Localisation and Mapping for an Autonomous Lawn Mower

Implementation and analysis of multiple configurations to provide localisation and mapping features for an autonomous lawn mower using measurements from heterogenous sensors

MARCO Boffo

Master's Programme, ICT Innovation, 120 credits
Date: June 22, 2021

Supervisors: Håkan Carlsson, Thiemo Voigt, Joakim Eriksson
Examiner: Haibo Li
School of Electrical Engineering and Computer Science
Host company: RISE Research Institutes of Sweden AB
Swedish title: Lokalisering och kartläggning för en autonom gräsklippare
Swedish subtitle: Implementering och analys av flera konfigurationer för att ge lokalisering och kartläggning för en autonom gräsklippare genom att smälta mätningar från heterogena sensorer

Abstract

DRAFT
Autonomous lawn mowers have been available to consumers for more than 20 years now. During this lapse of time, however, their configuration and performance have seen little improvements.

With the advancement in embedded devices and sensors, such a configuration is now obsolete. The features provided now are limited and still prone to errors. Moreover, the installed external infrastructure on which they rely, i.e. the boundary wire, requires maintenance.

This thesis analyses the implementation of a localisation and mapping module which allows for the implementation of more features. Multiple sensors to improve the localisation of the autonomous lawn mower are analysed, and the best configuration is presented. Additionally, mapping features are implemented over the positioning system to update the understanding of the robot's environment.

This solution could be used to improve the coverage feature of the autonomous lawn mower. Moreover, the availability of a more precise localisation could be the starting point for more advanced features, e.g. interaction with the lawn.

Keywords

Autonomous Mobile Robot, Sensor Fusion, Localisation, Mapping

Sammanfattning

DRAFT
Autonoma gräsklippare har varit tillgängliga för konsumenter i mer än 20 år nu. Under denna tid har dock deras konfiguration och prestanda sett små förbättringar.

Med framstegen inom inbäddade enheter och sensorer är en sådan konfiguration nu föråldrad. Funktionerna som tillhandahålls nu är begränsade och fortfarande utsatta för fel. Dessutom kräver den installerade externa infrastrukturen som de förlitar sig på, dvs. begränsningskabeln, underhåll.

Denna avhandling analyserar implementeringen av en lokaliserings- och kartläggningsmodul som möjliggör implementering av fler funktioner. Flera sensorer för att förbättra lokaliseringen av den autonoma gräsklipparen analyseras och den bästa konfigurationen presenteras. Dessutom implementeras kartläggningsfunktioner över positioneringssystemet för att uppdatera förståelsen för robotens miljö.

Denna lösning kan användas för att förbättra täckningsfunktionen för den autonoma gräsklipparen. Dessutom kan tillgängligheten av en mer exakt lokalisering vara utgångspunkten för mer avancerade funktioner, t.ex. interaktion med gräsmattan.

Nyckelord

Autonom mobil robot, sensorfusion, lokalisering, kartläggning

Sommario

DRAFT
I tosaerba autonomi sono disponibili per i consumatori da più di 20 anni. Durante questo lasso di tempo, tuttavia, la loro configurazione e le prestazioni hanno visto piccoli miglioramenti.

Con il progresso dei dispositivi e dei sensori integrati, una tale configurazione è ora obsoleta. Le funzionalità fornite ora sono limitate e ancora soggette a errori. Inoltre, l'infrastruttura esterna installata su cui si basano, ovvero il cavo perimetrale, richiede manutenzione.

Questa tesi analizza l'implementazione di un modulo di localizzazione e mappatura che consente l'implementazione di più funzionalità. Vengono analizzati più sensori per migliorare la localizzazione del rasaerba autonomo e viene presentata la configurazione migliore. Inoltre, le funzionalità di mappatura vengono implementate sul sistema di posizionamento per aggiornare la comprensione dell'ambiente del robot.

Questa soluzione potrebbe essere utilizzata per migliorare la funzionalità di copertura del rasaerba autonomo. Inoltre, la disponibilità di una localizzazione più precisa potrebbe essere il punto di partenza per funzionalità più avanzate, ad es. interazione con il prato.

Parole chiave

Robot mobili autonomi, fusione di sensori, localizzazione, mappatura

Acknowledgments

DRAFT
I would like to thank Thiemo Voigt and Joakim Eriksson for the opportunity to work at this challenging topic.

I would like to thank Håkan Carlsson for the supervision of the project. His insights and suggestions have been vital for the development of this thesis.

I would like to thank Haibo Li for the examination of the thesis.

I would like to thank Mikael Alexiusson for the Husqvarna collaboration.

I want to acknowledge my family and friends that supported me through the whole process.

Stockholm, June 2021
Marco Boffo

Contents

List of acronyms and abbreviations

SLAM Simultaneous Location and Mapping

ROS Robotic Operating System

WO Wheel Odometry

VO Visual Odometry

DR Dead Reckoning

MEMS MicroElectroMechanical System

RMSE Root Mean Square Error

UWB UltraWideBand

DOP Dilution Of Precision

HDOP Horizontal DOP

F2F Frame to Frame

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

ORB Oriented FAST and Rotated BRIEF

ICC Instantaneous Center of Curvature

NRAO National Radio Astronomy Observatory

NMEA National Marine Electronics Association

API Application Programming Interface

2D 2 Dimensional

3D 3 Dimensional

IMU Inertial Measurement Unit

GPS Global Positioning System

GPS-RTK GPS Real Time Kinematic

KF Kalman Filter

PF Particle Filter

EKF Extented Kalman Filter

IEKF Iterated Extented Kalman Filter

AEKF Adaptive Extented Kalman Filter

UKF Unscented Kalman Filter

HRP Husqvarna Research Platform

GNSS Global Navigation Satellite System

RPi Raspberry Pi

RGB Red Green Blue

RGB-D RGB-Depth

RTAB-Map Real-Time Appearance-Based Mapping

LMSE Least Mean Squared Error

ALM Autonomous Lawn Mower

Chapter 1

Introduction

Autonomous systems development has improved consistently in the later period, providing improvements to every day life. These systems are able to autonomously execute their tasks without human interaction and to adapt to dynamic settings. On the field of robotics, their support is relevant when tasks are repetitive and the human presence is not necessary. In the specific case of mobile robotics, autonomous navigation is required and those systems can achieve it using knowledge of their pose and surroundings [1].

The pose of a mobile robot can be estimated using sensors which provide measurements of their environment [2]. Multiple types of sensors are available, which differ in frequency, phenomenon captured, and reliability. Heterogeneous sensors provide measurements which need to be properly fused together to improve the individual knowledge that they provide.

The awareness and the precision related to the position in the working environment changes with respect to each application [2]. The first autonomous mobile robots available to the consumers were the indoor cleaning robots, where simple collision detection is enough to wander in the indoor environment. A more recent consumer application of such robots is the Autonomous Lawn Mower (ALM), which works in the more complicated outdoor setting.

ALMs mow the lawn using a random walk coverage approach [3], i.e. they move in a random direction until they detect a specific underground wire or until they collide with an object. They will then rotate towards another direction and keep on going with this behavior. To do so, they rely in a minimal set of sensors to achieve their goals in dynamic outdoor environments.

Boundary wires are laid underground to provide the outer limits of the lawn working area, and some guide wires are used to aid the mobile robot

with more complex tasks that the random behaviour will not be able to address, such as the autonomous return to the charging base and the navigation through narrow passages. These wires transmit a proprietary electrical current signal passing through and the mobile robot is able to detect it using three different magnetometer sensors, one positioned in the center front and the other two placed on each side. Through the analysis of the magnetic field direction and pulses, the lawn mower is able to understand if the wire defines a boundary or if it provides a guide to go towards another area of the lawn or back to the charging base. However, these external wires require regular maintenance which could be avoided with a more sophisticated model.

Additionally, the ALMs implement collision sensors, located near the rear wheels. When the mobile robot bumps into any firm object from the front, it will trigger push sensors situated on the chassis. It has been used to react in case of collision with unexpected objects, but eventually it could be used to map the environment with those objects and avoid them systematically.

In more sophisticated and recent models, additional frontal ultrasonic range sensors are installed to slow down the mobile robot before potentially colliding with objects in its trajectory.

The most advanced models also include Global Navigation Satellite Systems (GNSSs) receivers to keep track of the global robot position, for theft protection and lawn monitoring purposes. However, it is not precise enough on its own to provide real time localisation information for it to be used for navigation purposed by the ALM, but it can be used to detect that a particular side of the lawn has not been mowed in a while.

This configuration does not require them to understand their position on the world, as they just use a reactive architecture defined by a random behaviour motion pattern without any understanding of their surroundings [4]. These robots are situated, i.e. they are not taking into account events of the past and they cannot foresee their future interactions with the environment [5]. As such, they are not able to plan ahead their path and they just react to perceptions of their surroundings. They do not have a model of their world, and they do not need to update it in case of unexpected changes of their environment [4]. They are flexible and adaptive as they rely on the infrastructure manually installed for them [6], which however requires additional installation and maintenance.

1.1 Problem

The current method of autonomously mowing the lawn is not effective, since it is based on an external infrastructure to perform a reactive behaviour based

on a random walk coverage algorithm [3]. The problems that arise from this setting are multiple, as the external components require maintenance and the current algorithm cannot guarantee a complete deterministic coverage of the area. This configuration for the ALMs was given by constraints related to the low computational power available in the past. Moreover, the recent progresses in the performance of embedded devices and sensors have made the navigation model of ALMs obsolete, and as those components have also lowered their cost, their implementation is now feasible even for this consumer products. The combined availability of more advanced devices and more computational power enables for the implementation of real time applications to improve the performance of these mobile robots.

The performance of an ALM can be improved with a more advanced perception and action system used to implement deterministic techniques of coverage planning [2]. The problem that needs to be solved is the development of a more precise localisation and mapping module. The ALM, to perceive its surroundings and act accordingly, needs additional computational power and sensors than the ones available in the current configurations. With a more advanced set of components, it would be possible to reach a more accurate understanding of the mobile robot's pose inside a map of the lawn itself. An analysis of the best configuration of sensors, along with their measurements fusion, is needed to understand how to provide to costumers a more reliable product which needs no maintenance of the external infrastructure. The issue is related to finding the most appropriate technique to fuse all these sensors' measures in a setting where every downside is compensated and every useful aspect of those sensors is exploited and highlighted,

An aspect that is worth investigating is related to the removal of the need of external infrastructure used for localisation purposes. Instead of relying to a vulnerable boundary wire and with a set of sensors built in the ALM directly, without installing additional systems on the lawn, the implementation of such a system will allow for the offering of a more complete set of features to the users. Additionally, the removal of such a boundary wire or external infrastructure enable the ALM to cover larger areas of custom configuration.

Finally, the fused combination of given control commands and the whole set of sensors at disposal for this project has yet to be investigated in research, as usually fewer sensors are available.

Autonomously mowing the lawn will help saving time and avoid human intervention as much as possible. The steps needed to improve current systems are related to a dynamic management of the boundary of the lawn, eliminating the need for a boundary wire with its related installation and maintenance.

The knowledge about position and orientation of the mobile robot, with respect to a map, enables for the application of more deliberative architectures [7]. With this architecture, the pose with respect to a map will allow the ALM to make independent decisions and to plan its path to cover the lawn in a shorter amount of time, leaving a better pattern, avoiding unexpected objects, and saving energy and resources.

1.2 Research Goals

This master thesis investigates how to improve the performance of an ALM providing localisation and mapping features using a set of heterogeneous sensor directly mounted on it. It will focus on the implementation and analysis of a module based on the fusion of the measurements provided by different configurations of those sensors to remove the need of relying on external infrastructure. Finally, their configurations are investigated through multiple experiments, and drawbacks and improvements of each sensor are analysed.

The goal of this thesis is to provide an overview on how to develop a precise localisation module for an ALM. An analysis about different configurations of sensors and about how to fuse their measures using sensor fusion filters is presented. As end result, the ALM is able to operate in a specified environment without the need to intervene with the installation of additional infrastructure, but just with the addition of heterogeneous sensors installed directly on the mobile robot.

The focus of this degree project is the precise localisation inside a predefined given map to ensure that the ALM stays inside the virtual boundaries defined. Moreover, a map of the lawn is developed to allow for updates using collision events against unexpected object to provide a more dynamic view of the lawn.

The research goals to be reached by this project are the following:

- Providing an initial map and an initial location of the ALM in it, will the ALM be able to stay within the given boundaries with a specified margin for error?
- Will the ALM be able to correctly update the map by adding or removing eventual obstacles found by navigating in the map through collision detection?

In order to evaluate if the above goals have been reached, for an ALM made to cover an area of $5000\,\mathrm{m}^2$, the following research questions will be

evaluated:

- Identify if the mobile robot is able to stay within the boundaries defined, ensuring that the error in boundary violation will be inside the dimension of the ALM itself: around $50\,\mathrm{cm}$. These measures are defined to show that the lawn mower will not exceed its given boundaries, risking to damage itself, the lawn, or others.
- Identify if the ALM is able to return to the parking dock to recharge itself with an offset of a maximum of $1\,\mathrm{m}$ after a run of at least $500\,\mathrm{m}$. In this way, it will be relevant to show how the mobile robot is not drifting. Moreover, with this results it will be possible to correct eventual offsets once at the starting position.
- Identify if the ALM is able to localise a collision event within an error of at most 1% of the total distance already travelled, e.g. identify a tree with a maximum offset of $20\,\mathrm{cm}$ error after a $20\,\mathrm{m}$ run. This will be addressed to ensure that the mower is able to update precisely the map with the presence of unexpected objects.

1.3 Research Methodology

The research methodology [8] will follow a quantitative approach: acquiring measurements and using them to validate or not the formulated research questions through quantitative analysis. The assumptions will follow an objective and realistic paradigm where the final results will be evinced quantifying measures of the observations and gaining a better knowledge of the environment. The research method adopted will be of experimental nature to understand the cause and effect of the obtained measurements, improving where possible, and of descriptive nature to highlight the characteristics of the obtained system. A deductive approach will be used to test the theories and draw conclusions about the hypotheses described in the research questions. The research strategy adopted will be based on data collection through multiple case studies of experimental nature and the collected data will be analysed with computational mathematics. Statistical analysis will be used to test the quality of the obtained results. In particular, in this degree project, I will apply and discuss the validity, reliability, replicability, and ethics of those results.

The following tasks are required to achieve the above mentioned objectives. Literature study is the first step. The next step is related to

the improvement of the available localisation module performance, starting from the refinement of the sensor's drivers. The most relevant sensor fusion technique identified is implemented to merge heterogeneous sensors' measurements.Afterwards, different configurations of sensors are tested and a phase of tuning their attributes to reach more reliable and valid results is done. The theoretical validation of the sensor fusion system is performed with the usage of simulations of the sensor's measurements according to the required assumptions. Some outdoor experiments are performed and checked against the ground truth with some metrics to evaluate the improvements on the localisation performance. Finally, relying on the localisation results, the definition of initial virtual boundary is going to be provided with a first run of the ALM. The knowledge of the elements inside the obtained map is then improved using collisions events, always relying on the localisation improvements.

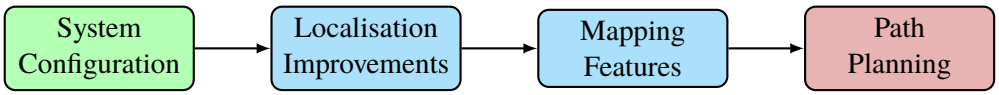


Figure 1.1: Main tasks of the project:
Improvement[green], new development[blue], and future development[red].

Since such ALMs are consumer products, some constraints will be taken into account, such as the configuration complexity and the sensors' cost analysis. Some aspects regarding this degree project are described here:

- The operation area is projected into a 2 Dimensional (2D) environment.
- The usage of an embedded device, such as a Raspberry Pi (RPi), limits the computational power of the module. Thus, the performance might not be as good as if a more powerful device would be used.
- A complete Simultaneous Location and Mapping (SLAM) algorithm is too computationally expensive to perform well in real-time with considering a dynamic outdoor environment. Instead, an easier approach is to split the two different phases. The localisation aspect will be achieved with a sensor fusion approach and the mapping happens after localisation, using collision events to update the knowledge of the predefined map.
- Relying on a camera in outdoor settings means that the weather conditions and time of mowing can affect the results. As such, the testing

will be performed in similar configurations, and, if possible, with an attitude towards limiting such a characteristic.

- A Range Finder Sensor was not considered as most of the outdoor environment will be sparse or empty. Thus, the sensor would not be able to provide valuable improvement to justify the expensive choice. Moreover, such a sensor requires a relevant amount of computational power to be run.

They provide the rationale for the project to investigate some aspects more than others. Some of them are directly limiting and they are not be investigated for such a project. Some others instead are part of the future works, discussed at the end of this thesis.

1.4 Structure of the thesis

In the following chapters the topics of this thesis are discussed as follows.

Chapter 2, Background, presents relevant information about mobile robots, localisation, and mapping. Some theoretical related works are presented to describe the state-of-the-art. It provides a summary about the lesson learned during the literature review.

Chapter 3, Methods, defines the localisation and mapping approaches chosen from the literature study. Moreover, the methodologies and experiments configurations to achieve the research goals are explained.

Chapter 4, Experiments and Results, defines the experiments developed to address the research questions. Moreover, the quantitative results obtained from them are shown to highlight the desired aspects.

Chapter 5, Discussions, explains and interprets the final results. The findings and their limitations are elaborated on.

Chapter 6, Conclusions, wraps up the thesis and how to improve further are provided. Finally, some reflections related to this thesis are presented.

Chapter 2

Background

This chapter presents relevant background research that is needed to understand the current state of the art about mobile robotics, the localisation field, and mapping aspects.

The specific coordinate frame of the wheeled mobile robot adopted for this project, with its kinematics and constraints, are described to present how the robotic lawn mower will be localised, controlled, and constrained.

Afterwards, the focus will be on the localisation topic, starting from an analysis of sensors both already available on the ALM and also those useful for positioning. The focus is about sensors which require no additional infrastructure installation. Additionally, techniques used to fuse their measurements will be described, shown, and analysed to highlight which technique is the most relevant for this project.

Furthermore, a brief analysis of mapping aspects is presented. Different mapping algorithms are available and only the most relevant approach for this thesis is discussed.

Finally, theoretical related works are briefly described to highlight their contributions, and a summary of the lessons learned from the literature study is presented.

2.1 Mobile Robots

Mobile robots are robotics systems which are able to move themselves on the environment they are in. The mobile robot used in this thesis equips two standard back wheels with differential drive, and two castor wheels on the front which allow free rotation around the wheel axle for multiple direction steering. An analysis of the coordinate and robot frame is presented, followed

by the definition of its kinematic model and by an overview of the constraints involved.

2.1.1 Coordinate Frame

The pose of the robot P_r on the global frame frame G is defined by the following Degree of Freedom (DOF) vector used to localise it in the 2D environment:

$$P_r = \begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T \tag{2.1}$$

where x_r, y_r, and θ_r define respectively the translation and rotation of the origin of the robot frame O_R with respect to the origin of the global frame O_G, as shown in figure 2.1.

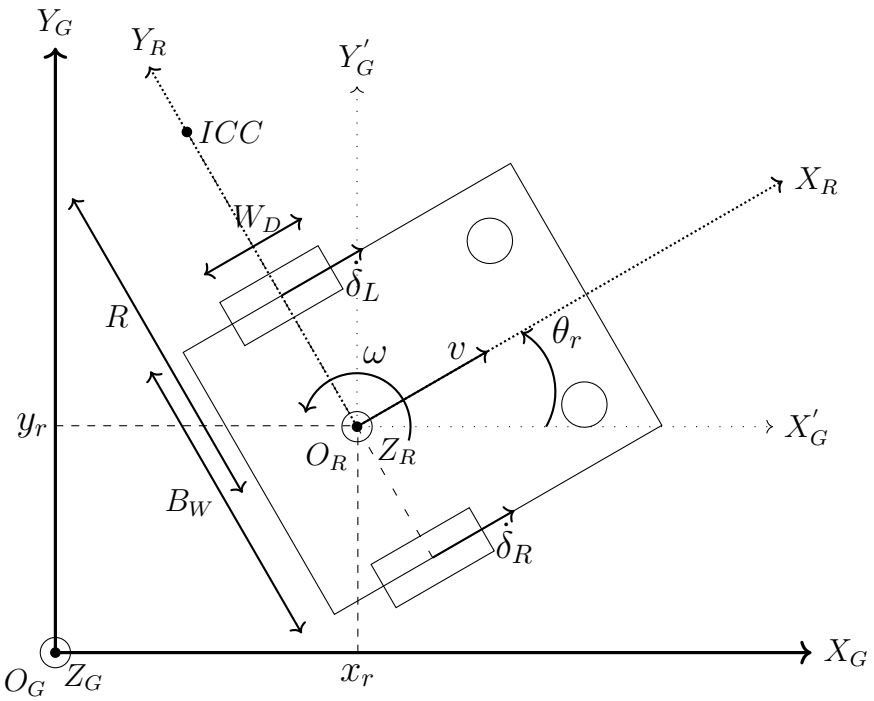


Figure 2.1: Kinematic model of a differential drive mobile robot.

The rotational and translational relations between frames is provided by the transformation matrix T_R^G, used to transform a point of the robot frame P_R to a point of the global frame P_G, as defined in the following equation:

$$P_G = T_R^G \cdot O_R \qquad \text{with} \qquad T_R^G = \begin{bmatrix} cos(\theta_r) & -sin(\theta_r) & x_r \\ sin(\theta_r) & cos(\theta_r) & y_r \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

For the scope of this thesis, just the kinematics have been considered as other aspects, such as torque, friction, physical implications, and in general the dynamics of the systems, are not relevant to be analysed for the scope of this thesis.

2.1.2 Kinematic Analysis

The ALM consists of two drive back wheels mounted along the same axis. Each wheel is actuated independently either in forward or backward rotation. By varying the velocity of each wheel, the robot rotates around a Instantaneous Center of Curvature (ICC) [9], ICC in figure 2.1, defined directly by its distance from the robot origin, R, varying the wheel velocities, $\dot{\delta}_L$ and $\dot{\delta}_R$, as in equation (2.3):

$$R = \frac{\dot{\delta}_R + \dot{\delta}_L}{\dot{\delta}_R - \dot{\delta}_L} \cdot \frac{B_W}{2} \tag{2.3}$$

where B_W defines the base width of the ALM as distance between wheels. Using this approach it is possible to vary the trajectory that the robot follows varying the wheels' velocities, i.e. in case the wheels' velocities are equal, R will result to be infinite and the robot will not rotate, following a straight trajectory. Some constraints about the possible trajectories are explained in section 2.1.3.

The rate of rotation about ICC is the same as the velocity of rotation around the robot frame origin O_R, defined by the time derivation of its orientation θ_r. This angular velocity, ω, is used to control the trajectory of the robot and it is defined as in equation (2.4).

$$\omega = \frac{\partial \theta_r}{\partial t} = \frac{\dot{\delta}_R - \dot{\delta}_L}{B_W} \tag{2.4}$$

The differential drive mobile robot is controlled directly through the desired linear velocity v along the X_R axis of the robot, and the desired angular velocity ω around the Z_R axis. The angular velocity is defined above in equation (2.4) and the linear velocity is defined as a transformation of the

angular velocity, or of the wheel velocities, as in equation (2.5).

$$v = \omega \cdot R = \frac{\dot{\delta}_R + \dot{\delta}_L}{2} \tag{2.5}$$

As the differential mobile robot is propelled by two separate motors mounted on the back wheels, its movements are defined by the wheel actuators and thus the desired control velocities need to be transformed into the proper wheels' velocities using the following equations (2.6).

$$\dot{\delta}_L = \omega \cdot (R - \frac{B_W}{2}) \qquad \dot{\delta}_R = \omega \cdot (R + \frac{B_W}{2}) \tag{2.6}$$

2.1.3 Constraints

This differential drive system is not able to freely move along the 2D environment, as the degrees of freedom, defined in (2.1), are less than degrees of mobility, given by the control velocities, v and ω. The system is thus called non-holonomic and a non-holonomic constraint has to be defined. The standard wheels used by the differential mobile robot are subject to such non-holonomic rolling constraint, which limits the sideways sliding by imposing that the wheel spins purely along the direction of the wheel. For example, to perform the rolling motion along the Y_R axis, the robot will have to vary the velocity of each wheel to rotate around its origin to align itself along that initial direction before moving forward. Non-holonomic constraints do not allow for integration of the differential equations to retrieve the final pose, since the displacements of each wheel are not sufficient to determine it. This forces the position and orientation to be estimated using integration of wheels velocities rather than their position displacements. However, with a sufficient sampling rate, an integrable approach can be used to estimate its pose, but the usage of incremental displacements provided by integration of wheels' velocity measurements will inevitably accumulate errors over time.

2.2 Sensors

For a robot to localise itself, it needs to perceive its surroundings and to do so it relies on its sensors' measurements. These sensors perceive and interpret different recorded phenomena based on their characteristics, and they can be divided in three main different types. *Interoceptive sensors* focus on internal data measured directly by the mobile robot. They output engineering

quantities which are used to estimate how the system is behaving internally, e.g. wheel encoders. *Proprioceptive sensors* measure robotic's interaction with the environment. They provide a value of how the movements of the mobile robot are perceived with respect to the outside, e.g. accelerometers, or gyroscopes. *Exteroceptive sensors* provide measures related directly to their perception of the external environment. The information they measure are related to phenomena happening at the environments around the mobile robot, e.g. magnetometers, GNSSs, or cameras.

As they measure different aspects, they perform differently. They provide advantages and disadvantages which can be combined in an intelligent manner to exploit their capabilities and limit their inaccuracies.

The available sensors that could be installed directly on the mobile robot, that do not require additional installation of infrastructures, and that can be used to improve localisation are described below.

2.2.1 Wheel Encoder

This interoceptive sensor detects the number of revolutions of the wheels, and they can be used to estimate their velocity [10]. An example of the optical wheel encoder is given in figure 2.2.

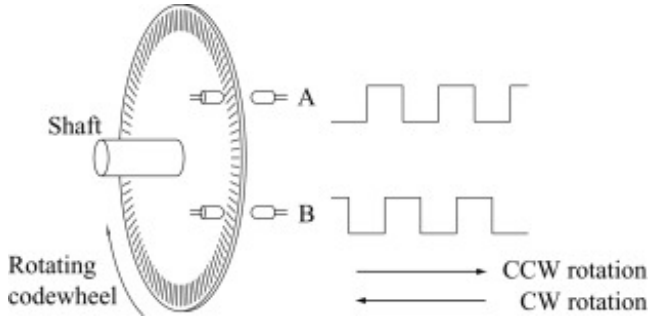


Figure 2.2: Optical wheel encoder and resulting encoded signal [11].

This sensor relies on a rotary encoding disk attached to the wheel axis on the end of the motor that estimates the relative angle change of the wheel. Rotary encoders can be of different kinds based on which phenomenon they observe: mechanical, optical, or magnetic. Incremental encoders estimate the relative rotation of the wheels by detecting the number of pulses, defined by step signals A and B in Figure 2.2, within certain time period using the internal clock of the embedded computer.

Wheel encoders are often chosen for Dead Reckoning (DR), i.e. the estimation of the pose using previous estimate and velocity, because of their fast measurement rate and their inexpensiveness. Related to the field of sensor fusion, these are key aspects to consider, as their employment is effective if combined with other sensors. They will be able to provide fast accurate velocity measures which could be used to compute a short-term accurate DR.

The estimate derived from the wheel encoders is called Wheel Odometry (WO) and it is subjects to errors and their accumulation. Some non-systematic errors related to these sensors are the miscalculations caused by slippage, uneven terrain, and other environmental issues which include wheels temperature and pressure of their tires. In order to improve its DR estimates, other sensors' measurements can be combined with its velocity estimates to get rid of the integration error accumulation on its position calculations.

2.2.2 Inertial Measurement Unit

An Inertial Measurement Unit (IMU) is an inertial sensor which is used to detect multiple phenomena and which measurements are relative to the inertial frame of platform they are attached to. They have both proprioceptive and exteroceptive characteristics as they contain both accelerometers, gyroscopes, and magnetometers. These sensors are placed in each orthogonal axis to provide measurements for each degree of freedom in free space, as shown in figure 2.3. Recent IMUs are developed on MicroElectroMechanical Systems (MEMSs) which are inexpensive and provide measurements with a high sampling rate.

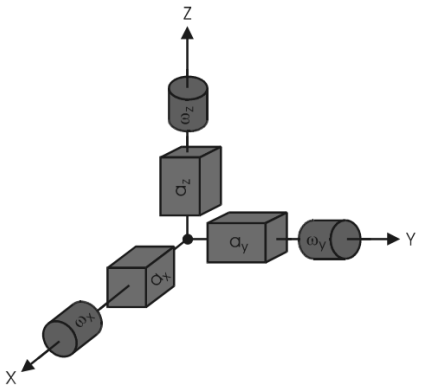


Figure 2.3: Structure of accelerometers and gyroscopes of an IMU [12].

Accelerometers measure the linear accelerations, gyroscopes output the rotational velocities, and magnetometers detect the earth magnetic field. Knowing these measures, the position and orientation can be estimated by integrating those signal over time with a DR approach. Their measurement are prone to deterministic errors, such as repeatable terms and temperature induced variations, and to stochastic errors as switch-on to switch-on variations, and in-run variations [13]. Deterministic errors can be compensated for through prediction and calibration, while stochastic ones are more difficult to model because of their noise and randomness. Accelerometers and Gyroscopes are sensors that need no external support, and they are not conditioned by external conditions or disturbances as they measure proprioceptive characteristics. Magnetometers instead might be conditioned by external disturbances as the magnetic field can be disturbed by multiple external factors, as any electronic device has a magnetic influence.

They provide accurate results over a short time span, but the integration of noisy results implies an accumulation of these errors over time. This phenomenon then leads to a diverging drift effect, if not corrected by other sensor measurements.

2.2.3 Global Navigation Satellite System

GNSSs are exteroceptive systems based on signals received from satellites orbiting earth. A GNSS receiver uses trilateration to estimate its position on the globe using time of flight of the signals sent by some satellites. The signals sent by satellites are composed by their trajectory, called ephemeris, along with their internal reference timestamp, and they are used to estimate their range as shown in figure 2.4.

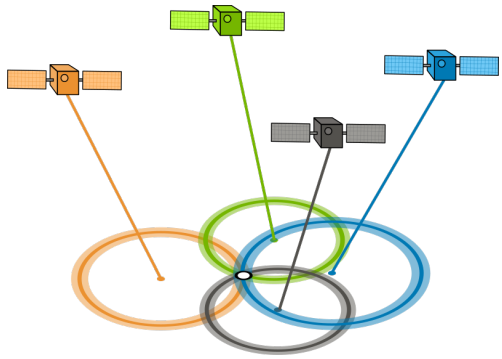


Figure 2.4: Trilateration for GNSS, including range error bars.

The configuration of perceived satellites contributes to their accuracy, as close position of several satellites implies that their range will be close, and their signals will not be effective. A more spread configuration of satellites' range reduces the Dilution Of Precision (DOP), which compares the position error to the satellite range error. Other source of errors are related to conditions which directly affect the time of flight of the signal, such as the presence of ionospheric and tropospheric delays, and the positioning in dense and noisy environments. The performance of these sensors is thus highly dependent on their environment.

For the GNSS receiver to estimate its position, it need to acquire measurements from at least four satellites. Three satellites signals are used to triangulate the 3 Dimensional (3D) position and the fourth is used for the synchronization of the clocks, essential to compensate eventual time differences between the receiver and the satellites. In case the perceived satellites are less then required, the receiver will have a system outage, not being able to estimate its position and velocity. As there exist multiple GNSS systems developed by distinct entities with different original purposes and number of satellites, some GNSS sensors are able to receive signals from different systems to improve their satellite coverage and their precision. With a higher number of perceived satellite signals, the receiver is able to use redundant information to increase its robustness to external disturbances and improve its position measurement accuracy.

The sampling rate of GNSS receivers is usually around one estimate per second, thus it is generally too slow to estimate the position for vehicles.

2.2.4 Camera

A camera sensor is an exteroceptive sensors, usually composed of lenses and light sensors, which gathers the Red Green Blue (RGB) information from the environment. Recent camera models are able to estimate stereopsis data, i.e. depth measurements can be estimated. A typology of such a sensor is the structured-light camera. It projects an infrared mesh pattern in the environment and then uses stereo cameras to perceive this pattern and it triangulates the depth based on the density of the projected points. Multiple factors can lower this sensor performances, such as excessive lightning conditions and reflecting materials which modify the infrared pattern.

These sensors can be used with a process called Visual Odometry (VO) to estimate the incremental movement of the camera, and the robot at which they are attached, by the changes detected in adjacent images with a Frame to

Frame (F2F) method. The VO algorithm detects the position of some specific features and then matches those landmarks in two consecutive images. By knowing their relative pose changes and the configuration of the camera, it is possible to estimate the corresponding camera movement The performances of this technique relies on the identification of landmarks, the more visible and variant landmarks are detectable, the more the pose estimation will be accurate. Those features are computed using different descriptors, e.g. Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB). The matching phase is guided by an initial estimate of the camera movement which projects previously found features to search just along that projection. The landmarks are matched after passing a consistency test and those displacements are triangulated to estimate the camera pose through a bundle adjustment step [14], as shown in figure 2.5.

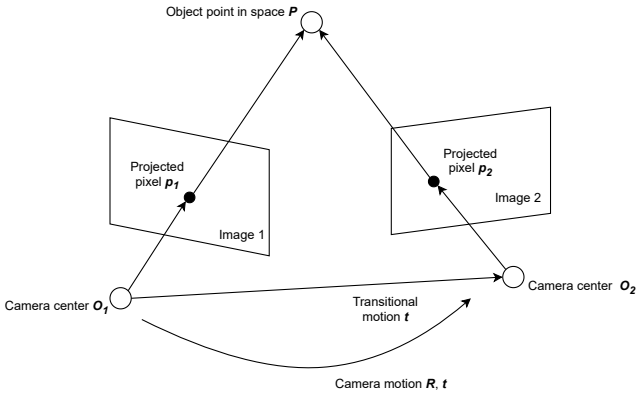


Figure 2.5: Motion estimation via F2F methodology [15] [16].

Instead of using a more complex SLAM algorithm where all the position of the landmarks are also used to estimate a map of the environment, in the VO case the landmarks are used just for the displacement analysis, not storing the landmark positions over time. Using this approach, just the previous state will be saved to estimate the offset, this results in faster computations, and thus an higher sampling frame rate. This is a key aspect, since we aim to use this approach in an embedded system for a real-time application. However, the estimates in velocity between frames derived from the VO is used to compute a DR localisation and it is subjects to drift.

Given the still relatively low frequency of the VO, high linear and angular velocities will render the VO estimation inaccurate, as it will be more difficult for the system to match identified landmarks when their displacement is bigger.

2.3 Sensor Fusion

In the field of localisation for mobile robots, sensor fusion is employed to combine redundant and complementary measurements from heterogeneous sensors to gain a more precise understanding of the pose in the environment. An issued related to the sensor fusion problem is the determination of the best configuration to combine the measurements of multiple different sensors. The objective of sensor fusion is to merge the knowledge acquired by these sensors to improve the position estimate with a more accurate solution with respect to any of the individual data sources trying to mitigate their drawbacks and to maintain their best provided features [17].

Sensor fusion could be characterised based on their competitive, complementary, and cooperative approaches [18]. Competitive fusion happens when the same attribute is estimated by different sensors to increase the system's robustness. Complementary fusion is related to the merging the observations of different phenomena to describe the complete system's behaviour. Cooperative fusion combines measures from different sources to provide information that will not be obtained by a single sensor.

Sensor fusion is related to independent measures and it can be differentiate into multiple categories: sensors, attributes, domains, and time [19]. The fusion across different sensors involves merging the same measured phenomenon from different devices to increase the robustness of the measured data, in a competitive approach. Fusion among different attributes is focused on sensors which estimates different but connected measures which might be related by kinematics or dynamics aspects, improving their precision in a complementary way. Different domains are fused when sensors measure the same attribute with a different range or frame, improving their performances when one of these sensors reaches its limits, through a cooperative procedure. Sensor fusion across time merges measure arrived at different times with increasing reliability to increase the evolution of the estimates.

The most common approach to sensor fusion is through the usage of statistical methods, where the uncertainties of the sensors are described by probabilistic models [20]. This section will describe the basics of some statistical filters used for sensor fusion purposes. The Kalman Filter (KF) will be detailed as it is the most basic among them, followed by an analysis of its non-linear correspondent, the Extented Kalman Filter (EKF). These different techniques available to fuse their heterogeneous measurements are described in their performances.

2.3.1 Kalman Filter

The KF was first developed by R.E. Kalman and published in [21] as a Bayesian estimator used in linear Gaussian systems. The usage of KF is ideal for real time, embedded, and sensor fusion systems as it can estimate dynamic systems disturbed by sensor noise with high performances. It does not require to store in memory the history of all the previous states and they are computationally efficient.

It uses a recursive probabilistic approach to efficiently filter discrete estimates of the states of a process and its estimated uncertainty, and then through new measurements it corrects its state and decreases its uncertainty with the goal of minimising the Least Mean Squared Error (LMSE). To reach an optimal estimation, at each iteration it performs a weighted sum of its current estimate using the prior estimate and new measurements, and weighting them based on their relative uncertainty. The KF holds information about its estimate and uncertainty through, respectively, the state vector $\mathbf{X}_t$, which contains the states and descriptions of the system, and the state covariance matrix P_t, which holds information about the variance between those states as statistical error indicator. It is structured as continuous cycle of its iterative process, composed of a prediction and correction phase.

The prediction phase estimates the next state based on the previous one, eventual control inputs, and adding Gaussian white noise to account for uncertainties in the prediction estimate. First of all, the KF computes the prediction of the current estimate $\hat{\mathbf{X}}_{t+1}$ using its previous estimate $\mathbf{X}_t$, the transition matrix A, and eventual inputs $B \cdot \mathbf{u}_t$, as in equation (2.7).

$$\hat{\mathbf{X}}_{t+1} = A \cdot \mathbf{X}_t + B \cdot \mathbf{u}_t \tag{2.7}$$

where the state, being a Gaussian random variable, can be described with its mean and covariance as $\mathbf{X}_t \sim \mathcal{N}(\mathbf{X}_t, P_t)$.Then, the prediction of the current state covariance $\hat{P}_{t+1}$ is computed using the previous state covariance P_t, the transition matrix A, and the state noise covariance Q, as in equation (2.8).

$$\hat{P}_{t+1} = A \cdot P_t \cdot A^T + Q \tag{2.8}$$

After a noisy measurement $\mathbf{Z}_{t+1}$ has been obtained, the correction phase corrects the prediction based on the uncertainty of both the prediction and the measurement to improve the estimation of the state. As first step in the correction, the prediction of the estimated measurements, $\hat{\mathbf{z}}_{t+1}$, is obtained

using the measurements matrix H, as in equation (2.9).

$$\hat{\mathbf{Z}}_{t+1} = H \cdot \hat{\mathbf{X}}_{t+1} \tag{2.9}$$

Then, the innovation vector, $\mathbf{Y}_{t+1}$, is computed as the difference between the estimated measurements and the actual measurements $\mathbf{Z}_{t+1}$:

$$\mathbf{Y}_{t+1} = \mathbf{Z}_{t+1} - \hat{\mathbf{Z}}_{t+1} \tag{2.10}$$

Its related innovation covariance, S, is given by the following formula, where R is the measurements process noise:

$$S = H \cdot \hat{P}_{t+1} \cdot H^T + R \tag{2.11}$$

The Kalman gain, K, is then computed to improve the estimate as a relation between state covariance and innovation covariance, related to the measures obtained as follows:

$$K = \hat{P}_{t+1} \cdot H^T \cdot S^{-1} \tag{2.12}$$

The corrected state, $\mathbf{X}_{t+1}$, is then obtained using the predicted state and the innovation vector weighted by the innovation covariance:

$$\mathbf{X}_{t+1} = \hat{\mathbf{X}}_{t+1} + K \cdot \mathbf{Y}_{t+1} \tag{2.13}$$

Finally, the corrected covariance matrix of the state, P_{t+1}, is obtained by equation (2.14).

$$P_{t+1} = (I - K \cdot H) \cdot \hat{P}_{t+1} \tag{2.14}$$

After this, the loop of equations restarts from the prediction phase by updating the time pedix from $t + 1$ to t, following the structure defined in figure 2.6.

Its many components can are structured in multiple ways. The initial state $\mathbf{x}_0$ can be defined either by zeros or starting estimates or according to the first measurements. The starting uncertainty of the filter P_0 can be initialized with the identity matrix or directly with the state noise covariance. In both cases, the filter will need some steps before converging to more stable estimates of state and covariance. The state noise covariance Q is harder to determine, as the transition equations would need to specify maximum bounds to define its components. The measurement noise covariance R is composed by the

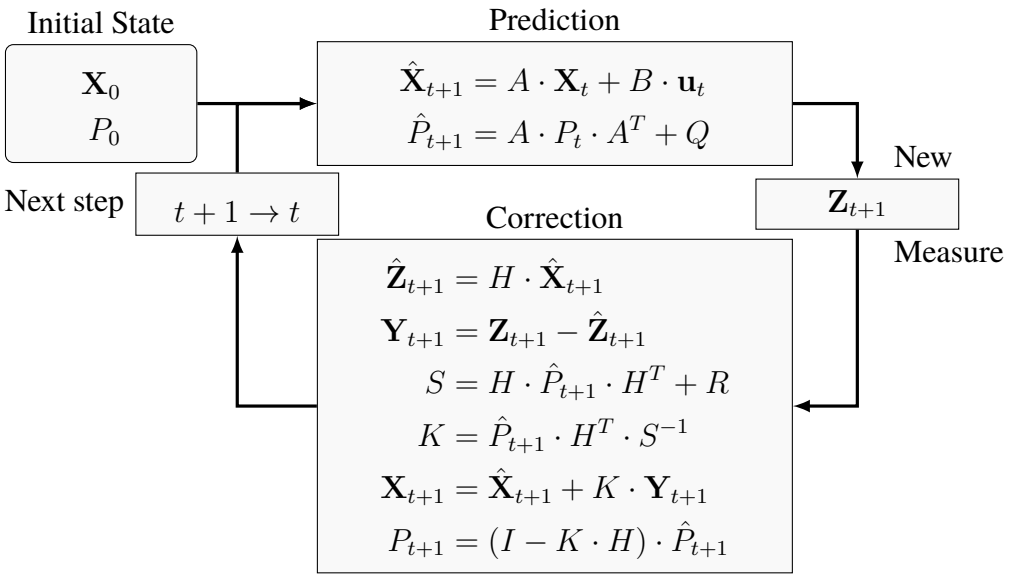


Figure 2.6: Structure of the Kalman Filter

variances of its components and they are usually estimated through calibration. If the state and measurement noises change over time, their matrices Q and R needs to be updated as the filter runs, as it is their relative ratio which determines how the Kalman gain K is defined to improve the estimates. In this case, the KF is defined as Adaptive [22] [23].

An numerical evaluation improvement over the standard KF is directly given by changing the state covariance correction step using the Joseph's form [24], defined by:

$$P_{t+1} = (I - K \cdot H) \cdot \hat{P}_{t+1} \cdot (I - K \cdot H)^T + K \cdot R \cdot K^T \tag{2.15}$$

As the standard state covariance correction in Equation (2.14) is composed by subtraction, which can result in a loss in symmetry and positive-definiteness due to rounding errors in a finite word length computer, the Joseph's form state covariance correction maintains positive and non-zero eigenvalues in P_t through a more computationally demanding equation [25].

For the filter to be considered an optimal estimator, it needs to exploit the properties of the Gaussian uncertainties, meaning that the disturbances in both the process and measurements need to be composed by Gaussian white noise. Moreover, the prediction and correction phases are formulated with the assumptions that the state $\mathbf{X}_t$ is markovian, i.e. that is evolution depends solely on its previous value, and that the measurement $\mathbf{Z}_{t+1}$ only depends on the

state $\mathbf{X}_{t+1}$, and not on previous measurements. When these assumptions are validated, the KF is proven to be the best linear estimator possible. However, it can be used with reasonable results even if the assumptions are not true, i.e. when the noise is not Gaussian and the system is non linear, but in those cases other filters will perform better and must be preferred, such as the ones described below. In any case, as the KF needs multiple iterations to stabilise, if its state is observable, the performance of the algorithm should not account for the first steps.

2.3.2 Extended Kalman Filter

The EKF is an extension of the KF used to estimate non linear systems, where equations (2.7) and (2.9) are defined respectively as follows:

$$\hat{\mathbf{X}}_{t+1} = f(\mathbf{X}_t) + \varepsilon \qquad \text{where } \varepsilon \sim \mathcal{N}(0, Q) \tag{2.16}$$

$$\hat{\mathbf{Z}}_{t+1} = h(\hat{\mathbf{X}}_{t+1}) + \delta \qquad \text{where } \delta \sim \mathcal{N}(0, R) \tag{2.17}$$

The problem related to non linear system is given by the fact that even though its inputs are Gaussian random variables, its outputs are not and therefore they could not be used for the propagation of the estimates and covariances with the KF equations.

The EKF deals with the non linearity of the system by propagating the estimates through a linearisation of those functions around the previous estimates of the system using the Taylor's expansion [26]. The obtained first order derivatives, defined as Jacobian matrices, are used as approximation for the linear transformation of the system [27]. The corresponding matrices will be substituted in the previously defined equations for the KF, as:

$$F = \frac{\partial f}{\partial \mathbf{X}_t} \qquad\qquad H = \frac{\partial h}{\partial \hat{\mathbf{X}}_{t+1}} \tag{2.18}$$

The linear approximation of the EKF is based on the first order derivatives, providing only first order accuracy. In case of highly non linear systems, this procedure can inject relevant errors in the system and cause divergence of the estimate.

An improvement on the performance of the EKF, regarding how to handle highly non-linear systems, has been defined in the Iterated Extented Kalman Filter (IEKF). As described in [28], the correction step is iterated along new estimates after every measure to achieve a more accurate estimation and a lower variance before a new prediction step.

2.4 Mapping

For a mobile robot to navigate in an area, it is essential to understand its position in a map that represents its environment. There exists two main typologies of maps: topological maps which describes the connectivity of significant places in the environment as nodes in a graph connected by paths as edges, and metric maps which employs a coordinate system to describe the proprieties of the environment. Moreover, maps can be defined based on their perspective: world-centric maps where a global coordinate space is used and every object has a global coordinate, and robot-centric maps where each entity is defined with respect to the robot's pose and measurements.

The description of a common used map which project a 3D environment into a 2D abstraction and which aids the localisation aspects is provided below.

2.4.1 Occupancy Grid

Occupancy grids are world-centric metric maps which are composed of occupancy probabilities cells. Their representation of the environment is defined by evenly spaced cells containing a random variable which describes the uncertainty of the presence of an obstacle within its coordinates [26]. While the mobile robot moves in that map, the measurements obtained can be used to update the random variables of nearby cells using approximates posterior estimates.

Using this methodology to define the internal map of the mobile robot could be used to virtually limit its environment by defining its borders with negative values. By initialising the map with a random uncertainty, the mobile robot will be able to move freely inside its boundaries and as soon as it approximates to an negative cell it will change its direction to avoid getting out of its limits. At the same time, it will be able to use its sensors to perceive its surroundings and update the occupancy cells based on the identifies obstacles within it. Then, once a better certainty about its knowledge of the map in reached, the mobile robot could employ path planning algorithms to improve its coverage performance.

2.5 Related Works

The improvement of a robotic lawn mower is a recent research topic. It has emerged thanks to a set of research outcomes which have provided tools to increase the possible features that a ALM can achieve. Previous development

of such mobile robots from scratch can be found in multiple reports. They provide a complete analysis of all the components and a comprehensive view about the features to be provided on an ALM. The complete development of a fully featured mobile robot for golf course mowing, as part of a yearly robotics course, is detailed in [29]. A recent master thesis about the development robotic lawn mower with Visual SLAM and terrain classifier is described in [30]. A bachelor thesis regarding the prototyping of a mower using a camera and GPS in available in [31]. Both of these works did not manage to deliver on the desired outcomes. This is given mostly by the lack of time considering they had to also build the infrastructure of the mower.

Different approaches for localisation on a Husqvarna Research Platform (HRP) have been developed: in [32], using GPS Real Time Kinematic (GPS-RTK) external infrastructure, and in [33], using UltraWideBand (UWB) sensors on the mower. In these cases however, there is the requirement for external devices to localise the ALM and this aspect is not part of the scope of this thesis.

A review about techniques for localisation and mapping for autonomous systems is provided in [34], where the focus is about the definition of sensors' performance and the description of different map features.

Extensive analyses of the techniques used by autonomous robots that have to deal with real-world sensor data and mapping using their measurements are provided in [26], [20] and [35], which are well-cited references for estimation tasks.

2.5.1 Localisation

Specifically for this project, given the configuration of the sensors, the algorithms and results described in [36] will provide a solid base for the localisation module. Moreover, a discussion about the improvements and limitations offered by each sensor is included, along with a comprehensive description for the implementation of the EKF in the Robotic Operating System (ROS) framework. Their `robot_localization`* ROS package implements multiple nodes to provide an easy way to fuse data from multiple and heterogeneous sensors. However, its performance and static configuration do not allow for the implementation of a real time system which needs updates with an high frequency.

Other approaches to sensor fusion using the KF to merge the measures of the available sensors are described in the literature. In [37], an analysis about

* http://wiki.ros.org/robot_localization

the difference of performance between KFs driven by the kinematic model or by the WO measures is described. As a conclusion, while their performance are similar, the kinematic driven prediction phase KF was preferred.

Some references which use an EKF to merge measures from a similar set of sensors available with a different mobile robot, than the one discussed in this thesis, are described below. In [38] an EKF is used to merge the WO and VO with GNSS and IMU measures. Multiple EKFs are employed in series in [39] to merge the WO and VO with GNSS and IMU measures. An adaptive approach using the `robot_localization` package to merge WO, GNSS measures, IMU measures, and VO is defined in [40]. In [13], an EKF is adopted to improve the localisation of a vehicle using WO, one GNSS receiver, and one IMU. In [41], an approach to deal with delayed measurements in the EKF is presented to fuse WO, one GNSS receiver, and one IMU. In [42] and [43], an EKF is adopted to provide localisation for a mobile robot using inexpensive sensors, such as GNSS receiver and IMU.

The adaptive KF has been in use for a while and some references are described below. An adaptive approach for the generation of the noise covariances in the KF to account for dynamic noise in sensor fusion to improve Global Positioning System (GPS) estimation with inertial measures from a IMU is defined in [44]. In [45], the adaptive approach is applied to an EKF. In [46], [47], and [48], adaptive approaches using EKF are adopted to merge measurements obtained respectively by one GNSS receiver, by one GNSS receiver and one IMU, and by two GNSS receivers and one IMU. In [49], an adaptive approach to sensor fusion using EKF is used to merge measurements from WO and VO.

Comparisons about different configurations of sensor fusion approaches are defined in the following references. In [50], a comparison about the fusion of WO, GNSS, and IMU using EKF and Unscented Kalman Filter (UKF), described in B.1, is presented. It highlights how the EKF provides more accurate and faster results than the UKF. In [51], an EKF is adopted to improve the localisation of a mobile robot using different configurations with a wheel encoder, a magnetic compass, one GNSS receiver, and one IMU. The most accurate results have been obtained by the approach fusing all their measures. In [52], different configurations of sensors including WO, GNSS receiver and IMU are fused with an EKF and analysed. The most accurate results have been obtained fusing all the sensors' measurements. Multiple KFs are used in [53] to merge VO and IMU, and to discuss the related improvements provided by different configurations. To improve the non-linearity performances, an implementation of the IEKF with a IMU and VO system is described in [54].

2.5.2 Mapping

The mapping feature has been discussed in the literature using multiple techniques. As the focus of this thesis is related to the usage of occupancy grids, their presence in the state-of-the-art implementation is analysed.

In [55], an algorithm to consider measurements from a range sensor to update the cells of an occupancy grid using Bayes Theorem is presented, specifying that it has been derived and improved from [56] and[26]. A similar approach has been presented in [57], where the sensor of choice to detect the environment is the sonar sensor. In these two approaches, two separate occupancy grids are used to describe the emptyness and occupancy aspects of the environment. A final process is then performed to merge their knowledge by using the most probable cells to provide the map.

2.6 Summary

The key lessons learned during this background analysis and description can be summarised by the following points:

- a localisation module could be built in a modular way starting from a general model and its kinematic analysis.
- the sensors available, which do not require an external infrastructure installations, are enough to measure the necessary phenomena to cover the states needed to improve the localisation aspects of a mobile robot;
- statistical sensor fusion techniques enables for the merging of measure from different and heterogeneous sensors with performances which are robust and fast enough to be implemented in an embedded device. The Adaptive Extented Kalman Filter (AEKF) provides good estimates and it can be tuned updating the noises which characterise the system according to sensor's measures, as the localisation scope requires;
- occupancy grids provide a 2D abstractions of the environment which can account both the definitions of its boundaries and the presence of obstacles within it;
- the usage of collision events to update the knowledge of the cells in the occupancy grid is possible using the Bayes Theorem.
- numerous related studies and implementations have been developed in similar topics to improve localisation and mapping features.

Chapter 3

Methods

In this chapter, the techniques adopted are explained from a theoretical perspective. The chosen methods and related models are discussed to highlight the rationale behind their use to reach and evaluate the research goals defined in Chapter 1.

The localisation system is defined by a sensor fusion approach to exploit the measurements available in a way that limits the different drawbacks of the sensors available. The sensor fusion filter that will be used is a AEKF, as it is a robust and adaptable. It enables for the fast and reliable fusion of heterogeneous sensors in a way that will enable the ALM to have a more accurate estimate of its pose.

The mapping approach is defined starting from the results of the localisation and exploiting the collision sensor installed in the ALM. Through the usage of these elements, a virtual boundary will be set and the presence of objects within it will be provided by the events fired by such a sensor.

Finally, the methodology adopted to evaluate the performances of these modules is defined.

The structure to provide localisation and mapping features is shown in Figure 3.1.

RISE AB, in collaboration with Husqvarna AB, is interested in the research and development of a refined approach to improve the performance of the ALM, thus allowing for the development of more features. The system will be built upon the HRP [61], a ROS enabled Husqvarna Automower 450X with additional sensors assembled upon it, as shown in Figure 3.2. It has been improved by a former thesis student [62] and it is now equipped with additional two IMUs, two GNSS signal receivers, and a RGB-Depth (RGB-D) camera.

The platform adopted, HRP, has been improved starting from its

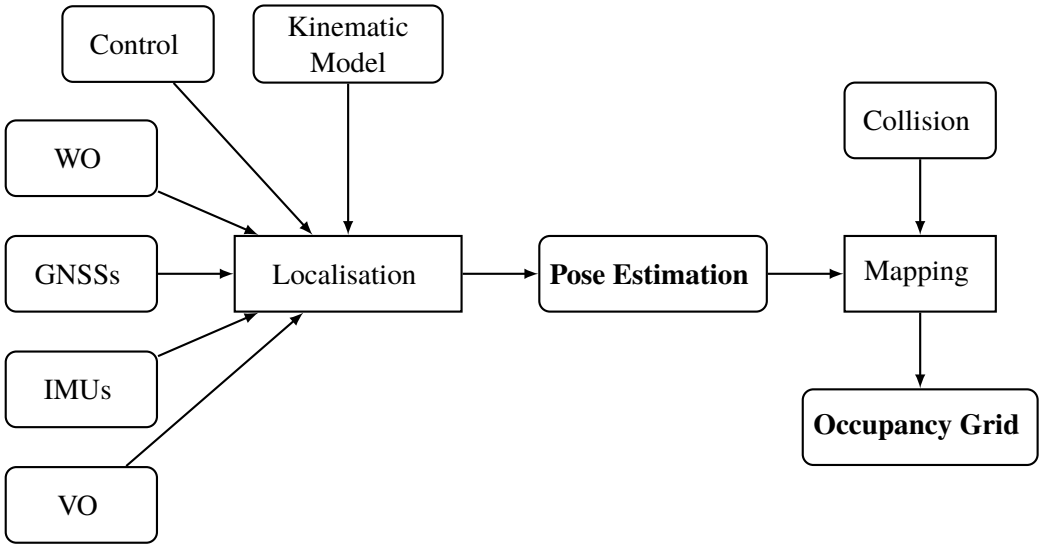


Figure 3.1: Structure of the Solution

configuration and sensors' drivers.

At first, an analysis of available platform is performed to gain a more comprehensive understanding of the current HRP implementation. The documentation of the HRP [61] and by the previous student [62] are going to be carefully comprehended before focusing on the study of the related state-of-the-art.

3.1 Hardware Configuration

HRP that holds the sensors and the raspberry pi, also proving the measurements of the wheel encoders and of the embedded GPS receiver. Computer to guide the ALM and make him move according to the desired path, also at the end the ALM was free to run its random path and use the algorithm implemented to stay inside the boundaries.

3.1.1 Raspberry Pi 4

On this device, most of the embedded devices will be attached. The HRP, the IMUs, and the camera.

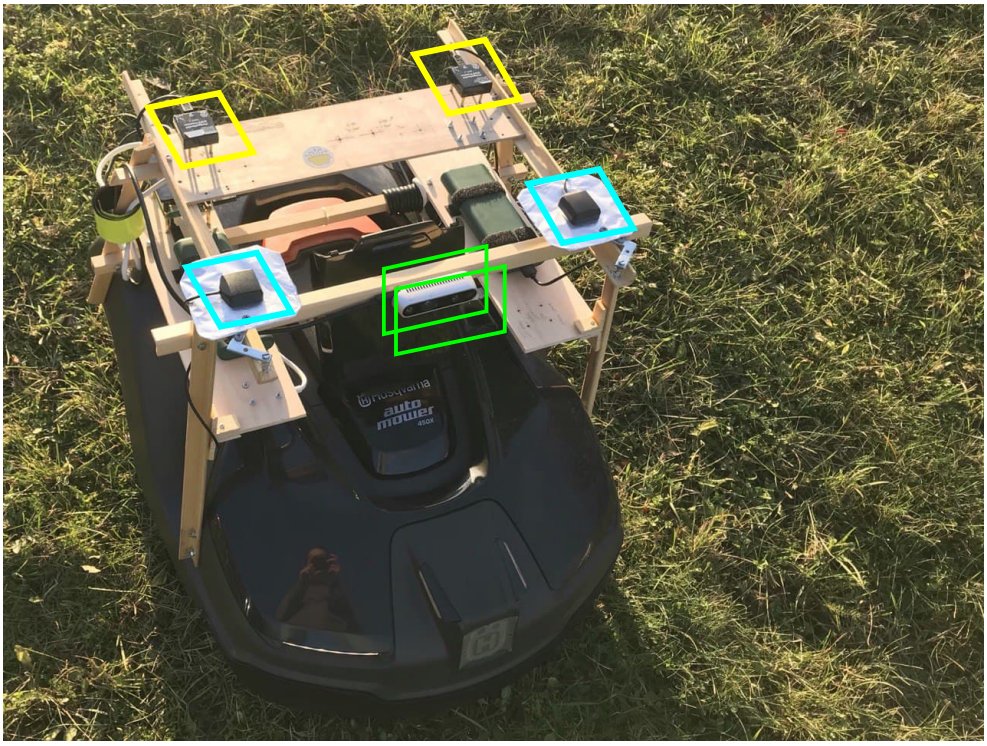
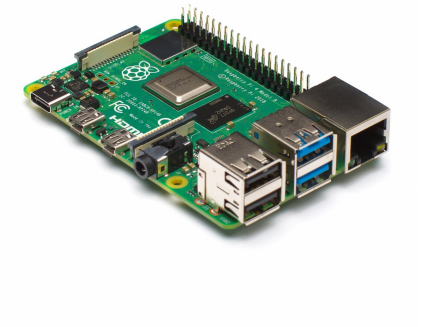
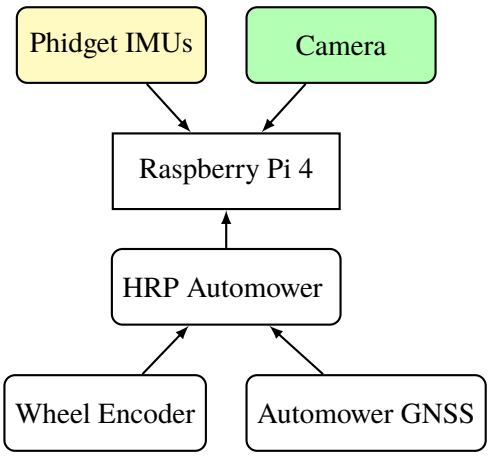


Figure 3.2: ALM Model with focus on additional sensors:
IMUs [yellow], GNSS receivers [cyan], and Camera [green]



(a) Raspberry Pi 4 Model B



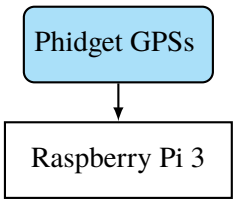
(b) Raspberry Pi 4 configuration

Figure 3.3: Raspberry Pi 4

3.1.2 Raspberry Pi 3

On this device, running Ubuntu 20, the Phidgets GPS sensors will be attached.

(a) Raspberry Pi 3 Model B



(b) Raspberry Pi 3 configuration

Figure 3.4: Raspberry Pi 3

3.2 Adaptive Sensor Models

Check for new measurements in background and dynamically create the components for the update step

3.2.1 Wheel Encoder Model

It is included in the motor of the wheel of the ALM, inside the Motor Kit shown in 3.5a. Measure the wheel displacement and provide WO, as explained in 3.

The performances have been examined and are available below.

(a) Motor Kit Automower 450X

Aspect	Value
Frequency	200 Hz
η_v	0.02 m/s
η_ω	0.02 rad/s

(b) Sensor's performance

Figure 3.5: Wheel Encoder

They are mounted in the wheel axis on both wheels.
Through the usage of wheel encoders attached to both these wheels, an estimated velocity for them is calculated and used to determine its pose. This open-loop technique to determine the position and orientation is called DR, as the new pose is estimated by using the last known pose and the new speed measurements.

The wheel encoders attached on each wheel provide an estimate of the number of pulses they detected. These encoder pulses need to be translated to an approximated wheel displacement through the transformation constant C_D provided by equation 3.1.

$$C_D = \frac{\pi \cdot W_D}{E_T} \tag{3.1}$$

where W_D is the estimated length of the wheel diameter and E_T is the encoder time resolution. This value is used to derive the velocity of each wheel separately, $\dot{\delta}_L$ and $\dot{\delta}_R$, by multiplying their encoder pulses, E_L and E_R respectively, with transformation constant and dividing it using the time delay Δ_t, as in equation 3.2.

$$\dot{\delta}_L = \frac{C_D \cdot E_L}{\Delta_t} \qquad \dot{\delta}_R = \frac{C_D \cdot E_R}{\Delta_t} \tag{3.2}$$

Using these single wheel displacements, it is possible to calculate the linear velocity along the x axis, v, and the angular velocity on the z axis, ω, with respect to the mobile robot frame using equations 3.3:

$$v = \frac{\dot{\delta}_R + \dot{\delta}_L}{2} \qquad \omega = \frac{\dot{\delta}_R - \dot{\delta}_L}{B_W} \tag{3.3}$$

where B_W is the estimated length of the base width of the back wheel axis.

These forward kinematics equations of this differential drive mobile robot are used to update the pose of the mobile robot, from time t after a time step Δ_t to time $t + 1$, in the global coordinate frame with the Euler first-order differential approximation [58], as shown in equation 3.4.

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \cos(\theta_t) & 0 \\ \sin(\theta_t) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \cdot \Delta_t \tag{3.4}$$

This method however is subject to multiple errors and inaccuracies, as the measurements and time delays will never be perfect. The systematic errors can be given by imperfectness of robot model as impreciseness in the wheel

diameters or wheel base measures, which are used to estimate the movements. However, the presence of non-systematic random errors as result of usage of imperfect measurements are more difficult to model, e.g. wheel slippage, uneven terrain, and external forces applied. The accumulative characteristic of these errors will break the stability of the system, making the estimate to drift after a period of time. These conditions render this DR model not adequate to determine the pose of the mobile robot, but its estimate could be used anyway to improve the positioning system.

$$\mathbf{v}_{Enc_t} = \mathbf{v} \qquad \boldsymbol{\omega}_{Enc_t} = \boldsymbol{\omega} \tag{3.5}$$

Measurements vector with uncertainty vector

$$\mathbf{Z} = \begin{bmatrix} \mathbf{v}_{Enc_t} \\ \boldsymbol{\omega}_{Enc_t} \end{bmatrix}^T \quad \mathbf{R} = \begin{bmatrix} \eta_{\mathbf{v}_{Enc}} \\ \eta_{\boldsymbol{\omega}} \end{bmatrix}^T \tag{3.6}$$

where $\omega_{\mathbf{x}_{Enc}} = \omega_{\mathbf{y}_{Enc}} = 1$ and $\omega_{\boldsymbol{\theta}_{Enc}} = \pi/180$, experimentally derived.

Measurement Matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

Measurement Jacobian Matrix

$$J_H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.8}$$

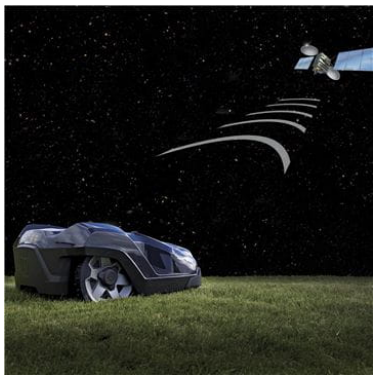
3.2.2 GNSS receiver Model

Measure the satellite positions to estimate its absolute position global values in the global coordinate frame.

Different GNSSs are available. A GPS receiver is embedded in the HRP, shown in 3.6a. Moreover, two additional GPS receivers has been added, shown in 3.7a.

The performances of PhidgetGPS ID: 1040_0B have been examined and are available below.

It is mounted in front, at coordinate x,y,z with orientation theta with respect to the robot coordinate frame.
The measures obtained by the GNSS receiver are transformed from WSG-84 degrees to local coordinates as specified below.



(a) Automower 450X GPS

Aspect	Value
Frequency	$1\,\mathrm{Hz}$
$\boldsymbol{\eta}_x$	$0.02\,\mathrm{m/s}$
$\boldsymbol{\eta}_y$	$0.02\,\mathrm{m/s}$
$\boldsymbol{\eta}_\theta$	$0.02\,\mathrm{rad/s}$

(b) Sensor's performance

Figure 3.6: GNSSs receivers adopted

(a) Performances

Aspect	Value
Frequency	$5\,\mathrm{Hz}$
$\boldsymbol{\eta}_x$	$0.02\,\mathrm{m/s}$
$\boldsymbol{\eta}_y$	$0.02\,\mathrm{m/s}$
$\boldsymbol{\eta}_\theta$	$0.02\,\mathrm{rad/s}$

(b) Sensor's performance

Figure 3.7: GNSSs receivers

Using specific math equations derived by [59], first a transformation from the geodetic coordinates in WSG-84 degrees estimated by the GPS receiver sensors to the ECEF coordinates is implemented. It is done so both for the initial coordinate and then used to check the difference with the ECEF translation from the geodetic coordinates of the successive reading of the GPS receiver.

From World Geodetic System: WGS 84 values

$$R_E = 6378137 \qquad \text{Equatorial radius (m)} \tag{3.9}$$

$$R_P = 6356752.3142 \qquad \text{Polar radius (m)} \tag{3.10}$$

From which the following can be derived the following values necessary

for the next algorithms:

$$F_R = \frac{R_E - R_P}{R_E} \qquad E_R = F_R \cdot (2 - F_R) \tag{3.11}$$

Using these values it is possible to transform from Geodetic WSG-84 Coordinates to ECEF coordinates using the Algorithm specified in 1.

Algorithm 1 Geodetic to ECEF Coordinates

Input: Latitude and Longitude (lat, lon) in WSG-84 degrees
Altitude (h) in meters
Output: ECEF Coordinates in meters (x, y, z)

1: $N = \dfrac{EqR}{\sqrt{1 - E_R \cdot \sin(lat)^2}}$
2: $x = (h + N) \cdot \sin(lat) \cdot \cos(lon)$
3: $y = (h + N) \cdot \cos(lat) \cdot \sin(lon)$
4: $z = (h + (1 - e\_sq) \cdot N) \cdot \sin(lat)$
5: return x, y, z

Finally, by exploiting the previous algorithm, it is possible to transform the Current Geodetic Coordinates into Local Coordinates by using some Reference Geodetic Coordinates and Algorithm 2.

In order to get the mean using the first measurements before the first command to move the automower, those values are stored in μ_{lat} and μ_{long}.

The formula used to obtain the distance between this mean value and the current value of the gps measurements is defined below.

$$\mathbf{x}_{GPS_t} = \text{geodetic2enu}(latC - latR) \tag{3.12}$$
$$\mathbf{y}_{GPS_t} = \text{geodetic2enu}(longC - lonR) \tag{3.13}$$

Then the orientation $z_{\boldsymbol{\theta}_{GPS_t}}$ is obtained by computing the direction between consecutive measurements.

$$z_{\boldsymbol{\theta}_{GPS_t}} = \arctan(\mathbf{y}_{GPS_{t-1}} - \mathbf{y}_{GPS_t}, \mathbf{x}_{GPS_{t-1}} - \mathbf{x}_{GPS_t}) \tag{3.14}$$

"""In this section we shall assume that wrapping operation amounts to enforcing the angular variable to be in the $[-\pi, \pi]$ interval, and we designate this operation as follows

$$w_\pi(x) = mod(x + \pi, 2\pi) - \pi \tag{3.15}$$

Algorithm 2 Geodetic to Local using Current and Reference Coordinates

Input: Current Latitude and Longitude ($latC$, $lonC$)
in WSG-84 degrees
Current Altitude (hC) in meters
Reference Latitude and Longitude ($latR$, $lonR$)
in WSG-84 degrees
Reference Altitude (hR) in meters
Output: Local Coordinates in meters (Δx, Δy, Δz)

1: xC, yC, zC = Geodetic to ECEF Coordinates ($latC$, $lonC$, hC)
2: xR, yR, zR = Geodetic to ECEF Coordinates ($latR$, $lonR$, hR)
3: $xD = xC - xR$
4: $yD = yC - yR$
5: $zD = zC - zR$
6: $\Delta x = -\sin(lonR) \cdot xD + \cos(latR) \cdot yD$
7: $\Delta y = -\cos(lonR) \cdot \sin(latR) \cdot xD - \sin(latR) \cdot \sin(lonR) \cdot yD + \cos(latR) \cdot zD$
8: $\Delta z = \cos(latR) \cdot \cos(lonR) \cdot xD + \cos(latR) \cdot \sin(lonR) \cdot yD + \sin(latR) \cdot zD$
9: return Δx, Δy, Δz

Note that when computing the difference between two angular variables, the wrapping effect of the circle should be taken into account, e.g., the difference between 178° and -178° should evaluate to 4°. This is achieved by the function $w_\pi(x)$ when the difference is given as the argument, i.e., difference between two angles x and y is computed as $w_\pi(x - y)$, as defined in [60]. """

$$\boldsymbol{\theta}_{GPS_t} = \boldsymbol{\theta}_t + w_\pi(z_{\boldsymbol{\theta}_{GPS_t}} - \boldsymbol{\theta}_t) \tag{3.16}$$

Measurements vector with uncertainty vector

$$\mathbf{Z} = \begin{bmatrix} \mathbf{x}_{GPS_t} \\ \mathbf{y}_{GPS_t} \\ \boldsymbol{\theta}_{GPS_t} \end{bmatrix}^T \quad \mathbf{R} = \begin{bmatrix} \omega_{\mathbf{x}_{GPS}} \\ \omega_{\mathbf{y}_{GPS}} \\ \omega_{\boldsymbol{\theta}_{GPS}} \end{bmatrix}^T \tag{3.17}$$

where $\omega_{\mathbf{x}_{GPS}} = \omega_{\mathbf{y}_{GPS}} = \text{HDOP} = \sqrt{\sigma_x^2 + \sigma_y^2}$ and $\omega_{\boldsymbol{\theta}_{GPS}} = \pi/8$, experimentally derived.

Measurement Matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.18}$$

Measurement Jacobian Matrix

$$J_H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

3.2.3 IMU Model

Measure the angular velocity, acceleration, and magnetic fields in the three orthogonal axis.

This PhidgetSpatial Precision 3/3/3 High Resolution - ID: 1044_1B, shown in 3.8a, has a 3-axis accelerometer, gyroscope and compass with high resolution readings at low magnitudes.

The performances have been examined and are available below.

(a) Performances

Aspect	Value
Frequency	$250\,\mathrm{Hz}$
$\boldsymbol{\eta}_\omega$	$0.02\,\mathrm{rad/s}$
$\boldsymbol{\eta}_a$	$0.02\,\mathrm{m/s^2}$

(b) Sensor's performance

Figure 3.8: IMUs

It is mounted in front, at coordinate x,y,z with orientation theta with respect to the robot coordinate frame.

"""As a general rule the closer is the IMU to the Centre of Gravity/Rotation the better; is to bear in mind that compensation for position offset reduces overall accuracy."""

The mobile robot is constrained to move and rotate around its center

of rotation, in a way that the displacement of the imu will not lower the performance, moreover, the angular velocity is the same for the whole robot, so the measured angular velocity can be fused directly.

Measurements vector, as obtained by IMU, with uncertainty vector.

The measurements obtained by the IMU are related to its own frame, so these measures need to be translated into the correct robot frame, by using the Rotation matrix.

$$\omega_r = R_i^r \cdot \omega_i \qquad \mathbf{a}_r = R_i^r \cdot \mathbf{a}_i \tag{3.20}$$

Using the newly obtained calibrated measurements, the measurement matrix and related noise matrix are defined as follows.

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{\omega}_{IMU_t} \\ \mathbf{a}_{IMU_t} \end{bmatrix}^T \qquad \mathbf{R} = \begin{bmatrix} \eta_{\boldsymbol{\omega}_{IMU}} \\ \eta_{\mathbf{a}_{IMU}} \end{bmatrix}^T \tag{3.21}$$

where $\omega_{\dot{\boldsymbol{\theta}}_{IMU}} = 1$ and $\omega_{\mathbf{a}_{IMU}} = 4$, experimentally derived.

Measurement Matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

Measurement Jacobian Matrix

$$J_H = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

3.2.4 Camera Model

Measure the displacement of the camera F2F and provide VO. The device that is used is the Intel® RealSense™ depth camera D435, shown in Figure 3.9a. It provides RGB-D images, i.e. both RGB and depth information.

"The Intel® RealSense™ depth camera D435 is a stereo solution, offering quality depth for a variety of applications. It's wide field of view is perfect for applications such as robotics or augmented and virtual reality, where seeing as much of the scene as possible is vitally important."

It is mounted in front, at coordinate x,y,z with orientation theta with respect to the robot coordinate frame.

Using the Real-Time Appearance-Based Mapping (RTAB-Map) package*[63] [64], more specifically the ROS version 0.20.9 for ROS Noetic.

* http://introlab.github.io/rtabmap/

(a) RGB-D Camera Performances

Aspect	Value
Frequency	$5\,\mathrm{Hz}$
$\boldsymbol{\eta}_v$	$0.02\,\mathrm{m/s}$
$\boldsymbol{\eta}_\omega$	$0.02\,\mathrm{rad/s}$

(b) Sensor's performance

Figure 3.9: Camera

Measurements vector, as obtained by the RTAB-Map package in ROS. Moreover, it also provides its own uncertainty vector error based on the uncertainty of the estimation over the images.

Odometry velocities merged, as the drift may still be an issue.

Measurements vector with uncertainty vector

$$\mathbf{Z} = \begin{bmatrix} \mathbf{v}_{Vis_t} \\ \boldsymbol{\omega}_{Vis_t} \end{bmatrix}^T \quad \mathbf{R} = \begin{bmatrix} \eta_{\mathbf{v}_{Vis}} \\ \eta_{\boldsymbol{\omega}_{Vis}} \end{bmatrix}^T \tag{3.24}$$

where $\omega_{\mathbf{x}_{Vis}} = \omega_{\mathbf{y}_{Vis}} = 1$ and $\omega_{\boldsymbol{\theta}_{Vis}} = \pi/180$, experimentally derived.

Measurement Matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.25}$$

Measurement Jacobian Matrix

$$J_H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.26}$$

3.3 Sensor Fusion Model

The localisation module is based on the merging of the information derived by the kinematic model, the control commands, and the sensors' measurements. The sensor fusion algorithm AEKF has been chosen for its time performances and reliability. It improves the KF allowing for the management of non-linear systems and the possibility to account for dynamic sensors' noise. It is defined as a Discrete time system, running with a frequency defined by the value Hz. Its implementation is hereby defined, starting from the Prediction phase, the different Measurements models of the heterogeneous sensors, and the Correction phase.

The system will start with all the state variables set to 0 and a know initial Covariance matrix P_0 set as an Identity I multiplied by 1^{-3}, i.e. diagonal matrix with those values.

3.3.1 Prediction Model

The prediction phase is driven by the kinematic model, as it is defined in Figure 2.1, following the conclusions of [37].

As assumption, the system will follow a Constant Velocity Model.

The state variables needed to provide the most accurate positioning in a 2D environment are defined in (3.27).

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}_t & \mathbf{y}_t & \boldsymbol{\theta}_t & \mathbf{v}_t & \boldsymbol{\omega}_t \end{bmatrix}^T \tag{3.27}$$

From the definition of the Kinematic Model in Figure 2.1, the first three positional elements, i.e. $\mathbf{x}_t$, $\mathbf{y}_t$, and $\boldsymbol{\theta}_t$, are related to the global frame. Instead the following inertial elements, i.e. $\mathbf{v}_t$, $\boldsymbol{\omega}_t$, and $\mathbf{a}_t$ are defined from the robot frame. As assumption, the system will start with a known initial state $\mathbf{X}_0$ of mean 0 for each element.

The transition matrix related to the states defined in Equation (3.27) and to the kinematic model of Figure 2.1 is given by (3.28).

$$A_t = \begin{bmatrix} 1 & 0 & 0 & \cos(\boldsymbol{\theta}_t) \cdot \Delta_t & 0 \\ 0 & 1 & 0 & \sin(\boldsymbol{\theta}_t) \cdot \Delta_t & 0 \\ 0 & 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

where $\boldsymbol{\Delta}_t$ is the time step used by the AEKF. It accounts for the transformation of the linear velocity, angular velocity, and the linear acceleration to update the position states that are used as DOF, as defined in Equation (2.1).

As defined in Section 2.1.2, the ALM is controlled using the linear and angular velocities. This configuration is used by the AEKF to improve the Prediction step using the following Control matrix B and states $\mathbf{u}_t$, as defined

in Equations (3.29).

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Delta_t & 0 \\ 0 & \Delta_t \end{bmatrix} \qquad \mathbf{u}_t = \begin{bmatrix} \mathbf{cmd_v} - \mathbf{v}_t \\ \mathbf{cmd}_{\boldsymbol{\omega}} - \boldsymbol{\omega}_t \end{bmatrix} \tag{3.29}$$

where Δ_t is the time step of the AEKF used to make the transition from command received and actual actuation more smooth by dynamically scaling the innovation, and $\mathbf{cmd_v}$ and $\mathbf{cmd}_{\boldsymbol{\omega}}$ are the commands sent to the robot.

The prediction of the estimate regarding the state variables is defined in the KF by equation (3.30).

$$\mathbf{X}_p = A \cdot \mathbf{X}_t + B \cdot \mathbf{u}_t + \boldsymbol{\eta} \tag{3.30}$$

As the Kalman Filter is not able to deal with non-linear systems, the Jacobian of the Transition matrix must be obtained to linearise the non-linear equations with a first-order Taylor expansion. The resulting matrix J_A, defined in (3.31), can then be used by the AEKF for the prediction of the Covariance matrix.

$$J_A = \begin{bmatrix} 1 & 0 & 0 & -\sin(\boldsymbol{\theta}_t) \cdot \Delta_t & 0 \\ 0 & 1 & 0 & \cos(\boldsymbol{\theta}_t) \cdot \Delta_t & 0 \\ 0 & 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.31}$$

The prediction noise matrix, given by Q in Equation (3.32), is defined by uncorrelated and zero-mean white noise, with the diagonal elements set dynamically as the related noise of the state variables.

$$Q = \begin{bmatrix} \sin(\boldsymbol{\theta}_t)^2 \cdot \frac{{\Delta_t}^2}{2} & -\sin(\boldsymbol{\theta}_t) \cdot \cos(\boldsymbol{\theta}_t) \cdot \frac{{\Delta_t}^2}{2} & 0 & -\sin(\boldsymbol{\theta}_t) \cdot {\Delta_t}^2 & 0 \\ -\sin(\boldsymbol{\theta}_t) \cdot \cos(\boldsymbol{\theta}_t) \cdot {\Delta_t}^2 & \cos(\boldsymbol{\theta}_t)^2 \cdot \frac{{\Delta_t}^2}{2} & 0 & \cos(\boldsymbol{\theta}_t) \cdot {\Delta_t}^2 & 0 \\ 0 & 0 & \frac{{\Delta_t}^2}{2} & 0 & {\Delta_t}^2 \\ -\sin(\boldsymbol{\theta}_t) \cdot {\Delta_t}^2 & \cos(\boldsymbol{\theta}_t) \cdot {\Delta_t}^2 & 0 & \Delta_t & 0 \\ 0 & 0 & {\Delta_t}^2 & 0 & \Delta_t \end{bmatrix} \cdot \sigma_a \tag{3.32}$$

where $\boldsymbol{\Delta}_t$ is the time step used by the AEKF and σ_a is the mean of the acceleration along the X axis of the robot frame, as measured by the IMUs or given by the costant $g = 9.81$ if no IMUs are available, as done in [42]. This formulation is provided to model the error in the process as a function of the

acceleration. The diagonal assumption means that the errors between variables are not directly associated, but their relationships and related propagation of errors will appear through the transition matrix Jacobian J_A [42].

Finally, the predicted covariance matrix, defined by $\hat{P}_{t+1}$ is updated during the prediction using the Jacobian J_A and the process noise covariance Q, as in Equation (3.33).

$$\hat{P}_{t+1} = J_A \cdot P_t \cdot J_A^T + Q \tag{3.33}$$

3.3.2 Correction Model

Gather all measurements and then use those combined measures for one single correction step, following a batch fusion approach.

Use of thread locks to avoid issues related to out of sequence measurements.

Joseph Form Covariance update is adopted, given the superior numerical performances obtained. (More time consuming) - Low improvements still relevant to be applied.

3.4 Mapping Configuration

The occupancy grid is defined using ROS. It is composed by a grid of cells of dimension $150\,\mathrm{m}$ x $150\,\mathrm{m}$, with the center positioned in the middle of this square at position ($75\,\mathrm{m}$, $75\,\mathrm{m}$), and with a resolution of $0.1\,\mathrm{m}$. In this way, it will be possible to manage an area of $5000\,\mathrm{m}^2$ whichever the configuration of the lawn.

3.4.1 Virtual Boundary

The external boundaries are defined as -1 values, and so if the automower finds itself close to or directly on a cell with a value of -1, it will change its orientation and continue on his tasks. The rest of the cells are initialised with value 50, meaning that the initial assumption is that the environment is uncertain and that there might be obstacles present within it.

At the first run of the automower, the estimated trajectory derived by the EKF will be used to set the corresponding boundary, i.e. by setting the cells it directly travelled through to -1.

After this guided procedure, the automower will have an estimate of its environment configuration and a virtual boundary to delimit its operations.

3.4.2 Map Update

After this phase of boundary definition, the ALM is left free to roam on that area. Even if it is defined with cells containing values of 0, it might still present unexpected objects and those are identified through the collision sensor.

The value of each cell then contains the probability that an object is present in that position, with those values being updated through an average of its previous occupancy probability and the eventuality of collision events detected by the ALM.

Each grid cell of the occupancy grid map contains different values of the probabilities that rely on the sonar range information. The probability for occupied grid cell should be equal to (0.98) for the maximum self-belief for various robotics task [3,4,5]. But the probability for the occupied grid cell varied as the range information from sonar varies corresponding to the grid cell under consideration [3].

Mathematical analysis for the investigation of the importance of probabilistic model in occupancy grid mapping is considered that is based on Bayesian technique. Bayesian theorem is an efficient technique to fuse the information of two occupancy grid map on the basis of probabilistic model as shown below.

To do this, conditional independence is assumed between grid cells.

The equation to update the cells is based on the Bayes Theorem. As defined in [55] and [57], then the equation to update it is shown below:

$$P_{i,j}^{t+1} = \frac{P_{i,j}^{t} \cdot P_{env}}{P_{i,j}^{t} \cdot P_{env} + (1 - P_{i,j}^{t}) \cdot (1 - P_{env})} \tag{3.34}$$

, where $P_{i,j}^{t}$ is the final value of cell i, j, $P_{i,j}^{t-1}$ is the previous value of the same cell, and P_{env} is the collision event value. This update equation has the important advantage that it is associative and commutative, enabling to incorporate measurements in any order.

3.5 Evaluation Framework

The evaluation configuration is defined, showing how the system developed is performing and how its performance are going to be evaluated.

Multiple tests are made. The teleoperation script available is used to control the mobile robot in velocity. The measures obtained by an experiments are stored in ROS bags which can be used later on to test the filter multiple

times with the same experiment.

Different configurations that are going to be tested provide different observabilty performance for the AEKF filter. observability

Finally, to evaluate if the objectives have been fulfilled and the results can be evaluated as valuable, the following steps are required:

- The quantitative measures need to be tested in a real use case experiment, as specified in the research questions.
- The data, acquired during these testing phases, will be used and analysed to check if the set of objectives have been reached.
- A statistical analysis over multiple experimental tests will provide more reliable results to be presented in the final report.

To gather data used to test and analyse the performance of the system, multiple outdoor tests have been carried on. Some tests have been made to define the calibration of the sensors against a ground truth identified using measure tape. Some other tests have been made to ensure that the system is behaving accurately while steering. Some tests have been made to check that the collision events are estimated with an certain degree of precision.

During these tests the rosbag utils provided by ROS has been exploited. Through this service it is possible to store every message sent to the topics, and they can be reused in the future to tune the algorithms. In this way, the real measurements gathered during outdoor testing can be re evaluated through the play of said messages and using a simulated parameter to reset the clock of ROS framework.

3.5.1 Ground Truth Generation

The ground truth is constructed manually by checking the actual path travelled by the robot with the usage of a meter. Setting the experiment and having the robot to follow that path manually, commanding it using the given teleoperation script.

3.5.2 Simulation design

Simulations will provide an overview of the sensor fusion should behave.

It will use the measures obtained by the Ground Truth. As a real measure is obtained by a sensors, a simulated measure is generated from the ground truth using a Gaussian white noise to satisfy the Kalman assumptions.

These simulated measures are used by the Kalman filter to offer its estimate.

3.5.3 Data Analysis Technique

To compare the performances of different configurations, different experiments have been run. The different performance offered by different configuration is provided comparing the measures of the desired aspects with the frequency of the EKF, i.e. at each step of the Sensor Fusion process the estimates of the pose are compared with the Ground Truth.

3.5.4 Evaluation Metric

Root Mean Square Error (RMSE) is used to compute the performance at each step. It uses the whole history of errors and then provides its value.

$$RMSE_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GroundX_j^{(i)} - CheckX_j^{(i)})^2} \tag{3.35}$$

where j indicates the value of the state that is evaluated, i indicates the step to be evaluated, N is the number of comparisons based on the steps computed, $GroundX_j^{(i)}$ is the ground truth value at that step, $CheckX_j^{(i)}$ is the value to be compared for performance evaluation.

Chapter 4

Experiments and Results

The results obtained from the localisation and mapping feature added to the ALM are shown.

4.1 Localisation Analysis

Analysis of the Simulated and Outdoor Experiment. Real Test configuration and environment in outdoor settings.

Test on real measurements gathered in different conditions of terrain, slope, and weather. Multiple tests have been repeated to measure the measurement's value noise of each sensors.

Model	**Sensors**						**Description**
	Control	WO	GNSS	GNSSs	IMUs	VO	
1	YES	YES	YES	NO	NO	NO	Original
2	YES	YES	YES	YES	YES	YES	Complete
3	YES	YES	YES	NO	YES	YES	Reliable
3-1	YES	YES	YES	NO	YES	NO	Reliable
4	YES	YES	NO	NO	YES	NO	Cheap solution
5	YES	NO	YES	NO	NO	YES	-WO+VO
6	NO	YES	YES	NO	YES	YES	No Control

Table 4.1: Experiments configuration

4.1.1 Simulated Experiment

The extensive and comprehensive experiment run with all the measures available is analysed here.

The following localisation performance have been obtained by the different configurations of sensors.

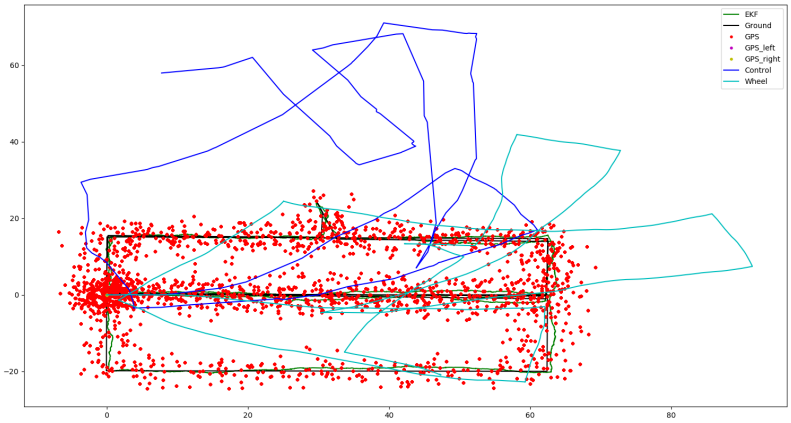


Figure 4.1: Simulation Experiment Results

These results have been evaluated using the RMSE metric, the values obtained are showed in the Table below.

Model	**Evaluation Criteria**						
	Final Estimate			**RMSE**			**Time step**
	$\mathbf{x} \pm \sigma_{\mathbf{x}}$[m]	$\mathbf{y} \pm \sigma_{\mathbf{y}}$[m]	$\boldsymbol{\theta} \pm \sigma_{\boldsymbol{\theta}}$[rad]	x[m]	y[m]	θ[rad]	$\Delta_t \pm \sigma_{\boldsymbol{\Delta}_t}$ [ms]
SIM 1	4.35 ± 1.25	4.10 ± 0.44	0.60 ± 0.04	10.35	5.10	1.6	4.23 ± 0.10
SIM 2	40.35 ± 11.25	4.1	0.6	10.35	5.10	1.6	4.23 ± 0.10
SIM 3	4.35	14.10	0.6	10.35	5.10	1.6	4.23 ± 0.10
SIM 4	4.35	4.10	0.6	10.35	5.10	1.6	4.23 ± 0.10
SIM 5	4.35	4.10	0.6	10.35	5.10	1.6	4.23 ± 0.10
SIM 6	4.35	4.10	0.6	10.35	5.10	1.6	4.23 ± 0.10

Table 4.2: Simulated experiments results

4.1.2 Outdoor Experiment

The extensive and comprehensive experiments run with all the measures available is analysed here.

The following localisation performance have been obtained by the different configuration of sensors.

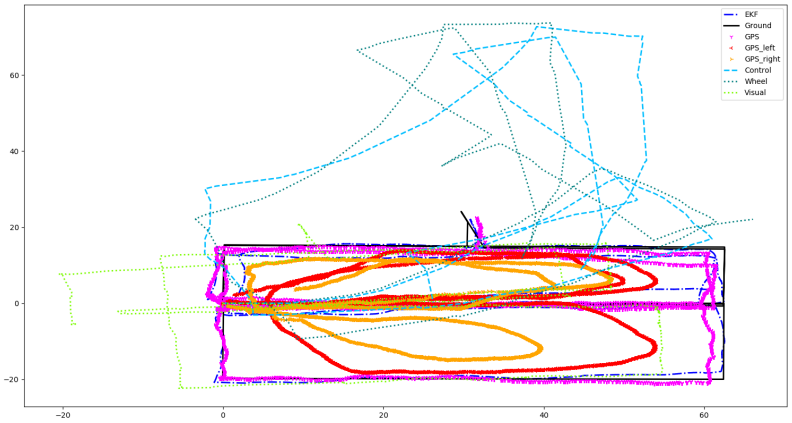


Figure 4.2: Outdoor Experiment Results

These results have been evaluated using the RMSE metric, the values obtained are showed in the Table below.

Model	**Evaluation**							
	Final Estimate			**RMSE**			**Time step**	
	x[m]	y[m]	θ[rad]	x[m]	y[m]	θ[rad]	μ[ms]	σ[ms]
1	4.35	4.10	0.6	10.35	5.10	1.6	4.23	0.1
2	4.3	4.10	0.6	10.35	5.10	1.6	4.23	0.1
3	10.35	4.10	0.6	10.35	5.10	1.6	4.23	0.1
4	4.35	4.10	0.6	10.35	5.10	1.6	4.23	0.1
5	10.35	4.10	0.6	10.35	5.10	1.6	4.23	0.1
6	4.35	4.10	0.6	10.35	5.10	1.6	4.23	0.1

Table 4.3: Outdoor experiments results

4.2 Mapping Analysis

The evolution of the mapping feature is defined here, from the initial setting, to the customised virtual boundary, and finally to the updating map behaviour.

The system adopted to localise the ALM during this phase is "*Model 2*" as defined in Section 4.1

4.2.1 Occupancy Grid Boundary

The virtual boundary is then set with the initial run setting of the personalized virtual boundary. The external area is set to the outdoor limited area.

With this virtual boundary, the ALM is able to stay within the designed area.

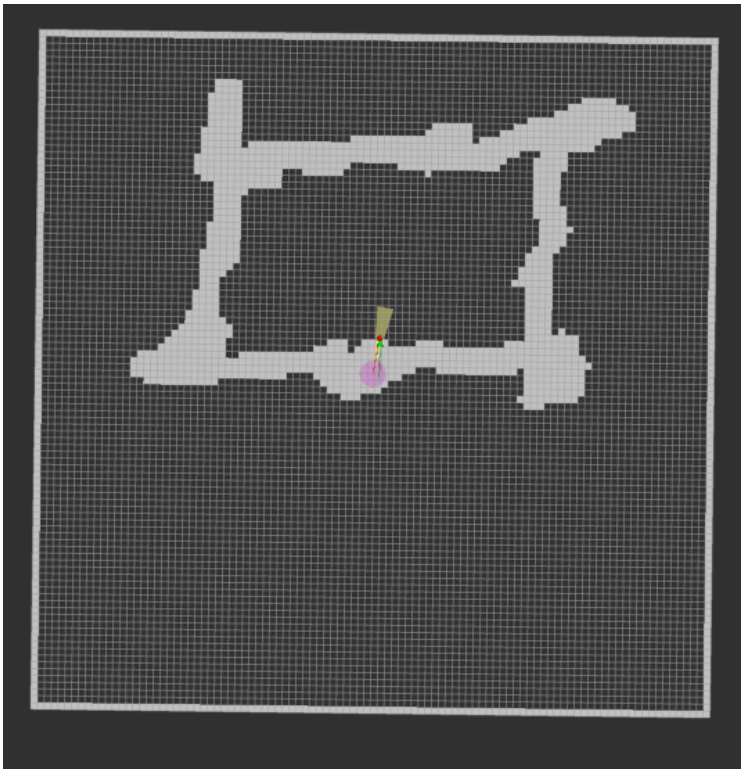


Figure 4.3: Personalised Boundary definition

4.2.2 Occupancy Grid Update

The virtual boundary is then set with the initial run setting of the personalised virtual boundary.

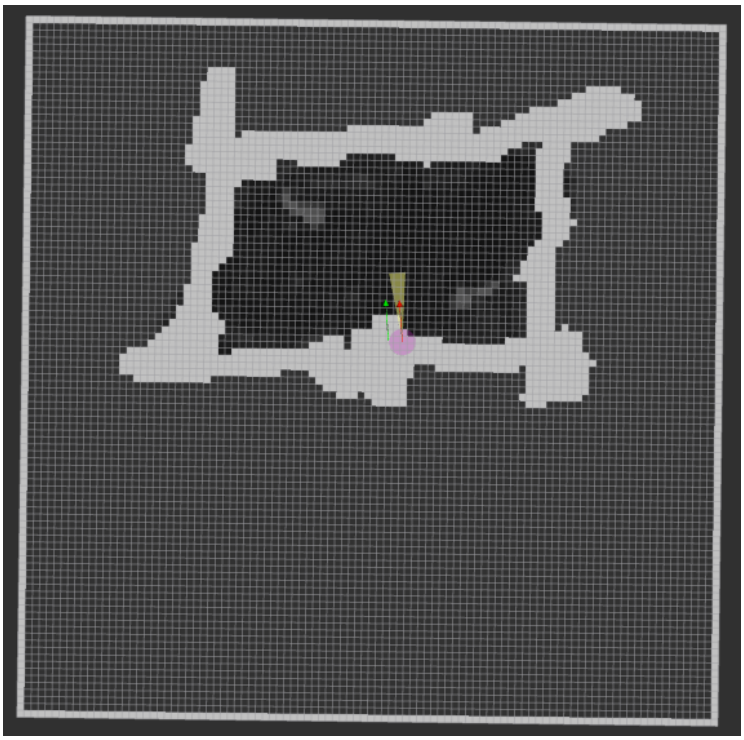


Figure 4.4: Update of the map inside the boundaries (lighter shade means probability of object given by collision events)

Chapter 5

Discussions

After this master thesis, two related topics have been investigated. The findings are discussed below.

5.1 Localisation

Using a sensor fusion approach, such as AEKF, it is possible to reach fast and robust localisation estimates.

Not all the sensors behaved as expected and the best configuration has not been achieved using all the sensors available as expected.

5.2 Mapping

With the improved localisation, it is possible to limit the mowing area of the ALM using a virtual boundary.

Using collision sensors and accounting for the robot pose uncertainty it is possible to provide a better understanding of the robot environment.

5.3 Limitations

As the system has been tested experimentally on the field, the conditions to run it outside were needed. This limited the available amount of time to do outdoor testing in a relevant matter, especially given the raining days.

Being the system an hardware implementation, different aspects needed to be tested before running the tests, removing time to more advanced tuning of the implementation.

Chapter 6

Conclusions

This project managed to fuse the measurements of multiple and heterogeneous sensors. It has been demonstrated how the implementation of this sensor fusion technique exploited a subset of sensors to achieve more accurate localisation estimates.

The mapping feature has been achieved by setting virtual boudaries through initial runs and then updating its knowledge using collision sensor events.

6.1 Limitations

A simple measurement tape, and accurate definition of the path with strings have been used to define the ground truth used to evaluate the system. A more precise way to compute the Ground Truth might be useful.

""" The limitations of rosbag record/play In the previous section you may have noted that the turtle's path may not have exactly mapped to the original keyboard input - the rough shape should have been the same, but the turtle may not have exactly tracked the same path. The reason for this is that the path tracked by turtlesim is very sensitive to small changes in timing in the system, and rosbag is limited in its ability to exactly duplicate the behavior of a running system in terms of when messages are recorded and processed by rosrecord, and when messages are produced and processed when using rosplay. For nodes like turtlesim, where minor timing changes in when command messages are processed can subtly alter behavior, the user should not expect perfectly mimicked behavior. """

6.2 Future works

The system could be expanded and improved in multiple directions.

Expanding the states of the system in order to navigate in a 3D environment instead than a simple 2D environment as in this project. It will be possible to account for slopes in the lawn and it could be interesting to evaluate its influence in the GPS readings.

Outliers detection in case of non reliable sensors and once the covariances noises matrices have been ultimately perfected. Moreover, it can be crucial when the sensors are not behaving correctly.

Intrinsic and Extrinsic Calibration
Currently, we are considering the case of a mobile robot whose odometry is not calibrated. In this case, the observability analysis will extend to the parameters characterizing the odometry error (e.g. wheel diameters, distance between the wheel).

Exploit depth information from the camera to provide a faster identification of collision events, eventually providing a map of the environment based on depth information instead of collision events. Instead of using the collision sensors to update the map, the camera, and its depth readings, could be used for object detection to identify and avoid the objects before collision.

It could be investigated a trade off between computations needed for a reliable SLAM implementation instead of an online implementation provided in this report.

Path planning algorithm as in [65] and [66], can be used to provide coverage features to the autonomous lawn mowers. A comprehensive overview about coverage is available in [67]. It discusses all the available techniques and their related strengths and weaknesses. More specific approaches to be investigated can be found in [68] and [69]. Regarding the avoidance of the objects for the unexpected detection of objects, a techniques is define in [70].

The last aspect of navigation program is locomotion [2].

6.3 Reflections

This project will address the following issues, including ethics and sustainability:

- Waste reduction: keeping the localisation module included in the automower will remove the need to install additional infrastructure

which could then be susceptible to deterioration.

- Bees preservation: Thanks to a more accurate control of the lawn with a given map, it will be possible to dynamically update it and preserve a part of the lawn to allow bees to prosper, without intervening on the boundary wire.
- Privacy issue: the images needed for the visual odometry module are going to be used inside the automower, not shared outside of it. They will be destroyed after their processing, avoiding breach of personal privacy in case of theft of the memory card.
- Economic aspects: as the ALM is a consumer product, the comparison between the available and proposed approach is relevant. A cheaper solution that provides more features might attract more customers.

6.3.1 Environmental

With a more precise localisation and navigation system the intervention on the lawn will be reduced. Also, the external boundary configuration produces some wastes that the virtual boundary do not have.

Moreover, with the possibility to set virtual boundaries for the automower, it will be possible to save part of the lawn for bees to find flowers to sustain themselves and prosper.

6.3.2 Security

The system is self-contained, and every information is stored inside the embedded Raspberry Pi 4. The single output that might be relevant to share can be the current position of the automower and the updated map that he has available for navigation.

The online system that doesn't store the history of the states also helps with the security of the information stored.

It can be easily protected through a password to avoid theft of information. The SD card could be encrypted also to avoid theft of it and retrieval of information through that memory.

6.3.3 Economical Analysis

As an economic results, by simply adding additional GPS, IMU, and RGB-D camera it is possible to remove the need of the boundary wire. As such, by

adding to the automower the cost of these additional sensors it will be possible to save money in the additional maintenance required. Overall, given the high reliability over time of these sensor, the consumers will be able to save money in the long run. However, the product-market fit might not be the best, since the company that will sell the automower also provides the boundary wire and removing that need will economically lower their revenues in that aspects. Anyway, the customers will be more willing to buy a system that requires less maintenance. So the trade-off will be between the more services offered and the lower revenues from selling additional infrastructures. Analysis of costs of sensors and usual boundary wire, plus related maintenance

It saves also the energy that now is given to the boundary wire to power the magnetic field that the automower need to detect its boundaries now.

References

[1] H. Sahin and L. Guvenc, "Household robotics: autonomous devices for vacuuming and lawn mowing [applications of control]," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 20–96, 2007. doi: 10.1109/MCS.2007.338262

[2] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Comput. Surv.*, vol. 53, no. 1, Feb. 2020. doi: 10.1145/3368961. [Online]. Available: https://doi.org/10.1145/3368961

[3] Karol, Ardıç, "A Conditional coverage path planning method for an autonomous lawn mower," Ph.D. dissertation, Middle East Technical University, 2016. [Online]. Available: https://webcache.googleusercontent.com/search?q=cache:Q-UGRS17DukJ:https://open.metu.edu.tr/handle/11511/25650+&cd=1&hl=en&ct=clnk&gl=se

[4] M. Wooldridge and N. R. Jennings, "Agent theories, architectures, and languages: A survey," in *Intelligent Agents*, M. J. Wooldridge and N. R. Jennings, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. ISBN 978-3-540-49129-3 pp. 1–39.

[5] J. P. MÜLLER, "Architectures and applications of intelligent agents: A survey," *The Knowledge Engineering Review*, vol. 13, no. 4, p. 353–380, 1999. doi: 10.1017/S0269888998004020

[6] M. Wahde, "Introduction to autonomous robots," *Lecture Notes from the course Autonomous Agents, Chalmers university of technology*, 2012.

[7] M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*. Elsevier Science, 1987. ISBN 978-0-12-801554-4

[8] A. Håkansson, "Portal of Research Methods and Methodologies for Research Projects and Degree Projects." CSREA Press U.S.A, 2013, pp. 67–73. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-136960

[9] F. A. Salem, "Dynamic and kinematic models and control for differential drive mobile robots," *International Journal of Current Engineering and Technology*, vol. 3, no. 2, pp. 253–263, 2013.

[10] M. Ben-Ari and F. Mondada, *Robotic Motion and Odometry*, 01 2018, pp. 63–93. ISBN 978-3-319-62532-4

[11] K. M. Lynch, N. Marchuk, and M. L. Elwin, "Chapter 21 - Sensors," in *Embedded Computing and Mechatronics with the PIC32*, K. M. Lynch, N. Marchuk, and M. L. Elwin, Eds. Oxford: Newnes, 2016, pp. 317–340. ISBN 978-0-12-420165-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124201651000214

[12] P. Lang, A. Kusej, A. Pinz, and G. Brasseur, "Inertial tracking for mobile augmented reality," vol. 2, 02 2002. doi: 10.1109/IMTC.2002.1007196. ISBN 0-7803-7218-2 pp. 1583 – 1587 vol.2.

[13] N. Magnusson and T. Odenman, "Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion," 2012, accepted: 2019-07-03T12:50:26Z. [Online]. Available: https://odr.chalmers.se/handle/20.500.12380/159412

[14] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, and M. Reinstein, "Robust data fusion of multimodal sensory information for mobile robots," *Journal of Field Robotics*, vol. 32, no. 4, pp. 447–473, 2015. doi: https://doi.org/10.1002/rob.21535. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21535

[15] X. Gao, T. Zhang, Y. Liu, and Q. Yan, *14 Lectures on Visual SLAM: From Theory to Practice*, 2017.

[16] T. Han, C. Xu, R. Loxton, and L. Xie, "Bi-objective optimization for robust rgb-d visual odometry," 11 2014. doi: 10.1109/CCDC.2015.7162218

[17] H. Mitchell, *Multi-Sensor Data Fusion: An Introduction*. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-71559-7. [Online]. Available: https://books.google.se/books?id=2hwcFSxQ1CAC

[18] K. Faceli, A. de Carvalho, and S. Rezende, "Combining intelligent techniques for sensor fusion," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, vol. 4, Nov 2002. doi: 10.1109/ICONIP.2002.1199023 pp. 1998–2002 vol.4.

[19] A. Weckenmann, X. Jiang, K.-D. Sommer, U. Neuschaefer-Rube, J. Seewig, L. Shaw, and T. Estler, "Multisensor data fusion in dimensional metrology," *CIRP Annals*, vol. 58, no. 2, pp. 701–721, 2009. doi: https://doi.org/10.1016/j.cirp.2009.09.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850609001759

[20] F. Gustafsson, *Statistical sensor fusion*. Studentlitteratur, 2010. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-74627

[21] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. doi: 10.1115/1.3662552. [Online]. Available: https://doi.org/10.1115/1.3662552

[22] *Adaptive Estimation and Maneuvering Targets*. John Wiley & Sons, Ltd, ch. 11, pp. 421–490. ISBN 9780471221272. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471221279.ch11

[23] R. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970. doi: 10.1109/TAC.1970.1099422

[24] J. Schmidt, "Analysis of Square-Root Kalman Filters for Angles-Only Orbital Navigation and the Effects of Sensor Accuracy on State Observability," May 2010. [Online]. Available: https://digitalcommons.usu.edu/etd/627

[25] Y. Bar-Shalom, X. Li, and T. Kirubarajan, "Estimation with applications to tracking and navigation: Theory, algorithms and software," 2001.

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 978-0-262-20162-9

[27] M. Schlosser and K. Kroschel, "Limits in tracking with extended kalman filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1351–1359, 2004. doi: 10.1109/TAES.2004.1386886

[28] Y. Xu, X. Chen, and Q. Li, "Adaptive Iterated Extended Kalman Filter and Its Application to Autonomous Integrated Navigation for Indoor Robot," Feb. 2014, iSSN: 2356-6140 Pages: e138548 Publisher: Hindawi Volume: 2014. [Online]. Available: https://www.hindawi.com/journals/tswj/2014/138548/

[29] "GroundsBot," Master of Science - Robotic Systems Development Program, School of Computer Science, Carnegie Mellon University., Tech. Rep. [Online]. Available: https://mrsdprojects.ri.cmu.edu/2017teama/

[30] B. Lukáš, "Robotic Lawn Mower," Master's thesis, Czech Technical University in Prague, Prague, May 2020. [Online]. Available: https://dspace.cvut.cz/bitstream/handle/10467/88202/F3-DP-2020-Bauer-Lukas-DP_elektro.pdf

[31] P. Andersson, F. Labe, J. Mejervik Derander, E. Moen, A. Nitsche, and E. Wennerberg, "Smart robot lawn mower Robot lawn mower without need for a boundary cable around the lawn," 2018, accepted: 2019-07-03T14:54:57Z. [Online]. Available: https://odr.chalmers.se/handle/20.500.12380/256114

[32] L. Oden and F. Stenbeck, *Localization and Mapping forOutdoor Mobile Robots with RTKGPS and Sensor Fusion : An Investigation of Sensor Technologies for theAutomower Platform*, 2017. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-216993

[33] F. Lensund and M. Sjöstedt, *Local positioning system for mobile robots using ultra wide-band technology*, 2018. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-232506

[34] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, "Where am i? localization techniques for mobile robots a review," in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 2020. doi: 10.1109/ICMRE49073.2020.9065135 pp. 43–47.

[35] H. B. Mitchell, *Multi-sensor data fusion: an introduction*. Springer Science & Business Media, 2007.

[36] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Intelligent Autonomous Systems 13*, ser. Advances in Intelligent Systems and Computing, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-08338-4_25. ISBN 978-3-319-08338-4 pp. 335–348.

[37] T. Larsen, K. Hansen, N. Andersen, and O. Ravn, "Design of kalman filters for mobile robots; evaluation of the kinematic and odometric approach," in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, vol. 2, 1999. doi: 10.1109/CCA.1999.801027 pp. 1021–1026 vol. 2.

[38] G.-S. Cai, H.-Y. Lin, and S.-F. Kao, "Mobile robot localization using gps, imu and visual odometry," in *2019 International Automatic Control Conference (CACS)*, 2019. doi: 10.1109/CACS47674.2019.9024731 pp. 1–6.

[39] R. Sun, Y. Yang, K.-W. Chiang, T.-T. Duong, K.-Y. Lin, and G.-J. Tsai, "Robust imu/gps/vo integration for vehicle navigation in gnss degraded urban areas," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 110–10 122, 2020. doi: 10.1109/JSEN.2020.2989332

[40] H. CHEN, "Gps-oscillation-robust localization and visionaided odometry estimation," Master's thesis, KTH, Machine Design (Dept.), 2019.

[41] K. Berntorp, K.-E. Årzén, and A. Robertsson, "Sensor fusion for motion estimation of mobile robots with compensation for out-of-sequence measurements," 2011, pp. 211–216.

[42] P. H. King, "A Low Cost Localization Solution Using a Kalman Filter for Data Fusion," Thesis, Virginia Tech, Apr. 2008, accepted: 2014-03-14T20:35:41Z. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/32384

[43] I. Skog and P. Händel, "A low-cost gps aided inertial navigation system for vehicle applications," in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–4.

[44] A. Mohamed and K. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of geodesy*, vol. 73, no. 4, pp. 193–203, 1999.

[45] F. Grandoni, A. Martinelli, F. Martinelli, S. Nicosia, and P. Valigi, "Sensor Fusion for Robot Localisation," in *Ramsete: Articulated and Mobile Robotics for Services and Technologies*, ser. Lecture Notes in Control and Information Sciences, S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi, Eds. Berlin, Heidelberg: Springer, 2001, pp. 251–273. ISBN 978-3-540-45000-9. [Online]. Available: https://doi.org/10.1007/3-540-45000-9_10

[46] Q. Kong, S. Xu, and S. Lee, "Using PDOP to estimate Kalman filters measurement noise covariance for GPS Positioning," 2012. [Online]. Available: /paper/using-PDOP-to-estimate-Kalman-filters-measurement-Kong-Xu/f8ea4a6b68af881f29ee77874108a77f79beb168

[47] A. Werries and J. M. Dolan, "Adaptive kalman filtering methods for low-cost gps/ins localization for autonomous vehicles," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-16-18, May 2016.

[48] Y. Hao, A. Xu, X. Sui, and Y. Wang, "A Modified Extended Kalman Filter for a Two-Antenna GPS/INS Vehicular Navigation System," *Sensors*, vol. 18, no. 11, p. 3809, Nov. 2018. doi: 10.3390/s18113809 Number: 11 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3809

[49] F. Chenavier and J. L. Crowley, "Position estimation for a mobile robot using vision and odometry," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, May 1992. doi: 10.1109/ROBOT.1992.220052 pp. 2588–2593 vol.3.

[50] T. Amador, "Robot Localization in an Autonomous Ground Vehicle," Feb. 2019, accepted: 2019-02-08T19:24:59Z Publisher: California State Polytechnic University, Pomona. [Online]. Available: http://dspace.calstate.edu/handle/10211.3/207956

[51] E. I. Al Khatib, M. A. Jaradat, M. Abdel-Hafez, and M. Roigari, "Multiple sensor fusion for mobile robot localization and navigation using the extended kalman filter," in *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, 2015. doi: 10.1109/ISMA.2015.7373480 pp. 1–5.

[52] L. A. Nguyen, P. T. Dung, T. D. Ngo, and X. T. Truong, "Improving the accuracy of the autonomous mobile robot localization

systems based on the multiple sensor fusion methods," in *2019 3rd International Conference on Recent Advances in Signal Processing, Telecommunications Computing (SigTelCom)*, 2019. doi: 10.1109/SIGTELCOM.2019.8696103 pp. 33–37.

[53] Y. Liu, R. Xiong, Y. Wang, H. Huang, X. Xie, X. Liu, and G. Zhang, "Stereo visual-inertial odometry with multiple kalman filters ensemble," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6205–6216, 2016.

[54] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[55] D. Joubert, "Adaptive occupancy grid mapping with measurement and pose uncertainty," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2012.

[56] L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid-based representation," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988. doi: 10.1109/ROBOT.1988.12145 pp. 727–733 vol.2.

[57] R. Singh and K. S. Nagla, "Sonar Sensor Model for the Precision Measurement to Generate Robust Occupancy Grid Map," *MAPAN*, vol. 34, no. 2, pp. 239–257, Jun. 2019. doi: 10.1007/s12647-018-0289-x. [Online]. Available: https://doi.org/10.1007/s12647-018-0289-x

[58] M. Braun, "First-order differential equations," in *Differential Equations and Their Applications: An Introduction to Applied Mathematics*. New York, NY: Springer New York, 1993, pp. 1–126. ISBN 978-1-4612-4360-1. [Online]. Available: https://doi.org/10.1007/978-1-4612-4360-1_1

[59] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975. doi: 10.1179/sre.1975.23.176.88. [Online]. Available: https://doi.org/10.1179/sre.1975.23.176.88

[60] I. Markovic, J. Cesic, and I. Petrovic, "On wrapping the Kalman filter and estimating with the SO(2) group," *arXiv:1708.05551 [cs]*, Aug.

2017, arXiv: 1708.05551. [Online]. Available: http://arxiv.org/abs/1708.05551

[61] "HusqvarnaResearch/hrp," Dec. 2020, original-date: 2017-03-16T09:20:21Z. [Online]. Available: https://github.com/HusqvarnaResearch/hrp

[62] Tianze, "TianzeLi/hrp_myversion," Feb. 2021, original-date: 2020-05-06T15:49:15Z. [Online]. Available: https://github.com/TianzeLi/hrp_myversion

[63] M. Labbé and F. Michaud, "Memory management for real-time appearance-based loop closure detection," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011. doi: 10.1109/IROS.2011.6094602 pp. 1271–1276.

[64] ——, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. doi: https://doi.org/10.1002/rob.21831 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21831. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21831

[65] I. Hameed, "Coverage path planning software for autonomous robotic lawn mower using dubins' curve," 07 2017. doi: 10.1109/RCAR.2017.8311915

[66] H. Van Pham and P. Moore, "Robot coverage path planning under uncertainty using knowledge inference and hedge algebras," *Machines*, vol. 6, no. 4, 2018. doi: 10.3390/machines6040046. [Online]. Available: https://www.mdpi.com/2075-1702/6/4/46

[67] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013. doi: 10.1016/j.robot.2013.09.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300167X

[68] I. Hameed, "Coverage Path Planning Software for Autonomous Robotic Lawn Mower using Dubins' Curve," Jul. 2017. doi: 10.1109/RCAR.2017.8311915

[69] T. M. Cabreira, P. R. Ferreira, C. D. Franco, and G. C. Buttazzo, "Grid-Based Coverage Path Planning With Minimum Energy Over Irregular-Shaped Areas With Uavs," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019. doi: 10.1109/ICUAS.2019.8797937 pp. 758–767, iSSN: 2575-7296.

[70] K. A. Daltorio, A. D. Rolin, J. A. Beno, B. E. Hughes, A. Schepelmann, M. S. Branicky, R. D. Quinn, and J. M. Green, "An obstacle-edging reflex for an autonomous lawnmower," in *IEEE/ION Position, Location and Navigation Symposium*, May 2010. doi: 10.1109/PLANS.2010.5507339 pp. 1079–1092, iSSN: 2153-3598.

[71] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, International Society for Optics and Photonics. SPIE, 1997. doi: 10.1117/12.280797 pp. 182 – 193. [Online]. Available: https://doi.org/10.1117/12.280797

[72] P. Del Moral, "Non linear filtering: Interacting particle solution," *Markov Processes and Related Fields*, vol. 2, pp. 555–580, 03 1996.

[73] K. Murphy and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York, NY: Springer New York, 2001, pp. 499–515. ISBN 978-1-4757-3437-9. [Online]. Available: https://doi.org/10.1007/978-1-4757-3437-9_24

[74] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[75] T. Foote, "tf: The transform library," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2013. doi: 10.1109/TePRA.2013.6556373 pp. 1–6.

Appendix A

Competitor Analysis

As this master thesis has been developed under the programme of EIT Digital*, an analysis of the competitor will be performed to highlight the current state of the art at an industrial level for ALMs which require no boundary wire installation, understanding their current implementations, and evaluating possible improvements.

Toadi† is a startup company based in Belgium which offers as only product an ALM of the same name. It employs a single visual sensor system to provide the localisation features. It can mow an area of $4800\,\mathrm{m}^2$. Its installation is performed though a track and follow approach to determine the boundaries of the lawn, it has been in testing phase and it is available to the public from May 2021.

The LUNA platform from Inertial Systems‡ employs all the sensors listed above to provide a localisation and navigation module, which has been used for ALMs. This platform was developed to enable engineers to directly embed autonomous capability into existing equipment avoiding the development of a localisation module, as it is detailed in this thesis.

Kingdom Technologies§ provide an ALM at a monthly fee to commercial clients. It employs sensor fusion techniques to merge several different sensors' measures to improve the positioning of the robot on a lawn that could reach an area of $7500\,\mathrm{m}^2$.

Ambrogio L60 Elite S+¶ requires no installation for areas up to $400\,\mathrm{m}^2$. It employs a grass sensor to detect the presence of grass under itself and it uses

* https://masterschool.eitdigital.eu/ † https://www.toadi.com/ ‡ https://inertialsense.com/autonomy/ § https://www.kingdom.garden/ ¶ https://www.ambrogiorobot.com/en/models/view/l60-elite-s

a specific robotic joint to recognise any holes or empty spaces. With these sensors it is able to navigate around a defined lawn using a reactive approach, but without the need for the boundary wire.

Different approaches which do not require the boundary wire installation but which still need some external infrastructures are the following.

Husvarna 550 EPOS* is a new ALM available for commercial customers which employs a GPS-RTK sensor to cover areas of $5000\,\mathrm{m}^2$. It needs an external and expensive receiver to correct the GNSS estimates of the robot and improve its localisation performance to just a few centimeters error.

iRobot Terra t7† is the ALM It relies on wireless navigation beacons installed on the garden to triangulate the position of the mower using their signals. However, the implementation of such technology has concerned the National Radio Astronomy Observatory (NRAO), as the original intended frequency they used were disturbing the measurements of that organisation, and they had to change their beacons using UWB signal technology.

The currently developed ALMs are still far from being mature and available to all customers. For this reason, an analysis of the best configuration of sensors to improve on the localisation aspects without the need for an external installation is provided, as improved localisation performance of autonomous mobile robots are required.

Numerous related studies and implementations have been developed in similar topics to improve localisation features, and the analysis of an optimal configuration of sensors which require no external installations is still relevant to the best of the found knowledge.

* https://www.husqvarna.com/us/products/robotic-lawn-mowers/models/automower-550-epos/ † https://www.irobot.co.uk/Terra

Appendix B

Highly-Non-Linear Sensor Fusion

The system analysed in this thesis is not characterised by high non-linearity given the high frame-rate available, and as such the adopted AEKF is robust, fast, and sufficient.

A brief description of other linearisations, such as the UKF, and other approaches, such as Particle Filter (PF), are provided in this appendix. Even if they are not used in this thesis, they are noteworthy to mention to provide a more comprehensive overview of sensor fusion techniques, in case of highly non linear systems.

B.1 Unscented Kalman Filter

The UKF is an alternative to the KF for estimation of highly non linear systems [71]. Instead of using a linearisation approach, the UKF is based on a deterministic sampling method. The estimates are propagated through the non linear system using sample values, called sigma points, defined by the unscented transform. This procedure provides two values along the mean of the estimate and symmetrically placed on its main uncertainty axes. Although the results are not a Gaussian random variables, their mean and standard deviation can be derived from them to maintain the KF assumption.

Its improvements in estimates and covariance accuracy are relevant over the EKF when the functions f and h are highly non-linear, as it is accurate to the second order Taylor's expansion. Moreover, it does not need to compute Jacobians of the non linear functions of the system [26].

B.2 Particle Filter

The PF provides another implementation of a Bayesian filter [72]. It is an iterative algorithm which saves multiple potential state estimate values, known as particles. The probability the state estimate value in a moment in time is given by the density of particles nearby which contains similar values. At each iteration, a sampling phase is performed generating temporary particles using the current estimated state. Afterwards an importance factor is calculated using new measurements of the state and it provides a likelihood value for the next estimations. In the resampling phase, the proper particles are updated by using the estimates provided by one of the temporary particles with a selection based on the importance factor. This process is continuously iterate and the particles will eventually converge to a proper estimate over several steps.

The Rao-Blackwellized PF extends it using both particles and Gaussian random variables as state variables [73].

Appendix C

Robotic Operating System

ROS[74] is an open-source framework, and not an actual operating system, widely adopted for building software used to control robotic systems. ROS works both with C++ and Python programmed scripts, allowing for developers with different backgrounds to develop upon it.

For these nodes to cooperate, they are connected by a middle-layer structure based on a network of topics or services, following a blackboard architecture of data sharing. It provides a graph-like structure were each program, described as node, can both publish and subscribe messages among the created and available topics.

Topics can be seen as variables which are used for streaming communication and share information between nodes. A ROS node publishes a specific message through a topic and every other node can subscribe to that topic and receive all the messages posted to it. A ROS system, to behave as expected, needs a master process which has the task of matching publishers and subscribers to their related topics.

These topics can be generated by sensors, actuators, planners, controllers of different kind and made available using this structure of topics, similar to a blackboard architecture. Each topic is defined by a specific message type which specifies its content in order to standardise its usage among applications and to simplify communication between nodes. Messages definition can be customised and they are stored inside the ROS package, defining the data sent within the topic which will use that definition.

Figure C.1 shows an example of a ROS graph where two subscriber nodes receive the messages posted on the topic by a publisher node.

The aspect of localisation is important for the robotics field, to understand where the robot is located with respect to the rest of the world, but also to

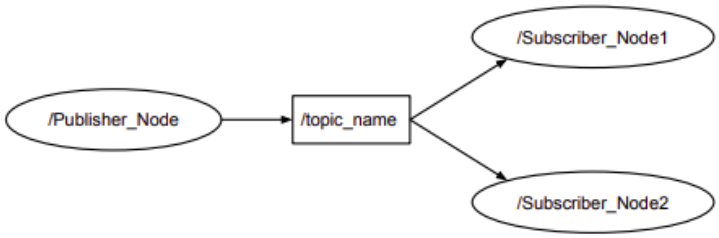


Figure C.1: ROS topics architecture

know how the sensors are positioned with respect to the robot in which they are installed. It is of prominent importance to be aware of the different pose of the connected sensors before fusing their information, as the measurements of each sensors are related to their specific coordinate frame.

Before performing sensors fusion, every different frames of the sensors need to be transformed into the base frame of the robot which they are measuring to be sure about that the measurements refers to the same coordinate frame.

ROS uses the TF[75] library to provide the tools needed to work with coordinate frames and to deal with their related transformation. It allows for the definition of each sensor rigid body transform to specify their related position with respect to the robot, and then the library will deal with all the other transformations by publishing messages regarding to the rotational and translational relations between frames. Commonly used frames are odom and base_link. The odom frame has the origin at the initial position P_{odom} of the robot and it is used to keep track of its moving behaviour. The base_link frame is rigidly attached to the robot base P_{base} and it is used to define the frames of its attached components.

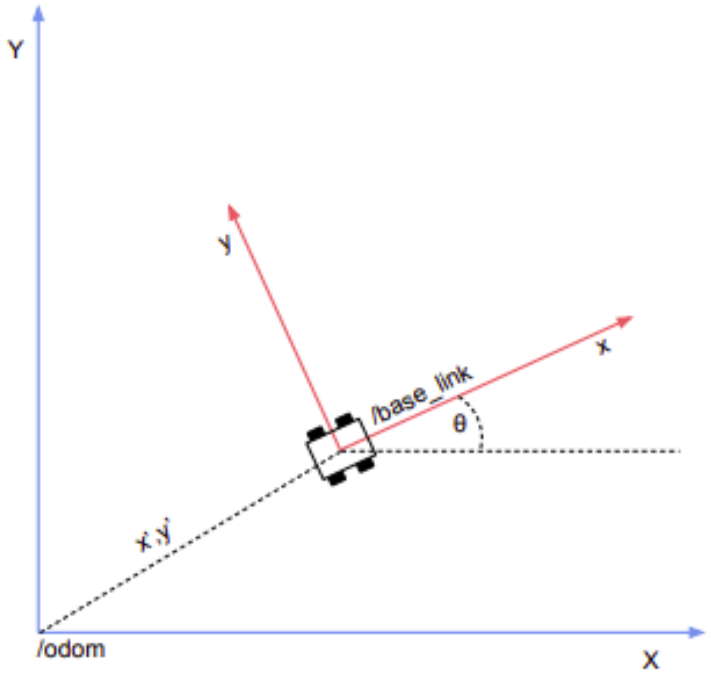


Figure C.2: ROS frames architecture

Appendix D

Repository

Everything that has been developed during this thesis could be found in https://github.com/boffomarco/hrp

The following files are the main contributions provided in this thesis.

D.1 System Configuration

.bashrc
network_file

Configuration of network settings between RPis and PC using ROS to be added in a file describing the procedure

D.2 Localisation Configuration

The improved localisation performance are provided using a personalised configuration of the sensors and the related sensor fusion of their measures.

sensors.launch
automower_safe.launch
imus_basic.launch
gps.launch
gps.cpp
rs_camera.launch
rtabmap.launch
BatchEKF.launch
BatchEKF_last.py
BatchEKF_lastRPi.py
BatchEKF_lastSimulated.py
GroundTruthSimulated.py

D.3 Mapping Configuration

OccupancyGrid.py

D.4 Results Configuration

plot_histogram.py
gps_plot.py

D.5 Software Configuration

C++ and Python development to work with the ROS framework, explained in details in Appendix C.

Network establishment to communicate wirelessly between Raspberry Pi and computer.

Communication through hotspot

D.5.1 Controller

The ALM behaviour is defined by the launching file `automower_hrp.launch`. It defines and launch the controller of the HRP.

D.5.2 Velocity Control

The script `hrp_teleop.lauch` as provided by the HRP platform is used to drive the ALM during the experiments. It controls the vehicle by setting its desired velocities: $\mathbf{v}v$ and $\boldsymbol{\omega}$. In this way it is possible to command the ALM to follow a trajectory.

D.5.3 Launching the sensors

The measurements are obtained from the heterogeneous sensors using the following methods.

The IMUs sensors are started using the file `imus_basic.launch`. In this file, the node defined by the `phidget_spatial` drivers is used, each sensor will belong to its group, and its coordiante frame is defined by the installation coordinates.

The additional Phidget GPS sensors are launched by the file `imus_basic.launch`. In the file, the node calls for a personalised driver that has been implemented to gather the National Marine Electronics Association (NMEA) data directly provided by the Phidgets Application Programming Interface (API). The information provided by this NMEA data structure

The additional Phidget GPS sensors are launched by the file `imus_basic.launch`. In the file, the node calls for a personalised driver that has been implemented to gather the NMEA data directly through the Phidgets API. The information provided by this NMEA data structure covers both the localisation estimate and its related accuracy, in form of Horizontal DOP (HDOP).

The camera added is exploited running the file `rs_camera.launch`. It is a customised launch file that has been improved starting from the camera own driver. As it does not provide the measurements directly, the VO package RTAB-Map is run via a customised `rtabmap.launch` file, obtained initially from the RTAB-Map package and then tuned for the scope of this thesis.

D.5.4 Launching the localisation feature

The localisation improvements are performed by the script in `AEKF.py`. It performs the AEKF elaborated in 3.3.

To perform in an asynchronous way, a binary lock is adopted to account when a measurement update is performed at the same time of a measurement receive.

D.5.5 Launching the mapping update

The mapping feature is provided by the implementation available in `Mapping.py`.

The initial phase is of setting the boundaries.

Then a specific message changes the

The collision events are then exploited to update the knowledge of the surroundings.

For DIVA


```
{
"Author1": {
            "Last name": "Boffo",
            "First name": "Marco",
            "Local User Id": "u100001",
            "E-mail": "boffo@kth.se",
            "ORCiD": "0000-0003-4282-0472",
            "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                             }
          },
"Degree": {"Educational program": "Master's Programme, ICT Innovation, 120 credits"},
"Title": {
            "Main title": "Localisation and Mapping for an Autonomous Lawn Mower",
            "Subtitle": "Implementation and analysis of multiple configurations to provide localisation and mapping features for an
autonomous lawn mower using measurements from heterogenous sensors",
            "Language": "eng" },
"Alternative title": {
            "Main title": "Lokalisering och kartläggning för en autonom gräsklippare",
            "Subtitle": "Implementering och analys av flera konfigurationer för att ge lokalisering och kartläggning för en autonom
gräsklippare genom att smälta mätningar från heterogena sensorer",
            "Language": "swe"
},
"Supervisor1": {
                "Last name": "Carlsson",
                "First name": "Håkan",
                "Local User Id": "u100003",
                "E-mail": "hakcar@kth.se",
                "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                 "L2": "Intelligent Systems" }
              },
"Supervisor2": {
                "Last name": "Voigt",
                "First name": "Thiemo",
                "E-mail": "thiemo.voigt@ri.se",
                "Other organisation": "RISE Research Institutes of Sweden AB"}
              },
"Supervisor3": {
                "Last name": "Eriksson",
                "First name": "Joakim",
                "E-mail": "joakim.eriksson@ri.se",
                "Other organisation": "RISE Research Institutes of Sweden AB"}
              },
"Examiner1": {
                "Last name": "Li",
                "First name": "Haibo",
                "Local User Id": "u100004",
                "E-mail": "haiboli@kth.se",
                "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                 "L2": "Human Centered Technology" }
              },
"Cooperation": { "Partner_name": "RISE Research Institutes of Sweden AB"},
"Other information": {
"Year": "2021", "Number of pages": "??,76"}
}
```