Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Μηχανολογων Μηχανικων

Εργαστηριο Αυτοματου Ελεγχου και Ρυθμισεως Μηχανων

Διπλωματικη Εργασια

Εκτίμηση Κατάστασης ενός Τετράτροχου Ρομπότ

Ασημένια Ζ. Λαϊνά

Επιβλέπων:

Κωνσταντίνος Κ. Κυριακόπουλος

Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

Page intentionally left blank.

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Μηχανολογων Μηχανικων

Εργαστηριο Αυτοματου Ελεγχου και Ρυθμισεως Μηχανων

Διπλωματικη Εργασια

Εκτίμηση Κατάστασης ενός Τετράτροχου Ρομπότ

Ασημένια Ζ. Λαϊνά

Επιβλέπων:

Κωνσταντίνος Κ. Κυριακόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

Page intentionally left blank.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Κωνσταντίνο Κυριακόπουλου που μου έδωσε την ευκαιρία να εργαστώ υπό την επίβλεψή του σε ένα τόσο οργανωμένο και σύγχρονο εργαστήριο πάνω σε ένα τόσο ενδιαφέρον θέμα, καθώς και όλα τα μέλη του εργαστηρίου για την συνεργασία που είχαμε και την προθυμία τους να βοηθήσουν κατά τη διάρκεια εκπόνησης της Διπλωματικής μου εργασίας. Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον Γιώργο Καρρά, Πάνο Μαράντο και Παναγιώτη Βλαντή για τη συμβολή τους, την υποστήριξή τους και την προθυμία τους να βοηθήσουν. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξή τους και την απεριόριστη αγάπη τους, καθόλη τη διάρκεια της ζωής μου, παρέχοντας τα εφόδια για να πραγματοποιήσω τα όνειρά μου και τους στόχους μου.

Page intentionally left blank.

Εκτίμηση Κατάστασης ενός Τετράτροχου Ρομπότ

Ασημένια Ζ. Λαϊνά

Περίληψη

Σκοπός αυτής της διπλωματικής είναι η εκτίμηση κατάστασης ενός τετράτροχου skid-steering ρομπότ με τη χρήση complementary filters. Η εκτίμηση της κατάστασης επιτυγχάνεται με τη ταυτόχρονη χρήση και τον κατάλληλο συνδιασμό μετρήσεων που λαμβάνονται από τους αισθητήρες που είναι τοποθετημένοι στο ρομπότ. Οι αισθητήρες αυτοί περιλαμβάνουν ένα IMU (Inertial Measurement Unit), GPS (Global Positioning System) και οδομετρία. Η συνδιασμένη χρήση των αισθητήρων με τη μέθοδο των complementary filters δεν απαιτεί μεγάλη επεξεργαστική ισχύ και οι υπολογισμοί πραγματοποιούνται πιο γρήγορα, παρέχοντας συχνότητα εκτίμησης της κατάστασης του ρομπότ ίση με αυτή του πιο γρήγορου αισθητήρα που είναι εγκατεστημένος σε αυτό. Το ρομπότ που χρησιμοποιήθηκε για την πραγματοποίση αυτής της διπλωματικής είναι σχεδιασμένο για χρήση σε εξωτερικό περιβάλλον και ελέγχεται με τηλεχειριστήριο.

Η συνδιασμένη χρήση των αισθητήρων επιτυγχάνεται με τη χρήση πρώτου βαθμού complementary filters για την εκτίμηση της θέσης και δευτέρου βαθμού για την εκτίμηση της ταχύτητας του ρομπότ. Η μέθοδος αυτή χρησιμοποιεί τον προσανατολισμό και τις επιταχύνσεις που λαμβάνονται από το IMU ως είσοδο του συστήματος και εξάγει τη θέση και την ταχύτητα του ρομπόρ στο τρισδιάστατο τοπικό σύστημα αξόνων του ρομπότ.

Πολλά πειράματα διεξάχθηκαν για την επιβεβαίωση της κατάλληλης διαμόρφωσης του συστήματος, καθώς επίσης και της αποδοτικότητας της χρήσης complementary filters. Τα αποτελέσματα εκτίμησης της ταχύτητας είναι αρκετά καλά και αυτά της εκτιμήσης της θέσης ικανοποιητικά, λαμβάνοντας υπόψιν ότι υπάρχει μόνο ένας αισθητήρας που δίνει την απόλυτη θέση του ρομπότ με ακρίβεια τριών (3) μέτρων.

Page intentionally left blank.

State Estimation of a Skid-Steering Mobile Robot using Complementary Filters

Asimenia Z. Laina

Abstract

The objective of this thesis is to estimate the state of a skid-steering mobile robot using complementary filters. The state estimation is achieved through fusing appropriately the measurements received from the sensors attached to the robot. The sensors include an Inertial Measurement Unit (IMU), a Global Positioning System (GPS) and the odometry. The sensor fusion using complementary filters is computationally simple and as such fast, providing at the same time estimation frequency equal to the fastest sensor used. This robot is aimed to be used in an outdoor uneven environment and is controlled by a joystick.

The sensor fusion is achieved by using first order complementary filters for the estimation of position and second-order complementary filters for the estimation of velocity. The fusion uses the orientation and the acceleration obtained from the IMU as an input and outputs the position and the velocity of the mobile robot in the three dimensions on the robot's local frame.

Many experiments were conducted to validate the appropriate configuration of the system as well as, the effectiveness of the sensor fusion using complementary filters. The results of the velocity estimation are really good and those of the position estimation are promising, considering there is only one sensor providing the absolute position with a minimum accuracy of three (3) meters (depending on the surroundings).

Page intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

Page intentionally left blank.

1 INTRODUCTION

The word robot was introduced in 1920 in a play by Karel Capek called R.U.U., or Rossum's Universal Robots. Robot comes from the Czech word robota, meaning forced labour or drudgery. In the play human-like mechanical creatures produced in Rossum's factory are docile slaves. The robots were presented to have the same figure as humans but worked tirelessly. However, it wasn't until the World War II that the first attempt to create wirelessly controlled mechanism of radioactive material happened. In 1961 it was also the first time when the notion of programming of a robot was introduced. In 1962 a robot was developed by Ernst with force sensors capable of storing boxes making it the first robot ever made to work in an unstructured environment. In 1973 the first robot's programming language was developed at Stanford University called WAVE. Through the years the requirements expected from robots were increased, especially from the industry, leading to improvements on the performance of them. During the nineties a huge progress concerning the development of robots occurred, especially in the sections of control algorithms, route planning and use of sensors. After that point, many robots were built; methods were improved leading to the current state. Robotics, as we know it today faces rapid changes, while there is a constant request for better, more accurate, more reliable and more capable robotic systems.[1]

A mobile robot is an automatic machine that is capable of movement in a given environment.[6] Mobile robots can be "autonomous" which means they are capable of navigating in an uncontrolled environment without the need for physical or electro-mechanical guidance devices.[2] Mobile robots are usually divided into two categories of legged and wheeled robots.

Techniques used for position determination of wheeled mobile robots (or simply, mobile robots) are classified into two main groups: relative positioning (position and orientation will be determined using relative sensors) and absolute positioning (techniques are referred to the methods utilizing a reference for position determination).[3]

Calculating position from wheel rotations using the encoders attached to the robot's wheels is called odometry. Although odometry is the first and most fundamental approach for position determination, due to inherent errors, it is not an accurate method.[3] That is one of the reasons why it is so important to fuse data from other sensors in a robust way. In most cases Kalman filter or a derivation of Kalman filter, such as Indirect Kalman filter (IKF), Extended Kalman filter (EKF) and Unscented Kalman filter (UKF) have been used to integrate the information, but more methods are available depending on the application; one of them being Complementary filters which were actually developed well before Kalman filters.

1.1 PROBLEM STATEMENT

The following diploma thesis presents a method on how to estimate the state of a skid-steering mobile robot using complementary filters. The main reason that led to choosing this method is the recent paper from Panos Marantos, Yannis Koveos, and Kostas J. Kyriakopoulos, *"UAV State Estimation using Adaptive Complementary Filters"*[34], in which are presented the satisfying and encouraging results of the complementary filters when used for the state estimation of UAVs (Unmanned Aerial Vehicles). Another reason is that there isn't a lot of research regarding the use of complementary filters in such kinds of problems, although it is a method well-known for several years. The challenge of state estimation is to minimize the errors occurring from the sensors and, fuse measurements in a way that exploits the best features each one has to offer. At the same time it is important to have an estimation frequency as high as the fastest sensor attached to the system.

(a) SUMMIT XL HL, Robotnik[37]

(b) XBOT, DRONYX[38]

FIGURE 1: DIFFERENT SKID-STEERING MOBILE ROBOTS

1.2 SIGNIFICANCE

In a world where the demands for better, more accurate, more autonomous, more capable robots are increasing daily, the significance of estimating the state of them is apparent. The robots must be in a position to answer the question "Where am I?" and perform a series of tasks depending on the answer. In order for them to be implemented in our lifestyles safely, the answer to this question should be accurate enough to eliminate accidents occurred because of their interaction with either humans or the environment. In this thesis a way of estimating the state of a mobile robot is presented. There are many other approaches to this problem and possibly more will pop up in the next years, but the implementation of each method should always take into consideration the kind of problem we are trying to solve and the system the mobile robot will function in. In modern applications, GPS, IMU and odometry won't be enough to provide an accurate state estimation and they are usually used along with other sensors and methods to improve the accuracy of the estimation.

1.3 SIMILAR WORK

There is usually a lot of research when it comes to sensor fusion from GPS and IMU sensors for the estimation of the absolute position of a robot. To improve position estimations, and at the same time maintain a low cost, it is important to combine information from different low-cost sensors using sensor fusion technology instead of using highly accurate but expensive GPS receivers, which would still suffer from limitations such as the inability of acquiring measurements in e.g. a tunnel.[13] One of the most common sensor fusion applications are the different kinds of Kalman filters. Some information is following in order to understand how this method works.

1.3.1 KALMAN FILTERS

"The Kalman filter in its various forms is clearly established as a fundamental tool for analyzing and solving a broad class of estimation problems." Leonard McGee and Stanley Schmidt

The Kalman filter enables estimation of past, present and future states of linear systems by using measurements in a fashion that minimizes the least mean squared error. There are, however, many systems that are nonlinear and in those cases Extended Kalman filter is used, in which the system is linearized around a working point.[13]

The Kalman filter is a recursive algorithm for estimating states in a system utilizing two sorts of information, measurements from relevant sensors and the mathematical model of the system. In the following lines is presented briefly the Kalman filter.

The Kalman filter operates by propagating the mean and covariance of the state through time. Suppose we have a linear discrete-time system given as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{1-1}$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are white, zero-mean, uncorrelated and have known covariance matrices.

Our goal is to estimate the state x_k based on our knowledge of the system dynamics and the availability of the noisy measurements $\{y_k\}$. If we have all of the measurements up to and including time k available for use in our estimate of x_k, then we can form an a posteriori estimate, $\hat{x}_k^+$. If we have all of the measurements before time k available for use in our estimate of x_k, then we can form an a priori estimate, $\hat{x}_k^-$.

It is important to note that $\hat{x}_k^-$ and $\hat{x}_k^+$ are both estimates of the same quantity, x_k.

However, $\hat{x}_k^-$ is our estimate of x_k before the measurement y_k is taken into account, and $\hat{x}_k^+$ is our estimate of x_k after the measurement y_k is taken into account.

Since we don't have any measurements available to estimate x_0, it is reasonable to form the $\hat{x}_0^+$ as the expected value of the initial state x_0:

$$\hat{x}_0^+ = E(x_0) \tag{1-2}$$

We also use the term P_k to denote the covariance of the estimation error.

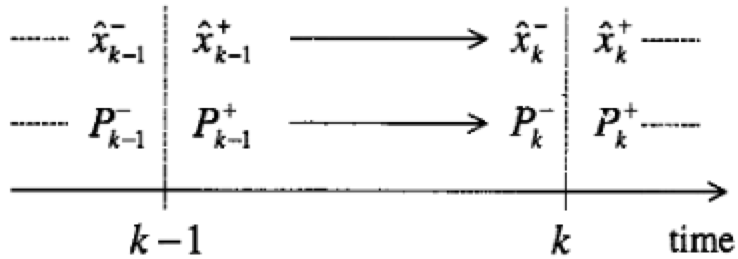

FIGURE 2: TIMELINE SHOWING A PRIORI AND A POSTERIORI STATE ESTIMATES AND ESTIMATION-ERROR COVARIANCES.[7]

We begin the estimation process with $\hat{x}_0^+$, our best estimate of the initial state x_0, and the use of the equation which shows how the mean of x propagates with time.

$$\hat{x}_1^- = F_0\, \hat{x}_0^+ + G_0 u_0 \tag{1-3}$$

In its general form,

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \tag{1-4}$$

Where, $F_k = e^{A\Delta t}$ and $G_k = \int_{t_k}^{t_{k-1}} e^{A(t_{k+1}-\tau)}B(\tau)d\tau$, and P the covariance of the state estimation error, is also being renewed as follows:

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{1-5}$$

Where, $Q_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Q_c(\tau)e^{A^T(t_k-\tau)}d\tau$ and Q_c is the covariance of a continuous-time white noise. For small values, $Q_{k-1} \approx Q_c(t_k)\Delta t$. We have derived the time-update equations for $\hat{x}$ and P and as such, the estimate of a constant x is computed (the estimate changes depending on the availability of the measurement y_k):

$$
\begin{aligned}
K_k &= P_{k-1}H_k^T(K_kP_{k-1}H_k^T + R_k)^{-1} \\
&= P_kH_k^TR_k^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k\big(y_k - H_k\hat{x}_{k-1}\big) \\
P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\
&= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\
&= (I - K_kH_k)P_{k-1}
\end{aligned}
\tag{1-6}
$$

Where, $\hat{x}_{k-1}$ and P_{k-1} are the estimate and its covariance before the measurement y_k is processed, and $\hat{x}_k$ and P_k are the estimate and its covariance after the measurement y_k is processed. Replacing in equation (1-6) $\hat{x}_{k-1}$ with $\hat{x}_k^-$, P_{k-1} with P_k^- , $\hat{x}_k$ with $\hat{x}_k^+$ and P_k with P_k^+, we obtain the measurement update equations [7]:

$$
\begin{aligned}
K_k &= P_k^-H_k^T(K_kP_k^-H_k^T + R_k)^{-1} \\
&= P_k^+H_k^TR_k^{-1} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\
P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \\
&= [(P_k^-)^{-1} + H_k^TR_k^{-1}H_k]^{-1} \\
&= (I - K_kH_k)P_k^-
\end{aligned}
\tag{1-7}
$$

To sum up, the Kalman filter algorithm is explained in words. Firstly, at t_0 the Kalman filter is provided with an initial estimate including its uncertainty (covariance matrix). Next, based on the mathematical model and the initial estimate a new estimate at t_1 is predicted. The uncertainty of the predicted estimate is calculated based on initial uncertainty and the process noise. At t_1 we also have obtained measurements from the sensors which give us new information about the states. Based on the accuracy of the measurements (measurement noise) and the uncertainty in the predicted estimate, the two sources of information are weighed and a new updated estimate valid at t_1 is calculated. The uncertainty of this estimate is also calculated. The algorithm continues for t_{i+1} predicting the new estimate as before, but based on t_i estimate.

This is the basic form of a Kalman filter. There are various others depending on the kind of problem we wish to solve, such as generalised forms and nonlinear ones. Some of them include the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) which are commonly used in robotics.

For more information, one could study the following references: [7], [8], [9].

As you may have noticed, the main disadvantage that arises when using Kalman filters is the computational complexity of this method. For this reason, another method is presented and applied in this thesis using complementary filters.

1.3.2 COMPLEMENTARY FILTERS

A simple estimation technique that is often used in the flight control industry to combine measurements is the complementary filter. This filter is actually a steady-state Kalman filter, also known as Wiener filter. The complementary filter users do not consider any statistical description for the noise corrupting the signals, and their filter is obtained by a simple analysis in the frequency domain.

The Wiener filter solution to this class of multiple-input estimation problems appeared in the literature well before Kalman published his classic paper.[10]

A basic figure of complementary filter is shown below, where x and y are noisy measurements of some signal z, and $\hat{z}$ is the estimate of z produced by the filter. Assuming that the noise of measurement y is mostly high frequency and the noise of measurement x is mostly low frequency, then a low pass filter $G(s)$ can be made to filter out the high-frequency noise in y. If $G(s)$ is low-pass, $[1 - G(s)]$ is the complement filtering out the low-frequency noise in x.

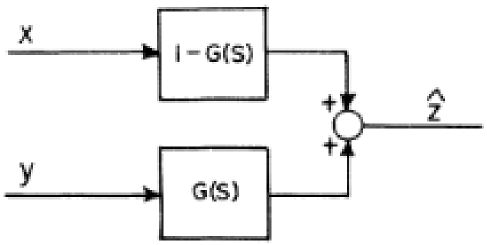

FIGURE 3: BASIC COMPLEMENTARY FILTER. IF G(S) IS A LOW-PASS FILTER, 1-G(S) IS A HIGH-PASS FILTER.[10]

At this point, it should be noted that complementary filters can be used for more than just two measurements, in which case the gains, that measurements are multiplied with, should sum up to no more than one.

A good research on Kalman and complementary filters that focuses on how someone could understand their use is the following: *"Integrated Navigation Systems and Kalman Filtering: A Perspective"*, from R. G. Brown.[11]

Below, follows different research studies on the estimation of the position and the orientation of a mobile robot.

1.3.3 STATE ESTIMATION OF A MOBILE ROBOT

As mentioned earlier, the primary method of estimating the state of a mobile robot is Kalman filter. In the work published from Slawomir Romaniuk and Zdzislaw Gosiewski, *"Kalman Filter Realization for Orientation and Position Estimation on Dedicated Processor"* one can see the implementation of this method. The measurements are obtained from an inertial measurement unit (IMU) and a Global Positioning System (GPS). At first, the measurements from the magnetometer, the accelerometer and the gyroscope are filtered in order to obtain the orientation of the robot and then, using the orientation and the accelerations, all measurements are transferred to a common reference system, where along with the data obtained from the GPS they are all filtered in order to estimate the position. Next, you can see the flow diagrams:

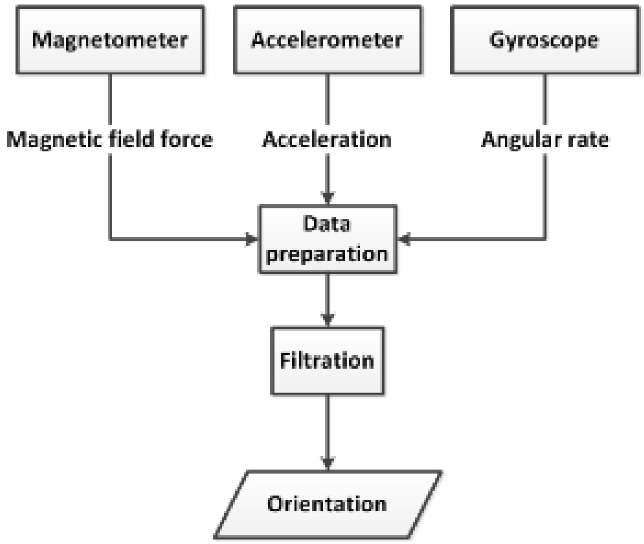

FIGURE 4: INFORMATION FLOW FOR ORIENTATION.[12]

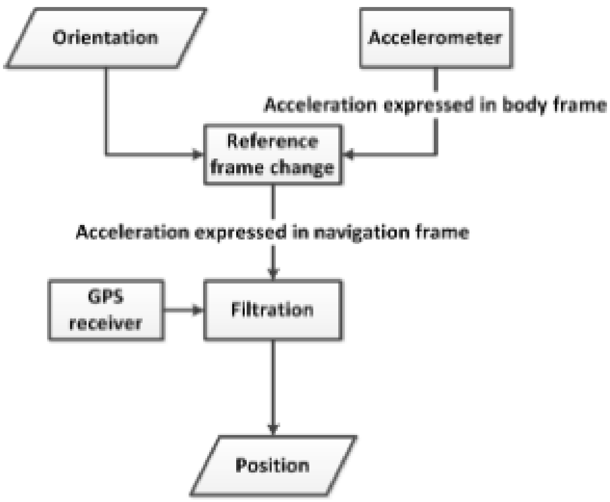

FIGURE 5: INFORMATION FLOW FOR POSITION ESTIMATION.[12]

The main advantage of this approach is achieving higher update frequency rated at about 50Hz (depending on the IMU frequency), which in comparison with 1Hz, characteristic to used GPS receiver, is great enhancement.[12]

There are a lot more different methods of implementing Kalman filter, such as "loosely coupling methods". In a paper though, it is mentioned that "tighter coupling methods should be used for better results.[17] On the other hand, the use of ultra-tight coupling methods is really difficult to approach and are characterised by computational complexity when compared to loosly coupling.[18]

In order to improve the results obtained, another research paper suggests the use of optical system in combination with a map.[14] This method of sensor fusion was developed and tested, showing that the position estimation is possible even when only two satellites are available. It should be noted though, that accuracy had a range from one (1) to ten (10) meters depending on which satellites couples were available.[15]

Another paper[16] suggests the use of a gyroscope, odometry and a map for the better estimation of the absolute position of a robot, approach that has been implemented to self-driving cars from Audi and Alfa Romeo.[16]

There is quite a lot of research when it comes to methods of improving the estimation of position and orientation of a robot in external environments using many other different methods such as rule or fuzzy-based fusion which are implemented along with Kalman filters.[19] Other methods include, lane tracking[20],[21],[22] , map matching[23],[24] , traffic sign localisation[25], [26] , Simultaneous Localisation and Mapping (SLAM)[28], [29] and INS/GPS as mentioned before.

While there are a lot of research papers when it comes to applications of Kalman filters, when it comes to applications of complementary filters the literature is quite limited. Some of them are presented below.

Firstly, the paper from Douglas Guimarães Macharet et al., *"Mobile Robot Localization in Outdoor Environments using Complementary Filtering"*, should be mentioned. In this particular research the use of complementary filters is suggested for outdoors applications, mostly because the position there should be estimated in three dimensions making the use of Kalman filters quite computationally complicated. Consequently, their goal was to find an appropriate model for robotic systems that includes sensors and motors, but is quite quick at start up and easy to calculate. The use of complementary filters was selected because of having low computational cost, faster dynamic responses and simple adjustment of the parameters of the algorithm. However, in this paper the main objective is on estimating the orientation of the robot and not its position. This article presents a localization system for mobile robots based on the complementary filtering technique to estimate the localization and orientation, through the fusion of data from IMU, GPS and compass. The results obtained through this system are compared positively with those obtained using more complex and time consuming classic techniques.[27]

A similar article that studies the use of complementary filters for the estimation of position and orientation is the following: *"Adaptive complementary filtering algorithm for mobile robot localization"*, from Armando Alves Neto et al.. For the experimental confirmation of the results, the Pioneer 3-AT is being used. It should also be noted that the results were quite satisfying and really close to those that came up from the implementation of more complex techniques such as the UKF.[30]

1.4 EXPECTED OUTCOME (WHAT WE DO-RECONSIDER TITLE)

In this Diploma thesis a way of estimating the state of the mobile robot Pioneer 2-AT is presented using complementary filters. The state is being estimated in the three dimensions and the approach used is similar to the one presented in the article, *"UAV State Estimation using Adaptive Complementary Filters"*, from Panos Marantos et al. [34]. The robot is equipped with sensors such as GPS and IMU and with the velocity measurements obtained from odometry the data are used to estimate the position and velocity of the mobile robot. The estimation is achieved by a GPS/INS filter using complementary filters. The robot should be able to know where it is at each given moment when moving in an outdoor environment, where GPS signal is available.

Many experiments were conducted in order to confirm the right configuration of the system as well as to evaluate the results of the filter.

1.5 STRUCTURE

This diploma thesis is divided into 6 chapters:

Chapter 2: In this chapter the problem is stated and the setup is presented along with the details concerning the sensors used.

Chapter 3: In this chapter the approach of solution is presented and explained in detail. Model and sensors are fused to give the estimation of the state of the skid-steering mobile robot.

Chapter 4: In this chapter the different experiments conducted to validate the correct setup of the system and the sensor fusion results are presented. Discussion and conclusions on the results is also provided.

Chapter 5: In this chapter issues for further research and improvements are given.

Chapter 6: In this chapter the function prototyping is explained, ways to use the robot are presented and common troubleshooting is presented.

2 TECHNICAL PROBLEM STATEMENT

2.1 SETUP

The robot used for the modeling of the system is the Pioneer 2-AT, which is property of the Control Systems Lab of the School of Mechanical Engineering, NTUA.

Pioneer is a family of mobile robots, both two-wheel and four-wheel drive. All are intelligent mobile robots, whose client-server architecture was originally developed by Kurt Konolige, Ph.D., of SRI International, Inc. and Stanford University.

ActivMedia's robots are truly intelligent, off-the-shelf mobile platforms, containing all of the basic components for sensing and navigation in a real-world environment, including battery power, drive motors and wheels, position-speed encoders and integrated sensors.

Pioneer 2-AT is a four-wheel drive, skid-steering mobile robot (SSMR) introduced for operation in uneven indoor and outdoor environments, including loose, rough terrain. Each side of the Pioneer AT is electronically and physically linked for evenly applied translational and rotational power and speeds.[5] As such, the two wheels of each side have the same angular velocity at any given moment. One of the motors from each side is equipped with a quadrature encoder with a resolution of 100 pulses per revolution.

The robot is equipped with a 20-MHz Siemens 88C166-based microcontroller, with independent motor-power and sonar microcontroller boards. Pioneer 2-AT also comes with a stall-detection system and inflatable pneumatic tires with metal wheels for much more robust operation in rough terrain, as well as the ability to carry nearly 30 kilograms (66lbs) of payload and climb a 60-percent grade. The maximum speed of the mobile robot is 0.8m/s.[5]

FIGURE 6: PIONEER 2-AT SKID STEERING MOBILE ROBOT.[5]

The robot is programmed in the Robot Operating System (ROS). It can be functioned either through ROS environment or manually using a joystick or by sending commands directly to it.

ROS

ROS is a meta-operating system created by Willow Garage, which provides not only an interface to the sensors and actuators attached to a robot, but allows the implementation of commonly used functionalities such as message passing between processes and package management.[4] It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.[31] The communication with the sensors and actuators is carried out through an IP network. ROS works under a client/server architecture, where one or more robots, with a set of attached sensors and actuators, upload a service to enable a communication channel between a remote client and the sensors. The client program can run on any computer that has a network connection to the robot or team of robots.[4]

The setup of the robot includes the sensors and also a joystick which was used to control the motion of the robot. The IMU attached to the robot is the MTi-G from the Xsens Technologies B.V which also incorporates a GPS sensor.

Following, in Figure 7 one can see the setup of the mobile robot.

FIGURE 7: SETUP OF THE MOBILE ROBOT.

The microprocessor can be seen attached to the middle of the robot and then the joystick used to control the robot, as well as the IMU connected to the satellites receiver. The MTi-G was positioned at the front of the mobile robot along its y axis close to the center of gravity in order to avoid any centripetal acceleration as an effect of the rotations of the vehicle but at the same time be far enough of the robot's motors to remain uninfluenced of the magnetic forces.

In addition, every time the robot boots it creates a Wi-Fi hotspot in which one can connect, access and control it.

As mentioned before the mobile robot used to conduct the sensor fusion is a skid-steering one and as such a mathematical model of a 4-wheel SSMR is presented below. At this point we should mention that the following work was produced by Krzysztof Kozłowski and published under the name *"Modelling and Control of a 4-wheel skid-steering mobile robot"*.[33]

The steering of an SSMR is achieved by differentially driving wheel pairs on each side of the robot. Although the steering scheme yields some mechanical benefits, the control of an SSMR is a challenging task because the wheels must skid laterally to follow a curved path.

Because of lateral skidding, velocity constraints occurring in SSMRs are quite different from the ones met in other mobile platforms where wheels are not supposed to skid. This implies that the control of this robot at the kinematic level only is not sufficient and, in general, demands the use of a properly designed control algorithm at the dynamic level, too.

2.2 MODEL

Here follows a mathematical description of an SSMR moving on a planar surface.

To consider the kinematic model of an SSMR, it is assumed that the robot is placed on a plane surface with the inertial orthonormal basis (X_g, Y_g, Z_g), see Figure 8. A local coordinate frame denoted by (x_l, y_l, z_l) is assigned to the robot at its center of mass (COM). According to Figure 8, the coordinates of COM in the inertial frame can be written as $COM = (X, Y, Z)$. Since in this work the plane motion is considered only, the Z-coordinate of COM is constant$(Z = const)$.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{2-1}$$

Where $\dot{X}$, $\dot{Y}$ denote the velocities, θ is the orientation of the robot and $v = [v_x \quad v_y \quad 0]$ is the vector of linear velocity expressed in the local frame. Also, because of the planar motion one can write $\dot{\theta} = \omega$.

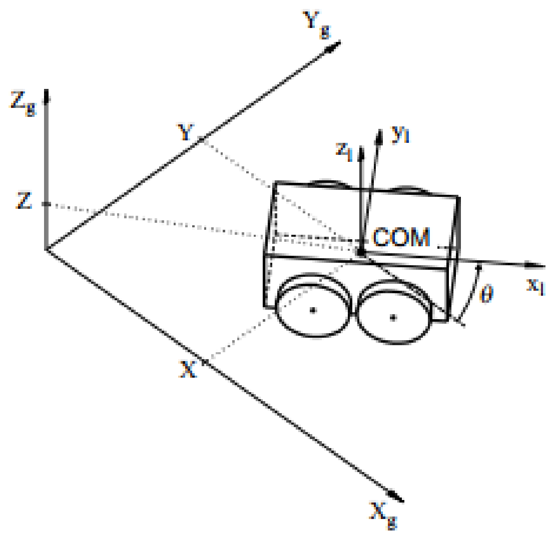

FIGURE 8: SSMR IN THE INERTIAL FRAME.[33]

In this description it is also considered a simplified case of the SSMR movement for which the longitudinal slip between the wheels and the surface can be neglected and thus:

$$v_{ix} = r_i \omega_i \tag{2-2}$$

Where, v_{ix} is the longitudinal component of the total velocity vector v_i of the $i-th$ wheel expressed in the local frame and r_i denotes the so-called effective rolling radius of that wheel. It is also assumed that the effective radius $r_i = r$ for each wheel.

$$\omega_w = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \tag{2-3}$$

Where ω_L, ω_R are the angular velocities of the left and right wheels respectively and v_L and v_R denote the longitudinal coordinates of the left and right wheel velocities.

$$\begin{bmatrix} v_x \\ \omega \end{bmatrix} = r \begin{bmatrix} \dfrac{\omega_L + \omega_R}{2} \\ \dfrac{-\omega_L + \omega_R}{2c} \end{bmatrix} \tag{2-4}$$

Where ω, is the angular velocity.

In Figure 9 a simplified scheme of the drive on the right side of the robot is depicted.

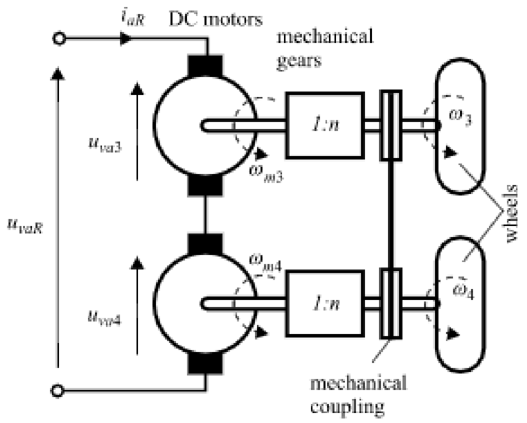

FIGURE 9: DRIVE SYSTEM ON THE RIGHT SIDE OF THE VEHICLE.[33]

2.3 SENSORS

In this section the technical data of the sensors used are being presented.

2.3.1 IMU

The main disadvantage IMUs are facing is that both gyroscope's and accelerometer's measurements drift over time infinitely. That's why we use a sensor fusion in order to improve this error that increases over time and obtain a more accurate estimation of the state of the robot.

As mentioned earlier, the IMU is the MTi-G sensor from Xsens, it is compact, lightweight and is an integrated GPS and MEMS Inertial Measurement Unit with a Navigation and Attitude and Heading Reference System processor. The IMU attached to it is working at a frequency of 100Hz. The device is configurable and has many settings and properties such as, different output modes of the measurements and manual calibration capabilities.

FIGURE 10: MTI-G XSENS SENSOR.[35]

2.3.2 GPS

On the GPS side of the MTi-G, the internal low-power signal processor runs a real-time Xsens Kalman Filter (XKF) providing inertial enhanced 3D position and velocity estimates. The MTi-G also provides drift-free, GPS enhanced, 3D orientation estimates, as well as calibrated 3D acceleration, 3D rate of turn, 3D earth-magnetic field data and static pressure (barometer).[32] However, it should be noted that for our application the Xsens Kalman Filter was disabled and raw GPS data were acquired in order to feed the complementary filter.

The GPS provides the absolute position of it, as well as the velocity in the three dimensions at a frequency of $10Hz$. The minimum error of this GPS in the x and y direction is 3 meters but, depending on the place and the surroundings it could sometimes reach 11 meters. When it comes to the altitude the error is even more making the GPS unreliable to use. This could be solved by also obtaining the pressure measurements of the IMU and fuse them with the altitude measurements of the GPS in order to improve the z axis accuracy. In this thesis, only the GPS's altitude measurements are taken into account. The antenna used for the GPS capabilities of the MTi-G was provided by the Control Systems Laboratory (CSL) instead of the one provided by the supplier of the sensor, in order to improve the reception between the buildings where some of the experiments took place.

FIGURE 11: GPS ANTENNA PROVIDED FROM THE CSL.

We should also mention that despite the fact that, the GPS is the only sensor that measures the absolute position and velocity, the need of a sensor fusion scheme arises because of the low update rate, lack of accuracy, introduced lag especially in velocity measurements and possible loss of signal.[34]

2.3.3 ODOMETRY

The robot has an optical Quadrature Encoder which is used to translate the angular position or movement into analog or digital signal. It produces two rectangular pulses with a phase difference of 90°. The robot is also equipped with an Odroid U2 computer with 1.7GHz computational power and 2GB of RAM. It is capable of performing the communication between the micro-controller and at the same time, communicating with another computer connected to the network. The robot has a PI control and has been programmed to perform the calculations it needs at specific time intervals. The encoder of the robot has a resolution of $100\left[\frac{clicks}{rev}\right]$. The robot is equipped with a speed reducer and has a reduction ratio of $n_{pulley} = \frac{25}{20} = 1.25$. As such the maximum angular speed of the motor is:

$$\omega_{motor} = \frac{v_{max}}{r} \cdot n \cdot n_{pulley} = 590.09\left[\frac{rad}{s}\right] \tag{2-5}$$

The maximum frequency of the pulses is:

$$f_{max} = \frac{\omega_{motor}}{2\pi} enc_{res} = 9391.85[Hz] \tag{2-6}$$

The controller of the robot can handle this frequency without any issues.

Having presented the kinematic model of the mobile robot and the sensors attached to it, we should also mention what are the inputs and outputs of our system when estimating the state of the robot.

2.4 SYSTEM'S INPUTS/OUTPUTS

The system's inputs are the orientation, the angular velocity and the acceleration in three dimensions of the robot as obtained from the IMU attached to it. The system's outputs, as obtained from the sensor fusion, are the position of the robot and the linear velocities in the three dimensions xyz of the robot's local frame. The state estimation package for the 2at mobile robot prints and updates the following values:

Orientation	$phi, theta, psi$ (obtained directly from the IMU)
Position	x, y, z (obtained from the sensor fusion)
Velocity	u, v, w (obtained from the sensor fusion)

In the next chapter, the way the sensor fusion was applied will be presented and analysed.

3 APPROACH OF SOLUTION

In this chapter we address the complete state estimation problem of a skid-steering mobile robot, while using the low-cost sensors presented before with bias variations and higher levels of noise. The estimation of the position and the velocity is achieved by complementary filters combining the various sensors.

The frame in which the sensor fusion takes place is the mobile's local frame expressed in the North East Down (NED) coordinates. All measurements received from the sensors are expressed in this frame before any other calculations are performed.

FIGURE 12: LOCAL NED FRAME OF SSMR.

The preparation of the measurements is presented below, data are transferred to the appropriate frame and then, the sensor fusion process is presented and analysed.

3.1 GPS

It should be mentioned that the MTi-G has a built in Kalman filter to fuse the measurements from the IMU and the GPS, which was disabled in order to receive the raw GPS data that are needed for our application. The MTi-G was configured in a way that provides us with the raw GPS measurements.

GPS position values are obtained in Latitude (φ), Longitude (λ) and Altitude (h), then transferred into the Earth Centered Frame (ECEF - Earth-Centered Earth-Fixed) and finally transferred into the robot's NED frame. The x, y, z axes of the robot will match the local NED axes. The longitude measures the rotational angle (ranging from $-180°$ to $180°$) between the Prime Meridian and the measured point. The latitude measures the angle (ranging from $-90°$ to $90°$) between the equatorial plane and the normal of the reference ellipsoid that passes through the measured point. The height (or altitude) is the local vertical distance between the measured point and the reference ellipsoid. This is the Geodetic Coordinate System.

The ECEF coordinate system rotates with the earth around its spin axis. As such, a fixed point on the earth surface has a fixed set of coordinates. The origin and axes of the ECEF coordinate system are defined as follows:

- The origin is located at the center of the Earth.
- The $Z - axis$ is along the spin axis of the earth, pointing to the north pole.
- The $X - axis$ intersects the sphere of the earth at 0° latitude and 0° longitude.
- The $Y - axis$ is orthogonal to the $Z -$ and $X - axes$ with the usual right-hand rule.[40]

The NED frame is defined as follows: the "North" axis points North in the local meridian direction and the "East" axis points East in the local parallel direction.

These directions span a Cartesian plane on the Local Tangent Plane (LTP). The final, "Down" axis is perpendicular to the other two axes and points towards the Earth, to complete a right-handed coordinate system. Note that the "Down" axis doesn't point to the center of the Earth, but is defined by the other two axes and its direction depends on the latitude and longitude of the origin of the NED frame.

The origin of the NED frame is fixed in ECEF coordinates.[41] It can be chosen arbitrarily at a point on the surface at the operational site of the robot.

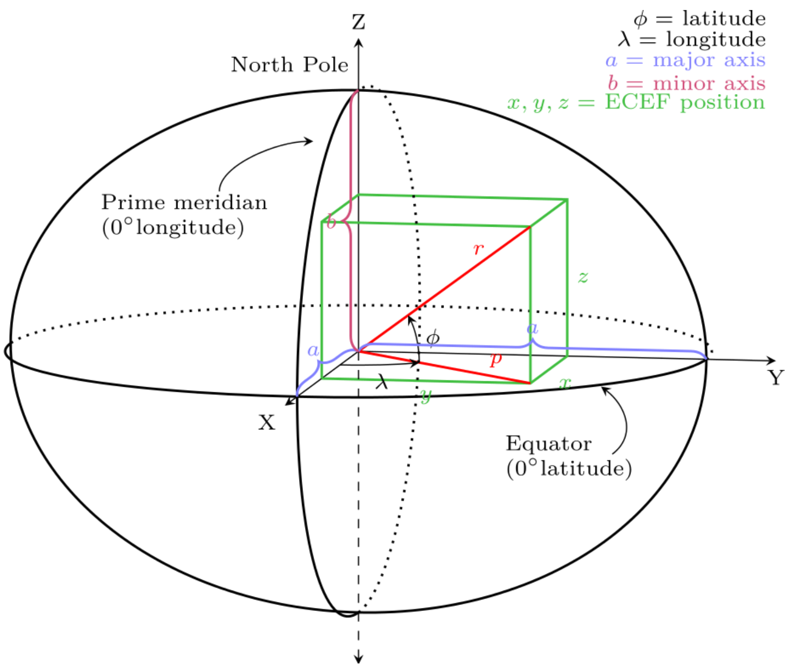

FIGURE 13: EARTH CENTERED, EARTH FIXED COORDINATES IN RELATION TO LATITUDE AND LONGITUDE.[39]

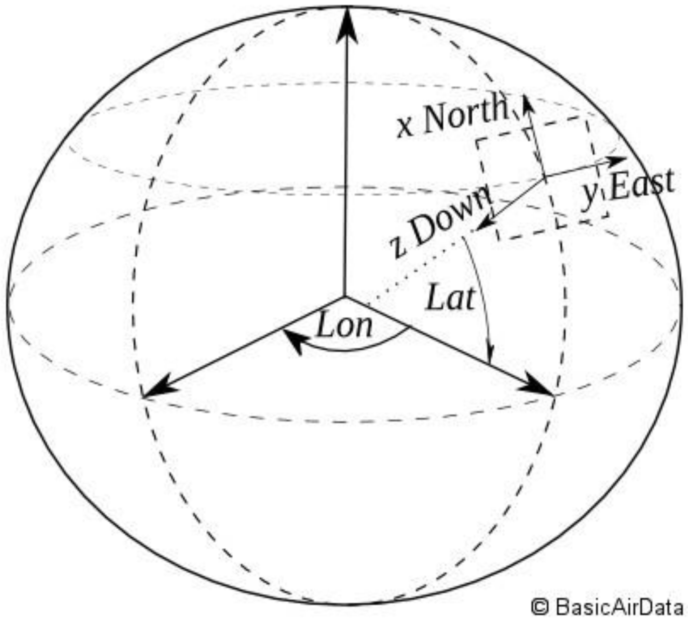

FIGURE 14: NED FRAME.

Below, is presented the process of the aforementioned transformations.

The earth centered coordinates are obtained as follows:

$$
\begin{aligned}
x_{ecef} &= \left(\frac{R_{ea}}{\sqrt{1-f^2\sin^2(lat)}} + alt\right)\cos(lat)\cos(lon) \\
y_{ecef} &= \left(\frac{R_{ea}}{\sqrt{1-f^2\sin^2(lat)}} + alt\right)\cos(lat)\sin(lon) \\
z_{ecef} &= \left(\frac{(1-f^2)R_{ea}}{\sqrt{1-f^2\sin^2(lat)}} + alt\right)\sin(lat)
\end{aligned}
\tag{3-1}
$$

Where, $R_{ea} = 6378137m$ is the semi-major axis of earth and, $f = 0.003352810664747$ is the earth flattening, lat, lon, alt are latitude, longitude and altitude respectively.

Now, from the ECEF we can transfer the coordinates to the robot's local NED:

$$
\begin{aligned}
x = -sin(lat)\cos(lon)\left(x_{ecef} - x_{ecef,0}\right) \\
\qquad - \sin(lat)\sin(lon)\left(y_{ecef} - y_{ecef,0}\right) \\
\qquad + \cos(lat)\left(z_{ecef} - z_{ecef,0}\right) \\
y = -\sin(lon)\left(x_{ecef} - x_{ecef,0}\right) + \cos(lon)\left(y_{ecef} - y_{ecef,0}\right) \\
z = -\cos(lat)\cos(lon)\left(x_{ecef} - x_{ecef,0}\right) \\
\qquad - \cos(lat)\sin(lon)\left(y_{ecef} - y_{ecef,0}\right) \\
\qquad - \sin(lat)\left(z_{ecef} - z_{ecef,0}\right)
\end{aligned}
\tag{3-2}
$$

Having obtained the position on the NED frame of the robot, the GPS data are ready to be used in the sensor fusion process.

3.2 IMU

The IMU is providing the orientation of the robot as long as the acceleration and the angular velocity of it in three dimensions, all being used as inputs. The orientation of the robot is obtained in quaternions and is then translated into Euler angles, which are easier to be used when the application is a mobile robot. The IMU is located at the front of the robot and the data are obtained at the NED frame of the sensor, so the measurements received are being transferred to the NED frame at the robot's center of mass.

The quaternions are translated into Euler angles:

$$\varphi = \tan^{-1}\left(2\frac{q_0 q_1 + q_2 q_3}{1 - 2(q_1^2 + q_2^2)}\right)$$

$$\theta = \sin^{-1}\big(2(q_0 q_2 - q_3 q_1)\big) \qquad (3\text{-}3)[35]$$

$$\psi = \tan^{-1}\left(2\frac{q_0 q_3 + q_1 q_2}{1 - 2(q_2^2 + q_3^2)}\right)$$

The acceleration is being transferred to the body frame as follows:

$r_{imu} = (0.212, 0.0, 0.0)$, is the position of the IMU in relation to the robot's center of mass.

$a = a + \omega \times (\omega \times r_{imu})$, acceleration transferred to body frame.

The acceleration in the z direction (gravity), after the first iteration is being calculated as the mean value of the previous ones, providing better accuracy.

Lastly, first position is set as the reference point, and the route of the robot is expressed in relation to the first point.

3.3 ODOMETRY

The encoders of the robot are providing measurements of angular velocity of the left and right wheels and encoder measurements for left and right wheels as well. Following are the equations being used to translate those measurements into the angular velocity of the robot and the orientation of it. The velocities obtained from odometry are used along with those from GPS for the x direction.

$R = 0.11m$, measured radius of robot's tires.

$d = 0.1905 \cdot 1.63m$, geometric characteristic of skid-steering mobile robots, as presented in figure 15.

$encres = 8187.5$

$looptime = 15 \cdot 10^{-3}s$

The right and left angular velocities are calculated through the encoders' measurements:

$$\omega_L = \frac{2\pi\left(\frac{encoder_L}{encres}\right)}{looptime}$$

$$\omega_R = \frac{2\pi\left(\frac{encoder_R}{encres}\right)}{looptime}$$

The angular velocity of the mobile robot: $\omega = -\frac{(\omega_R - \omega_L)\cdot R}{2\cdot d}$

The linear velocity of the mobile robot: $u = \frac{(\omega_R + \omega_L)\cdot R}{2}$

In the state estimation code it has also been implemented a way of calculating the robot's position using the odometry but this way has not being used in the state estimation, as it would add more errors to the calculation.

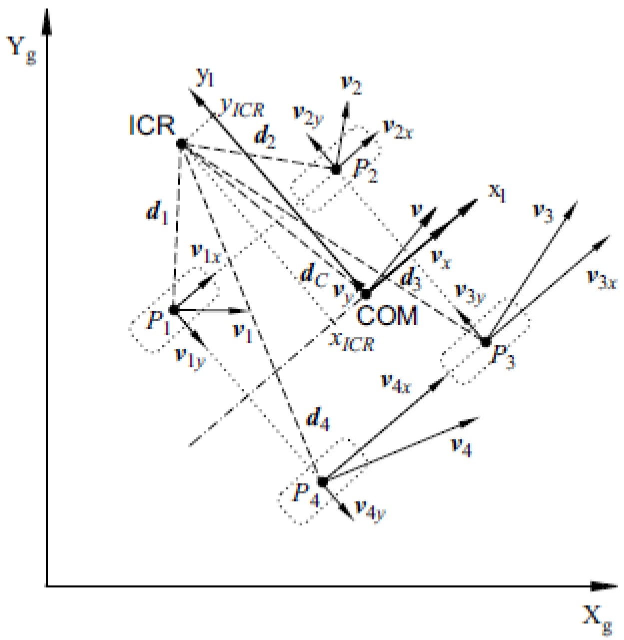

FIGURE 15: WHEEL VELOCITIES

3.4 SENSOR FUSION

Having transferred all the measurements into a common coordinate frame we are ready to implement the sensor fusion in order to estimate the state of the robot. As mentioned to a previous chapter the method used is complementary filters.

An observer fusing the data obtained from the different sources using complementary filters is a good approach to the state estimation problem. In a later step this approach would be also helpful in the use of a controller for the robot.

The observer used is based on the "*UAV State Estimation using Adaptive Complementary Filters*"[34] reference, and is estimating the translational velocity and accelerometer biases using second-order complementary filters and first order for the position estimation. This is implemented by fusing the GPS, the IMU and the odometry. Below follows the GPS/INS observer:

$$\begin{aligned}
\hat{p}_N^k &= \hat{p}_N^{k-} + k_{gps}^{pN}\left(p_N^{gps,k} - \hat{p}_N^{k-}\right) \\
\hat{p}_E^k &= \hat{p}_E^{k-} + k_{gps}^{pE}\left(p_E^{gps,k} - \hat{p}_E^{k-}\right) \\
\hat{p}_D^k &= \hat{p}_D^{k-} + k_{gps}^{pD}\left(p_D^{gps,k} - \hat{p}_D^{k-}\right) \\
\hat{v}_I^k &= \hat{v}_I^k + \boldsymbol{K}_{gps}^{v}\left(v_I^{gps,k} - \hat{v}_I^k\right) \\
\hat{b}_{a,I}^k &= \hat{b}_{a,I}^{k-} - \boldsymbol{K}_{gps}^{a}(v_I^{gps,k} - \hat{v}_I^k)
\end{aligned} \tag{3-4}$$

Where, $\hat{p}_I^{k-} = \hat{p}_I^{k-1} + \Delta t \hat{v}_I^{k-1}$, $\hat{v}_I^{k-} = \hat{v}_I^{k-1} + \Delta t(\hat{R}_{k-1}^T a_B^{y,k-1} + g - \hat{b}_{a,I}^{k-1})$ and $\hat{b}_{a,I}^{k-} = \hat{b}_{a,I}^{k-1}$ are the priori estimation of the state x, $k_{gps}^{p_N}$ and $k_{gps}^{p_E}$ are positive gains derived from the selected low-pass cut-off frequencies of the GPS position measurements. $\boldsymbol{K}_{gps}^{v}, \boldsymbol{K}_{gps}^{a}$ are positive diagonal matrices which are derived from the selected high-pass cut-off frequencies and damping factors of the vehicle acceleration.

It should be noted that, if there is no GPS available, the MTi-G cannot make a reliable estimation of position or velocity. In a different approach it could be chosen to estimate the xy position and the u velocity using the odometry, but in this thesis this wasn't chosen because it would add extra errors to the estimation, as the position estimation suffers from many errors. Though, as mentioned in the *3.3 Odometry section* above, the code has been implemented in the state_estimation_2at.py file for future reference and/or use. First, in order to estimate the position, the velocity has to be integrated and then, calculate the new position based on the previous one, meaning the errors are magnified from the one time step to the next one.

3.5 S/W ENGINEERING

In this chapter the function coded and used in the ROS package to estimate the state of the mobile robot is presented and explained in detail.

In the previous chapters a presentation of how the measurements were handled before being used in the state estimation was made, and a brief explanation of the complementary equations that were used was also described. Here, a more detailed explanation of the code and the function *StateEstimation* will be presented and explained.

Firstly, the necessary initial values are being defined. For the first measurement the biases are set to zero and the state estimation takes place using only the odometry measurement for the velocity (GPS signal not still available) and the position is forced to zero. In this last case it is assumed that the robot is stationary when the first state estimation is being made. Though, it should be noted that in the code has also been implemented a way of using the GPS values right from the start, but in that case it would be safer to use a delay before calculating the first estimation, as it might take a while for the GPS to find satellites and give the first measurements.

Having initialised, time and first position then the state estimation function can use the previous measurements to estimate and approximate more accurately the mobile's state.

For the orientation, as mentioned earlier, only the IMU measurements are being used. The first Euler angles are also initialised and then subtracted from any other measurement so as to give the change of orientation comparing to the initial state.

Lastly, in case of the GPS losing its signal, a message is presented to the user informing of this situation. The estimation of the velocity then takes into consideration only the velocity obtained from the odometry. The estimation of the position only takes into consideration the a priori estimation which is based in the robot's kinematic model.

4 RESULTS

In this section, a summary of the experiments and the results is going to be presented as long as the different scenarios that were taken into consideration and the check tests that were performed in order to ensure that all the data received were accurate and that sensors were tuned in the right way. There were performed many tests and experiments but only the main ones will be extensively presented and the rest will be mentioned in short. Some of the experiments were performed outdoors when GPS data were needed and some were conducted indoors for the check and regulation of the rest of the sensors and hardware.

4.1 EXPERIMENTAL SETUP

In the different experiments performed, when control of motion for the robot was needed, the joystick was used. When it came to the indoors ones all sensors except the GPS were up and running and for the outdoor, all sensors were active. The indoors experiments were performed in the Control Systems Laboratory, while the outdoors were performed in different places around the university campus depending on the purpose of the experiment and will be specified later on.

The purpose of the first test that was conducted was to identify that everything is working properly. In this test it was made clear that the way one can connect to the robot wasn't practical and as such the robot was programmed to create its own Wi-Fi hotspot upon booting making it easy for the handler to connect and control. One more test was also conducted to check the behavior of the joystick. It seemed that the response of it was quite aggressive and the gains were readjusted to help the control of the robot's movement become smoother.

Other tests that were performed indoors included checking the accuracy of the orientation data and how they might be influenced by the IMU being positioned near the motors of the robot.

Firstly, the robot was turned off and positioned in a known direction. Only the IMU was working at this point and the values of the quaternions were obtained. As a next step, the robot was turned on but stationary at the same orientation and the same measurements of the IMU were recorded. Lastly, the robot was moving in the direction of the orientation specified in order to ensure that at worst case scenario (full power of motors) the influence of the magnets to the IMU was negligible. These actions were repeated several times to ensure that many data were available to reach a safe conclusion. After obtaining all the measurements, the data were compared and there was no more than 3 degrees difference with the motors on and running. That ensured the right set up of the IMU on the robot.

The next step was to ensure the accuracy of the orientation obtained from the IMU, because this was the only sensor providing measurements of orientation. There were performed many in place rotations around the robot's axis and checked with pre-specified points to make sure the rotations were properly measured. The results were really good and there was no need for any further checking of the orientation of the robot.

There were conducted two more experiments with all the systems up and running, the purpose of the first one was mostly to check that all the transformations used in the filter were correct and to also get an approximate estimation of how well the filter works. The second one was a ground-truth experiment.

The two aforementioned experiments will be presented in detail below.

4.2 VERIFICATION OF SYSTEM SETUP AND SENSOR FUSION BEHAVIOUR

4.2.1 SCENARIO

The first scenario was conducted on the parking space of the School of Mechanical Engineering. The path was drawn on the ground using chalk and measured with a measurement tape. The sensors were given time firstly, to heat up and secondly, in order for the GPS to find the satellites and reduce its error at 3 meters. The robot was controlled through the joystick over the path twice to check the data it was giving us. The main reason for this experiment was to check that the transformations used to transfer all measurements to body frame were correctly coded and calculated.

4.2.2 RESULTS

In the next figure one can see how the path looked like. The shape is like a flag in order to force the robot to turn 90 degrees each time and as such have a clear understanding of how the measurements should look like. Following one can see, the velocities:

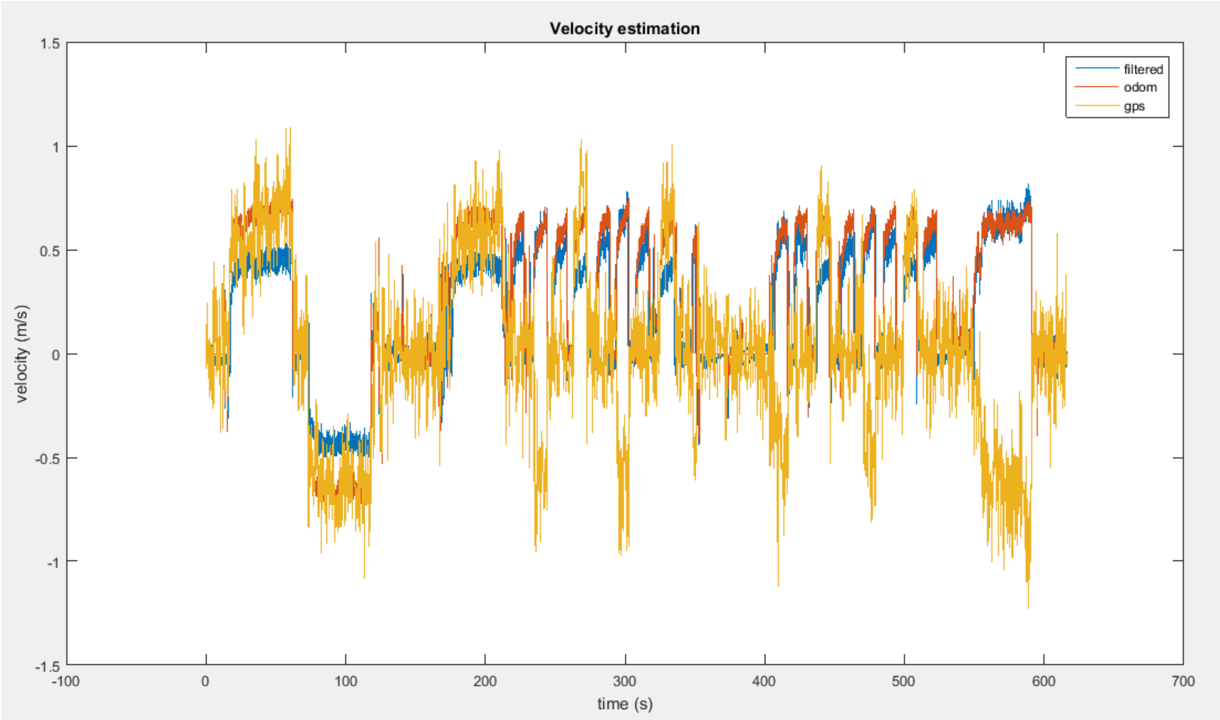

FIGURE 16: VELOCITY MEASUREMENTS AND VELOCITY ESTIMATION.

Forward and backward movements seem to be correctly aligned. In the cases where one can see that GPS and odometry measurements are in an opposite direction coincide with the case where the accuracy of the GPS is really bad and the filter recognizes that and uses only the odometry measurements for this period of time. In the following picture one can note this behavior.

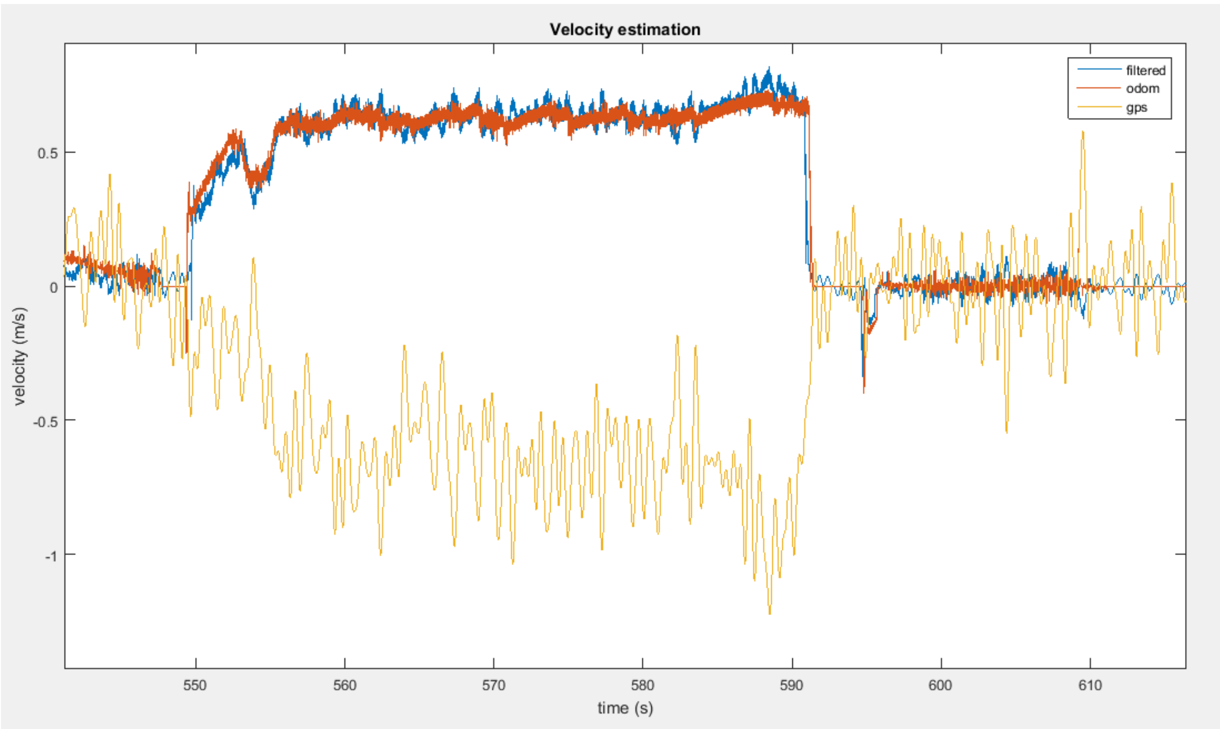

FIGURE 17: FILTER BEHAVIOR UNDER BAD GPS ACCURACY.

One more interesting observation in this velocities graph is the fact that the filter can really well calculate the zero velocities, where the GPS usually gives really inaccurate measurements. The orange line is the odometry measurement which we know is completely accurate at zero velocities, and it is noticed that the estimated (filtered) velocity is around zero as well (blue line).

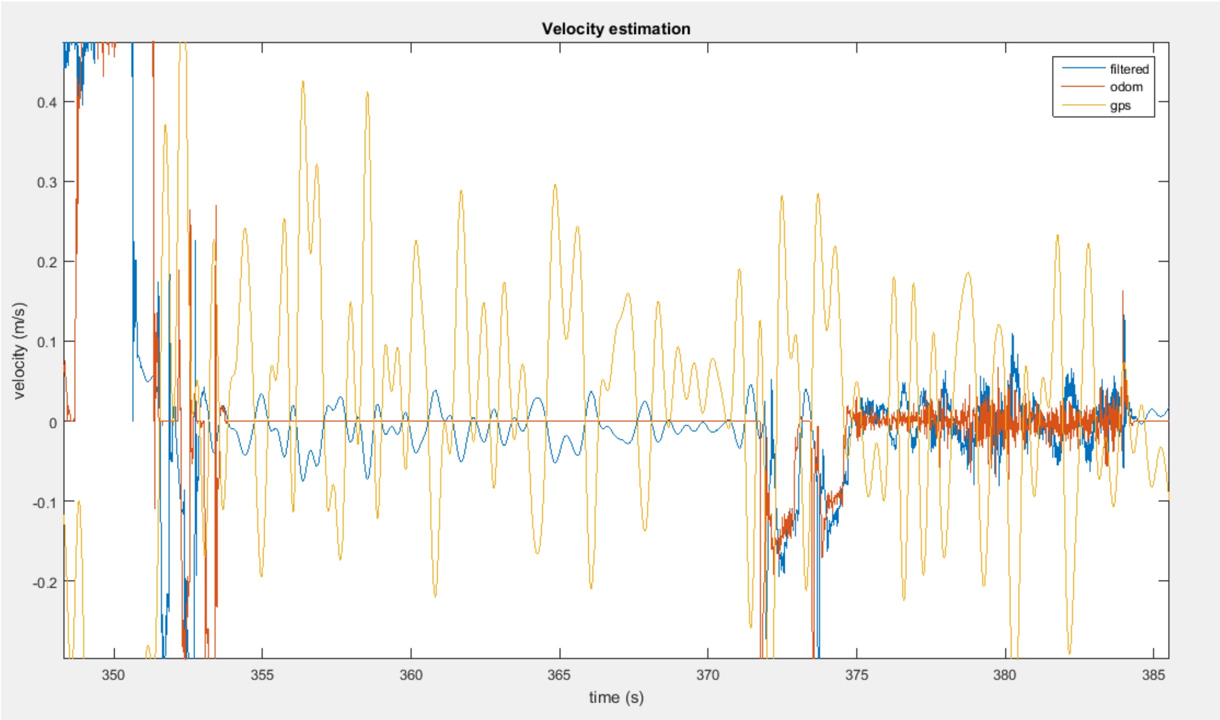

FIGURE 18: FILTER BEHAVIOR ON ZERO VELOCITY.

In the following graphs is presented the position of the robot.

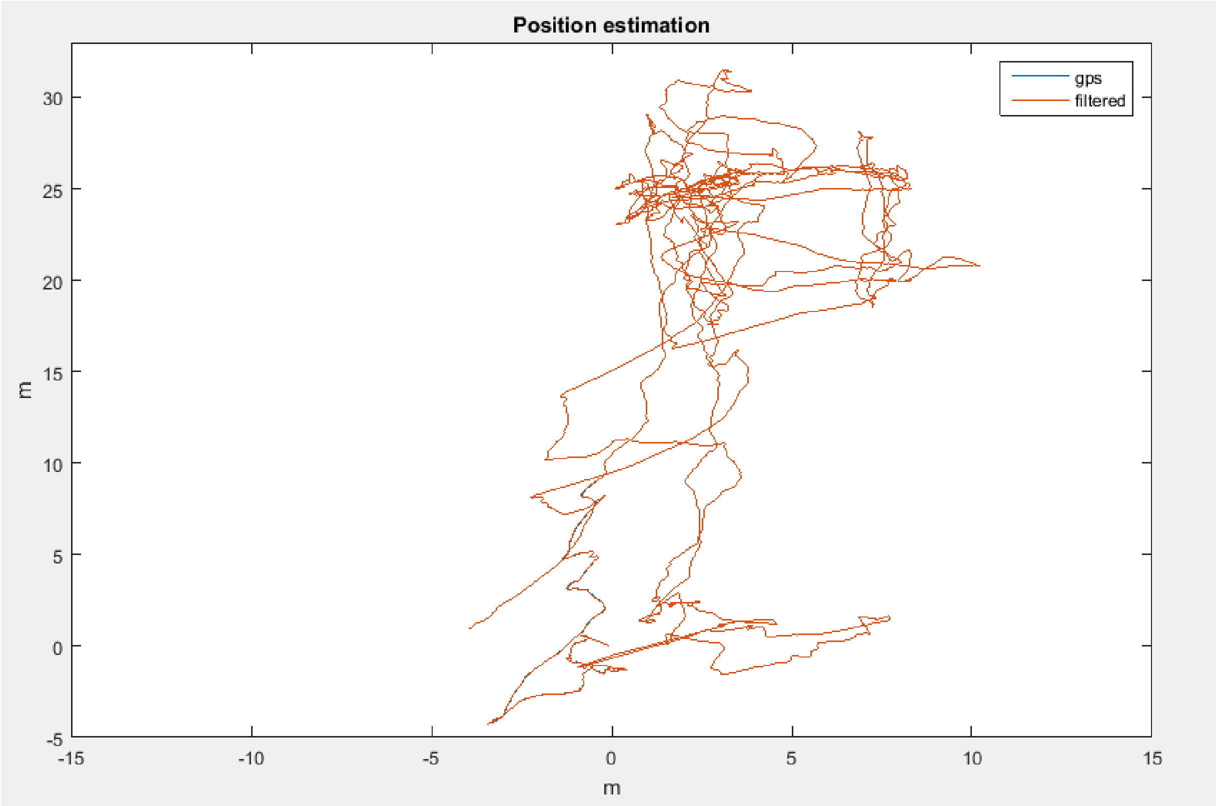

FIGURE 19: GPS AND ESTIMATED ROUTE OF THE ROBOT

The drawn route was as presented below:

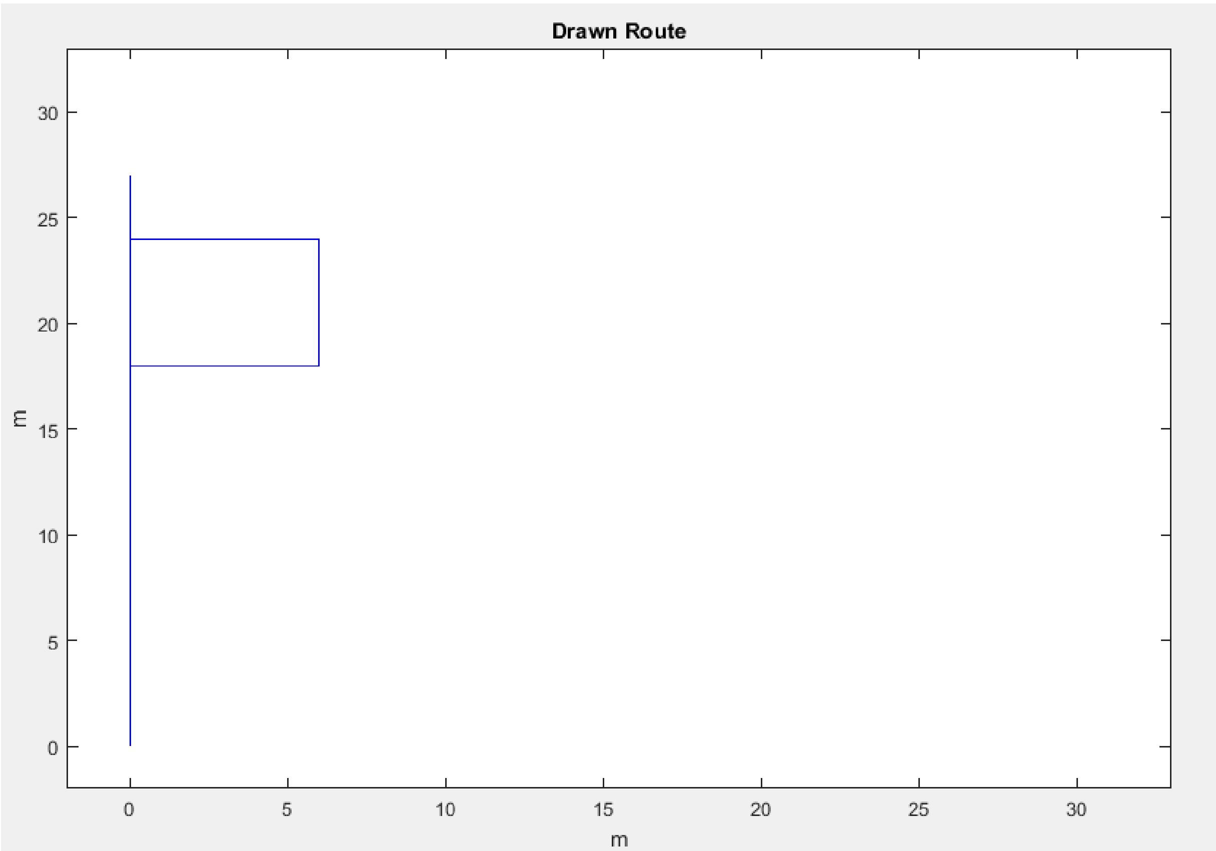

FIGURE 20: DRAWN ROUTE OF EXPERIMENT

The experiment was successful since everything was configured correctly, but it can be seen that the GPS has big inaccuracies leading to not that accurate results. One more thing that was noticed and will be apparent in the Ground Truth experiment is the fact that the robot isn't following the joystick commands as smooth as it should leading to a difficult control of its route.

4.3 GROUND TRUTH

4.3.1 SCENARIO

The last experiment performed was the one to decide the overall accuracy of the sensor fusion. It should be noted that the 3 meters accuracy could not be reduced as this is the accuracy of the only sensor providing the absolute position. The reason sensor fusion is used is only in order to minimize the errors of the different sensors, such as drift or non-zero velocity provided by the GPS as explained earlier.

The experiment took place at the university's gym court. The terrain is rough consisting of soil and small rocks. This potentially increases the difficulty for the robot to be driven and the skid-steering phenomena. The experiment setup was consisted of 24 checkpoints as displayed below. The route designed for the robot to follow is a parallelogram of the following dimensions, $28m \times 12m$. Checkpoints (1) through (4), (6) through (16) and (18) to (24) are positioned in 3 meters linear distance between them, while checkpoints (4) through (6) and (16) through (18) are positioned in 5 meters linear distance between them. The distance between those points was chosen to be at least the minimum accuracy of the GPS. The robot was controlled through the joystick and the experiment was recorded by a camera attached to a drone.

FIGURE 21: SETUP OF EXPERIMENT. THE DRONE RECORDING THE EXPERIMENT CAN BE SEEN AS A SHADOW IN THE PICUTRE. PIONEER 2AT AND TWO CHECKPOINTS CAN ALSO BE SEEN.

The Pioneer 2-AT was driven from the one checkpoint to the next, making a stop to each one. The reason why, this way was chosen to perform the experiment is because that way we could check during the post process where the robot is, as the velocity obtained from the odometry would be zero.

Before the experiment started, it was given some time to the sensors to heat up and to the GPS to find the satellites and reduce its error at four (4) meters.

FIGURE 22: SENSORS BEING PREPARED, ROBOT DRIVEN TO ITS START POINT.

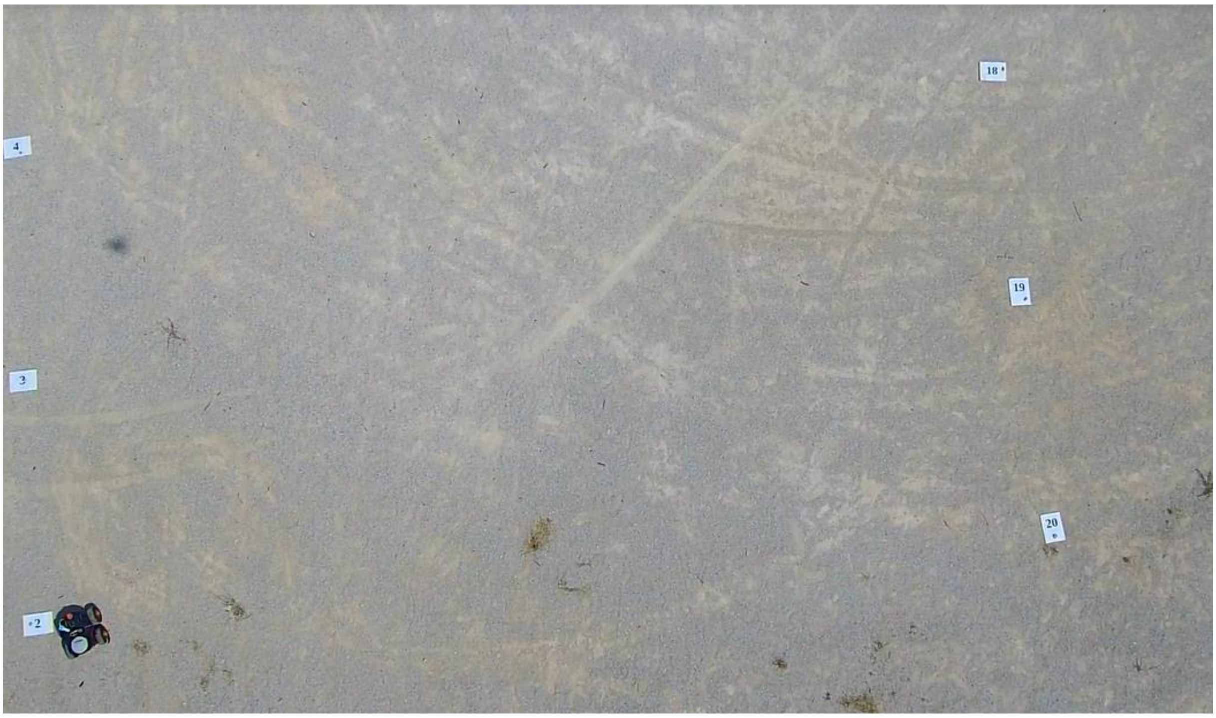

FIGURE 23: SETUP OF EXPERIMENT, WHERE SOME CHECKPOINTS ARE VISIBLE.

FIGURE 24: CAMPUS'S GYM COURT WHERE THE EXPERIMENT TOOK PLACE.

4.3.2 RESULTS

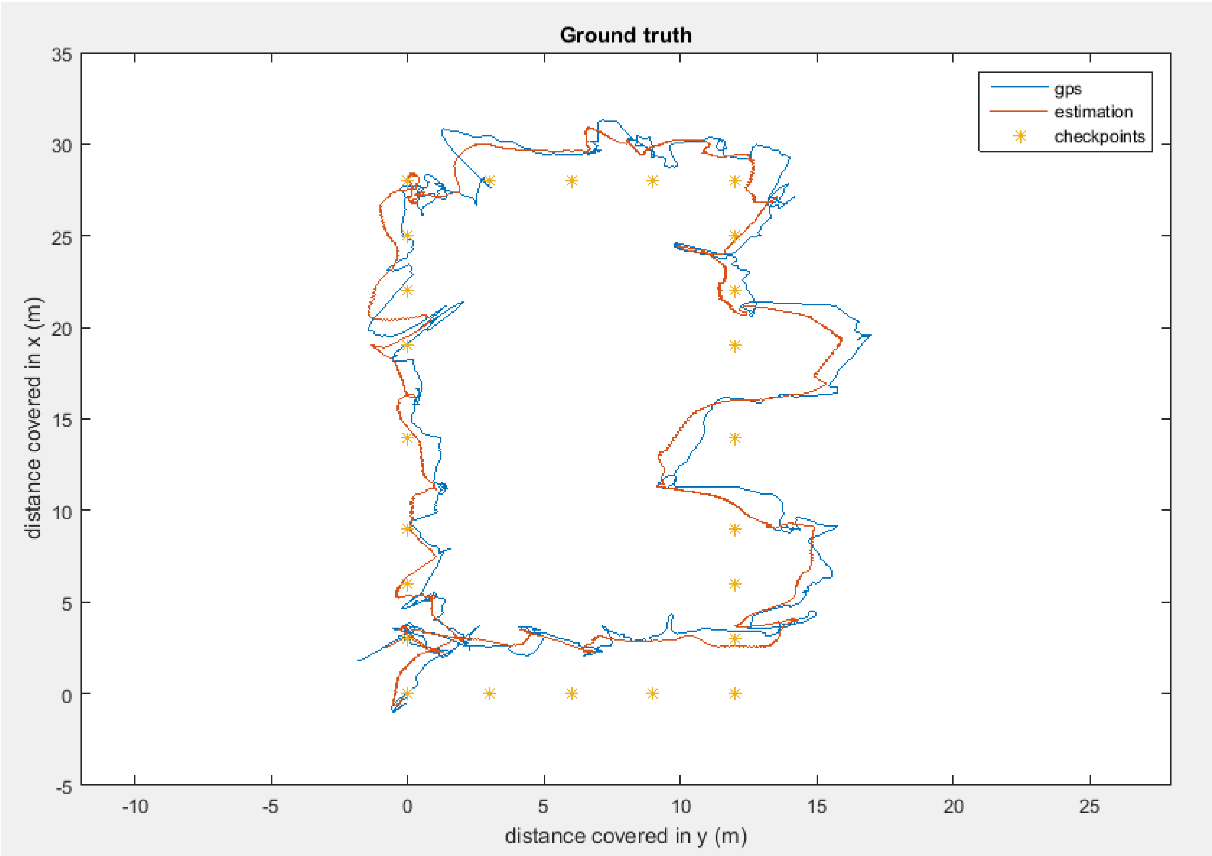

FIGURE 25: ROUTE OF THE ROBOT, AS MEASURED FROM THE GPS AND AS CALCULATED USING COMPLEMENTARY FILTERS.

As mentioned earlier the robot couldn't be properly controlled using the joystick. As a result, when someone is looking at the position graph and has in mind the rectangular setup of the route might come to the conclusion that the state estimation is quite inaccurate. The fact is that the robot was actually following this weird non linear path, as it was unable due to its PID control to follow the commands properly. Sometimes though, the GPS inaccuracy was increased to 8 meters and in these cases as can be seen in the video the robot was kept stationary longer to reduce the error.

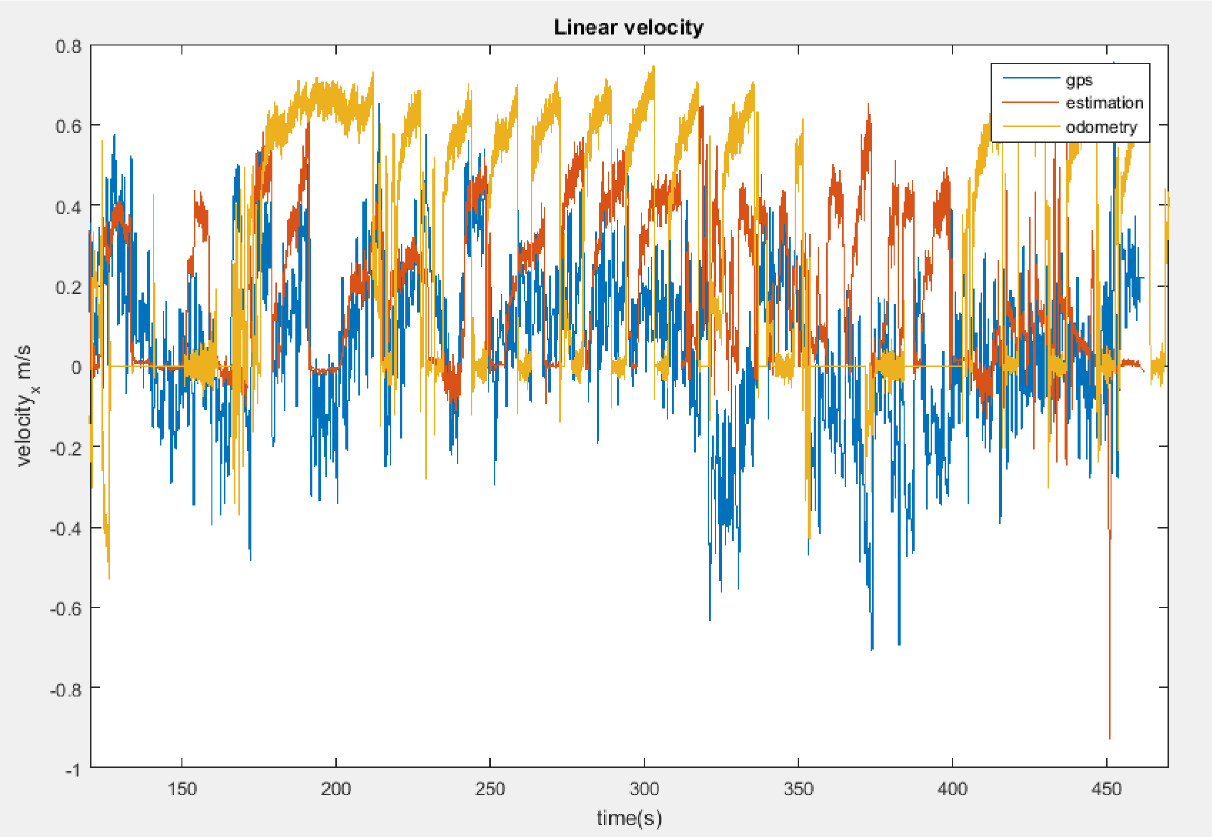

FIGURE 26: VELOCITY OF THE ROBOT, AS MEASURED FROM THE GPS, THE ODOMETRY AND AS CALCULATED FROM THE SENSOR FUSION.

In this experiment the velocity seems to be more inaccurate and the truth is that on the surroundings where the experiment took place the GPS was suffering from big errors. In the post process the gains were changed to see whether better results could be achieved but it only made the results worse. Though, the zero value of the velocity is still really well estimated from the sensor fusion.

It becomes obvious that the setup of the robot and sensors being used need to be changed or improved. In a different application maybe the estimation of the position could also take into consideration the odometry. Lastly, GPS sensors have also been improved in the last years providing better accuracy, but they still remain inaccurate depending the application they are going to be used.

There are many ways this method could be improved and many of them were presented in the introduction of this thesis. The use of complementary filters though seems to give a good estimation without the complexity of using Kalman filters.

4.4 DISCUSSION

Having performed the Ground truth test a few things became obvious. It is important to take into consideration the facts the led to the previously presented results. Firstly, the robot wasn't able to be controlled properly during the experiments leading to misleading assumptions of the estimation being inaccurate. The estimation seems to be giving really good results, and this can be clearly seen in the velocity graphs. When it comes to position estimation, the only sensor that measures absolute position is the GPS which is known to have at least 3 meters inaccuracy and this cannot be avoided unless a more accurate sensor is used for absolute position measurements. It is important to keep this in mind while evaluating the results.

5 ISSUES FOR FURTHER RESEARCH

Until this point it should be clear that without a more accurate sensor we wouldn't be able to achieve more accurate results. The main advantage of the proposed method is the reduction of the computational complexity of the overall system, which enables increasing the sampling frequency of the measurements with a consequent improvement in the accuracy of the estimates. Another, advantage is that drift can be also measured and estimated minimizing the errors in the y direction.

When it comes to estimating the position z direction, the GPS suffers from large amount of errors. In that case a pressure measurement fused with the GPS measurement would result in a better estimation of altitude.

The sensor fusion process could also be adjusted to provide state estimation, even when no GPS data are available using the other sensors attached to the robot. Furthermore, an adaptive process could be implemented in the case which the GPS error is really big. The gains could be readjusted to take into consideration mainly the IMU and odometer data.

Finally, sensor fusion using complementary filters for improving absolute position estimates using GPS, IMU and odometry is not sufficient to provide a robust and accurate system for automotive applications. In cases where the accuracy is of the essence, other methods should be used in conjunction with this one, such as lane tracking, and traffic sign localization together with map matching. There is a lot of research on these methods and many references can also be found in the introduction of this thesis, or in the bibliography section.

6 APPENDIX

6.1 FUNCTION PROTOTYPING

In this section the functions used in the file mobile_state will be explained.

In the main code the appropriate initialization of the variables is defined and the node needed for ROS is created. ROS is subscribing to the sensors' topics and the StateEstimation message is printed.

There are different functions for the different sensors, where the appropriate calculations are taking place. Those functions are:

- *callback_odom*
- *callback_gps*
- *callback_imu*

Every callback handles the measurements by transferring them to the appropriate frame, or expressing them in a different way.

In the *odom callback* the angular and linear velocity is being calculated from the encoders and the appropriate calculations for estimating the position using the odometry is coded, while not used for the sensor fusion.

In the *GPS callback*, the latitude, longitude and altitude obtained from the sensor are translated in terms of x, y, z in the body frame. (The first positioned is supposed to be the (0,0,0).

In the *IMU callback* the quaternions are being expressed in Euler angles, the acceleration is transferred to the body frame and lastly the first angles are saved as a reference.

In the *llatoecef* function, the transformation from latitude, longitude and altitude to the Earth Center Frame is taking place and feeded into the *lla2xyz* functions which transfers this to the robot's xyz frame.

In the *R_bf2ned* function, the transfer from the North-East-Down frame to the body frame is calculated.

Lastly, the *StateEstimation* function is performing the estimation using the calculated and transferred values from the previous functions and the complementary filters to publish the state of the robot.

6.2 USE OF THE ROBOT

Before one is able to use the sensors attached to the robot and the state estimation resulting from the sensor fusion, the appropriate ROS packages must be installed (assuming that ROS is already installed on the PC).

Firstly, the installation of the rosserial package is necessary typing the following commands:

```
$ sudo apt-get update
$ sudo apt -get install rosserial
$ sudo apt-get install rosserial-arduino
```

arduino_mr
In order for ROS to be able to send messages to the robot, the `arduino_mr` package, which includes the messages, must be installed. All is needed is for the folder to be copied in the folder ~`catkin_ws/src` and the run the following command:

```
$ catkin_make
```

While the system is at the ~/catkin_ws/ directory.

Odometry
In the same way the odometry folder is also copied in the directory ~catkin_ws/src and the following command should be run:

```
$ catkin_make
```

Joy_to_arduino
For the joystick to work, the installation of the following packages is needed:

```
$ rosdep install joy
$ rosmake joy
```

For information on how to configure the joystick (if further configuration is needed) please refer to the [36] reference.

The appropriate privileges must be assigned to the joystick in order for it to work:

```
$ sudo chmod a+rw /dev/input/jsX
```

The joystick is ready to be used. Last step is to copy the folder joy_to_arduino in the directory ~`/catkin_ws/src` and run the command:

```
$ catkin_make
```

If a different method of sending messages to the robot is needed, instead of the joystick, please refer to reference [36].

Delete ROS libraries

For the arduino_mr package to be used the following steps should be followed:

Deleting, if existing, the folder:

```
~sketchbook/libraries/ros_lib
```

And running the command:

```
$ rm -rf ~/sketchbook/libraries/ros_lib

$ rosrun rosserial_arduino make_libraries.py ~/sketchbook/libraries/
```

The last command should be run in order for the appropriate ROS libraries to be created.

Arduino Libraries

Two more libraries are necessary, the `PWM` and the `PID_motor`. Copy the two respective folders in the directory `~/sketchbook/libraries/`.

Using the Robot

In order to be able to use the robot and control it using the joystick, the following procedure should be followed:

1. Connect to the robot:
   a. Boot the robot.
   b. Connect to its Wi-Fi, which is listed under the name *pioneer* using the code: *pioneer123*.
   c. Through the terminal access control of the robot using ssh command *ssh -X linaro@10.10.0.1* inserting the code *linaro* when prompted.
   d. Run roscore.
   e. Launch the file *atx2_architecture.launch* located in the following package *arduino_mr*, which initiates the required files to control the robot.
   f. Launch the file all_to_joy.launch from the package state_estimeation_u_comp_filters, which allows all processes to run and the user to control the robot through the joystick.
2. The state estimation of the robot is being published to the message *mobile_state*.
   a. The state estimation can be run as follows:
      ```
      $ rosrun state_estimeation_u_comp_filters
      state_estimation_2at.py
      ```

Observing the messages being published
For accessing any topic, run:

```
$ rostopic echo <topic_name>
```

The list of open topics is viewed by:

```
$ rostopic list
```

The topic's messages can be viewed by:

```
$ rostopic info <topic>
```

For info on each topic's message:

```
$ rosmsg show <topic name>
```

6.3 TROUBLESHOOTING

- **ARM Computer:** In case the Arduino isn't communicating with the PC equipped with an ARM processor (e.g. Odroid, Raspberry Pi,...) the rosserial-arduino package may not work properly. In that case install the following package:

  ```
  $ cd ~/catkin_ws/src/
  $ git clone https://github.com/chuck-h/rosserial.git
  $ cd ..
  $ catkin_make
  $ catkin_make install
  $ source catkin_ws/install/setup.bash
  ```

- In order for the command *catkin_make* to run properly one should always run it from the `~/catkin_ws` directory.
- `roscore` must always be running before any action is taken considering the ROS environment.
- It may happen that the *raw GPS data won't be published*. In that case the user should reconfigure the xsens to print the raw data, because it might have lost its configurations.
- If any *communication issues* occur between the robot and the PC, run the following command on the robot's terminal:

  ```
  $ export LC_ALL=C
  ```

BIBLIOGRAPHY

[1] Richard M. Murray, Zexiang Li, S. Shankar Sastry, *"A Mathematical Introduction to Robotic Manipulation",* 1994 (Richard M. Murray, 1994)

[2] https://en.wikipedia.org/wiki/Mobile_robot

[3] Andon Venelinov Topalov, "*Recent Advances in Mobile Robotics*", 2011

[4] Jose-Marcio Luna, Rafael Fierro, "On the configuration of Sensors and Actuators on a Pioneer 3AT robot", October 2010

[5] "*Pioneer 2 / PeopleBot, Operations Manual*"

[6] Koji Ohmori, Kunio Sakamoto, "*Automatic Mobile Robot Control and Indication Method Using Augmented Reality Technology*"

[7] Dan Simon, "*Optimal State Estimation, Kalman, Hinfinity and Nonlinear Approaches*", 2006

[8] Stergios I, Roumeliotis, Gaurav S. Sukhatme, George A. Bekey, "*Circumventing Dynamic Modelling: Evaluation of the Error-State Kalman Filter applied to Mobile Robot Localization*", 1999

[9] Peter S. Maybeck, "Stachastic models, estimation, and control", 1979

[10] Walter T. Higgins, JR, *"A Comparison of Complementary and Kalman Filtering"*, 1975

[11] R. G. Brown, *"Integrated Navigation Systems and Kalman Filtering: A Perspective",* 1972

[12] Sławomir Romaniuk, Zdzisław Gosiewski, *"Kalman Filter Realization for Orientation and Position Estimation on Dedicated Processor",* 2014

[13] Niklas Magnusson, Tobias Odenman, *"Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion"*, 2012

[14] Vishisht Gupta, *"Vehicle Localisation Using Low-Aaccuracy GPS, IMU and Map-Aided Vision"*, Phd dissertation, The Pennsylvania State University, 2009

[15] John W. Allen, "Use of vision sensors and lane maps to aid gps-ins navigation" Master's thesis, Auburn University, 2011

[16] Dragan Obradovic, Henning Lenz, and Markus Schupfner, *"Fusion of sensor data in siemens car navigation system"*, 2007

[17] C. Chen, *"Low-cost loosely-coupled GPS/odometer fusion: a pattern recognition aided approach"*, 2008

[18] National Technical Meeting of The Institute of Navigation, Anaheim, CA, *"Performance Analysis and Architectures for INS-Aided GPS Tracking Loops"*, January 2003

[19] David McNeil Mayhew, *"Multi-rate sensor fusion for gps navigation using kalman filtering"*, Master's thesis, Virginia Polytechnic Institute and State University, 1999

[20] John W Allen, *"Use of vision sensors and lane maps to aid gps-ins navigation"*, Master's thesis, Auburn University, 2011.

[21] ZuWhan Kim, *"Robust lane detection and tracking in challenging scenarios"*, 2008

[22] S. Sivaraman and M.M. Trivedi, *"Improved vision-based lane tracker performance using vehicle localization"*, 2010

[23] Dragan Obradovic, Henning Lenz, and Markus Schupfner, *"Fusion of sensor data in siemens car navigation system"*, 2007

[24] M. Jabbour, P. Bonnifait, and V. Cherfaoui, *"Road tracking for multi-hypothesis localization on navigable maps"*, 2008

[25] J. Schmackers and A. Glasmachers, *"Landmark based fast positioning for sensor data fusion; receiver design and measurement results"*, 2011

[26] Yanlei Gu, T. Yendo, M.P. Tehrani, T. Fujii, and M. Tanimoto, *"Traffic sign detection in dual-focal active camera system"*, 2011

[27] Douglas Guimarães Macharet, Armando Alves Neto, Víctor Costa da Silva Campos, Mario Fernando Montenegro Campos, "*Mobile Robot Localization in Outdoor Environments using Complementary Filtering"*, 2010

[28] Tim Bailey, *"Mobile Robot Localisation and Mapping in Extensive Outdoor Environments"*, Phd dissertation, University of Sydney, 2002

[29] P. Newman, D. Cole and K. Ho, *"Outdoor SLAM using Visual Appearance and Laser Ranging"*, 2006

[30] Armando Alves Neto*, Douglas Guimarães Macharet, Víctor Costa da Silva Campos, Mario Fernando Montenegro Campos, *"Adaptive complementary filtering algorithm for mobile robot localization"*, 2009

[31] http://www.ros.org/about-ros/

[32] MTi-G User Manual and Technical Documentation, Revision H

[33] Krzysztof Kozłowski, *"Modelling and Control of a 4-wheel skid-steering mobile robot"*, 2004

[34] Panos Marantos, Yannis Koveos, and Kostas J. Kyriakopoulos, *"UAV State Estimation using Adaptive Complementary Filters"*, accepted for publication

[35] Jose-Luis Blanco, *“A tutorial on SE(3) transformation parameterizations and on-manifold optimization”*, 2013

[36] Πούλιας Κωνσταντίνος, *“Ανάπτυξη Συστήματος Ελέγχου Κίνησης Τροχοφόρων Ρομπότ”*, Διπλωματική Εργασία, 2015

[37] http://www.robotnik.eu/mobile-robots/summit-xl-hl/

[38] http://www.dronyx.com/xbot-tracked-mobile-robot/

[39] https://en.wikipedia.org/wiki/Geographic_coordinate_system

[40] Guowei Cai, Ben M. Chen, Tong Heng Lee, *“Unmanned Rotorcraft Systems”*, 2011

[41] http://www.basicairdata.eu/knowledge-center/background-topics/coordinate-system/