TÉCNICO LISBOA

VIENA

Veículo Inteligente Elétrico de Navegação Autónoma

Documentation Guide

Current authors:

Bruno Tibério: bruno.tiberio@tecnico.ulisboa.pt

Revision: 1

September 12, 2018

Acknowledgments

The group VIENA would like to acknowledge all past and present contributors that are making this project alive and growing. Following is a list of them in no particular order.

IST For allowing the creation of this project and providing a variety of tools and of course the facilities.

FST Lisboa Contributtor with main parts that made possible the powertrain. Among them are two essential parts, the battery pack and inverter.

Dr. João Fernandes Co-supervisor of project.

Dr. João Sequeira Co-supervisor of project.

Dr. Paulo Branco Co-supervisor of project.

Pedro Costa Former FST leader. Valuable knowledge contributor to get us started with parts supplied by FST team as well as many suggestions to problems that arise or can arise in future. His experience as former FST leader, saved us may time in problems.

André Agostinho FST member. Provided us help and information specially with battery pack.

André Antunes Former FST member. Enlightened us in the very beginning of CAN connection with inverter.

AEIST/ BPI In partnership, provided funding for the project as one of the winners in an open competition fundings.

Contents

List of Tables

List of Figures

Nomenclature

σ_P	Standart deviation associated to a position reading provided by a GNSS receiver.
σ_V	Standart deviation associated to a velocity reading provided by a GNSS receiver.
$\Omega_\times$	Skew symmetric matrix of angular velocities used to represent cross products as matrix multiplications.
$\hat{q}$	Normalized quaternion.
$\otimes$	Quaternion product operation generally described as Hamilton product.
q^*	Quaternion conjugate.
q	Quaternion.
${}^bA$	Physical quantity defined in superscript frame (in this case b, body frame).
${}^b_wR_{2D}$	A rotation matrix reduced to only two dimensions.
${}^b_wR$	Rotation matrix that maps physical quantities defined in subscript frame (in this case w, world frame) to superscript frame (in this case b, body frame).
a posteriori	Latin expression denoting, in this case, an estimate after performing correction using recent information.
a priori	Latin expression denoting, in this case, an estimate before atually having information about it and before possible correction.
intrinsic	Intrinsic rotation means that rotation is applied about an axis of the moving frame .
lever arm	A distance correction for the point of application of a physical quantity or to describe the same quantity in a specific location.
polyfit	Polynomial fit function available in Matlab®.
pure quaternion	Pure quaternion is a quaternion with real part equal to zero.

Acronyms

9DOF Nine Degrees of Freedom.

A-GNSS Assisted GNSS.

ADC Analog to Digital Converter.

AHRS Attitude and Heading Reference System.

CEP Circular error probable.

DC Direct Current.

DIY Do It Yourself.

DOF Degrees of Freedom.

ECEF Earth Centered, Earth Fixed.

EEPROM Electrically-Erasable Programmable Read-Only Memory.

EKF Extended Kalman Filter.

ENU East North Up.

EPOS Easy Positioning System.

FS Full Scale.

FST Lisboa Formula Student Team Lisboa.

GNSS Global Navigation Satellite System.

GPS Global Position System.

I^2C Inter-Integrated Circuit bus.

IC Integrated Circuit.

IMU Inertial Measurement Unity.

INS Inertial Navigation System.

IST Instituto Superior Técnico.

LSB Least signigicant bit.

MAC Media Access Control.

MARG Magnetic Angular Rate and Gravity.

MCU Microcontroller Unit.

MEMS Microelectromechanical systems.

OS Operating System.

PCB Printed Circuit Board.

rms	root mean square.
Rpi	Raspberry PI.
SBC	Single Board Computer.
SoC	System on Chip.
SPI	Serial Peripheral Interface bus.
UART	Universal Asynchronous Receiver/Transmitter.
VIENA	Veiculo Inteligente Elétrico de Navegação Autónoma.
WGS84	World Geodetic System.
ZVU	Zero Velocity Update.

Chapter 1

Introduction

Instituto Superior Técnico (IST) is currently developing research in autonomous electrical vehicles, namely converting from commercial vehicles with upgraded power management. This provides a motivating/attracting setup for new students and a testbed for industry solutions and it is the main motivation for this work.

The main goal of the project is a conversion of an electrical car (Fiat Elettra) property of IST into an autonomous vehicle as framework for future projects or research in this field and also energy efficiency. Within the conversion, researchers, student and collaborators knowledge acquired during academic cycle is put into test and evaluated in a real situation.

With environmental issues and technological waste in mind, this project has also been focused on the reuse of parts and material from other IST projects by the simple fact that they have been replaced by improved versions or will no longer serve the current goals of those projects. Not only it is given a new propose for those parts but it will also allow the cost reduction.

This guide aims to be auxiliary documentation and future memory source as part of the IST project named Veiculo Inteligente Elétrico de Navegação Autónoma (VIENA) and as well as final report for fellowship BL43/2018.

Although all the instructions are given based on Linux operating system, they should be similar to any other Operating System (OS) and the majority of code developed is made in Python, intending to be as much cross-platform as possible.

1.1 Guide Outline

This documentation guide is organized in four main chapters, server, main sensors, controllers, miscellaneous. In the server is provided information about the used hardware, designed pieces, software configuration needed to get started and connections. In main sensors will be discussed the work done with the current available sensors. New sensor additions should be documented under this chapter. In controllers will be reported mainly hardware controllers and software necessary to connect, configure or communicate with it. Miscellaneous contain other parts that do not fit particularly inside any of those chapters but are necessary or may help in the project.

1.2 Contributions

During the fellowship BL43/2018, it was made the main contributions:

1. Development of 3D printed parts for allowing the control of steering wheel without disassembling or violating the integrity of steering shaft column
2. Development of a library in Python based on CAN bus protocol to enable interconnection between software and hardware to control the steering wheel.
3. Development of prototype circuit to add CAN communication for main computer
4. Development and design of PCB for future CAN communication to replace the protoboard for main computer
5. Identification of the function that relates the steering wheel position with angle of the front wheel in a bicycle model of the car.
6. Mounting of available sensors (two GNSS unities and one IMU 9DOF)
7. Calibration of the inertial sensor.
8. Updated code related with inertial sensor.
9. Updated code related with GNSS unities.

Although not initially planned, but because the change of hardware and/or adversities found during the fellowship, following additional work was also contributed:

1. Study of interconnection of the main battery pack
2. Study of software used for management of main battery pack
3. Study of hardware necessary for management of main battery pack
4. Study and initial software developement for invertor received from FST Lisboa (in progress)
5. Creation of an headless structure and an AP station based on Rpi to communicate with vehicle
6. Initial development of an mqtt based design to control the vehicle or check its status.

Main code contributions are available under Github repository in https://github.com/brtiberio/VIENA and will be kept up to date as soon as possible. The developed 3D pieces are also present in that repository and the companion drawings are shown in this guide also. The developed PCB schematics and board layouts are also presented in this guide.

Chapter 2

Main Server configuration

In order to control the current features already developed and used in the car it is used a Raspberry PI (currently version 3 model B). The Raspberry PI, from now on denoted as Rpi, is a Single Board Computer (SBC) affordable and low power, widely used among community and developed by Raspberry Pi Foundation.

2.1 Requirements

For using the SBC it is required the following hardware:

- **Raspberry Pi SBC** Recommended version at least 3 since it has a built in wifi.
- **MicroSD card** Recommended with 8Gb capacity at least.
- **Power Supply** Recommended 2.5 Amps . While developing, it is used as the main power for the SBC.
- **Host computer (any OS)** with internet and ability to read MicroSD cards.
- **Ethernet cable** for connecting to router/switch for internet access. It can also be used an crossover cable connecting directly to PC if it is able to share internet connection.

2.2 Initial Preparation

This preparation will focus on providing instructions to a minimal headless setup. For this reason, the recommended OS image version is the Raspbian Lite.

For this step it is recommended to be used the Etcher software for burning the OS image into microSD card. It is assumed the user is familiar with ssh capable software, for example Putty. The next steps are mainly based in the documentation guide provided by [1].

1. Download the OS image, Raspbian Lite version in here
2. Connect the MicroSD to host computer.
3. Open Etcher and:

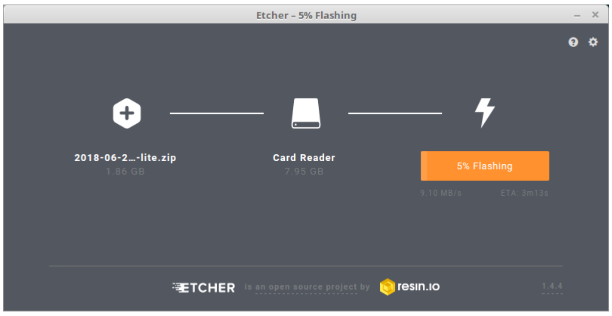


Figure 2.1: Etcher flashing image to microSD card

(a) Select OS image or zip file that have been downloaded.

(b) Select the SD card you wish to write your image to.

(c) Review your selections and click 'Flash!' to begin writing data to the SD card.

(d) after successful write, continue to next step.

4. From microSD card, open the boot partition.
5. Create a new blank file with name "ssh" without extension. This will allow to enable the ssh daemon at first boot time.
6. Remove card from host computer and insert on raspberry PI.
7. Plug in the Ethernet cable and connect the Rpi to the same local network as host computer.
8. Plug in the power supply to Rpi.

If everything went as expected, the Rpi will start to boot and prepare the first setup. It will be seen led light blinking indicating activity. After a few minutes, it should be possible to access it remotely via ssh

2.2.1 Remote access

The next step is to find the ip address of the Rpi. For example, in linux terminal type `arp -a`. In figure 2.2 is seen an output example. In red stroke is shown the Media Access Control (MAC) address of a Rpi. Every MAC is unique and the first three bytes are fixed in every Rpi wich correspond to organizationally unique identifier [2] associated to Raspberry Pi Foundation, B8:27:EB [3].

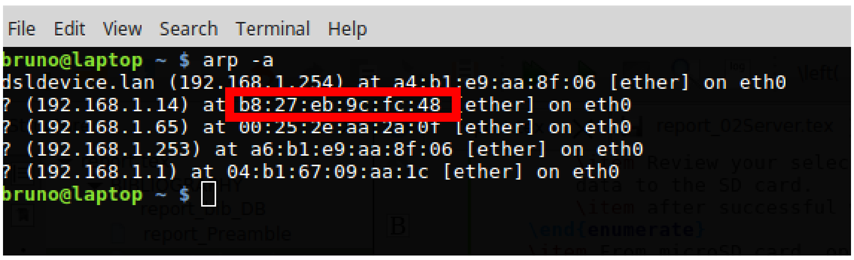


Figure 2.2: arp -a command example output

Perform the first login with using the default login details as shown in table 2.1. Using the discovered ip for the Rpi, in Linux, the first login may be performed using `ssh pi@<ip-of-Rpi>` command and entering the default password.

username:	pi
password:	raspberry

Table 2.1: Default login details for Rpi

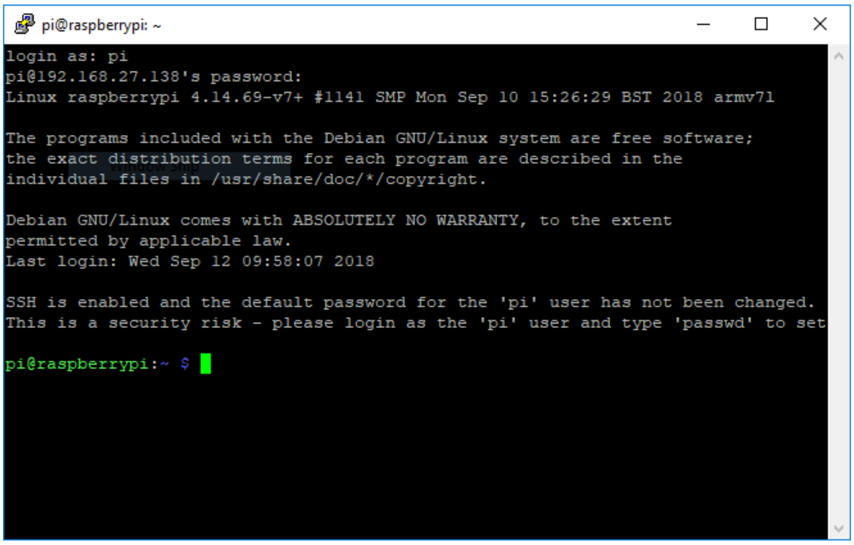


Figure 2.3: SSH sucessfull login

2.2.2 Initial update and configuration

If the user as been granted with permission to login, the next steps are used to perform a few tweaks. To do that user must use the command `sudo raspi-config`. Example output is seen in figure 2.4.

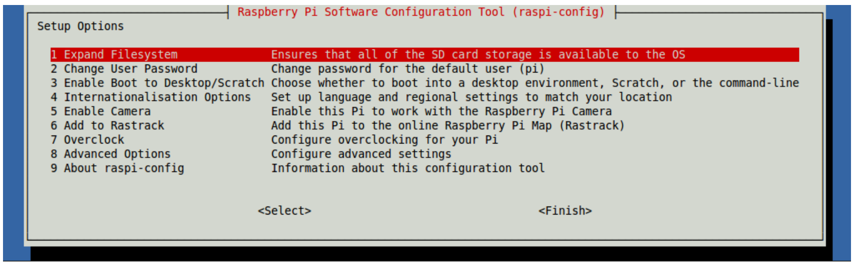


Figure 2.4: Raspi-config example output

- Set Keyboard Layout
- Update to lastest software
- Set Timezone
- Set language [optional]
- Change default login password
- Configure Wifi-zone [if present]
- Enable Serial Peripheral Interface bus (SPI) for CAN controller

- Enable Inter-Integrated Circuit bus (I^2C) for RTC
- Set hostname
- change default password
- change memory for GPU

The current settings are presented in table 2.2.

Username:	pi
Password:	fiatelettra
Hostname:	raspberrypi
wait for network at boot:	no
Language:	en_GB.UTF-8 UTF-8
Timezone:	Europe → Lisbon
Keyboard layout:	pt_PT
WIFI country:	PT_Portugal
SPI:	on
I2C:	on
Memory split:	16 (minimum since is running headless)

Table 2.2: Suggested details for Rpi

After successfully changed the settings, perform a full update and upgrade by running:

```
sudo apt update && sudo apt upgrade -y
```

2.3 Prepare CAN interface

The current protoboard uses the Integrated Circuit (IC) MCP2515 as CAN network controller and MCP2551 as CAN transceiver. Figure 2.5 shows the used protoboard with highlighted connections names and parts.

The schematic circuit implemented in the protoboard is present in figure 2.6. Current raspbian kernel, automatically support the can interaction via SPI to IC MCP2515, making it available via SocketCAN. To enable that, it is necessaries to configure MCP2515 driver on device tree overlay and install the can-utils package.

- Install can-utils using `sudo apt install can-utils`
- Enable MCP2515 overlay by using `sudo nano /boot/config.txt` and append at end:

```
#CAN bus controllers
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835
```

- To enable the interface at boot time, use `sudo nano /etc/network/interfaces` and append at end:

```
auto can0
iface can0 inet manual
  pre-up /sbin/ip link set can0 type can bitrate 1000000 \
  triple-sampling on restart-ms 10
  up /sbin/ifconfig can0 up
  down /sbin/ifconfig can0 down
```

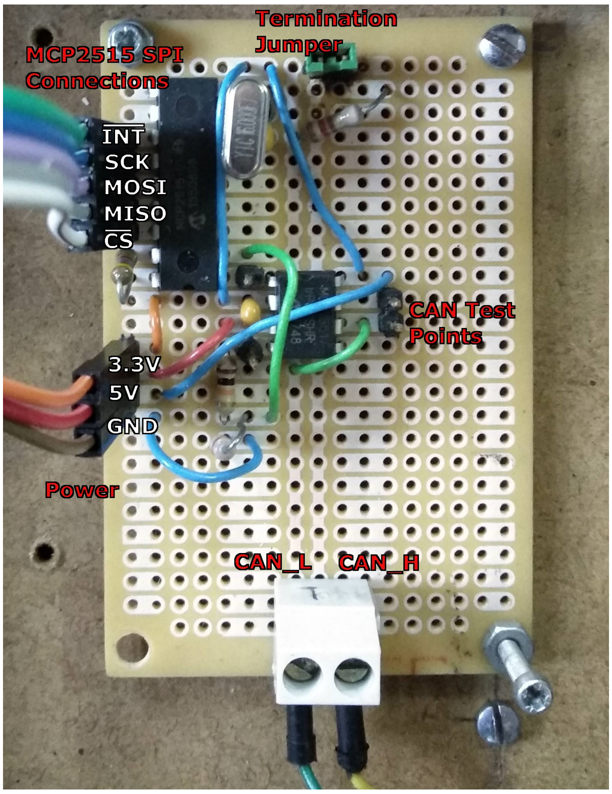


Figure 2.5: Protoboard designed for CAN interface

2.4 Troubleshooting

The command `arp -a` do not show my Rpi

In that case, try use the `sudo nmap -sS 192.168.1.0/24`, assuming the 192.168.1.0/24 is your local network. This is a time consuming command!

Cannot find my Rpi IP. Is it even running?

Maybe there is some problem with boot or bad microSD reading. The easiest solution is to connect a monitor and keyboard. Check messages at boot time. If nothing seems strange, manually login and then check if ehternet connection is ok using `ifconfig`

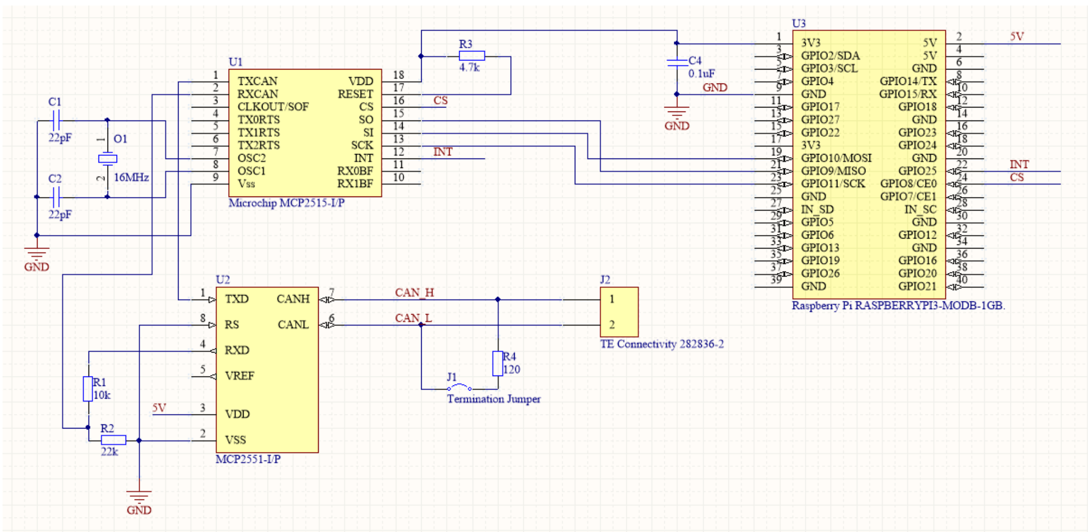


Figure 2.6: CAN protoboard schematic

Chapter 3

Code Guidelines

In this chapter is described the main guidelines used for coding and documentation as well as relevant suggestions. Since the majority of code is developed in Python, the guidelines are provided for that language. However similar suggestion may apply for the other languages cases. Here follows a list of sugestions:

Git

The first suggestion is the code version control system, Git. It has a lot integrations with majority of IDE's and have lot of free tools to manage it. It allows easy contributions from multiple users. Documentation and guides to become familiar with it can be found at try.github.io

Python3

Since Python core team is dropping support for python 2.7.x in 2020 [4] and some of important package developers are also dropping support for it [5], the chosen **version is the 3.X.**

Use virtual environment

Typically, every Linux distribution uses python to run critical routines. Perturbing the main ecosystem of python packages should be avoided, at least during development phase. Raspbian is no exception, so it is suggested to use virtual environments. It is used to create a isolated local installation in the directory you are working. Since version 3.5 the official recommended tool for creating virtual environments is the venv [6]. To create a virtual environment use the following console command `python3 -m venv /path/to/new/virtual/environment` or assuming the user is currently in desired local path, simplify to `python3 -m venv .` To activate/deactivate the local environment follow the specification accordingly to used OS as present in table 3.1, using the same assumption as before. After activation, the shell prefix will change from the default to `(<path of venv>)` which is a easy away to check if environment is active or not.

Requirements files (requirements.txt)

Requirement files are an easy away to install all the requisites necessary to run the code and ensure repeatability of installations. Unless any particular reason, a restriction of any package version should be avoided. Requirements file can be as simple as a list of necessary packages or more complex struture, if needed [7]. Installation of requirements is done by running the following command `pip install -r requirements.txt`

Google style docstrings

A good documentation is a key point for keeping good readability of a project. Docstrings format from Google Style Guide [8] is adopted. Many projects can fail from bad documentation.

Sphinx with automation

Similar to previous point, sphinx will allow the auto documentation generation from code. Is a popular tool for creating documentation and is used also in readthedocs.org, that allows to integrate documentation update with commits in the three main git repository management services, GitHub, Bitbucket, and GitLab. Using autodoc and napoleon extension allow an easy maintenance of code documentation.

Argparse

Use argparse to create clean and readable argument handling if necessary. It auto generates help, description and handles unknown options.

Logging

Since the main intention is to run programs in a headless computer, the logging module should be used instead of typical print() function. Not only it helps keep tracking of events defined in code, but also keeps track of other modules events. It also allows the creation of multiple destinations using the handlers and each handler can have its own format. Typical handlers used in project are files, console and websockets (currently only in development branchs)

Shell	Activate	Deactivate
	POSIX	
bash/zsh	source ./bin/activate	deactivate
fish	source ./bin/activate.fish	deactivate
csh/tcsh	source ./bin/activate.csh	deactivate
	Windows	
cmd	.\Scripts\activate.bat	deactivate
PowerShell	.\Scripts\Activate.ps1	deactivate

Table 3.1: Activation/deactivation of venv OS specific

Chapter 4

Batteries

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1 Main battery packs

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.1 Connections

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.2 Power requirements

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.3 Enabling power

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.4 List of CAN messages

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2 Auxiliary battery pack

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 5

Sensors

In this chapter will is briefly explained several notations used along the report and important definitions that help the reader to walk through the work. Many parts of the text presented here are extracted from [9] where full description is available and user should refer to it, if necessary. The notation system of leading superscripts and subscripts is used to denote relative frames orientation or general physical quantities (vectors, points). For frames orientations, leading subscript refers to the frame being represented with respect to the frame in leading superscript. For example let R be a rotation matrix. Using the notations stated before, ${}^{a}_{b}R$ describes orientation of frame b with respect to frame a. For physical quantities, a vector is represented in the frame defined by is leading superscript, ${}^{a}v$ and in similar way points follow the same rule, ${}^{a}P = [{}^{a}x, {}^{a}y, {}^{a}z]$.

5.1 Definition of frames

The frames cited are Earth Centered, Earth Fixed (ECEF), World Geodetic System (WGS84), East North Up (ENU) and body frame. The ECEF, WGS84 are just auxiliary frames used to define the ENU frame which will be considered the world frame.

5.1.1 ECEF and ENU Frame

ECEF coordinate system defines a referential axis where the origin is defined as the center of Earth, X axis is defined through the intersection of the plan defined by zero latitude line (Equator) and plan defined by zero longitude line (prime meridian). The X-axis orientation is considered positive from center towards the point defined by zero latitude and zero longitude. Z axis is defined by line intersecting origin and both Poles, being positive towards North Pole. Y axis is perpendicular to the plan defined by X and Z axis and it positive direction is defined by right hand rule.

ENU coordinate system is a local coordinate system where the origin is located at a user defined point in ECEF coordinate system, with Y axis pointing towards North Pole and X axis pointing towards East. The plan defined by X and Y axis is tangent to the WGS84 frame on the origin of ENU. Z axis

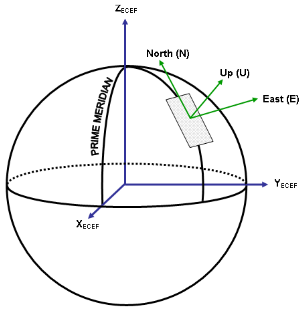


Figure 5.1: ECEF frame and Local ENU frame (source:wikipedia)

.

express the altitude from defined local plane (see figure 5.1). The ENU frame is considered in this work as the reference frame and will be denoted with superscript or subscript w.

Given a point of reference in ECEF frame, it is necessary to find the corresponding latitude and longitude of reference point (X_r, Y_r, Z_r). To do it is necessary to use the parameters of WGS84 presented in the table 5.1

WGS 84 Defining Parameters[10]		
Parameter	**Notation**	**Value**
Semi-major axis	a	6 378 137.0 m
Reciprocal of flattening	$1/f$	298.257 223 563
Semi-minor axis	b	6 356 752.3142 m
First eccentricity squared	e^2	6.694 379 990 14x10^{-3}
Second eccentricity squared	e'^2	6.739 496 742 28x10^{-3}

Table 5.1: WGS84 parameters necessary to transform ECEF coordinates into ENU

Using set of equations (5.1) to estimate the latitude (λ_r) and longitude (φ_r) for the reference coordinate point. The final transformation results in applying (5.2) to ECEF physical quantities [11].

$$p = \sqrt{{X_r}^2 + {Y_r}^2} \tag{5.1a}$$

$$\theta = \arctan\left(Z_r \frac{a}{pb}\right) \tag{5.1b}$$

$$\lambda_r = \mathrm{arctan2}(Y_r, X_r) \tag{5.1c}$$

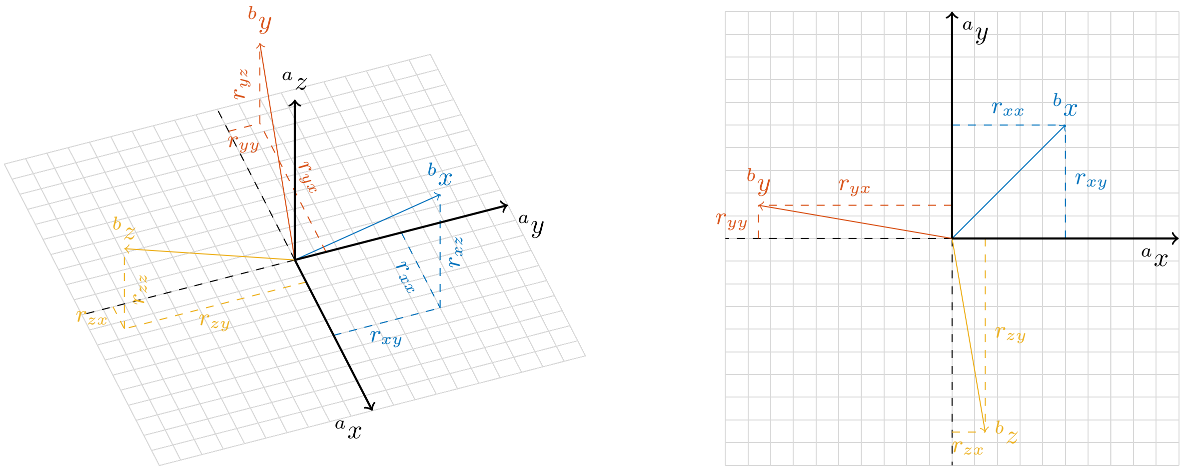
$$\varphi_r = \arctan\left(\frac{Z_r + e^2 b \sin^3(\theta)}{p - e'^2 a \cos^3(\theta)}\right) \tag{5.1d}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ENU} = \begin{bmatrix} -\sin(\lambda_r) & \cos(\varphi_r) & 0 \\ -\sin(\varphi_r)\cos(\lambda_r) & -\sin(\varphi_r)\sin(\lambda_r) & \cos(\varphi_r) \\ \cos(\varphi_r)\cos(\lambda_r) & \cos(\varphi_r)\sin(\lambda_r) & \sin(\varphi_r) \end{bmatrix} \begin{bmatrix} X - X_r \\ Y - Y_r \\ Z - Z_r \end{bmatrix}_{ECEF} \tag{5.2}$$

5.1.2 Body frame, *b*

Each sensor present in the razor board is aligned to match the sensor axes printed in the board as seen in figure 5.5. For simplicity, the razor sensor is placed as possible near the middle point of the bisector segment between the two wheel axes in such a away that YY axis as marked in the figure 5.5 is pointing towards the front of vehicle, XX axis is pointing to the right side of the car and ZZ axis is pointing to the top. This way, if the Euler angles describing the orientation of body frame related to world frame are all equal to zero, it means the axes in each frame are coincident apart from an offset in origin. Rotation angles are considered positive following the right hand rule in each axis. The origin of body frame is equal to intersection of rear wheel axis with the bisector defined above.

5.2 Euler angles and Rotation Matrix



(a) 3D view of axes frames example

(b) XY plane of axes frame example

Figure 5.2: Two frames axes example

The rotation matrix is a tool used to describe transformation of coordinates from one frame to another and this also describe orientation of one frame relative to another frame. The convention used in this work is represented by (5.3) and maps quantities described in frame b to frame a. Comparing the structure of (5.3) with figure 5.2 it is seen that columns of ${}^a_bR$ represent each unity vector defining all axes of frame b.

$$
{}^a_bR = \begin{bmatrix} {}^ar_{xx} & {}^ar_{yx} & {}^ar_{zx} \\ {}^ar_{xy} & {}^ar_{yy} & {}^ar_{zy} \\ {}^ar_{xz} & {}^ar_{yz} & {}^ar_{zz} \end{bmatrix} \tag{5.3}
$$

Rotation matrices belong to the orthonormal group and one important property is that ${}^a_bR {}^a_bR^T = I$ meaning ${}^a_bR^T = {}^a_bR^{-1}$. Also ${}^a_bR^T$ is equal to ${}^b_aR$ [12][13].

Euler angles are another form to describe orientation of one frame relative to another by defining three angles. The Euler angles convention adopted in this work is the Tait–Bryan intrinsic rotation sequence Z-Y-X, meaning referential a axes, represented in figure 5.2, can be mapped into referential b by performing sequential rotations, first along ZZ axis by an angle ψ, second along the resulting YY axis by an angle θ and final rotation along the resulting XX axis by an angle ϕ. In figure 5.3 is represented the sequence necessary to rotate frame a into frame b with the respective Euler angles to achieve the same orientation as expressed in figure 5.2. It is also represented the intermediary axes resulting from each sequential rotation and it positive direction. Each individual rotation is expressed by is own rotation matrix using the respective Euler angle associated and the final orientation is the result of successive matrices multiplications as shown in equation 5.4 where the intermediary axes sequence is denoted as expressed in the figure 5.3.

$$
\begin{aligned}
{}^b_aR &= {}^b_2R_x(\phi)^2_1R_y(\theta)^1_aR_z(\psi) \\
&= \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\psi) sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}
\end{aligned} \tag{5.4}
$$

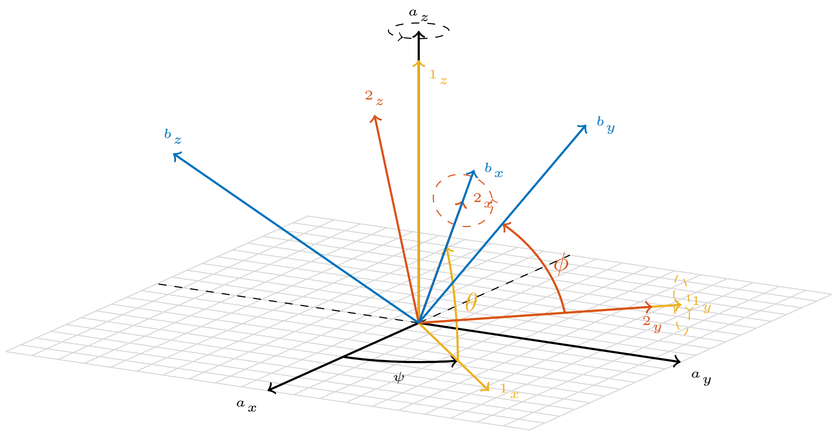


Figure 5.3: Two frames axes example - Euler angles. In this case, the angles are $\psi = 45^o, \theta = -45^o, \phi = 45^o$

5.3 Quaternion

During the work an Attitude and Heading Reference System (AHRS) algorithm based on [14] and derived in [9] is implemented and it uses quaternions. Basic definitions for understanding the filter terminology are explained in this section following the same notation as present by [9] in case the user as no notion of quaternions at all.

Quaternion is a complex number in four dimension that can be used, similar to rotation matrices and Euler angles, to represent orientation of frames with respect to others. The Euler rotation theorem and the Rodriguez formula are the basis for quaternion representation of rotations but will not be discussed. The theorem states it is possible to describe an orientation of frame relative to other by performing a rotation of α along an axis r that is defined by the points that remain static relative to the original frame. [15] quickly resumes the main idea about quaternions, why they can be used to express orientations and relation between them and Euler and Rodriguez formulations.

A quaternion can be represented by (5.5) where q_1 is the norm of it, q_2, q_3 and q_4 are complex coordinates with *i, j, k* being the axis versors. If a quaternion is normalized, it is denoted with a circumflex accent as shown in (5.6).

$$q = q_1 + q_2 i + q_3 j + q_4 k$$
$$q = [q_1\; q_2\; q_3\; q_4] \tag{5.5}$$

$$\hat{q} \implies \|q\| = 1 \tag{5.6}$$

Using the same notation as stated before, ${}^{w}_{b}q$ represent the orientation of body frame with respect to world frame. With (5.5) in mind, the following list of properties/operations are summarized in this list:

Identities All quaternions multiplications must obey to the following set of properties described in equation 5.7.

$$i^2 = j^2 = k^2 = ijk = -1 \tag{5.7a}$$

$$ij = -ji = k \tag{5.7b}$$

$$jk = -kj = i \tag{5.7c}$$

$$ki = -ik = j \tag{5.7d}$$

Quaternion multiplication The quaternions multiplication, denoted by $\otimes$, is defined by the Hamilton product as in equation 5.8

$$\begin{aligned} a \otimes b &= [a_1\; a_2\; a_3\; a_4] \otimes [b_1\; b_2\; b_3\; b_4] \\ &= \begin{bmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{bmatrix}^T \end{aligned} \tag{5.8}$$

Quaternion conjugate The quaternion conjugate describes the inverse rotation and is defined as equation 5.9.

$${}^{w}_{b}q^* = {}^{b}_{w}q = [q_1\; -q_2\; -q_3\; -q_4] \tag{5.9}$$

Vector rotation Let's define the following quaternion representation of the same vector but in each referential by using is pure quaternion as ${}^{w}v = [0\, {}^{w}x\, {}^{w}y\, {}^{w}z]$ and ${}^{b}v = [0\, {}^{b}x\, {}^{b}y\, {}^{b}z]$. The rotation of vector v from one frame to other, using a quaternion, is performed by equation 5.10.

$${}^{b}v = {}^{w}_{b}\hat{q} \otimes {}^{w}v \otimes {}^{w}_{b}\hat{q}^* \tag{5.10}$$

Composed rotations The composition of rotations can be described in quaternions as the product between quaternions. For example the sequence ${}^{a}_{b}\hat{q} \rightarrow {}^{b}_{c}\hat{q}$ is equal to ${}^{a}_{c}\hat{q}$ and is defined as equation 5.11.

$${}^{a}_{c}\hat{q} = {}^{b}_{c}\hat{q} \otimes {}^{a}_{b}\hat{q} \tag{5.11}$$

5.4 Novatel OEM4-G2L FlexPak

(a) Novatel OEM4-G2L FlexPak receiver.

(b) Novatel GPS-701-GG antenna

Figure 5.4: Novatel GNSS devices used (source:NovAtel)

Each Novatel GPS sensor is composed with Novatel OEM4-G2L FlexPak receiver and Novatel high performance GPS-701-GG antenna. The Novatel OEM4-G2L FlexPak receiver provides position and velocity estimations using GPS broadcasted satellites radio signals. It has two communications ports that can be configured independently. Additional to common GPS protocols, it has support for custom Novatel protocols (which will be used) providing the best possible estimation the receiver can do making it as simple as possible from the point of view of user. The receiver can handle both L1 and L2 signals provided by satellites [16], however, antenna module only can handle L1 frequency signal (1575.42MHz)[17]. Power supply to FlexPak enclosure should range from 6 to 18 volts DC with typical consumption of 5W. Without differential GPS methods, the Novatel receiver can achieve up to 1.8 meters precision Circular error probable (CEP) with single point operation. Output logs data rate is up to 20Hz in binary mode and 10Hz in ASCII format.

Since the receivers used are industrial grade, no particular effort is made to modeling any error source or trying to improve the solution given by it. However is important to have minimal knowledge the limitations of those types of receivers. It is trusted that the reported position and velocity information is the best the sensor could possibly estimate.

5.4.1 GNSS main errors

The main cause of Global Navigation Satellite System (GNSS) calculations errors are presented in table 5.2. While some can't be controlled by user (satellite clocks, orbit clocks), others can be mitigated using differential GNSS techniques, multi-frequency (currently civilians have access to more than one), satellite based augmentations system and multi-constellation or using modeling the error source. Multipath is probably the most user dependent but in some situations hard to avoid for example in streets surround with high buildings general denoted as urban canyons.

Source	Value (up to)
Satellite clocks	± 2m
Orbit Errors	± 2.5m
Inospheric Delays	± 5m
Tropospheric Delays	± 0.5m
Receiver Noise	± 0.3m
Multipath	± 1m

Table 5.2: Main source of errors in calculations using GNSS [18]

5.5 Sparkfun Razor IMU 9DOF

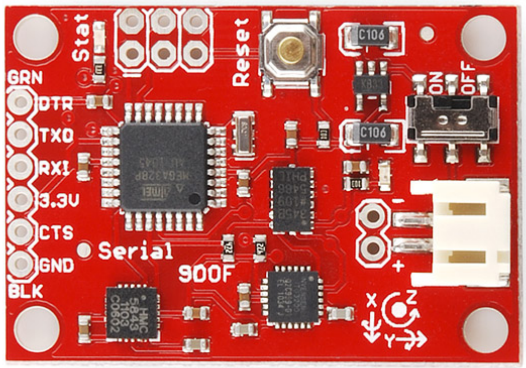


Figure 5.5: Razor 9DOF IMU.

Razor IMU 9DOF is an electronics board, developed by Sparkfun Electronics, which includes an accelerometer, a gyroscope and a magnetometer, each one with three axis of sensibility. Since this board is based on the Arduino project, the chip sensors are interconnected by an Atmel® ATmega328p Microcontroller Unit (MCU) which allows the configuration of sensors, handles the readings and processing of their respective outputs.

5.5.1 ADXL345 Digital Accelerometer

The ADXL345 is a Microelectromechanical systems (MEMS) sensor with three axis accelerometer made by Analog Devices . It has an adjustable acceleration scale range value up to $\pm 16g$. It is possible to measure both dynamic accelerations resulting from motion and shock or measure static accelerations such as gravity, which allows this device to be used as a tilt sensor also. Acceleration deflects the a proof mass attached to one differential capacitor plate and unbalances it, resulting in a sensor output whose amplitude is proportional to acceleration [19].Conversion of outputs is made through 10 to 13 bit Analog to Digital Converter (ADC) according to selected range to maintain the same scale of 4mg/LSB.

5.5.1.1 Accelerometer sensor model

The output of accelerometer is proportional to the sum of acceleration in each axis. However the accelerometer is not capable of distinguish if the acceleration is caused by a true acceleration produced by motion or produced by a static force, generally gravity force. As so, equation (5.12) describes the general output of accelerometer [20] where, ${}^{b}A_{ext}$ is the sum of real external acceleration due to linear or rotational dynamics in the body frame, ${}^{b}_{w}R$ is the rotation matrix mapping quantities in world frame into sensor frame, ${}^{w}g$ represents the fictitious acceleration due to gravity in the world frame, ${}^{b}A_0$ is an offset and δA_ϵ describes addictive gaussian noise. Ideally, G should be equal to identity matrix but in fact it represents the product of two matrices, one describing the cross-axis influence and the other a scale factor for each axis [20].

$$ {}^{b}A_s = G[{}^{b}A_{ext} + {}^{b}_{w}R\,{}^{w}g\,] + {}^{b}A_0 + \delta A_\epsilon \tag{5.12} $$

5.5.1.2 Accelerometer calibration

The suggested calibration method by the manufacturer [21] assumes that cross-axis influence is small enough and can be neglected, this way, G should only be represented by a diagonal matrix with the scale factors for each axis. However, if more precision for the application is needed, calibration using the ellipsoid fitting approach should be made [22].

For this application it will be used the suggested calibration by manufacturer resulting in exposing the sensor to six position combination while resting. This means the ${}^{b}A_{ext}$ will be zero and accelerometer will be only influenced by the gravity force, 1g.

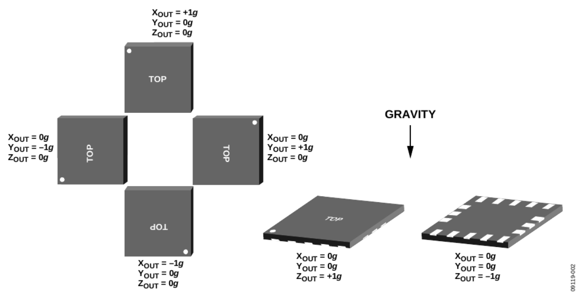


Figure 5.6: ADXL345 calibration poses and expected output [21].

Gains and offset [LSB]	Razor
Z gain	253.66
Z offset	-3.47
Y gain	266.27
Y offset	-3.18
X gain	266.04
X offset	13.83

Table 5.3: Summary of values extracted and calculated for each axis of accelerometer in both sensors

Aligning one axis at a time with gravity vector, as seen in figure 5.6, the mean value of several samples is calculated. Using these means for each axis and the known acceleration values($\pm$1g), it is calculated the slope and offset of the respective line passing through the two points. Calculated values (see table 5.3) will be stored in the internal EEPROM memory of MCU to be loaded at boot time of each razor. The conversion between LSB and g unities is given by the relation $254LSB/g$ or $3.9mg/LSB$ [19]

5.5.2 ITG3200 Digital Gyroscope

The ITG-3200 is a microelectronic mechanical integrated circuit chip with three axis independent gyroscopes, built by Invensense able of measuring angular velocities up to $\pm 2000^\circ /s$. When the gyroscopes are rotated about any of the sensor axes, the Coriolis effect causes a deflection that is detected by a capacitive pick-off circuit proportional to angular velocity. The measured signal is filtered and converted using internal ADC. The scale factor that allow to convert the output values to angular velocities in degrees per second is, according to its datasheet [23], factory calibrated and equal to $14.375 LSB/^\circ .s^{-1}$.

5.5.2.1 Gyroscope sensor model

For a given axis, the gyroscope output is proportional the angular velocity sensed of that axis and is given by equation (5.13) where ${}^b\omega_s$ represents the vector output of the sensor in body frame at instant t, ${}^b\omega_{real}$ is the real angular velocity vector applied to sensor in body frame, ${}^b\omega_0$ is the bias vector term and $\delta\omega_\epsilon$ addictive gaussian noise. Since the sensor is described in [23] as factory calibrated, G which represent a scaled factor correction, is expected to be equal to identity matrix. Note that the equation model (5.13) is in fact a simplified version of a more accurate model described in [20]. As stated in [20] cross-axis misalignment and linear acceleration sensitivity are expected to be small for the application purpose so they are considered irrelevant.

In general, the offset value depends on the temperature value inside chip. Once internal stability is reached, offset values tends to be constant [24]. Electrical temperature stability can be achieved after a few minutes of operation. At the beginning of initialization of Razor, several samples are acquired internally in the microprocessor and mean value is calculated for each axis, defining this way the offset constant for each axis.

$${}^b\omega_s = G\,{}^b\omega_{real} + {}^b\omega_0 + \delta\omega_\epsilon \tag{5.13}$$

5.5.3 HMC5843 Digital compass

The HCM5843 sensor is a digital compass chip designed by Honeywell, designed for low field magnetic sensing. This sensor uses anisotropic magnetic resistor technology, making it immune to vibration noises, and the use of Wheatstone bridge makes it more robust to noise. A material with anisotropic magnetic resistor properties changes its resistance according to absolute value and direction of magnetic field.

5.5.3.1 Magnetometer model

The magnetometer sensor model follows a similar structure as two previous sensors and is given by equation (5.14). The ${}^bH_s$ is the value sensed by the sensor in body frame. C is a matrix resulting from the multiplication of matrices containing influence of cross-axis, gain scale factor and soft-iron interference [25]. ${}^wh_e$ represent the geomagnetic vector of Earth, ${}^bh_{offset}$ is an offset vector which describes the influence of zero-field offset and the ferromagnetic masses fixed relative to body frame usually denoted as hard-iron. δh_ϵ represents gaussian additive noise.

$$^{b}H_s = C\,^{b}_{w}R^{w}h_e\, + {}^{b}h_{offset} + \delta h_\epsilon \tag{5.14}$$

5.5.3.2 Geomagnetic field

The geomagnetic field can be described as 3D vector in each point as seen in [26]. As previous defined, $^{w}h_e$ represent the geomagnetic vector of Earth. Let us define now, $^{w}h_0$ as the horizontal projection of $^{w}h_e$. The angle between $^{w}h_0$ and $^{w}h_e$ is denominated as inclination, I. As stated in [26], the horizontal projection of the geomagnetic vector points to the magnetic north which may not be aligned with the north axis of reference frame, ^{w}Y. This defines the magnetic declination angle, D, as the angle between horizontal projection and the ^{w}Y axis.

For Lisbon, the magnetic declination angle, in January 2017, defined in the reference frame is equal to 2.48° , inclination angle is equal to 52.77° and the $^{w}h_e$ is equal to $[-1121.5, 26479.8, -34884.7]^T$ nanoteslas [27]

5.5.3.3 Hard and soft-iron compensation

The compensation for hard and soft-iron effects can be done using a geometric approach as described by [25] [28] [29]. If the geomagnetic vector is known and sensor perfect calibrated, free of any type ambient distortion, points collected are expected to belong a surface of an origin centered sphere with radius equal to the magnitude of geomagnetic field $^{w}h_e$. Distortions, cross-axis influence, scale-factor inequalities between axis, causes the expected sphere to be deformed into a ellipsoid. Collecting points in 3D space by performing rotations of the sensor frame, allow us to fit the data to an ellipsoid surface and after determinate the matrix C and offset vector $^{b}h_{offset}$.

However since the body frame is a vehicle, it is not physically possible to collect points based in rotations of YY and XX axis. Only ZZ axis rotations are available and the data instead of belonging to an ellipsoid surfaces, will belong to an ellipse in Z-plane as a result from that plan cutting the ellipsoid in some undetermined z coordinate. Because of restriction stated before it will be considered that the influence in ZZ axis is zero and the scale factor is equal to one. Offset in ZZ axis will also be considered zero. In [9] is shown that this assumption is not relevant. (5.14) will be rearranged to equation (5.15) form.

$$\begin{aligned} ^{b}H_s &= C\,^{b}_{w}R^{w}h_e\, + {}^{b}h_{offset} + \delta h_\epsilon \\ ^{b}_{w}R^{w}h_e &= C^{-1}[^{b}H_s - {}^{b}h_{offset} - \delta h_\epsilon] \end{aligned} \tag{5.15}$$

Without loss of generality, $C^{-1}\delta h_\epsilon$ will result also in a gaussian vector so it will be rewritten as δh_ϵ. Equation (5.15) can take now be arranged into (5.16) where $^{b}H_c$ is the corrected value after calibration process.

$$
\begin{aligned}
{}^{b}_{w}R(t)^{w}h_e &= C^{-1}[^{b}H_s - {}^{b}h_{offset} - \delta h_{\epsilon}] \\
{}^{b}H_c &= C^{-1}[^{b}H_s - {}^{b}h_{offset}] + \delta h_{\epsilon}
\end{aligned} \tag{5.16}
$$

The matrix C^{-1} and ${}^{b}h_{offset}$ is obtained as described by [28] using the *ellipse_fit*[1]. The function *ellipse_fit* returns the least square estimate that best fits the data collected from sensor and the parameters that define the estimated ellipse as described in this list:

- **semi-major** - The length of major semi axis of ellipse.
- **semi-minor** - The length of minor semi axis of ellipse.
- x_0 - The offset in the XX axis of ellipse.
- y_0 - The offset in the YY axis of ellipse.
- α - Rotation angle between semi-major axis and XX axis in reference frame.

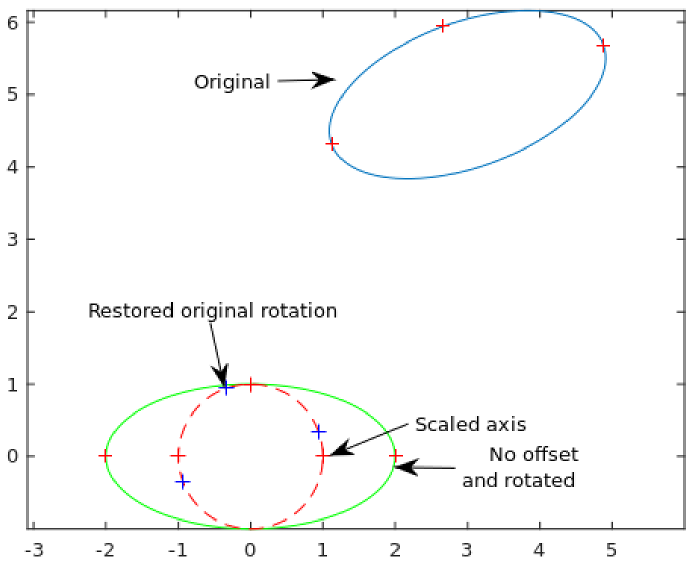


Figure 5.7: Simulation of the calibration steps applied to hypothetical ellipse data.

Take note the fact that since is not physically possible to perform rotations in 3D space, the calibration will only be reflect the XX and YY axis. Using the values returned, the calibration process consists in the following sequential steps:

1 Remove offset by using equation (5.17) with ${}^{b}h_{offset} = [^{b}x_0, {}^{b}y_0, 0]^T$

$$
{}^{b}H_s = {}^{b}H_s - {}^{b}h_{offset} \tag{5.17}
$$

2 Align semi-major axis of ellipse with the XX axis of reference frame using a rotation of $-\alpha$ by using equation (5.18)

[1] https://www.mathworks.com/matlabcentral/fileexchange/22423-ellipse-fit

$$
\begin{aligned}
{}^{b}H_s &= R_{cal}\,{}^{b}H_s \Leftrightarrow \\
\Leftrightarrow {}^{b}H_s &= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{b}H_s
\end{aligned}
\tag{5.18}
$$

3 Apply a scaling matrix S to make semi-major axis the same length as semi-minor axis using (5.19)

$$
\begin{aligned}
{}^{b}H_s &= S\,{}^{b}H_s \Leftrightarrow \\
\Leftrightarrow {}^{b}H_s &= \begin{bmatrix} \frac{semi-minor}{semi-major} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{b}H_s
\end{aligned}
\tag{5.19}
$$

4 Restore the initial rotation α to the scaled data using the transpose of R_{cal} as seen in (5.20)

$$
\begin{aligned}
{}^{b}H_s &= R_{cal}^{T}\,{}^{b}H_s \Leftrightarrow \\
\Leftrightarrow {}^{b}H_s &= \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{b}H_s
\end{aligned}
\tag{5.20}
$$

The ${}^{b}H_s$ resulting from step four is in fact the expected corrected reading of sensor, ${}^{b}H_c$. In figure 5.7 visually represented the steps applied to a hypothetical ellipse. Resuming, all sequential steps in one equation results in (5.21).

$$
{}^{b}H_c = R_{cal}\,S\,R_{cal}^{T}[{}^{b}H_s - {}^{b}h_{offset}] \tag{5.21}
$$

Comparing the structure of (5.21) with (5.16) this implies that C^{-1} is equal to $R_{cal}\,S\,R_{cal}^{T}$.

The results of calibration are presented for one of the devices in figure 5.8 and the obtained are expressed in the table 5.4. The trajectory used is not important as long as it is able perform one or more rotation of the vehicle in the horizontal plan. In figure 5.8c is shown the results before and after calibration for the estimation of Euler ψ angle. It is also present the estimation of the same angle using the the one of the gps unities as a reference for the comparison.

(a) Trajectory used for calibration.

(b) Magnetometer calibration

(c) ψ angle estimation before and after calibration of magnetometers

Figure 5.8: Magnetometer calibration steps applied to real data

	Razor 1
${}^{b}h_{offset}$ $[nT]$	$[-2.5828\ \ -2.1703]^{T}\mathrm{x}10^{4}$
α $[deg]$	107.5238
$\frac{semi-minor}{semi-major}$	0.9567
R_{cal}	$\begin{bmatrix} -0.3011 & -0.9536 & 0 \\ +0.9536 & -0.3011 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
C^{-1}	$\begin{bmatrix} +0.9961 & +0.0124 & 0 \\ +0.0124 & +0.9606 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 5.4: Magnetometers calibration results.

Chapter 6

Steering

The control of steering wheel take advantage of motor for steering assist originally present in the car. The motor is controlled by a positioning controllers designed for brushed DC and brushless DC motors with encoders. denominated Easy Positioning System (EPOS) by many manufactures

6.1 Maxon EPOS 70/10 controller

Figure 6.1: Maxon EPOS 70/10 controller

The controller available to use is the Maxon EPOS 70/10. It includes CAN network interface and a RS-232. The relevant electrical specs can be seen in table 6.1. The full specifications as well as connections layout must be seen in [30]. The controller contains two led in a single package that is used as status reference. Table 6.2 present the status of device based on pattern of leds shown by it.

Description	**Min.**	**Max.**	**Unity**
Power supply voltage V_{CC} (Ripple $<$10%)	11	70	V_{DC}
Max. output voltage		$0.9 \cdot V_{CC}$	V_{DC}
Max. output current I_{max} ($<$1sec)		25	A
Continuous output current I_{cont}		10	A
Sample rate PI - current controller	10K	10K	Hz
Sample rate PI - speed controller	1K	1K	Hz
Sample rate PI - positioning controller	1K	1K	Hz
Max. speed (motors with 2 poles)		25K	rpm

Table 6.1: EPOS electric specifications

Description	Red	Green
The EPOS is in state: • Switch ON Disabled • Ready to Switch ON • Switched ON • The power stage is disabled	OFF	Blinking ($\approx$ 1Hz)
The EPOS is in state: • Operation Enable • Quick Stop Active • The power stage is enabled	OFF	ON
EPOS is in Fault State	ON	OFF
EPOS is in temporary state Fault Reaction Active	ON	ON
There is no valid firmware on the EPOS (due to a failed firmware download)	ON	Flashing

Table 6.2: Maxon EPOS 70/10 Led status

6.2 Steering sensor support

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.3 Calibration process

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.4 interface library

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 7

NovatelOEM4 GPS Library

Date 24 Jul 2018

Version 0.4

Author Bruno Tibério

Contact bruno.tiberio@tecnico.ulisboa.pt

This module contains a few functions to interact with Novatel OEM4 GPS devices. Currently only the most important functions and definitions are configured, but the intention is to make it as much complete as possible.

A simple example can be run by executing the main function wich creates a Gps class object and execute the following commands on gps receiver:

- **begin:** on default port or given port by argv[1].
- **sendUnlogall**
- **setCom(baud=115200):** changes baudrate to 115200bps
- **askLog(trigger=2,period=0.1):** ask for log *bestxyz* with trigger ONTIME and period 0.1
- wait for 10 seconds
- **shutdown:** safely disconnects from gps receiver

Example:

```
$python NovatelOEM4.py
```

Bibliography

[1] Raspberry Pi Foundation. Installing operating systems images, April 2018. URL https://www.raspberrypi.org/documentation/installation/installing-images/README.md. (2018).

[2] Wikipedia. Mac address, 2018. URL https://en.wikipedia.org/wiki/MAC_address. (2018).

[3] Wireshark. Oui lookup tool, 2018. URL https://www.wireshark.org/tools/oui-lookup.html. (2018).

[4] Python Software Foundation. Status of python branches, 2018. URL https://devguide.python.org/#status-of-python-branches. (2018).

[5] Numpy Developers. Plan for dropping python 2.7 support, 2017. URL https://www.numpy.org/neps/nep-0014-dropping-python2.7-proposal.html. (2018).

[6] Python Software Foundation. Python 3 documentation, 2018. URL https://docs.python.org/3/. (2018).

[7] Python Software Foundation. Pip 18.0 user guide - requirements file format, 2018. URL https://pip.pypa.io/en/stable/reference/pip_install/#requirements-file-format. (2018).

[8] Google Python Team. Google python style guide, 2018. URL https://github.com/google/styleguide/blob/gh-pages/pyguide.md#38-comments-and-docstrings. (2018).

[9] B. Tibério. An online filter study for inertial properties estimation based on low-cost sensors. Msc thesis, Instituto Superior Técnico, November 2017. URL https://fenix.tecnico.ulisboa.pt/downloadFile/1126295043835604/Thesis.pdf. (2018).

[10] Wikipedia. Geodetic datum, 2017. URL https://en.wikipedia.org/wiki/Geodetic_datum#Earth_reference_ellipsoid. (2017).

[11] Wikipedia. Geographic coordinate conversion, 2017. URL https://en.wikipedia.org/wiki/Geographic_coordinate_conversion#From_ECEF_to_ENU. (2017).

[12] J. S. Sequeira. Introdução à robótica. Introductory notes about Robotic concepts and definitions, 2016. URL http://users.isr.ist.utl.pt/~jseq/caps1-6.pdf.

[13] Wikipedia. Euler's rotation theorem, 2017. URL https://en.wikipedia.org/wiki/Euler%27s_rotation_theorem. (2017).

[14] S. O. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Technical report, University of Bristol, 2010. URL http://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf.

[15] J. V. Verth. Understanding Quaternions - Presentation at GDC 2013, 2013. URL https://www.essentialmath.com/GDC2013/GDC13_quaternions_final.pdf. (2017).

[16] NovAtel. *OEM4 Family - USER MANUAL - VOLUME 1*, 2005. URL https://www.novatel.com/assets/Documents/Manuals/om-20000046.pdf.

[17] NovAtel. Antennas GPS-701-GG GPS-702-GG pinwheel ® antennas enhance flexibility and reduce costs dual constellation for enhanced positioning, 2015. URL https://www.novatel.com/assets/Documents/Papers/GPS701_702GG.pdf. (2017).

[18] NovAtel. An introduction to gnss, 2015. URL https://www.novatel.com/an-introduction-to-gnss/. (2017).

[19] *ADXL345 Product Specification*. Analog Devices, Inc, 2015. URL http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf. Revision E.

[20] Vectornav. Vectornav support library - calibration, N/A. URL http://www.vectornav.com/support/library/calibration. (2017).

[21] T. Tusuzki. *ADXL345 Quick Start Guide*. Analog Devices, Inc, June 2010. URL http://www.analog.com/media/en/technical-documentation/application-notes/AN-1077.pdf. Rev. 0.

[22] T. Pylvänäinen. Automatic and adaptive calibration of 3D field sensors. *Applied Mathematical Modelling*, 32(4):575–587, 2008. ISSN 0307904X. doi: 10.1016/j.apm.2007.02.004. URL http://www.sciencedirect.com/science/article/pii/S0307904X07000297.

[23] *ITG-3200 Product Specification*. Invensense, 2011. URL https://www.invensense.com/wp-content/uploads/2015/02/ITG-3200-Datasheet.pdf. Revision 1.7.

[24] O. J. Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, 2007. URL https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf.

[25] T. Ozyagcilar. *Calibrating an eCompass in the presence of Hard and Soft-Iron Interference*. Freescale Semiconductor, Inc, November 2015. URL http://cache.freescale.com/files/sensors/doc/app_note/AN4246.pdf. Rev. 4.0.

[26] IPMA. Geomagnetismo, 2005. URL http://www.ipma.pt/pt/enciclopedia/geofisica/geomagnetismo/. (2017).

[27] NOAA. Magnetic field calculators, 2017. URL https://www.ngdc.noaa.gov/geomag-web/#declination. (2017).

[28] M. Caruso. Applications of magnetic sensors for low cost compass systems. *IEEE 2000. Position Location and Navigation Symposium (Cat. No.00CH37062)*, pages 1–8, 2000. doi: 10.1109/PLANS.2000.838300. URL https://cdn.shopify.com/s/files/1/0038/9582/files/lowcost.pdf.

[29] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira. Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1293–1306, 2011. ISSN 00189251. doi: 10.1109/TAES.2011.5751259. URL http://www.dem.ist.utl.pt/poliveira/Invest/05751259.pdf.

[30] Maxon Motor, AG. *EPOS 70/10 Hardware Reference*. Maxon Motor, AG, 752380-04 edition, December 2008. URL https://www.maxonmotor.com/medias/sys_master/root/8803613802526/300583-Hardware-Reference-En.pdf. (2018).