千葉工業大学

CHIBA INSTITUTE OF TECHNOLOGY

UAV Indoor Localization in Real Time with a Multi Mobile Robot Formation

複数移動ロボットシステムを用いる屋内 UAV ロボットの実時間自己位置推定に関する研究

A DISSERTATION

Author:
RUIZ BRITO Luis Arturo

Advisor:
WANG Zhidong

Submitted in fulfillment of the requirements for the degree of PhD in Engineering in the

Department of Engineering

March, 2017

CHIBA INSTITUTE OF TECHNOLOGY

Abstract

Faculty of Engineering

Department of Engineering

PhD in Engineering

UAV Indoor Localization in Real Time with a Multi Mobile Robot Formation

by RUIZ BRITO Luis Arturo

For an Unmanned Aerial Vehicle (UAV) robot, to be correctly located in indoor environments is of great importance. By successfully reaching a commanded position in a given place, the robot can efficiently carry on assigned tasks. To aid on the subject of localization, this works presents an UAV three dimensional localization method, which is performed by means of trilateration by a multi robot formation of Omnidirectional Wheeled Mobile Robots (OWMR). The formation of ground robots has the ability to move freely anywhere reachable in the environment. As the formation explores the environment, it has an uncertainty from measured sensors which propagates and affects the predicted UAV position. The OWMR uncertainties translate into errors in positions which are handled using estimation algorithms to keep the UAV location accurate. An extended Kalman filter is implemented as well as a particle filter and their results compared. While the former approach has good tracking properties the later has good response to the nonlinearities in the system. Localization is not attainable everywhere on a map and areas where it can be hard to correctly perform the localization are described in a solvability map. This map has properties that can be employed to select the robot formation shape and into the design of favorable paths for the UAV motion. The solvability map is calculated by means of the derived probability distribution of the measured uncertainties of the trilateration algorithm. To accomplish these objectives, control schemes are correctly implemented and verified in the OWMR formation to carry on individual and combined tasks. All the concepts are taken from theory, starting from researches in the field, and they are supported with numerical simulations and real time experiments in a networked multi robot environment.

Keywords: indoor localization, localization, multi robot, omnidirectional, OWMR, real time, solvability, UAV

千葉工業大学

要旨

工学学科
工学専攻
工学博士課程

複数移動ロボットシステムを用いる屋内 UAV
ロボットの実時間自己位置推定に関する研究

ルイスブリト　ルイスアルトゥーロ

屋内環境で移動する無人飛行ロボット（UAV）にとっては、正確な自己位置を推定することは極めて重要であり、指定された目標位置へ正確な移動は、UAV ロボットの効率的に各種作業の遂行に大きく寄与する。GPS による位置推定が利用できない屋内環境においては、測位センサやカメラによる周囲環境に基づく自己位置推定手法には、センサ自身及び処理に多くの電力を消費し、UAV ロボットの長期間な作業に影響している。本研究では、屋内環境で活動する複数の移動ロボットのフォーメーションによる UAV ロボットの実時間の三次元自己位置推定の手法を提案し、移動ロボットの位置及び UAV ロボットの計測の不確実性による UAV 位置推定の誤差伝搬及び自己位置推定の可解性問題を研究し、自己位置推定の性質を解明し、UAV との協調運動の制御戦略を提案し、数値シミュレーション及び実機を用いて検証を行った。提案した UAV の三次元自己位置推定手法は、複数移動ロボットが構築したフォーメーションから三辺測量手法に基づくものであり、移動ロボットに搭載したカメラから測定した UAV ロボットとの距離と SLAM アルゴリズムから推定した移動ロボットの位置の二種類の情報で実装できる。従って提案する自己位置推定法は、移動ロボットの種類や構造に依存しなく、異なる座標系で制御されるロボットに実装でき、不整地で作業するロボットフォーメーションからも適用できる優位性がある。本論文では、不確実性を有する移動ロボットのフォーメーションから UAV ロボットの自己位置推定システムの構築及びアルゴリズムを提案し、拡張カルマンフィルターとパーティクルフィルターの二つの実装を示し、数値シミュレーションと実機ロボットを通して有効性を示した。さらに、実環境で不確実性の要素からロボットシステムの運動制御への影響について、理論面から複数移動ロボットのフォーメーション誤差及び計測誤差から UAV ロボットの位置推定への伝搬特性の定式化と、自己位置推定の可解性マップ（Solvability Map）という新しいコンセプトの提案を行った。その上、確率方程式による三次元可解性マップの高速計算アルゴリズムを提案し、フォーメーションと三次元可解性との関係を分析すると共に、自己位置推定の可解性を維持する移動ロボットのフォーメーション制御戦略を構築し、実システムに実装を行い、実験を通して有効性を示した。以上のように、本論文では、複数移動ロボットのフォーメーションによる UAV ロボットの自己位置推定の新しい協調ロボットシステムを提案した上、不確実性を有するシステムにおいて理論的な要素を研究し、誤差伝搬および可解性マップに基づく協調運動制御の手法を確立した。これらの成果は、今後実システムへの応用において、有効なシステム構築の指針と協調運動制御の手段となる。

Para Esther y Luis Arturo

(To Esther and Luis Arturo)

Acknowledgements

This represents the consummation of one of the hardest goals I have achieved in my life. With it I learned about my own capabilities and that I have no limits to reach my goals when I propose myself to something. I learned about not only my academic field but also about how important are the things you do in your life. I give myself a standing ovation, for never give up.

I would like to thank my mother and father, Esther and Luis Arturo, for his infinitely love and support, for sharing with me all your experiences and teaching me about life. I know how much the culmination of this means to you both. To my brothers, Paulina, Claudia and Carlos for being there at all times and I just want you to know how much I love you. There has been not a moment I do not think about you all.

To her . . . ♥, my motivation, everything I do is because of you.

Thanks to all the people I have befriended up to this moment. Persons I have laugh with, shared experiences, learned and grown. Friends who helped me to settled down in moments I lost my way. I enjoy so much to spend time with all of you and this would not have been possible without all the people involved in my life. I just want to meet you all again and again and have you in my life.

My everlasting gratitude to the people of Japan for their support on this work through the Monbukagakusho scholarships program. I have learned so much about this beautiful country and its culture through all the people I have met, my lab mates, my Japanese instructors, the CIT professors and staff and even that little girl that once called me "abunai hito" on the train station. The Japanese perception of life have changed me forever. It was always one of my life goals to visit this awesome place and I ended up having one of the most rewarding experiences of my life.

And finally, thanks to professor Zhidong Wang, my advisor. I have shared so many good and bad moments with you while working at this institution. Thank you for all the support during these years and for giving me this opportunity.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of abbreviations

CCW	Counter Clock Wise
CDF	Cumulative Density Function
CW	Clock Wise
DOP	Dilution of Precision
EKF	Extended Kalman Filter
GPS	Global Positioning System
IKF	Iterated Kalman Filter
IMU	Inertial Measurement Unit
LRF	Laser Ranger Finder
MRL	Multi Robot Localization
OWMR	Omnidirectional Wheeled Mobile Robot
PDF	Probability Density Function
PF	Particle Filter
r.v.	Random Variable
RANSAC	Random Sample Consensus
SEP	Spherical Error Probable
SLAM	Simultaneous Localization And Mapping
SM	Solvability Map
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

List of Symbols

c_m	Encoder conversion factor	$\mathrm{m/pulses/rev}$
f_{abc}	Trilateration function	m
g^p	OWMR dynamics	m
k_3	Height factor	$\mathrm{unitless}$
l_{iu}	Distance measurements	m
$\mathbf{p}_C$	Center of formation	m
$\mathbf{p}_i$	OWMR positions	m
$\mathbf{p}_i^n$	OWMR positions by encoders	m
$\mathbf{p}_i^r$	OWMR positions by LRF	m
$\mathbf{p}_u$	UAV position	m
$\mathbf{R}$	UAV measurement covariance	$1/\mathrm{m}^2$
$\mathbf{Q}^n$	OWMR encoder measurement covariance	$1/\mathrm{m}^2$
$\mathbf{Q}^p$	OWMR measurement covariance	$1/\mathrm{m}^2$
$\mathbf{Q}^r$	OWMR LRF measurement covariance	$1/\mathrm{m}^2$
$\mathbf{R}$	UAV process covariance	$1/\mathrm{m}^2$
$\mathbf{R}^n$	OWMR process covariance	$1/\mathrm{m}^2$
SM	Solvability map	$\%$
$\mathbf{u}_C$	Formation input speed	$\mathrm{m\,s^{-1}}$
$\mathbf{u}_i$	OWMR input speed	$\mathrm{m\,s^{-1}}$
$\mathbf{u}_t$	UAV input speed	$\mathrm{m\,s^{-1}}$
$\mathbf{v}_{n\psi}$	OWMR local velocity	$\mathrm{m\,s^{-1}}$
$\mathbf{v}_{xy\psi}$	OWMR global velocity	$\mathrm{m\,s^{-1}}$
$\mathbf{v}_{123}$	OWMR wheel tangential velocities	$\mathrm{m\,s^{-1}}$
$\mathcal{X}$	PF estimator particles	m
$\mathbf{z}_i^p$	OWMR position measurement signal	m
$\mathbf{z}_t$	UAV position measurement signal	m
$\boldsymbol{\omega}_{enc123}$	OWMR wheel encoder speed	$\mathrm{pulses/s}$
Σ	EKF estimator covariance	$1/\mathrm{m}^2$
ψ_C	Formation angle	rad
π	π	$3.141\,592$

1 Introduction

Robotics is a field that has seen huge improvements in the last years. While at the beginning robotics focused on how a robot could perform in the environment, dynamics were one of the main researched areas in the field. How to land theory into real systems was a real issue. Nowadays, thanks to advancements in technology, robotics is moving more and more to help machines understand the environment and interact with it.

One of the goals in the field for a robot is to be truly autonomous: to perform activities by its own without human intervention. In disaster zones or difficult access areas there are many scenarios where a robot is needed for inspection, taking samples, make a survey of the area and/or feedback all the information it can gather to evaluate further actions.

Anywhere a robot, specifically an Unmanned Aerial Vehicle (UAV), has to perform an activity, it needs to know where it is, its position and the positions of objects around it. With that information, this robot can successfully carry on given tasks. Due to interpretations like this one, an UAV localization within a given environment is one of the main topics of research nowadays. The need to know were objects are located in indoor environments is of interest because it allows to analyze areas, minimizing risks to humans and/or damages to equipment.

In outdoor environments localization has the advantage of access to the Global Positioning System (GPS) technology and the tasks a robot needs to perform in an open environment usually do not need to be extremely accurate. For indoor localization, network arrangements have proven to work to mimic GPS technology; fixed stations are placed everywhere required inside a building and they transmit a signal to a receiver which has the ability to compute its position based on the signals it receives.

Following the network approach, stations can be fixed cameras, used to build motion capture (Mo-cap) systems. Systems as VICON and Optitrack have proved to have an exceptional accuracy. On the down side the systems have to be setup before using them, so it is not optimal and in no way convenient in disaster situations.

1.1 About this project

The idea of this research comes from taking a proven static localization system for indoor environments to a dynamic level. Making an analogy to GPS technology, the stations as well as the receiver can move freely in a indoor scenario. This also comes in hand with the interactions between multiple robots. It has been recognized that some times a single robot might not be able to handle all the requirements for a given tasks. It may be too small or too big, too slow or too fast, too heavy or too light. Thus, it has been proposed to use multi robot approaches to carry on complex duties. The advantages of one type of robot can potentially minimize the deficiencies of other types of robots. The system overview is depicted in Fig. 1.1.

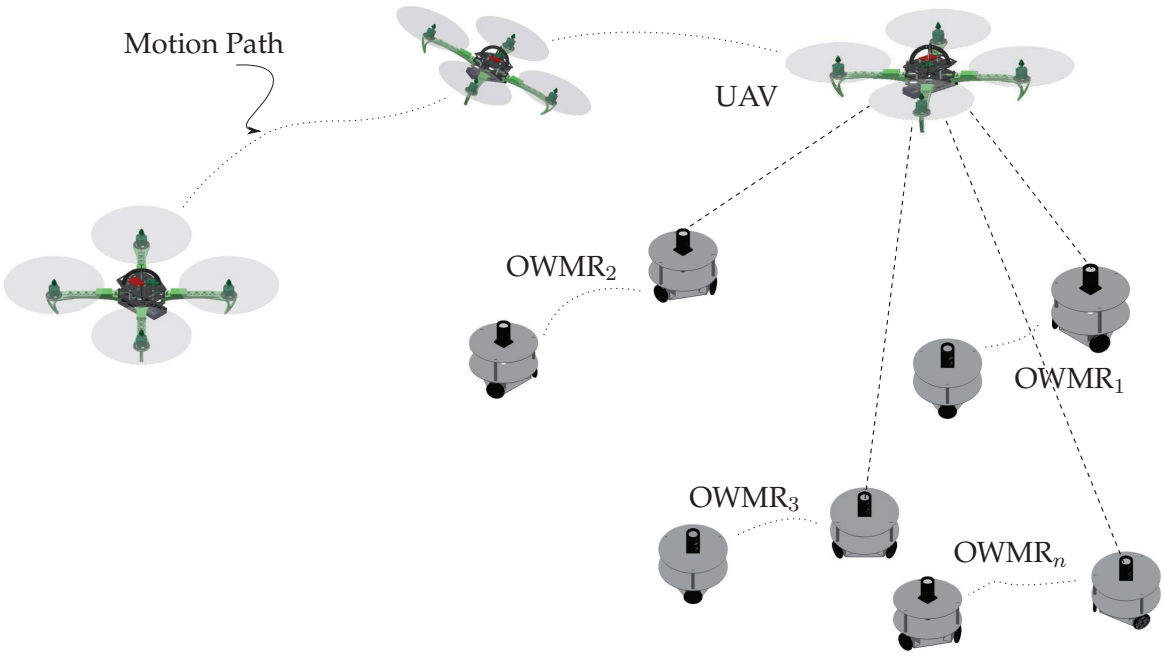


FIGURE 1.1: System overview. The proposed multi robot approach intended to be used for this project. A formation of OWMR will localize an UAV in three dimensions in a dynamic environment.

For this project, the mentioned moving stations are a group of Omnidirectional Wheeled Mobile Robot (OWMR) in a multi robot formation, and the receiver is an UAV. An UAV needs to trade the amount of weight it can carry between sensors needed for its own performance and the ones used for the tasks it was given. Thus, to aid in the performance of the UAV, it is proposed to use a multi robot approach to calculate the UAV position.

In this multi robot architecture, the UAV only requires to handle the necessary sensors for the task assigned taking in account its wide range and dynamic 3D motion, using as much flying time as possible. Wheeled multi robot formations can be easily equipped with high resolution cameras and powerful on board image processing computers (making power consumption not so critical) to perform both Simultaneous Localization And Mapping (SLAM) and UAV localization tasks.

The localization is executed as a trilateration in three dimensions. Trilateration is used by GPS satellite systems to localize targets on the surface of the Earth but have limitations in indoor scenarios as the satellites are occluded by the environment. Now, the same approach can be taken for indoor scenarios if instead of satellites we talk about a multi robot formation. In trilateration, the unknown is called the receiver, the UAV in this project, and the measurements came from stations or receivers, which in this case will be the robot formation of OWMR.

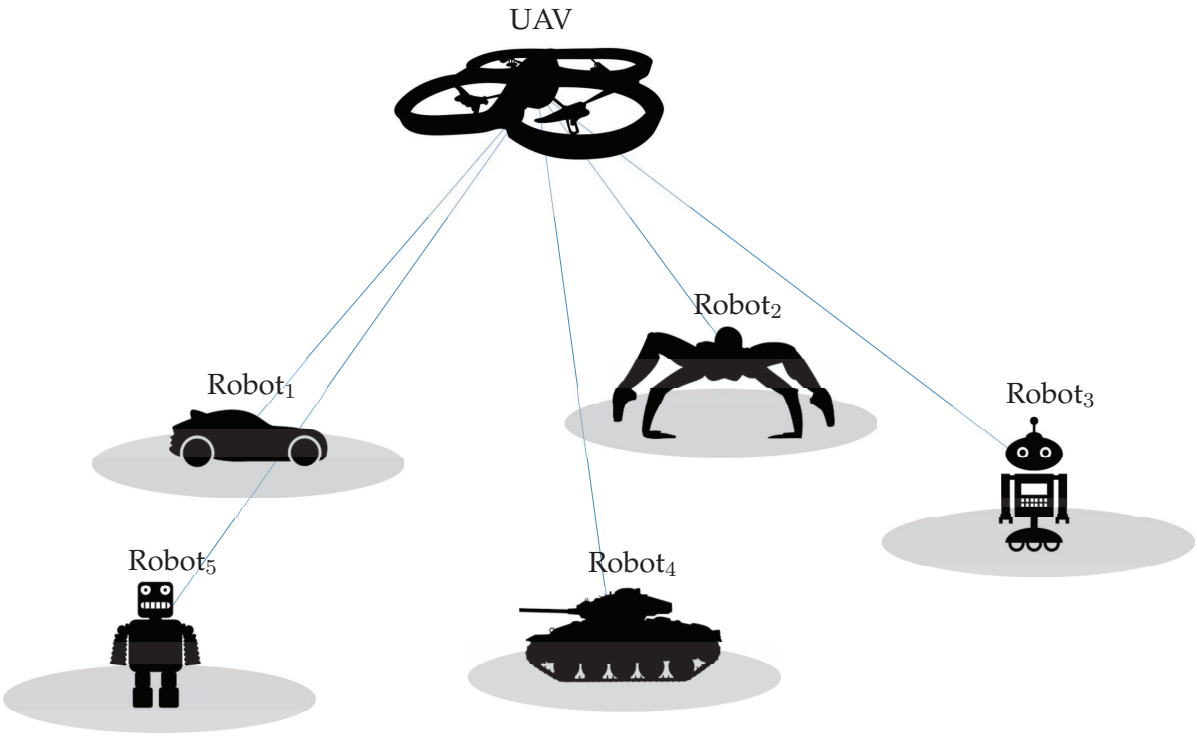


FIGURE 1.2: Different types of robots for localization. The proposed multi robot localization is intended to be used with any kind of robots as long as they can calculate their positions and the range measurements required to trilaterate.

Notice that as long as they can calculate their own positions and the distance they have to the UAV, the robots of this formation can have any shape, and any dynamics can define their motion. At this stage we are focusing more on the algorithms for the localization and formation control, so using OWMR is more convenient. The motion of the robot formation is independent of the drone motion and can be of great aid to the UAV performance. The trilateration based method do not necessary constraints the mobile robots to a flat surface when the formation of mobile robots is known. This allows the method to be applied for disaster investigation tasks on even and uneven terrains, see Fig. 1.2.

1.2 Thesis organization

In this thesis the localization in three dimensions is performed, and to achieve it, this thesis is organized as follows:

- Chapter 1 , Introduction.
- Chapter 2 , Background and Related Research.
- Chapter 3 , Localization by Trilateration.
- Chapter 4 , Sensed Measurements for Localization.
- Chapter 5 , UAV Localization with a Multi Robot Formation.
- Chapter 6 , Uncertainty Propagation and Accuracy Analysis.
- Chapter 7 , Solvability due to Uncertainty in Measurements.
- Chapter 8 , Formation Selection and Control.
- Chapter 9 , Numerical Simulations.
- Chapter 10 , Real Time Localization Experiments.
- Chapter 11 , Conclusions.
- Appendix A , Caley-Menger Determinants.
- Appendix B , Jacobians of g^p.
- Appendix C , Derivation of Partials of H_{zt}.
- Appendix D , Derived Probability Distributions.

In Chapter 1 the purpose of this work is explained and in Chapter 2 the background information is provided. Then, in Chapter 3 the trilateration is explained and Chapter 4 has the methodology to obtain the measurements required for this task. Chapter 5 defines how the multi robot localization is performed. Chapters 6 and 7 show some of the methods used to analyze formations with the localization method presented in this work. Chapter 8 presents some formation control algorithms used in this project. Finally, Chapters 9 and 10 show some numerical and real time experiments to prove the algorithms proposed. The conclusions are presented in Chapter 11 and in Appendices A to D some of the mathematical tools employed are explained.

1.3 Remarks

During the realization of the Ph.D. studies, 2 journal papers and 2 conference papers were published. These publications backup the interest of the scientific community in the research carried on, as well as the methods employed to solve the problem. The publications written are listed below:

- [1] S. Magariyama, H. Matsumoto, L. Ruiz, *et al.*, "Estimation of a feasible dynamic caging zone for multirobot object transportation", in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, 2014, pp. 508–514.
- [2] L. Ruiz and Z. Wang, "System design for the localization of an uav in 3d using a mobile multi-robot platform", *JOURNAL OF JAPAN SOCIETY FOR DESIGN ENGINEERING*, vol. 52, no. 2, pp. 81–97, 2017.
- [3] L. Ruiz and Z. Wang, "Error propagation and solvability in uav localization based on trilateration with a multi robot formation", *Int. J. of Mechatronics and Automation*, Accepted, In Print, 2017.
- [4] L. Ruiz and Z. Wang, "Real time multi robot 3d localization system using trilateration", in *IEEE International Conference on Robotics and Biomimetics, 2016, International Conference on*, 2016, pp. 1510–1515.

Publication [1] was used as an introduction to multi robot environments and the research methodology. In [2] the concept of this investigation was proposed. In [2]–[4], the general concepts for the trilateration, localization and measurements mentioned in the publications is explained (Chapters 3 to 5). The propagation of the error from publications [3] and [4] is discussed in Chapter 6 as well as how the uncertainties in the problem were envision. The solvability map concept was introduced in this thesis, Chapter 7 is divided in the Montecarlo method from publication [2] and the derived probability closed form formula from publication [3]. The formation control algorithms in Chapters 8 are taken from publications [2] and [3]. Finally, [4] shows the real time implementation, defining how the project was taken from theory to a real time system using robots and systems developed in the laboratory.

2 Background and Related Research

2.1 Localization with stations and receivers

There are many methods to compute the location of an object having a set of stations and receivers, being one of the most studied triangulation and trilateration[5], [6]. In both methods the stations can be robots which know their positions in the environment, and the receiver, is a robot with an unknown position. As depicted in Fig. 2.1 for two dimensions, $\mathbf{p}_1$ and $\mathbf{p}_2$ are the stations and $\mathbf{p}_u$ is the receiver. Triangulation can determine the pose of an object, its coordinates and orientation based on angle measurements, θ_1 and θ_2. Trilateration can calculate the position based on distance measurements l_1 and l_2. When using trilateration the receiver's orientation cannot be computed, but, this can be mitigated using additional measurements from another sensor, e.g. an Inertial Measurement Unit (IMU). The problem as shown in Fig. 2.1 can be solved using trigonometric identities with an increased difficulty in three dimensions.

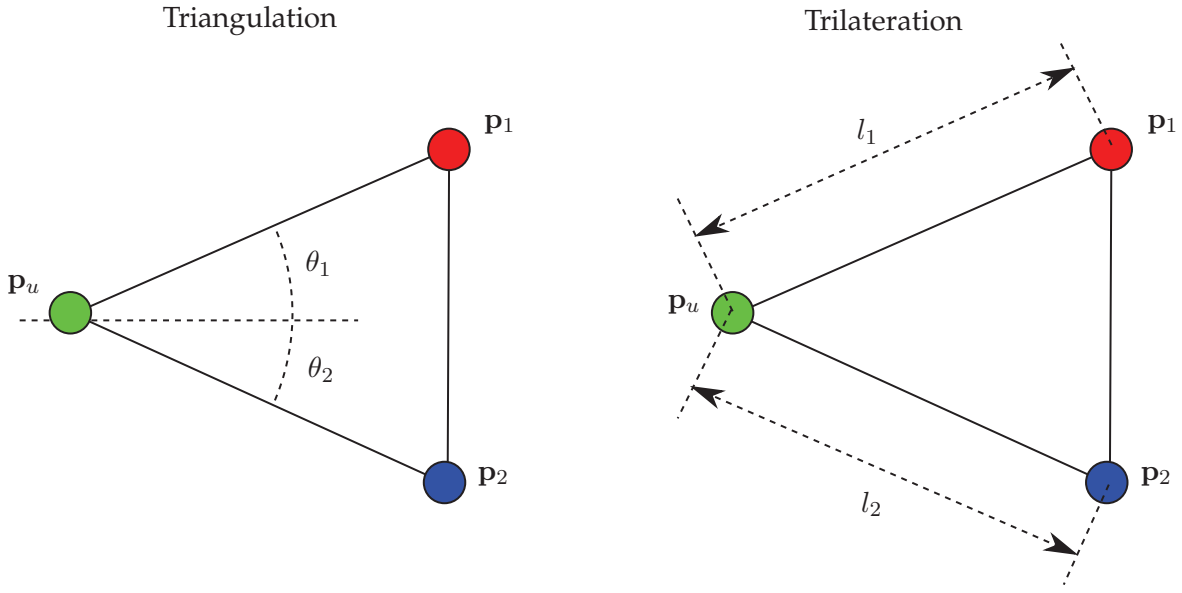


FIGURE 2.1: Triangulation and trilateration. Triangulation and trilateration differ in the way the measurements are taken, and thus, how the problem is solved. $\mathbf{p}_1$ and $\mathbf{p}_2$ have known coordinates and $\mathbf{p}_u$ is the unknown.

Triangulation is used widely in vision localization researches as the three point resection problem or the PnP problem[6]–[8]. Here two images of the same object from different perspectives can be used to retrieve the object position, see Fig. 2.2.

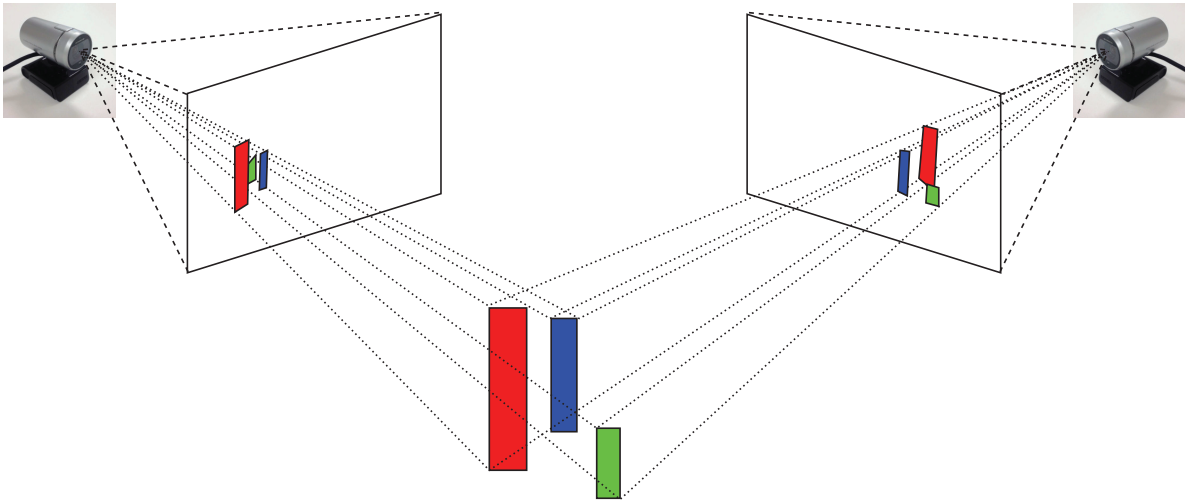


FIGURE 2.2: PnP problem. Triangulation is used to calculate the pose of an object if two or more images exist of that object. A set of features detected in more than one image is used to retrieve the pose.

As any object can be defined by several features in a given image, this method need to iterate to find the best pose for a given set of points, normally using Random Sample Consensus (RANSAC) approaches[9]–[12]. This method strongly depends on coordinates, specially their relationship, and the orientation errors will greatly affect the accuracy of the localization. As a vision system, in general, some preprocessing calibration helps to limit this error but as a multi mobile robots, this will be not easy and feasible as what fixed multi-camera system can do.

Localization by trilateration is used extensively by GPS devices to position targets on the surface of the Earth. In [13] it is mentioned that GPS has the disadvantages costly hardware and power requirements as well as the need for a free line of sight to GPS satellites. The same GPS principle can be applied at a smaller scale for robotic applications. If one thinks about a multi robot scheme, each robot can be considered to do the functions of a GPS satellite while other robots can work as the receivers. Trilateration is chosen as the method for this research as we want a system that is a simple as possible while still having the robustness for a good localization approach, since only the distance is required, the method to get it determines the sources of noise in the measurements[5]. In this research the method chosen is an AR Tag vision system. Finally, another advantage over triangulation is that when measuring angles a small error in orientation translates into big errors in position, specially in noisy environments.

In [14] three methods to perform trilateration are discussed: quadratic equation, Caley-Menger determinant and non linear least squares. Exact methods with close form equations can be found in [15] (Caley-Menger determinant) and in [16] (polynomial algorithm). Similar researches have been applied in this field in two dimensions using fixed stations. In [17] a group of submarines is used to trilaterate their mutual positions to navigate more accurately. In [13] it is introduced how a network of nodes can mutually compute their positions in a network localization scheme. The work of [14] has in mind fixed stations to create intelligent spaces for indoor scenarios. In Fig. 2.3 the network approach is depicted, several transmitters are placed inside a building, the information provided by the transmitters can be read by an external device capable of measure the distance it has to each station.

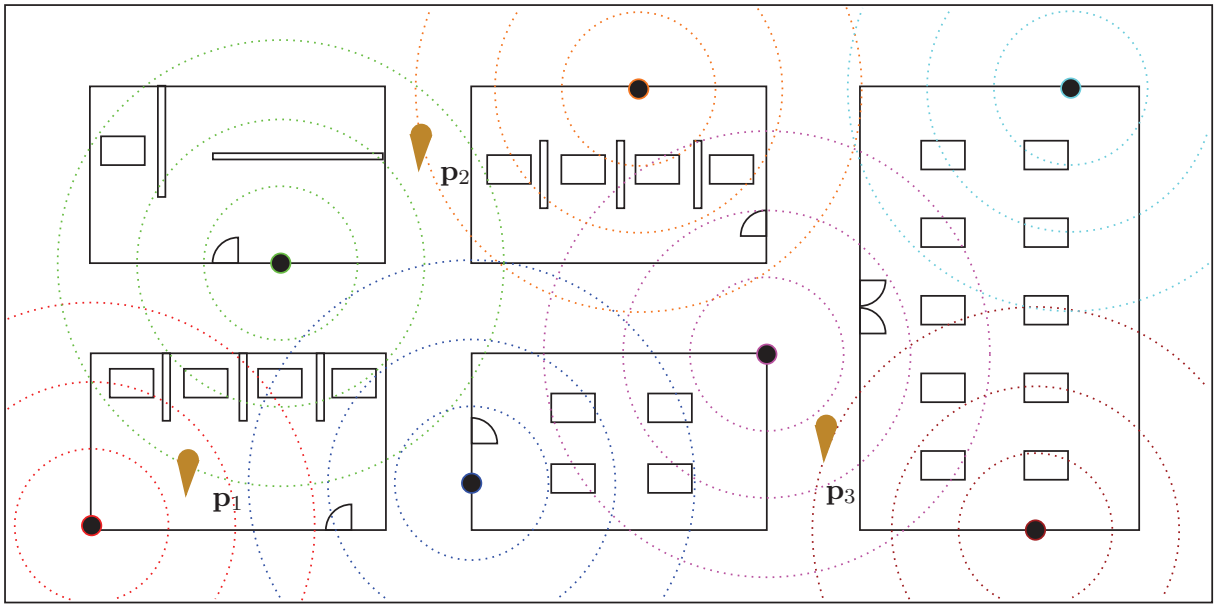


FIGURE 2.3: Network localization for indoor environments. A network array of station can be deployed in indoor environments to localize objects within the range of the stations measurements.

The methods described are good providing an horizontal position, for height measurements they must be fusion with data from additional sensors, e.g. barometric sensors[16], if the receiver is to close to the base plane or far away from the formation. The method chosen for this research is the Caley-Menger determinant, it has a straightforward error analysis and a close form solution which can be easily implemented in real time applications, also, it does not require linearization, so the execution is quick in real time systems. This thesis presents a work which has the novelty of execution for trilateration in indoor scenarios in a three dimensional space with moving stations taking in account the uncertainties in the system.

2.2 Errors in localization

The positions of stations with respect of the receiver cause an error in trilateration, which can be indexed as a Dilution of Precision (DOP)[18], an effect of the apparent increase of the error. DOP researches have been done previously[19], [20] to address DOP. In [19] a triangulation method is used and the propagation of errors is introduced as a robot changes its position with respect of the stations. In [21] it is identified that noisy measurements degrade the trilateration in four aspects:

1. Uncertainty or no intersection.
2. Ambiguity or mirror solution.
3. Error propagation.
4. Non consistency between different sets of readings.

The item 1 is handled in this research introducing the concept of Solvability Map (SM) as a measure of the trilateration output, describing a region where the trilateration can be computed, Chapter 7. Item 2 is handled in the trilateration algorithm by defining the formation direction, it will be explained in Chapter 3. The aspect 3 will be derived in Chapter 6. Item 4 is handled by redundancy between different sets of robots.

The intersection error has been addressed in [22] reducing no solvable areas by 50% adjusting the range measurements knowing the size of the error covariances, this methods needs to compute all the possible combinations of addition and subtraction of errors. Then, [23] deals with the flip ambiguity, setting safe conditions to perform trilateration. The work of [24] analyzes the complexity of trilateration with noise, stating that the problem is intractable. Having a network of stations, a localization algorithm is proposed to track different objects dealing with the non consistency in [25]. The way the error propagates from measurements in stations to the position of the receiver is mathematically defined with covariance matrices. In [26] it is explained how the covariances in an autonomous vehicle depends on the Jacobians with respect of the state position and the input vector. This is vastly mentioned in a probabilistic way in [27] and a calculation using multi step prediction is performed in [28].

Since the trilateration solves a localization problem, the uncertainty in positions and measurements has an error that increments with time. The effects of systematic errors by erroneous coordinates at the stations has also been explained taking in account that the errors in the stations propagates to the trilateration output[29], [30]. In Fig. 2.4 it is shown how the error propagates in different positions $\mathbf{p}_u$, when $\mathbf{p}_1$ and $\mathbf{p}_2$ have different covariances.

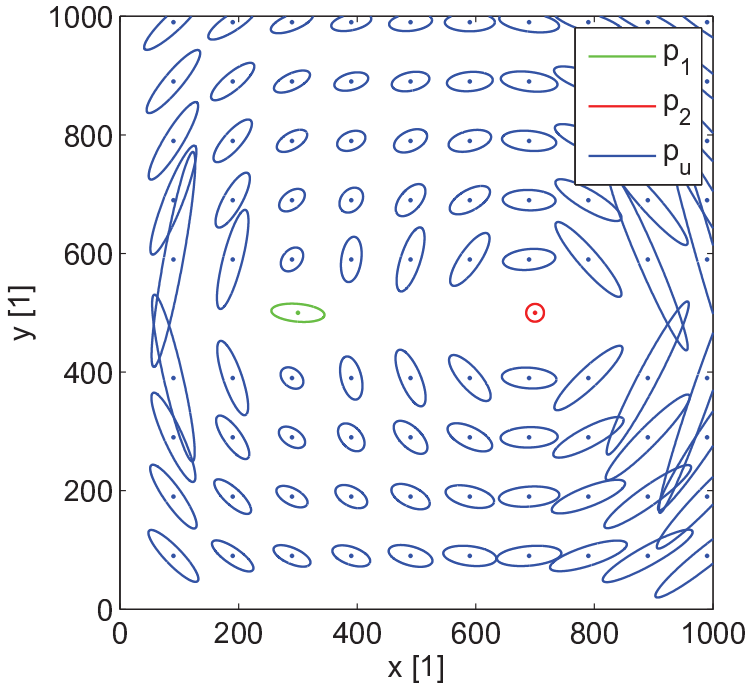


FIGURE 2.4: Error propagation in 2D. In this case the points $\mathbf{p}_1$ and $\mathbf{p}_2$ have covariances Σ_1 and Σ_2 which propagate to the unknown position $\mathbf{p}_u$. Σ_u is shown for several positions $\mathbf{p}_u$.

2.3 Multi robot localization

With the advent of new technologies robotics are taken on the field of perception to whole new levels. Vision algorithms[31], [32] have taken odometry and SLAM to a top notch recognition levels which are easily translated to the human understanding. SLAM techniques allow for a robot to build maps and at the same time the robot can know its position inside that map[33]. Laser Ranger Finder (LRF) sensors are used extensively for this application. Research has been done using UAV to localize several Unmanned Ground Vehicle (UGV)[34]. The UAV is used to calculate the position of the UGV by means of image processing using a camera looking down, and by doing so, it updates their positions, reducing the UGV position errors (see Fig. 2.5). Similar approaches with air ground robot teams are shown in [35] and [36]. All these works are presented for outdoor environments.

For indoor environments it is presented in [37] a group of robots inside a building that can mutually calculate their positions using themselves as features. One of them moves while the others stay static and serve as reference, it stops and another robot starts the same process, see Fig. 2.6. This way the covariances are reduced improving the localization.

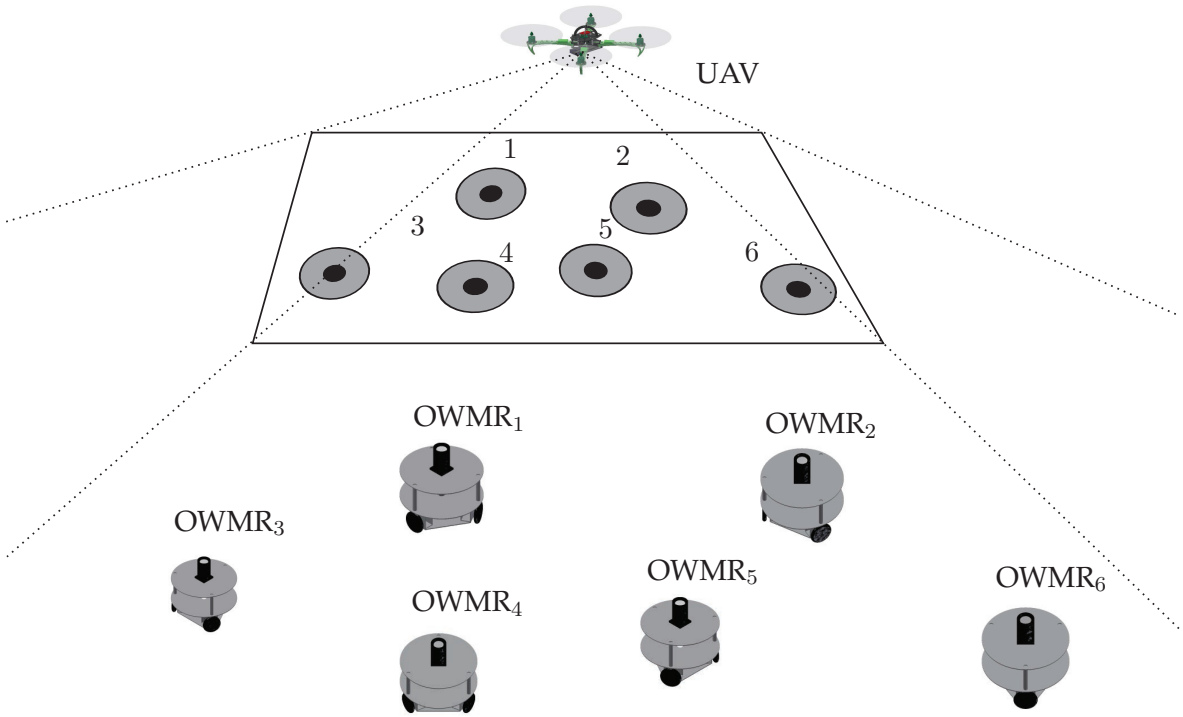


FIGURE 2.5: Aerial localization. An UAV has been used to localize the position of UGV using vision localization techniques, the UAV update the positions of the UGV.

Works like [38] and [39] show a group of UAV and UGV working cooperatively to achieve SLAM by grouping the individual outputs of each robot. In [40] a multi robot scheme is employed using manifolds where the planar trajectories of different robots are put together to build a map. Research is moving more and more towards real time applications, and with it, it is needed to use as many sources of information as possible to define the position of a robot.

As the localization by trilateration depends on the stations positions and distance measurements, the output may not be accurate and an estimator technique can be used to improve the localization. Researches have shown how to minimize the effect the errors using an Extended Kalman Filter (EKF)[41] showing the dynamics involved in the problem. Whatever method is used for estimation the problem statement is the same: calculate a prediction of the possible state of the process at a time t based on a given input $\mathbf{u}_t$, knowing the previous state $\mathbf{x}_{t-1}$, then, update that prediction with measurements $\mathbf{z}_t$ to get a estimated state $\mathbf{x}_t$[42]. If the nonlinearities highly influence the output, methods like the Iterated Kalman Filter (IKF)[43] or the Particle Filter (PF)[44] can also be implemented as estimators. These methods are computationally more costly but they are a better option dealing with the nonlinearities in the system.

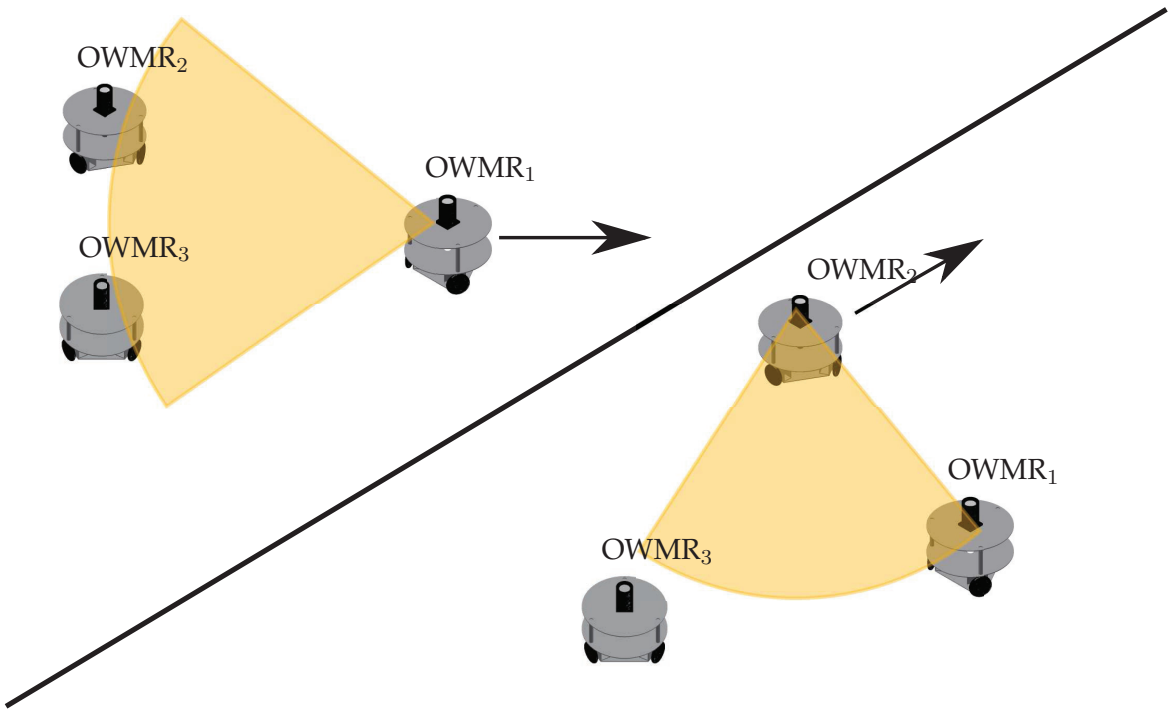


FIGURE 2.6: Mutual localization. In a mutual localization approach one robot moves while it uses other robots as features to localize itself. Then, another robot can execute the same process. In this way the robots mutually express their positions and reduce their localization errors.

Lastly, for this project is required that the interaction between the robots is not strongly centralized. Although the OWMR send the sensed information back to a central computer they can act on its own, and no robot is in charge of the formation at a very strict level. Rather, the formation will be defined based on the circumstances of the problem. This approach has been studied in [45].

2.4 Omnidirectional robots

A position for a mobile robot can be expressed by absolute sensors, and dead reckoning sensors. The former take information from the environment to localize an object and the later take measurements from “inside" the robot[46], [47]. Knowing this, one of the most powerful properties of estimators is that they can fusion data from different measurements to have a better estimate[48]–[51].

Estimators are vastly used in localization where dead reckoning positioning like odometry, is only good for small distances but have to be updated with absolute sensor readings as SLAM techniques or GPS devices after a long distance has been traveled. The dynamics of OWMR are well understood and good controllers based on odometry readings have been applied[52]–[54]. The OWMR can move in any direction without changing its orientation, their practicality in real time scenarios depends on how they are build. If the construction is taking in account rough terrains or not will determine the kind of environments these kind of robots can be employed.

The main source for localization in an OWMR is its odometry measurement from encoders, this measurement has errors intrinsic to the sensors attached to the wheels as well as systematic and non systematic errors. Systematic errors accumulate constantly and are due to the parameter uncertainties in the dynamics from the sensor readings to the vehicle displacement. Non systematic errors are unrelated to the robot, like slippage and uneven terrain[55], [56]. The error in odometry has been defined for vehicles showing the increase over time of the position error[57], [58]. Examples of calibration for OWMR are proven to improve the localization of this kind of robots and have been studied before[59], [60] and the slippage in the wheels has also been addressed in previous researches[61]–[63]. Non systematic errors are better handled using SLAM techniques as an external feature is used as reference to correct the position.

2.5 Conclusions

This thesis is intended to aid in the localization of objects in indoor environments. To put the concepts explained in this work to practice requires the understanding of the whole system integration and the interaction of its different parts. Concepts like SM are defined for the first time in this thesis and are intended to give more insight on how to handle properties of trilateration for multi robot formations. Also, this project can serve as an explanatory guide for the mathematical tools explained in its chapters.

3 Localization by Trilateration

Trilateration is used extensively in GPS applications to find the position of object on the surface of the Earth. If a GPS receiver has clear view to four GPS satellites it can calculate its position.

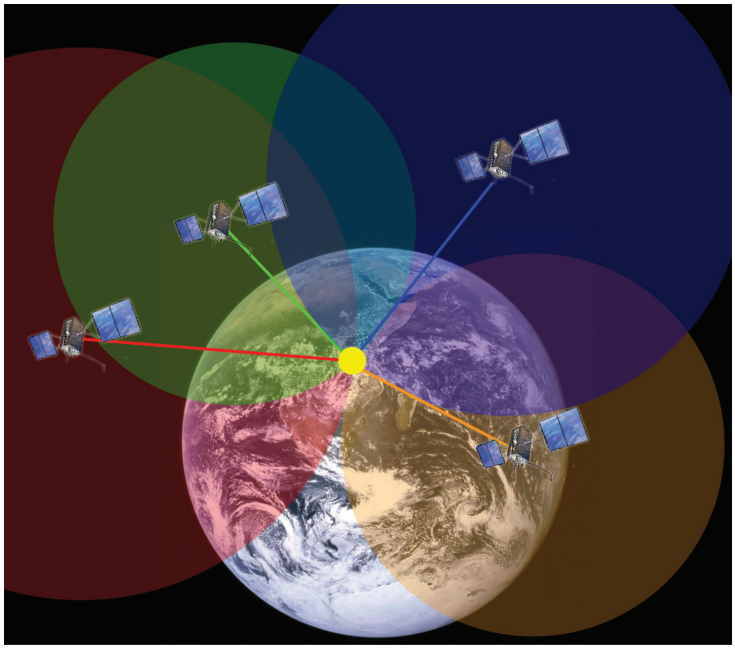


FIGURE 3.1: Earth's GPS application. The GPS system needs the measurements from 4 satellites to calculate the position of an object on the surface of the Earth.

Mathematically speaking, trilateration consist, in a very simple scheme, in finding where a group of n spheres intersect. These spheres have center coordinates denoted by $\mathbf{p}_i$ with a radius l_{iu}. The intersection point will be denoted by $\mathbf{p}_u$. Thus, to find the coordinates of $\mathbf{p}_u$, the solution of the system of equations (3.1) is required.

$$\|\mathbf{p}_u - \mathbf{p}_i\| = l_{iu}, \qquad i \in 1, \ldots, n \tag{3.1}$$

Locations at $\mathbf{p}_i$ are denoted stations while $\mathbf{p}_u$ is called the receiver. In this work, $\mathbf{p}_i$ are OWMR which are able to maintain their locations, and $\mathbf{p}_u$ is the UAV robot which position is localized by the proposed system. Graphically, a trilateration graph can be drawn[24], Fig. 3.2.

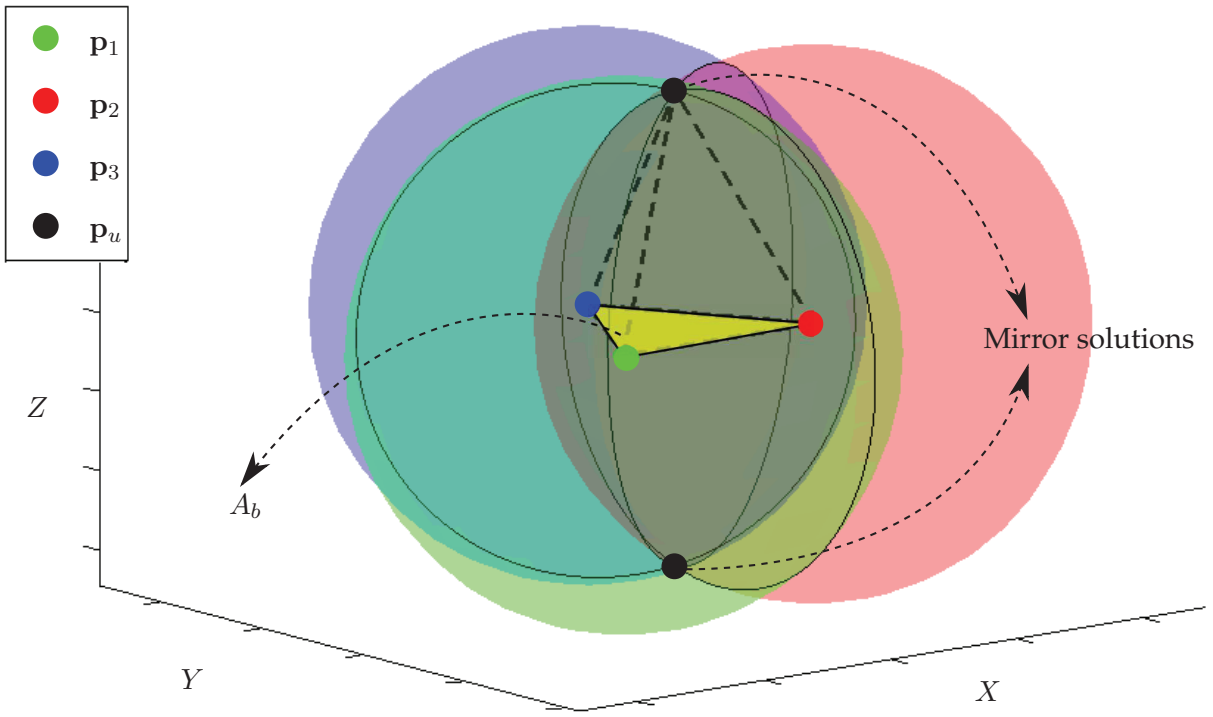


FIGURE 3.2: Trilateration graph. A non unique solution is denoted by the intersection of three spheres as two possible solutions for $\mathbf{p}_u$. The mirror solution can be discarded by knowing if $\mathbf{p}_u$ is above or below to the base plane A_b.

At least four spheres ($n = 4$) are needed for a unique solution of $\mathbf{p}_u$. The intersection of two spheres is a circle, also shown in Fig. 3.2, and by adding a third sphere only two points are left as the possible locations for $\mathbf{p}_u$.

These two solutions are mirrored by the plane A_b. A_b is denoted as the base plane, formed by the locations of the stations $\mathbf{p}_i$. Now, if the receiver at $\mathbf{p}_u$ is known to lie above or below A_b, then, the true solution can be rightly selected.

3.1 Solution to trilateration

The Caley-Menger determinant[15] is used in this thesis as the chosen method to solve the trilateration problem, it has a closed form solution, it has a straight forward implementation. Also Caley-Menger determinants have geometric properties that allow for a better understanding of the mathematical concepts behind the equations.

Using Caley-Menger determinants, for three points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, $\mathbf{p}_u$ is given by:

$$\mathbf{p}_u = \mathbf{p}_{ub} \pm k_3(\mathbf{v}_{12} \times \mathbf{v}_{13}) \tag{3.2}$$

$\mathbf{p}_{ub}$ is the projection of $\mathbf{p}_u$ on the base plane A_b and k_3 is the height factor, k_3 will be explained in Section 3.2. $\mathbf{v}_{12}$ and $\mathbf{v}_{13}$ are vectors defined as:

$$\mathbf{v}_{12} = \mathbf{p}_2 - \mathbf{p}_1 \tag{3.3}$$

$$\mathbf{v}_{13} = \mathbf{p}_3 - \mathbf{p}_1 \tag{3.4}$$

$\mathbf{p}_{ub}$, is calculated as:

$$\mathbf{p}_{ub} = \mathbf{p}_1 + k_1\mathbf{v}_{12} + k_2\mathbf{v}_{13} \tag{3.5}$$

The constants k_1, k_2 and k_3 are calculated by means of the Caley-Menger[1] determinants ($D(\cdot)$) as:

$$k_1 = -\frac{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3; \mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_u)}{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)} \tag{3.6}$$

$$k_2 = \frac{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3; \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_u)}{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)} \tag{3.7}$$

$$k_3 = \frac{\sqrt{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_u)}}{D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)} \tag{3.8}$$

k_1, k_2 and k_3 set a scaling factor in the directions required to have the position $\mathbf{p}_u$ correctly located. The trilateration variables can be seen in Fig. 3.3.

3.2 True solution

The square root in (3.8) implies the mirror solution when using only three positions $\mathbf{p}_i$. In [15], the sign of k_3 is not dealt concisely in a mathematical way. On the other hands, k_3 is defined as the height divided by the norm of the vector normal to the base plane.

The term $k_3(\mathbf{v}_{12} \times \mathbf{v}_{13})$ in (3.2) is the height with direction, and that direction is given by the unitary normal vector $\mathbf{p}_n$ given by:

$$\mathbf{p}_n = \frac{\mathbf{v}_{12} \times \mathbf{v}_{13}}{\|\mathbf{v}_{12} \times \mathbf{v}_{13}\|} \tag{3.9}$$

[1] In Appendix A the calculation for Caley-Menger determinants $D(\cdot)$ is explained.

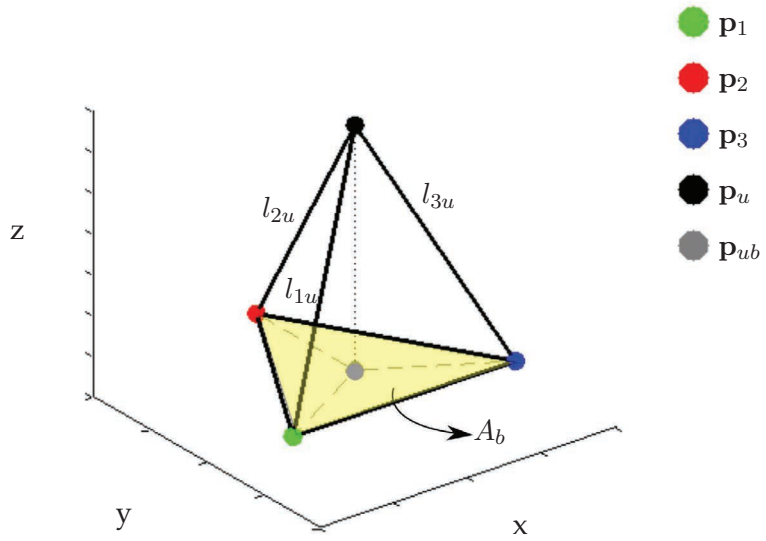


FIGURE 3.3: Trilateration variables. The measurements required for trilateration are shown. Positions $\mathbf{p}_i$ and distances l_{iu}.

If an observer is sitting on the UAV at $\mathbf{p}_u$, the OWMR formation at $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, can be seen to be arranged in a Clock Wise (CW) or Counter Clock Wise (CCW) fashion. For CCW formations, k_3 is positive ("+") because $\mathbf{p}_u$ is in the direction of $\mathbf{p}_n$, otherwise is negative ("−") because it is in the opposite direction to $\mathbf{p}_n$. In Fig. 3.4 the concept for the sign of k_3 is depicted.

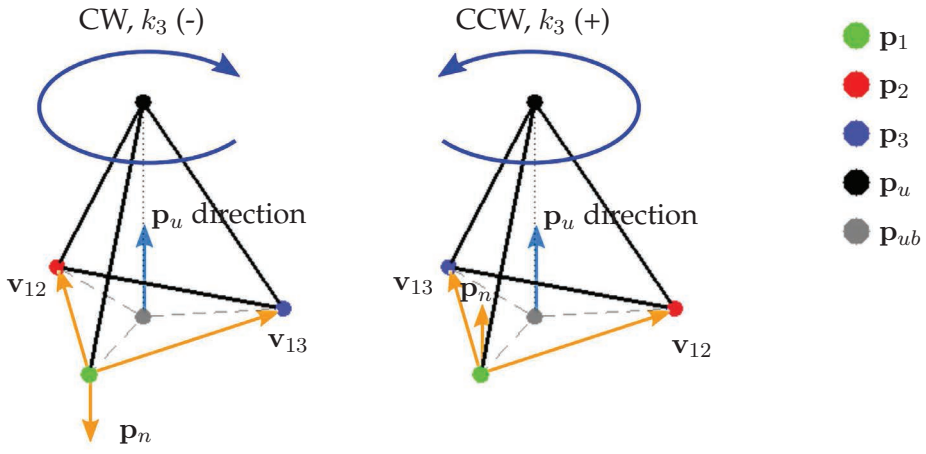


FIGURE 3.4: k_3 sign. The sign of k_3 changes due to $\mathbf{p}_n$ direction which in turns defines if the formation is in CW or CCW direction.

Any point on A_b can be used to create an auxiliary vector:

$$\mathbf{p}_{aux} = \mathbf{p}_i + \mathbf{p}_n = \begin{bmatrix} x_{aux} \\ y_{aux} \\ z_{aux} \end{bmatrix} \tag{3.10}$$

Then it can be tested if the formation is in CW or CCW rotation, Algorithm 1 depicts this process, where n_{err} is a threshold error.

Algorithm 1 Sign of k_3.

```
1: procedure GETK3SIGN(p_i,p_n)
2:     p_aux ← p_i + p_n
3:     if z_i − z_aux ≤ n_err then
4:         return 1, CCW
5:     else
6:         return −1, CW
7:     end if
8: end procedure
```

It is expected that the UAV is always above the OWMR formation. There are applications where that is not always true, e.g. a submarine. In that case the "if" condition changes to:

$$z_i - z_{aux} \geq n_{err} \tag{3.11}$$

3.3 No solution for $\mathbf{p}_u$

If the points $\mathbf{p}_i$ are collinear there is no solution for $\mathbf{p}_u$. The points $\mathbf{p}_i$ are collinear if the Caley-Menger determinant of the points in the base A_b is zero, that is:

$$D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = 0 \rightarrow \text{Collinear} \tag{3.12}$$

This means that the points in A_b cannot be on the same line or the area of A_b will be zero.

If the spheres defined as in Fig. 3.2 do not intersect, there is no solution to the trilateration, as shown in Fig. 3.5 and Fig. 3.6. Here, two cases are shown when the intersection does not happen. In the former, the spheres never make contact because they are far apart, and in the later, one of the spheres is inside the other.

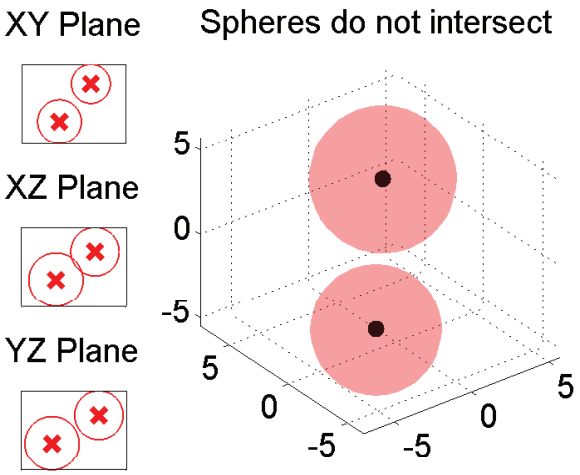


FIGURE 3.5: No sphere intersection case 1. The spheres do not intersect because they are far apart and thus, never make contact, the "air" gap between them is clearly visible.

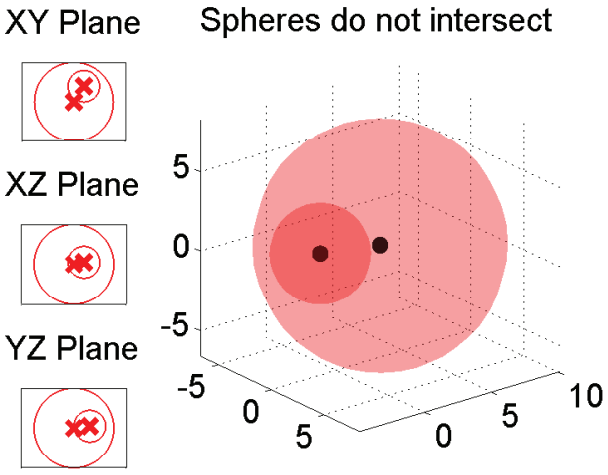


FIGURE 3.6: No sphere intersection case 2. The spheres do not intersect because one of them is completely inside the other, which means there is not a single contact point between them.

Now, mathematically, two spheres with centers at $\mathbf{c}_i$ and $\mathbf{c}_j$, radius r_i and R_j, intersect in a circle with radius $a_{ij} \geq 0$[64]. For three spheres, $n = 3$, the intersection is tested as:

$$\mathbf{p}_u \in \exists a_{12} \cap \exists a_{13} \cap \exists a_{23} \tag{3.13}$$

where a_{ij} is given by:

$$a_{ij} = \frac{1}{2d_{ij}}\sqrt{(-d_{ij}+r_j-R_i)(-d_{ij}-r_j+R_i)(-d_{ij}+r_j+R_i)(d_{ij}+r_j+R_i)}, a_{ij} \geq 0 \quad (3.14)$$

and d_{ij} is the distance from center to center:

$$d_{ij} = \|\mathbf{c}_j - \mathbf{c}_i\| \quad (3.15)$$

All the distances a_{ij} ($i \in 1,2,3$, $j \in 1,2,3$, $i \neq j$) must be greater or equal than zero to have a solution for $\mathbf{p}_u$, which means that the square root inside (3.14) has to be positive. For two spheres located at $\mathbf{p}_1 = [-1.4982, -0.8650, 0]^\intercal$ and $\mathbf{p}_2 = [0, 1.7300, 0]^\intercal$ with radius of 3.46 units, the intersection radius a_ij is 3.118 units, as can be appreciated in Fig. 3.7.

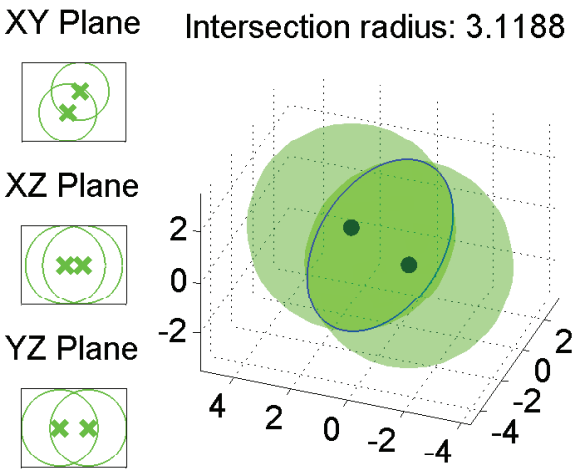
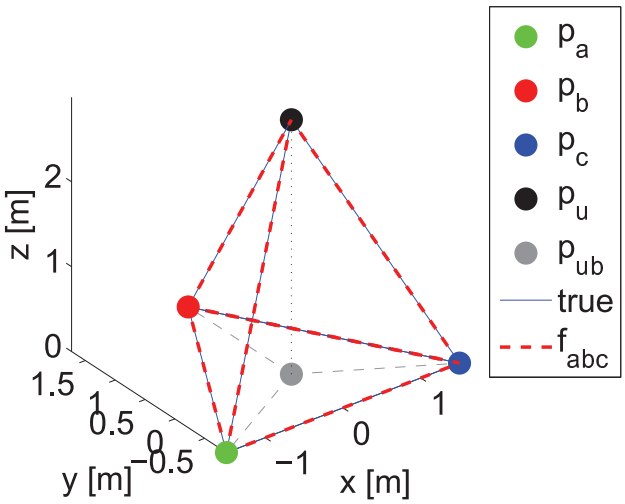


FIGURE 3.7: Sphere intersection radius. Two spheres intersect in a circle with radius a_ij. For two spheres located at $\mathbf{p}_1 = [-1.4982, -0.8650, 0]^\intercal$ and $\mathbf{p}_2 = [0, 1.7300, 0]^\intercal$ with radius of 3.46 units, the intersection radius a_ij is 3.118 units.

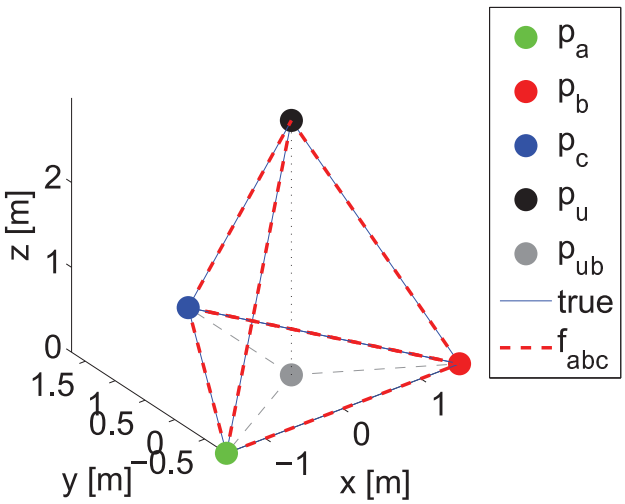
3.4 Trilateration function

If more robots are added to the OWMR multi robot formation, then $n \geq 3$ and three points $\mathbf{p}_a$, $\mathbf{p}_b$ and $\mathbf{p}_c$ have to be selected to perform the trilateration. Thus, a suitable notation will be required for the trilateration. This notation will be denoted as the function f_{abc}.

$$f_{abc} = \mathbf{p}_u = \text{Trilateration}(\mathbf{p}_a, \mathbf{p}_b, \mathbf{p}_c, l_{au}, l_{bu}, l_{cu}) \quad (3.16)$$



(a) CW formation.



(b) CCW formation.

FIGURE 3.8: f_{abc} numerical simulation 1. Three points $\mathbf{p}_i$ are arranged in a triangle centered at the origin with the point $\mathbf{p}_u$ held above the plane A_b. Both CW and CCW orientations are tested.

so, the coordinates of $\mathbf{p}_u$ are:

$$f_{abc} = \mathbf{p}_u = \begin{bmatrix} x_u \\ y_u \\ z_u \end{bmatrix} \tag{3.17}$$

The value of $\mathbf{p}_u$ does not have to be used for trilateration. The positions $\mathbf{p}_i$ and distances l_{iu} are obtained from sensors mounted on the OWMR. Also, remember that the intersection verification has to be done three times, Eq. (3.14).

This method to compute the trilateration allows for a three dimensional calculation of the point $\mathbf{p}_u$, which is in accordance with the objectives of this thesis. Several simulations were performed to validate the algorithm. Three points are arranged in a triangle centered at the origin, in the first simulation $\mathbf{p}_u$ is held above the points $\mathbf{p}_i$, the test was performed in CW and CCW orientations, see Fig. 3.8.

Then a simulation was performed with $\mathbf{p}_u$ outside the base plane A_b, shown in Fig. 3.9. The position $\mathbf{p}_u$ is correctly calculated.

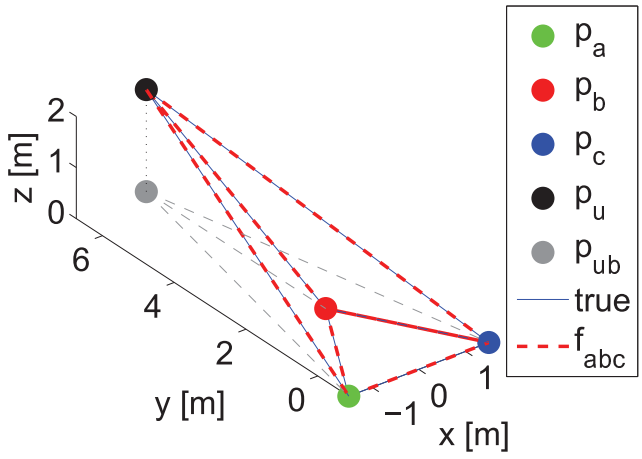


FIGURE 3.9: f_{abc} numerical simulation 2. The position of $\mathbf{p}_u$ can be correctly calculated even if it is outside the base plane A_b.

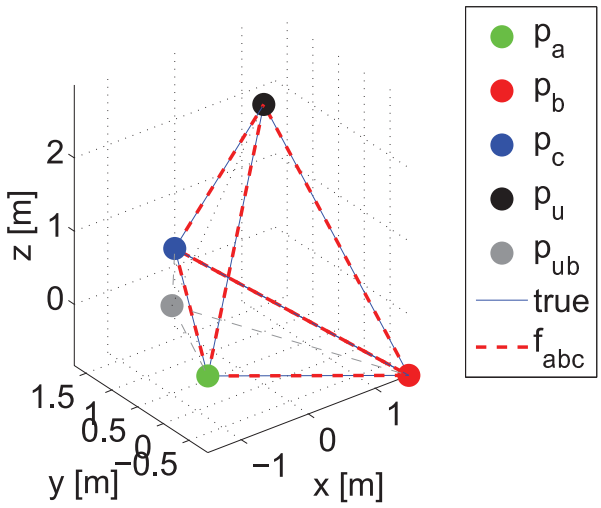


FIGURE 3.10: f_{abc} numerical simulation 3. Even if the base plane is tilted, the position $\mathbf{p}_u$ is correctly computed, validating the three dimensional execution.

Also, the system is tested in a tilted condition, see Fig. 3.10. It can be appreciated that the solution can be found correctly, validating the three dimensional approach presented in this work. For this test, the points $\mathbf{p}_i$ are located at:

$$\mathbf{p}_a = \begin{bmatrix} -1.4982 \\ -0.8650 \\ 0.2114 \end{bmatrix} \tag{3.18}$$

$$\mathbf{p}_b = \begin{bmatrix} 1.4982 \\ -0.8650 \\ -0.8351 \end{bmatrix} \tag{3.19}$$

$$\mathbf{p}_c = \begin{bmatrix} 0 \\ 1.7300 \\ 0.2358 \end{bmatrix} \tag{3.20}$$

All the tests were performed without adding noise, which mean they are fully solvable and in Chapter 6 the effects of noisy measurements will be analyzed. Finally, Algorithm 2 summarizes the trilateration process.

If, e.g., only three robots are employed OWMR_1, OWMR_2 and OWMR_3, then $a = 1$, $b = 2$ and $c = 3$.

Algorithm 2 Trilateration.

procedure TRILATERATION($\mathbf{p}_a$,$\mathbf{p}_b$,$\mathbf{p}_c$,l_{au},l_{bu},l_{cu})
  if COLLINEAR($\mathbf{p}_a$,$\mathbf{p}_b$,$\mathbf{p}_c$) **then**
    return FAIL
  end if
  if INTERSECTION($\mathbf{p}_a$,$\mathbf{p}_b$,l_{au},l_{bu})< 0 **then**
    return FAIL
  end if
  if INTERSECTION($\mathbf{p}_a$,$\mathbf{p}_c$,l_{au},l_{cu})< 0 **then**
    return FAIL
  end if
  if INTERSECTION($\mathbf{p}_b$,$\mathbf{p}_c$,l_{bu},l_{cu})< 0 **then**
    return FAIL
  end if
  $\mathbf{v}_{ab}$ and $\mathbf{v}_{ac} \leftarrow$ Eq. 3.3 and Eq. 3.4
  k_1, k_2 and $k_3 \leftarrow$ Eq. 3.6, Eq. 3.7 and Eq. 3.8
  $\mathbf{p}_n \leftarrow$ Eq. 3.9
  GETK3SIGN($\mathbf{p}_i$,$\mathbf{p}_n$)
  $\mathbf{p}_{ub} \leftarrow$ Eq. 3.5
  $\mathbf{p}_u \leftarrow$ Eq. 3.2
  return $\mathbf{p}_u$
end procedure

3.5 Conclusions

The mathematical tools required to compute the trilateration were explained in this chapter sections. The geometric aspects of the problem were analyzed. Among the methods available to solve the problem, the Caley-Menger determinant was chosen for its simplicity and usefulness in three dimensions. An algorithm to perform the trilateration was developed taking in account failure scenarios. This algorithm is to be implemented in real time and some basic numerical tests were executed to validate the good performance in the execution.

4 Sensed Measurements for Localization

As stated in Chapter 3, localization by means of trilateration needs as inputs the positions $\mathbf{p}_i$ and the distances l_{iu}. These inputs are information gathered from sensors mounted on the robots, and this robots, are employed as the stations in trilateration. The inputs are required to solve the UAV position $\mathbf{p}_u$ in Algorithm 2. Each robot must be capable to measure its own position in world frame coordinates:

$$\mathbf{p}_i = [x_i, y_i, z_i]^\mathsf{T} \tag{4.1}$$

Also it is needed to provide the distances l_{iu} to $\mathbf{p}_u$:

$$l_{iu} = \|\mathbf{p}_u - \mathbf{p}_i\| \tag{4.2}$$

The way $\mathbf{p}_i$ is calculated has to be independent of the trilateration algorithm. That means that as long as a robot can provide $\mathbf{p}_i$, regardless of the dynamics, any type of robot can be employed, e.g. the algorithm works the same if using an OWMR, a differential drive vehicle or a legged humanoid. In this work a formation of OWMR is employed.

The same concept applies for the distances l_{iu}, they can be calculated with any method as long as they provide the lengths between positions $\mathbf{p}_i$ and $\mathbf{p}_u$. It is assumed that l_{iu} is calculated center to center. In this research distances l_{iu} are measured using monocular cameras employing a fiducial system, reducing the complexity that image tracking systems have. Each OWMR is independent in the sense that it can take commands on its own, but at the same time is part of a multi robot formation used to perform the Multi Robot Localization (MRL). Next, it is explained in detail how $\mathbf{p}_i$ and l_{iu} are obtained.

4.1 Positions $\mathbf{p}_i$

The OWMR robots used have holonomic behavior, meaning that they can move in any direction without restriction, e.g. they do not need to rotate to change direction. Any wheel of the OWMR has an angular velocity ω. Using the right hand rule, if the wheel rotates to the outside of the robot along the center of the wheel, then the wheel speed is negative, and positive if the rotation is to the inside of the robot, see Fig. 4.1.

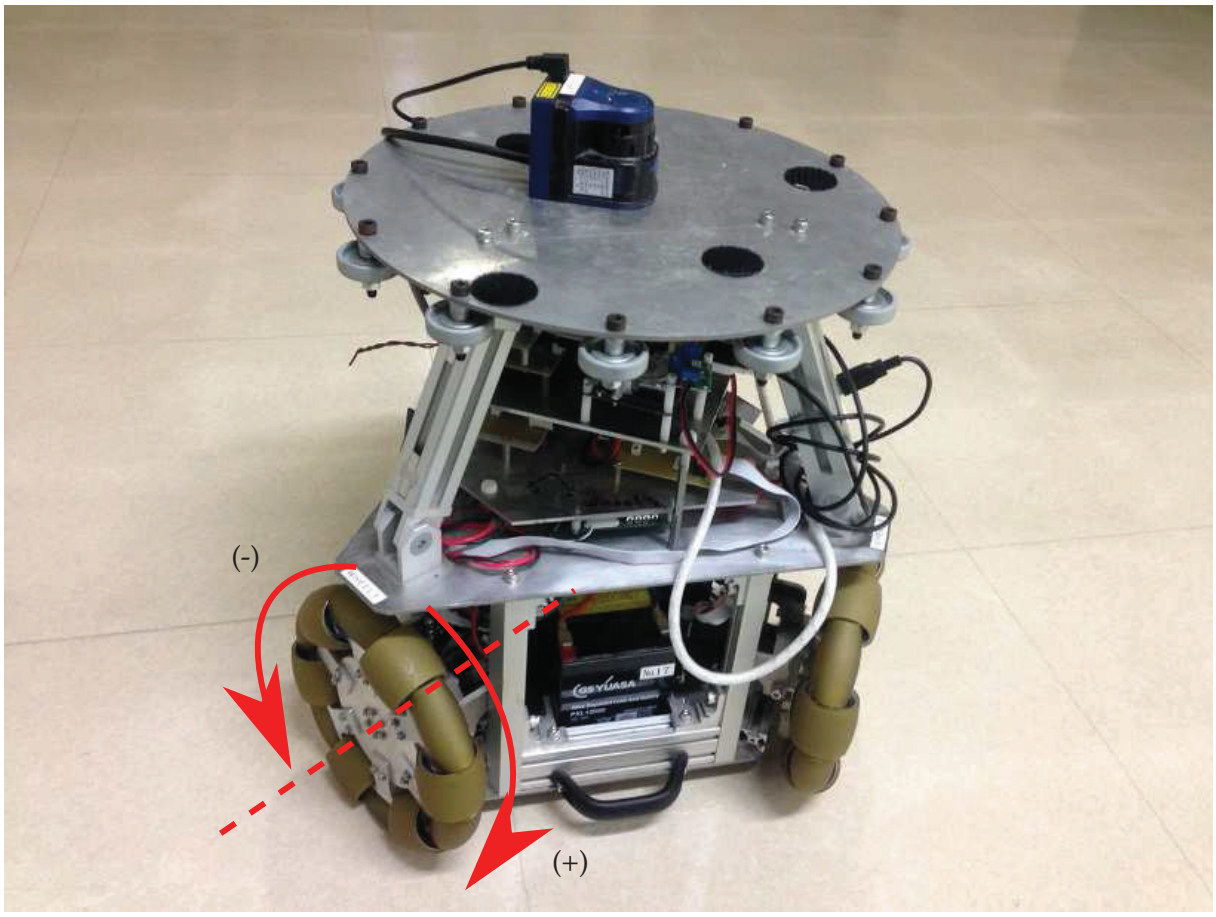


FIGURE 4.1: OWMR wheel speed direction. Taking the right hand rule, if the wheel rotates to the outside of the robot ω is negative, otherwise is positive.

The angular velocities are read from encoders mounted on the shaft of motors, using a gear drive with ratio n. From the encoders, the wheel speeds are given in $[\text{pulses/s}]$. A wheel encoder speed ω_{enc} is related to a wheel tangential velocity v_t trough a conversion factor c_m,

which gives the tangential displacement of the wheel in SI units, and is calculated as:

$$c_m = \pi d/(nC_e) \tag{4.3}$$

where, d is the wheel diameter, n the gear ratio and C_e the encoder resolution. Then, the wheel tangential displacement v_t is given by:

$$v_t = c_m \omega_{enc} \tag{4.4}$$

Two types of OWMR are used[1] (Fig. 4.2) and both types are controlled via speed commands given in the world frame coordinate system or on the robot frame. Speeds in the world frame are denoted by:

$$\mathbf{v}_{xy\psi} = [\dot{x}, \dot{y}, \dot{\psi}]^\intercal \tag{4.5}$$

where $\dot{x}$ and $\dot{y}$ are the linear speeds and $\dot{\psi}$ is the angular speed[2].

Speeds in the local frame are:

$$\mathbf{v}_{n\psi} = [v, v_n, \dot{\psi}]^\intercal \tag{4.6}$$

where v is the forward speed in the local x direction of the robot, v_n the transversal local speed in the y direction. $\dot{\psi}$ is the same in both global and local frames.

Lastly, the tangential speeds are, from (4.4):

$$\mathbf{v}_{123} = [v_{t1}, v_{t2}, v_{t3}]^\intercal \tag{4.7}$$

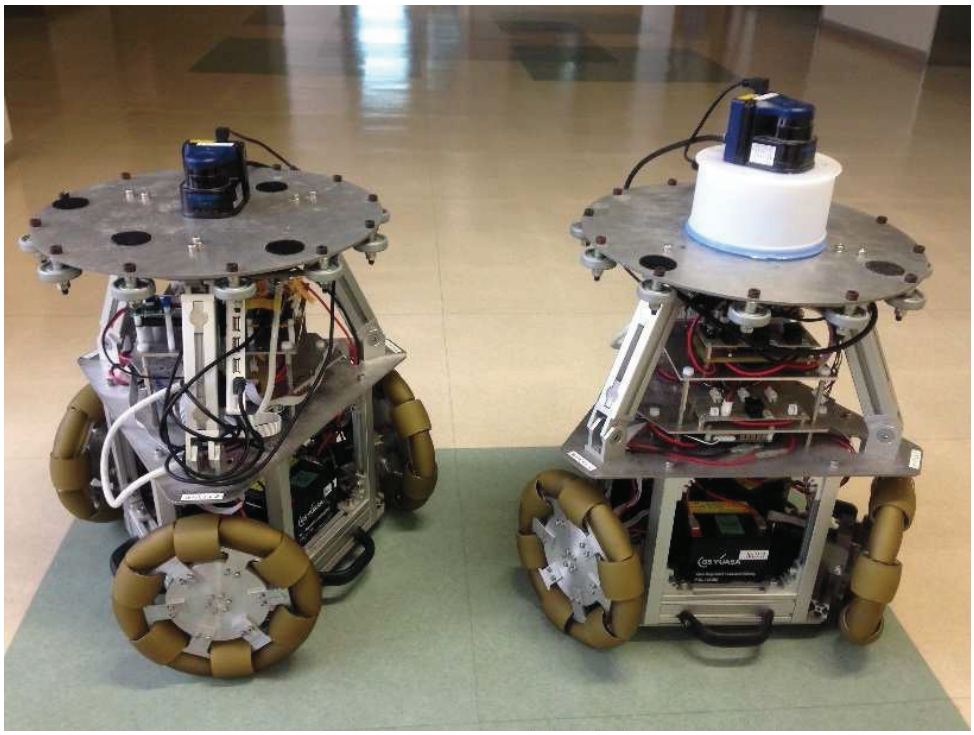
$\mathbf{v}_{123}$ and $\mathbf{v}_{n\psi}$ are related by the geometry of the robot, that is:

$$\mathbf{v}_{n\psi} = G\mathbf{v}_{123} = \begin{bmatrix} 0 & -\sqrt{3}/3 & \sqrt{3}/3 \\ 2/3 & -1/3 & -1/3 \\ 1/3L & 1/3L & 1/3L \end{bmatrix} \mathbf{v}_{123} \tag{4.8}$$

Then, $\mathbf{v}_{xy\psi}$ is calculated with a transformation from local to global coordinates:

$$\mathbf{v}_{xy\psi} = B\mathbf{v}_{n\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v}_{n\psi} \tag{4.9}$$

[1]The OWMR were openly named worker and mini worker.
[2]The OWMR only move in two dimensions



(a) Worker robot.

(b) Mini worker robot.

FIGURE 4.2: OWMR robots. The robots used as stations in trilateration, they calculate their positions based on speed commands. The robots in (a) are capable to perform SLAM while the robots in (b) can only measure their positions from odometry readings.

The plant P relates $\mathbf{v}_{123}$ and $\mathbf{v}_{xy\psi}$, P can be obtained through the Jacobian.

$$\mathbf{v}_{xy\psi} = BG\mathbf{v}_{123} = P\mathbf{v}_{123} \tag{4.10}$$

World speeds, local speeds and tangential wheel speeds, $\mathbf{v}_{xy\psi}$, $\mathbf{v}_{n\psi}$ and $\mathbf{v}_{123}$ respectively, are depicted in Fig. 4.3. $\dot{\psi}$ is the same in global and local frames.

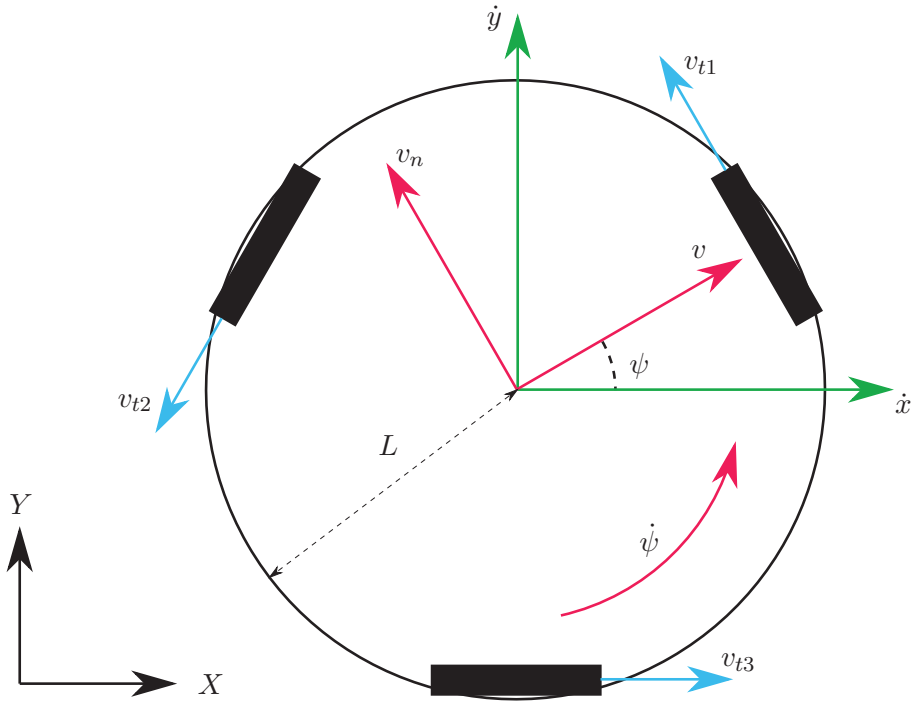


FIGURE 4.3: OWMR speeds. World speeds, local speeds and tangential wheel speeds, $\mathbf{v}_{xy\psi}$, $\mathbf{v}_{n\psi}$ and $\mathbf{v}_{123}$ respectively, are depicted. $\dot{\psi}$ is the same in global and local frames.

A vector $\mathbf{p}_i$ is defined as the state vector, it has the two dimensional OWMR pose with an orientation ψ[52]:

$$\mathbf{p}_i = [x_i, y_i, \psi_i,]^\intercal \tag{4.11}$$

The control is called $\mathbf{u}_i$, the input state, it can be delivered in global or local coordinates:

$$\mathbf{u}_i = \mathbf{v}_{xy\psi} = [\dot{x}_i, \dot{y}_i, \dot{\psi}_i]^\intercal \tag{4.12}$$

$$\mathbf{u}_i = \mathbf{v}_{n\psi} = [v, v_n, \dot{\psi}_i]^\intercal \tag{4.13}$$

where (4.12) is expressed in global coordinates and (4.13) in local ones. The wheels are attached to motors $\mathbf{m}_{123}$, and the motor speeds are read by encoders $\boldsymbol{\omega}_{enc123}$:

$$\mathbf{m}_{123} = [m_1, m_2, m_3]^\intercal \tag{4.14}$$

$$\boldsymbol{\omega}_{enc123} = [\omega_{enc1}, \omega_{enc2}, \omega_{enc3}]^\intercal \tag{4.15}$$

The control loop in Fig. 4.4 is implemented on the OWMR to get the positions $\mathbf{p}_i$. The loop runs at a frequency of 100 Hz. P_{GL} is the plant in global or local coordinates depending on the input $\mathbf{u}_i$, (4.16) and (4.17) respectively.

$$P_{GL} = BG \tag{4.16}$$

$$P_{GL} = G \tag{4.17}$$

At each iteration, small errors accumulate over time when reading the encoders $\boldsymbol{\omega}_{enc123}$ and to keep the motors $\mathbf{m}_{123}$ running and the commanded velocity a PID controller is implemented.

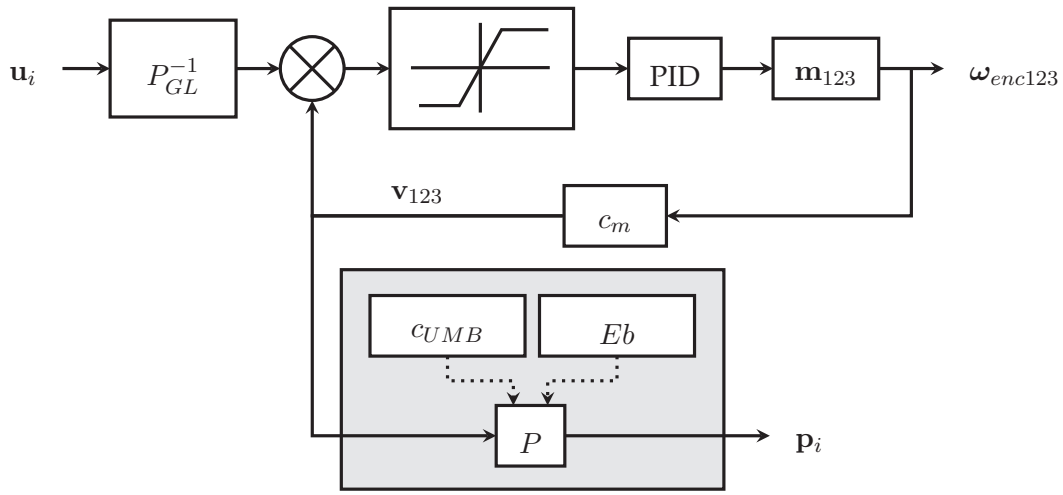


FIGURE 4.4: OWMR Control Loop. The control diagram as implemented in the OWMR is shown. The PID control is used to keep the input $\mathbf{u}_i$ at the required value. Also, the OWMR is calibrated for the wheel diameter and base sizes.

To test the implementation of the motor speed PID controller, one OWMR was commanded to move in the y direction at 0.1 [$\mathrm{m\,s^{-1}}$]. The commanded speed is constant but the encoder output is noisy and to keep it around the commanded speed the PID controls the velocity of the motors $\mathbf{m}_{123}$. Fig. 4.5 shows the results.

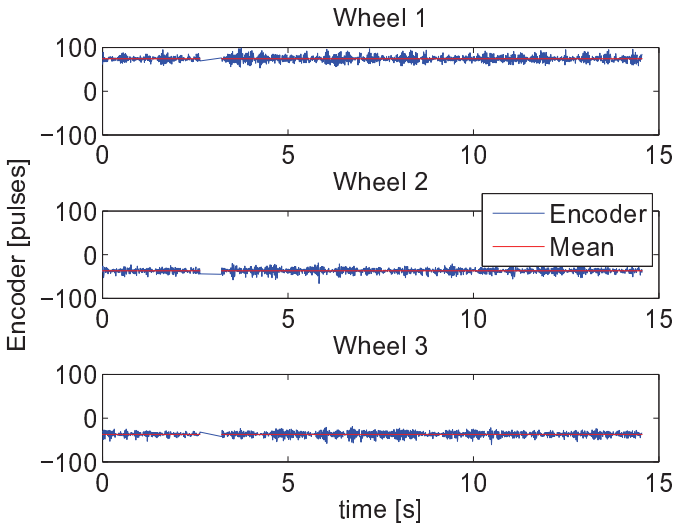


FIGURE 4.5: OWMR PID. When reading the encoders there is an error around a mean value and the PID keeps the input $\mathbf{u}_i$ near this value.

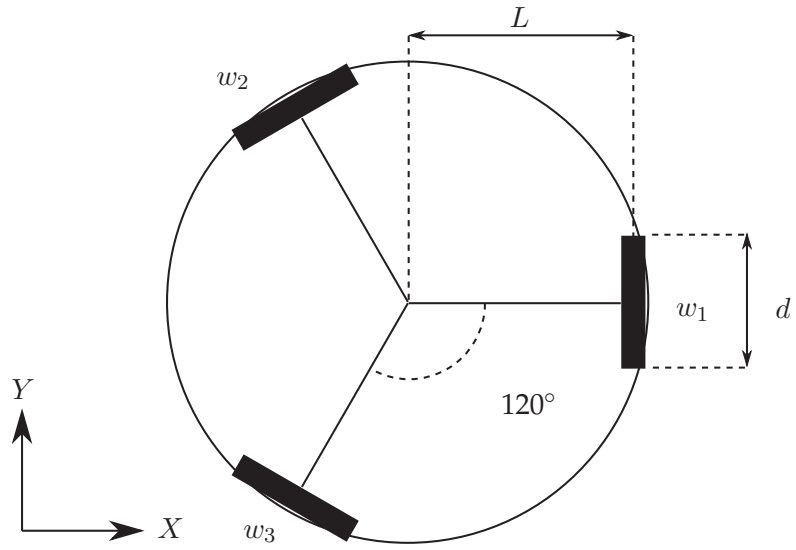


FIGURE 4.6: OWMR top view. The variables used to calculate the dynamics are shown, the length arm L, the wheel diameter d and the wheel speeds ω_i.

The OWMR have systematic errors due to the mechanical aspects of the robot. This means the robot is not symmetrical perfect. In Fig. 4.6 the arm length L and the wheel diameter d are shown. L and d are not the same in all directions and they can be corrected by calibration.

The correction is done using the UMB benchmark[56]. The calibration fixes the wheel diameters d and the base arm lengths L and it is intended to be used on differential drive vehicles. In order to use it on a 3 wheeled OWMR it is proposed in this work to perform the calibration for each wheel as if it were the right wheel, e.g. for wheel 1, $w_1 = w_R$ (see Fig. 4.7).

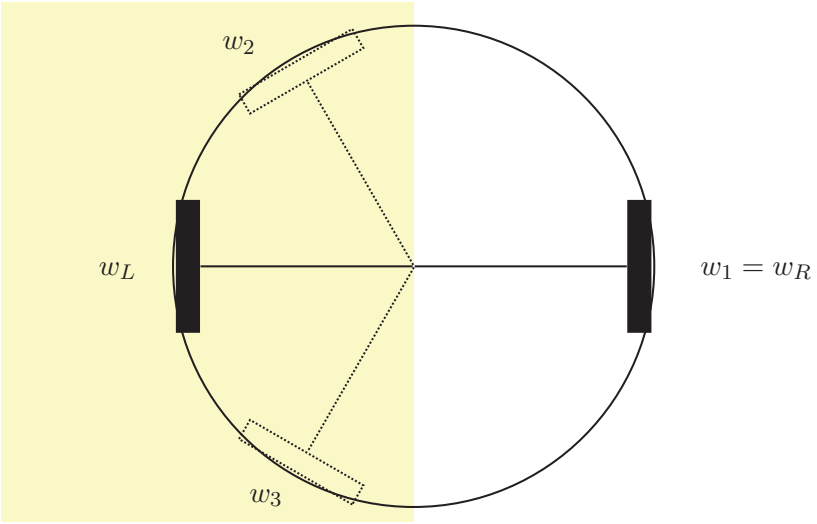


FIGURE 4.7: Wheel 1 UMB setup. The wheels 2 and 3 are taken as the left wheel to perform the UMB calibration.

The UMB calibration was performed for each wheel following the paths in Fig. 4.8. The calibration is not accurate if the robot rotates, that is, when ψ is big. For straight paths in the x and y directions this calibration is enough to get an estimated position $\mathbf{p}_i$.

Then, the world speed is, applying the calibration parameters:

$$\dot{\mathbf{p}}_i = Pc_{UMB}\mathbf{v}_{123} = BG\mathbf{v}'_{123} \tag{4.18}$$

$$\mathbf{v}'_{123} = c_{UMB}\mathbf{v}_{123} = \left[v'_1, v'_2, v'_3\right]^\intercal \tag{4.19}$$

$c_{UMB} \in \mathbb{R}^{3x3}$ is the wheel diameter correction diagonal matrix and the wheel shafts are corrected as:

$$L' = LE_b \tag{4.20}$$

where Eb is the UMB wheelbase correction factor.

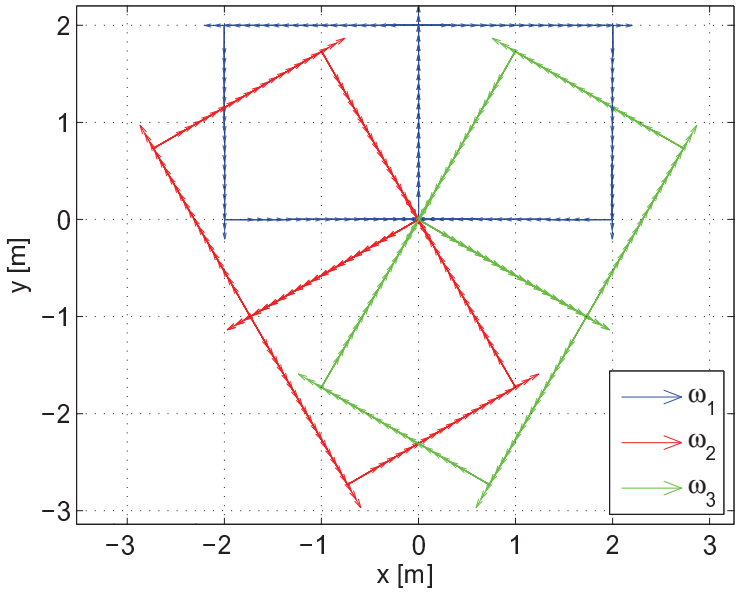


FIGURE 4.8: Test paths for UMB calibration. The paths are taken accordingly to the wheel taken as the right side wheel. The test have to be done three times, each for each wheel.

The calibration parameters for one of the robots are presented in Table 4.1.

TABLE 4.1: UMB calibration parameters for one robot. The calibration is done using the UMB benchmark, the values for each parameter are shown.

Parameter	Value
c_{UMB}	$[1.01378, 1.01060, 1.00719]$
E_b	$[1.00056, 0.99883, 0.99605]$

Eq. (4.18) used with the control loop in Fig. 4.4 gives an inverted calibration, that is, the OWMR will not move straight but rather they will give an estimate of the true position, this is because of limitations on the embedded CPU on board the OWMR. Finally, at each time step δ_t $\mathbf{p}_{it}$ is given by the dynamics g^p.

$$g^p = \mathbf{p}_{it} = \mathbf{p}_{i(t-1)} + \dot{\mathbf{p}}_{it}\delta_t \tag{4.21}$$

The process standard deviations were taken from the encoder readings in each wheel compared against the real input, see Fig. 4.9. The odometry readings have an offset when compared against the reference speed.

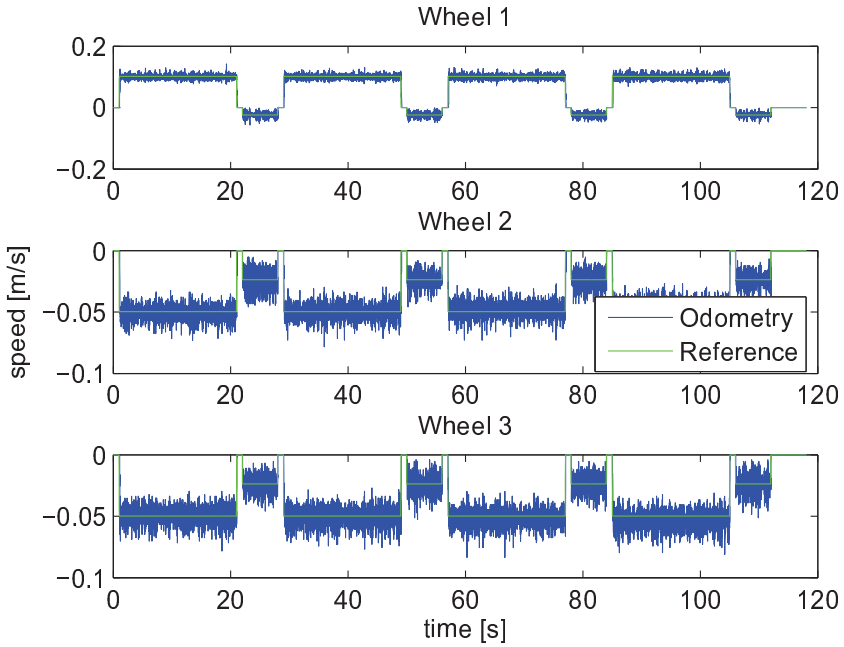


FIGURE 4.9: Wheel readings from UMB tests. The OWMR process covariance $\mathbf{R}^n$ is taken from the errors in the encoders.

The errors for one of the wheels is shown in Fig. 4.10, using this information the distribution of the error measurements can be calculated, as an histogram, shown in Fig. 4.11. The equivalent probability density function of this histogram is shown in red, based on a Gaussian normal distribution. The fit to the error is good, so the mean and the standard deviation can be fitted and added to the dynamics in (4.21).

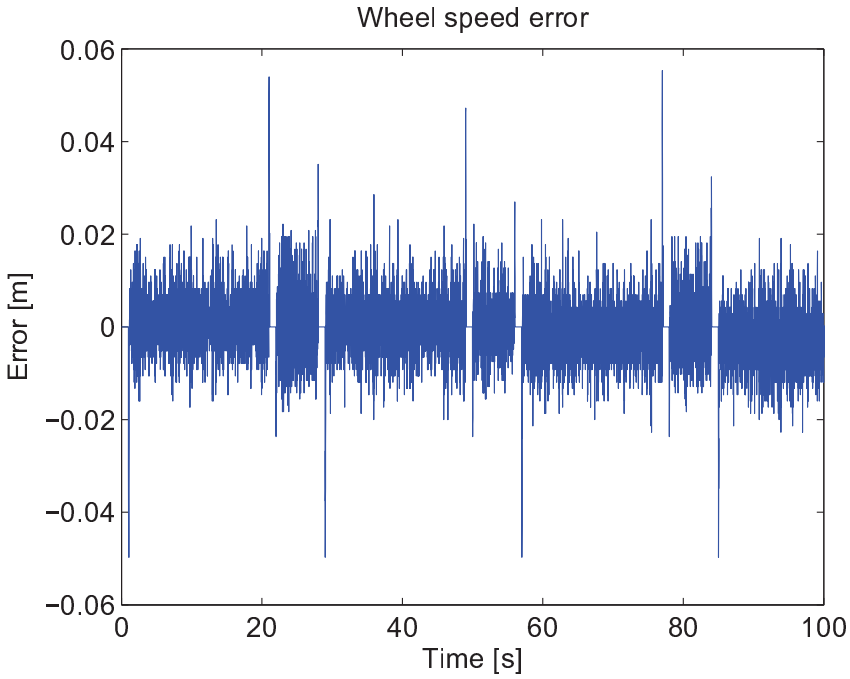


FIGURE 4.10: Wheel speed error for one wheel. The error as read from one wheel of an OWMR is shown.

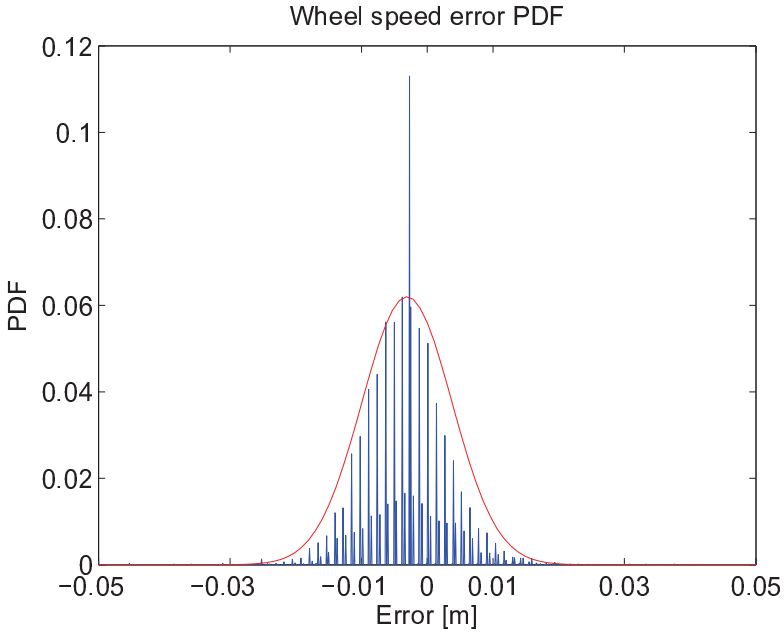


FIGURE 4.11: Wheel speed error PDF for one wheel. The distribution of the error size for one wheel of an OWMR is shown. In red, it is shown the equivalent PDF of this distribution based on a Gaussian normal distribution.

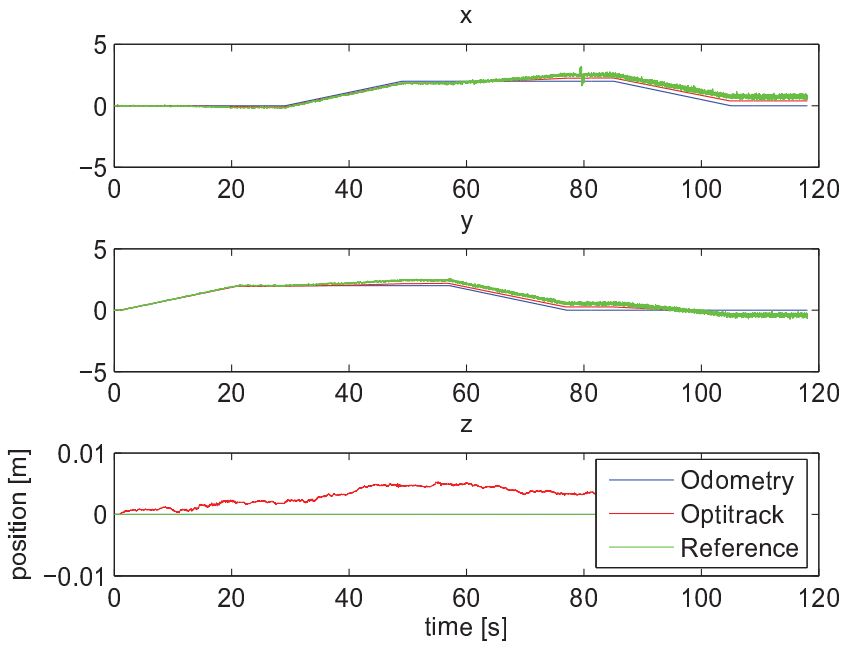


FIGURE 4.12: Position readings from UMB tests. The coordinates x, y and z are taken from different sources to calculate the errors in positions.

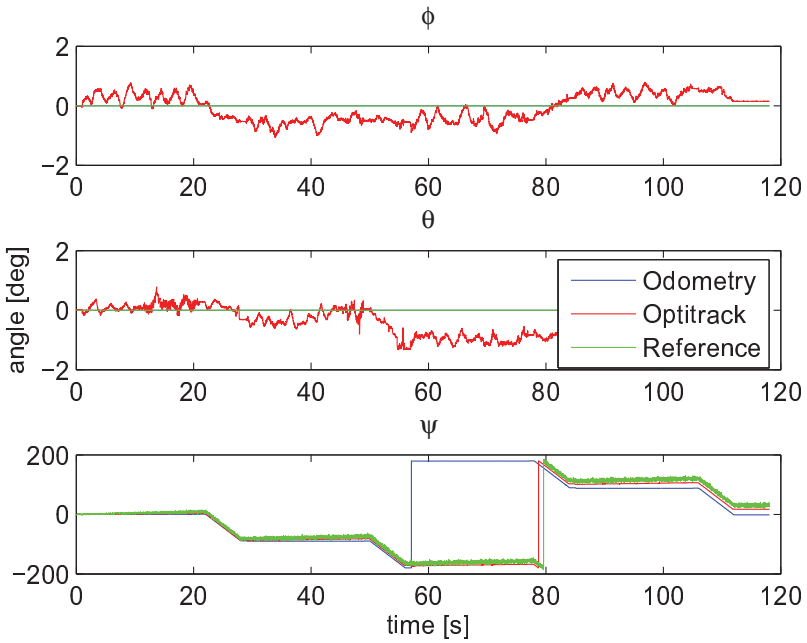


FIGURE 4.13: Orientation readings from UMB tests. The values ρ, θ and ψ are taken from different sources to calculate the errors in orientation.

The positions, x, y and z is shown in Fig. 4.12. As it can be seen the error increases with time, but that is expected as the error adds up at each iteration. The orientations ϕ, θ and ϕ are depicted in Fig. 4.13. As it can be seen, since the OWMR move in two directions the errors in z, θ and ψ can be neglected. The errors related to the measurements x, y and ψ are shown in Fig. 4.14. Results were compared against an Optitrack camera motion system.

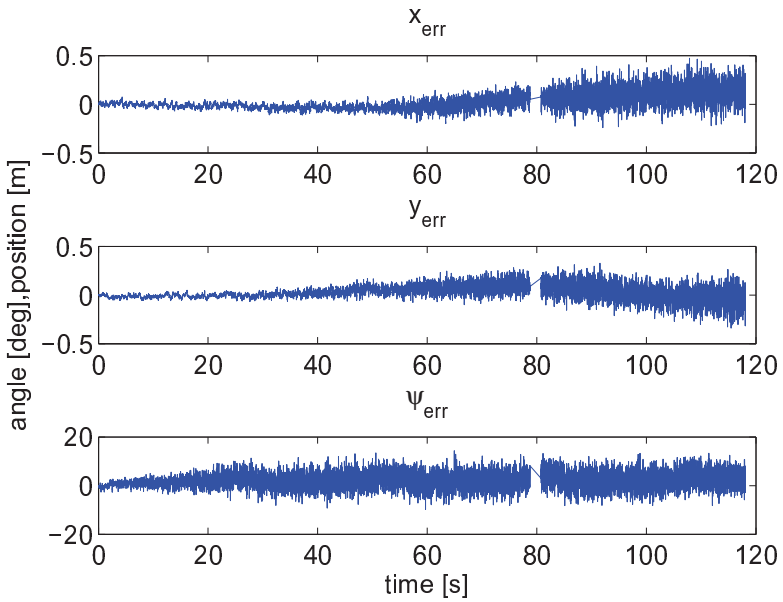


FIGURE 4.14: Errors from UMB tests. The OWMR measurements covariance $\mathbf{Q}^n$ is taken from the errors in the state variables x, y and ψ.

The fit for a Gaussian distribution of the error in x is shown in Fig. 4.15. Due to the error being incremental, the Gaussian fit is not exact. With all this information, the standard deviations σ_R^n and σ_Q^n can be obtained, they are:

$$\sigma_R^n \approx [0.0075, 0.0075, 0.007]^\intercal \tag{4.22}$$

$$\sigma_Q^n \approx [0.095, 0.095, 0.06]^\intercal \tag{4.23}$$

where σ_R^n and σ_Q^n are related to the covariances $\mathbf{R}^n$ and $\mathbf{Q}^n$ respectively. Finally, a simulation was performed using an EKF in a circular path to show how Σ_{pi} changes within a path, the results are shown in Fig. 4.16.

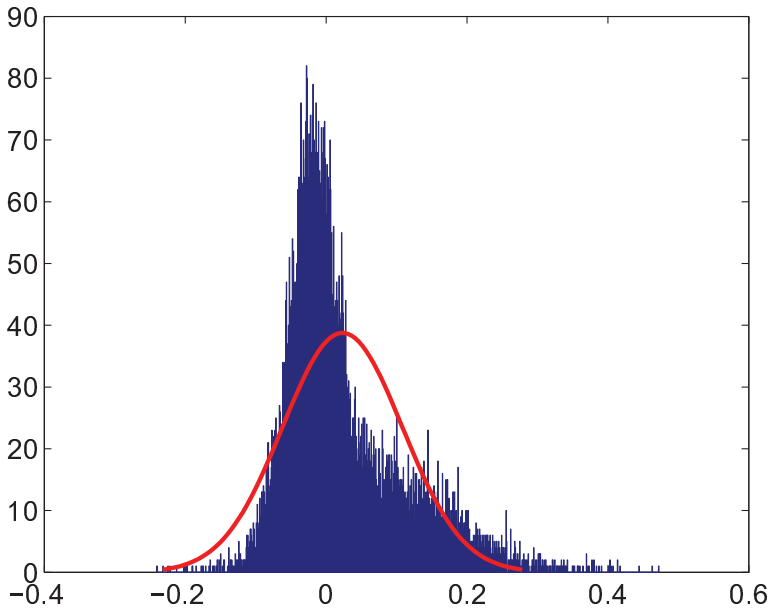


FIGURE 4.15: Gaussian fit for the error in the position in x. The Gaussian fit for the error in x is shown. Due to the error being incremental, the Gaussian fit is not exact.

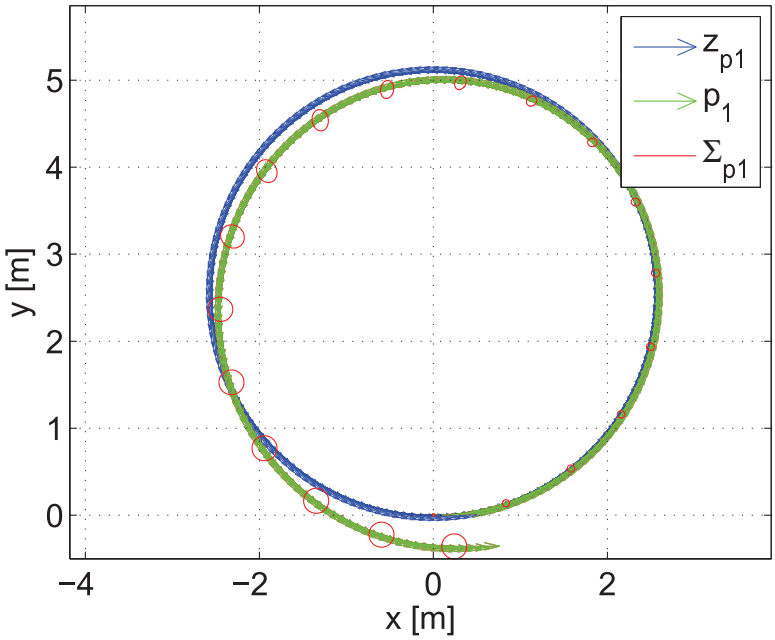
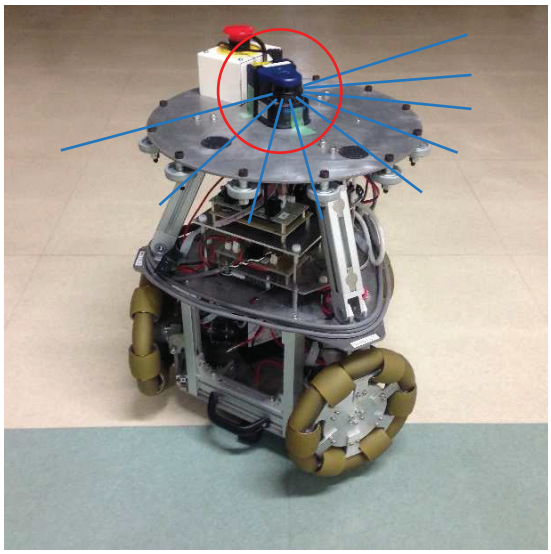


FIGURE 4.16: OWMR$_1$ circular path simulation. The covariance in the OWMR increases as it moves. With SLAM techniques that can be fixed. The OWMR does not rotate, $\dot{\psi} = 0$. $\mathbf{\Sigma}_{p1}$ with 1 σ.

4.2 SLAM

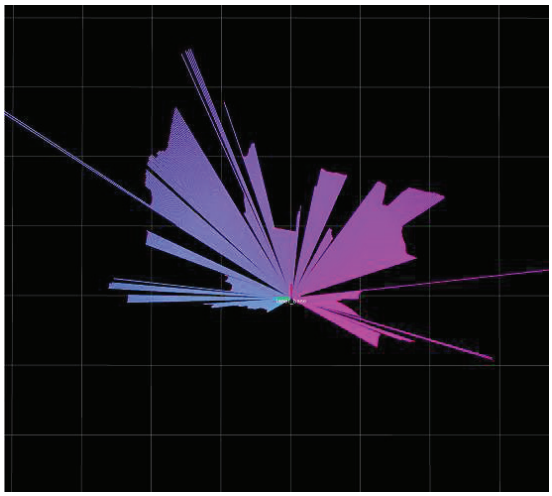
When the pose $\mathbf{p}_i$ is taken only from odometry readings, it is good only for small distances. In the long run, the errors accumulate and also, if the robot drifts, it can lose its position.

The pose can be improved or fixed using SLAM localization techniques. For that the Worker OWMR are equipped with an LRF to localize themselves within a given map (see Fig. 4.17).

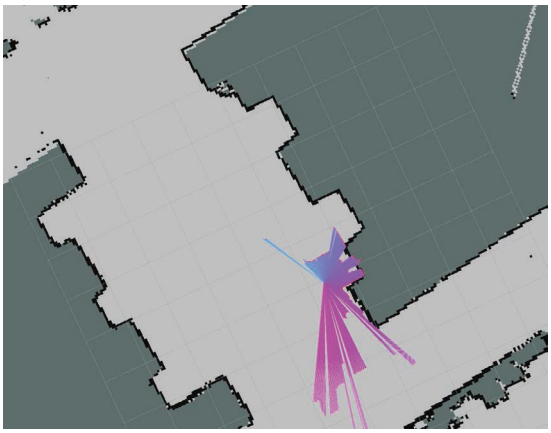


(a) OWMR with an LRF.

(b) Map built with a robot using an LRF.



(c) LRF graphic interface.



(d) LRF in a map.

FIGURE 4.17: Worker robot with an LRF. The Worker OWMR are equipped with an LRF which allows for localization within a map.

The errors in $\mathbf{p}_i$ propagate to the position $\mathbf{p}_u$. To get this propagation a Kalman filter is implemented in the OWMR to get a correct estimate of the covariances in $\mathbf{p}_i$. Localization with encoders will be defined as $\mathbf{p}_i^n$ and localization with an LRF will be defined as $\mathbf{p}_i^r$.

The position $\mathbf{p}_i^r$ is given at a much lower rate than $\mathbf{p}_i^n$, and each time $\mathbf{p}_i^r$ is updated it has to be fused with $\mathbf{p}_i^n$, this can be achieved with an EKF in the update step.

In the prediction step, the new believes are calculated:

$$\bar{\mu}_t^p = g^p(\mathbf{u}_{it}, \mathbf{p}_{i(t-1)}) \tag{4.24}$$

$$\bar{\Sigma}_t^p = G_t^p \Sigma_{t-1}^p (G_t^p)^\intercal + \mathbf{R}_t^p \tag{4.25}$$

then, the Kalman gain is given by:

$$K_t^p = \bar{\Sigma}_t^p (C_t^p)^\intercal \left(C_t^p \bar{\Sigma}_t^p (C_t^p)^\intercal + \mathbf{Q}_t^p \right)^\intercal \tag{4.26}$$

and, finally, the update is:

$$\mu_t^p = \bar{\mu}_t^p + K_t^p \left(\mathbf{z}_t^p - C_t^p \bar{\mu}_t^p \right) \tag{4.27}$$

$$\Sigma_t^p = (\mathbf{I} - K_t^p C_t^p) \bar{\Sigma}_t^p, \mathbf{I} \in \mathbb{R}^{3\times 3} \tag{4.28}$$

where g^p is Eq. (4.21), Σ^p is the Kalman covariance, G_t^p is the Jacobian of (4.21) with respect of the state, that is:

$$G_t^p = \frac{\partial g^p(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} \tag{4.29}$$

The process covariance $\mathbf{R}_t^p$ is the propagation of the error, calculated as[26], [27]:

$$\mathbf{R}_t^p = G_t^p \Sigma_{t-1}^p (G_t^p)^\intercal + V_t^p \mathbf{R}_{t-1}^p (V_t^p)^\intercal \tag{4.30}$$

where the initial value of Σ^p is $\mathbf{R}^n$, and V_t^p is the Jacobian of (4.21) with respect of the input:

$$V_t^p = \frac{\partial g^p(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{u}_t} \tag{4.31}$$

In appendix B the derivation of (4.29) and (4.31) can be found.

The measurements are provided from two different sources, odometry and an LRF.

$$\mathbf{z}_{it}^p = \begin{bmatrix} \mathbf{p}_{it}^n \\ \mathbf{p}_{it}^r \end{bmatrix} \tag{4.32}$$

Fig. 4.18 shows the EKF implementation in the OWMR.

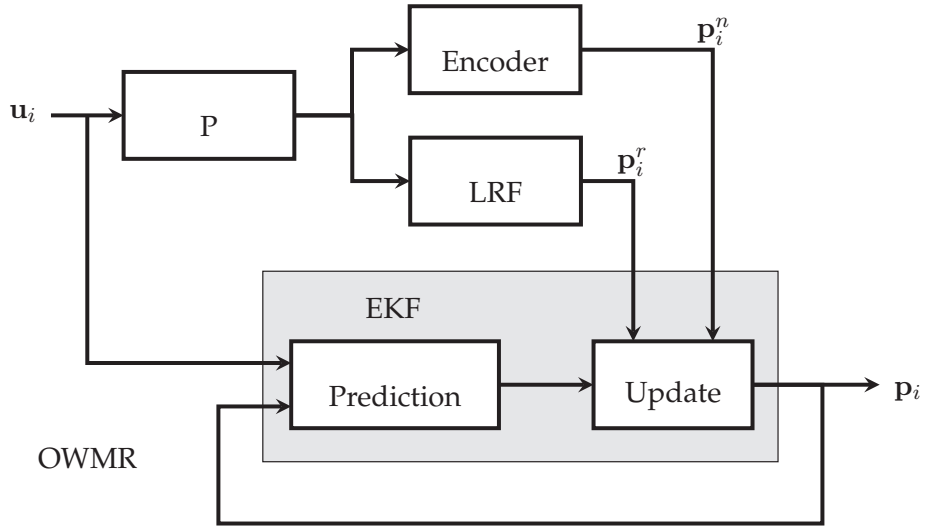


FIGURE 4.18: EKF implementation in the OWMR. The update step in the OWMR EKF is made up of two measurements.

The matrix C_t^p is:

$$C^p = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}, \qquad \mathbf{I} \in \mathbb{R}^{3\times 3} \tag{4.33}$$

The measurement covariance at each time step is:

$$\mathbf{Q}_t^p = \begin{bmatrix} \mathbf{Q}_t^n & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_t^r \end{bmatrix} \tag{4.34}$$

where $\mathbf{Q}_t^n = \mathbf{Q}^n$ is constant and $\mathbf{Q}_t^r$ is given by the SLAM algorithm implemented[27]. The update times for $\mathbf{p}_i^r$ are much slower than the ones for $\mathbf{p}_i^n$, so the equations have to be adjusted accordingly if only $\mathbf{p}_i^n$ is available.

One of the robots was used with SLAM following a square path, the approach taken was fusion the data every time the SLAM algorithm updated the position of the OWMR. The results are presented in Fig. 4.19.

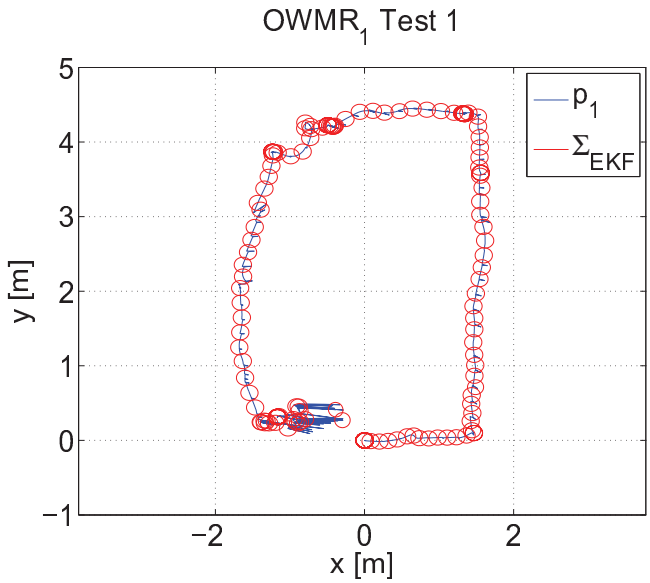


FIGURE 4.19: $OWMR_1$ circular path simulation. The covariance in the OWMR increases as it moves. With SLAM techniques that can be fixed. The OWMR does not rotate, $\dot{\psi} = 0$. $\mathbf{\Sigma}_{p1}$ with 1 σ.

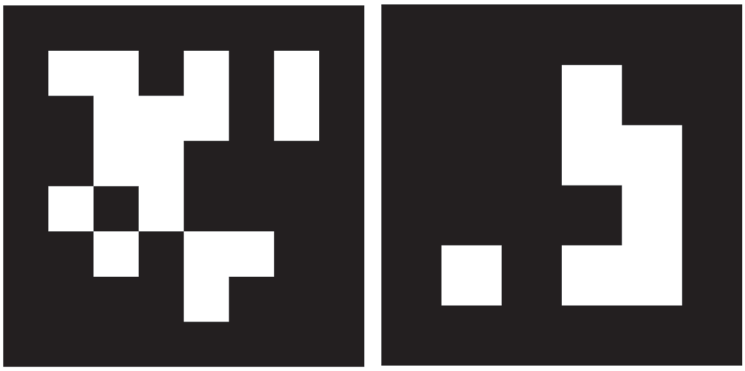
4.3 Distances l_{iu}

(a) Monocular camera.

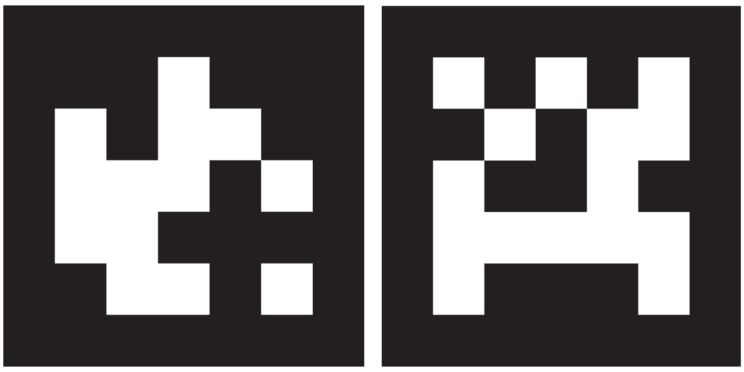
(b) Mini worker OWMR with camera mounted.

FIGURE 4.20: OWMR as used in the experiments. The mini worker is shown as used in the experiments. The monocular cameras are located looking upward.

The distances l_{iu} are measured using cameras mounted on the OWMR, they are measured center to center from the cameras to the UAV. The cameras used are depicted in Fig. 4.20.



(a) Apriltag 1. (b) Apriltag 2.



(c) Apriltag 3. (d) Apriltag 4.

FIGURE 4.21: Apriltags. Different Apriltags are shown. The software can detect the patterns and based on them calculate the distance from a camera to the tag, or the object where the tag is mounted.

World coordinates are translated to image pixel positions by:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \tag{4.35}$$

By (4.35) it can be seen that the depth is lost in the image, so it is needed a way to calculate it, so the distance can be retrieved. Normally, the three dimensional pose is calculated solving the PnP problem[7]. In this research we are only interested on the distance form the image, thus, the fiducial image system Apriltags[65] was chosen for this tasks. This software can calculate the full pose but we are only interested in the distances provided from it.

In Fig. 4.21 some examples of Apriltags are shown. The software has an internal library that allows for the tag identification, and once identified, the distance from the camera to the tag can be retrieved.

In Fig. 4.22 it is shown where the Apriltag is mounted on the UAV for the experiments, and also, it can be seen that the range distance is measured center to center, from the camera to the Apriltag.

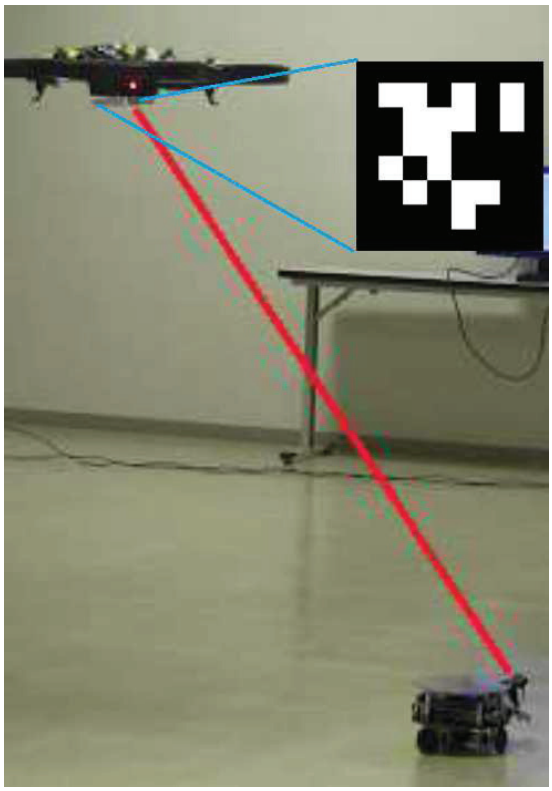


FIGURE 4.22: Apriltag mounted in the UAV. Apriltag mounted in the UAV as it is being used in the experiments. The distance is measured center to center, from the camera to the Apriltag.

To get better results, the cameras are calibrated using OpenCV[66] with an asymmetrical circle pattern, see Fig. 4.23. A calibration software detects this pattern and based on how it truly has to be seen, it calculates the calibration parameters needed to correct an uncalibrated image. These calibration parameters are passed to the Apriltags software and the calibration is done automatically.

The error using Apriltags is calculated by comparing the software distance against the real one, the results are shown in Fig. 4.24. Three runs were performed measuring the distance at intervals of 0.10 m. The error using this fiducial system is around 0.05 m, good enough for the purposes of this research.

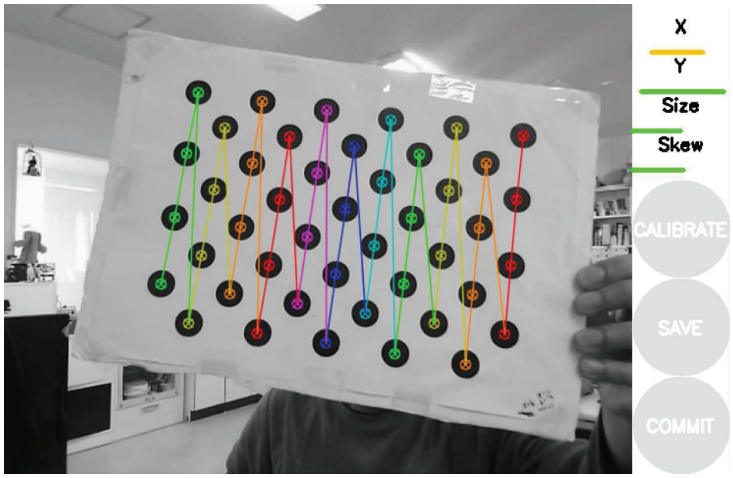


FIGURE 4.23: Asymmetric circles pattern. A calibration software detects this pattern and based on how it truly has to be seen calculates the calibration parameters for a monocular camera.

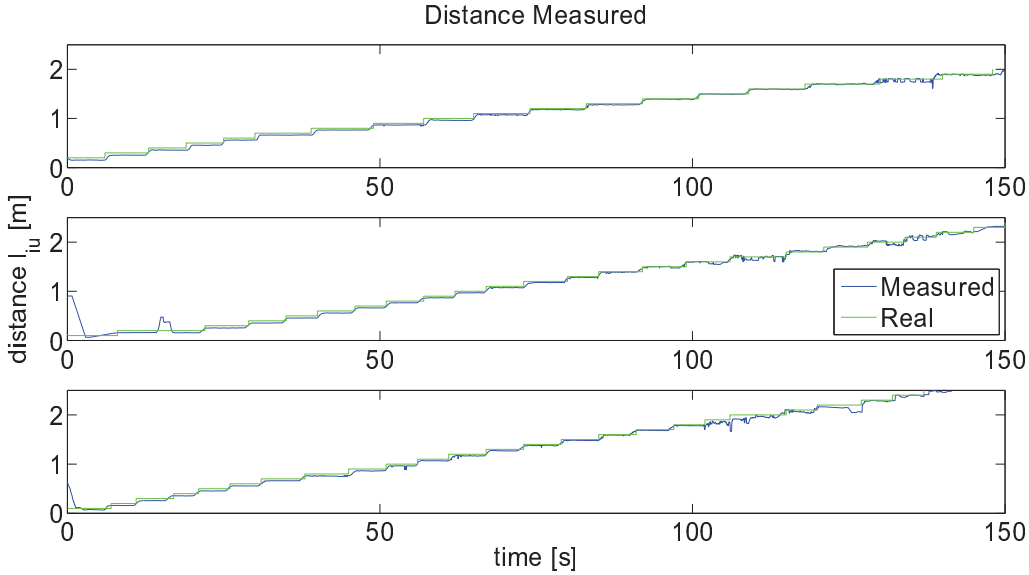


FIGURE 4.24: Apriltags error. To calculate the error when using the Apriltags different distances were measured and compared to a ground truth. Then the standard deviation was taken.

4.4 Conclusions

In this chapter it was explained how the measurements required are obtained. The methods shown are specific to the methodology of this work. As stated before, the measurements are independent of the localization algorithms, as long as the information required is provided by the robots any method can be used. Positions $\mathbf{p}_i$ depend on the robot's dynamics, through odometry readings. Usually, a SLAM technique has to be use to update the measurements so the true position is as real as possible so the robot can perform successfully in the environment. For the distances l_{iu}, in this work, it is used an image processing technique. Other researches use network approaches with radio transmitters to calculate these distances. The following chapters will describe how this information is used in the multi localization algorithm.

5 UAV Localization with a Multi Robot Formation

The multi robot localization calculates the position of an UAV using a formation of OWMR. The OWMR are stations in a GPS framework, and the UAV the receiver. The control diagram is shown in Fig. 5.1.

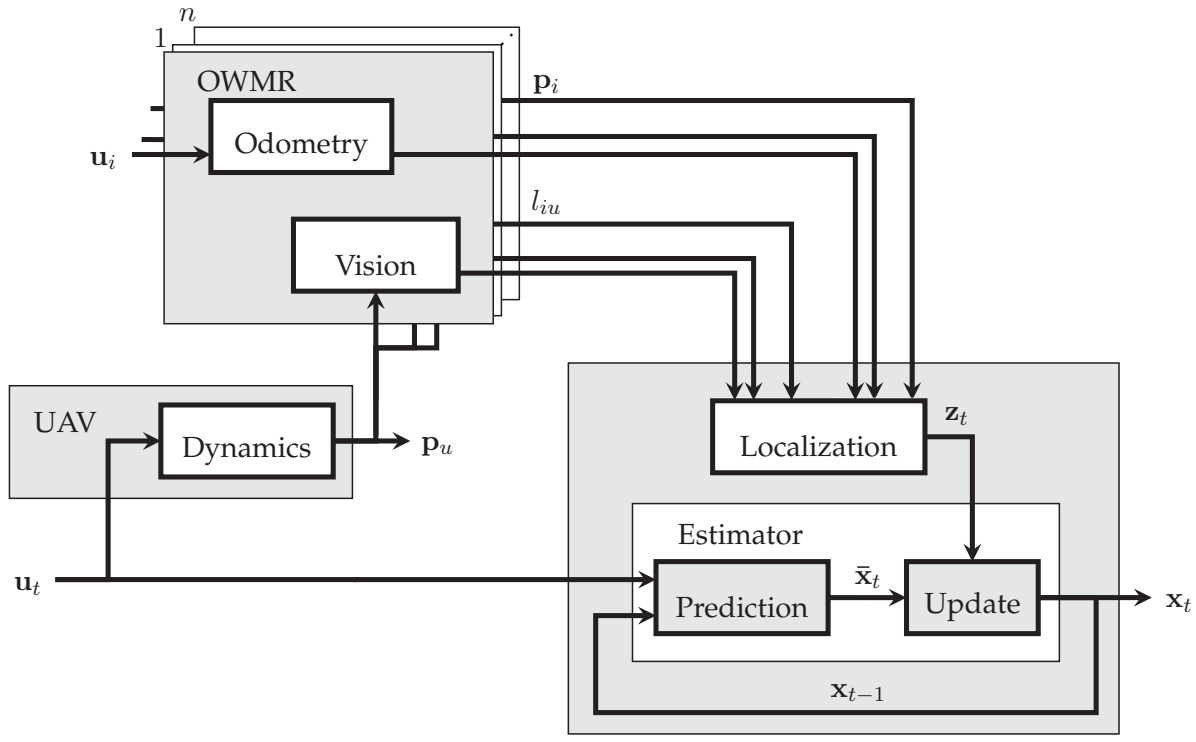


FIGURE 5.1: Multi Robot Localization (MRL) control. The OWMR formation calculates the UAV position. It uses a trilateration approach with independent measurements $\mathbf{p}_i$ and l_{iu} coming from sensors mounted in the OWMR.

The MRL calculates a measurement $\mathbf{z}_t$ and uses an estimator to keep the position $\mathbf{p}_u$ accurate. The variables involved in the algorithm are presented in Table 5.1.

TABLE 5.1: MRL algorithm variables. The variables involved in the MRL algorithm are described.

Variable	Description
l_{iu}	Distance measurements.
$\mathbf{p}_u$	UAV position, the unknown variable.
$\mathbf{p}_i$	OWMR position measurements.
$\mathbf{u}_i$	Input to the OWMR.
$\mathbf{u}_t$	Input to the UAV.
$\mathbf{x}_t$	MRL output, estimated position $\mathbf{p}_u$.
$\mathbf{z}_t$	Measurement for MRL, Eq. (3.16) and Algorithm 2.

The trilateration was explained in Chapter 3, the calculations for positions $\mathbf{p}_i$ and the distances l_{iu} were explained in Chapter 4. In the multi robot approach of this work the number of OWMR can vary, so the number of OWMR is denoted n. Only three robots are needed to perform Algorithm 2 and the selection of which robots can compute it should be based optimality or simple OWMR positions within the formation. Also a redundancy approach can be taken, having different group of robots calculating the position $\mathbf{p}_u$. Depending on the application one approach may be better than others.

The MRL control block uses an estimator, it is used as the measurements in position and distances have errors within them. The errors in positions $\mathbf{p}_i$ and distances l_{iu} have covariances which increase over time. Errors in positions and distances propagate to the UAV position and it is needed to filter the signal $\mathbf{z}_t$ to get an accurate solution for $\mathbf{p}_u$.

The estimation consist in a prediction and an update steps. In the prediction a process model $g(\mathbf{u}_t, \mathbf{x}_{t-1})$ approximates the solution for $\mathbf{p}_u$ based on the input $\mathbf{u}_t$. Generally this process model is just the dynamics of the system.

Then, in the update step, the data is fusion with the measurement signal $\mathbf{z}_t$ to get the true position $\mathbf{p}_u$. The measurement model is defined as $h(\mathbf{x}_t)$.

5.1 Process and measurements models

The unknown position is the three dimensional location of an UAV, but the MRL can be used to track anything, which depending on the situation, can be an UAV, a person inside a building or simply a marker on the ceiling. So the best approach is to use a velocity model. This velocity model is defined as:

$$\mathbf{x}_t = g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \epsilon_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \epsilon_t, \epsilon_t = \mathcal{N}(0, \sigma_R) \tag{5.1}$$

with an input vector:

$$\mathbf{u} = [\dot{x}, \dot{y}, \dot{z}]^\intercal \tag{5.2}$$

and a state vector:

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^\intercal \tag{5.3}$$

A is the process matrix (5.4) and B is the input matrix (5.5).

$$A = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.4}$$

$$B = \begin{bmatrix} \mathbf{I}dt \\ \mathbf{I} \end{bmatrix} \tag{5.5}$$

The measurement model is defined as the trilateration function (3.16):

$$\mathbf{z}_t = h(\mathbf{x}_t) + \delta_t = f_{abc} + \delta_t, \delta_t = \mathcal{N}(0, \sigma_Q) \tag{5.6}$$

ϵ_t and δ_t are noise vectors, assumed to be multivariate Gaussian distributions with mean 0 and variances σ_R and σ_Q, respectively. In equations (5.4) and (5.5), $\mathbf{I}$ and $\mathbf{0}$ are the identity and null matrix, respectively, in $\mathbb{R}^{3x3}$.

Two methods for estimation are addressed, an EKF and a PF. Both methods were implemented following [27]. The methods were implemented having in mind that they can be interchangeable and do not have to interfere with the processing of the MRL algorithm.

While the EKF proves to be more robust to initial conditions the PF have properties that make it more robust in the presence of non linearities. The Kalman filter is one of the most used techniques for estimation due to its simplicity but as the problem to solve gets more complicated other techniques must be used.

5.2 Extended Kalman filter

The extended Kalman filter is a very powerful tool to estimate the state of a dynamic system. It is widely used in robotic applications as well as in localization systems. This filter has good response as long as the linearization at the required state positions represents the non linear model accurately. On the other hand, the EKF is very robust to initial conditions as it can adjust the state values to track the measurement signal adequately.

The implementation of the EKF is very straight forward and the computational cost is very low. The EKF algorithm is presented in Algorithm 3.

Algorithm 3 Extended Kalman filter.

1: **procedure** EXTENDEDKALMANFILTER($\mathbf{x}_{t-1}$,Σ_{t-1},$\mathbf{u}_t$,$\mathbf{z}_t$)
2: $\bar{\mathbf{x}}_t \leftarrow g(\mathbf{u}_t, \mathbf{x}_{t-1})$
3: $\bar{\Sigma}_t \leftarrow G_t \Sigma_{t-1} G_t^\intercal + \mathbf{R}_t)$
4: $K_t \leftarrow \bar{\Sigma}_t H_t^\intercal (H_t \bar{\Sigma}_t H_t^\intercal + \mathbf{Q}_t)^{-1}$
5: $\mathbf{x}_t \leftarrow \bar{\mathbf{x}}_t + K_t(\mathbf{z}_t - h(\bar{\mathbf{x}}_t))$
6: $\Sigma_t \leftarrow (\mathbf{I} - K_t H_t)\bar{\Sigma}_t$
7: **return** $\mathbf{x}_t$, Σ_t
8: **end procedure**

$\mathbf{R}_t$ is the process covariance and $\mathbf{Q}_t$ is the measurement covariance. They are related to σ_R and σ_Q respectively. Steps 1 and 2 are the prediction and steps 4, 5 and 6 are the update to the state. $\mathbf{Q}_t$ is constant but $\mathbf{R}_t$ propagates from positions $\mathbf{p}_i$ and distances l_{iu} to $\mathbf{p}_u$. It will be explained in Chapter 6.

The EKF uses the process and measurements Jacobians, G_t and H_t. G_t is the Jacobian of g given by:

$$G_t = g'(\mathbf{u}_t, \mathbf{x}_{t-1}) = \frac{\partial g(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} = A \tag{5.7}$$

H_t is the Jacobian of h defined by:

$$H_t = h'(\mathbf{x}_{t-1}) = \frac{\partial h(\mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{5.8}$$

The EKF in essence linearize the dynamics at every point. So if the system is highly non linear it is very prompt to fail because it depends on the accuracy of model g to make a proper linearization for G_t.

5.3 Particle filter

The PF deals with the linearization concern of the EKF by consensus. Several particles are taken around the true value requested based on the propagation of the error in the process, then, the particles that are closer to the measurement are employed to calculated the estimated signal. The down side of the PF is that it is very prone to fail if the initial conditions are not good enough. Algorithm 4 shows the particle filter (PF).

Algorithm 4 Particle filter.

procedure PARTICLEFILTER($\mathcal{X}_{t-1}$,$\mathbf{u}_t$,$\mathbf{z}_t$)
  $\mathbf{R}'_t \leftarrow (\mathbf{R}_t^{-1} + H_t^\intercal \mathbf{Q}_t^{-1} H_t)^{-1}$
  $\mathbf{Q}'_t \leftarrow (\mathbf{Q}_t + H_t \mathbf{R}_t H_t^\intercal)$
  for $m = 1$ to M **do**
    $\bar{\mathbf{x}}_t^{[m]} \leftarrow g(\mathbf{u}_t, \mathbf{x}_{t-1}^{[m]})$
    $\bar{\mathbf{z}}_t^{[m]} \leftarrow h(\bar{\mathbf{x}}_{t-1}^{[m]})$
    $\omega_t^{[m]} \leftarrow \mathcal{N}(\bar{\mathbf{z}}_t^{[m]} | \mathbf{z}_t, \mathbf{Q}'_t)$
  end for
  $\omega_t \leftarrow \omega_t / \sum_{m=1}^{M} \omega_t^{[m]}$
  for $m = 1$ to M **do**
    draw i with probability $\propto \omega_t^{[i]}$
    add $\bar{\mathbf{x}}_t^{[i]}$ to $\mathcal{X}_t$
  end for
  $\mathbf{x}_t \leftarrow$ MEAN($\mathcal{X}_t$)
  $M_{new} \leftarrow$ SIZE($\mathcal{X}_t$)
  for $m = M_{new} + 1$ to M **do**
    $\mathbf{x}_t^{[m]} \leftarrow \mathbf{x}_t + \epsilon'_t$, $\epsilon'_t = \mathcal{N}(\mathbf{0}, \mathbf{R}'_t)$
  end for
  return $\mathbf{x}_t$,$\mathcal{X}_t$
end procedure

For this filter, a set of M points, called particles are created, defined as:

$$\mathcal{X} = \mathbf{x}^{[m]}, m \in \{1, \ldots, M\} \tag{5.9}$$

Each of these particles is located within a threshold of the position $\mathbf{p}_u$, see Fig. 5.2. Usually, this threshold is represented by the process covariance $\mathbf{R}_t$. So, at each time step, each particle is:

$$\mathbf{x}_t^{[m]} = \mathbf{x}_t + \epsilon_t, \epsilon_t = \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \tag{5.10}$$

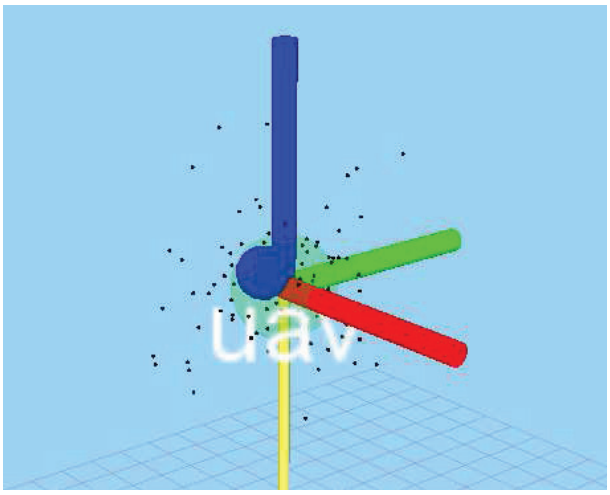


FIGURE 5.2: Particles dispersed around a position $\mathbf{x}_t$. Each particles is located within a threshold of the position $\mathbf{p}_u$. Usually, this threshold is represented by the process covariance $\mathbf{R}_t$.

All the particles are propagated through the dynamics of g and the measurements h to have a set of particle measurements $\mathcal{Z}$.

$$\mathcal{X}_t = g(\mathbf{u}_t, \mathcal{X}_{t-1}) \tag{5.11}$$

$$\mathcal{Z}_t = h(\mathcal{X}_t) \tag{5.12}$$

The true measurement $\mathbf{z}_t$ is given a weight factor based on the measurements covariance $\mathbf{Q}_t$, and then, the closest particles to this measurement are taken to calculate the true position $\mathbf{p}_u$ using for example, the mean value.

$$\mathbf{p}_u = \text{MEAN}(\mathcal{X}_t) \tag{5.13}$$

The PF algorithm was slightly modified from the one in [27]. For the algorithm to work it is needed to know how the error propagates from measurements to the process. In step 2 and 3 optimal process and measurements covariances ($\mathbf{R}'_t$ and $\mathbf{Q}'_t$) are calculated based on an optimal approach using the Jacobian H_t[44]. Doing that inside the filter allows for interchangeability between the EKF and the PF algorithms in the implementation.

Steps 5 to 7 are the prediction. Step 5 and 6 are the propagation of the state through the model, Eq. (5.11) and (5.12), step 7 is the probability of the measurement which is then normalized in step 9.

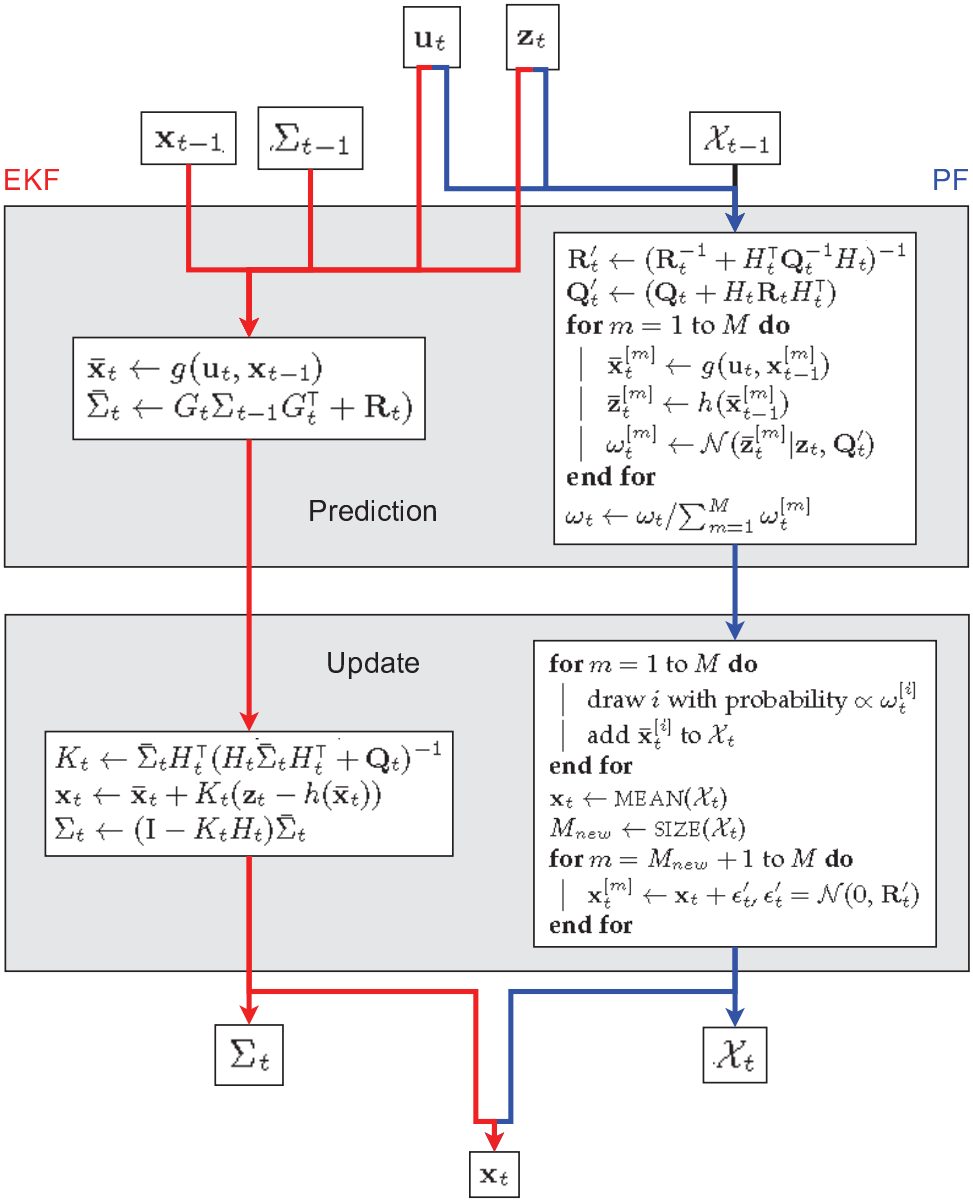


FIGURE 5.3: EKF and PF comparison chart. Both estimators consist in a prediction and an update steps but the way they perform the operation affect which estimator is better for certain circumstances.

The update consists of a resampling and particles addition. Steps 11 and 12 are the resampling which keeps only the particles close enough to the measurement $\mathbf{z}_t$. After resampling, the number of particles decreases and it is needed to add more particles to $\mathcal{X}_t$. The new particles are generated around the mean of $\mathcal{X}_t$, calculated in step 14. New particles are created in step 17 by adding a normal Gaussian random vector ϵ'_t to $\mathbf{x}_t$ using the optimal process covariance $\mathbf{R}'_t$, Eq. (5.13).

In Fig. 5.3 a comparison between the flow of data in the EKF and the PF is carried on.Both estimators consist in a prediction and an update steps but the way they perform the operation affect which estimator is better for certain circumstances.

5.4 Conclusions

The UAV position is meant to be found by a group of ground robots, so, in this chapter it was explained how the localization will be performed is such scenarios for the MRL approach. The mathematical tools required for the correct deployment of the MRL algorithm were described. Two estimation techniques were studied, an EKF and a PF, and both techniques can be used with the localization method. The estimators require a process and measurements model, which were described in this chapter. Defining the process model as a velocity model allows for the MRL to be employed in localizing any object and how the trilateration model was translated into the measurements model was also defined.

6 Uncertainty Propagation and Accuracy Analysis

In network environments the trilateration is performed between static stations and moving receivers, so the error in trilateration is attributed solely to the measurements l_{iu}. In the case of moving stations the error in the positions $\mathbf{p}_i$ cannot be neglected because there is an error that accumulates over time. Errors in positions $\mathbf{p}_i$ and distances l_{iu} propagate to $\mathbf{p}_u$, so the localization needs an estimation method like an EKF to properly calculate the position $\mathbf{p}_u$[15]. The localization accuracy can be changed by modifying the formation of OWMR and the uncertainty propagation can provide a reference for designing and controlling the formation of OWMR for some requested accuracy properties of the UAV.

6.1 Dilution of precision

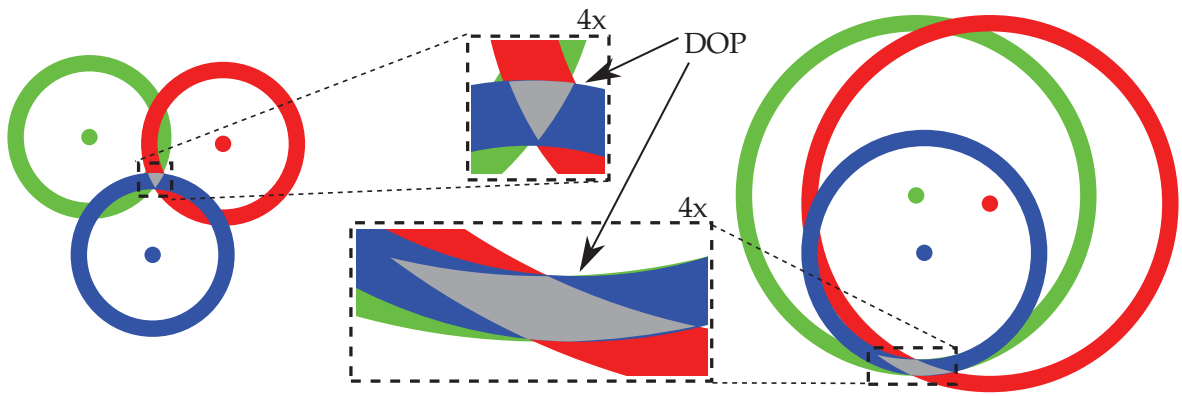


FIGURE 6.1: Dilution of precision (DOP) for different configurations. The robots are denoted by dots and the error size is denoted by the width of the rings. On the left, the intersection region is small, depicted in gray, because all the robots are equidistant to the intersection. On the right side, as the solution moves and the configuration changes, the uncertainty region increases.

In trilateration with noisy measurements the output suffers from dilution of precision (DOP)[18] as the geometry of the problem changes. In Fig. 6.1 this concept is depicted for the two dimensional case. On the left side an intersection region is shown for all the points having the same distance to the unknown. On the right side it can be appreciated that if the distances l_{iu} change the uncertainty in the true position increases.

Taken this concept to the three dimensional case is evident that the dilution also happens. In Fig. 6.2 the OWMR are located below the UAV at equidistant distances. The positions $\mathbf{p}_i$ and distances l_{iu} have errors that make the solution lie within an intersection region, depicted as a particle distribution of possible solution for $\mathbf{p}_u$.

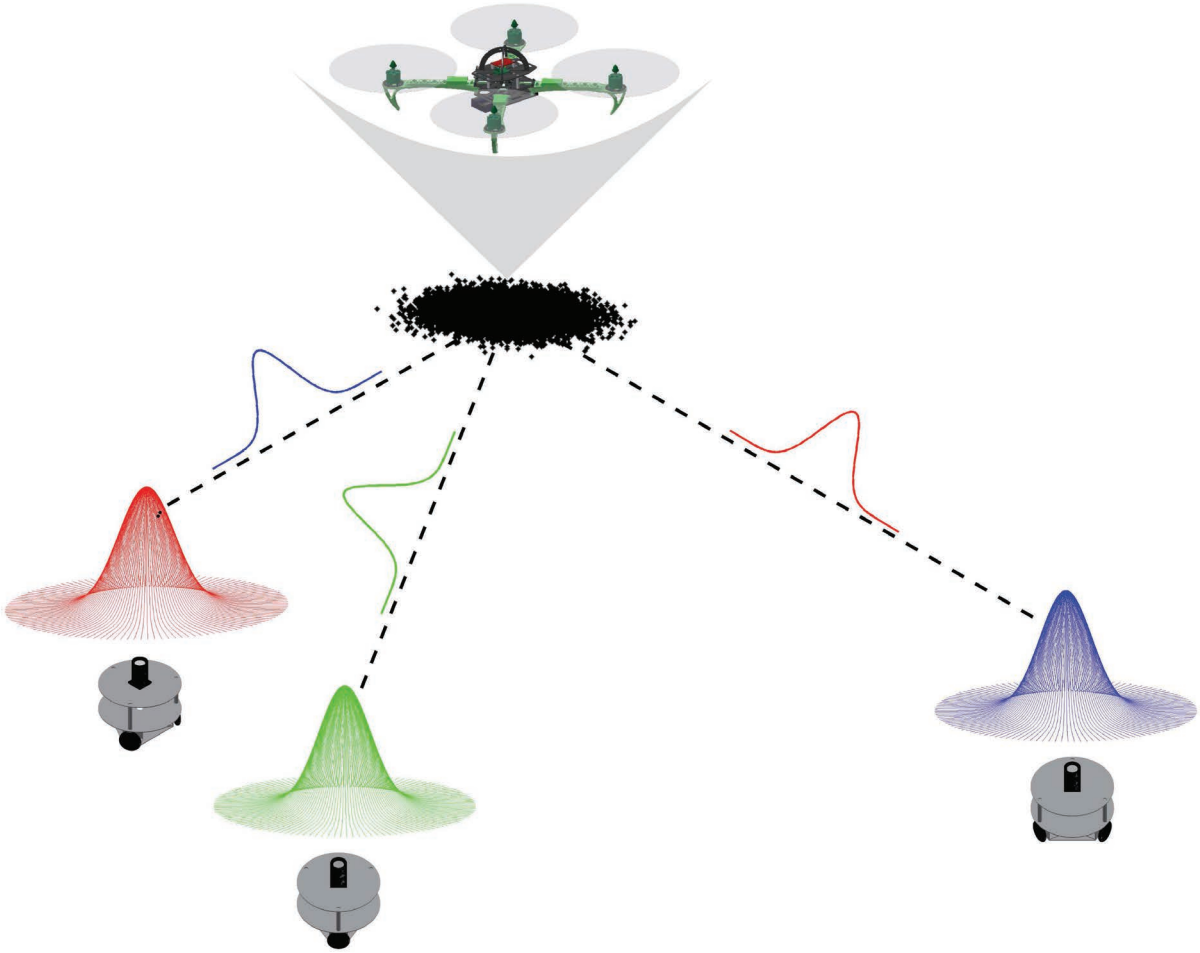


FIGURE 6.2: DOP in 3D case 1. The OWMR uncertainty is denoted as two dimensional Gaussian distributions and the range measurements uncertainty are depicted as linear Gaussian distributions. The intersection region is shown in three dimensions as a particle distribution. The particle distribution is calculated with a Montecarlo approach.

As the positions $\mathbf{p}_i$ and distances l_{iu} change, the possible solutions for $\mathbf{p}_u$ dilute, changing the intersection region, in Fig. 6.3 the dilution is shown when the UAV moves away from the formation. The errors on the horizontal plane are small compared to the errors in the z direction but that can be mitigated using sensor fusion techniques.

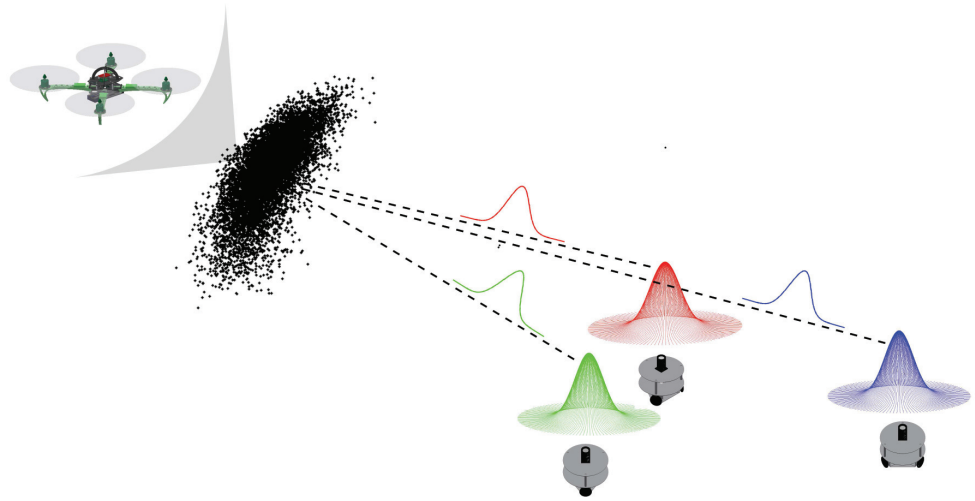


FIGURE 6.3: DOP in 3D case 2. If $_{|u}$ moves away from the formation, the intersection region dilutes as the positions $\mathbf{p}_i$ and distances l_{iu} change. The z direction is the more affected by the error propagation.

In Fig. 6.2 and Fig. 6.3 the particle distribution is taken with a Montecarlo methodology but it can be calculated by means of the Jacobian of the trilateration function.

6.2 Uncertainty propagation

Measurements errors propagate to a measurement covariance $\mathbf{Q}_t$[25], [26], [30]. The UAV position $\mathbf{p}_u$ depends on measurements coming from six independent sources, three OWMR positions $\mathbf{p}_i$ and three distances l_{iu}, with covariance matrices Σ_{pi} and Σ_l, respectively.

$$\mathbf{Q}_t = H_{zt}\mathbf{Q}_\Sigma H_{zt}^\intercal \tag{6.1}$$

where $\mathbf{Q}_\Sigma$ is:

$$\mathbf{Q}_\Sigma = \begin{bmatrix} \mathbf{\Sigma}_{p1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{p2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_{p3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_l \end{bmatrix} \tag{6.2}$$

$\mathbf{\Sigma}_{pi}$, $i \in 1, 2, 3$ are the position covariances of the corresponding OWMR$_i$, which are taken from the EKF implemented on the OWMR:

$$\mathbf{\Sigma}_{pi} = \begin{bmatrix} \sigma^2_{xxi} & \sigma^2_{xyi} & \sigma^2_{xzi} \\ \sigma^2_{yxi} & \sigma^2_{yyi} & \sigma^2_{yzi} \\ \sigma^2_{zxi} & \sigma^2_{zyi} & \sigma^2_{zzi} \end{bmatrix}, \qquad i \in 1, 2, 3 \tag{6.3}$$

$\mathbf{\Sigma}_l$ is the distance measurements covariance and $\mathbf{0}$ is the null matrix in $\mathbb{R}^{3\times3}$:

$$\mathbf{\Sigma}_l = \begin{bmatrix} \Sigma_{l1} & 0 & 0 \\ 0 & \Sigma_{l2} & 0 \\ 0 & 0 & \Sigma_{l3} \end{bmatrix} = \begin{bmatrix} \sigma^2_{l1} & 0 & 0 \\ 0 & \sigma^2_{l2} & 0 \\ 0 & 0 & \sigma^2_{l3} \end{bmatrix} \tag{6.4}$$

$H_{zt} \in \mathbb{R}^{3\times12}$ is the Jacobian of f_{abc} with respect of the measurements $\mathbf{q}_z$:

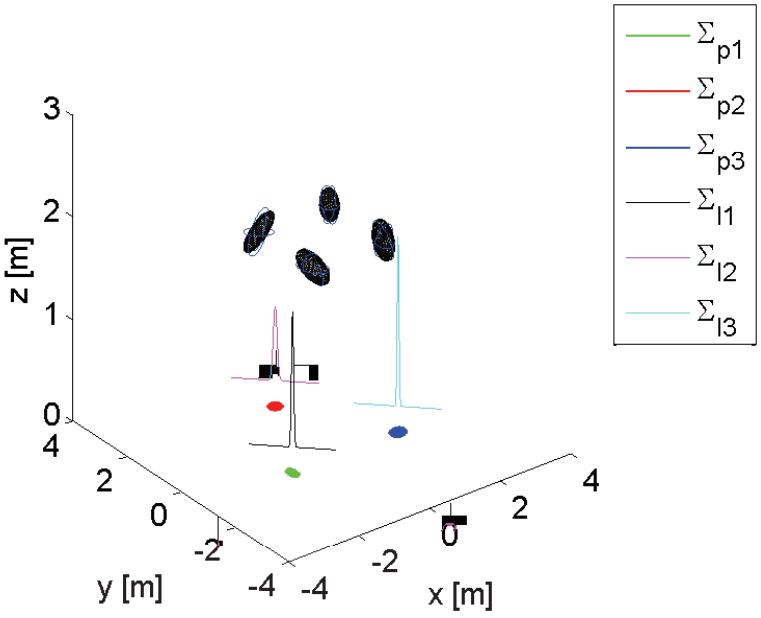
$$H_{zt} = \frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{q}_z} = \frac{\partial f_{abc}}{\partial \mathbf{q}_z} = \frac{\partial \mathbf{p}_u}{\partial \mathbf{q}_z} = \begin{bmatrix} \frac{\partial \mathbf{p}_u}{\partial \mathbf{p}_1} & \frac{\partial \mathbf{p}_u}{\partial \mathbf{p}_2} & \frac{\partial \mathbf{p}_u}{\partial \mathbf{p}_3} & \frac{\partial \mathbf{p}_u}{\partial \mathbf{l}} \end{bmatrix} \tag{6.5}$$

$$\mathbf{q}_z = \begin{bmatrix} \mathbf{p}_1^\intercal & \mathbf{p}_2^\intercal & \mathbf{p}_3^\intercal & l_{1u} & l_{2u} & l_{3u} \end{bmatrix}^\intercal = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{l} \end{bmatrix} \tag{6.6}$$

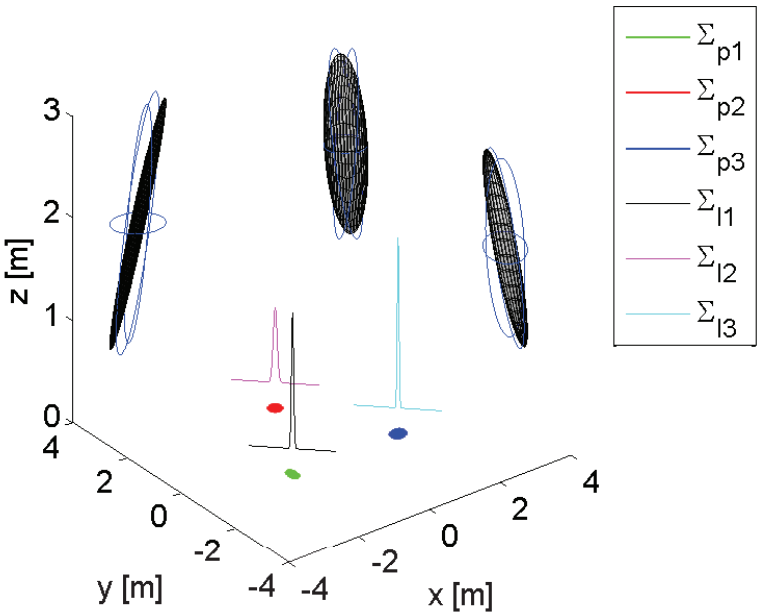
It is assumed that the positions $\mathbf{p}_i$ are independent of each other, otherwise the off diagonal elements in (6.2) are not zero. Now, to solve (6.5), it can be expanded as:

$$\begin{aligned} H_{zt} = \frac{\partial \mathbf{p}_1}{\partial \mathbf{q}_z} + \frac{\partial k_1 \mathbf{v}_{12}}{\partial \mathbf{q}_z} + \frac{\partial k_2 \mathbf{v}_{13}}{\partial \mathbf{q}_z} \pm \frac{\partial k_3 \mathbf{v}_{12} \times \mathbf{v}_{13}}{\partial \mathbf{q}_z} \\ = \frac{\partial \mathbf{p}_1}{\partial \mathbf{q}_z} + k_1 \frac{\partial \mathbf{v}_{12}}{\partial \mathbf{q}_z} + \mathbf{v}_{12} \frac{\partial k_1}{\partial \mathbf{q}_z} + k_2 \frac{\partial \mathbf{v}_{13}}{\partial \mathbf{q}_z} \\ + \mathbf{v}_{13} \frac{\partial k_2}{\partial \mathbf{q}_z} \pm k_3 [\mathbf{v}_{12}]_x \frac{\partial \mathbf{v}_{13}}{\partial \mathbf{q}_z} \mp k_3 [\mathbf{v}_{13}]_x \frac{\partial \mathbf{v}_{12}}{\partial \mathbf{q}_z} \pm (\mathbf{v}_{12} \times \mathbf{v}_{13}) \frac{\partial k_3}{\partial \mathbf{q}_z} \end{aligned} \tag{6.7}$$

where $[\cdot]_x$ is the skew symmetric matrix defining the cross product, $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_x \mathbf{b}$. The signs in (6.7) are taken accordingly to the sign of k_3 back in section 3.2. In Appendix C the derivation for the partials of H_{zt} is explained.



(a) $\mathbf{Q}_t$ near the center of formation.



(b) $\mathbf{Q}_t$ away from formation.

FIGURE 6.4: $\mathbf{Q}_t$ at different positions. Three OWMR are positioned at the vertexes of an equilateral triangle of three meters centered at the origin. As the position $\mathbf{p}_u$ changes the measurements l_{iu} change, and also, the errors associated with the position $\mathbf{p}_i$ make the matrix $\mathbf{Q}_t$ to change due to the geometry in the problem. $\mathbf{\Sigma}_{p1} \models [0.08, 0.02, 0]^\intercal @75°$, $\mathbf{\Sigma}_{p2} \models [0.06, 0.06, 0]^\intercal @120°$, $\mathbf{\Sigma}_{p3} \models [0.08, 0.06, 0]^\intercal @30°$ and $\mathbf{\Sigma}_l \models [0.025, 0.045, 0.02]^\intercal$.

A numerical simulation was executed to verify $\mathbf{Q}_t$ changes. Three OWMR are positioned at the vertexes of an equilateral triangle of three meters centered at the origin. The UAV moves around the formation, $\mathbf{\Sigma}_{pi}$ and $\mathbf{\Sigma}_l$ were given constant random values, the results are presented in Fig. 6.4. $\mathbf{\Sigma}_{p1} \models [0.08, 0.02, 0]^\intercal$ rotated 75 degrees, $\mathbf{\Sigma}_{p2} \models [0.06, 0.06, 0]^\intercal$ rotated 120 degrees, $\mathbf{\Sigma}_{p3} \models [0.08, 0.06, 0]^\intercal$ rotated 30 degrees and $\mathbf{\Sigma}_l \models [0.025, 0.045, 0.02]^\intercal$[1]. The UAV was hold at 2 meters and moved in the x and y direction. The covariances of $\mathbf{p}_i$ are represented by error ellipses, the covariances l_{iu} by Gaussian distributions and $\mathbf{Q}_t$ by error ellipsoids[2]. As the UAV moves away from the formation $\mathbf{Q}_t$ increases.

The DOP can be used to measure the error in $\mathbf{Q}_t$. It is computed as a unit less index for the error size of $\mathbf{Q}_t$[19]. In trilateration DOP is given as:

$$\text{DOP} = \log \begin{vmatrix} \frac{\partial l_{1u}}{\partial x_u} & \frac{\partial l_{1u}}{\partial y_u} & \frac{\partial l_{1u}}{\partial z_u} \\ \frac{\partial l_{2u}}{\partial x_u} & \frac{\partial l_{2u}}{\partial y_u} & \frac{\partial l_{2u}}{\partial z_u} \\ \frac{\partial l_{3u}}{\partial x_u} & \frac{\partial l_{3u}}{\partial y_u} & \frac{\partial l_{3u}}{\partial z_u} \end{vmatrix}^{-1} = \log \begin{vmatrix} \frac{x_u - x_1}{l_{1u}} & \frac{y_u - y_1}{l_{1u}} & \frac{z_u - z_1}{l_{1u}} \\ \frac{x_u - x_2}{l_{2u}} & \frac{y_u - y_2}{l_{2u}} & \frac{z_u - z_2}{l_{2u}} \\ \frac{x_u - x_3}{l_{3u}} & \frac{y_u - y_3}{l_{3u}} & \frac{z_u - z_3}{l_{3u}} \end{vmatrix}^{-1} \tag{6.8}$$

DOP increases exponentially so a $\log$ is used in (6.8). As DOP does not take in account the off diagonal elements in (6.1) is not suitable as a measurement of the error in $\mathbf{Q}_t$. Thus, the error can be calculated as the Spherical Error Probable (SEP), defined as trace of the Cholesky factorization of $\mathbf{Q}_t$:

$$\text{SEP} = 0.59 \;\; \text{tr}(\text{Cholesky}(\mathbf{Q}_t)) \tag{6.9}$$

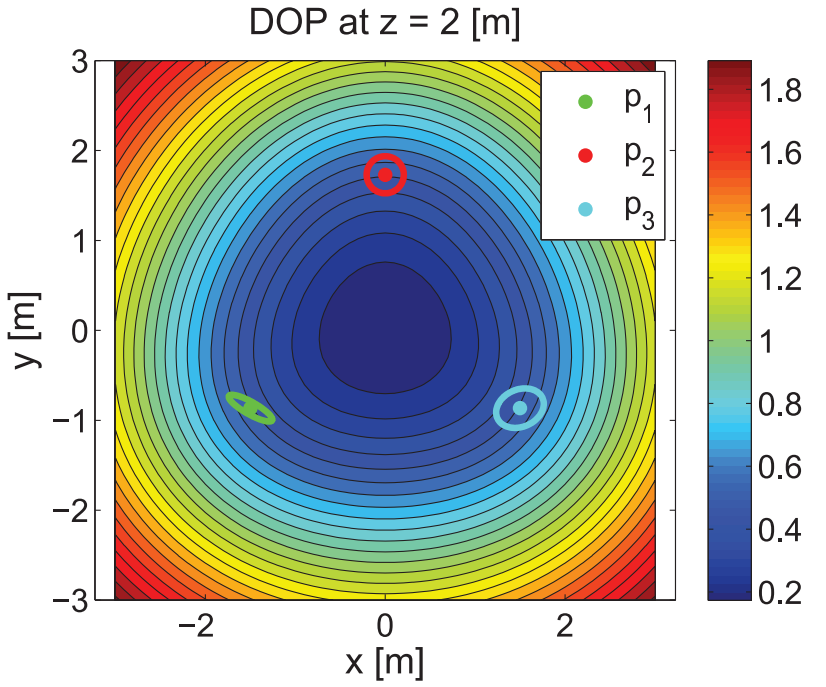
In order to include the off diagonal elements of $\mathbf{Q}_t$ it is proposed to calculate the SEP as:

$$\text{SEP} = 0.59\sqrt{\det \mathbf{Q}_t} \tag{6.10}$$

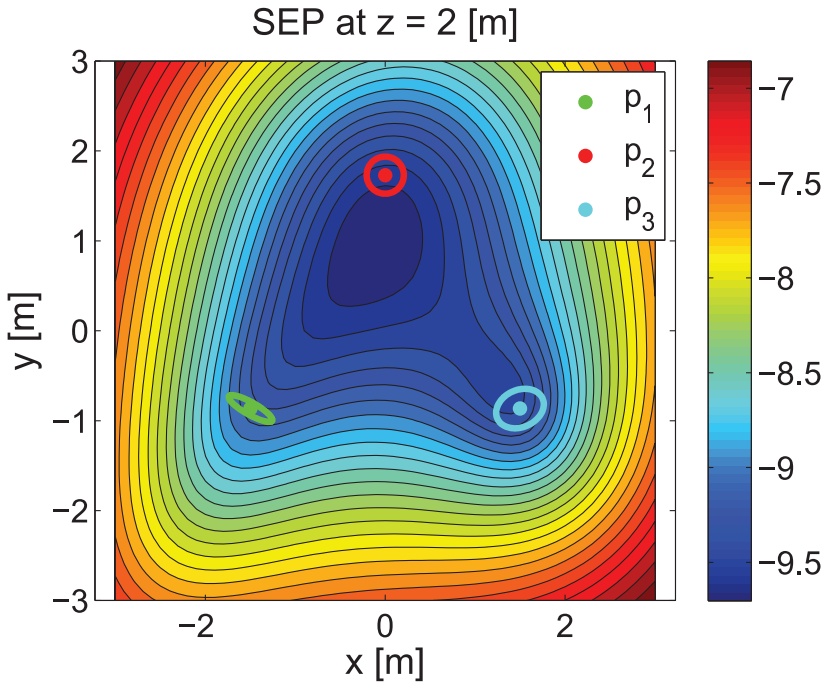
A plot for DOP and the SEP is shown in Fig. 6.5. The results are given in logarithmic scale. Fig. 6.5(b) clearly shows the effects of the propagation of the error from OWMR positions and distance measurements to the position $\mathbf{p}_u$. SEP has a more accurate description of the error in trilateration than DOP. DOP takes in account that the stations are fixed in a given position. If they were to move, then there is an error associated with the dynamics of the motion. The plots are not interchangeable, e.g. a value of 1 in DOP is not the same as a value of 1 in SEP.

[1] A covariance matrix $\mathbf{Q}$ is rotated as $\mathbf{RQR}^\intercal$ where $\mathbf{R}$ is a rotation matrix

[2] The covariances are drawn using an eigen value decomposition, where the eigen vectors denote the axis of the ellipses and ellipsoids, and the eigen values denote how much it stretches in a certain direction.



(a) Dilution of precision.



(b) Spherical error probable.

FIGURE 6.5: DOP and SEP. SEP has a more accurate description of the error in trilateration than DOP because it takes in account the off diagonal elements in the covariance. The plots are not interchangeable, e.g. a value of 1 in DOP is not the same as a value of 1 in SEP. $\mathbf{\Sigma}_{p1} = [0.08^2, 0.02^2, 0]^\intercal @ 75°$, $\mathbf{\Sigma}_{p2} = [0.06^2, 0.06^2, 0]^\intercal @ 120°$, $\mathbf{\Sigma}_{p3} = [0.08^2, 0.06^2, 0]^\intercal @ 30°$ and $\mathbf{\Sigma}_l = [0.025, 0.045, 0.02]^\intercal$.

6.3 Conclusions

In this chapter is was explained how the error propagates from the OWMR and range measurements to the UAV position. It was explained how the error dilutes as the geometry in the problem changes. The mathematical steps to derive of the error propagation was executed and demonstrated with numerical simulations. Finally it was defined how to measure the propagation error, with two measurements studied: DOP and SEP. It was noticed that the SEP is better to measure a covariance than the DOP as it takes in account all the values in the covariance matrix of the propagation. The error propagation is very important for mobile robots as the uncertainty changes with the motion. It cannot be neglected that if a formation of OWMR move, their uncertainty will have an effect on robots or elements that use this information.

7 Solvability due to Uncertainty in Measurements

In real scenarios the measurements $\mathbf{p}_i$ and l_{iu} are noisy. As stated in Section 3.3, when the UAV is close to the base plane or away from the OWMR formation it may happen that the intersection of the spheres does not happen and so, the trilateration can not be computed[1]. In Chapter 3, Eq. (3.14) states that three distances a_{ij} must exist, if one of them is less than zero, that configuration is considered unsolvable, mathematically:

$$\text{SM}_{abc}(\mathbf{p}_u) = \mathbf{p}_u \exists \iff a_{ij} \in \mathbb{R}_0^+, \qquad \{i,j\} \in \{a,b,c\}, i \neq j \tag{7.1}$$

where $\exists$ is the exist operator and $\mathbb{R}_0^+$ is the set of all real numbers greater or equal than zero and $\text{SM}_{abc}(\mathbf{p}_u)$ is the probability that the point $\mathbf{p}_u$ is solvable for f_{abc}. For any position $\mathbf{p}_u$, the solvability is the percentage for a solution in trilateration.

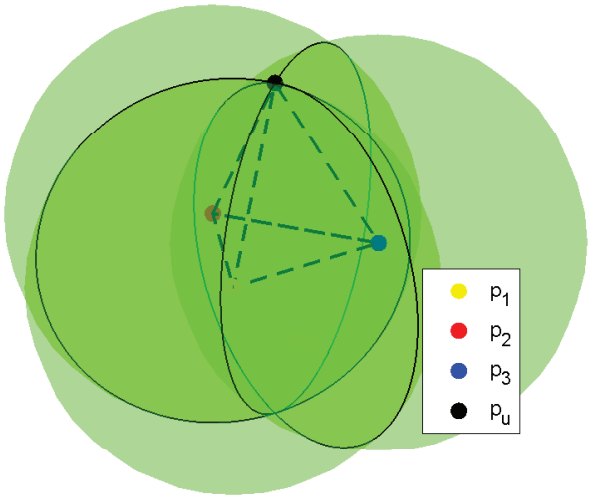
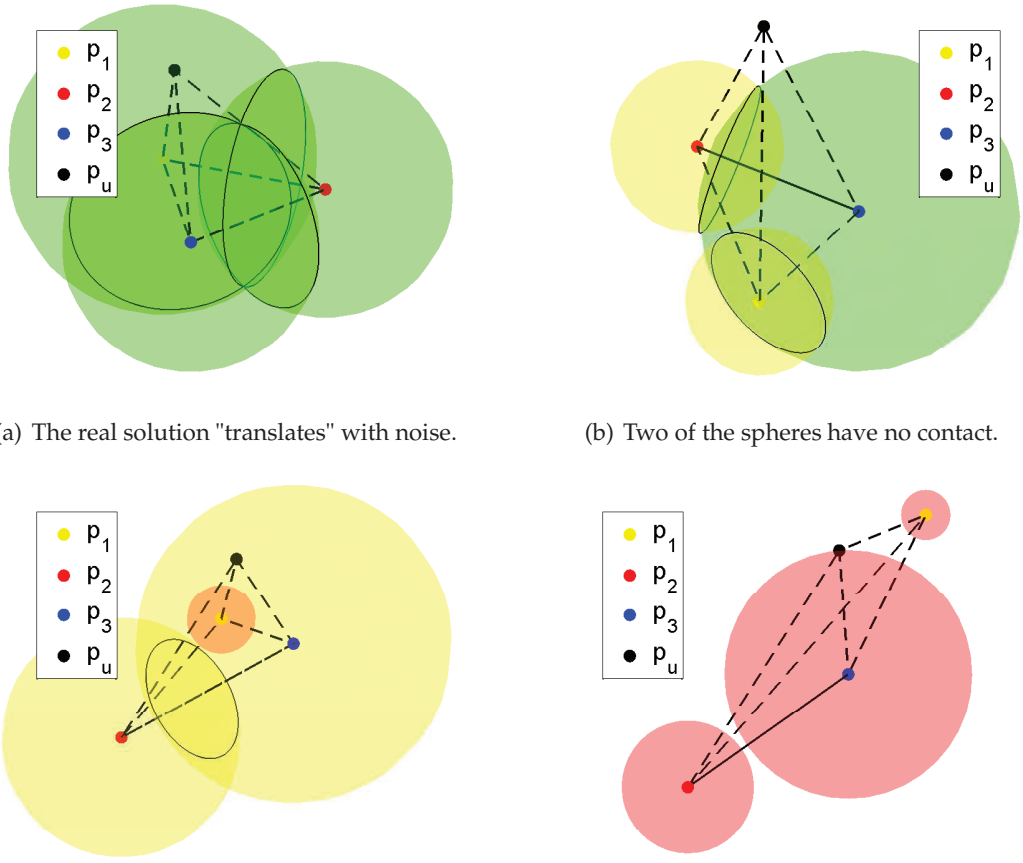


FIGURE 7.1: $\mathbf{p}_u$ solution in a noiseless environment. Without noise the solution to $\mathbf{p}_u$ is exactly at the intersection of three spheres, given the known positions $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ and their corresponding range measurements l_{iu}.

[1]Mathematically, (3.16) has a solution but physically it does not.

This percentage can be calculated by means of the Probability Density Function (PDF) and the Cumulative Density Function (CDF) for the UAV position $\mathbf{p}_u$. If three points, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ exist in a noiseless environment, the solution $\mathbf{p}_u$ lies at the intersection of three spheres with radiuses $l_i u$, as depicted in Fig. 7.1. On the other hand, if the measurements for positions and ranges are disturbed by noise the solution cannot be found. Fig. 7.2 shows some examples of cases where the intersection is affected by noise, notice that the errors are exaggerated so the concept can be clearly illustrated. Green means contact with two spheres. Yellow means contact with just one sphere. Red means no contact at all.



(a) The real solution "translates" with noise.

(b) Two of the spheres have no contact.

(c) One sphere is completely inside another.

(d) No contact at all.

FIGURE 7.2: Sphere intersection affected by noise. Some examples of cases where the intersection is affected by noise. Notice that the errors are exaggerated so the concept can be clearly illustrated. Green means contact with two spheres. Yellow means contact with just one sphere. Red means no contact at all.

In Fig. 7.3 an example where noise does not affect the outcome is explained. In the upper part the measurements are noiseless, so the intersection lies exactly at $\mathbf{p}_u$, in the green spot. In the lower part, even do the measurements are corrupted by noise there is still an intersection, and thus, a solution for the localization.

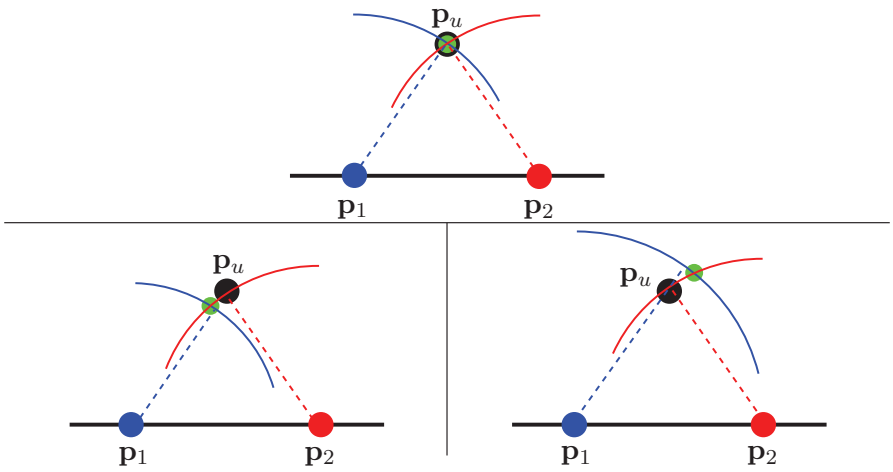


FIGURE 7.3: Intersection explained, case A. In the upper part the measurements are noiseless and a solution lies at $\mathbf{p}_u$, in the green spot. In the lower part, even do the measurements are corrupted by noise there is an intersection.

For a different configuration, a solution may not be feasible, as shown in Fig. 7.4.

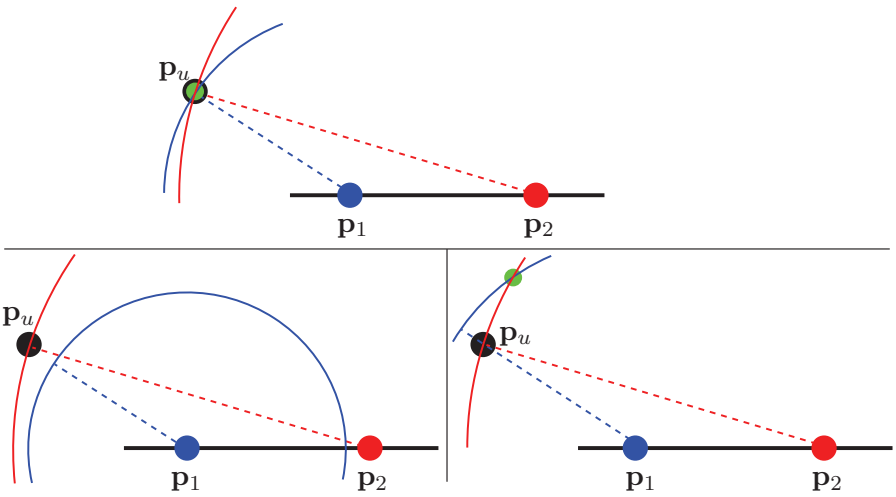


FIGURE 7.4: Intersection explained, case B. For a different noiseless configuration, there is a solution, see upper graph, green spot. With noise, the measurements change and a solution may not be feasible, as shown in the lower left graph.

7.1 Solvability using Montecarlo approximation

A Montecarlo approach can be used to calculate the PDF and the CDF for $\mathbf{p}_u$, e.g. if n_{iter} iterations are taken with noisy measurements, $\text{SM}_{abc}(\mathbf{p}_u)$ can be computed as:

$$\text{SM}_{abc}(\mathbf{p}_u) = \frac{\sum_{n=1}^{n_{iter}} \mathbf{p}_u \exists \iff a_{ij} + \epsilon \in \mathbb{R}_0^+}{n_{iter}}, i \in \{a,b,c\}, j \in \{a,b,c\}, i \neq j, \epsilon\, \mathcal{N}(0,\sigma) \quad (7.2)$$

where ϵ is noise represented by a normal Gaussian distribution with zero mean and variance σ. For this method to work and give accurate results, n_{iter} has to be as high as a thousand iterations.

In Fig. 7.5 the PDF and CDF of (7.2) are shown for a OWMR formation centered at the origin with the UAV above them at 1.25 meters. The standards deviation in positions $\mathbf{p}_i$ and distances l_{iu} is 0.1 m and 0.1 m, respectively. As it can be seen, in this case $\text{SM}_{abc}(\mathbf{p}_u) = 1$ at all times, which means that the solvability is guaranteed for this configuration.

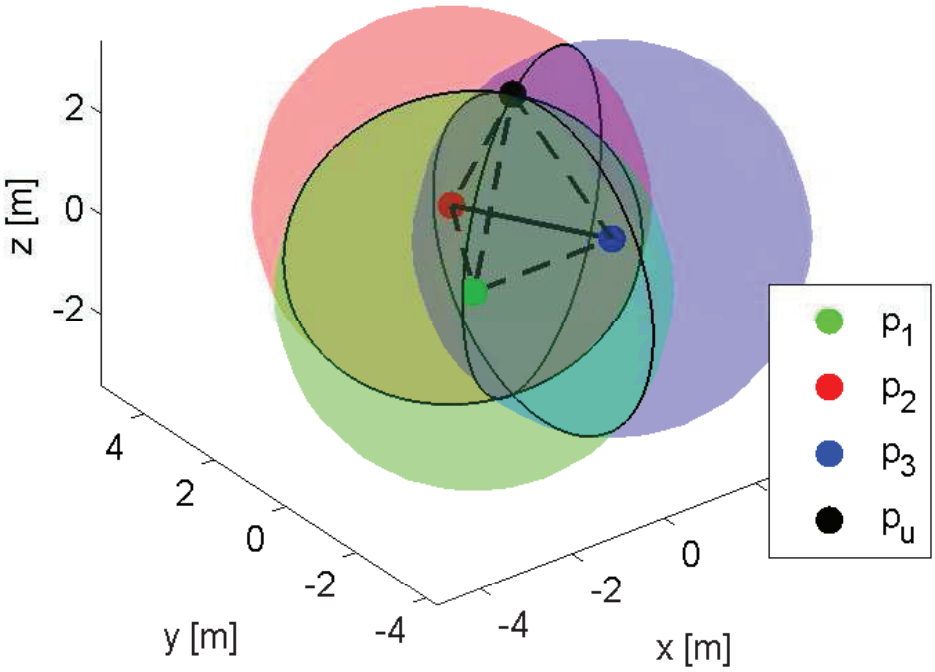
If $\mathbf{p}_u$ where to move along one of the axis of the base plane A_b then the solvability decreases, as shown in Fig. 7.6. In this case $\text{SM}_{abc}(\mathbf{p}_u) = 0.74$ which means that there is a 26% probability that the trilateration cannot be solved.

Fig. 7.5 and 7.6 show cases for a single $\mathbf{p}_u$ location. As the UAV can be anywhere on the environment it is better to represent the solvability as a map SM instead of a single point. In this case, the SM will represent the probabilities at all positions $\mathbf{p}_u$. Mathematically:

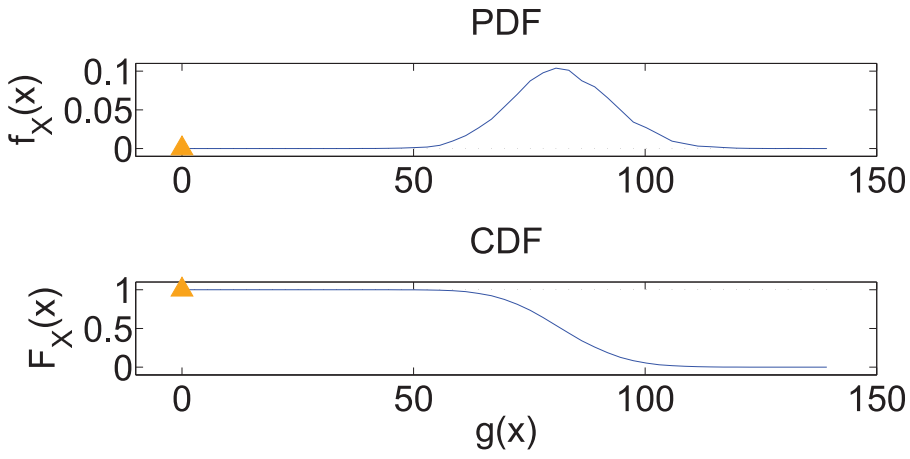
$$\text{SM}_{\forall abc}(\mathbf{p}_u) = \text{SM}_{abc}(\mathbf{p}_u) \forall \{x_u, y_u, z_u\} \in \mathbb{R} \quad (7.3)$$

where $\forall$ is the for all operator.

A SM was built for a OWMR formation centered at the origin with standards deviation in positions $\mathbf{p}_i$ and distances l_{iu} of 0.1 m and 0.1 m, respectively. The UAV is positioned at all possible positions spanning an area of 20 by 20 meters at different heights. The results are shown in Fig. 7.7. In Fig. 7.8 an SM is plotted in three dimensions. The stations are given by their 2D distributions and the shape of the region covered by the SM can be appreciated.

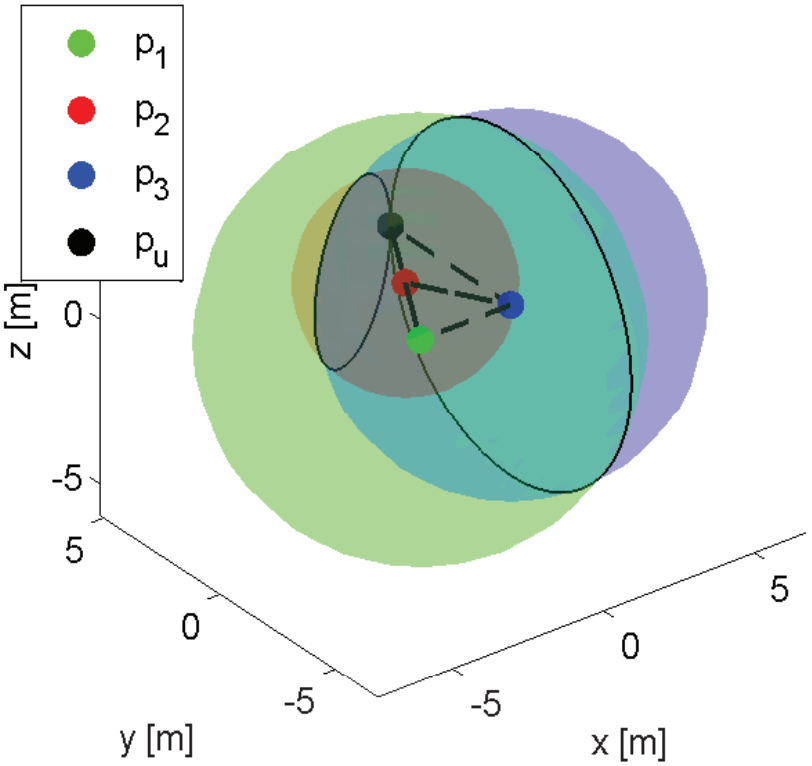


(a) Trilateration problem used for Montecarlo case A.

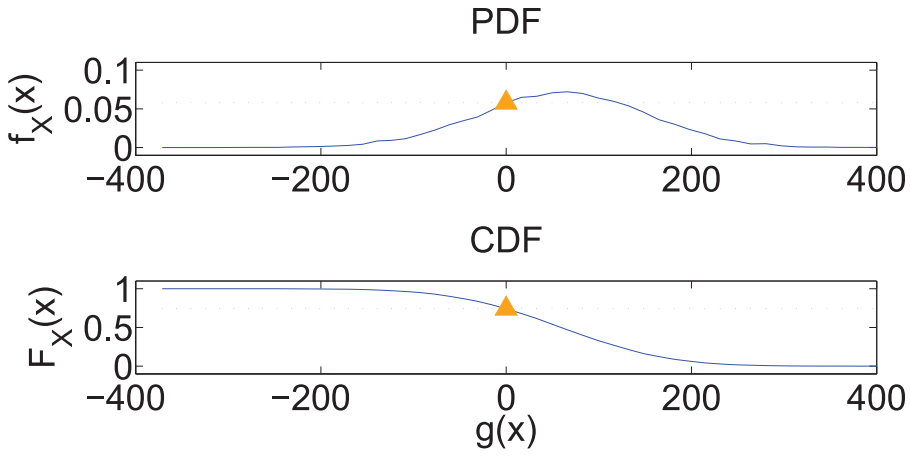


(b) PDF and CDF.

FIGURE 7.5: $\text{SM}_{abc}(\mathbf{p}_u)$ case A. The PDF and CDF are shown for the formation depicted using the Montecarlo method. The CDF is 1, which means that the problem is solvable at all times, and thus, a solution for $\mathbf{p}_u$ is guaranteed.

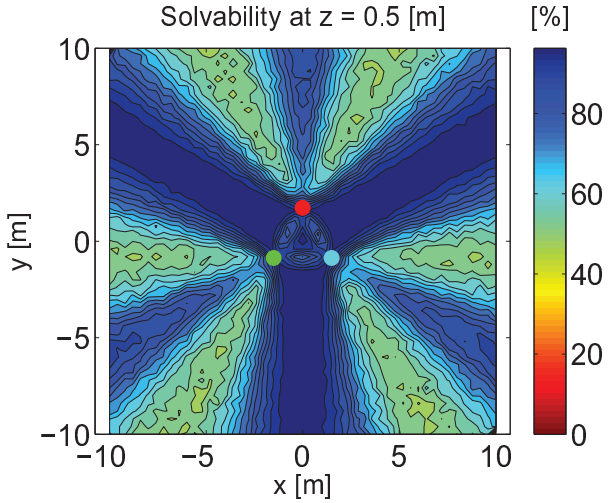


(a) Trilateration problem used for Montecarlo case B.

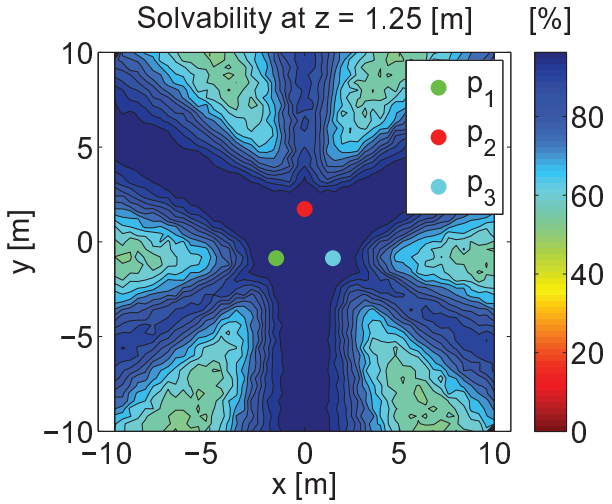


(b) PDF and CDF.

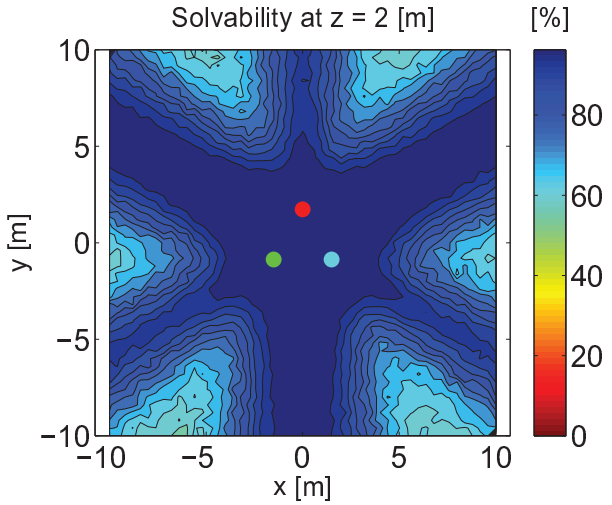
FIGURE 7.6: $SM_{abc}(\mathbf{p}_u)$ case B. The PDF and CDF are shown for the formation depicted using the Montecarlo method. Now the solvability drops to 0.74, meaning that there is 26% of probability that the problem does not have a solution for $\mathbf{p}_u$ in this configuration.



(a) Blue areas are fully solvable.



(b) SM change with height.



(c) Solvability increases with height.

FIGURE 7.7: Solvability maps $\mathrm{SM}_{abc}(\forall \mathbf{p}_u)$. The SM is shown for an area of 20 by 20 meters at heights of 0.5, 1.25 and 2 meters. The random nature of the Montecarlo method can be perceived. As the height increases so does the solvability.

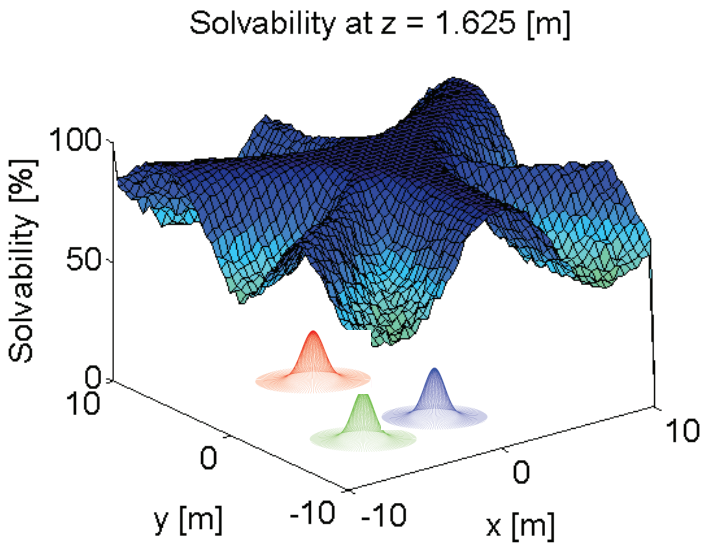


FIGURE 7.8: 3D view of an SM. The stations are given by their 2D distributions and the shape of the region covered by the SM can be appreciated.

As it can be seen, zones along the direction of the axis of the OWMR formation have higher probability for failure, and that is expected since two positions $\mathbf{p}_i$ can easily form a collinear formation with $\mathbf{p}_u$ along these axes. The solvability is also proportional to the height, the higher the bigger probability for a solution to trilateration. Heights less than one meter are not used for long times in real scenarios but they are mention as the solvability can be clearly noticed.

7.2 Solvability using derived probability distribution

The Montecarlo method is not favorable for real time applications, a SM is computationally costly to calculate. Using probability theory [67], [68] a closed form solution will be calculated based on the derived probability of the multivariate PDF. A derived probability distribution maps the PDF of a Random Variable (r.v.) to a function using that r.v.. A quick review of derived probability distributions can be found on Appendix D.

The expression inside the square root of (3.14) has to be positive to guarantee an intersection between two spheres i and j. A function $g(\mathbf{x})$ can be defined as a probability function to represent the intersection of two spheres:

$$g(r, R, d) = g(\mathbf{x}) = (-d + r - R)(-d - r + R)(-d + r + R)(d + r + R), g(\mathbf{x}) \in \mathbb{R}_0^+ \qquad (7.4)$$

For the purpose of the SM, the real variables r, R and d will be defined as values taken from independent r.v. $\mathbf{r}$, $\mathbf{R}$ and $\mathbf{d}$, that is, they are related using a probability as:

$$\mathrm{P}(\mathbf{r} = r) \tag{7.5}$$

$$\mathrm{P}(\mathbf{R} = R) \tag{7.6}$$

$$\mathrm{P}(\mathbf{d} = d) \tag{7.7}$$

The triplet of variables in (3.14) form a random vector $\mathbf{x}$:

$$\mathbf{x} = [r, R, d]^\intercal \tag{7.8}$$

The following notation will come in useful:

$$g(r, R, d)_{ij} = g(r_i, R_j, d_{ij}) = g(\mathbf{x})_{ij} \tag{7.9}$$

The PDF of $\mathbf{r}$, $\mathbf{R}$ and $\mathbf{d}$ is a probability function given by a multivariate normal Gaussian distribution:

$$f_X(r, R, d) = f_X(\mathbf{x}) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\intercal \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})) \tag{7.10}$$

where $\boldsymbol{\mu}$ is the mean and Σ the covariance:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_r & \mu_R & \mu_d \end{bmatrix}^\intercal \tag{7.11}$$

$$\Sigma = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_R^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix} \tag{7.12}$$

The derived probability will map the distributions of $\mathbf{r}$, $\mathbf{R}$ and $\mathbf{d}$ to the distribution of $g(r, R, d)$, expressed as:

$$f_X(g(\mathbf{x})) = \sum_{n=1}^{4} \frac{f_X(\mathbf{x}_n)}{|\det(J(\mathbf{x}_n))|}, \qquad r \geq R \tag{7.13}$$

where $\mathbf{x}_n$ are the four roots of $g(\mathbf{x})$ given by:

$$\mathbf{x}_n = \begin{bmatrix} r_n & R & d \end{bmatrix}^\intercal, \qquad n = 1,2,3,4 \tag{7.14}$$

$$\begin{bmatrix} r_{1,2} \\ r_{3,4} \end{bmatrix} = \begin{bmatrix} \pm((4R^2d^2 - g(\mathbf{x}))^{\frac{1}{2}} + R^2 + d^2)^{\frac{1}{2}} \\ \pm(R^2 - (4R^2d^2 - g(\mathbf{x}))^{\frac{1}{2}} + d^2)^{\frac{1}{2}} \end{bmatrix} \tag{7.15}$$

and $J(\mathbf{x})$ is the Jacobian of $g(\mathbf{x})$. As $g(\mathbf{x})$ is a single function, the denominator of (7.13) is calculated using (7.16) and (7.17) setting R and d as auxiliary variables:

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial g(\mathbf{x})}{\partial r} & \frac{\partial g(\mathbf{x})}{\partial R} & \frac{\partial g(\mathbf{x})}{\partial d} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.16}$$

$$\det(J(\mathbf{x})) = 4R^2r + 4d^2r - 4r^3 \tag{7.17}$$

Finally, if $r < R$, then r and R have to be swapped to cover all the range of the r.v..

The probability of the intersection of two spheres is given by the CDF of (7.13). This CDF cannot be expressed in closed form because it requires to solve to a triple integral which does not have an exact solution. This integral can be inferred numerically as the cumulative sum of the PDF. First a PDF must be built, and a single loop with 50 iteration is enough to get an acceptable result for the CDF.

$$F_X(g(\mathbf{x})) = 1 - \sum_{n=1}^{50} f_X(g(\mathbf{x})) \tag{7.18}$$

Notice that the CDF has to be inverted to get the real value for the solvability. A non inverted probability is $\mathrm{P}(\mathbf{x} \leq x)$. The process has to be calculated three times for any three OWMR a, b and c. So the probability at each position $\mathbf{p}_u$ is calculated using (7.18) as (7.19). That means that if any two spheres do not intersect, then the trilateration does not have a solution.

$$\mathrm{SM}_{abc}(\mathbf{p}_u) = \operatorname{argmin} \begin{bmatrix} F_X(g(\mathbf{x})_{ab}) \\ F_X(g(\mathbf{x})_{ac}) \\ F_X(g(\mathbf{x})_{bc}) \end{bmatrix} \tag{7.19}$$

The process to calculate a sphere intersection is presented in Algorithm 5. Here, n_{PDF} is the number of points in the PDF. The vectors $\mathbf{r}$, $\mathbf{R}$ and $\mathbf{d}$ are monotonic vectors centered at a value x of the form $x - 3\sigma_x$ to $x + 3\sigma_x$.

Algorithm 5 Probability sphere intersection.

```
1: procedure ProbabilitySphereIntersection(r,R,d)
2:     if r < R then
3:         swap(r,R)
4:     end if
5:     MakeVector(r,R,d)
6:     g(r, R, d) ← (7.4)
7:     for i = 1,n_PDF do
8:         f_X ← (7.13)
9:     end for
10:    g_0 ← Find(g(x) ≥ 0)
11:    for i = 1,g_0 do
12:        F_X ← F_X + f_X
13:    end for
14:    return 1 − F_X
15: end procedure
```

The Montecarlo method requires a large number of iterations to give a good result. Two a Montecarlo runs are shown with 50 and 1000 iterations. As it can be seen, a large number of iterations is required to converge to an acceptable result.

Fig. 7.9 uses the Montecarlo method with 50 iterations, it is clearly visible that the regions did not reach a final value.

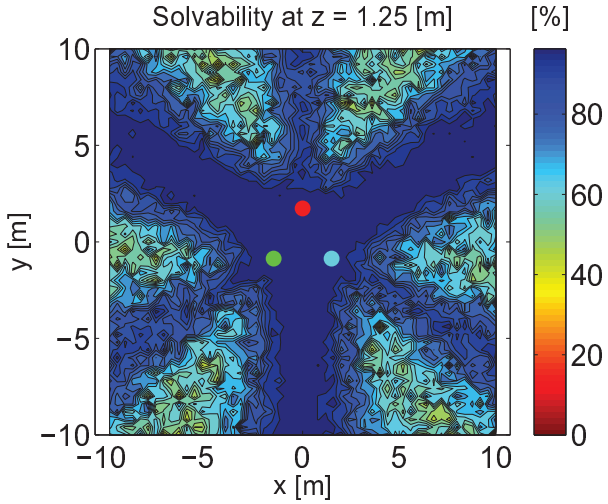


FIGURE 7.9: Solvability map using Montecarlo with 50 iterations 50 iterations are not enough to get a good result for the solvability

A good approximation is achieved with 1000 iterations (Fig. 7.10), although it is very time consuming.

Finally, in Fig. 7.11 the SM using the closed form formula is shown. The closed form formula gives an exact result and is computationally fast compared to the Montecarlo method.

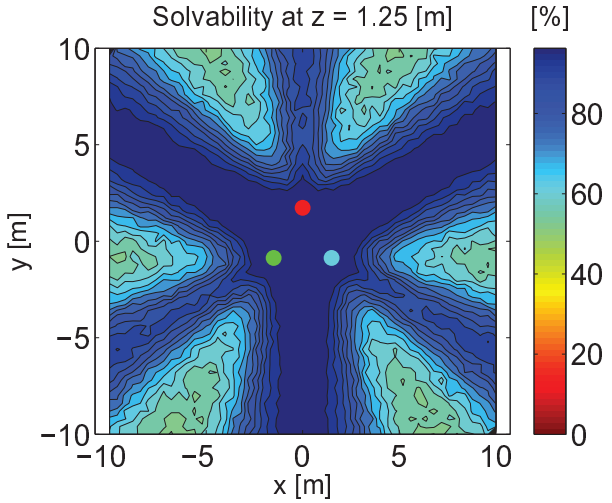


FIGURE 7.10: Solvability map using Montecarlo with 1000 iterations. 1000 iterations need a big amount of time to get the SM done.

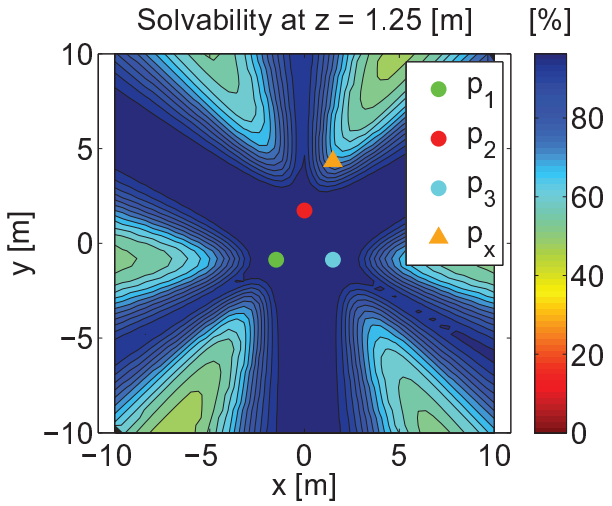


FIGURE 7.11: Solvability map using closed form formula. The closed form formula gives an exact result and is computationally faster compared to the Montecarlo method.

In Fig. 7.11, a point $\mathbf{p}_x$ is shown, for this point its PDF and CDF are shown in Fig. 7.12. It can be seen that the closed formula gives a better answer than the Montecarlo method.

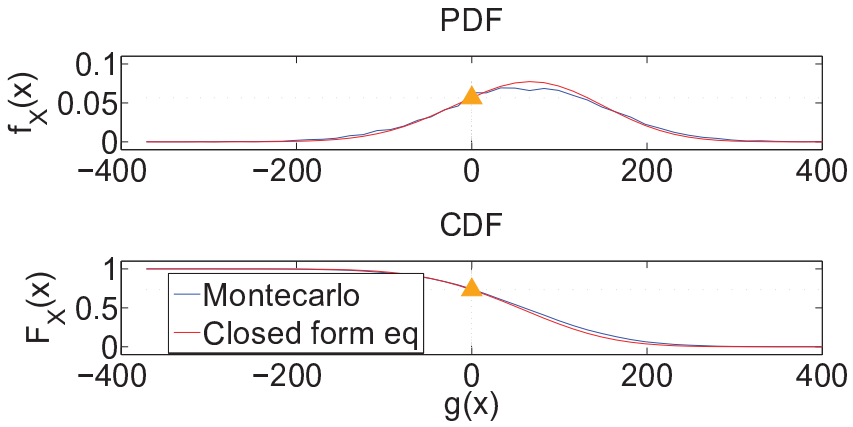


FIGURE 7.12: PDF and CDF of $g(\mathbf{x})$ at a point $\mathbf{p}_x$. The PDF and CDF are shown for a point within an SM to show the differences between the Montecarlo and the closed formula methods.

In Table 7.1 a comparison of the execution times between the Montecarlo method and the closed form formula is shown. The times presented are the mean of an area of 51 by 51 $\mathbf{p}_u$ different locations, that is, the mean of 2601 positions $\mathbf{p}_u$. It can be seen that the closed form formula is extremely fast compared to the Montecarlo method.

TABLE 7.1: Solvability execution time comparison. Comparison of execution times between the Montecarlo method and the closed form formula.

Method Time [s]	Montecarlo 50 iterations	Montecarlo 1000 iterations	Closed form formula
Run 1	0.0013	0.0237	2.0659e-5
Run 2	0.0011	0.0273	2.1434e-5
Run 3	0.0011	0.0250	2.6701e-5
Run 4	0.0012	0.0261	2.1640e-5
Run 5	0.0011	0.0243	2.2138e-5
Mean **2601 positions $\mathbf{p}_u$**	**0.00116**	**0.02528**	**2.25144e-5**

The solvable areas depend on the shape of the OWMR formation, so if the formation changes, so does the solvability. The formation can be changed accordingly to the requirements of the UAV or the environment the robots are located, e.g. a corridor. Two slim formations were chosen, see Fig. 7.13 and Fig. 7.14, as in indoor scenarios corridors will are likely to exist in great part of the environment.

The multi robot configuration can be exploited when more than three OWMR are available. For that it is needed to take combinations of three robots ($\wp(C_3^n)$) at a time to perform the trilateration. The solvable zones of each combination can be superimposed and then areas

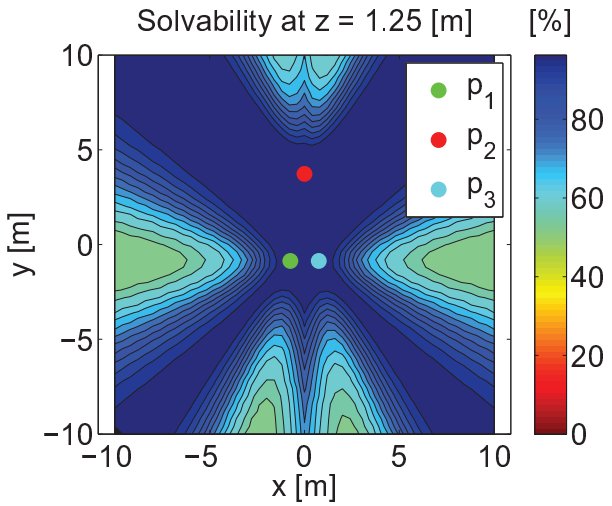


FIGURE 7.13: Solvability map for “Vertical" formation. As the indoor scenario changes the formation can be changed accordingly to the requirements of the environment. Slim formations are useful in e.g. corridors.

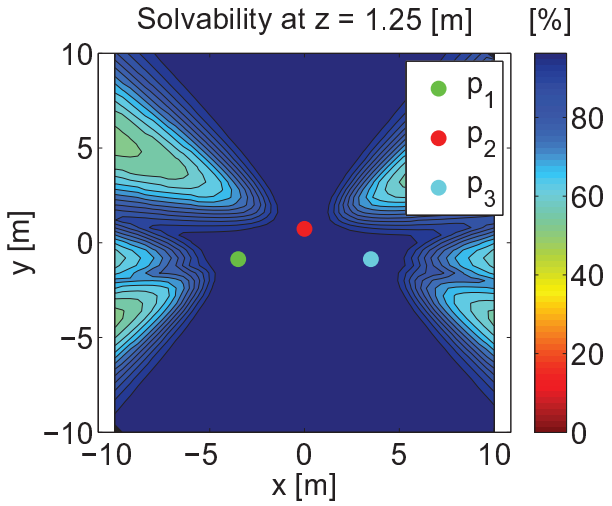


FIGURE 7.14: Solvability map for slim “Horizontal" formation. As the indoor scenario changes the formation can be changed accordingly to the requirements of the environment. Slim formations are useful in e.g. corridors.

without a solution using a selection of robots may have a solution with a different selection. Each combination has a SM, and these maps can be added to increase the solvability in a certain area. SM maps are added as:

$$\mathrm{SM}_{abc}(\forall \mathbf{p}_u) = \mathrm{argmax}(\forall \mathrm{SM}_{abc}(\forall \mathbf{p}_u)), \{a, b, c\} \in \wp(C_3^n) \tag{7.20}$$

In (7.19) the minimum value is taken because if the intersection of two spheres is not viable then the trilateration cannot be computed. On the other hand, in (7.20), the maximum value is taken because it is only needed that one formation f_{abc} can compute the position $\mathbf{p}_u$.

In Fig. 7.15 two mirrored formations are added. A map $\mathrm{SM}_{123}(\forall \mathbf{p}_u)$ is added to the formation with map $\mathrm{SM}_{134}(\forall \mathbf{p}_u)$.

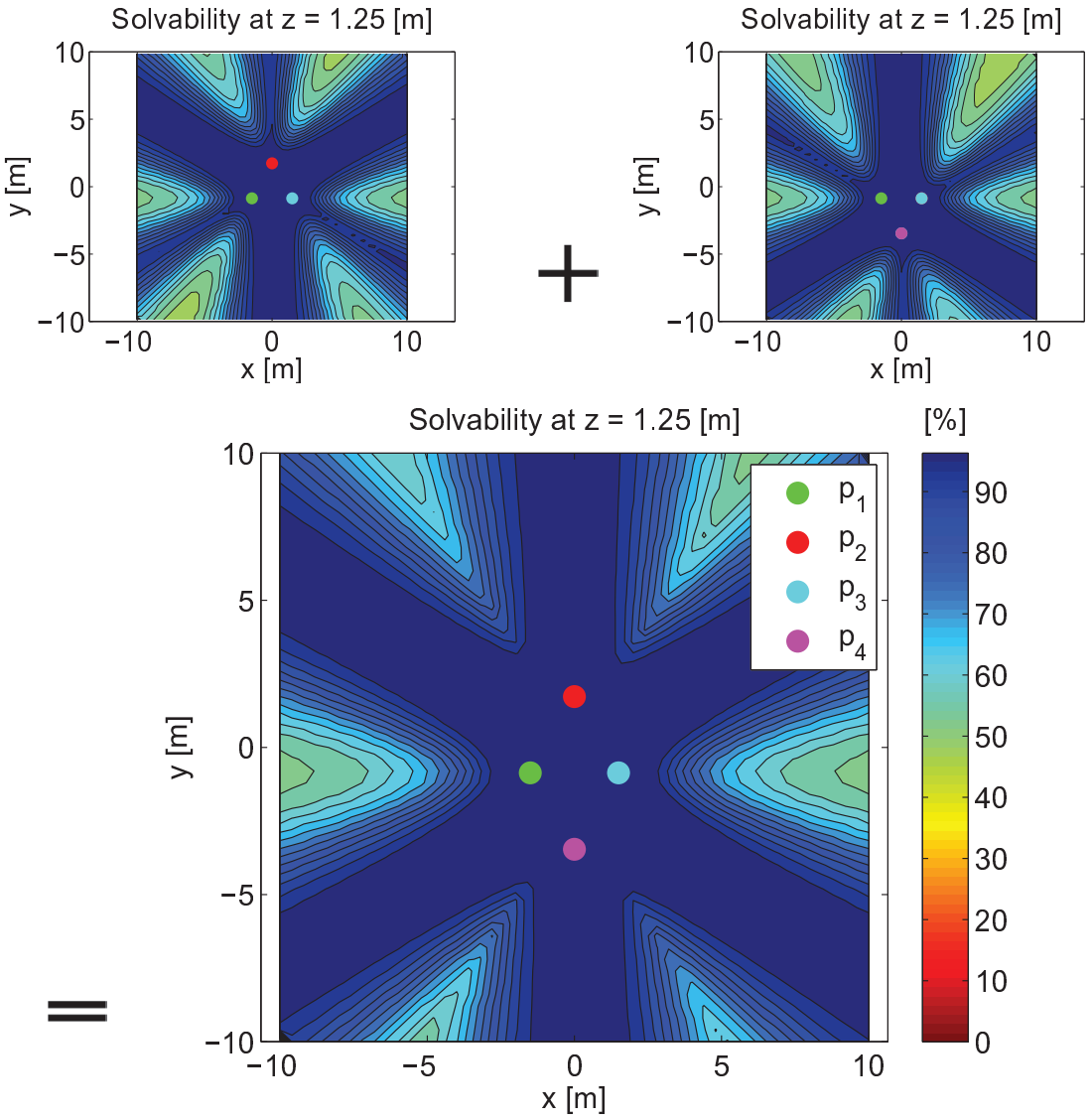


FIGURE 7.15: Addition of two SM maps. Two mirrored formations are added, its SM maps complement each other increasing the solvability and guaranteeing the localization execution.

A third formation $SM_{235}(\forall \mathbf{p}_u)$ can be added to the result in Fig. 7.15, it is shown in Fig. 7.16. Just by adding a third formation the solvability greatly increases. With this it is confirmed that by incrementing the number of robots used for localization better results are to be expected.

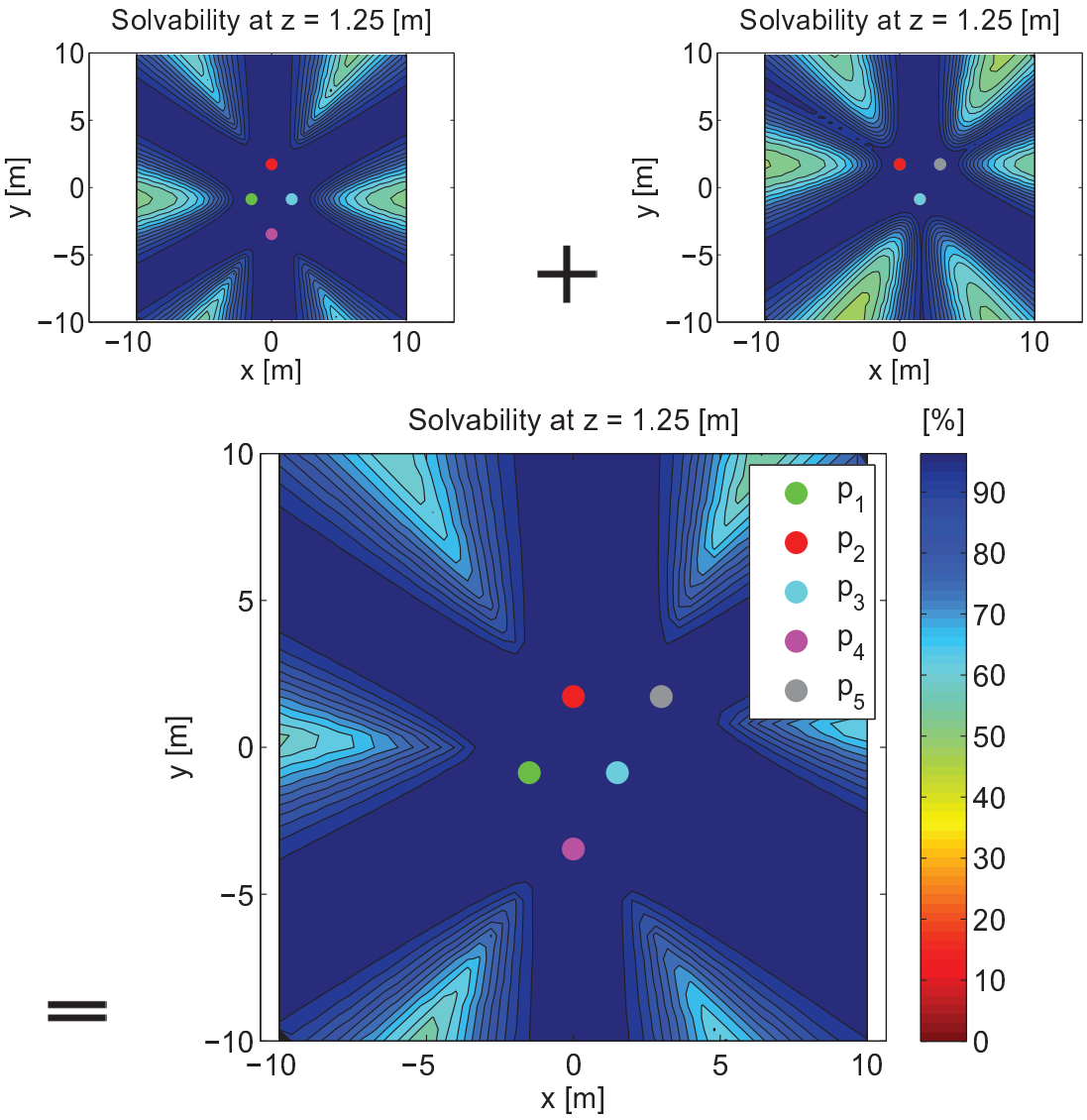
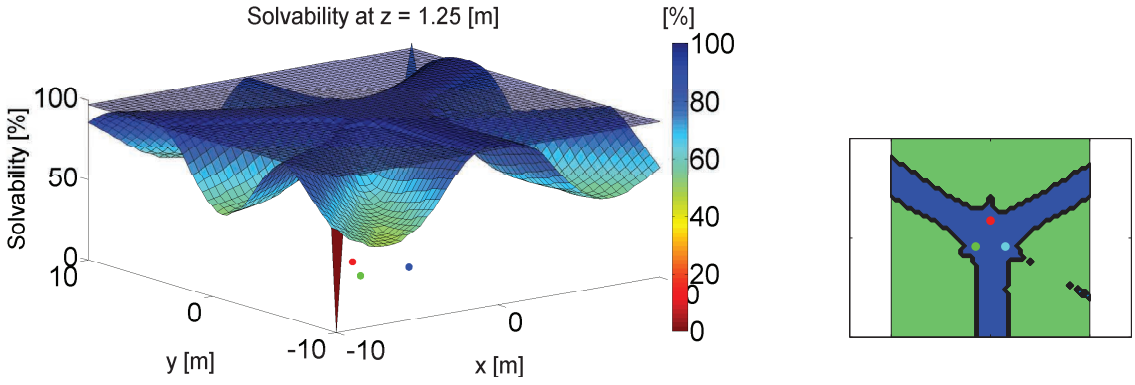


FIGURE 7.16: Addition of three SM maps. As many combination of OWMR can be added as long as the localization algorithm can be executed for the group of robots chosen.

Since an UAV moves in a three dimensional space the solvability can be represented in a three dimensional space. If a threshold SM_{thres} is set, a SM plane can be extracted. Setting the threshold at $SM_{thres} = 97\%$ in Fig. 7.17(a) gives the resulting plot drawn in Fig. 7.17(b). As a result, the SM plane only has two values, solvable and not solvable.

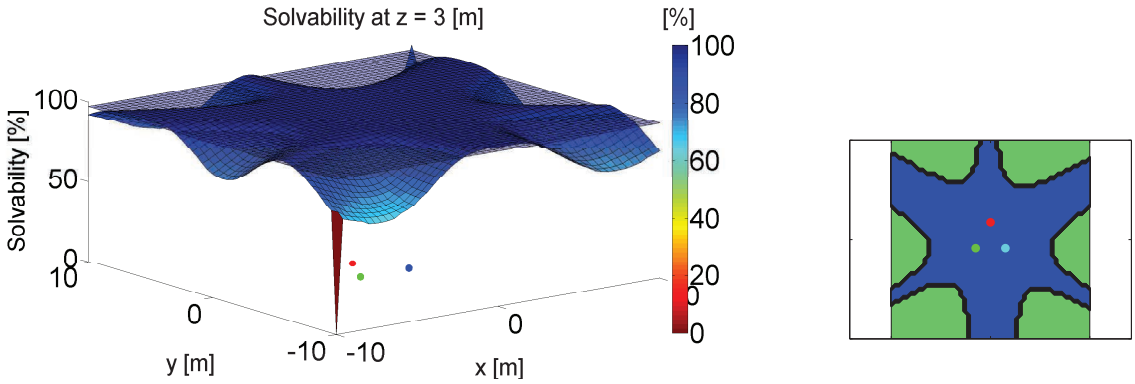


(a) SM plot with threshold plane at 97%.

(b) Threshold plane.

FIGURE 7.17: Solvability at 1.25 meters with $SM_{thres} = 97\%$. The solvability only has two values, 0 and 1 but this allows for a three dimensional SM.

In Fig. 7.18(a) the same approach is carried on but on a map at three meters to show the differences at different heights. The SM plane is presented in Fig. 7.18(b).

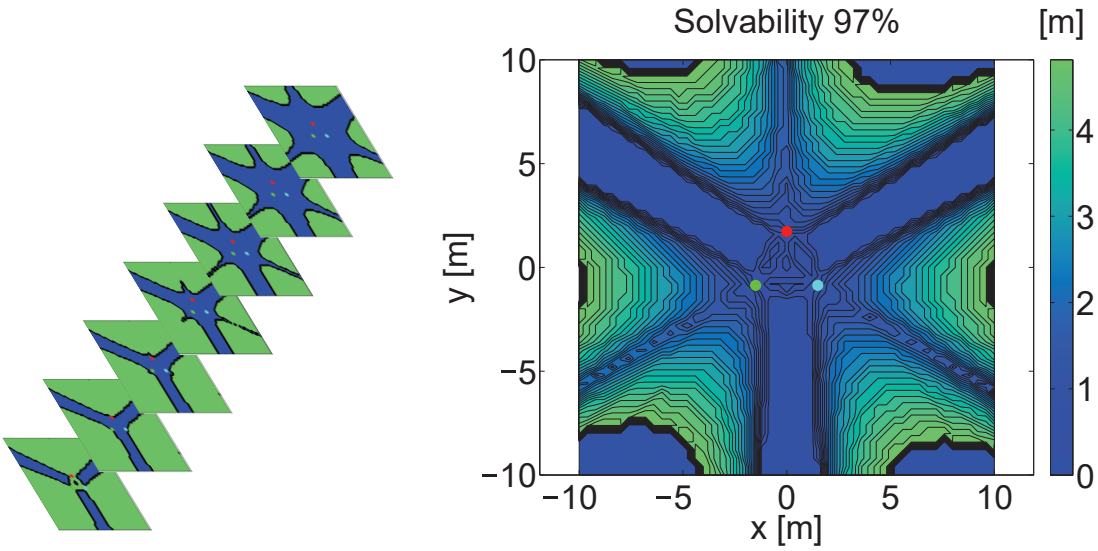


(a) SM plot with threshold plane at 97%.

(b) Threshold plane.

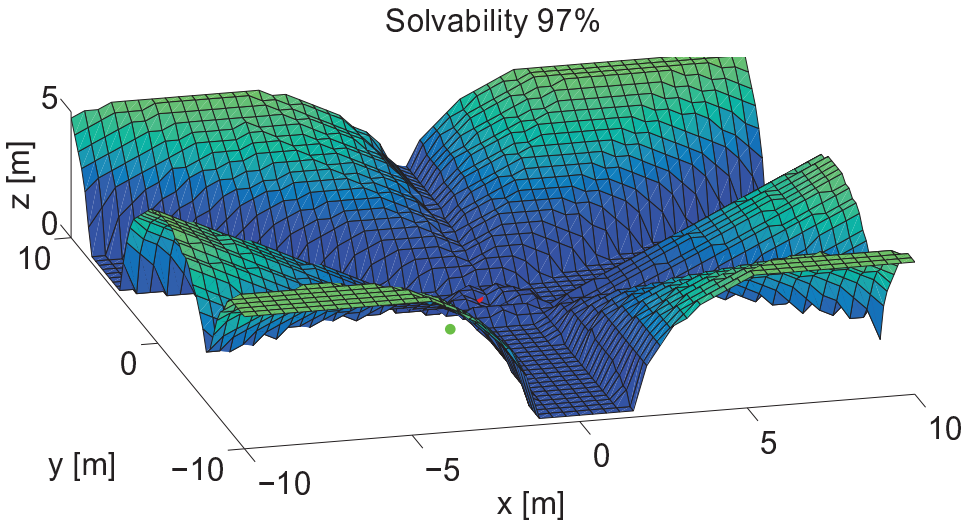
FIGURE 7.18: Solvability at 3 meters with $SM_{thres} = 97\%$. At different heights, the same threshold gives a different solvability.

Stacking several SM planes allows for a three dimensional SM construction. This is presented in Fig. 7.19. where it is shown a three dimensional SM plot from 0 to 5 meters with $SM_{thres} = 97\%$.



(a) SM stacked, 97%.

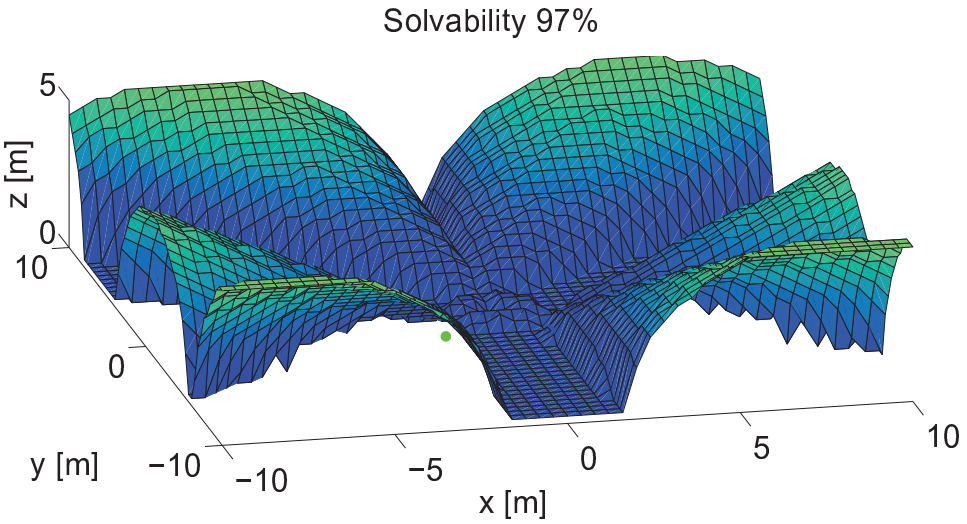
(b) Upper view.



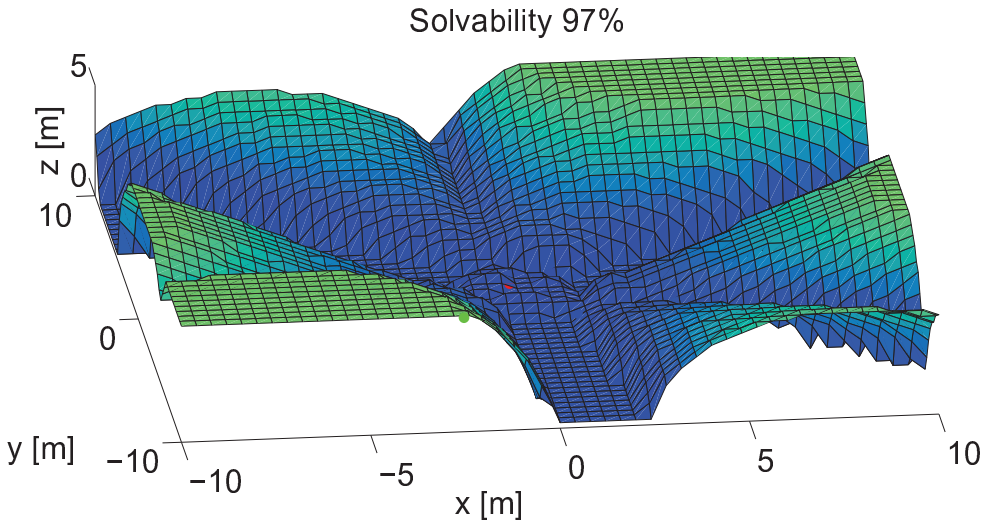
(c) 3D view

FIGURE 7.19: Solvability maps in 3D. If 2D maps are stacked a three dimensional SM can be built. The solvable areas are everywhere above the “mountains".

In a dynamic system the errors may change over time, if this happens the SM also change. If the errors in positions $\mathbf{p}_i$ change the SM do not change too much, see Fig. 7.20.



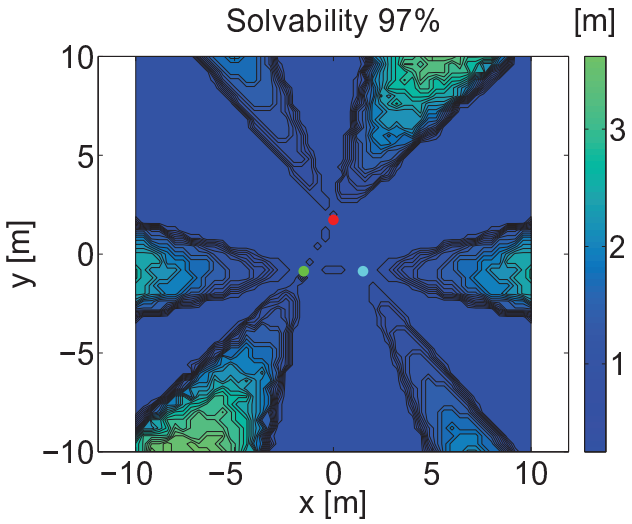
(a) Solvability with $\sigma_p = 0.01$ m.



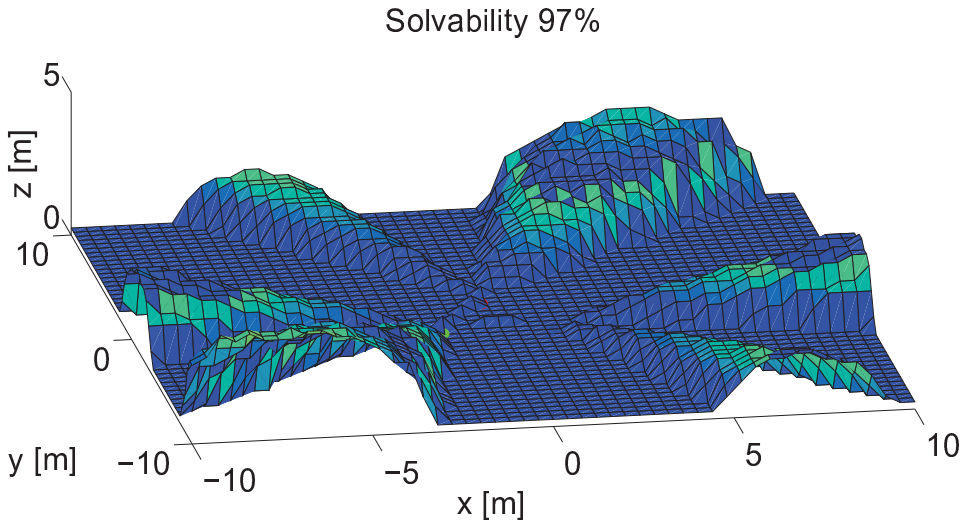
(b) Solvability with $\sigma_p = 0.3$ m.

FIGURE 7.20: SM with different variances in positions $\mathbf{p}_i$. The errors in positions do not affect the solvability as it can be appreciated.

On the other hand, if the errors in distance measurements change, then the solvability is greatly affected. This is presented in Fig. 7.21 and Fig. 7.22.

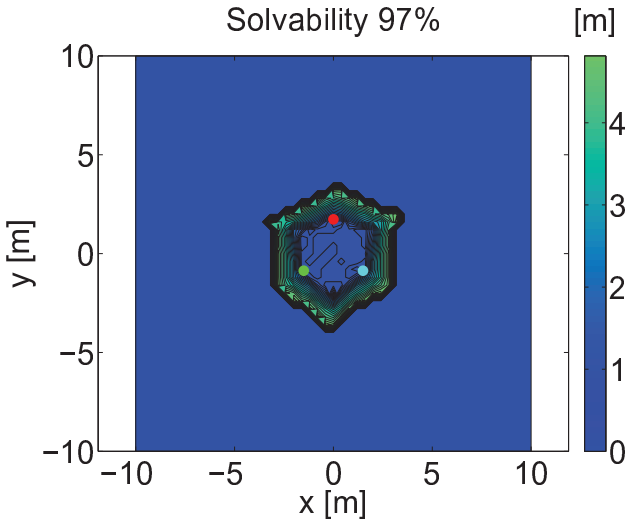


(a) 2D view Solvability with $\sigma_l = 0.01$ m.

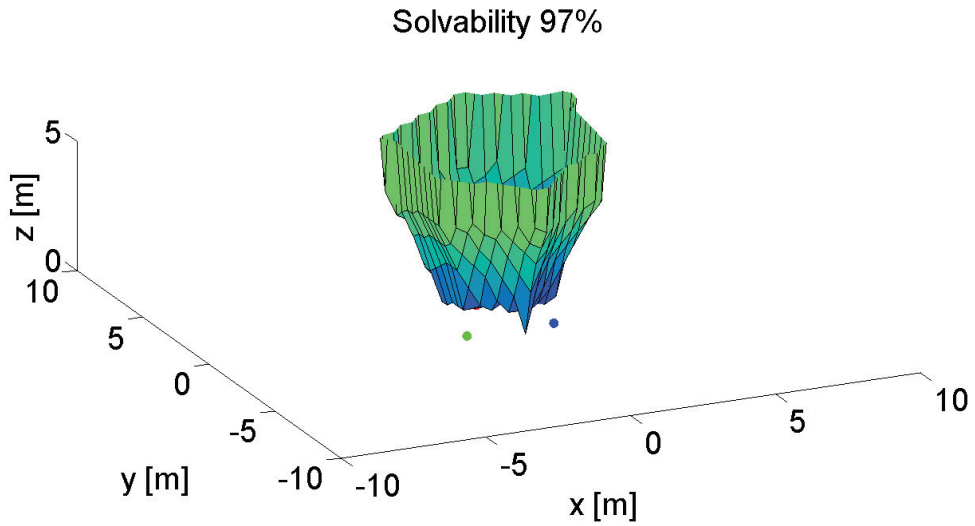


(b) 3D view Solvability with $\sigma_l = 0.01$ m.

FIGURE 7.21: SM with a small variance in distance measurements $\sigma_l = 0.01$. The solvability is greatly increased. The non solvable areas are now at a lower altitude.



(a) 2D view Solvability with $\sigma_l = 0.3$ m.



(b) 3D view Solvability with $\sigma_l = 0.3$ m.

FIGURE 7.22: SM with a big variance in distance measurements $\sigma_l = 0.3$. The solvability is reduced as only a small region can be calculated.

7.3 Conclusions

The concept of solvability map SM was introduced in the presented chapter. It was shown that the regions were the trilateration problem is computable can be known. This regions lay along the axis of the triangle formed by the base plane A_b. The solvability can be used to select the shape of the multi robot formation or to design paths for the tracked object. The SM was introduced first as a conceptual method using the Montecarlo approach and then, using a derived probability function is was shown that it can be calculated with a closed form formula with faster and more accurate results, which can be applied in real time systems.

8 Formation Selection and Control

For multi robot formations, it is needed to define how robots will behave within the group and which robots will perform certain actions. In this chapter it is explained how the multi robot formation behaves under the localization algorithm proposed. First, knowing that the trilateration algorithm needs three points working as stations it will be defined how this triad of robots is chosen, then a simple formation control is explained to drive the robots motion.

8.1 Triad representation for function f_{abc}

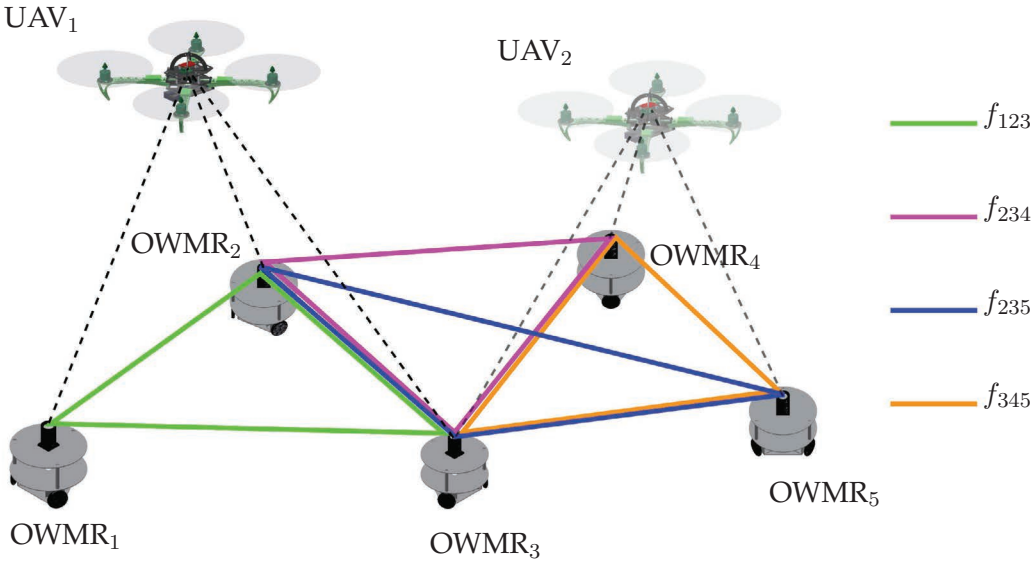


FIGURE 8.1: Example of different selections of formation f_{abc}. Different formation selection are shown when having five robots available to perform the localization, any triad f_{abc} can perform the trilateration. At position 1 f_{123} is selected but at position 2 f_{345} is a better option.

In Chapter 3, the function f_{abc} was defined to represent the trilateration, Eq. (3.16), this is useful because it allows to put all the possible combinations in a simple notation, where f_{abc} is any combination of three robots taken out of the formation, a triad, represented as:

$$f_{abc} \in \wp(C_3^n), n = \text{number of robots} \tag{8.1}$$

where n is the number of OWMR available in the multi robot formation, equal or larger than three. As only three robots are required to perform the trilateration, if a formation has, e.g. 5 robots, then $n = 5$ and there are 10 different formations that can be selected with three robots at a time, $C_3^5 = 10$. To illustrate this concept, in Fig. 8.1 some of the formations taken of a group of 10 OWMR are depicted. In the real system, the selection may vary depending on the requirements of the task commanded to the multi robot formation.

8.2 Robot triad selection

The triad of robots used to perform the trilateration can be chosen based on the shortest distances l_{iu}, so the robots a, b and c can be chosen by the a property from the UAV to the OWMR. An algorithm algorithm to select three robots based on the distances measured from all OWMR is a feasible concept that be implemented on the formation.

$$\{a, b, c\} \in \arg\min(l_{iu}), \qquad i \in \{0, \ldots, n\} \tag{8.2}$$

On the other hand, DOP and SM can also be used to define characteristics within the multi robot formation. As an example of an algorithm representing this approach it is proposed to use the DOP as a measure of the proximity of the OWMR to the UAV. As shown in Chapter 6, the DOP changes with the distance from the center of the formation. So, as long as the DOP is less than a threshold value DOP_{thres}, the formation triad will be the same and it will change to the next formation suited for trilateration with a smaller DOP value when the threshold is reached, see Fig. 8.2 for the diagram illustrating this concept.

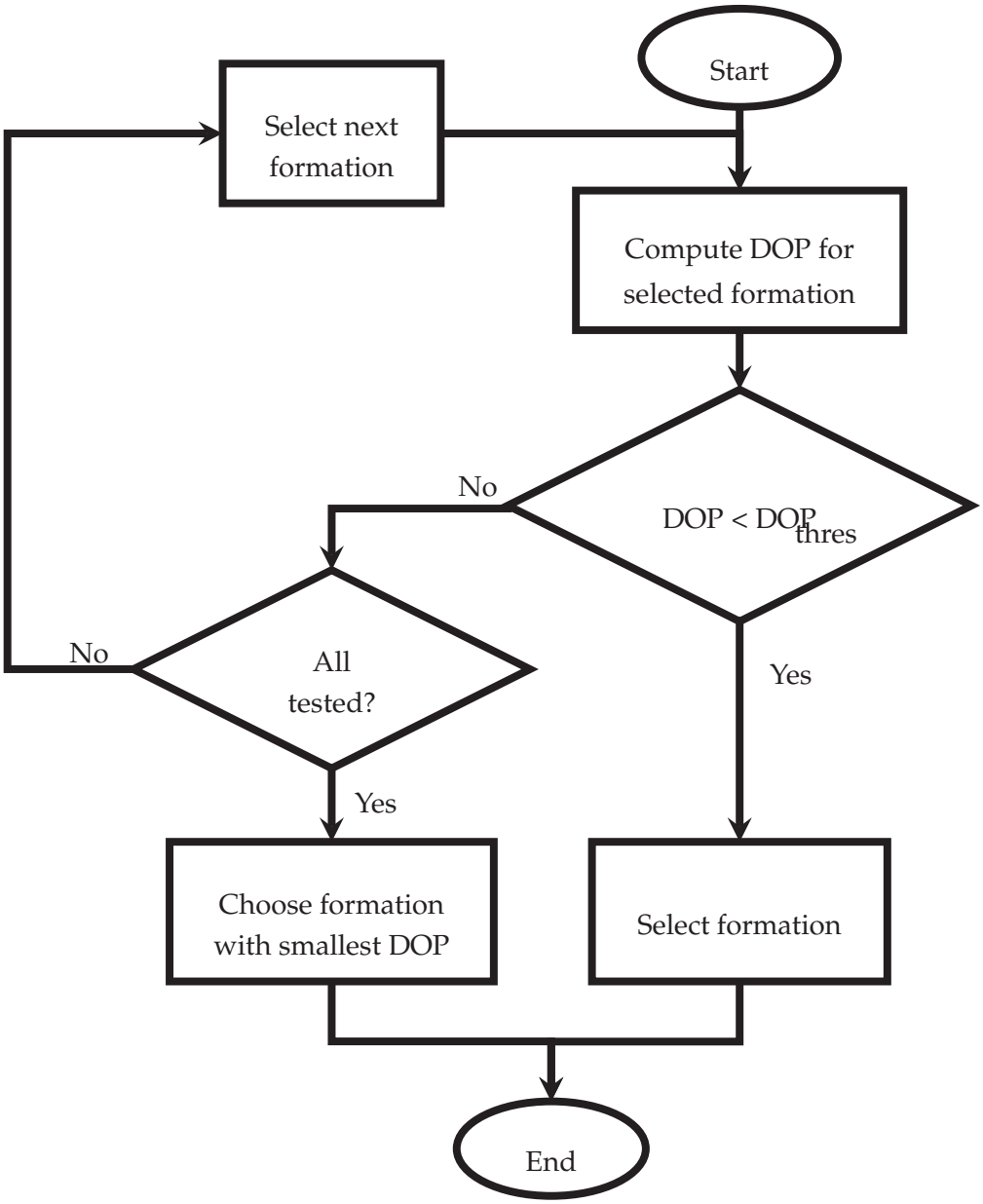
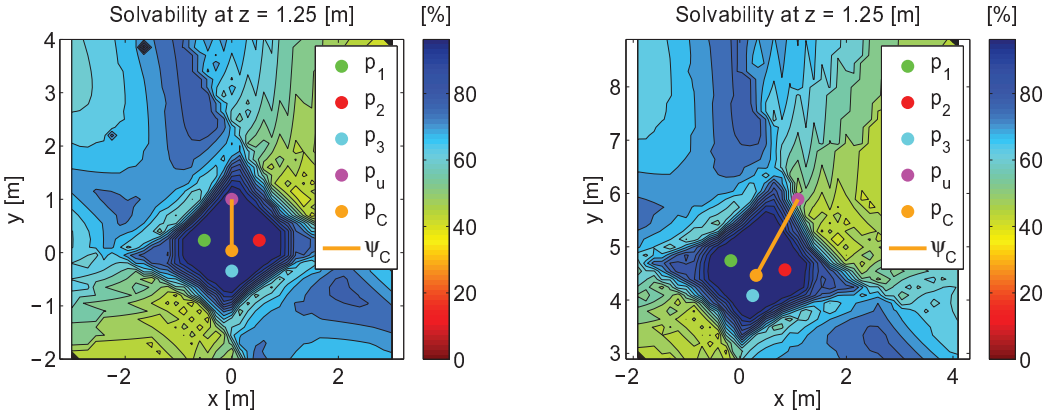


FIGURE 8.2: Triad selection based on DOP. The selection of robots to perform the trilateration is based on the DOP value of the formation f_{abc}. The formation with the smallest DOP value is used in the localization algorithm.

8.3 Formation control

Take Fig. 8.3, the SM can be used to drive the OWMR formation. On the right, the UAV at $\mathbf{p}_u$ is in a good position inside the SM. On the left the UAV enters a non solvable area, and thus, the formation has to move to keep the UAV inside a solvable area. A control $\mathbf{u}_C$ is being applied to the OWMR formation, following the UAV motion. A numerical simulation is performed in Section 9.3.



(a) Solvability at position A.

(b) Solvability at position B.

FIGURE 8.3: SM for formation control. The SM can be used to drive the OWMR formation. On the right, the UAV at $\mathbf{p}_u$ is in a good position inside the SM. On the left the UAV enters a non solvable area, and thus, the formation has to move to keep the UAV inside a solvable area. A control $\mathbf{u}_C$ is being applied to the OWMR formation, following the UAV motion.

For the formation control it is needed to define where robots lie within the formation. A simple method is to define the center coordinates among a group of robots. The center of the OWMR formation $\mathbf{p}_C$ is defined as the mean of the OWMR positions[1].

$$\mathbf{p}_C = \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} \tag{8.3}$$

Each OWMR its able to compute its own position $\mathbf{p}_i$ and the position $\mathbf{p}_C$ is sent to all of them through the network. Also, each robot can handle input commands $\mathbf{u}_i$ and formation commands $\mathbf{u}_C$. In Fig. 8.4, the OWMR control scheme is depicted including the formation control.

[1]If the circumcenter is taken and the number of robot increases it is not convenient.

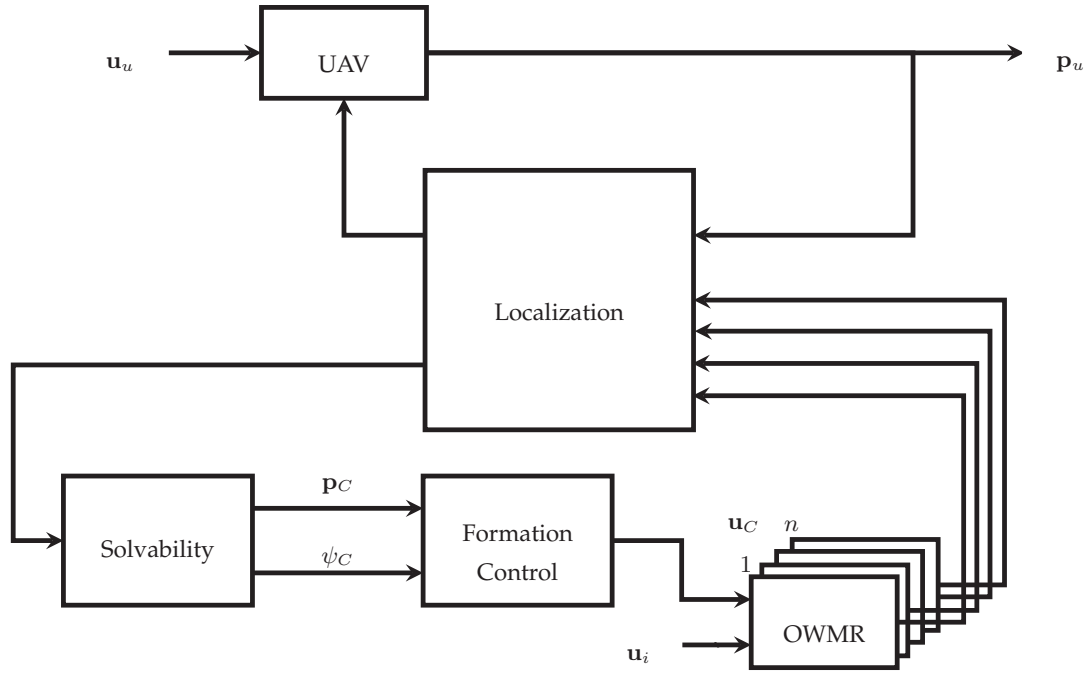


FIGURE 8.4: OWMR control scheme with formation control. The SM is used to calculate a formation control signal $\mathbf{u}_C$, this signal is a vector that can be sum up to the motion of the OWMR.

The control $\mathbf{u}_C$ is given by:

$$\mathbf{u}_C = \begin{bmatrix} u_x \\ u_y \\ u_\omega \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ \omega_C \end{bmatrix} \tag{8.4}$$

where u_x is the formation velocity in the x direction, u_y the formation velocity in the y direction and u_ω is the angular velocity around $\mathbf{p}_C$. For a formation with only three elements, the formation has an orientation ψ_C in the direction perpendicular to the vector $\mathbf{v}_{12} = \mathbf{p}_2 - \mathbf{p}_1$. These concepts are depicted in Fig. 8.5.

The signal $\mathbf{u}_C$ commands the OWMR formation, and then each OWMR calculate how they are moving within the formation. In a time step δ_t, ω_C produces a rotation R_C around $\mathbf{p}_C$.

$$R_C = \begin{bmatrix} \cos(\omega_C \delta_t) & -\sin(\omega_C \delta_t) \\ \sin(\omega_C \delta_t) & \cos(\omega_C \delta_t) \end{bmatrix} \tag{8.5}$$

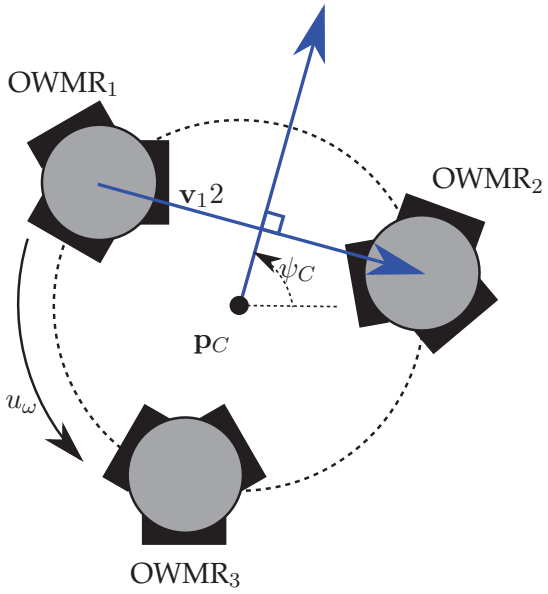


FIGURE 8.5: Formation variables. A formation f_{abc} centered at $\mathbf{p}_C$ has an orientation ψ_C. It can rotate around $\mathbf{p}_C$ with an angular velocity u_ω and the forward motion is in the direction of perpendicular to $\mathbf{v}_1 2$.

Each OWMR calculates its velocity by estimating how much they are moving with respect of the formation center center $\mathbf{p}_C$. The translation $\mathbf{p}_t$ each robot experience is:

$$\mathbf{p}'_t = \mathbf{p}_i - \mathbf{p}_C \tag{8.6}$$

$$\mathbf{p}'_r = \mathbf{p}'_t \tag{8.7}$$

$$\mathbf{p}'_i = \mathbf{p}'_r + \mathbf{p}_C \tag{8.8}$$

where the orientations ψ_i is ignored. The velocity around $\mathbf{p}_C$ is:

$$\mathbf{v}_{\omega i} = \frac{\mathbf{p}'_i - \mathbf{p}_i}{\delta_t} \tag{8.9}$$

If l_{Cu} is greater than a threshold the formation move towards $\mathbf{p}_u$ with a linear velocity equal to:

$$\mathbf{v}_{xyi} = k_{xy} l_{Cu} \mathbf{n}_{Cu} \tag{8.10}$$

where k_{xy} is a constant and $\mathbf{n}_{Cu}$ is the unitary vector from $\mathbf{p}_C$ to $\mathbf{p}_u$. The total velocity each robot OWMR$_i$ has to apply due to the control $\mathbf{u}_C$ is:

$$\mathbf{v}_{Ci} = \mathbf{v}_{\omega i} + \mathbf{v}_{xyi} \tag{8.11}$$

The SM, Chapter 7, can be used to determine if the formation of OWMR needs to move or rotate to maintain a good solvability percentage while the formation is moving. This gives a mobile localization platform an advantage over static network localization indoor environments.

The UAV angle ψ_u is the angle of the formation to the UAV, defined as the angle of the vector $\mathbf{v}_{Cu} = \mathbf{p}_u - \mathbf{p}_C$. The UAV enters within a threshold into the unsolvable area and an error angle can be calculated:

$$\psi_e = \psi_u - \psi_C \tag{8.12}$$

where ψ_e can be used to send a velocity command to the formation and the distance l_{Cu} keeps the formation close enough to the UAV. The error ψ_e causes the formation to rotate with angular velocity ω_C.

8.4 Conclusions

This chapter covered some properties for the OWMR formation when executing the multi robot localization (MLR) algorithm and they will be used in the following chapters to test it. The notation for the triad of robots was defined as the selection of three OWMR used to perform the localization. A simple algorithm for triad selection based on the DOP value was presented. A formation control scheme was defined using the SM as the measurement for a control input to the formation of OWMR. This control delivers a signal $\mathbf{u}_C$ that the OWMR can use to move within the formation.

9 Numerical Simulations

In this section the multi robot localization algorithm will be tested with numerical simulations. The objective is to validate the theory is employed as expected. The simulations were performed in Matlab[69] using as estimators an EKF and a PF as described in Chapter 5. Two simulation are performed to show the results of triad selection, for this simulations it is proposed to use a formation control based on the DOP as explained in Section 8.2. Then, one more simulation is performed to show the control input $\mathbf{u}_C$ with the solvability range adjustment method of Section 8.3.

9.1 Linear path with formation selection

If several robots are used in the multi robot formation, they can be arranged to cover a larger area, and considering this fact if the UAV moves through a large area maybe the OWMR cannot keep with the velocity of the drone, but many of them can cover the flying area. If five robots are available, the different formations f_{abc} that can be used are listed in Table 9.1. Assigning a number to the formation will be helpful in this experiment.

TABLE 9.1: Possible formations with five robots. All the possible formations having five robots are shown $\wp(C_3^5)$.

Formation#	1	2	3	4	5	6	7	8	9	10
Formation	f_{123}	f_{124}	f_{125}	f_{134}	f_{135}	f_{145}	f_{234}	f_{235}	f_{245}	f_{345}

To begin with, the first simulation shows the UAV passing through an static formation of five OWMR.

The problem setup is shown in Fig. 9.1. The five robots are located at:

$$\mathbf{p}_1 = [-4.4947, -3.4600, 0]^\intercal \tag{9.1}$$
$$\mathbf{p}_2 = [0, 3.4600, 0]^\intercal \tag{9.2}$$
$$\mathbf{p}_3 = [7.4911, -2.5950, 0]^\intercal \tag{9.3}$$
$$\mathbf{p}_4 = [14.9822, 1.7300, 0]^\intercal \tag{9.4}$$
$$\mathbf{p}_5 = [22.4734, -5.1900, 0]^\intercal \tag{9.5}$$

The motion path is in the x direction and the initial formation to be used for localization is f_{123}. The base plane A_b is located at the camera centers since the distances are taken center to center from these cameras to the UAV position. In this test the OWMR are static.

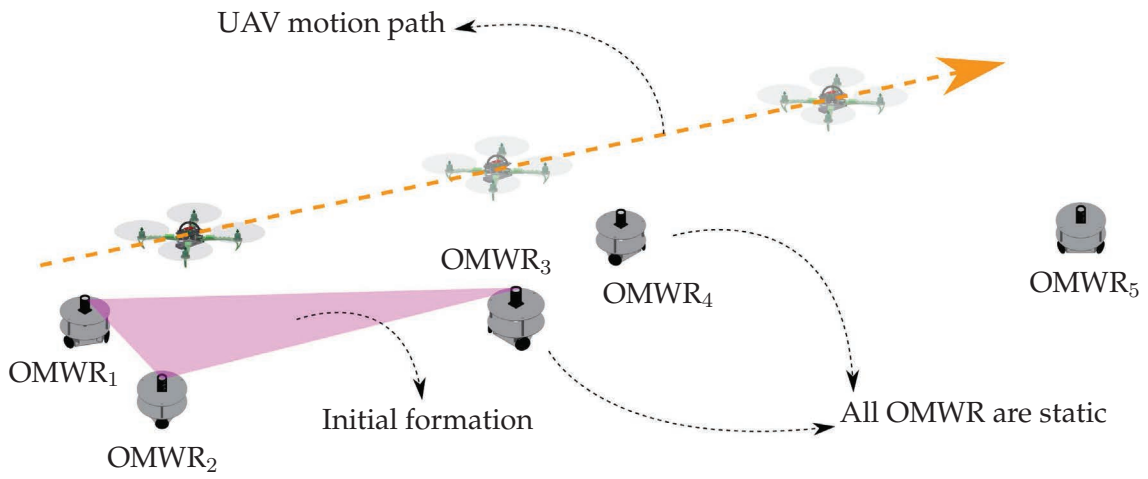
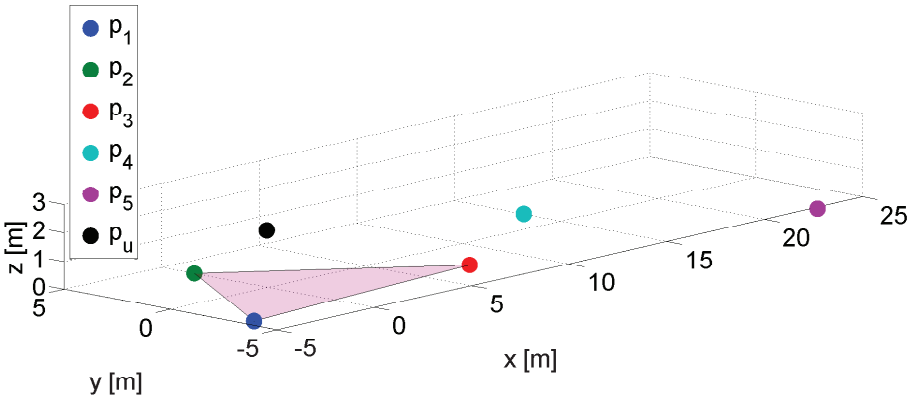
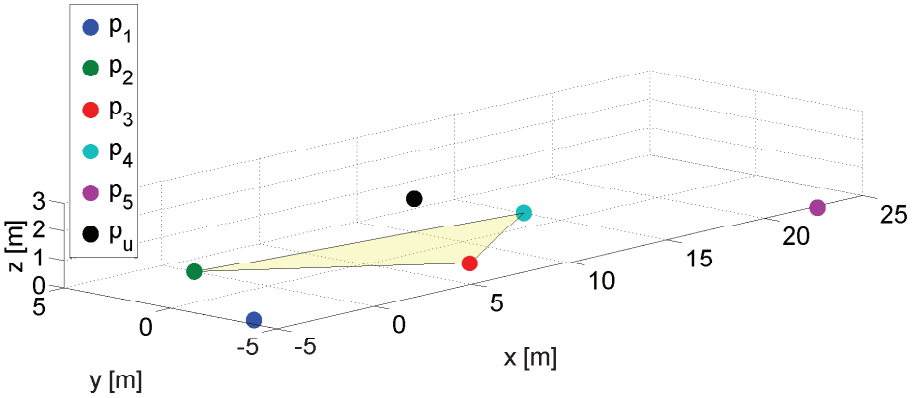


FIGURE 9.1: Linear path simulation setup. The motion path is in the x direction and the initial formation to be used for localization is f_{123}. The base plane A_b is located at the camera centers since the distances are taken center to center from these cameras to the UAV position. In this test the OWMR are static.

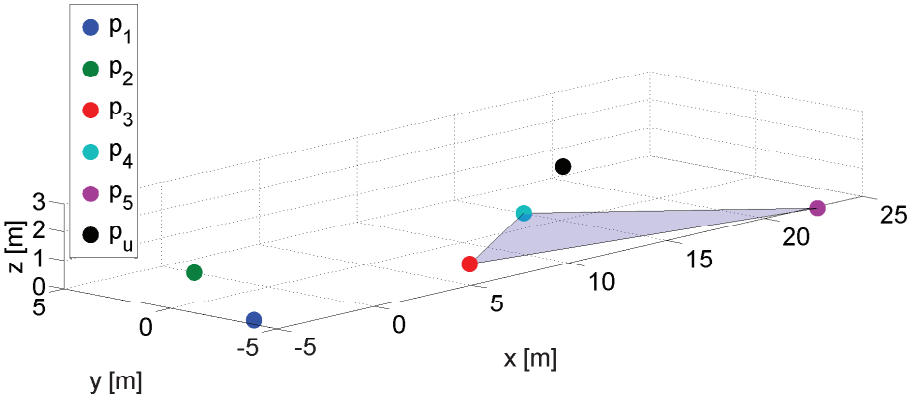
When the robot is near formation f_{123} one would expect the UAV to use that formation, when it approaches f_{234} the formation is required to change, and the same concept applies for f_{345}. In Fig. 9.2 it is depicted positions for $\mathbf{p}_u$ where different formations f_{abc} are expected to be selected.



(a) Formation selection, case f_{123}.



(b) Formation selection, case f_{234}.



(c) Formation selection, case f_{345}.

FIGURE 9.2: Formation f_{abc} selection at different positions $\mathbf{p}_u$. The formation selected f_{abc} needs to change accordingly to the UAV position.

The solution to the simulation is shown in Fig. 9.3, the UAV moves in a straight path in the x direction at 1 m/s. As the UAV moves through the formation, the selection f_{abc} changes. The time at which these changes happen are depicted by gray walls and the formation changed due to a change in the the DOP value based on the algorithm of Section 8.2. The DOP threshold was set at 3[1].

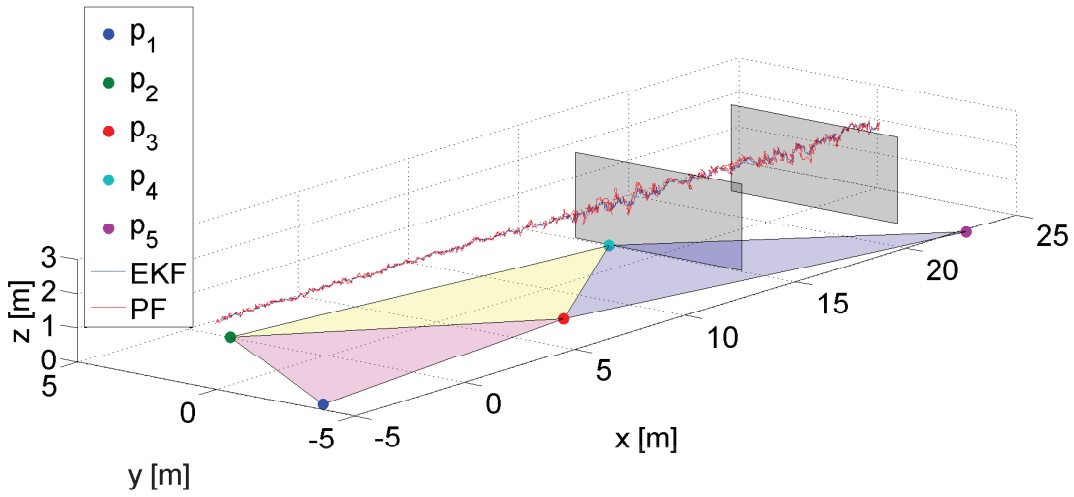


FIGURE 9.3: Linear path simulation results. Several OWMR can be arranged to cover a large area, then, if the UAV passes through them, a different triad can be selected to perform the localization.

Five robots are used so the possible combinations are the same as the ones presented in Section 8.1. For this simulation the formations were listed as $\{1, 7, 10, 2, 4, 3, 8, 9, 6, 5\}$, so the first available formation is f_{123}, the second f_{234} and so on. Results are shown in Fig. 9.4, the selection selected is correct according to the path taken by the UAV.

In this simulated environment the EKF performs better than the PF, but later it will be seen in Chapter 10 that the PF is more robust in real time applications in the presence of non linearities. The errors in the EKF are within 0.1 m (Fig. 9.5).

[1] A good value for the DOP has to be tested for the requirements.

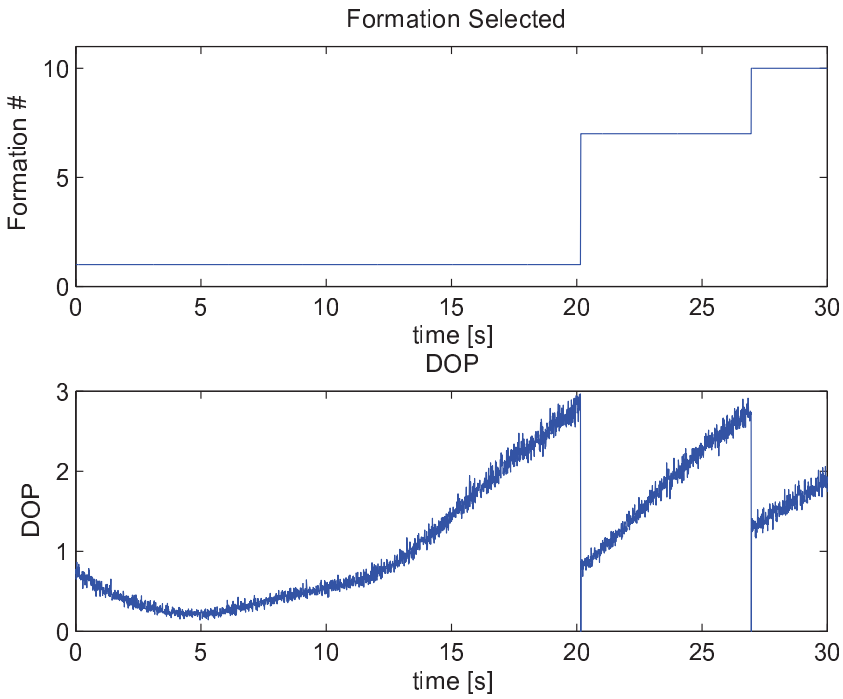


FIGURE 9.4: Linear path simulation formation selection. The triad selected in the multi robot formation is picked accordingly to the DOP value sensed by the OWMR. The number of the formation selected is accordingly to Table 9.1.

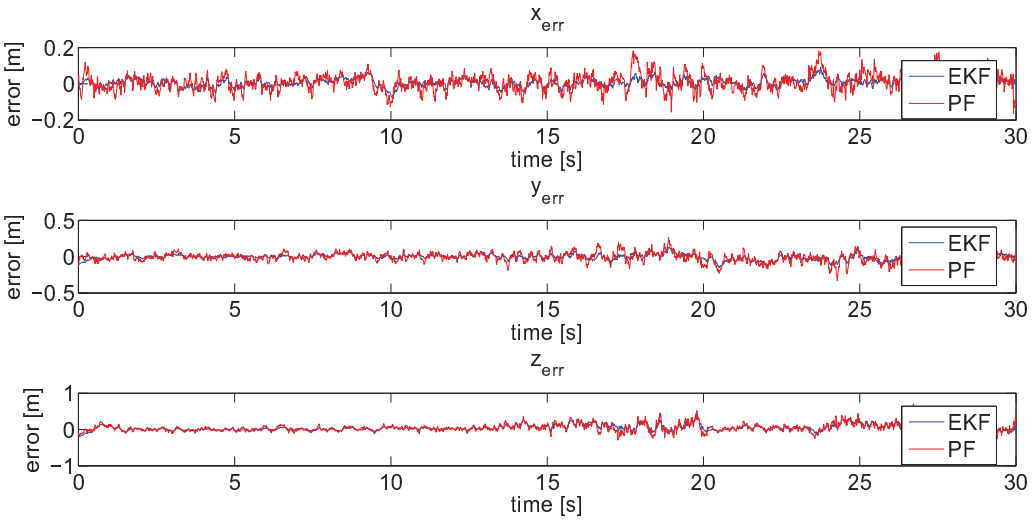


FIGURE 9.5: Linear path simulation errors. The EKF estimator has a smoother response than the PF estimator. The errors in the EKF are within 0.1 m.

9.2 Circular path with moving formation selection

The second simulation consists in the UAV moving in a circular motion around the multi robot formation of OWMR as shown in Fig. 9.6. The UAV will replicate this motion three times.

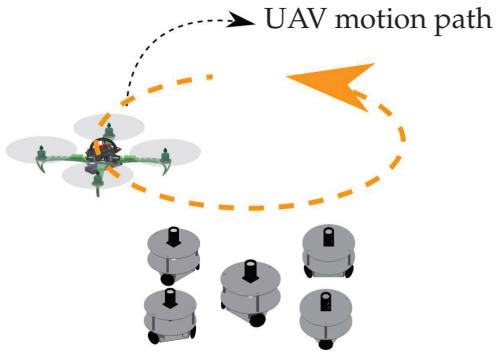


FIGURE 9.6: Circular path simulation setup 1. The UAV is moving in a circular motion around the multi robot formation of OWMR. This motion will be replicated three times.

Then, at the same time, the OWMR formation will move in the x direction. The motion for all the robots is depicted in Fig. 9.7.

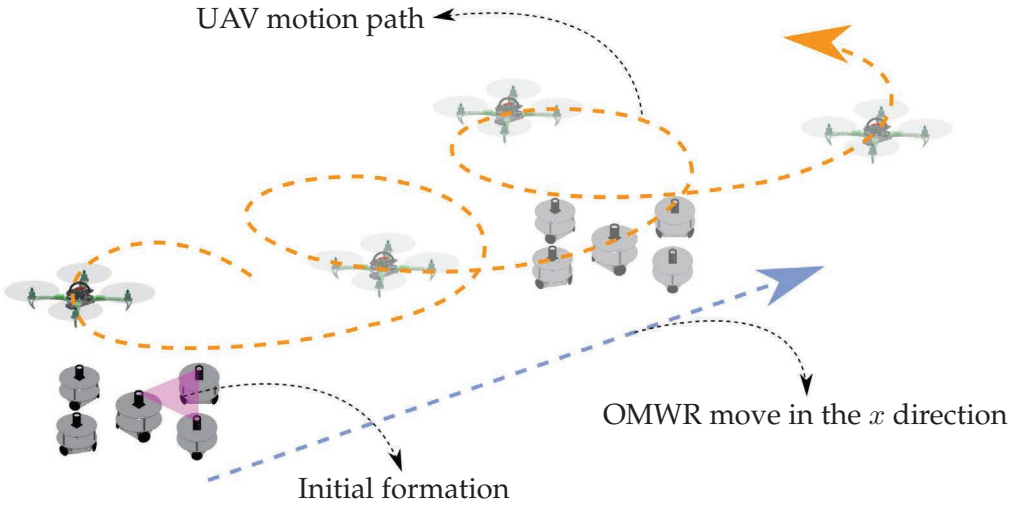


FIGURE 9.7: Circular path simulation setup 2. At the same time that the UAV is circling the OWMR formation, the formation will move in the x direction as shown.

The resulting motion is shown in Fig. 9.8 for a test using an EKF and a PF estimators. For this test the formations were listed as $\{4, 6, 3, 1\}$, see Table 9.1.

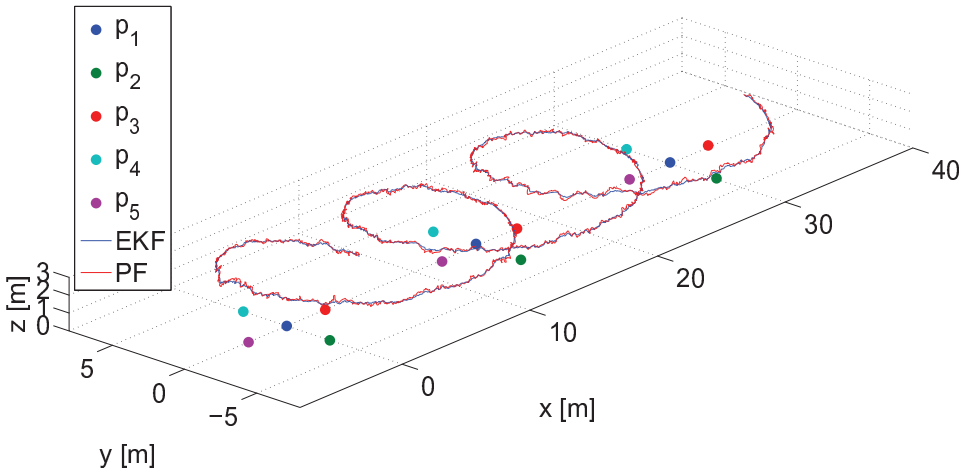


FIGURE 9.8: Circular path simulation results. The UAV moves around an OWMR formation while the formation is moving forward. The selected formation f_{abc} changes accordingly to the UAV position.

The correct formation is selected along the path taken by the robots, see Fig 9.9. The triad selected in the multi robot formation is picked accordingly to the DOP value sensed by the OWMR. Since the UAV moves around the OWMR the formation selected is repeated each time the UAV makes a turns round the multi robot formation.

The errors are shown in Fig 9.10. The EKF shows a smaller error than the PF. This happens because the environment is simulated but in real time scenarios implemented on actual robots, the non linearities affect the behavior of the filters, so it will be shown that the particle filter is more robust to nonlinearities.

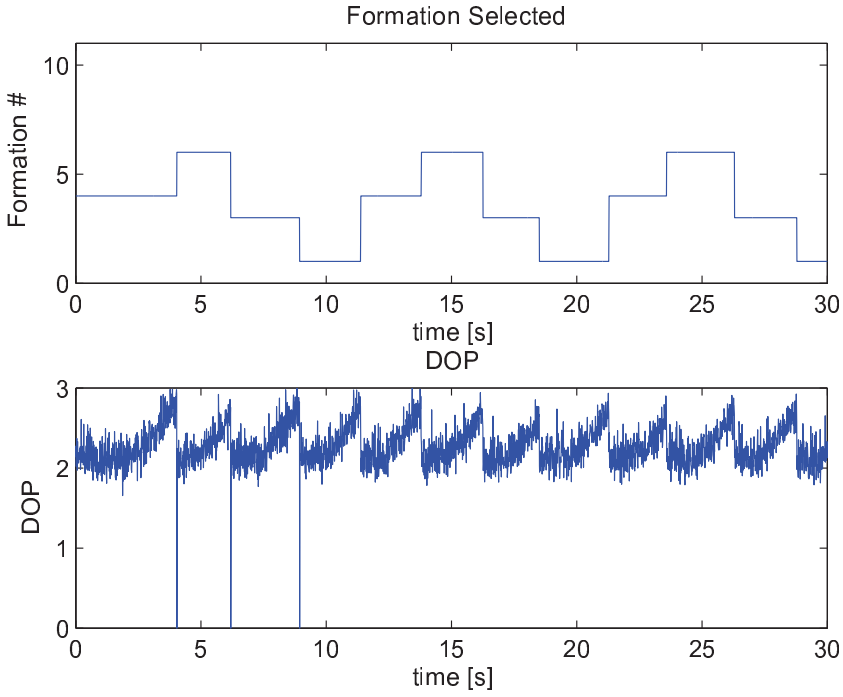


FIGURE 9.9: Circular path simulation formation selection. The triad selected in the multi robot formation is picked accordingly to the DOP value sensed by the OWMR. Since the UAV moves around the OWMR the formation selected is repeated each time the UAV makes a turns round the multi robot formation.

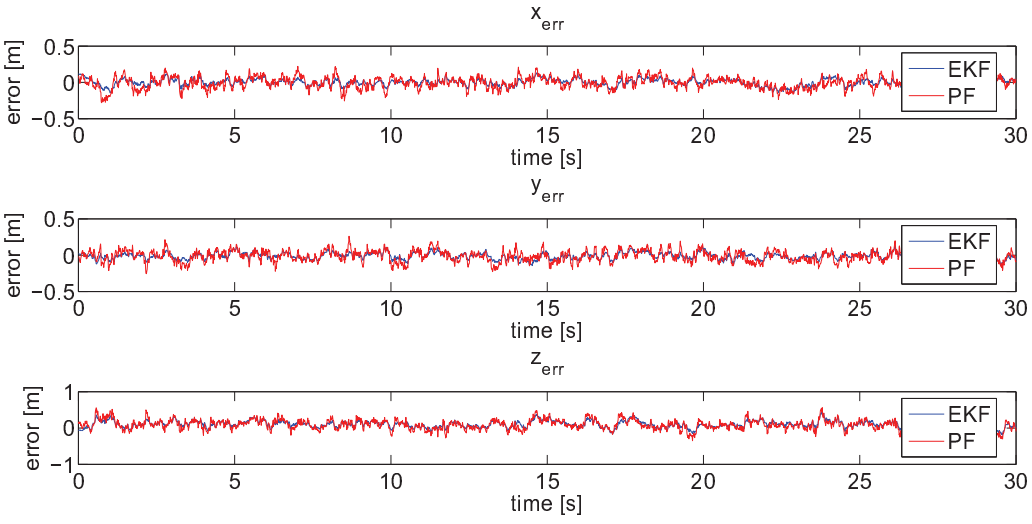


FIGURE 9.10: Circular path simulation errors. The UAV moves around an OWMR formation while the formation is moving forward. The errors are acceptable and as it can be seen the EKF performs better than the particle filter.

9.3 Formation control simulation with solvability range adjustment

In this test it is used a solvability range adjustment, in which the solvability is used to drive the orientation of the OWMR formation. A numerical simulation is executed simulating a OWMR formation moving in a corridor and turning. In Fig. 9.11 the problem is depicted. The formation moves in the y direction and when it moves to the x direction the formation has to rotate. Since the robots are OWMR they do not need to turn but using an SM a control signal $\mathbf{u}_C$ can be used to make the formation turn.

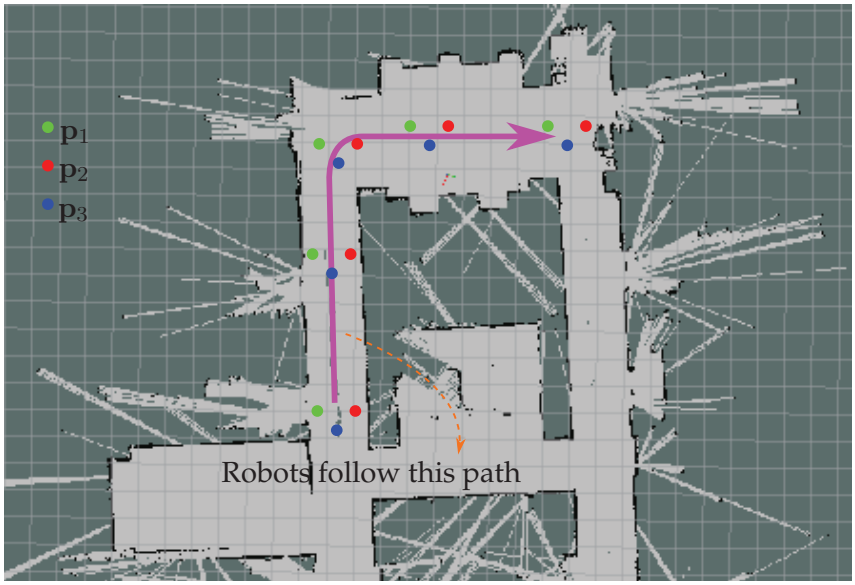


FIGURE 9.11: Solvability range adjustment test path. Test path used for the simulation of the solvability range adjustment control. The UAV moves along a corridor and then it turns. Since the robots are OWMR they do not need to turn but using an SM a control signal $\mathbf{u}_C$ can be used to make the formation turn.

Fig. 9.12 shows three positions A, B and C along the expected path the OWMR formation is following. The positions A, B and C denote the initial position, a position where the control input takes place and the final position, respectively.

The distance l_{12} has a ratio $1.5 : 1$ to the distances l_{13} and l_{23}, this allows for a solvable region in front of the formation movement. In Fig. 9.13 the initial SM at position A is presented.

The OWMR formation rotates at point B to keep the UAV in front of it, see Fig. 9.14. The control $\mathbf{u}_C$ is being applied because the UAV enters an area with less solvability, sending a control signal $\mathbf{u}_C$ and making the formation to turn, following the UAV motion.

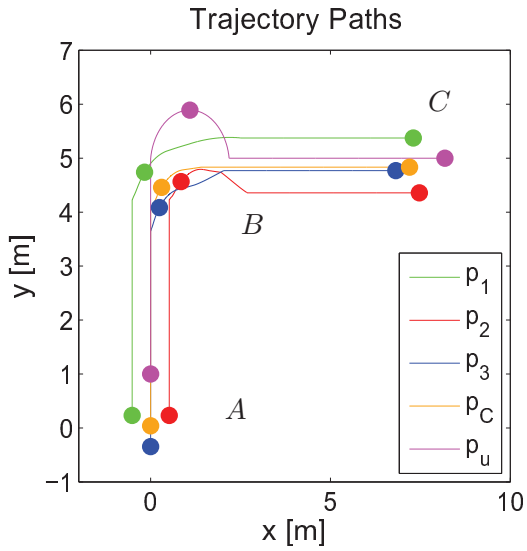


FIGURE 9.12: OWMR paths for solvability range adjustment. How each OWMR moves during the test for solvability range adjustment control test is shown. The positions A, B and C denote the initial position, a position where the control input takes place and the final position, respectively.

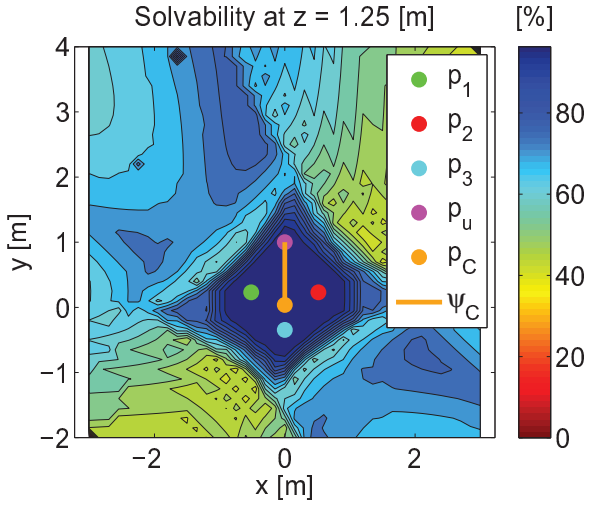


FIGURE 9.13: SM at position A. Initial SM at the beginning of the solvability range adjustment control test. A solvable region lies in front of the formation movement.

At point C it is noticed that the formation is slightly rotated, that is because the OWMR need to track $\mathbf{p}_u$ even do the direction of motion is not on the same orientation. The SM in C is shown in Fig. 9.15.

The resulting control input for the solvability range adjustment consist on a linear component and a angular component. The control each robot applies is different as they do not move

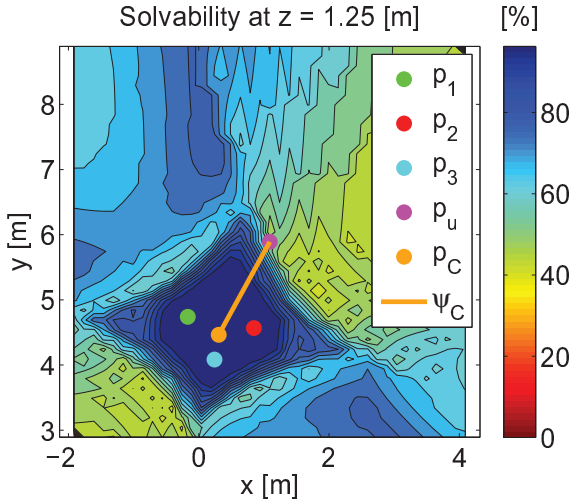


FIGURE 9.14: SM at position B. The control $\mathbf{u}_C$ is being applied because the UAV enters an area with less solvability, sending a control signal $\mathbf{u}_C$ and making the formation to turn, following the UAV motion.

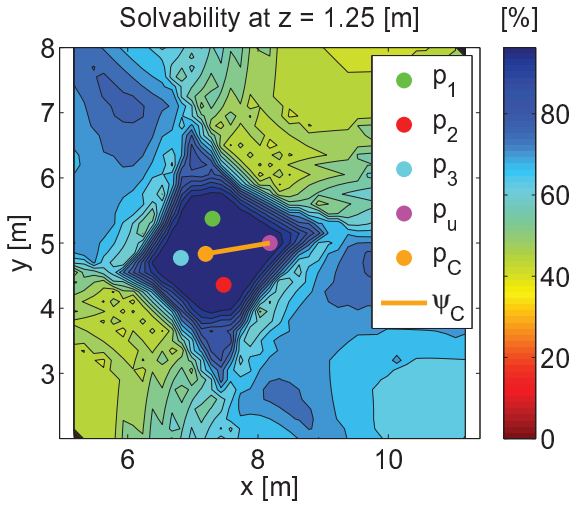


FIGURE 9.15: SM at position C. Final SM at the end of the solvability range control test. The formation is slightly rotated, that is because the OWMR need to track $\mathbf{p}_u$ even do the direction of motion is not on the same orientation.

in the same manner within the formation. The control input is shown in Fig. 9.16.

Setting a control mode for linear, angular or a combination a both, it is possible to see how the control is being applied in each phase of the motion. For this test, the control mode is shown for the times when the controller kicks in (Fig. 9.17), each robot applies the control as needed according to their positions within the formation.

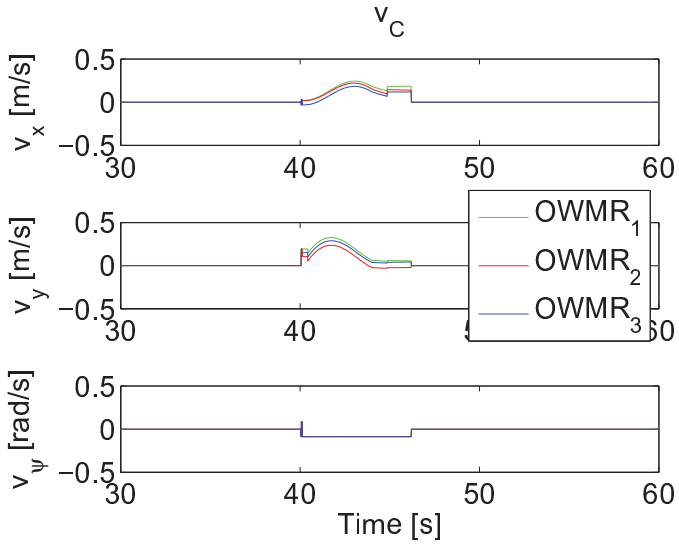


FIGURE 9.16: $\mathbf{u}_C$ applied by the OWMR formation. The control each robot applies is different as they do not move in the same manner within the formation.

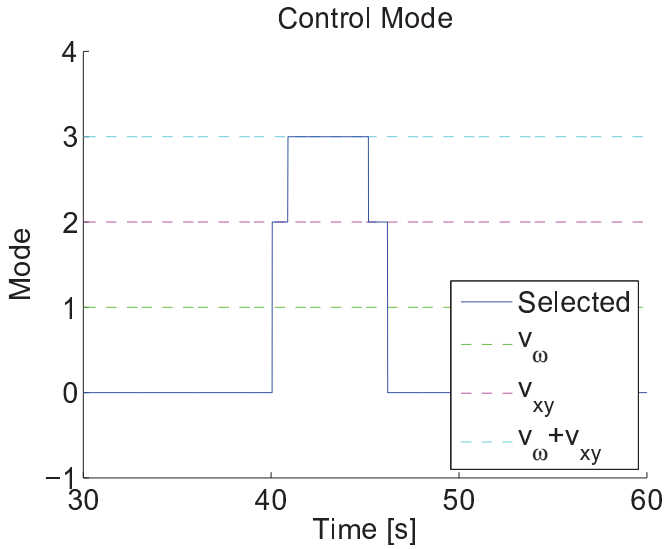


FIGURE 9.17: Solvability range adjustment simulation control mode. The control $\mathbf{u}_C$ has a linear and angular modes, these modes can be used alone or combined. The control mode is shown for the times when the controller kicks in.

9.4 Conclusions

In this chapter three simulations were performed to validate the algorithms presented in previous chapters. It was shown that the localization algorithm proposed is a viable option and be used to compute the UAV position in three dimensions. Also a multi robot environment was used to define how a different triad of robots can be used to perform the trilateration more adequately. Two simulation were performed to present the concepts of triad selection. A simulation was performed to present how the SM can be used to drive a control signal for the multi robot formation, and change the orientation of the OWMR formation to better calculate the UAV position.

10 Real Time Localization Experiments

In this section the concepts presented in this work are implemented in a real time system using robots. The OWMR calculate their position $\mathbf{p}_i$ as explained in Section 4.1 and using the cameras mounted on them the distances l_{iu} are obtained, Section 4.3. Each OWMR can move independent from the others, they have an Odroid[70] CPU on board that processes all the sensed information required for the localization. The sensed signals, $\mathbf{p}_i$ and l_{iu}, are transmitted via WiFi to a central computer where the multi localization algorithm calculates the position of the UAV. The data transmission is done using the middleware robotics operating system ROS[71], which is used extensively by the robotics community. The OWMR send data at a 100 Hz, the camera distance readings are sent at 5 to 7 Hz and the multi robot localization algorithm is run at a 100 Hz. Finally, an OptiTrack Motion Camera System[72] is used as the ground truth to validate the algorithms and the implementation.

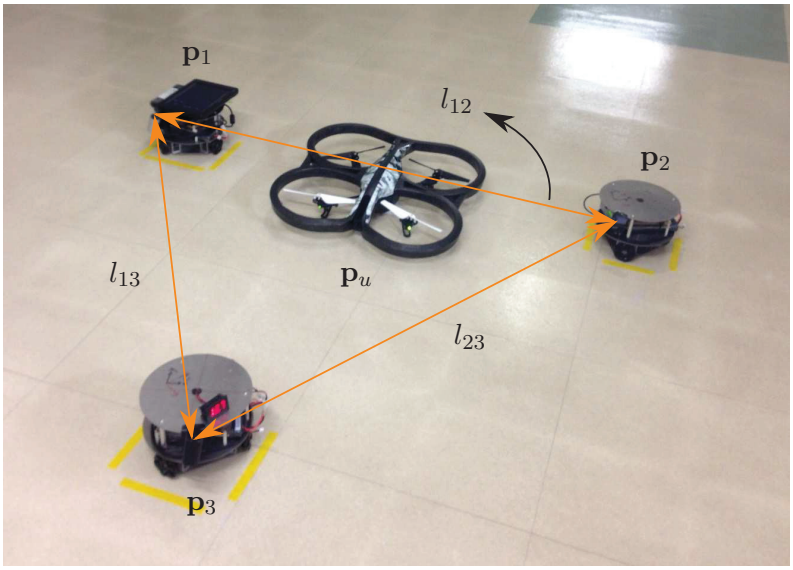


FIGURE 10.1: Robot ID assignation for real time testing. Each robot is assigned a position $\mathbf{p}_i$ which also the ID to identify it.

For the real time tests a group of three OWMR is used. These robots move together in a triangular formation in a predetermined pattern and the UAV moves above them. The OWMR execute the localization algorithm using the trilateration as defined in Chapter 5. The robots are freely named OWMR_1, OWMR_2 and OWMR_3 for the tests, with respective coordinates are $\mathbf{p}_i$. The distances l_iu are taken from cameras mounted vertically on the OWMR. The OWMR do not rotate on their axis so the distances l_{ij} between them are taken from the camera lenses. The plane A_b has a vertical offset that has to be taken in account when taking the calculated UAV position $\mathbf{p}_u$. In Fig. 10.1 and Fig. 10.2 the variables for the real time testings are shown.

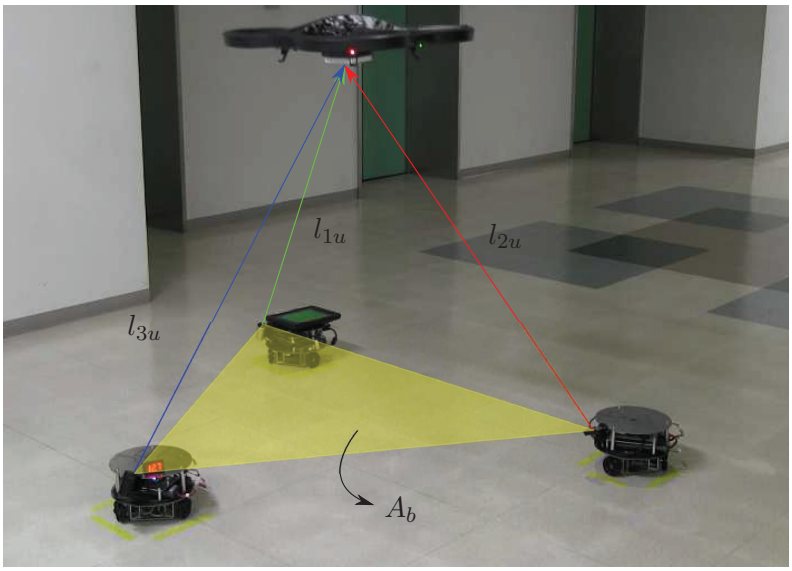


FIGURE 10.2: Camera distances and base plane in a real time environment. The camera distances l_{iu} and the base plane A_b are also defined in for test. The range measurements are taken from the camera lenses to the base of the UAV.

10.1 System initialization

The robots need to know their initial positions, they can be measured directly in the x and y directions. On the other hand, the formation is triangular, and the law of cosines can be used to get the initial positions $\mathbf{p}_i$ knowing the distances the robots have to each other. The distances between the robots are measured as l_{12}, l_{13}, and l_{23}. Then, OWMR_1 is positioned at the origin:

$$\mathbf{p}_1 = [0, 0, 0]^\intercal \tag{10.1}$$

Following, OWMR_2 is located in the x axis next to OWMR_1:

$$\mathbf{p}_2 = [l_{12}, 0, 0]^\intercal \tag{10.2}$$

Finally for OWMR_3 the law of cosines is applied, so:

$$\mathbf{p}_3 = [x_3, y_3, 0]^\intercal \tag{10.3}$$

where[1]:

$$x_3 = l_{13} \cos \theta_{1\_23} \tag{10.4}$$

$$y_3 = -l_{13} \sin \theta_{1\_23} \tag{10.5}$$

$$\cos \theta_{1\_23} = (l_{12}^2 + l_{13}^2 - l_{23}^2)/(2 l_{12} l_{13}) \tag{10.6}$$

As noticed, the reference frame is initialized with a right hand system pointing up. The initialization problem is depicted in Fig. 10.3.

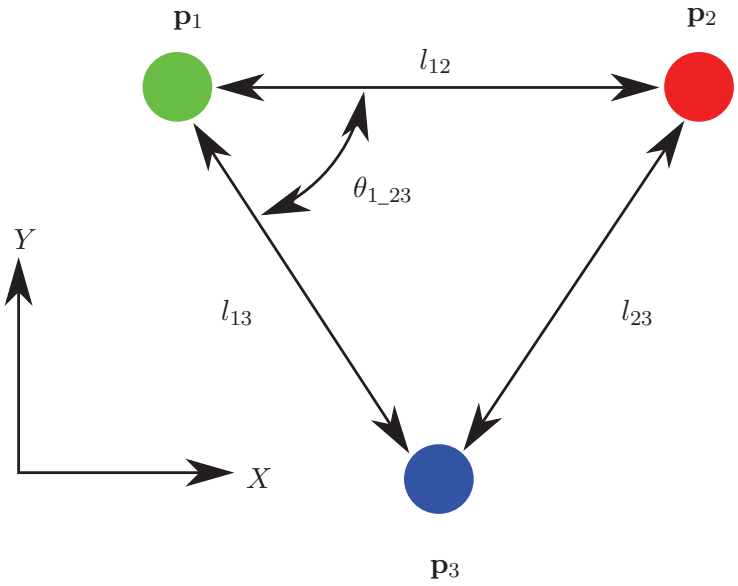


FIGURE 10.3: Reference frame setup. The reference frame is initialized using the law of cosines, which is convenient to obtain the initial coordinates $\mathbf{p}_i$ from measurements l_{ij}.

[1]The notation θ_{i_jk} is used.

The process covariance $\mathbf{R}_t$ is related to the process standard deviation and it is set at:

$$\sigma_R = [0.05, 0.05, 0.05, 0.05, 0.05, 0.05]^\intercal \tag{10.7}$$

The measurement covariance $\mathbf{Q}_t$, is the error propagation from measurements, explained in Chapter 6. For the test performed the distances were set at $l_{12} = 0.98$ m, $l_{13} = 0.92$ m and $l_{23} = 0.95$ m. The velocity of the OWMR is 0.1 m/s.

10.2 Circular path test

For the first test the robot formation moves in a circular path with radius of 1 m while the UAV is help above the formation, as shown in Fig. 10.4. The multi robot formation moves and calculates the position $\mathbf{p}_u$ of the UAV based on the measurements $\mathbf{p}_i$ and l_{iu}.

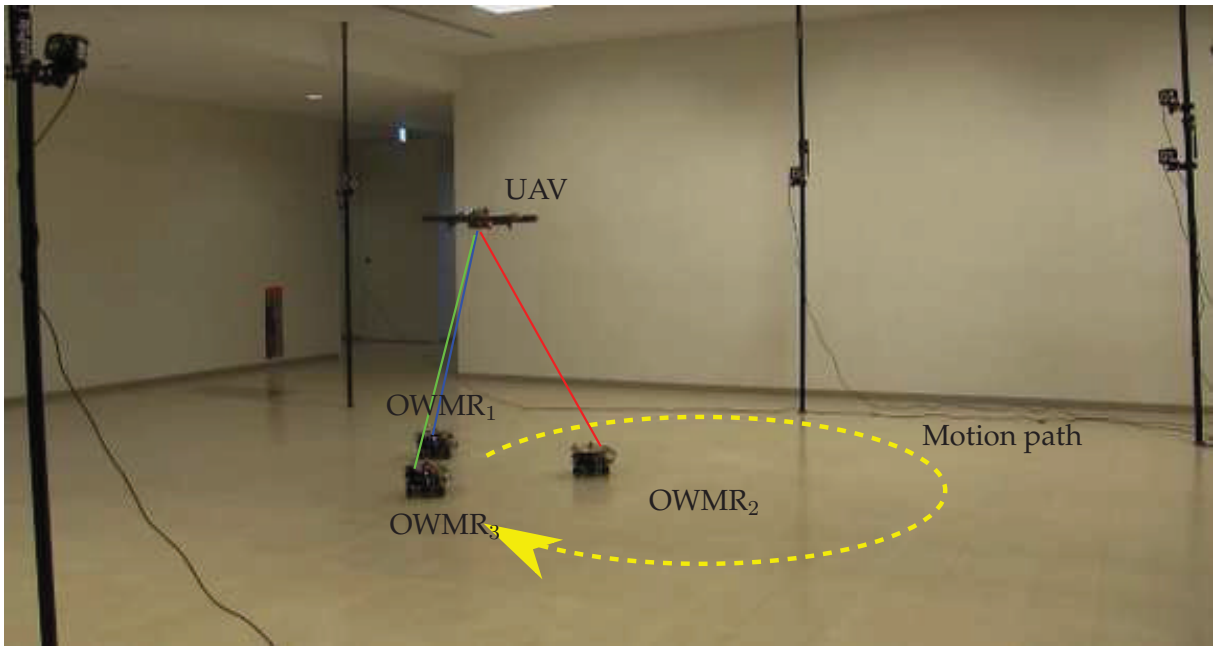


FIGURE 10.4: Real time circular path test. The multi robot formation moves in a circular path, then it calculates the position $\mathbf{p}_u$ of the UAV based on the measurements $\mathbf{p}_i$ and l_{iu}.

The positions x, y and the orientation ψ are depicted in Fig. 10.5.

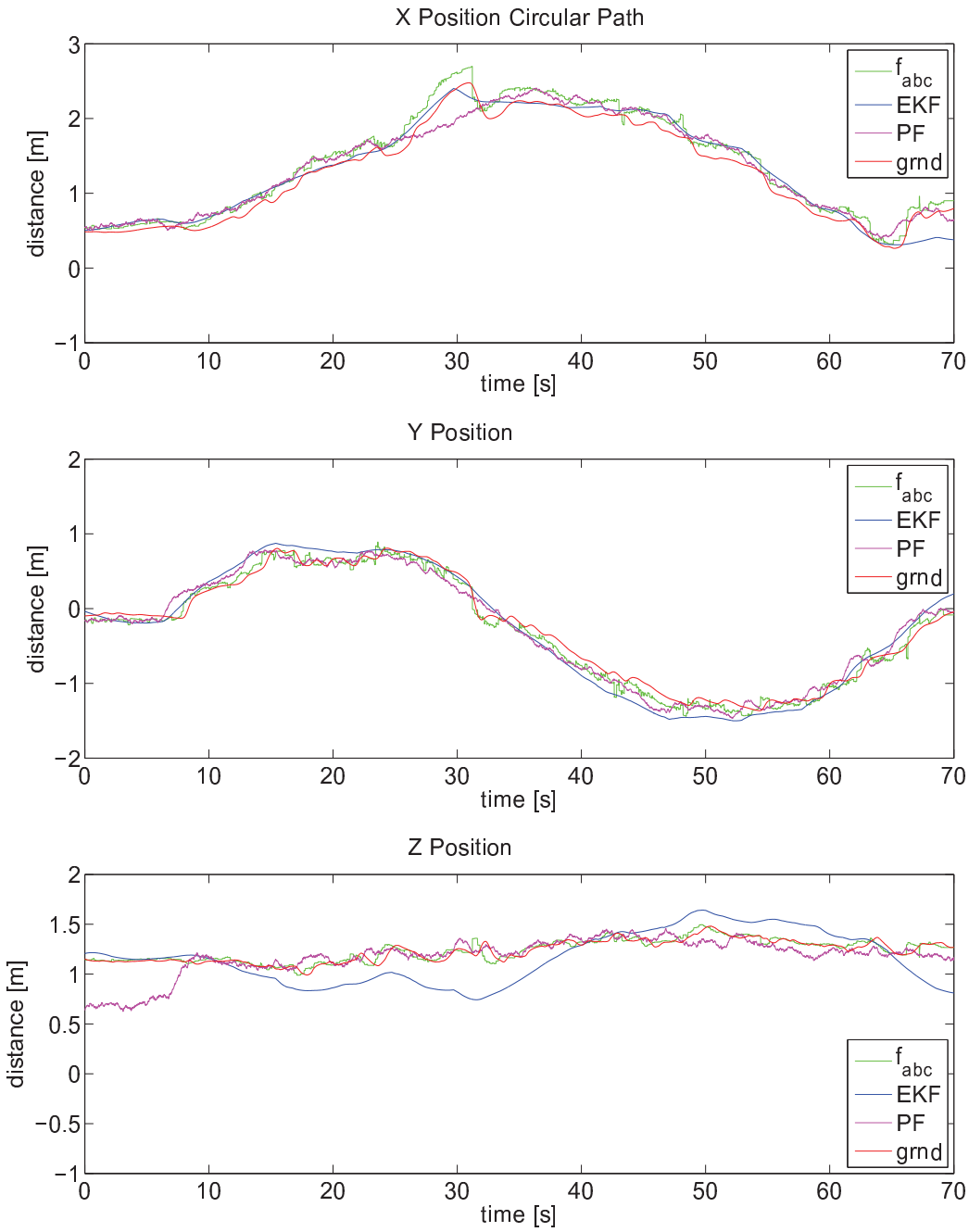


FIGURE 10.5: Positions in the circular path test. The positions calculated by the mobile OWMR formation matches the ground truth positions.

The path followed by the OWMR is shown in Fig. 10.6. The covariances of $\mathbf{p}_2$ are visible. The OWMR have errors in the motion and that is why they do not return exactly to their initial positions.

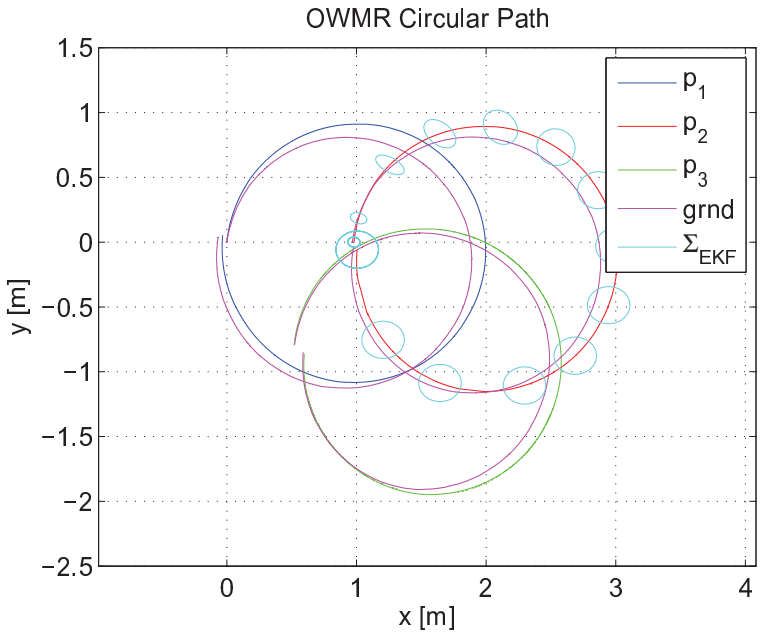


FIGURE 10.6: Circular path followed by the OWMR. The covariances of $\mathbf{p}_2$ are visible. The OWMR have errors in the motion and that is why they do not return exactly to their initial positions.

The positions calculated by the mobile OWMR formation matches the ground truth positions of the Optitrack system and the errors shown in Fig. 10.7 validate the accuracy of the system. As expected the errors are bigger in the z direction. Both estimators, EKF and PF, can track the UAV effectively. The PF is very sensitive to initial conditions. The EKF is not affected so much by the initial conditions as the algorithm can follow the sensed signal $\mathbf{z}_t$ in a better way at the beginning of the test. Even do the EKF has a smoother response it can be seen that the PF responds better to non linearities, as can be seen around second 30. In the z direction the PF outperforms the EKF.

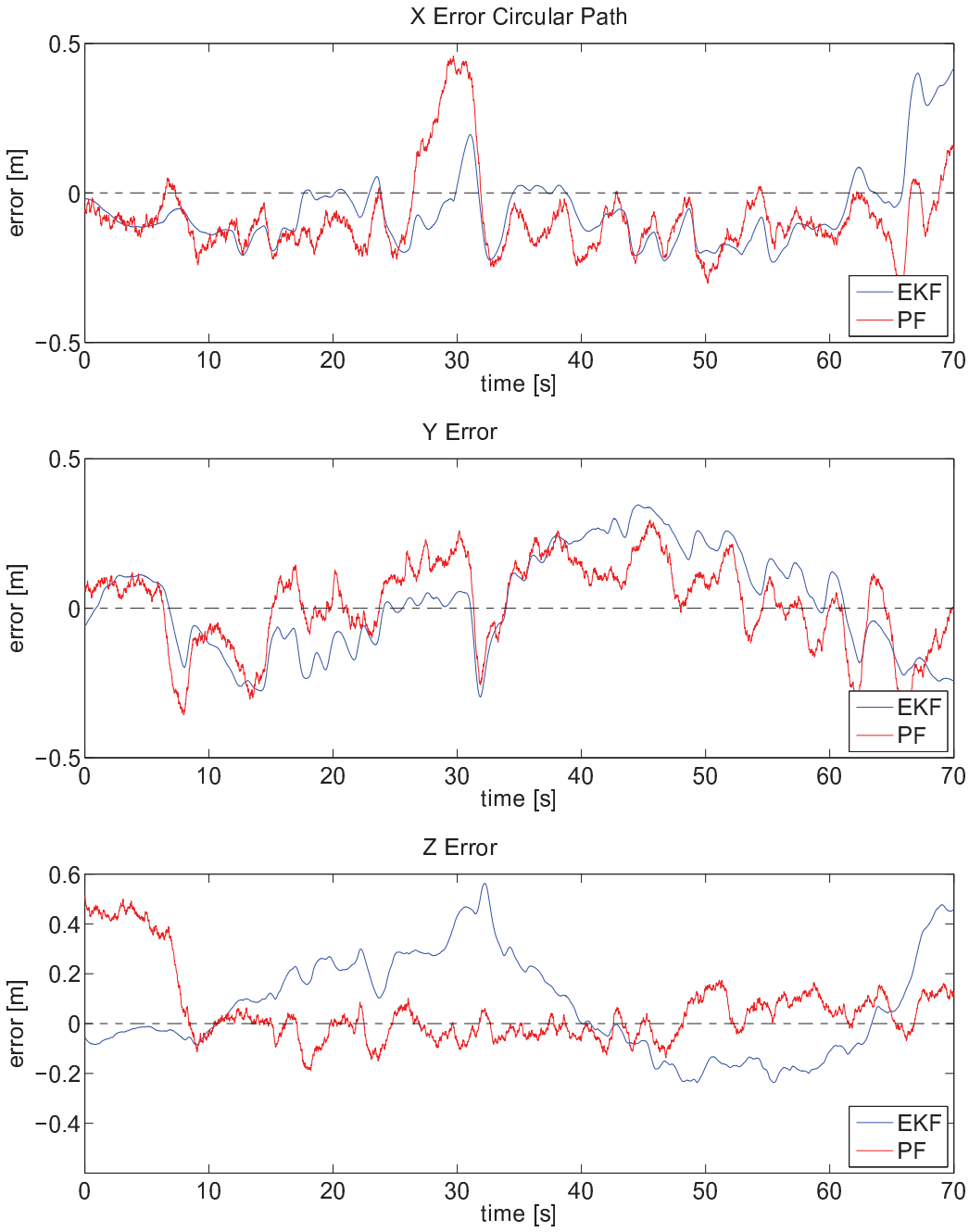


FIGURE 10.7: Positions errors in the circular path test. The errors validate the multi robot localization algorithm proposed. As expected the errors are bigger in the z direction.

10.3 Square swipe test

In this test, the multi robot formation swipes a square area of 2 by 2 meters. Again, the UAV is help above the formation, see Fig. 10.8. The position $\mathbf{p}_u$ of the UAV is calculated using the multi robot localization algorithm.

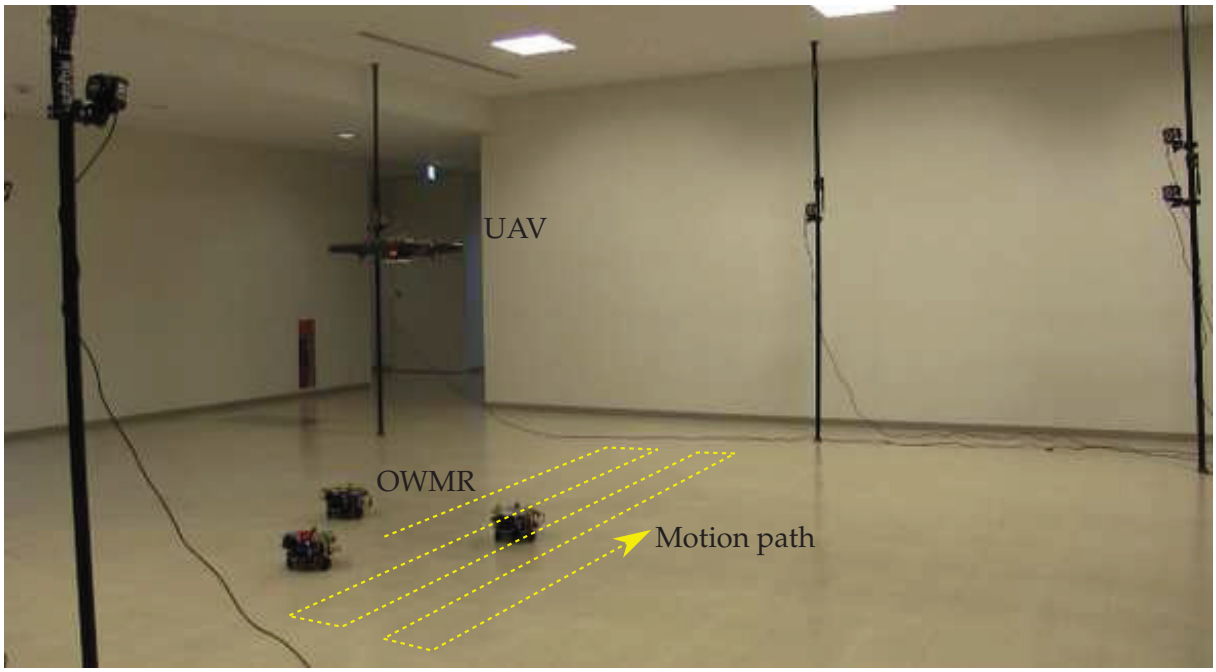


FIGURE 10.8: Real time square path test. The multi robot formation swipes a squared area as it calculates the position $\mathbf{p}_u$ of the UAV based on the measurements $\mathbf{p}_i$ and l_{iu}.

Fig. 10.9 shows the results for positions x and y and orientation ψ. At this point the OWMR do not perform SLAM, so their covariances increase until it is impossible for the estimators to keep tracking on the UAV. On the other hand, as opposed to the simulations performed in Chapter 9 the PF outperforms the EKF as it is more robust to the uncertainty, giving that the initial conditions are met.

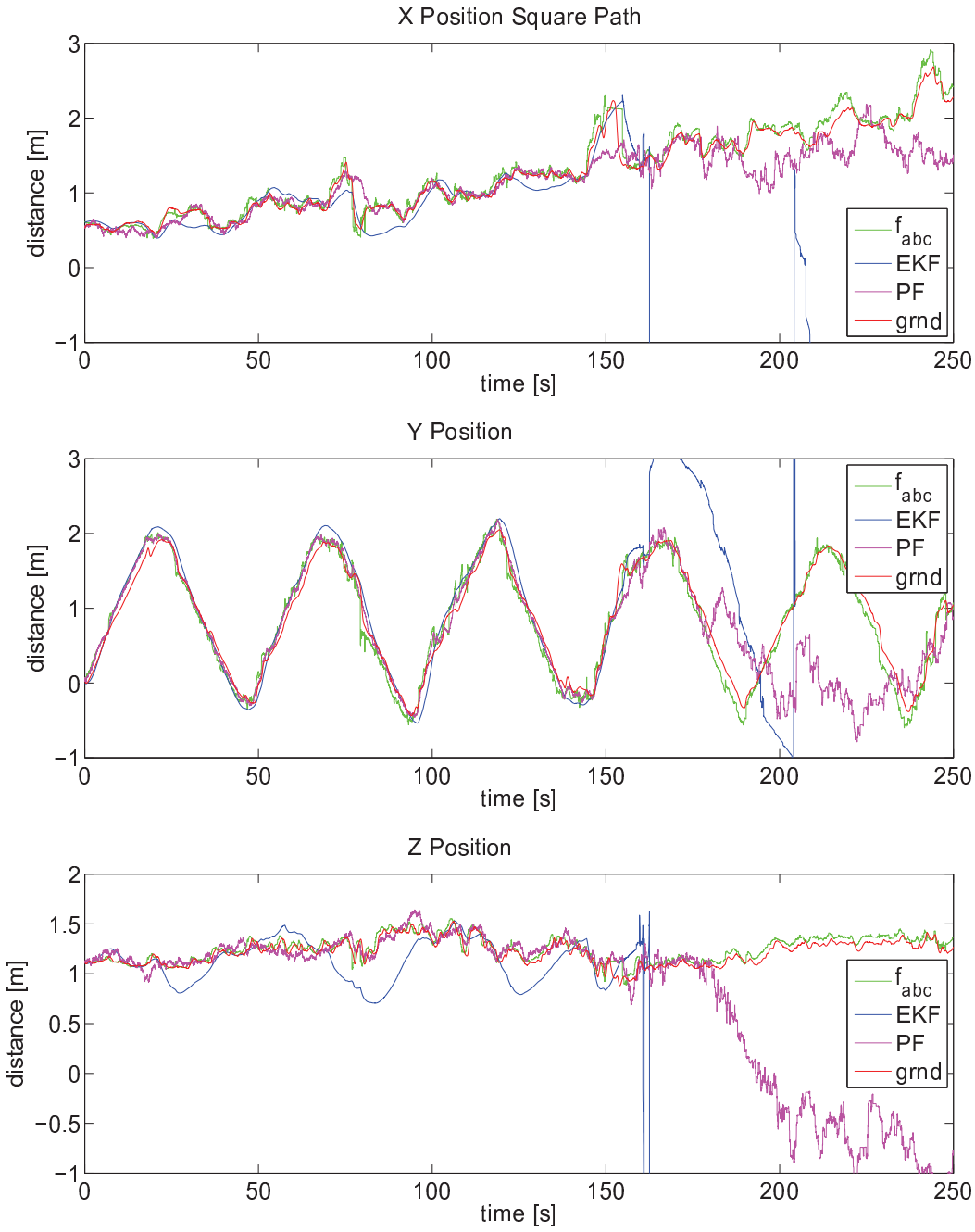


FIGURE 10.9: Positions in the square path test. The OWMR swipe a square area, due to the increments in the error covariance the OWMR end up losing tracking of the UAV.

The path followed by $OWMR_3$ is depicted in Fig. 10.10. It is seen that $OWMR_3$ thinks it is moving correctly but in reality it is not. The ground truth path, in red, shows how the robot deviates due to errors.

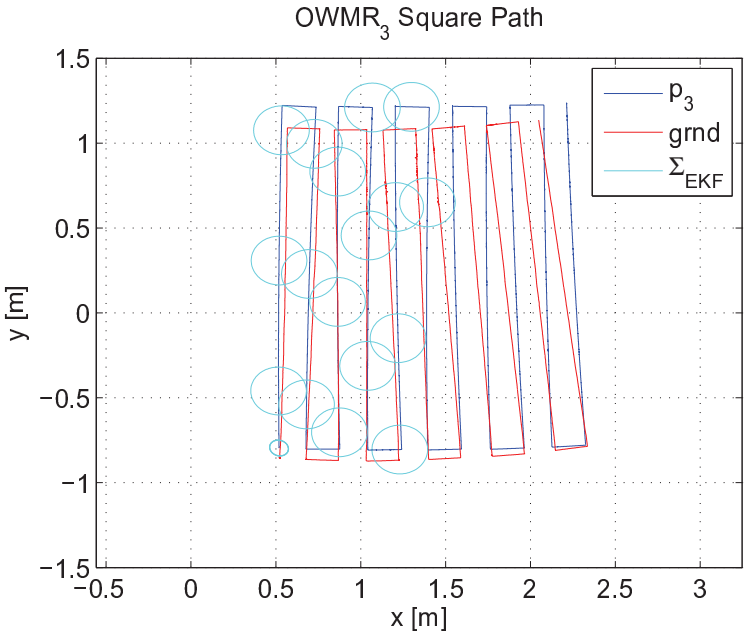


FIGURE 10.10: Path followed by the OWMR. The ground truth path, in red, shows how $OWMR_3$ deviates due to errors.

The errors in positions are shown in Fig. 10.11. The multi robot localization algorithm works. Tracking is maintained as long as three minutes without improving the uncertainty in the OWMR. In the tracked regions, the errors are below ± 25 centimeters. The EKF shows a smoother response but the PF is more robust tracking the sensed signal in all directions.

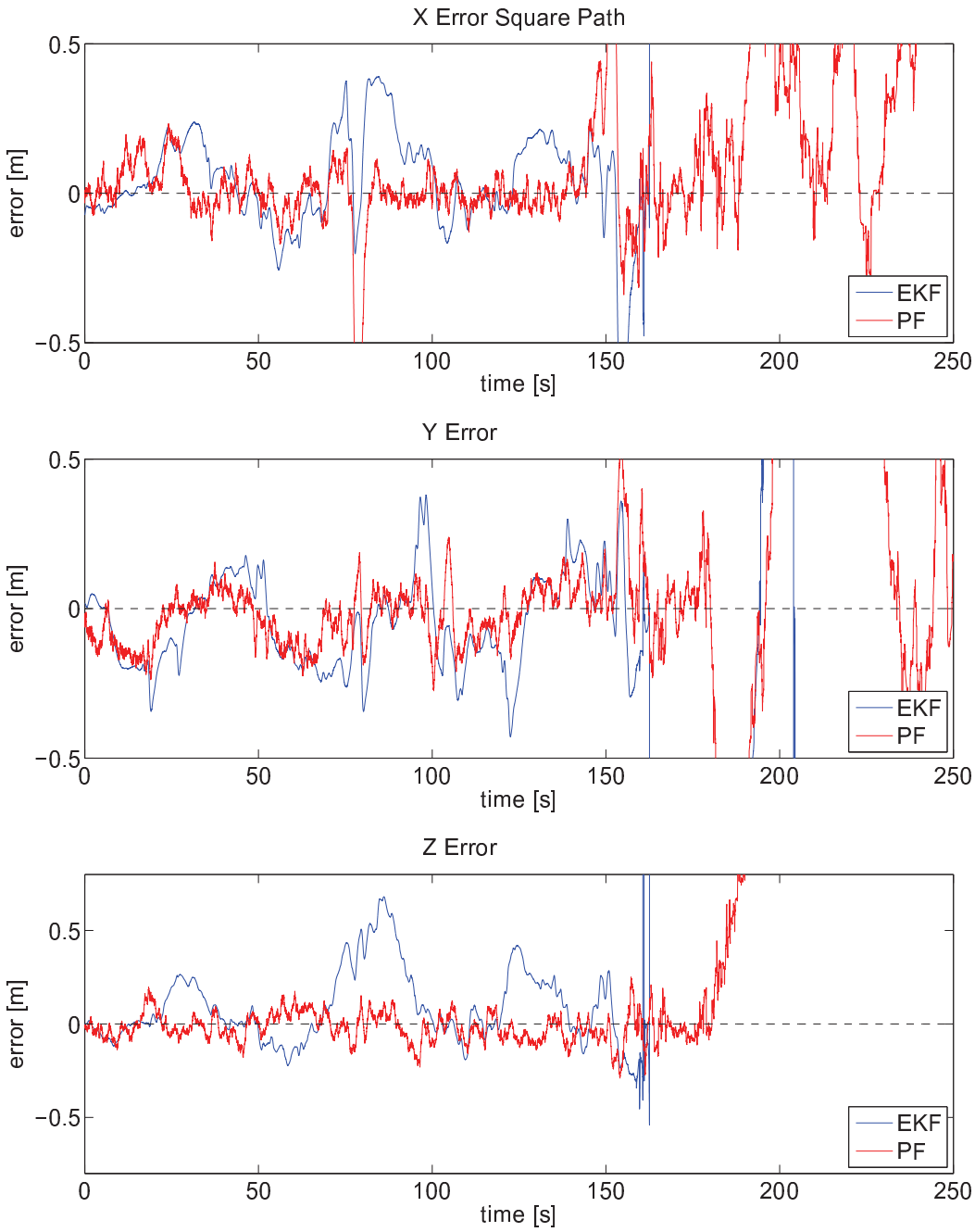


FIGURE 10.11: Positions errors in the square path test. The OWMR lose tracking of the UAV as the errors increment, the PF estimator is more robust to the EKF.

10.4 Real time solvability

The closed form formula for solvability, Eq. (7.19), allows for the calculation of SM in real time. The real time testing of Section 10.2 was used to calculate the SM in real time. In Fig. 10.12 and 10.13 the results are presented. The size of the map calculated defines the speed of the calculation.

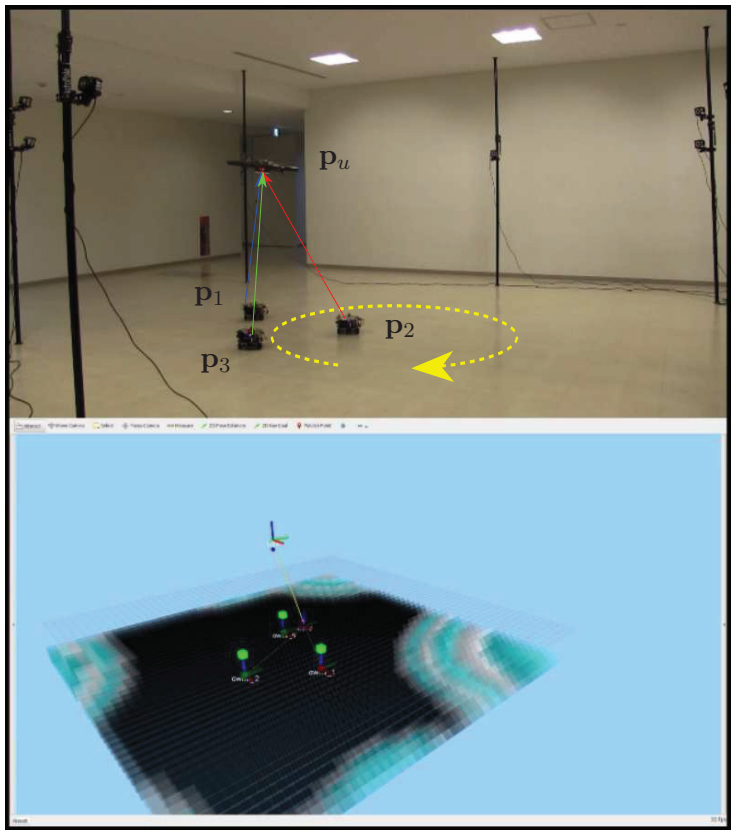


FIGURE 10.12: Real time calculation of solvability maps (SM) 1. Experiment result for an SM in a circular path motion with a triangular OWMR formation. Dark areas have higher solvability.

SM are calculated in real time, and although there is a small delay in the calculation, it all depends on the CPU capabilities and the size of the map required. Even do, it is fast enough to detect non solvable areas and take the pertinent actions.

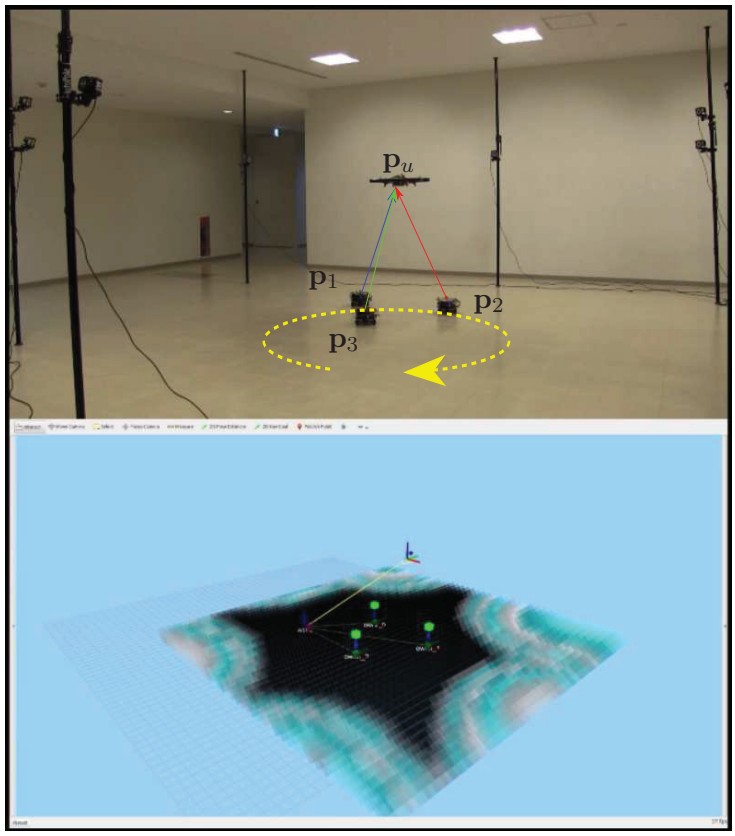


FIGURE 10.13: Real time calculation of solvability maps (SM) 2. Experiment result for an SM in a circular path motion with a triangular OWMR formation. Dark areas have higher solvability.

10.5 Conclusions

The concepts of this thesis were taken from theory into a real time application using real robots, in a multi robot environment. It was demonstrated that only the measurements from positions $\mathbf{p}_i$ and range measurements l_{iu} are required to perform the multi robot three dimensional localization of the UAV in real time. All the algorithms were validated and the localization was compared against a ground truth provided by an Optitrack Motion Camera System. By using other kind of robots, e.g., legged or all terrain vehicles this localization approach can be used in zones of difficult access for humans, and this can be of great aid to the UAV performance.

11 Conclusions

This work presented a three dimensional multi robot localization of an UAV in real time by a mobile multi robot platform consisting of a group of OWMR. The system was conceptualized as a tool for indoor environments where localization techniques still find it difficult to localize object due to occlusion to GPS readings or the unfamiliarity of unexplored environments like disaster areas. Also, taken into account the fact that sensor devices mounted on an aerial robots diminish the power capabilities of flying devices, this localization approach can be of great aid for an UAV performance. The problem was envisioned as a platform that does not depend on a previous setup to work, it is "ready to go", and thus, useful on disaster scenarios.

The basics of the problem where revised and derived. Trilateration was chosen as the mathematical method to solve the problem, allowing any kind of robot to be used with the system, regarding the kinematics involved in the robot motion. As long as the robot can measure the requirements for the problem, positions $\mathbf{p}_i$ and distances l_{iu}, it can be employed and the orientation is not an issue from this approach. With this in mind the selection of the type of robot depends entire on the situation. This method can be applied successfully in three dimensions for indoor scenarios and it can be implemented in a mobile multi robot formation. The robots chosen as platform for this task were OWMR because their dynamics simplify many of the tasks and they are convenient for research purposes.

The propagation of errors was derived from positions $\mathbf{p}_i$ and distances l_{iu} to the covariance of the position $\mathbf{p}_u$ of the UAV, being these sources independent measurements. As the errors can be represented by Gaussian noise and there are tools that permit to calculate the effects of this disturbances in the final measurement. The apparent increase of the error was explained as the DOP. The propagation was built by means of the covariances in the measurements using the dynamic model of the localization algorithm. How this traduces from concepts, to the mathematical requirements to calculate the solution was explained. The covariance in $\mathbf{p}_u$, $\mathbf{Q}_t$, can be measured using metrics as the DOP or SEP, being the later more accurate as it takes in account all the elements in the matrix defining the error in $\mathbf{Q}_t$. This error can be used to select robots within a formation and that was shown in numerical simulated noisy environments.

The rate of failure of trilateration was defined as a map, called Solvability Map, or SM. This map is can be useful to select formation shapes or the direction of motion. As a first procedure it was opted to use a Montecarlo execution, but it was proven that a faster, more accurate and efficient method can be calculated using probability theory by means of the derived probability function of the intersections of two spheres. The solvability analysis was carried on for different formation shapes and different number of robots, showing that if the number of robots increases so does the area were the localization problem is feasible. It was shown that the regions were the trilateration problem is computable can be known. This regions lay along the axis of the triangle formed by the base plane A_b. The solvability can be used to select the shape of the multi robot formation or to design paths for the tracked object.

The robotic mobile platform presented is capable to calculate the position of any external object, not only an UAV and the results are intended to aid in the localization of objects in indoor environments. To put the concepts explained in this work to practice requires the understanding of the whole system integration and the interaction of its different parts. Concepts like SM are defined for the first time in this thesis and are intended to give more insight on how to handle properties of trilateration for multi robot formations. Also, this project can serve as an explanatory guide for the mathematical tools explained in its chapters. This system is flexible as different estimation techniques can be easily implemented if the goal of the mission requires it. Here an EKF and a PF were studied and their differences compared. All the tools required to solve the problem were developed keeping in mind that the system had to be reliable and feasible with the technologies currently available, but as technology improves this methodology with also improve giving better results.

The theory was validated with simulations and real time experiments. The simulations paved the road to understand the requirements for the real time implementation of the multi robot localization system. It was shown how a multi robot formation can handle data transmissions in a multi robot environment and it was stated that robots need to be independent but at the same time they have to work together to achieve a common goal. The achievements of this thesis landed in two journals and one conference papers, showing the interest of the robotics community in this research and the abilities of the author to carry on a scientific investigation. All the concepts where studied from previous researches to novel concepts deployed in this thesis, from theoretical concepts that can be found on books related to this subject to the real time validation.

With the aid of new achievements in SLAM and sensing devices this system can be incorporated to many robotic fields, like perception and slam, to improve the results for localization problems. In the future, a multi robot localization mobile platform can be employed for surveillance and data analysis in zones of difficult access or with little opportunity to use extra sensors in UAV devices. As interaction between robots inside a formation were defined and taken into account, this work is also the base for future researches with multi robot formations.

A Caley-Menger Determinants

Having two sets of points $[\mathbf{p}_1, \ldots, \mathbf{p}_n]$ and $[\mathbf{q}_1, \ldots, \mathbf{q}_n]$, its Caley-Menger bi-determinant is:

$$D(\mathbf{p}_1, \ldots, \mathbf{p}_n; \mathbf{q}_1, \ldots, \mathbf{q}_n) = 2\left(\frac{-1}{2}\right)^n \begin{vmatrix} 0 & 1 & \cdots & 1 \\ 1 & D(\mathbf{p}_1, \mathbf{q}_1) & \cdots & D(\mathbf{p}_1, \mathbf{q}_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & D(\mathbf{p}_n, \mathbf{q}_1) & \cdots & D(\mathbf{p}_n, \mathbf{q}_n) \end{vmatrix} \tag{A.1}$$

If $\mathbf{p}$ and $\mathbf{q}$ are the same sequence, then:

$$D(\mathbf{p}_1, \ldots, \mathbf{p}_n; \mathbf{p}_1, \ldots, \mathbf{p}_n) = D(\mathbf{p}_1, \ldots, \mathbf{p}_n) \tag{A.2}$$

A Caley-Menger determinant has some useful geometric meanings for the distance d, area A and volume V:

$$D(\mathbf{p}_1, \mathbf{p}_2) = d(\mathbf{p}_1, \mathbf{p}_2)^2 \tag{A.3}$$

$$D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = 4A^2 \tag{A.4}$$

$$D(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = 36V^2 \tag{A.5}$$

In trilateration the following relation is very useful:

$$D(\mathbf{p}_i, \mathbf{p}_u) = l_{iu}^2 \tag{A.6}$$

In [15] some factorizations for $D(\cdot)$ can be found.

The following notation will be used to represent the Caley-Menger determinants:

$$D(\mathbf{p}_a, \mathbf{p}_b) = D_{ab} \tag{A.7}$$

$$D(\mathbf{p}_a, \mathbf{p}_b; \mathbf{p}_a, \mathbf{p}_c) = D_{ab;ac} \tag{A.8}$$

In order to solve Eq. 3.6 and Eq. 3.7, the Caley-Menger determinant of different sequences are needed, but following Eq. A.1 it can be noticed that the elements of the determinants are represented by distances, so:

$$D_{123;134} = -\frac{1}{4}\begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & D_{13} & l_{14}^2 \\ 1 & D_{12} & D_{23} & l_{24}^2 \\ 1 & D_{13} & 0 & l_{34}^2 \end{vmatrix} \tag{A.9}$$

$$D_{123;124} = -\frac{1}{4}\begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & D_{12} & l_{14}^2 \\ 1 & D_{12} & 0 & l_{24}^2 \\ 1 & D_{13} & D_{23} & l_{34}^2 \end{vmatrix} \tag{A.10}$$

B Jacobians of g^p

G_t^p is the Jacobian of (4.21) with respect of the state:

$$G_t^p = \frac{\partial g^p(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} = \begin{bmatrix} 1 & 0 & G_x^p \delta_t \\ 0 & 1 & G_y^p \delta_t \\ 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

where δ_t is the time step, G_x^p is:

$$\begin{aligned} G_x^p = (& v_3'(b_1 \cos(\psi) - \sqrt{3}/3(b_1 + 2b_2)\sin(\psi)) \\ & + v_2'(b_1 \cos(\psi) + \sqrt{3}/3(b_1 + 2b_3)\sin(\psi)) \\ & - v_1'((b_2 + b_3)\cos(\psi) + \sqrt{3}/3(b_2 - b_3)\sin(\psi)))/(b_1 + b_2 + b_3) \end{aligned} \tag{B.2}$$

and G_y^p is:

$$\begin{aligned} G_y^p = (& v_3'(b_1 \sin(\psi) + \sqrt{3}/3(b_1 + 2b_2)\cos(\psi)) \\ & + v_2'(b_1 \sin(\psi) - \sqrt{3}/3(b_1 + 2b_3)\cos(\psi)) \\ & - v_1'((b_2 + b_3)\sin(\psi) - \sqrt{3}/3(b_2 - b_3)\cos(\psi)))/(b_1 + b_2 + b_3) \end{aligned} \tag{B.3}$$

V_t^p is the Jacobian of (4.21) with respect of the input:

$$V_t^p = \frac{\partial g^p(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{u}_t} = \frac{B'}{b_1 + b_2 + b_3}(\begin{bmatrix} G_1' & G_2' & G_3' \end{bmatrix} + \delta_t \begin{bmatrix} G' & G' & G' \end{bmatrix})\delta_t \tag{B.4}$$

where B' is:

$$B' = \begin{bmatrix} \cos(\psi') & -\sin(\psi') & 0 \\ \sin(\psi') & \cos(\psi') & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

with ψ' given by:

$$\psi' = \psi + \frac{\text{enc}_1 c_m c_{UMB1} + \text{enc}_2 c_m c_{UMB2} + \text{enc}_3 c_m c_{UMB3}}{b_1 + b_2 + b_3}\delta_t \tag{B.6}$$

G'_1, G'_2 and G'_3 are defined as:

$$G'_1 = \begin{bmatrix} \frac{\sqrt{3}}{3}(b_2 - b_3), & b_2 + b_3, & 1 \end{bmatrix}^\intercal \tag{B.7}$$

$$G'_2 = \begin{bmatrix} -\frac{\sqrt{3}}{3}(b_1 + 2b_3), & -b_1, & 1 \end{bmatrix}^\intercal \tag{B.8}$$

$$G'_3 = \begin{bmatrix} \frac{\sqrt{3}}{3}(b_1 + 2b_2), & -b_1, & 1 \end{bmatrix}^\intercal \tag{B.9}$$

And G' is calculated as:

$$\begin{aligned} G' = \frac{1}{b_1 + b_2 + b_3}(&\text{enc}_3 c_m c_{UMB3} \begin{bmatrix} b_1 \\ \frac{\sqrt{3}}{3}(b_1 + 2b_2) \\ 0 \end{bmatrix} \\ &+ \text{enc}_2 c_m c_{UMB2} \begin{bmatrix} b_1 \\ \frac{\sqrt{3}}{3}(-b_1 - 2b_3) \\ 0 \end{bmatrix} \\ &- \text{enc}_1 c_m c_{UMB1} \begin{bmatrix} b_2 + b_3 \\ \frac{\sqrt{3}}{3}(b_3 - b_2) \\ 0 \end{bmatrix}) \end{aligned} \tag{B.10}$$

C Derivation of Partials of H_{zt}

To solve (6.5), six partial derivatives must be solved. Three are partials of vectors: $\mathbf{p}_1$, $\mathbf{v}_{12}$ and $\mathbf{v}_{13}$, which are matrices in $\mathbb{R}^{3\times12}$. And three are partials of scalar values: k_1, k_2 and k_3, which are vectors in $\mathbb{R}^{1\times12}$. In all the equations presented:

$$\mathbf{v}_{ab} = \mathbf{p}_b - \mathbf{p}_a \tag{C.1}$$

The following notation will be used to represent the Caley-Menger determinants:

$$D(\mathbf{p}_a, \mathbf{p}_b) = D_{ab} \tag{C.2}$$
$$D(\mathbf{p}_a, \mathbf{p}_b; \mathbf{p}_a, \mathbf{p}_c) = D_{ab;ac} \tag{C.3}$$

The partials of vectors are:

$$\frac{\partial \mathbf{p}_1}{\partial \mathbf{q}_z} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C.4}$$
$$\frac{\partial \mathbf{v}_{12}}{\partial \mathbf{q}_z} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C.5}$$
$$\frac{\partial \mathbf{v}_{13}}{\partial \mathbf{q}_z} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{C.6}$$

where $\mathbf{I}$ and $\mathbf{0}$ are the identity and null matrix in $\mathbb{R}^{3\times3}$. Now, the partial of the scalars k_i, $i \in 1, 2, 3$, can be written in matrix form as:

$$\frac{\partial k_i}{\partial \mathbf{q}_z} = \begin{bmatrix} \frac{\partial k_i}{\partial \mathbf{p}_1} & \frac{\partial k_i}{\partial \mathbf{p}_2} & \frac{\partial k_i}{\partial \mathbf{p}_3} & \frac{\partial k_i}{\partial \mathbf{l}} \end{bmatrix} \tag{C.7}$$

$$\mathbf{q}_z = \begin{bmatrix} \mathbf{p}_1^\intercal & \mathbf{p}_2^\intercal & \mathbf{p}_3^\intercal & l_{1u} & l_{2u} & l_{3u} \end{bmatrix}^\intercal = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{l} \end{bmatrix} \tag{C.8}$$

Using (3.6), (3.7) and (3.8), the partials of k_i can be obtained.

$$\frac{\partial k_1}{\partial \mathbf{p}_i} = -\frac{1}{D_{123}}\frac{\partial D_{123;13u}}{\partial \mathbf{p}_i} + \frac{D_{123;13u}}{D_{123}^2}\frac{\partial D_{123}}{\partial \mathbf{p}_i} \tag{C.9}$$

$$\frac{\partial k_2}{\partial \mathbf{p}_i} = \frac{1}{D_{123}}\frac{\partial D_{123;12u}}{\partial \mathbf{p}_i} - \frac{D_{123;12u}}{D_{123}^2}\frac{\partial D_{123}}{\partial \mathbf{p}_i} \tag{C.10}$$

$$\frac{\partial k_3}{\partial \mathbf{p}_i} = \frac{1}{2D_{123}\sqrt{D_{123u}}}\frac{\partial D_{123u}}{\partial \mathbf{p}_i} - \frac{\sqrt{D_{123u}}}{D_{123}^2}\frac{\partial D_{123}}{\partial \mathbf{p}_i} \tag{C.11}$$

Then, it is only needed to calculate the partials $\partial D_{123}/\partial \mathbf{p}_i$, $\partial D_{123u}/\partial \mathbf{p}_i$, $\partial D_{123;13u}/\partial \mathbf{p}_i$ and $\partial D_{123;12u}/\partial \mathbf{p}_i$. For that, the following partials are defined:

$$\frac{\partial D_{ab}}{\partial \mathbf{p}_a} = \frac{\partial \|\mathbf{p}_b - \mathbf{p}_a\|^2}{\partial \mathbf{p}_a} = -2\mathbf{v}_{ab} \tag{C.12}$$

$$\frac{\partial D_{ab}}{\partial \mathbf{p}_b} = \frac{\partial \|\mathbf{p}_b - \mathbf{p}_a\|^2}{\partial \mathbf{p}_b} = 2\mathbf{v}_{ab} \tag{C.13}$$

In all the equations:

$$A_b = \frac{\sqrt{D_{123}}}{2} \tag{C.14}$$

$$V = \frac{\sqrt{D_{123u}}}{6} \tag{C.15}$$

The partials $\partial D_{123}/\partial \mathbf{p}_i$ are:

$$\frac{\partial D_{123}}{\partial \mathbf{p}_1} = D_{12}(-\mathbf{v}_{13} + \mathbf{v}_{12}) + D_{13}(-\mathbf{v}_{12} + \mathbf{v}_{13}) + D_{23}(-\mathbf{v}_{12} - \mathbf{v}_{13}) \tag{C.16}$$

$$\frac{\partial D_{123}}{\partial \mathbf{p}_2} = D_{12}(-\mathbf{v}_{23} - \mathbf{v}_{12}) + D_{23}(\mathbf{v}_{12} + \mathbf{v}_{23}) + D_{13}(-\mathbf{v}_{23} + \mathbf{v}_{12}) \tag{C.17}$$

$$\frac{\partial D_{123}}{\partial \mathbf{p}_3} = D_{23}(\mathbf{v}_{13} - \mathbf{v}_{23}) + D_{13}(\mathbf{v}_{23} - \mathbf{v}_{13}) + D_{12}(\mathbf{v}_{23} + \mathbf{v}_{13}) \tag{C.18}$$

$\partial D_{123u}/\partial \mathbf{p}_i$ derivations are as follows:

$$\begin{aligned}\frac{\partial D_{123u}}{\partial \mathbf{p}_1} = \frac{1}{2}((&-(l_{1u}^2 l_{3u}^2 + l_{2u}^2 l_{3u}^2 - l_{1u}^2 l_{2u}^2 - l_{3u}^4) \\ &+ 2D_{12}l_{3u}^2 + D_{13}D_{23} - D_{13}(l_{2u}^2 + l_{3u}^2) - D_{23}(l_{1u}^2 + l_{3u}^2))\mathbf{v}_{12} \\ &+ (-(l_{1u}^2 l_{2u}^2 + l_{2u}^2 l_{3u}^2 - l_{1u}^2 l_{3u}^2 - l_{2u}^4) \\ &+ 2D_{13}l_{2u}^2 + D_{12}D_{23} - D_{12}(l_{2u}^2 + l_{3u}^2) - D_{23}(l_{1u}^2 + l_{2u}^2))\mathbf{v}_{13})\end{aligned} \tag{C.19}$$

$$\frac{\partial D_{123u}}{\partial \mathbf{p}_2} = \frac{1}{2}(((l_{2u}^2 l_{3u}^2 + l_{1u}^2 l_{3u}^2 - l_{2u}^2 l_{1u}^2 - l_{3u}^4) - 2D_{12}l_{3u}^2 - D_{23}D_{13} + D_{23}(l_{1u}^2 + l_{3u}^2) + D_{13}(l_{2u}^2 + l_{3u}^2))\mathbf{v}_{12} + (-(l_{2u}^2 l_{1u}^2 + l_{1u}^2 l_{3u}^2 - l_{2u}^2 l_{3u}^2 - l_{1u}^4) + 2D_{23}l_{1u}^2 + D_{12}D_{13} - D_{12}(l_{1u}^2 + l_{3u}^2) - D_{13}(l_{2u}^2 + l_{1u}^2))\mathbf{v}_{23}) \quad \text{(C.20)}$$

$$\frac{\partial D_{123u}}{\partial \mathbf{p}_3} = \frac{1}{2}(((l_{3u}^2 l_{2u}^2 + l_{1u}^2 l_{2u}^2 - l_{1u}^2 l_{3u}^2 - l_{2u}^4) - 2D_{13}l_{2u}^2 - D_{23}D_{12} + D_{23}(l_{1u}^2 + l_{2u}^2) + D_{12}(l_{3u}^2 + l_{2u}^2))\mathbf{v}_{13} + ((l_{3u}^2 l_{1u}^2 + l_{1u}^2 l_{2u}^2 - l_{2u}^2 l_{3u}^2 - l_{1u}^4) - 2D_{23}l_{1u}^2 - D_{13}D_{12} + D_{13}(l_{1u}^2 + l_{2u}^2) + D_{12}(l_{3u}^2 + l_{1u}^2))\mathbf{v}_{23}) \quad \text{(C.21)}$$

$\partial D_{123;13u}/\partial \mathbf{p}_i$ are calculated as:

$$\frac{\partial D_{123;13u}}{\partial \mathbf{p}_1} = \frac{1}{2}((l_{3u}^2 - l_{1u}^2 + D_{13})\mathbf{v}_{12} + (l_{1u}^2 - 2l_{2u}^2 + l_{3u}^2 - 2D_{13} + D_{12} + D_{23})\mathbf{v}_{13}) \quad \text{(C.22)}$$

$$\frac{\partial D_{123;13u}}{\partial \mathbf{p}_2} = \frac{1}{2}((l_{1u}^2 - l_{3u}^2 - D_{13})\mathbf{v}_{12} + (l_{1u}^2 - l_{3u}^2 + D_{13})\mathbf{v}_{23}) \quad \text{(C.23)}$$

$$\frac{\partial D_{123;13u}}{\partial \mathbf{p}_3} = \frac{1}{2}((2l_{2u}^2 - l_{1u}^2 - l_{3u}^2 + 2D_{13} - D_{12} - D_{23})\mathbf{v}_{13} + (l_{3u}^2 - l_{1u}^2 - D_{13})\mathbf{v}_{23}) \quad \text{(C.24)}$$

Finally, $\partial D_{123;12u}/\partial \mathbf{p}_i$ are:

$$\frac{\partial D_{123;12u}}{\partial \mathbf{p}_1} = \frac{1}{2}((2l_{3u}^2 - l_{1u}^2 - l_{2u}^2 - D_{13} - D_{23} + 2D_{12})\mathbf{v}_{12} + (l_{1u}^2 - l_{2u}^2 - D_{12})\mathbf{v}_{13}) \quad \text{(C.25)}$$

$$\frac{\partial D_{123;12u}}{\partial \mathbf{p}_2} = \frac{1}{2}((l_{1u}^2 + l_{2u}^2 - 2l_{3u}^2 - 2D_{12} + D_{13} + D_{23})\mathbf{v}_{12} + (l_{2u}^2 - l_{1u}^2 - D_{12})\mathbf{v}_{23}) \quad \text{(C.26)}$$

$$\frac{\partial D_{123;12u}}{\partial \mathbf{p}_3} = \frac{1}{2}((-l_{1u}^2 + l_{2u}^2 + D_{12})\mathbf{v}_{13} + (l_{1u}^2 - l_{2u}^2 + D_{12})\mathbf{v}_{23}) \quad \text{(C.27)}$$

$\partial k_i/\partial \mathbf{l}$ are defined in [15], they are:

$$\frac{\partial k_1}{\partial \mathbf{l}} = \begin{bmatrix} \frac{\partial k_1}{\partial l_{1u}} & \frac{\partial k_1}{\partial l_{2u}} & \frac{\partial k_1}{\partial l_{3u}} \end{bmatrix} = \frac{1}{4A_b^2}\begin{bmatrix} l_{1u}D_{23;13}, -l_{2u}D_{13}, l_{3u}D_{12;13} \end{bmatrix} \quad \text{(C.28)}$$

$$\frac{\partial k_2}{\partial \mathbf{l}} = \begin{bmatrix} \frac{\partial k_2}{\partial l_{1u}} & \frac{\partial k_2}{\partial l_{2u}} & \frac{\partial k_2}{\partial l_{3u}} \end{bmatrix} = \frac{1}{4A_b^2}\begin{bmatrix} -l_{1u}D_{23;12}, l_{2u}D_{13;12}, -l_{3u}D_{12} \end{bmatrix} \quad \text{(C.29)}$$

$$\frac{\partial k_3}{\partial \mathbf{l}} = \begin{bmatrix} \frac{\partial k_3}{\partial l_{1u}} & \frac{\partial k_3}{\partial l_{2u}} & \frac{\partial k_3}{\partial l_{3u}} \end{bmatrix} = \frac{1}{24A_b^2V}\begin{bmatrix} -l_{1u}D_{123;32u}, -l_{2u}D_{123;13u}, -l_{3u}D_{123;21u} \end{bmatrix} \quad \text{(C.30)}$$

D Derived Probability Distributions

If a process has various outcomes for a given event, then those outcomes can be presented as a r.v., which is a function whose domain is the set of all the outcomes. A widely used probability function is the Gaussian distribution[1], which can be seen in natural processes and is commonly used as noise for dynamic environments. A Gaussian distribution is defined by its mean and covariance:

$$f_X = f_X(x) = (2\pi\sigma)^{-\frac{1}{2}} \exp(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}) = \mathcal{N}(\mu, \sigma^2) \tag{D.1}$$

This PDF has a CDF given by:

$$F_X = F_X(x) = \int_{-\infty}^{\infty} f_X dx \tag{D.2}$$

If x is used in a function $g(x)$, then the effects of the probability distribution of x, f_X, can be translated into $g(x)$.

$$f_X(g(x)) = \frac{f_X(x_1)}{|g'(x_1)|} + \frac{f_X(x_2)}{|g'(x_2)|} + \cdots + \frac{f_X(x_n)}{|g'(x_n)|} \tag{D.3}$$

where $\{x_1, x_2, \cdots, x_n\}$ are the roots of $g(x)$. Now, if two functions $g(x, y)$ and $h(x, y)$ were to be functions of two r.v. x and y, the joint density function is given by:

$$f_{XY}(g(x,y), h(x,y)) = \frac{f_{XY}(x_1, y_1)}{|J(x_1, y_1)|} + \frac{f_{XY}(x_2, y_2)}{|J(x_2, y_2)|} + \cdots + \frac{f_{XY}(x_n, y_n)}{|J(x_n, y_n)|} \tag{D.4}$$

where $J(x, y)$ is the Jacobian:

$$J(x,y) = \begin{vmatrix} \frac{\partial g(x,y)}{\partial x} & \frac{\partial g(x,y)}{\partial y} \\ \frac{\partial h(x,y)}{\partial x} & \frac{\partial h(x,y)}{\partial y} \end{vmatrix} \tag{D.5}$$

The concepts of Eq. (D.4) and Eq. (D.5) can be expanded to more than two dimensions.

[1] A Gaussian distribution is also called normal distribution.

Bibliography

[1] S. Magariyama, H. Matsumoto, L. Ruiz, and Z. Wang, "Estimation of a feasible dynamic caging zone for multirobot object transportation", in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, 2014, pp. 508–514.

[2] L. Ruiz and Z. Wang, "System design for the localization of an uav in 3d using a mobile multi-robot platform", *JOURNAL OF JAPAN SOCIETY FOR DESIGN ENGINEERING*, vol. 52, no. 2, pp. 81–97, 2017.

[3] ——, "Error propagation and solvability in uav localization based on trilateration with a multi robot formation", *Int. J. of Mechatronics and Automation*, Accepted, In Print, 2017.

[4] ——, "Real time multi robot 3d localization system using trilateration", in *IEEE International Conference on Robotics and Biomimetics, 2016, International Conference on*, 2016, pp. 1510–1515.

[5] A. Shareef and Y. Zhu, *Localization using extended kalman filters in wireless sensor networks*. Kalman Filter Recent Advances, Applications, pp. 297-320, Victor M. Moreno, and Alberto Pigazo (Ed.)., 2009, ISBN: 9789533070001.

[6] V. Pierlot and M. V. Droogenbroeck, "A new three object triangulation algorithm for mobile robot positioning", *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 566–577, 2014, ISSN: 1552-3098.

[7] R. I. Hartley and P. Sturm, "Triangulation", *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[8] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]", *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011, ISSN: 1070-9932.

[9] Nister, "Preemptive ransac for live structure and motion estimation", in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, 199–206 vol.1.

[10] D. Nister, "An efficient solution to the five-point relative pose problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004, ISSN: 0162-8828.

[11] D. H. Stewénius, C. Engels, and D. D. Nistér, "Recent developments on direct relative orientation", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[12] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation", in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 2969–2976.

[13] T. Eren, O. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization", in

INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4, 2004, 2673–2684 vol.4.

[14] K.-W. Lee, J.-B. Park, and B.-H. Lee, "Dynamic localization with hybrid trilateration for mobile robots in intelligent space", *Intelligent Service Robotics*, vol. 1, no. 3, pp. 221–235, 2008, ISSN: 1861-2784.

[15] F. Thomas and L. Ros, "Revisiting trilateration for robot localization", *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 93–101, 2005, ISSN: 1552-3098.

[16] D. E. Manolakis, "Efficient solution and performance analysis of 3-d position estimation by trilateration", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1239–1248, 1996, ISSN: 0018-9251.

[17] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization", in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 3415–3422.

[18] R. B. Langley, "Dilution of precision", *GPS World*, 1999.

[19] A. Kelly, "Precision dilution in triangulation based mobile robot position estimation", in *In Intelligent Autonomous Systems*, 2003.

[20] F. Kaya and S. M., "A computer simulation of dilution of precision in the global positioning system using matlab", in *Proceedings of the 4th International Conference on Electrical and Electronic Engineering*, 2005.

[21] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization", *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2010, ISSN: 1045-9219.

[22] M. M. Zaniani, A. M. Shahar, and I. A. Azid, "Trilateration target estimation improvement using new error correction algorithm", in *2010 18th Iranian Conference on Electrical Engineering*, 2010, pp. 489–494.

[23] H. Akcan and C. Evrendilek, "Reducing the number of flips in trilateration with noisy range measurements", in *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Acess*, ser. MobiDE '13, New York, New York: ACM, 2013, pp. 20–27, ISBN: 978-1-4503-2197-6.

[24] C. Evrendilek and H. Akcan, "On the complexity of trilateration with noisy range measurements", *IEEE Communications Letters*, vol. 15, no. 10, pp. 1097–1099, 2011, ISSN: 1089-7798.

[25] J. V. Kleef, J. Bergmans, L. Kester, and F. Groen, "Multiple-hypothesis trilateration and tracking with distributed radars", in *2006 9th International Conference on Information Fusion*, 2006, pp. 1–7.

[26] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics", in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, 1987, pp. 850–850.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics (intelligent robotics and autonomous agents)*. The MIT Press, 2005, ISBN: 0262201623.

[28] J. Dunik and M. Simandl, "Estimation of state and measurement noise covariance matrices by multi-step prediction", *{IFAC} Proceedings Volumes*, vol. 41, no. 2, pp. 3689 –3694, 2008, 17th {IFAC} World Congress, ISSN: 1474-6670.

[29] D. E. Manolakis and M. E. Cox, "Effect in range difference position estimation due to stations' position errors", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 329–334, 1998, ISSN: 0018-9251.

[30] A. Bahr and J. J. Leonard, "Minimizing trilateration errors in the presence of uncertain landmark positions.", in *EMCR*, 2007.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system", *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015, ISSN: 1552-3098.

[32] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 15–22.

[33] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i", *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006, ISSN: 1070-9932.

[34] R. T. Vaughan, G. S. Sukhatme, F. J. Mesa-Martinez, and J. F. Montgomery, "Fly spy: Lightweight localization and target tracking for cooperating air and ground robots", in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds. Tokyo: Springer Japan, 2000, pp. 315–324, ISBN: 978-4-431-67919-6.

[35] P. R. Tony Stentz Alonzo Kelly and O. Amidi, "Integrated air/ground vehicle system for semi-autonomous off-road navigation", in *Robotics Institute Technical Report, AUVSI Symposium*, 2002, Paper 18.

[36] B. Grocholsky, S. Bayraktar, V. Kumar, C. J. Taylor, and G. J. Pappas, "Synergies in feature localization by air-ground robot teams", in *Experimental Robotics IX, The 9th International Symposium on Experimental Robotics [ISER 2004, Singapore, 18.-21. June 2004]*, 2004, pp. 352–361.

[37] R. Kurazume and S. Hirose, "Development of a cleaning robot system with cooperative positioning system", *Autonomous Robots*, vol. 9, no. 3, pp. 237–246, 2000, ISSN: 1573-7527.

[38] R. K. S. Oshima Y. Iwashita, "Development of 3d scanning system using multiple robots and a precise laser scanner", in *JSME Conference on Robotics and Mechatronics on*, 2015.

[39] K. M. Y. Kasuya F. Hashikawa, "Development of an information sharing system among distributed sensors and mobile robots for construction of intelligent spaces", in *JSME Conference on Robotics and Mechatronics on*, 2015.

[40] A. Howard, G. S. Sukhatme, and M. J. Mataric, "Multirobot simultaneous localization and mapping using manifold representations", *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1360–1369, 2006, ISSN: 0018-9219.

[41] R. Khan, S. U. Khan, S. Khan, and M. U. A. Khan, "Localization performance evaluation of extended kalman filter in wireless sensors network", *Procedia Computer Science*, vol. 32, pp. 117 –124, 2014, The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), ISSN: 1877-0509.

[42] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]", *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012, ISSN: 1053-5888.

[43] B. M. Bell and F. W. Cathey, "The iterated kalman filter update as a gauss-newton method", *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993, ISSN: 0018-9286.

[44] F. Gustafsson, "Particle filter theory and practice with positioning applications", *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010, ISSN: 0885-8985.

[45] K. Xu and P. Song, "A coordination framework for weakly centralized mobile robot teams", in *The 2010 IEEE International Conference on Information and Automation*, 2010, pp. 77–82.

[46] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "Gps/imu data fusion using multisensor kalman filtering: Introduction of contextual aspects", *Information Fusion*, vol. 7, no. 2, pp. 221 –230, 2006, ISSN: 1566-2535.

[47] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, "Robust localization using relative and absolute position estimates", in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, 1134–1140 vol.2.

[48] S. Mau, *What is the kalman filter and how can it be used for data fusion*, Robotics Math 16-811, Dec. 2005.

[49] D. M. Mayhew, "Multi-rate sensor fusion for gps navigation using kalman filtering", Master's thesis, Virginia Tech, Apr. 1999.

[50] L. G. Stephen J., "Development of a gnss-based multi-sensor vehicle navigation system", in *Proceedings of the 2000 National Technical Meeting of The Institute of Navigation*, Anaheim, CA, 2000, pp. 268–278.

[51] P. Eberhard and Q. Tang, "Sensor data fusion for the localization and position control of one kind of omnidirectional mobile robots", in *Multibody System Dynamics, Robotics and Control*, H. Gattringer and J. Gerstmayr, Eds. Vienna: Springer Vienna, 2013, pp. 45–73, ISBN: 978-3-7091-1289-2.

[52] J. Gonçalves, J. Lima, and P. Costa, "Real time tracking of an omnidirectional robot - an extended kalman filter approach.", in *ICINCO-RA (2)*, J. Filipe, J. Andrade-Cetto, and J.-L. Ferrier, Eds., 978-989-8111-31-9, INSTICC Press, 2008, pp. 5–10.

[53] R. Rojas and A. G. Forster, "Holonomic control of a robot with an omni-directional drive", *KI Kunstliche Intelligenz*, 2006.

[54] Y. Liu, J. J. Zhu, R. L. W. II, and J. Wu, "Omni-directional mobile robot controller based on trajectory linearization", *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 461 –479, 2008, ISSN: 0921-8890.

[55] G. Antonelli, S. Chiaverini, and G. Fusco, "A calibration method for odometry of mobile robots based on the least-squares technique: Theory and experimental validation", *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 994–1004, 2005, ISSN: 1552-3098.

[56] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, 1996, ISSN: 1042-296X.

[57] A. Martinelli and R. Siegwart, "Estimating the odometry error of a mobile robot during navigation", *Autonomous Systems Lab Swiss Federal Institute of Technology Lausanne*, 2003.

[58] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot", in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, 1997, 2783–2788 vol.4.

[59] Y. Maddahi, A. Maddahi, and N. Sepehri, "Calibration of omnidirectional wheeled mobile robots: Method and experiments", *Robotica*, vol. 31, no. 6, 969–980, 2013.

[60] Y. Maddahi, S. M. Hosseini, and N. E. Mastorakis, "How to improve positional accuracy in redundant omnidirectional mobile robots", in *Recent Advances in Electrical Engineering and Educational Technologies*, 2014, pp. 22–26.

[61] R. L. Williams, B. E. Carter, P. Gallina, and G. Rosati, "Dynamic model with slip for wheeled omnidirectional robots", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 285–293, 2002, ISSN: 1042-296X.

[62] D. Chugo, H. Asama, K. Kawabata, H. Kaetsu, and T. Mishima, "Plural wheels control based on slip estimation", in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5558–5563.

[63] D. Chugo, K. Kawabata, H. Kaetsu, H. Asama, T. Mishima, and K. Takase, "Slip reduction control using plural wheel information for step-climbing vehicle", in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, 2009, pp. 2373–2378.

[64] E. W. Weisstein, "Sphere-sphere intersection from mathworld–a wolfram web resource", *[Online]. Available: Http://mathworld.wolfram.com/ Sphere-SphereIntersection.html*, 2015.

[65] E. Olson, "Apriltag: A robust and flexible visual fiducial system", in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[66] G. Bradski, "Opencv", *Dr. Dobb's Journal of Software Tools*, 2000.

[67] A. Papoulis, *Probability, random variables, and stochastic processes*, 1st. McGraw-Hill, 1965, ch. 7, pp. 201–205.

[68] S. J. D. Prince, *Computer vision: Models, learning, and inference*, 1st. New York, NY, USA: Cambridge University Press, 2012, ISBN: 1107011795, 9781107011793.

[69] MATLAB, *Version 8.1.0.604 (r2013a)*. Natick, Massachusetts: The MathWorks Inc., 2013.

[70] Hardkernel, "Odroid", *Http://www.hardkernel.com/*, Accessed 2016.

[71] 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.

[72] NaturalPointInc, "Optitrack", *Http://www.optitrack.com/*, Accessed 2016.