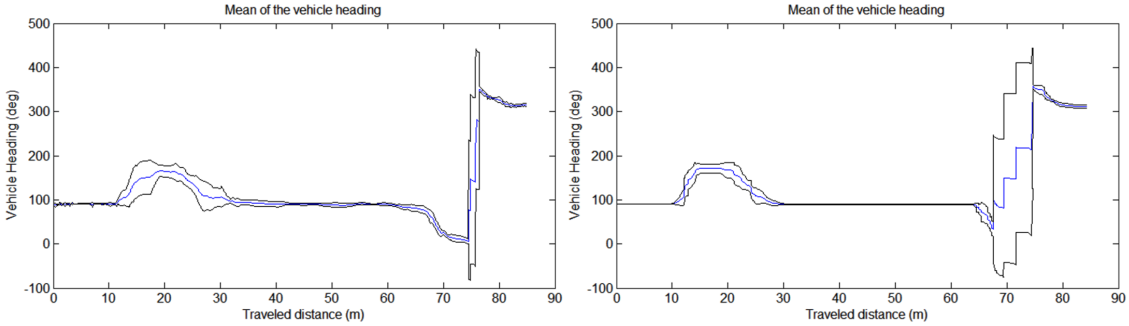

Figure B.33: Vehicle Heading for user 8

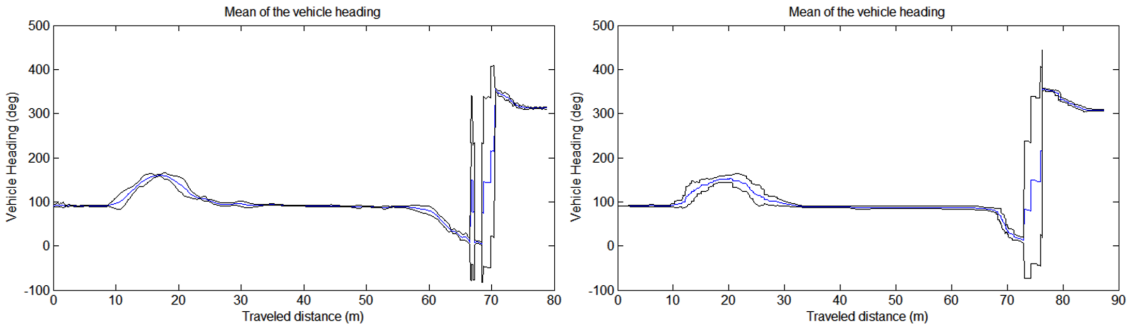

Figure B.34: Vehicle Heading for user 9

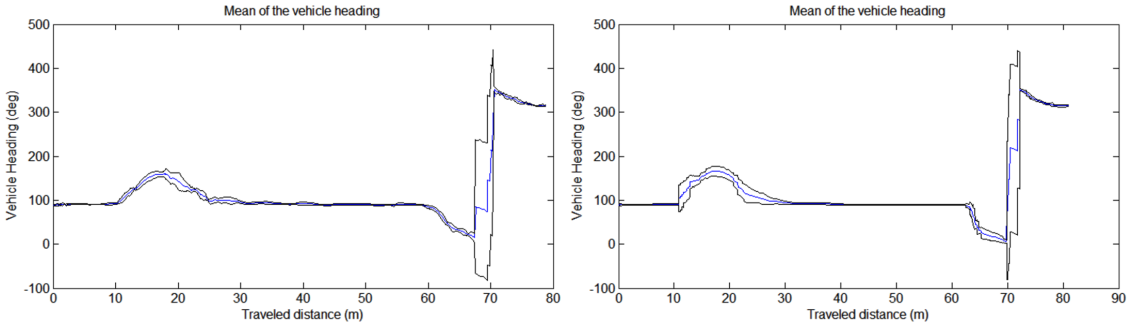

Figure B.35: Vehicle Heading for user 10

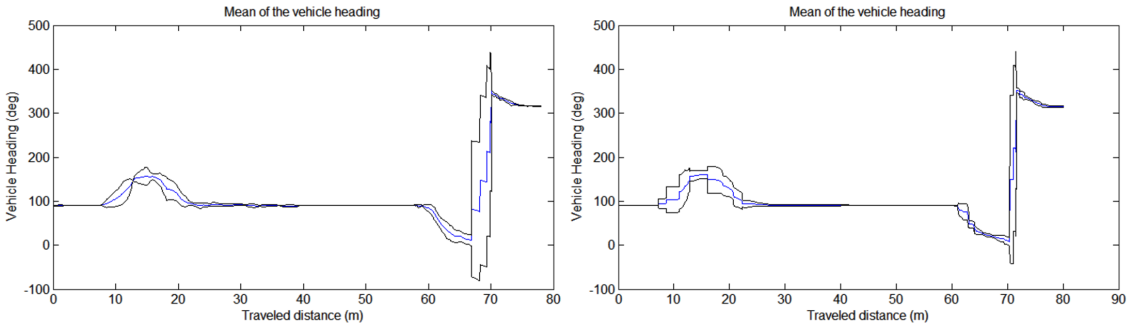

Figure B.36: Vehicle Heading for user 11

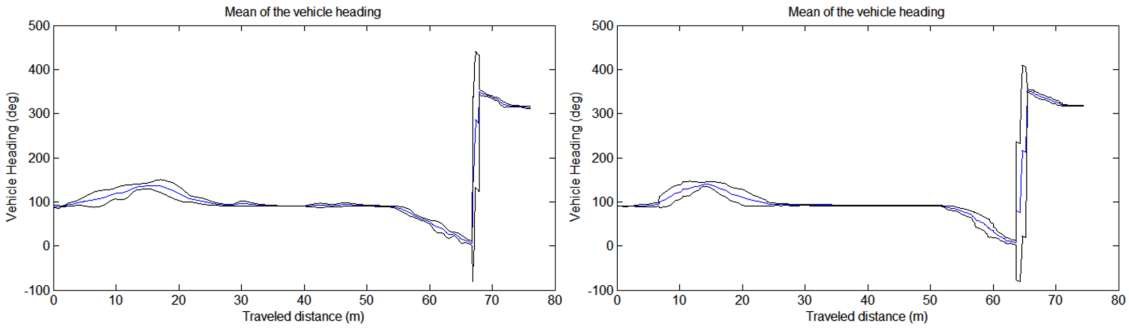

Figure B.37: Vehicle Heading for user 12

B.5 Wheel velocities

From Figures B.38 to B.49 are shown the average wheel velocities for all users. On the left are the results for Gamepad v.2 and on the right for JRD. The blue curve are the average values and the black one is the standard deviation.

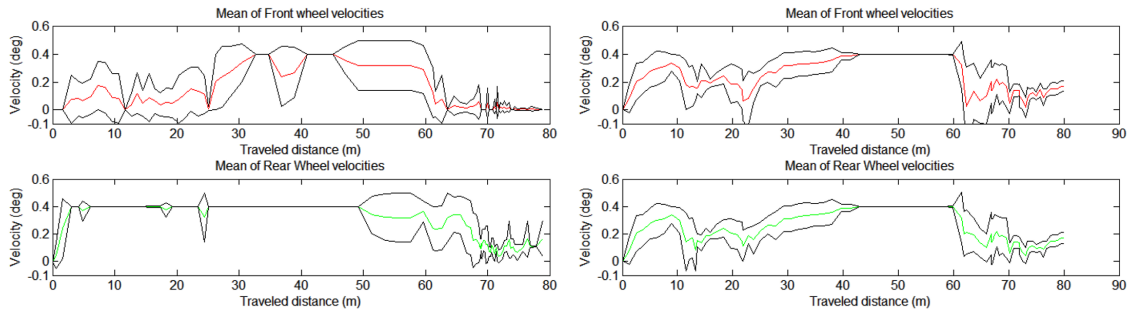

Figure B.38: Wheel velocities of user 1

B.6 Individual results - Duration, energy and trajectory length

In Table B.1 are shown the average values for the duration, energy and trajectory length for each trial of all users, each set of five trials correspond to a different user, the first six sets correspond to test group 1 and the rest to group 2.

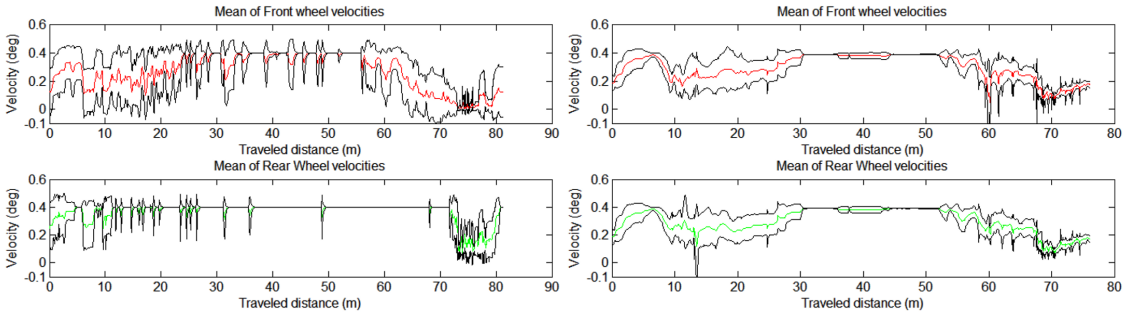

Figure B.39: Wheel velocities of user 2

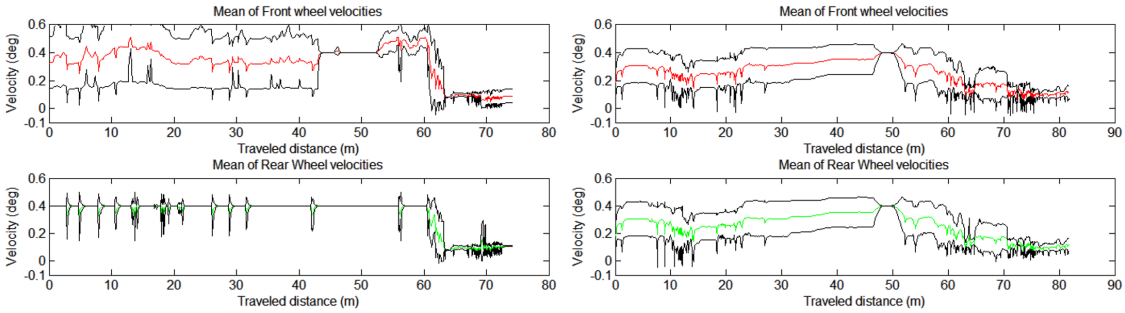

Figure B.40: Wheel velocities of user 3

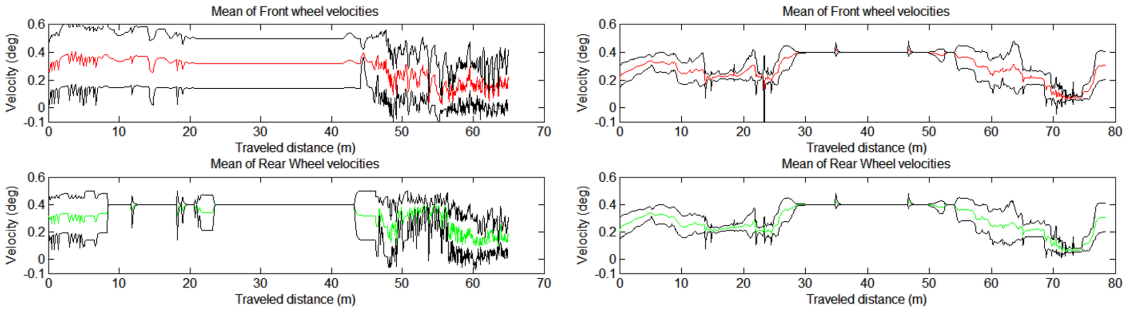

Figure B.41: Wheel velocities of user 4

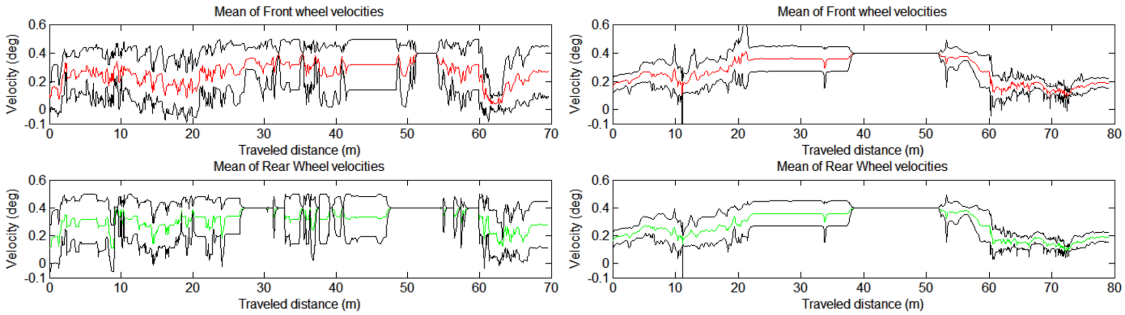

Figure B.42: Wheel velocities of user 5

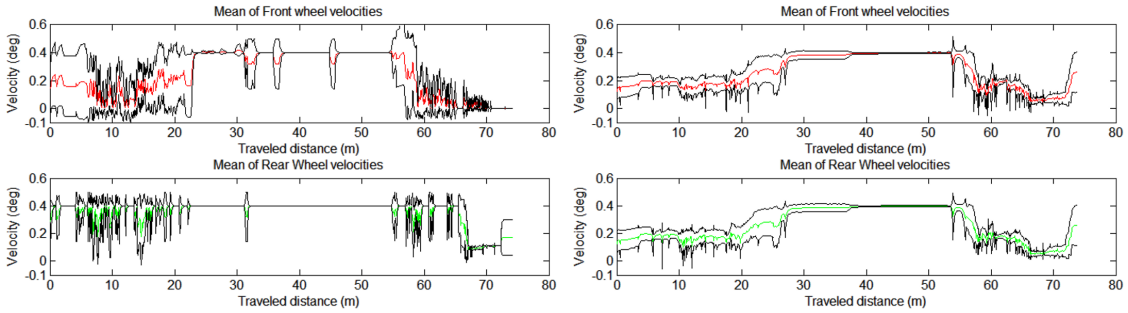

Figure B.43: Wheel velocities of user 6

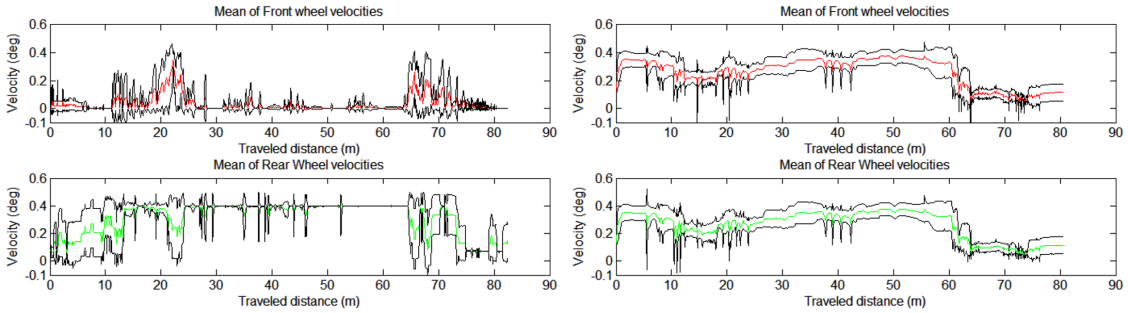

Figure B.44: Wheel velocities of user 7

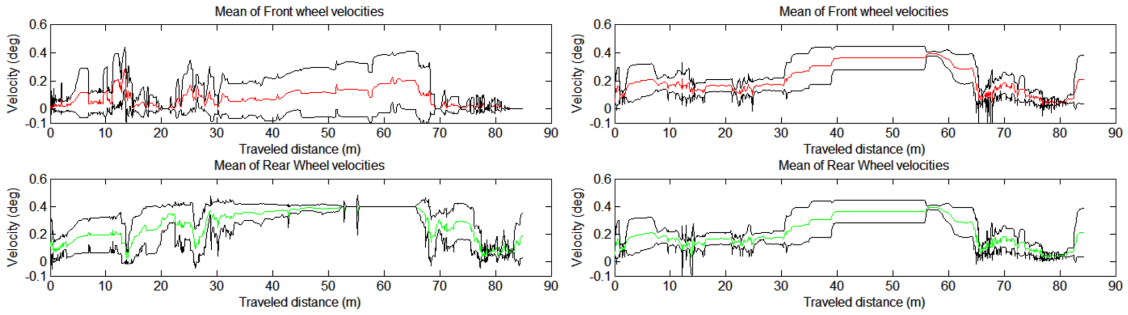

Figure B.45: Wheel velocities of user 8

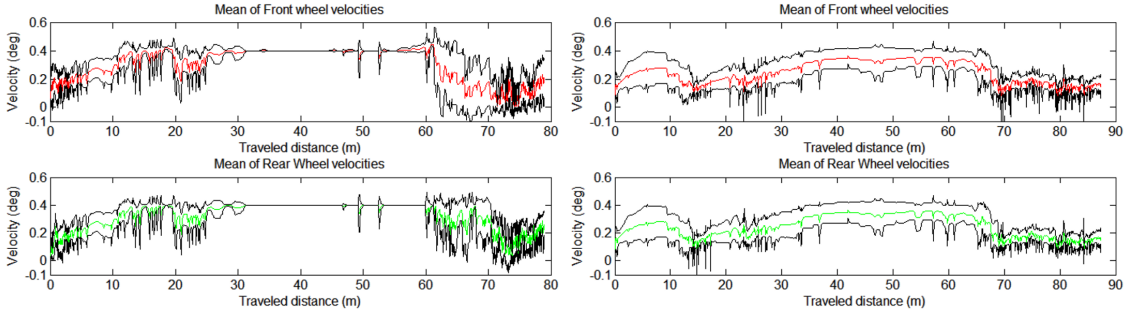

Figure B.46: Wheel velocities of user 9

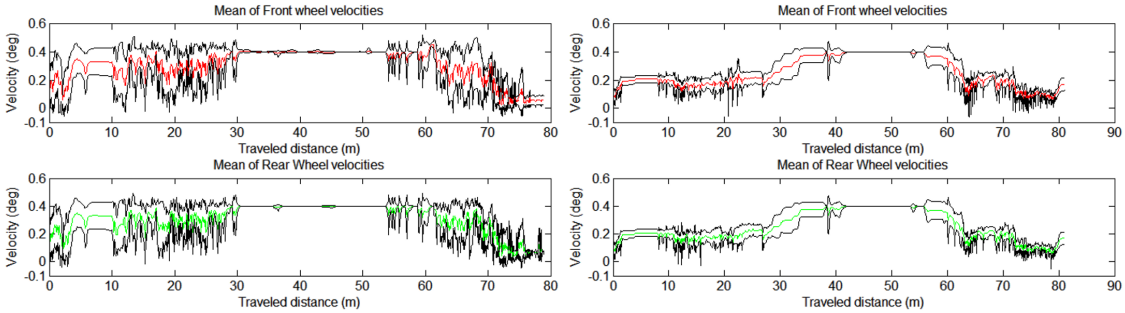

Figure B.47: Wheel velocities of user 10

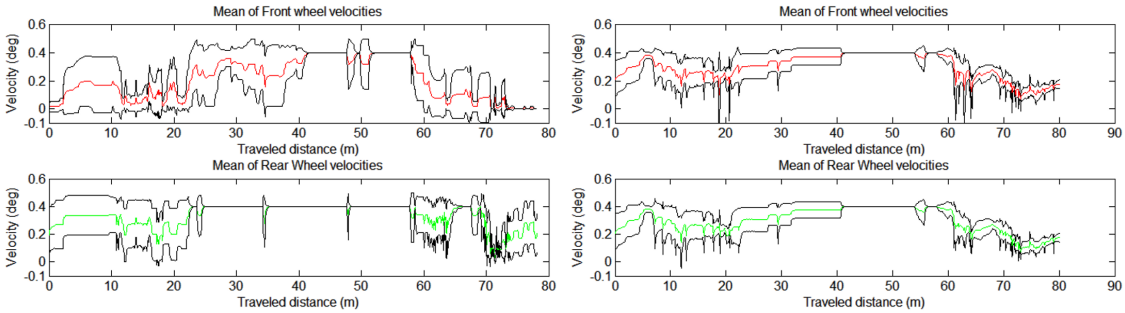

Figure B.48: Wheel velocities of user 11

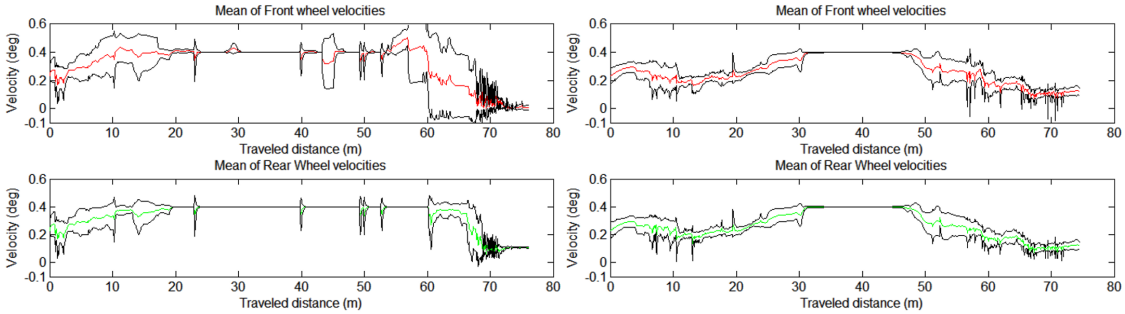

Figure B.49: Wheel velocities of user 12

Table B.1: Duration, energy and trajectory length by user

Gamepad v.2					JRD			
#	Duration (s)	Energy FW	Energy RW	Length (m)	#	Duration (s)	Energy FW	Energy RW
1	120,00	296,91	400,61	98,36	1	120,00	296,91	400,61
2	117,60	59,47	366,83	79,69	2	117,60	59,47	366,83
3	118,80	74,744	271,23	78,75	3	118,80	74,74	271,23
4	122,60	39,00	306,31	78,86	4	122,60	39,00	306,31
5	124,50	143,91	270,98	80,71	5	124,50	143,91	270,98
1	82,44	263,36	541,07	77,30	1	82,44	263,36	541,07
2	86,36	279,01	539,60	75,85	2	86,36	279,01	539,60
3	83,08	544,44	390,93	75,12	3	83,08	544,44	390,93
4	94,92	510,65	675,13	78,61	4	94,92	510,65	675,13
5	70,60	342,31	844,82	81,34	5	70,60	342,31	844,82
1	122,56	188,08	1121,88	76,42	1	122,56	188,0	1121,80
2	63,36	15,52	522,67	72,12	2	63,36	15,52	522,67
3	70,88	14,93	590,23	74,65	3	70,88	14,93	590,23
4	60,52	12,00	403,43	75,46	4	60,52	12,00	403,43
5	55,68	10,93	407,91	74,27	5	55,68	10,93	407,91
1	71,72	185,80	429,00	77,26	1	71,72	185,81	429,00
2	124,16	79,02	478,51	76,59	2	124,16	79,02	478,51
3	82,60	242,07	430,20	74,78	3	82,60	242,07	430,20
4	65,24	224,66	384,90	64,98	4	65,24	224,66	384,90
5	68,16	157,21	543,54	65,33	5	68,16	157,21	543,54
1	129,00	634,61	835,67	86,84	1	129,00	634,61	835,67
2	76,56	1062,40	992,40	81,71	2	76,56	1062,40	992,49
3	83,40	706,99	861,56	80,34	3	83,40	706,99	861,56
4	64,56	1189,70	797,28	81,42	4	64,56	1189,70	797,28
5	64,40	873,34	804,95	69,66	5	64,40	873,34	804,95
1	85,24	204,34	637,80	75,51	1	85,24	204,34	637,80
2	93,76	330,49	520,25	77,00	2	93,76	330,49	520,25
3	83,68	187,80	542,88	77,43	3	83,68	187,88	542,88
4	79,36	369,04	735,46	74,25	4	79,36	369,04	735,46
5	94,48	200,71	490,52	80,42	5	94,48	200,71	490,52
1	122,56	188,08	1121,80	76,42	1	122,56	188,08	1121,80
2	63,36	15,52	522,67	72,12	2	63,36	15,52	522,67
3	70,88	14,93	590,20	74,65	3	70,88	14,93	590,23
4	60,52	12,00	403,43	75,46	4	60,52	12,00	403,43
5	55,68	10,90	407,91	74,27	5	55,68	10,93	407,91
1	226,80	71,10	337,51	82,63	1	226,80	71,10	337,51
2	135,96	203,30	459,30	85,02	2	135,96	203,30	459,30
3	163,72	269,67	505,03	91,28	3	163,72	269,67	505,03
4	406,76	429,73	446,88	94,34	4	406,76	429,73	446,88
5	286,36	183,90	319,57	86,07	5	286,36	183,90	319,57
1	103,04	98,71	723,50	78,85	1	103,04	98,714	723,50
2	100,36	79,02	1202,40	80,16	2	100,36	79,02	1202,40
3	78,12	176,50	443,03	79,04	3	78,12	176,56	443,03
4	88,28	37,30	537,29	79,41	4	88,28	37,30	537,29
5	73,00	38,53	394,57	78,94	5	73,00	38,53	394,57
1	118,52	62,75	913,12	79,45	1	118,52	62,75	913,12
2	97,88	40,68	892,03	78,91	2	97,88	40,68	892,03
3	158,84	27,5	1108,50	81,48	3	158,84	27,59	1108,50
4	125,56	27,09	712,87	79,19	4	125,56	27,09	712,87
5	133,20	51,82	854,17	81,94	5	133,20	51,82	854,17
1	86,08	118,60	277,26	76,42	1	86,08	118,63	277,26
2	135,92	118,74	237,14	77,67	2	135,92	118,74	237,14
3	122,08	167,32	287,46	78,84	3	122,08	167,32	287,41
4	73,88	84,70	318,76	78,17	4	73,88	84,70	318,76
5	85,36	158,66	438,96	76,18	5	85,36	158,66	438,96
1	74,56	13,09	346,59	70,28	1	74,56	13,09	346,59
2	77,92	45,77	364,11	76,15	2	77,92	45,77	364,11
3	72,84	32,17	265,94	71,23	3	72,84	32,17	265,94
4	72,04	31,00	312,65	75,21	4	72,04	31,00	312,65
5	66,88	14,27	434,14	76,15	5	66,88	14,27	434,14