Development of Cooperation Between Flying Robot, Ground Robot and Ground Station with Fuzzy Logic and Image Processing

von der Naturwissenschaftlich-Technischen Fakultät (Fakultät IV)
Department Elektrotechnik und Informatik
der Universität Siegen

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

genehmigte Dissertation
von

M.Sc. Rami Ahmad Al-Jarrah

1. Gutachter : Prof. Dr.-Ing. Hubert Roth
2. Gutachter : Prof. Dr.-Ing. Madjid Fathi

Tag der mündlichen Prüfung: 23 April 2015

der Universität Siegen, 23 April 2015

Abstract

The aerial robots represent an interested and rich area of research because they are very useful to perform complex tasks such as image acquisition of unreachable areas, localization and tracking targets. To develop blimp system that is appropriate in diversity scenarios, an intelligent control with high autonomy degree is required. Thus, the main challenge in context of designing and navigation is the combination between intelligent control framework and small-light hardware components. Therefore, we design blimp robot based on embedded system; then, we present several fuzzy sets models that should deal with autonomous, navigation and visual tracking problems. These models are empirically designed by combining the possibilities distributions theory with fuzzy logic.

On the other hand, the use of cooperative robots system dramatically increases since it is appropriate solution in relation to performance, efficiency and reliability. The development of such cooperation system is one of the most demanding goals in artificial intelligence research and the most challenging issues here are the localization and tracking targets. Thus, this dissertation addresses the problem of tracking robots in parallel with achieving the cooperative behavior based on computer vision system and artificial intelligent control to improve the performance and efficiency of such system. Moreover, the thesis considers the use of computer vision system that can be applied to navigation and tracking missions. This system has been designed to detect and track either ground robots or 3D aerial object. In addition, considering use of wireless sensor network for estimation multi-targets locations and the distances between them is presented. Finally, the approaches that were developed in the thesis have been carried out and evaluated on the blimp robot. The extensive experiments have been done to demonstrate, analyze and validate the capabilities of such system as well as the properties of the proposed algorithms.

Zusammenfassung

Flugroboter repräsentieren ein interessantes und weites Gebiet für Forschungen, da sie sehr vielseitig einsetzbar sind, um komplexe Aufgaben zu erfüllen, wie Bildaufnahmen in schwer zugänglichen Gebieten, Lokalisierung und Verfolgung von Zielen. Bei der Entwicklung eines Luftschiff-Systems, das in einer Vielzahl von Szenarien eingesetzt werden kann, ist eine intelligente Steuerung mit hoher Autonomie von Nöten. Folglich ist die Hauptherausforderung im Kontext des Designs und Navigation die Kombination zwischen einem intelligenten Steuerungssystem und einer kleinen und leichten Hardware. Deshalb entwickeln wir einen Luftschiffroboter basierend auf einem eingebetteten System, dann präsentieren wir verschiedene Fuzzy-Modelle die Probleme der autonomen Navigation und Zielverfolgung lösen. Diese Modelle sind empirisch durch die Kombination von Wahrscheinlichkeitsverteilungen mit Theorien der Fuzzylogik erstellt worden. Andererseits wird durch die Verwendung von kooperativen Robotersystemen die Leistung, Effizienz und Zuverlässigkeit drastisch gesteigert. Die Entwicklung eines solchen kooperativen Robotersystems ist eines der anspruchsvollsten Ziele bei der Erforschung künstlicher Intelligenzen und dabei sind die größten Herausforderungen Lokalisierung und Zielverfolgung. Somit adressiert diese Dissertation das Problem der parallelen Verfolgung von Robotern durch das Erreichen von kooperativen Verhalten basierend auf Bildverarbeitung und Künstlicher Intelligenz, um die Leistung und Effizienz eines solchen Systems zu verbessern. Darüber hinaus betrachtet diese Arbeit, ob Systeme zur Bildverarbeitung bei der Navigation und Zielverfolgung eingesetzt werden können. Dieses System wurde entwickelt um Bodenroboter oder 3D Luftobjekte zu erkennen und zu verfolgen. Außerdem wird eine Methode zur Positionsschätzung von multiplen Zielen mit Hilfe eines drahtlosen Sensornetzwerkes präsentiert. Schließlich wurden die oben erwähnten Ansätze der Arbeit auf ein reales Luftschiff angewendet und getestet. Umfangreiche Experimente wurden durchgeführt, analysiert, um die Fähigkeiten und Eigenschaften des entwickelten Systems und der entwickelten und verwendeten Algorithmen zu validieren.

Acknowledgments

First, and foremost, I am thankful to Allah Almighty for helping me to complete this PhD thesis work. Indeed, it is rare for someone to learn and contribute into new field of science and knowledge. Certainly, this work is not only an example of such research, but it also merely reflects my continuation in a certain direction inspired from my supervisor Prof. Dr.-Ing. Hubert Roth. Before entering Siegen University, I could not have imagined how rich and broad my study under him would be, and how far my education would advance at the hands of such an inspiring professor. Without his vision and strong support over more than three years none of these could have been possible. Studying under him has been an immense privilege. Moreover, I would really appreciate his support during my study and his confidence to allow me attends in several international conferences. I would like to thank my second supervisor Prof. Dr.-Ing. Madjid Fathi for his kindly help.

I am especially grateful to Die Aktionsgemeinschaft zur Forderung wissenschaftlicher Projecte (AFP) in Siegen University for partially financial support a part of my project and especially much thanks to Prof. Dr.-Ing. Thomas Carolus for his support and help.

The colleagues and friends I have made in the academic community over the years have also been a great source of support and stimulation: warm thanks to Dipl. Ing. Peter Sahm, Dipl. Ing. Hille Christof and Aamir Shahzad for their inspiration and discussing and laughing at leisure time. I am grateful to all RST members for discussions, sharing experience and respect during all of these years especially Frau Birgit Hoffmann, Frau Judith Düber and Frau Kristina Borchert. I would like to thanks MSc. Reduane Ait-Jellal, Mr. Hamza Kabori, Dr.-Ing. Mustafa Jaradat for support and encouragement during these years.

Finally and most importantly, I would like to thank all my family for believing in me through the years and providing me with enthusiastic and inspirations; I owe them for who I am now: mother, father, sisters and brothers thanks for all support, love, encouragement and care.

Table of Contents

List of Figures

List of Tables

Notations

The following are acronyms and notations that used throughout this thesis.

AAPI	Allocation for Private Internets.
CDF	Complementary Cumulative Distribution Function.
CRLB	Cramer-Rao lower bound.
GCC	Ground Control Center.
GMP	Generalized Modus Ponens.
GOU	Gondola On-board Unit.
GUI	Graphical User Interface.
HOT	Higher Order Terms.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
IID	Identical Independent Distribution.
IMU	Inertial Measurement Unit.
IPM	Inverse Perspective Mapping.
LAN	Local Area Network.
LKT	Lucas Kanade Technique.
MSE	Mean Square Error.
MU	Main Unit.
OOK	On-Off Keying.
OS	Operating System.
P2P	Peer-to-Peer.
PSO	Practical Swarm Optimization.
PWM	Pulse Width Module.
RANSAC	Random Sample Consensus.
SIFT	Scale Invariant Feature Transform.
SRF	Ultrasonic Sensors.
SURF	Speeded UP Robust Feature algorithm.

TCP	Transmission Control Protocol.
UAVs	Unmanned Aerial Vehicles.
UDP	User Datagram Protocol.
WLAN	Wireless Local Area Network.
WSNs	Wireless sensor networks.

Chapter 1 Introduction

The main objective of this thesis is to present contributions in the field of aerial blimp robot to state the problems of navigation, path planning, detecting, tracking and localization. More precisely, to develop embedded blimp robot system with artificial intelligent control. This control based on fuzzy logic and possibilities distribution due to the fact that some decisions based on possibilistic rather than probabilistic. Thus, the fuzzy variables are associated with the possibility distribution in the same principle as the random variables are associated with a probability distribution. In addition, this thesis proposes vision-based guidance and estimation for the targets tracking. Then, proposed wireless sensor networks for multi-targets tracking in order to estimate the locations and orientation of the blimp robot system and the targets as well as the distance between them. Therefore, this chapter firstly presents the motivation, then, the main objectives of the research carried out. Finally, the outlines and main contributions of the thesis are presented.

1.1. Motivation

Clearly and transparently, the technologies for autonomous Unmanned Aerial Vehicles (UAVs) have dramatically developed and advanced. Thus, the research on aerial autonomous systems has become affordable for many research centers. The UAVs have been mainly used in military applications. Nevertheless, it is obvious also that they have been widely used in civil applications, for instance, used in disaster and emergency situations. Because UAVs have higher mobility and maneuverability respecting to ground robots, they are more ideal and suitable for special missions and tasks. More precisely, UAVs could be very useful to perform complicated missions for example they can be used to provide images data for exploring areas as well as detect, localize and track other targets in the environment.

The most important challenge in this area is to design aerial robot with high degree of autonomy. Despite the small helicopters exploring fast the different environments, unfortunately their measurements could not provide a high resolution in the exploration area since they fly at high altitude [1, 2]. On the other hand, the blimp robots have recently gained importance, not only in the research studies, but also as a useful means of advertising, climate research as well as surveillance and tracking. Therefore, the blimp robot not only presents and creates a very important opportunity to explore the environment, but also it increases the efficiency of exploration. The blimp robot is known as a special kind of airship that has no rigid structure body. It has better performance than small helicopters or drones due to some advantages such as low speed, low noise, long time hovering, and much less energy consumption [1, 3-5]. It is successful applicable in monitoring and exploring tasks as well as tracking other vehicles. Since the blimp robot depends on low density Helium gas to balance the envelope weight, it does not need any motor actions to maintain at a certain height (altitude).

In addition, the autonomy system is a system that is capable to drive the most essential behaviors without relying on any external control. Moreover, to design a more applicable system for different conditions, a high autonomy degree of software system is needed. In addition, there are several common requests for any robot such as small size and low weight. For instance, let us consider an autonomous blimp that has the mission to track and follow ground robot in unknown area. In such scenario, fundamental tasks to be handled by the blimp are operating on different areas, interacting with other robots developed for same mission and find the exact tracked robot. To construct such system that has the mobility to operate in unknown environments and the ability to perform such tasks, common solutions aim to keep the blimp's size and weight a small and light as possible. Furthermore, this system must have ability to perform its tasks and missions with high level of communication and powerful software platform. The powerful software platform is based on Embedded Linux software which is a computer software written to control the Gumstix Overo Air COM (Computer-On-Module which represents the main core of the blimp system).

On the other hand, some applications have more complexity and require the cooperation between several robots such as the UAV with ground robots or with another UAV. In fact, without condoning that the cooperation is not required in some cases, but it really has the ability to increase the robustness in these applications. However, this kind of cooperation will add more challenges to the research due to the fact that it needs much awareness for some important issues like the communication protocol between the robots and the ground control station as well as among the robots themselves. Another challenge is the control system that has high priority and it should be robust and intelligent to deal with any changes in the environment during the mission. As it is known that the fuzzy logic control is the most widely artificial intelligent control used in several applications. We aim to implement different approaches to control the main behaviors of the blimp robot. However, the main challenge of implementing such control is how to design the fuzzy knowledge base.

Moreover, the increase of the implemented applications in robot research has made the computer vision a decisive and important key factor for them. Thus, several robots have capability of carrying professional cameras which have made the research goes beyond the classical use of visual system as an “eye”. Let’s be more precise, the visual information can be used as an accurate and high performance data that are suitable for control, navigation, detection and visual tracking. It is become obvious now that a visual camera sensor provides reliable data that are useful for designing different intelligent control systems and applications with high capacity, precision and performance.

The most important question that might come to the mind is how to use the visual information collected from the vision sensor in the blimp's control system?. The answer for this question is simply solved by using the visual servoing theory which is known as a vision-based robot control. In our case, it is defined as the technique that could use the visual feedback information which is extracted from the blimp vision sensor (Caspa camera) to control the blimp behaviors. The visual servoing information has refined information about blimp position and attitude as well as a dynamic and timely data such as the presence of targets within the blimp field of view or their relative position with respect to the blimp. However, the target tracking is a challenging problem especially for blimp robot that uses only a single vision sensor since this kind of sensors could not provide metric distances or the range to the target. Also, this challenge becomes more complex and difficult when the targets are dynamic and aerial. Nevertheless, the visual sensors could also be used for increasing the capabilities of the blimp and exploiting the vast amount of information about the targets and environment. The key factors that motivate this thesis are: using visual information, and the integration of blimp intelligent control, which is focused on developing novel and practical applications.

Another important issue for robots system is the localization which is inevitable challenge and it is not solved until now. The localization problem takes its essential aware dramatically since it is the main task for several applications such as navigation, autonomous robotic movement and tracking. One of the most successful solutions for the localization is based on the use of Wireless Sensor Networks (WSNs). They consist of a large number of low-cost and low power sensors, which are deployed in an environment to collect observations and preprocess them to obtain local decisions. They have ability to sense, process and communicate over short distance. Therefore, they can localize every target in the environment, localize any new target coming close to the area, as well as estimate the distances between the targets.

Therefore, in this thesis we seek the design of small size and low weight blimp robot, then design of visual and artificial intelligent systems and test it on actual experiments. The approaches have been tested for the detection and tracking other robots in the environment as well as help the blimp in navigation tasks as it will be described in next chapters. In addition, the approaches which developed are extendable to other applications or other robots. Finally, wireless sensor network will be designed and implemented in order to localize the targets as well as the blimp in the environment.

1.2. Objectives

Therefore, the main objectives of this research that are described in this thesis are the following:

1. To design a high degree of autonomy, small size and low weight blimp robot system. This system has the mobility to operate in unknown environment and the ability to perform the tasks with powerful software platform. The powerful software platform is based on Embedded Linux software which is a computer software written to control the Gumstix Overo Air COM (Computer-On-Module). Even though Gumstix Overo Air COM and Summit Expansion board have small size and low weight, but the combination between them will present a full-sized Linux computer that might be programmed to

carry out and handle several functions in blimp robot. In order to improve the behaviors of the blimp robot, empirical experiments will be developed to study the sensors characteristics and reduce the drawbacks and uncertainties in these sensors.

2. To Design artificial intelligent control, more precisely fuzzy sets models to perform the navigation tasks and to enable the blimp's system to autonomously drive the necessary actions to fulfill its mission and to maintain its policy during the navigation time. In this thesis, we aim to solve the most important problems with fuzzy logic which is how to design the fuzzy knowledge base or the membership functions. Therefore, the combination between the possibility theory and frequency distribution theory leads to design the fuzzy sets model. These models will not only be used for controller purposes, but also some of them to deal with the sensors uncertainties, thus, they correct the information and reduce the drawbacks.
3. To study the fuzzy image processing techniques such as fuzzy edge detection and fuzzy c-means, then, analyze them in order to implement them on the blimp robot. Moreover, propose fuzzy edge detection algorithm and shape-color features techniques for the blimp robot to detect the target and the obstacles in the environment, then to generate the global optimal path with avoiding obstacles for ground targets by enhanced genetic algorithm by modified search A*.
4. To develop and implement computer vision algorithms, which are suitable running on the onboard blimp system and providing accurate information for the control system. The research aims to develop accurate and fast algorithms for object tracking and identification among others as well as to increase the autonomous control capabilities of the blimp robot. This robust visual system will be designed to track targets by considering some important criteria such as performance, repeatability, accuracy and speed. In addition, we will study different approaches and methods for object detection and seek a detection method which allows having good detection with scaled invariant, rotation invariants, robust against noise, and most convenient for our demands. Then, we will correct the prediction position information from the vision system by using fuzzy sets model.

5. To define and design Ground Control Center (GCC) which is the main core of the cooperative perception framework. This is very necessary to record and map the collected data for better decision. Thus, in order to enhance sensing capabilities and to record gathers data on the environment, the real time communications protocols will be designed.
6. To state and estimate the 3D aerial or ground target tracking problem when the blimp robot uses single vision sensor. This problem is really a challenge as the single vision sensor has a common problem which is projecting 3D object into 2D image plane, thus, it could not measure the range between the target and the blimp. This problem comes more complex when the target is an aerial object.
7. To solve the multi-targets tracking problem and to estimate the state of the targets. The estimation process contains many uncertainties such as noise in the measurements and the motion of the target is unknown. The 3D target tracking in WSNs will be considered and we will address the problem of estimating locations of different targets in 3D environment. After estimation of the locations, the distances between the targets could be found especially if the targets are aerial robots.

1.3. Problem Statement

Recently, the cooperation of multi-robots systems has become widely used in several applications such as search and rescue, military, exploration and surveillance. The main reason for using such systems is that they are an appropriate solution in terms of costs, performance, efficiency and reliability. Because the complexity of the robots is increasing, this leads to more aware and researches in the cooperation between several robots [6]. In regarding with that, the design and development of cooperative autonomous robot teams with efficient performance and artificial intelligent behaviors has become one of the most challenging goals in the robot research area. The target detection, localization, and tracking is very important for these types of applications, it can help to localize one robot to another in the environment. The tracking algorithm which uses vision sensors must have capability to cope with several variations in the environment such as lighting and illumination, size of targets, the state and pose of targets, as well as it must be with low computational complexity.

In accordance with that, the design of an artificial intelligent aerial blimp robot could help dramatically to improve the feasibility and suitability of cooperative robotics. However, there are some restrictions to build the blimp system such as the payload, space available for attaching hardware and sensor devices on gondola as well as the unstable wireless communication. In addition, this system does not base on any external control commands and it must display a high degree of autonomy with minimum computational power. The existence of intelligent control as well as navigation algorithm are important to take the best to accomplish the tasks which means add more challenges to the blimp and more uncertainties. Therefore, the blimp must be able to cope with these uncertainties in its location by using artificial system that can deal with noises and drawbacks. Also, to perform the autonomous navigation and tracking tasks for the blimp, it is required to maintain the blimp at constant altitude, ability to avoid obstacles during the missions, and then capable to detect and track the complex target robot. In order to fulfill this demand, it is very essential and necessary to measure velocities and the pose with an accurate sensor for blimp robot. In addition, find the pose of the target related to the blimp. Although the blimp and ground robots have different characteristics, it is possible to develop a unique sensing and perception collaborative system by merging all of these characteristics together. This system could provide aerial images and perception along with the ground robot. Hence, the ground robot could be able to navigate from an initial position to a target point without colliding with other obstacles. To perform these tasks successfully the blimp sensing should have ability to track the target robot and scan the environment to provide ground robots with aerial navigational information. Therefore, in this thesis we approach the challenge of setting up the autonomous blimp system. Thereto, we will present solutions for the following:

1. Design a blimp system including appropriate hardware devices and software platform.
2. Introduce artificial intelligent methods for navigation, tracking and control.
3. Present artificial algorithms to deal with uncertainties, drawbacks and correct the sensors information.

4. Introduce a robust approach for detecting and localization other objects based on vision system.

For these strategies and dilemmas, we present, in this thesis, possibilities distributions and fuzzy sets models techniques not only to reduce the uncertainties in the blimp sensors data, but also to enable the blimp's system to autonomously drive the necessary actions to fulfill its mission and to maintain its policy during the navigation time. These models and strategies based on collected data experimentally after studying the sensors behaviors, analyzed these data in order to propose intelligent models and, in this way, these models could fit and be useful for the current conditions.

On the other hand, computer vision plays a significant role in the demand for many civil applications and it takes much importance and interests because of the optimization of the computational power, the reduction of hardware size and weight as well as the development in image processing techniques. In fact, many robots have good capability to carry cameras. Then, the visual information can be fit and used for our navigation, detection and tracking demands. Also, a vision algorithm must fulfill several requirements such as real time and high precision performance. The main object for the visual tracking system is the ability to identify the target correctly and continuously on the image sequence with respect to some parameters like camera rotation and translations. As it was mentioned earlier, the visual servoing theory (vision-based robot control) is the most common tracking visual technique used in robotics research. The visual servoing information could be used to refine information about blimp position and attitude as well as a dynamic and timely data like the presence of target within the field of view and its relative position with respect to the blimp. For this purpose, the implementation of visual computer algorithms, which is suitable running on the limited performance of the onboard blimp system and able to provide accurate and fast information for the control, has been developed in this thesis. The research framework aims to develop and design fast and accurate algorithms for object tracking and identification among others as well as to increase the autonomous control capabilities of the blimp robot.

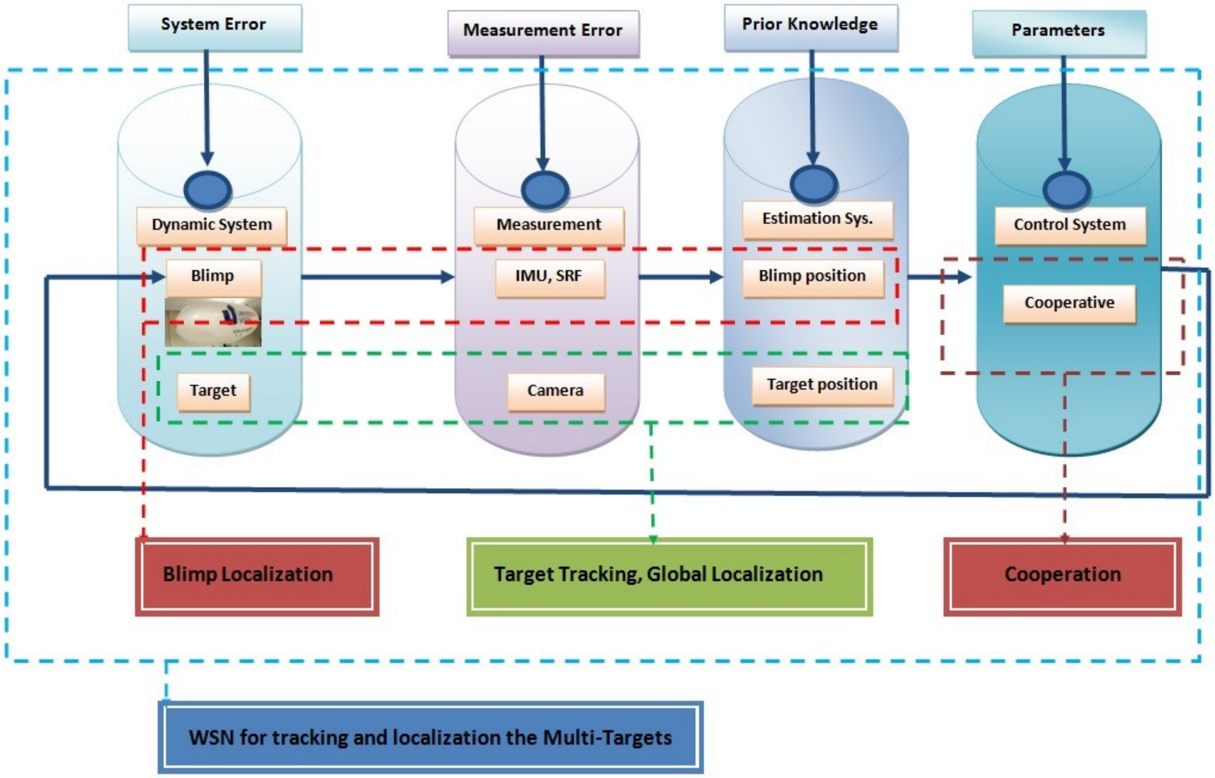


Figure 1. 1 Target tracking scheme using blimp robot.

The decomposition of the problem is shown in Fig. 1.1. The problem can be decomposed into four sub-problems: blimp localization, target tracking, cooperation, and finally the localization and tracking of the whole system based on Wireless Sensor Networks (WSNs). The available measurements are based on Inertial Measurement Unit (IMU) sensor data for blimp robot localization and UltraSonic Range Finder sensors (SRF) which have been used for altitude measurement and to avoid obstacles. In addition, the vision sensor Caspa camera is used for target detection and tracking.

The algorithm focuses on the single-blimp-target problem so that the blimp performs robust target tracking. Actually, there are four main important factors in order to design robust and reliable tracking for such a system. First, if the environment is complicated and complex, the tracking system must be capable to recover from lost tracking. Second, the prior knowledge of target motion which might be random or unpredictable because tracking solutions may be different regarding to the movement of the targets. However, in real- applications, the target motion model is often unavailable. Third, the targets we attempt to track are any complex and moving objects like ground robot or aerial object. Finally, since the blimp robot has no control over the target motions. Thus, based on the current blimp pose and the current target position, the blimp controllers must cope with these issues and provide the blimp an ability to move toward the target for the best position.

Therefore, the target tracking problem has several categories since it has been studied by many researchers and communities for various applications. In this dissertation, we will address several key challenges for single target-based moving object tracking. First, how to separate the target (which could be aerial or ground target) from other motion objects, and then how to detect and track moving targets. For this purpose, the computer vision methods algorithms for tracking will be described. However, the most popular criterion in this area is the accuracy for the tracking algorithm that is able to estimate the state of a target with less noisy measurements. The uncertainty of target position estimation could be reduced by using fuzzy sets model that are able to correct the prediction position information in the computer vision system. For path planning problem, cooperation among the blimp and the target is essential to improve tracking performance and accuracy. Hence, the fuzzy edge detection and shape-color features technique has been used to separate the target in the environment as well as to detect the obstacles in its path. The fuzzy motion planning and genetic algorithm have been used in order to find the trajectory and path for the robot in order to navigate from start point to end point in unknown environment.

In fact, the vision system has the ability to detect multi targets in the environment which can be globally localized related to the blimp position as it is shown in Fig. 1.2. However, due to the possibility to track and follow multi targets on same time it is a challenge to seek another technique which can detect and localize multi-targets. Moreover, the tracking strategy which is based on WSNs will provide the information about the whole system and it is sufficient for such purposes. Unlike other published work and research, in this work we will address the problem of estimating locations of different targets in 3D environment. In order to get local decisions each sensor makes several observations. Then, these decisions will be modulated using On-Off Keying (OOK). The fusion center has ability to detect, collect and process all decisions to estimate the targets locations. This process in the fusion center is using the maximum likelihood estimator. It is not only for localization purposes, but also it provides more accurate data about the state of the multi robots besides it can estimate the distances between the targets and the blimp even though they are aerial. Fig. 1.3 shows one example of two targets placed in 3D space to be localized using binary sensor network. The real locations of the 2 targets are shown as red circles while the estimated locations are shown as black circles. A top view of the WSN with the two targets and the estimated locations are shown in Fig. 1.4.

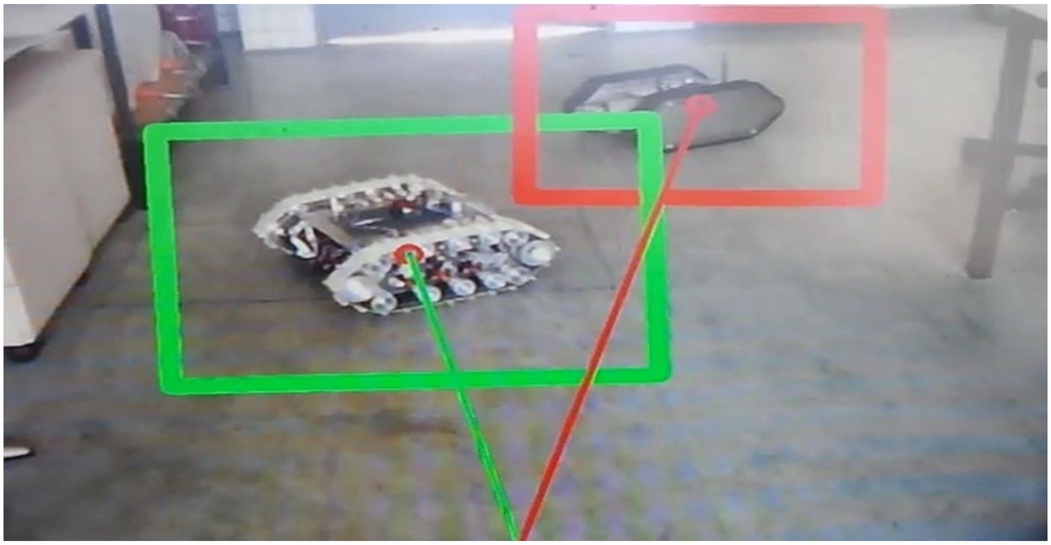


Figure 1. 2 Multi-Target detection and localization by vision system.

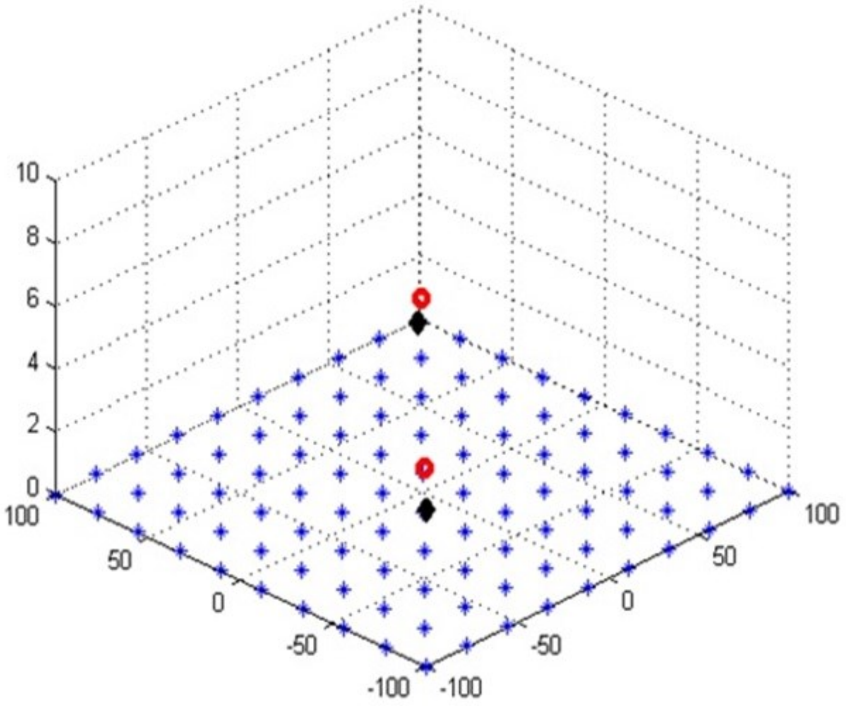


Figure 1. 3 3D view of two targets in WSNs environment.

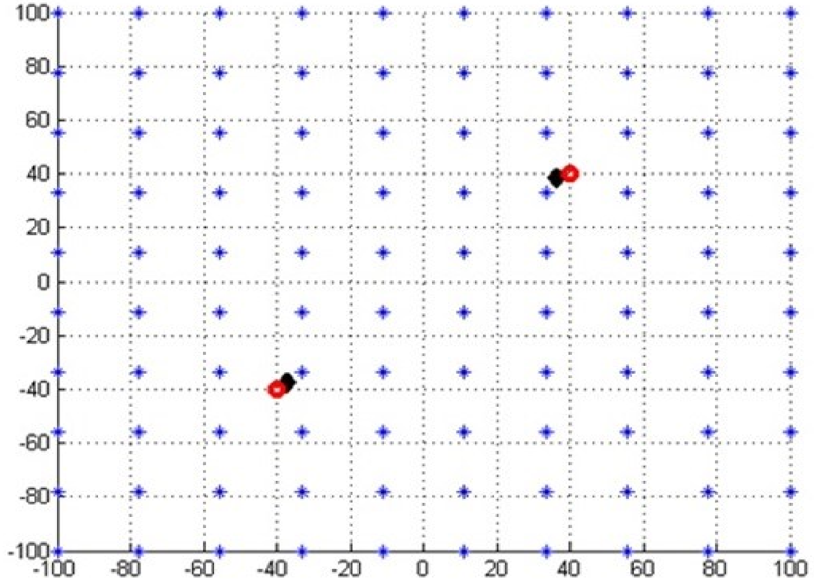


Figure 1. 4 2D view of two targets in WSNs environment.

1.4. Outline and main contributions

With this thesis, we contribute some novel approaches to the field of aerial robotics as well as tracking and cooperative system by developing and researching solutions for realizing an autonomous blimp system. We design a flexible but still powerful blimp platform and demonstrate its capabilities. Furthermore, we investigate several intelligent control approaches as well as computer vision for embedded systems. The thesis consists of six chapters complemented with four appendixes. The summary of the chapters contents will be presented here:

Chapter One: describes the motivation of the research work and the objectives. It describes also the problem statement as well as the contributions.

Chapter Two: It describes the literature reviews to identify what has been done before regarding to our issues. This includes the reason that blimp has taken high priority in our research work. Then, it discusses the vision technology regarding to the target detection and tracking. In addition, talk about WSN technology and its advantages for localization and tracking. Finally, brief information about the classical control system and artificial intelligence control regards to advantages and disadvantages.

Chapter Three: This chapter deals with the cooperative perception issues in multi-robots system as well as the target tracking problems. A discussion on its main problem and some important definitions are presented. Then, the approaches which have been used to track systems are described. Also, it discusses the fuzzy sets models and control approaches which have been implemented in this thesis. It provides also a method to solve the most common problem in the fuzzy logic and fuzzy control that is how to design the fuzzy knowledge base without relies on a human expert or simulation studies. Moreover, the combination between the possibility theory with the fuzzy sets leads to model the complex systems experimentally and empirically without regard to the present a simulation model study or an expert. Finally, the wireless sensor network which can be used for localization of the targets in the environment.

Chapter Four: This chapter summarizes the most image processing and computer vision approaches and methods which have been used during this thesis. First, different approaches are introduced in order to find relevant visual features and matching useful for object detection, identifying and tracking regions of interest in the image sequences. Then, this tracking method is used to estimate a projection model in the camera image space. Finally, the robust embedded visual algorithms implemented on on-board blimp to give accurate and fast information. This information will feed the blimp's control system based on visual servoing theory. It was called here robust due to the ability to detect targets with high robust capability against the noise and it was named embedded because it will run on an Embedded Gumstix board which runs Embedded full Linux.

Chapter Five: It discusses the implementation of the computer vision systems as well as the servoing fuzzy control models on the real tests by using the blimp robot.

Chapter Six: End of the thesis and a discussion about the research works and concludes with the future work direction.

However, some parts of this thesis were published at international conferences and international journals. The areas of the conferences vary in topics and they are related in the fields of: Telematics, Computer vision and Image Processing, Robotics, Mechatronics as well as Neural Network and Intelligent Artificial. In more details, the main contributions of this thesis are the following:

[1] Rami Al-Jarrah, Aamir Shahzad and Hubert Roth. **Path Planning and Motion Coordination for Multi-Robots System using Probabilistic Neuro-Fuzzy**. *2nd IFAC Conference CESCIT 2015. Slovenia 2015*.

[2] Rami Al-Jarrah and Hubert Roth. **WSNs and On-board Visual Fuzzy Servoing on Blimp Robot for Tracking Purposes**. *4th international Conference on Information Computer Application, Amsterdam, Netherlands, 12-13 February, 2015*.

[3] Rami Al-Jarrah and Hubert Roth. **Multiple 3D Target Tracking in Binary Wireless Sensor Network**. *International Conference on Robotics and Mechatronics (ICROM2014),* July 6-7, 2014, Nottingham, UK, 2014.

[4] Rami Al-Jarrah, Bashra Kadhim, Aamir Shahzad and Hubert Roth. **Towards a Heterogeneous Navigation Team of Aerial-Ground Robots based on Fuzzy Image Processing**. *International Conference on Robotics and Mechatronics(ICROM),* July 6-7, 2014, Nottingham, UK, 2014.

[5] Rami Al-Jarrah and Hubert Roth. **Visual Fuzzy Control for Blimp Robot to Follow 3D Aerial Object**. *The 8th International Conference on Neural Network and Artificial Intelligence (ICNNAI 2014),* 3-6 June, 2014, Brest State Technical University, Belarus. **Published in: Communications in Computer and Information Science, Springer International Publishing, Vol. 440, pp. 98-111, 2014**.

[6] Rami Al-Jarrah and Hubert Roth. **Intelligent Tracking and Trajectory Navigation Approach for Blimp Robot**. *International Journal of Enhanced Research in Science Technology and Engineering*. Vol. 3 (1), pp. 198-204, 2014.

[7] Rami Al-Jarrah and Hubert Roth. **Visual Fuzzy Logic Path Planning Controller for Mobile Robots**. *International Journal of Enhanced Research in Science Technology and Engineering*. Vol. 3 (1), pp. 215-221, 2014.

[8] Rami Al-Jarrah and Hubert Roth. **Ground Robot Navigation Relies on Blimp Vision Information**. *3rd International Conference on Information Computer Application (ICIC2014). February 20-21, 2014, Barcelona, Spain, 2014.*

[9] Rami Al-Jarrah, Radouane Ait Jellal and Hubert Roth. **Blimp Based on Embedded Computer Vision and Fuzzy Control for Following Ground Vehicles**. *3rd IFAC Symposium on Telematics Applications (TA 2013). November 11-13, 2013. Seoul, Korea, 2013.*

[10] Rami Al-Jarrah and Hubert Roth. **Design Blimp Robot based on Embedded System & Software Architecture with High Level Communication and Fuzzy Logic**. *9th International Symposium on Mechatronics & its Applications (ISMA13), Proceeding in IEEE , Amman, Jordan, April 9-11, 2013.*

[11] Rami Al-Jarrah and Hubert Roth. **Developed Blimp Robot Based On Ultrasonic Sensors Using Possibilities Distributions and Fuzzy Logic**. *5th International Conference on Computer and Automation Engineering ICCAE, Jan 2013, Bruxelles- Belgium*, 2013.

[12] Bashra Kadhim Oleiwi, Rami Al-Jarrah, Hubert Roth. **Multi Objective Optimization of**

Trajectory Planning of Non-holonomic Mobile Robot in Dynamic Environment Using Enhanced GA by Fuzzy Motion Control and A*. *Journal of Communications in Computer and Information Science, Springer. vol. 440, pp 34-49, 2014*.

[13] Bashra Kadhim Oleiwi, Rami Al-Jarrah, Hubert Roth. **Intelligent Hybrid Approach for Multi Robots-Multi Objectives Motion Planning Optimization**. *International Journal of Enhanced Research in Science Technology and Engineering. vol. 3, no. 3, pp. 539-550. 2014*.

[14] Aamir Shahzad, Rami Al-Jarrah and Hubert Roth. **Teleoperation of AutoMerlin by Inculcating FIN Algorithm**. *3rd International Conference on Control, Robotics and Cybernetics ICCRC 2015, well held in Berlin, Germany during August 13-14, 2015.*

[15] Aamir Shahzad, Rami Al-Jarrah and Hubert Roth. **Telecontrol of AutoMerlin Robot employing Fuzzy Logic**. *3rd International Conference on Control, Robotics and Cybernetics ICCRC 2015, well held in Berlin, Germany during August 13-14, 2015.*

[16] Bashra Kadhim, Rami Al-Jarrah, Hubert Roth. **Integrated Motion Planning and Control for Multi Objectives Optimization and Multi Robots Navigation**. *2nd IFAC Conference CESCIT 2015. Slovenia 2015*.

Chapter 2 Literature Reviews

The best start point when we are seeking for any solution is to identify what has been done before regarding to that issue, the state of the art and the worldwide trends around the problem of interest.

2.1. Aerial Robots

Recently, the cooperative of aerial-ground robot systems has become widely used in several applications such as search and rescue, exploration and surveillance as well as target detection and tracking for either military or civilization. The main reason for using such system is that they are an appropriate solution in terms of costs, efficiency, performance, and reliability. In this way, the number of robots is actually increasing which makes the system more complex, therefore many researchers start designing the cooperation among multi-robots [6, 7]. In accordance with that, designing and developing an artificial intelligent team consists of cooperative autonomous aerial-ground robots with efficient performance has been one of the most challenging goals in robotic research. The cooperation of such system has involved state of problems such as efficient navigation, path planning, exploration, localization, target tracking, and suitable communications. As a result, the control of single robot such as the blimp could help dramatically to improve the feasibility and suitability of cooperative robotics by developing inter-robot communication that ensures the autonomy and the individual requirements of the involved robots.

Since the development in new technologies has been dramatically increased as well as several applications come to our real life, the UAVs have been taken a higher priority for many universities and governments. This interest is due to the wide variety of applications in civil and military fields such as search and rescue, surveillance, target detection and tracking. One might probably wonder what is the purpose of tracking another robot in the environment?. This is simply regarding to the fact that the tracking can help and improve the ability of the aerial robot to cooperate with other robots in the system and share information about the target. Perhaps the small airplane such as helicopters (drones) can explore the different terrains and areas. Unfortunately, their measures do not give a good resolution for the exploration area since they fly fast at high height. In addition, it is not easy to change the direction immediately in case they want to track and follow an object which moves at the ground [1]. Drones would have to be equipped with highly accurate sensors and provided by vision systems that are able to reliably identify intended targets based on very limited and misleading information. However, none of the currently operational drone can reliably distinguish between legitimate military targets and civilian persons or objects or even take precautions to avoid erroneous targeting [8].

Therefore, it becomes very essential to look for another solution which has more advantages over drones. The airship robot is one of the most important solutions in this field. The unmanned airship system not only creates an opportunity to explore the environment, but also it increases the efficiency of such exploration. In addition, it gains more interest recently because of its advantages such as long time hovering, efficient cost and low noise [1-5]. These advantages make the airship reaches anywhere to remote or to access difficult regions. Nevertheless all of these advantages, but in order to build an artificial blimp robot system, an eye (camera) is also important for this kind of robot. It can provide a high artificial visual system that simulates the principle and characteristics of the human eye system. Openly, the blimp robot equipped with a vision payload can perform a wide range of these applications. However, the single vision sensor has a common problem which is lack of ability to provide the range information of the target. Some researchers solved this problem by adding another sensor which can measure the distances. Unfortunately, this method will increase the blimp system's cost by adding new hardware. Thus, if the robot can measure the range by only single vision sensor that could help to reduce the overall cost of the robot.

2.2. Computer Vision on Robot Applications

Moreover, the robot vision tracking technology is utilized widely because of its advantages such as reliability and low cost. It is applied widely in the fields of security, traffic monitoring, object recognition and tracking. Several researches have been presented and devoted to vision flight control, navigation, tracking and object identification [9-12]. It is clearly and without doubt that the color object recognition, which can adapt any changes in illumination, was most under study research topic and most of its aspects were almost solved. Korodi et al. [13] have proposed a visual information algorithm for cooperative robotics. Every robot ,which is in a dynamic environment, is realizing a cooperative action and has to be able to accomplish its task by detecting an object as well as following it in the environment. Visual information also has been proposed on aerial robotics for flying information [14]. They describe a simple and straightforward algorithm for multiple UAVs in a formation with no communication between the vehicles. It was assumed that relative kinematics parameters are available to each UAV from an on-board passive sensor followed by estimation processes and a controller which may use visual information. Several approaches also have been proposed for UAV following circular paths in order to track a ground moving object [15]. In this project, the problem of following a moving target took into account that the target may have higher maneuverability as well as they try to keep the autonomous unmanned vehicle as close as possible to the target all the time. Fukao et al. [16] presented the motion segmentation to improve the detection and tracking of point features on a target object. Then, the inverse optimal tracking control was applied on the blimp robot in order to track point features precisely. However, this approach has some disadvantages like sensitivity to light and the amount of image data.

Let us be more specific and transparent in this area of research. In fact, when we review the most researches in this area, in particular, we must deep study the vision tracking approaches. Numerous treatises and projects have been done on this rich topic in where it gains its significance and importance dramatically. So, one might find many studies which have been proposed for vision tracking methods such as the target tracking based on relatively simple color [17], template matching features [18], background subtraction and object classification [19], and feature based approaches tracking [20, 21] . However, many algorithms could not meet the requirements of real time and robustness due to large amount of video data. Therefore, it becomes important to improve the accuracy and real time capability of tracking algorithms. In order to improve the detection of targets in real time and robustness, Lowe [22] has proposed the Scale Invariant Features Transform (SIFT), and then the Speeded Up Robust Features (SURF) has been proposed [23]. Because the processing time of SURF algorithm is faster than of SIFT, interest point detection (keypoint detection) of SURF algorithm is used for real-time processing especially limited on-board vision system. As a result, the design and implementation of a real time embedded vision system for intelligent blimp robot, which includes an on-board embedded hardware system and control algorithm, becomes very important. This significance not only comes to make the blimp more autonomous , but also to provide it with the most new vision techniques to track another robots in the environment. It is very important to mention here that SURF is much new, intelligent, fast and accurate algorithm among other approaches in where the complex target robot could intelligently distinguish among other objects. It's an exciting to have such algorithm that can easily recognize and localize any specific target even though it is in size invariant or rotation invariant.

In fact, the most common way in order to classify the vision system or the computer vision in robots depends on the complexity degree of the robot applications as it is illustrating in Fig. 2.1. In the first level, image processing technique are employed to process the captured images, Then, the visual data can be fused with other sensors and actuators to make specific applications. Finally, the visual information is used to control the blimp actuators. The most famous process in visual system is called a visual tracking that is analyzing of sequential images to identify a reference pattern and to follow the interest point over time. There are many tracking methods and approaches in which the algorithms could based on color, shape or features [24]. In visual odometry, the images will be analyzed to extrapolate the robot movement, estimate the position, and also estimate the robot orientation [25]. If the process uses the visual data to determine object position as well as the safe path, it will be called visual navigation [26]. In regarding to the visual navigation, many navigation approaches and algorithms have been proposed such as in structured environments using white line recognition [27], corridor navigation [28], more complicated techniques that combine valid planer with the visual localization [29], visual and navigation techniques to perform obstacle avoidance [30].

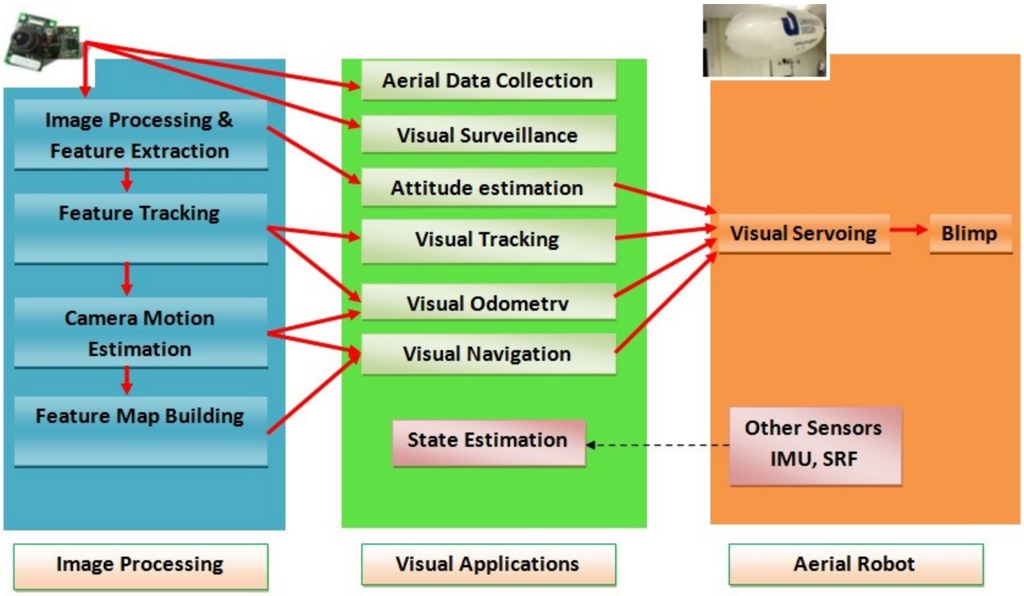


Figure 2. 1 Vision scheme.

2.3. Tracking and Navigation

On the next stage, the cooperation between the aerial and ground robots has been studied and developed because such system might provide aerial imagery and perception along with ground robot inspection capabilities to help in dangerous tasks as well as to explore large unstructured environments. Hence, the ground robots may be able to navigate from an initial position to a target point without colliding with other vehicles and obstacles. This navigation process might be described as the problem of finding a suitable, better and free collision path motion of the ground robot, while obstacle mapping includes using the sensing capabilities to gain representative of unknown obstacles that is useful for navigation [31].

Despite of having different characteristics, it is possible to develop a unique sensing and perception collaborative system between blimp and ground robots. In other words, as a typical lighter-than-air vehicle, the autonomous blimp robot is a unique platform for several applications such as telecommunication, broadcasting relays, and scientific exploration [32]. On the other hand, the visual servoing and navigation, especially for humans and vehicles, are currently the most active research topics in computer vision. In [33], they study a multi-robots system depending on a vision-guide quad-rotor and described some methods to take off, land and track over the ground robot. However, the quadrotor does not be able to provide information to the ground robot about the surrounding environment. Also, [34] presents a motion-planning and control system based on visual servoing without cameras on board. In [35], they integrated and fuse vision data from the aerial and ground robots for best target tracking and for allowing leveraging of multi-domain sensing and increase opportunities for improving line sight communication. These studies and others have been tested and implemented for several purposes such as environment monitoring, fire detection, fighting and remote sensing system [36-38].

Also, pose-base methods have been employed for some applications like autonomous landing on moving objects. These methods are important in order to estimate the 3D object position [39]. The image based methods also have been used widely for positioning [40]. In [41] they deal with this problem by proposing a measurement model of the vision sensor based on the specific image processing technique that is related to the size of the target. However, a nonlinear adaptive observer is implemented for the problem and the performance of the proposed method has been verified through numerical simulations. For the bearing measurement sensor, the maximization of determinant and fisher information matrix has been studied to generate the optimal trajectory [42, 43]. However, these approaches are hard to implement because of the high computational load. In addition, the tracking problem has been studied by using a robust adaptive observer and the intelligent excitation concept [44-46]. Note that the observer is only applied to the systems whose relative degree is one, and therefore it could not be implemented to higher relative degree systems. The 3D object following method has been developed based on visual information to generate a dynamic look and move control architecture for UAV [47]. Also, aerial object following using visual fuzzy servoing has been presented [48]. However, they approach the problem of the tracking by exploiting the color characteristic of the target which means to define a basic color to the target and assuming a simple colored mark to track it. This process is not always perfect and might face problems due to the changes in color over time. Also, the designs of the fuzzy controllers were made based on the excellent results after trials and errors method.

We are interested in tracking targets using aerial blimp robot as the tracking device because the blimp can cover and explore a wide and big area over time, which means the number of sensors required for tracking can be kept small. As it was mentioned earlier the target tracking problem has been studied in several research areas with many applications. In the computer vision community, the tracking algorithms have been developed to track visual targets by using camera sensors [49-55]. In addition, in the wireless sensor network, the problem has utilized deployed sensors to monitor moving objects in the environment [56-60]. The most work in the single tracker- single target category focuses mainly on signal processing techniques for target detection and probabilistic filter to filter out noisy measurements. The Kalman filter and the particle filter have been applied to the single target tracking problem successfully [61, 62]. Even though the particle filter is a simple, but effective algorithm to estimate the posterior probability distribution recursively, which is appropriate for real-time applications. In addition, its ability to perform multi-modal tracking is attractive for multiple objects detection and tracking [63]. In order to detect and track a visual target (ground robot or object) using a single camera, the computer vision was introduced and presented the visual servoing technique of many applications and robots systems such as autonomous target capturing and control non-holonomic mobile robots [64-68]. The mobility of the tracker is extremely important for robotics tracking and cooperation; thus, based on the possibility degree of the tracker motions, the tracking problem can be classified as the following: stationary, pan/tilt/zoom, planar, and unrestricted [69]. The computer vision community has introduced several algorithms and methods to stabilize the camera by features [70-72] or optical flow [73-77]. These approaches focus on how to estimate the transformation between images even though the target is moving. Once the moving object has been identified, then, it needs to be tracked in the scene. Finally, in accordance to the vision system, the detection and classification of the targets will be more flexible by using the pan–tilt–zoom cameras [78].

2.4. Control System

One of the most common and important problems is the controller algorithms for the blimp robot. In fact, the scientific community is still debating between the classical control system (e.g. PD or PID) and artificial intelligence control (such as fuzzy logic or genetic algorithm). The preference, advantages or disadvantages between them are still ongoing. Every supporter of a particular opinion from both sides try to highlight the performance and advantages compared with the other. The fuzzy control system does not have much better characteristics in time domain than PID classical system, but still it can deal with nonlinear system. In [79] the comparative results between the fuzzy controller and the PID controller show that both might be slightly better than the other for different environments, scenarios and conditions. Whilst a better PID requires online adjustment for its parameters, fuzzy controller does not affect the system behavior too much and needless for parameters modification [80, 81]. Also, the fuzzy controller might perform better with the uncertainties [80, 81].

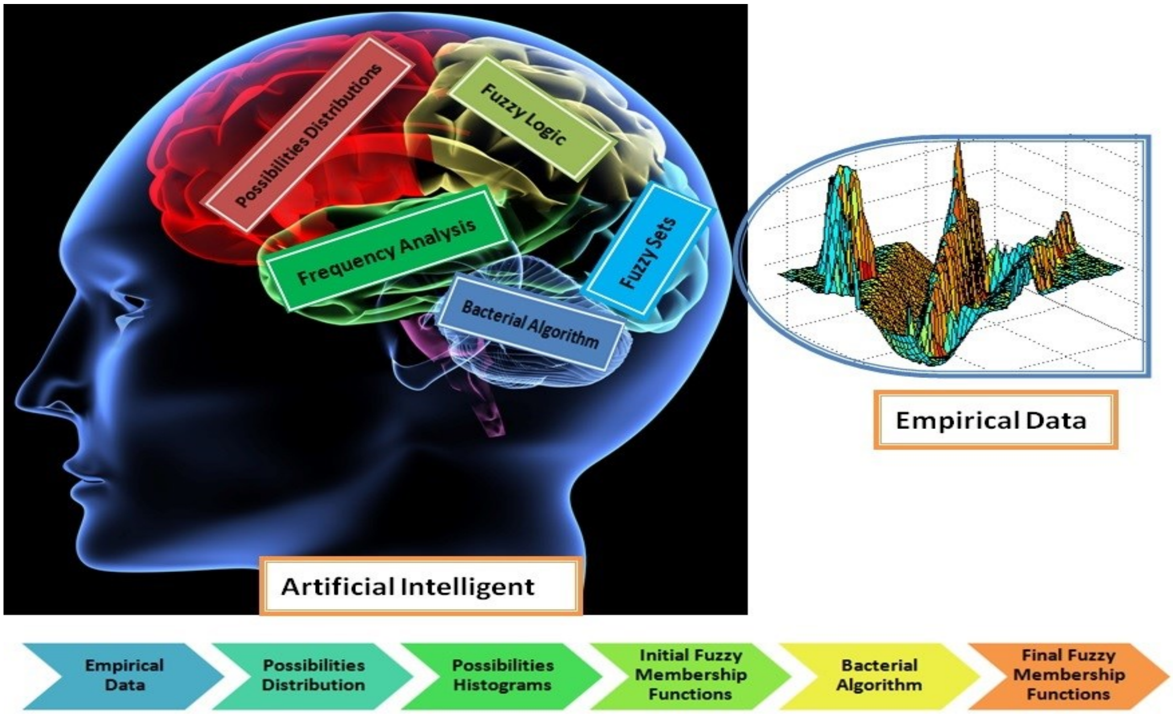


Figure 2. 2 **The combination between fuzzy and possibilities theories.**

There are two important issues in fuzzy control. First, since there are complex operations like fuzzification and defuzzification, the computing time is much longer in fuzzy than in the classical control. As a results, it is better to optimize the fuzzy rule base in order to reduce the computing time. Second, the most important problems with fuzzy are how to design the fuzzy knowledge base or the membership functions. Moreover, the possibility theory could deal with the uncertainties in any system and can handle incomplete information. The most important characteristic of such theory is the ability to link itself strongly to the fuzzy intervals [82].

Therefore, the combination between the possibility theory, bacterial algorithm and fuzzy logic sets will lead to model the complex systems empirically without regard to the presence of the expert or even simulation studies as it is shown in Fig. 2.2. This combination is not only essential in order to design the fuzzy controllers, but also it can be used to design fuzzy sets models which can be used to reduce the noises and drawbacks in the system.

2.5. Localize Multi Targets

In the past few decades, WSNs have been studied extensively in the literature. The WSNs have wide applications start from military surveillance to security and monitoring environment, cognitive radio networks due to their high mobility, flexibility, and cost effectiveness. They contain a large number of low-cost and low-power sensors, which are deployed in an environment to collect observations and preprocess them to obtain local decisions [83-87]. Target localization has been considered as one of the most important aspects in WSNs which enables and gives ability to track the target [88]. Because it is not possible to estimate and localize the target in 3D plane (especially if we are looking for the altitude values) with a single sensor, there is a need to fuse the multi sensors data. Many algorithms have been proposed in the literature to localize the target, i.e., time domain of arrival (TDOA), direction of arrival (DOA), received signal strength indicator (RSSI), and energy based methods [89-95]. In [96], maximum likelihood estimator has been used to estimate and localize the target where each sensor sends a row of quantized decisions to the fusion center. Depending on all rows sent by different sensors, the fusion center uses maximum likelihood estimator to estimate the location of the target. Moreover, the efficiency of maximum likelihood estimator is compared with the weighted average algorithm and the Cramer-Rao lower bound. Further work, channel aware target localization is considered in [97] where the estimation process carried out using the quantized sensor data in addition to the fading channel statistics. In this thesis, we consider a 3D multi target localization problem where targets are flying within the range of the sensor network. On the other hand, sensors are assumed to be ground sensors where they are deployed above the ground surface plane. Moreover, sensors modulate their decisions using

OOK technique which has been proposed recently for WSNs because it is a power efficient modulation technique and enables sensors to use censoring scheme (send/no send).

2.6. Conclusion

In this chapter, the related works from different research areas are discussed. The combination between all of these areas lead us to identify what has been done before regarding to our issue and to deep understand such problem and find the solutions to reach the objects of such thesis.

Chapter 3 Cooperative Perception, Tracking and Control System

This chapter deals with the cooperative perception issues in multi-robots systems and the target tracking problems as well as the design of the fuzzy sets models. A discussion on the main problems and some important definitions are presented. Moreover, the related work on robot perception is summarized. Then, the approaches which have been used for the tracking system. Then, the fuzzy sets and possibilities theories are introduced. Finally, the wireless sensors network which can be used in indoor and/or outdoor environments to track multi objects (aerial or ground objects) is presented.

3.1. Introduction

Actually, the robot perception system is employed to estimate the static and dynamic characteristics of the surrounded environment such as obstacle position, the target position, and robot position. This estimation is obtained from the data gathered by the different sensors mounted on the robot such as ultrasonic sensors and vision sensors. In the case of the multi robots, if the information that each robot obtains about the environment can be shared among them, each robot obtains a better picture of the surrounded environment and the perception will be increased. Moreover, from the viewpoint of the perception, the robots can collect data, then analyze them in order to develop accurate and safe actions as well as to establish a cooperation during the missions. Thus, this cooperative behavior between the robots might be defined as the collaboration action between multi robots which share individual information to estimate the environment state which leads to estimate the state of robots themselves. Moreover, the robots use their sensors to update and estimate their local knowledge about position and velocity of the surrounding obstacles or targets. If a robot has the ability to communicate with another robot in the environment, then the uncertainty knowledge about the environment can be reduced. For example, the aerial robot explores the environment to detect the ground target robot and other obstacles, and then, provides the ground robot with navigational information. Another good example in this area is considering that we can provide the ground robot, which has no prior information about its environment, with the optimal trajectory and path by using intelligent system such as fuzzy-genetic algorithm. The aerial map images will transfer to the ground control station. Then, the global optimal path avoiding obstacles is generated by enhanced genetic algorithm as it is shown in Fig. 3.1 and Fig. 3.2. Details about this novel work can be found in [98].

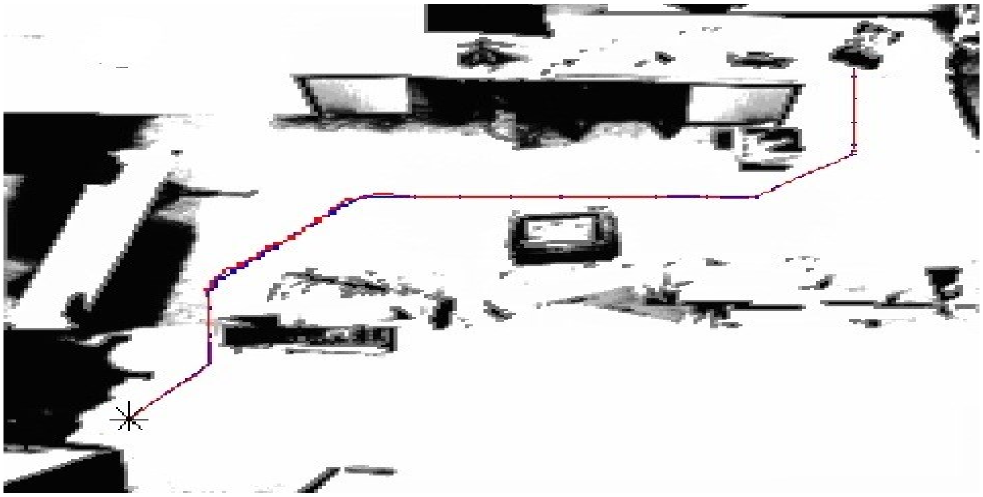


Figure 3. 1 Fuzzy edge detection and optimal Path trajectory.

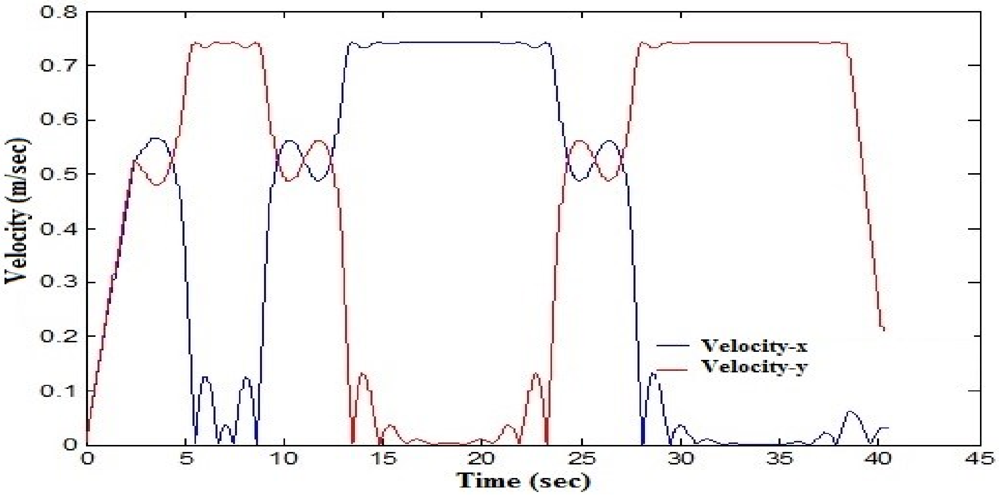


Figure 3. 2 The x and y velocity of the mobile robot.

Another applicable example is the recognition and identification in the rescue situations. The process of identifying human victims among other things with impressive results is a challenge task because many humans in the images are not the desired targets. By using teleoperated robot based on artificial control and computer vision, this challenge can be done and allows the single operator to control the agent robot via internet by considering significant amounts of data that are transmitted between the robots and the Ground Control Center (GCC). With the availability of these data the operator can scan the environment looking for the victims. The artificial control gives the robot ability to navigate avoiding any obstacle, then, the computer vision capable of detecting the human, and then send information to GCC or blimp robot.

The cooperation and sharing information among robots has a variety of challenges that must be addressed. These challenges are not only from a robotics perspective, but also from some other disciplines such as artificial intelligence and sensor networking communities. As it is shown in Fig. 3.3, the main issues are the intersection between control, perception and communication in which it is responsible for attaining the adapting, networking and decision making that will provide the capabilities for efficient operations [78]. More precisely, the aerial-ground robots system can adapt any change in the environment. For example, the teleoperated ground robot (AutoMerlin) has artificial control in order to avoid obstacles during the mission even though the operator could give a wrong command. This kind of system represents a tool to gather more images and data from the environment to display or record them on GCC for further investigations and for better decision making and then an efficient search operation as it is shown in Fig. 3.4. The AutoMerlin robot navigates in the environment in order to explore it based on operator commands (Joystick commands) that have been sent via internet. However, the artificial control could detect any moving object in the environment and in this case it will not response to the operator command.

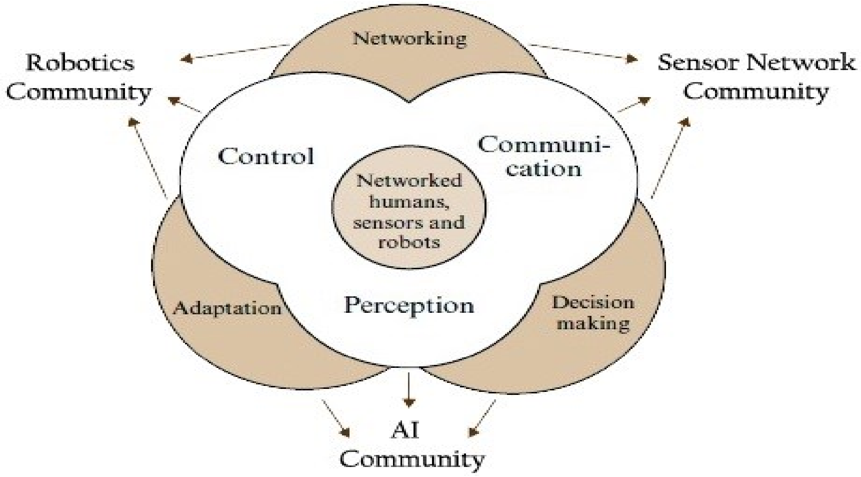


Figure 3. 3 Major challenges for robots [78].

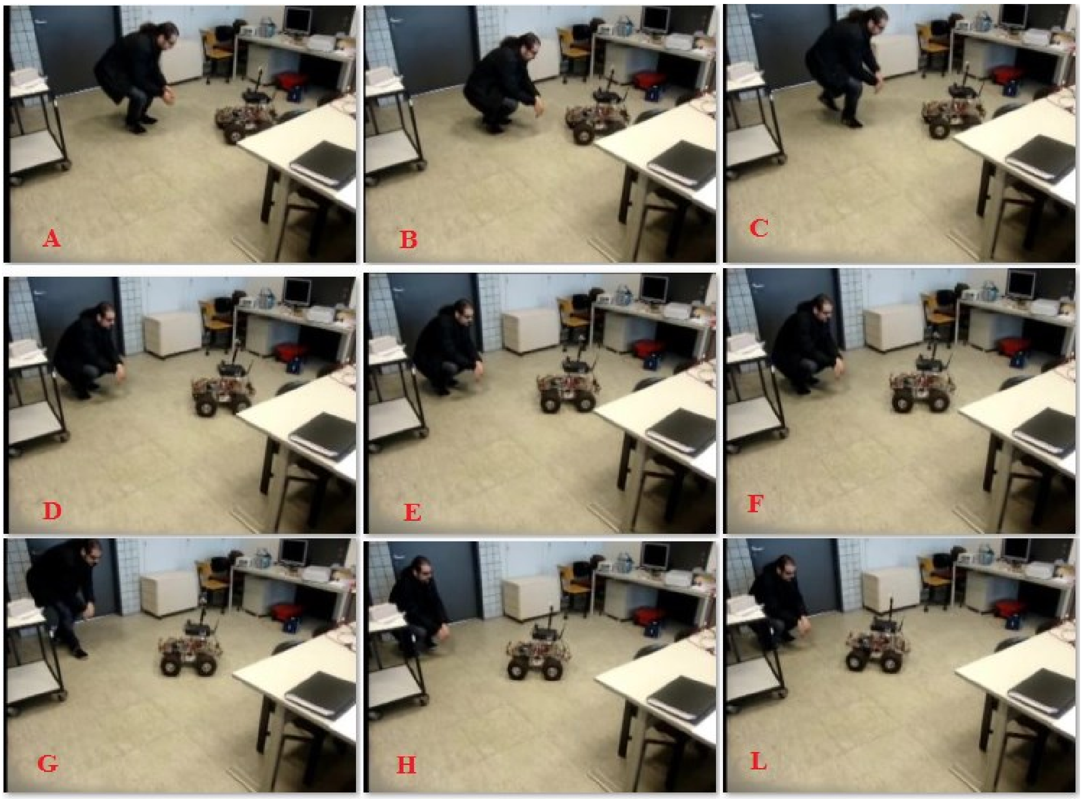


Figure 3. 4 AutoMerlin Robot detecting moving object.

Therefore, in order to understand the situation here we need to state the major addressed issues:

1- Ground Control Center (GCC): it is very important in such task to record and map the data collected by the robots for better decision. This interface could be used to receive all flight data acquired from blimp onboard system, and then the aerodynamic information will display and plot in real time as well as store to provide the information for further flight investigation and analysis.

2- Control: The control system is really a challenging task because of the mechanical complexities of the aerial blimp robot and the characteristics of the environments. Therefore, the fuzzy logic and fuzzy controllers will be designed and implemented because of several reasons such as the simplicity and the powerfulness of the fuzzy methodology to solve and control the main behaviors of the system. Also, fuzzy is capable of providing remarkably simple ways to draw definite conclusions from the vague information.

3- Communications: in order to increase and enhance the sensing capabilities as well as to record gathered data of the environment, blimp robot should have ability to communicate in real-time with GCC or with a robot team in the environment. Whilst at the lower level the communications enable to state feedback of the robot, at higher level the robots have the ability to share information for many purposes such as coordination control and planning [98]. However, there is typically a destroyed communication infrastructure and ad hoc networks are likely to become saturated [99].

4- Tracking: The target tracking problem is defined as the method or technique which is used to estimate the location of a target based on the sensors measurements as well as to estimate the state of this target. As it was mentioned in Chapter 2, the target tracking problem has several categories and it has been studied by diverse research communities in different applications. The target tracking algorithms has widely applications range from surveillance to military purposes. In addition, the motion tracking and estimation of moving targets in the environment is also a fundamental capability for safe navigation and path planning applications. The tracking algorithm in this research focuses on the single-blimp-target problem by using vision system. Thus, the blimp robot performs robust target tracking. However, because the sensors measurements might be inaccurate due to some factors like uncertainties, noises and the unknown motion of the target, it is better first to reduce these noises and uncertainties. Then, to realize the autonomous aerial-ground target following, an efficient and accurate vision-based object detection and localization algorithms is proposed in this thesis by using visual system and servoing system.

3.2. Blimp Robot System

This section will give an overview and brief description of the blimp system that we will use in the following chapters to demonstrate and evaluate the proposed strategies. A brief technical review is provided to describe the small and low-weight hardware components for the blimp robot. In more details, our blimp system consists of a processing unit or the core of the system, a networking device, several sensors and actuators, as well as an artificial intelligent software framework. Then, the key reasons to select the Gumstix-Overo-Air (the core system) that resembles an embedded system will be discussed [100]. In general, the airships without an internal supporting framework are called blimps. The envelope of such blimp basically maintains its shape by using the higher pressure of the lifting gas inside the envelope, more precisely the Helium gas. Fig. 3.5 shows our blimp system consisting of an envelope, a gondola, and other electronic devices.

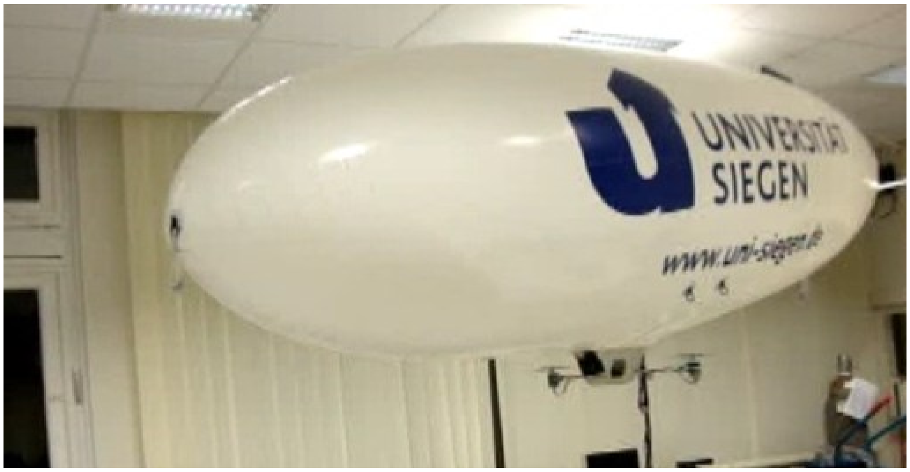


Figure 3. 5 **The blimp robot.**

The interface among the on-board Linux operating system, microcontroller on-board and the devices is providing a simple, easy, and powerful system to handle the blimp main behaviors [101]. By empirically tests, we evaluate the performance of the components, and design artificial fuzzy sets models in order to deal with noises and uncertainties in the system. Therefore, in order to demonstrate the capability of the complete system we are integrating the blimp system into navigation issues to enable the system flies in indoor environments at a certain altitude and whilst avoiding obstacles as well as tracking targets.

3.2.1. Blimp System Component

The blimp robot is a kind of under-actuated and non-holonomic system. Thus, the control degrees of freedom are less than the total degrees of freedom. Hence, blimps are not able to control their lateral movements and it is not easy to control them in 3D spaces. Also, blimp's characteristics have some restrictions considering its electronic hardware. For any blimp system, if the envelop volume gets higher, the ascending force will be affected and thus increased and as a result the possible payload increases. Based on these features, we have several objects to reach during the design such as minimize the weight of the needed hardware equipments and develop appropriate, efficient and robust artificial intelligent control algorithms for autonomous flight, navigation, localization and tracking tasks.

The main components of the blimp system are shown in Fig. 3.6 which shows gondola onboard unit (GOU) with all the electronic components. One self-imposed goal of the project was the design and development of a system with a maximum degree of flexibility and high degree of ability to be continuously improved in the future. Thus, the hardware system is separated into several main parts.

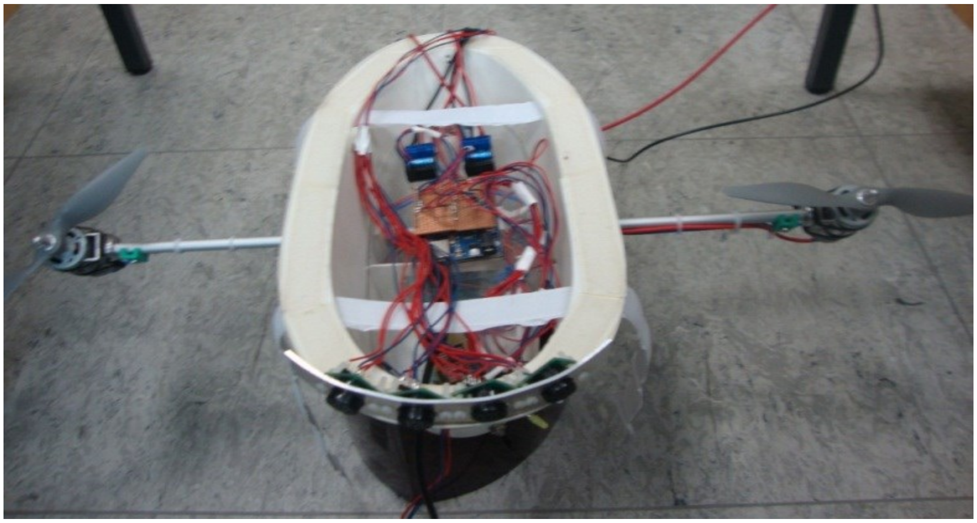
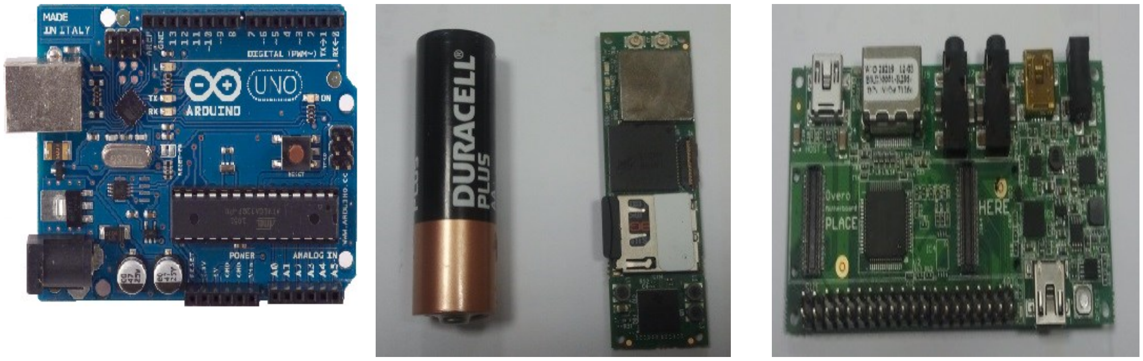


Figure 3. 6 **Gondola onboard unit.**

1. The Main Unit (MU): This unit is the core of the blimp system which is distributed among the Gumstix-Overo-Air-COM and Arduino-UNO. Arduino-Uno is a microcontroller board based on the ATmega328 as it is shown in Fig. 3.7A. Indeed, this µcontroller will be used for low level control and it was chosen due to its ability to interface with other components in the system as well as its ability to handle several simple string commands in order to control the power of the main motors of propellers and the setting of the servo motor for the shaft. On the other hand, the Gumstix computer-on-module will be used for high level controller and image processing. It is a tiny and small single OMAP3503 based on Texas Instruments OMAP3 systems on chip built on the ARM Cortex-A8 architecture as it is illustrating in Fig. 3.7B. It has a weight of 5.6g and communicates via 802.11g. Gumstix uses full embedded Linux software framework and

in order to have a Linux image for the Gumstix we need to consider some steps. First, fetch and track the dependencies, cross-compile packages and build complete images by using BitBake (Appendix A). After building, the image and the kernel are transferred to the Gumstix through a serial connection or through Ethernet network. However, for simplicity a Summit Gumstix expansion board has been used. It forms the interface between the MU and other components. This board provides some ports for generic devices communicating via UART or SPI, 2 connectors for I2C devices and power supply for the complete system. It has a weight of 16.5g and it is illustrating in Fig. 3.7C.

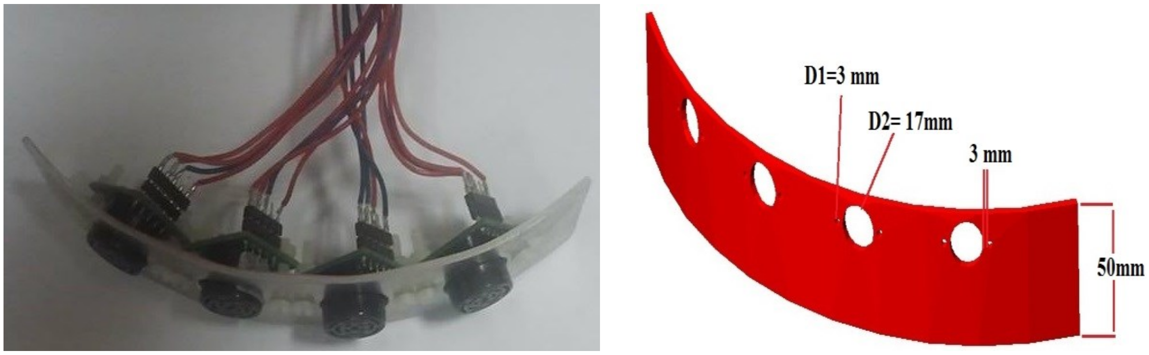


A. Arduino Uno. **B. Gumstix Overo Air.** **C. Summit Gumstix Board.**

Figure 3. 7 **The core of the blimp system.**

2. Actuators and Motors: In order to control the pitch and thrust two motors are mounted on each side of the gondola. They are attached to a shaft which can be rotated up to 180 degree by a servo. The third motor in the blimp robot system is mounted on the tail to control the yaw. However, to control the speed of each motor, a driver based on discrete MOSFET H-bridge was used. The motor driver will enable bidirectional control of one DC brushed motor and it supports a wide range of voltage from 5.5 to 30 V.
3. Sensors: the most fundamental requirement for autonomous navigation of any robot is the ability to localize itself. In blimp robot, information about the (roll, pitch, and yaw) angles are measured by the Inertial Measurement Unit (IMU). The information obtained from this IMU are used as input for predictive filters to estimate the current position of the blimp robot based on the previous position of the blimp. We should note that GPS-based localization is not applicable because our blimp system is targeted on indoor

environments. In addition, four ultrasonic sensors SRF02 are mounted in front side of gondola to be used for obstacles avoidance. A light metal sheet has been designed using AutoCAD software, and then it was made in the Siegen University Labs. Actually, many experiments have been done to test the characteristics of the SRF02 sensors. It had been found that using 18 degree between any two adjacent sensors is the optimal angle to mount them. This decision was based on the Ultrasonic sensor characteristics after experimental results. These four sensors will be mounted on this arc as shown in Fig. 3.8. The altitude during the flight and mission was verified and controlled via the fifth ultrasonic sensor that is downward-facing mounted at gondola's bottom. These sensor modules are attached to the I2C bus and have a weight of 4.6 g each. For tracking purposes, a Caspa camera sensor has been mounted in front of the gondola. It has a weight of 22.9g. It is using an optical filter to cut out the IR range and receives only visible spectrum light. It is based on MT9V032 sensor that has some good features. First, it has maximum frame rate 60 fps at full wide-VGA resolution 752x480 with angle of view 94.4^{o}. Second, the data output is 10-bit parallel with extended light sensitivity. Finally, this sensor is global shutter that can be electronic or mechanical shutter. Whilst this shutter system is off, the pixels still could be read because the camera store their charges. On the other hand, if the shutter is open, the vision sensor will collect the light to read the pixel one by one, and then make the pixels ready for the next frame. The main advantage of the global shutter over the rolling one is that the former exposes the entire image simultaneously and when moving will not create a distorted picture.



A. Four Ultrasonic Sensors **B. The Arc metal sketch**

Figure 3. 8 **The arrangement of the ultrasonic sensors.**

3.2.2. Software Architecture

The Open-Embedded software with reference to the Operating System (OS) and the communication has high flexibility and maintainability, and a fast configuration. This software is not only building framework software for embedded Linux, but also it offers a cross-compile environment which allows developers to create complete Linux distributions for any embedded systems. For Gumstix device, the Open-Embedded is used to track and fetch dependencies, cross-compile packages, and then build complete images using BitBake. After build the image and the kernel by combining the Open Embedded and BitBake, and then they are transferred to the Gumstix. This kernel, the core of the OS, is not only playing a significant role as a bridge between applications and the actual data processing, but it has also the responsibility to manage the systems resources and to facilitate the communication between software component and hardware. Despite its small dimensions the Gumstix-Overo-Air board runs a full Linux OS that provides several interfaces to access the connected hardware and simplifies the configuration of the blimp as a wireless client of the network. Fig. 3.9 shows the overall system architecture. Whilst the Atmega328 will handle low level controller system, the Gumstix will handle the high level controller such as fuzzy sets models, fuzzy controllers, and image processing. The I2C is a common peripheral bus for embedded systems and it was found that it is one of the easiest ways to design the communication between the Gumstix and Arduino board. The I2C bus in this system is initialized at 400 kHz since the Arduino appears to work perfectly without any problems at this speed. However, the logic levels coming from the Gumstix expansion board are 1.8V and most I2C devices will require 5V or 3.3V on the I2C (SDA and SCL). Therefore, physically in order to connect Gumstix with Arduino a logical level converter for the voltage transition had been used as shown in Fig. 3.10. The Gumstix is processing and handling the information of the sensors and executing the commands to Arduino to generate the Pulse Width Module (PWM) to control the motors and the setting of the servo. More information about the Gumstix board and embedded Linux system can be found in Appendix A.

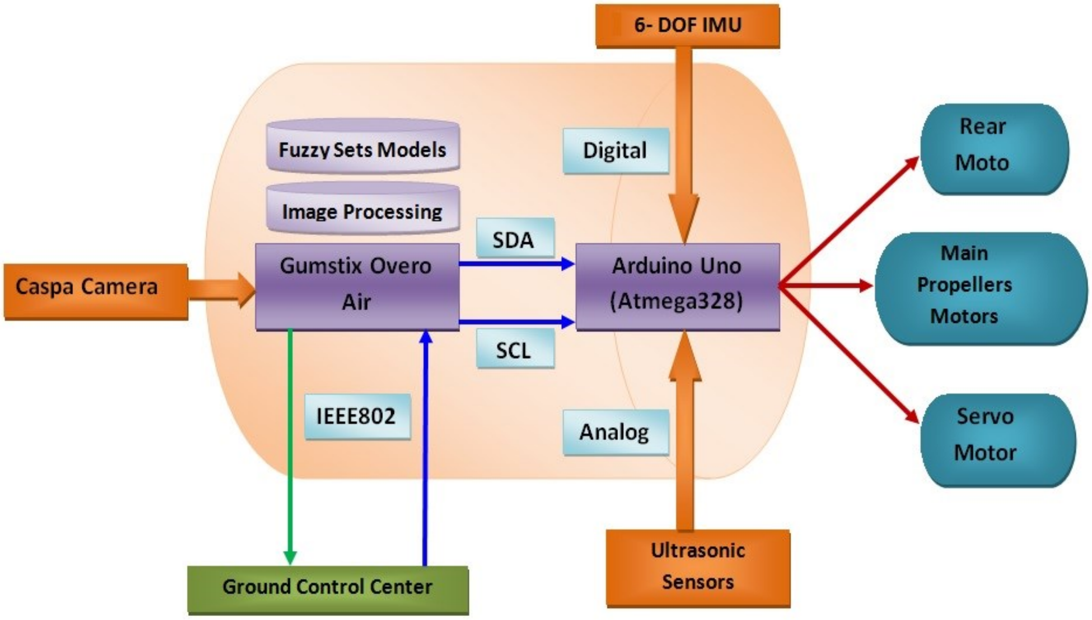


Figure 3. 9 The overall system architecture.

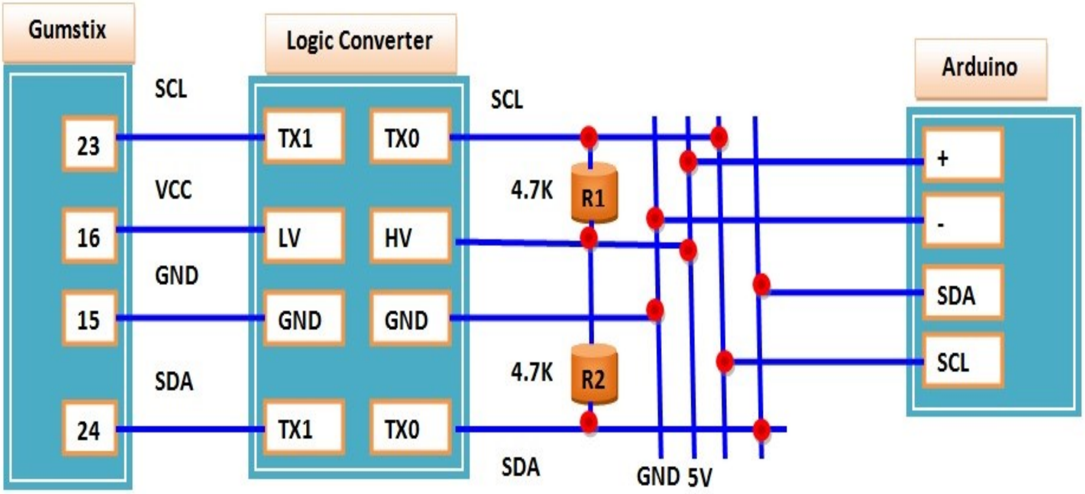


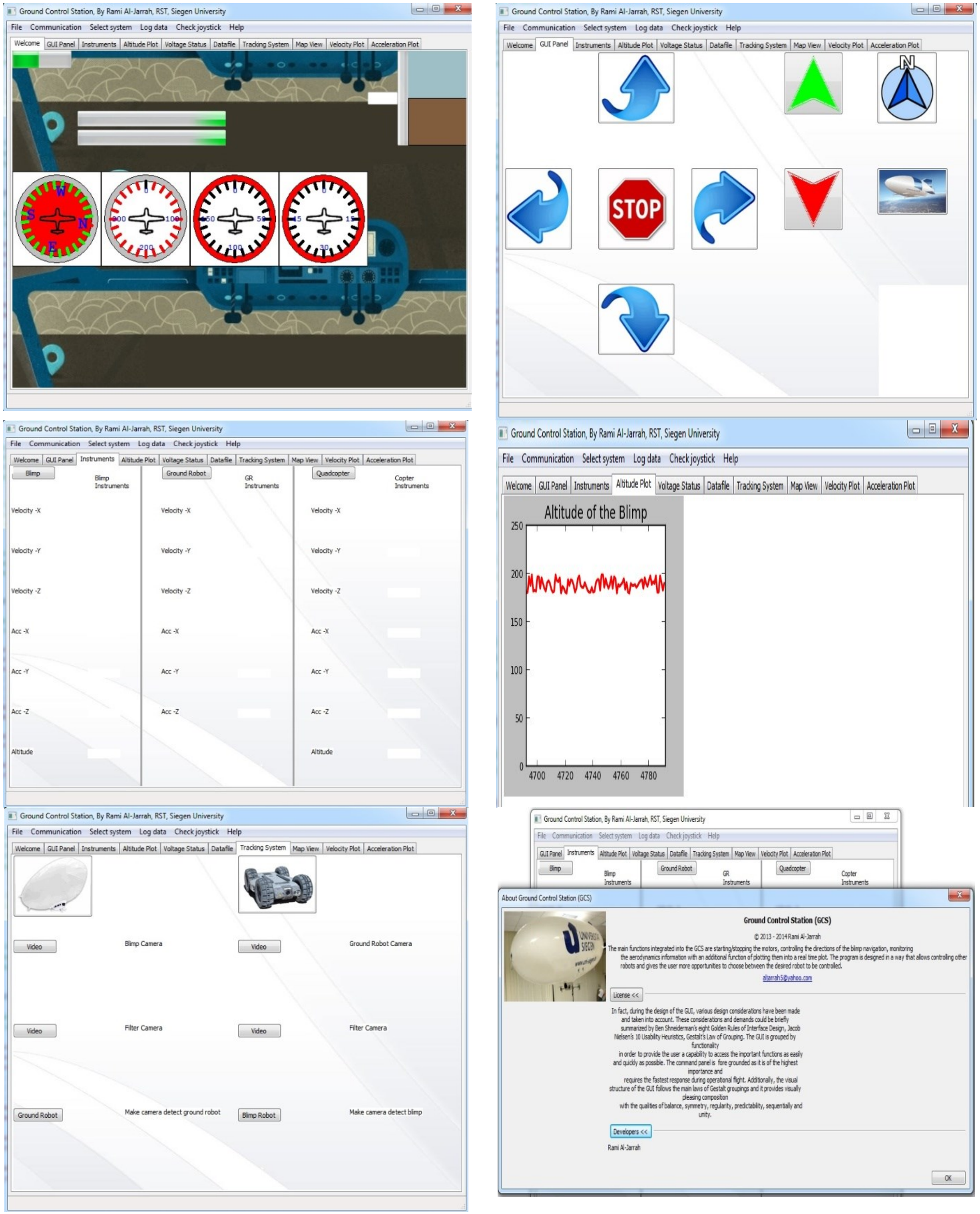
Figure 3. 10 I2C protocol between Gumstix and Arduino.

3.3. Ground Control Station

The main functions integrated into the Ground Control Station (GCS) are starting/stopping the motors, controlling the directions of the blimp navigation, monitoring the aerodynamics information with an additional function of plotting them into a real time plot. The program is designed in a way that can perform our desire as well as to allow controlling other robots in the system and gives the user more opportunities to choose between the desired robots to be controlled as it is illustrating in Fig. 3.11.

The tool used for creating the Graphical User Interface GUI was written by the Python language. The Python is high level language which has many advantages and several attractive features such as it is more dynamic than compiled languages which means at run-time the user can generate the codes and add new variables to classes. The Python is built-in run-time that can check and detect any bugs (errors) in the code by examining the structural elements of this code to decrease development time. The Python's modules and packages system makes it well suited for large-scale development projects. It is very powerful integration of libraries for sending and receiving information to distanced devices. A main role in the program has the WxPython, which is a GUI toolkit for the cross-platform application programming interface for the Python language.

In fact, during the design of the GUI, various design considerations have been made and taken into account. These considerations and demands could be briefly summarized by Ben Shneiderman's eight Golden Rules of Interface Design (Appendix B. Part A), Jacob Nielsen's 10 Usability Heuristics (Appendix B. Part B), Gestalt's Law of Grouping (Appendix B. Part C). The GUI is grouped by functionality in order to provide the user a capability to access the important functions as easily and quickly as possible. The command panel in the GUI has the highest priority during operational flight because it is required fast response during the mission. In addition, the visual GUI's shape and structure achieves the Gestalt laws and it gives a nice visually appearance with high qualities of some criteria such as symmetry, unity and sequentially. Details about the GCS , GUI frames and wedges could be found in Appendix E.



Ground Control Station (GCS)

© 2013 - 2014 Rami Al-Jarrah

The main functions integrated into the GCS are starting/stopping the motors, controlling the directions of the blimp navigation, monitoring the aerodynamics information with an additional function of plotting them into a real time plot. The program is designed in a way that allows controling other robots and gives the user more opportunities to choose between the desired robot to be controlled.

aljarrah5@yahoo.com

In fact, during the design of the GUI, various design considerations have been made and taken into account. These considerations and demands could be briefly summarized by Ben Shneiderman's eight Golden Rules of Interface Design, Jacob Nielsen's 10 Usability Heuristics, Gestalt's Law of Grouping. The GUI is grouped by functionality in order to provide the user a capability to access the important functions as easily and quickly as possible. The command panel is fore grounded as it is of the highest importance and requires the fastest response during operational flight. Additionally, the visual structure of the GUI follows the main laws of Gestalt groupings and it provides visually pleasing composition with the qualities of balance, symmetry, regularity, predictability, sequentially and unity.

Rami Al-Jarrah

Figure 3. 11 **The GCS and GUI.**

3.4. Fuzzy Sets Model and Possibilities Distributions

The classical fuzzy semantics are interpretations of fuzzy sets that represent cognitive categories and the system measurements based on linguistic variables. However, the most common and famous problem in the fuzzy control is how to design the membership functions and the rules-base (how to design the fuzzy knowledge base). Therefore, the possibility theory which helps to design the fuzzy sets from the possibility histograms will lead to model our complex system. In this chapter, we will provide a brief introduction about the fuzzy logic and the combination between the possibility theory and the fuzzy intervals. Then, we will develop efficient, powerful and intelligent fuzzy sets models for the blimp robot based on possibilities theory and frequency distributions. These models can deal with the uncertainties and drawbacks from the system sensors to improve the blimp performance during the missions. Also, they can be used for the blimp visual control navigation purposes.

3.4.1. Introduction to Fuzzy Logic

Fuzzy sets were originally introduced by Lotfi Zadah [102] to model the uncertainty of the real world applications and to represent the imprecise and vague information. Later, he introduced fuzzy logic [103] to extend the conventional Boolean or traditional logic in order to handle the intermediate values between "true" and "false". He presented the process "fuzzification" as a methodology to generalize the specific theory from a crisp to fuzzy form by applying the "extension principle" [102]. Then, several researchers have applied these fuzzy principles on different areas such as control, decision making and pattern recognition. Not only the fuzzy logic is used and implemented to model the vague information, but also it can be used to model the complex system when they are difficult to be described mathematically. Hence, some researchers such as Sugeno [104] and others [105, 106] have expanded the fuzzy logic into fuzzy systems model. Moreover, the fuzzy model will describe the system behavior and will make the relation between the inputs and the outputs of the system in a form of **IF-THEN** rules which is known as the fuzzy knowledge rule base. In order to design the fuzzy system model there are several approaches for this purpose. The most well-known approaches are based on the knowledge of the expert, simulation studies , fuzzy clustering algorithms [107].

3.4.2. Introduction to Fuzzy Sets

Because the crisp categories of the classical logic could be ***1*** or ***0*** { either ***true*** or ***false***}, the representing of ambiguous or imprecise events are very difficult in the real world applications. This is because these crisp values could allow two possible levels of sets: the element belongs to the set ***A*** or it belongs to another set ***B*** {it is not a member ***A***} [102]. In order to solve this problem Lotfi Zadah [102] presented Fuzzy sets theory that allows representing values between ***0*** and ***1***. Every engineer knows that the real world is not yes or no, on the contrast, it contains infinite number of decisions {in between} for which classical logic is not able to cope with. There are several vagueness even in our communication languages which we are using daily to share information and knowledge among us. Fuzzy sets theory is really the appropriate theory to represent and cope with such knowledge. For example, let us consider that ***"Blimp velocity is fast",*** the ***fast*** is fuzzy in the sense since it could not be sharply defined. How can we define the concept of ***fast*** ? and how can we quantify that statement?. In the classical set theory and in fuzzy sets theory the statement ***"Blimp velocity is fast"*** can be equivalent to ***"Blimp velocity belongs to the set of fast speed"***. However, the difference between the two approaches is how someone can define what is the **degree of belief** for this velocity or how much it belongs to this set. In the classical set theory, the mathematical representation of the set of ***fast speed*** might be done by indicator function. If the definition of ***fast*** is : ***"the speed which is greater than 10m/s"***, we can formalize the answer as **true** or **false**. Considering the possible speed to be [0,∞) and ***A*** is subset denotes the ***fast speed***, ***x*** is the blimp velocity. Then, $\boldsymbol{A} = \{\boldsymbol{x}|\boldsymbol{x} \in [\boldsymbol{0}, \infty) \& \, x > 10\}$**.**The indicator functions is $\mathrm{\chi(x) = \{1 \text{ if } x \in A \text{ or } 0 \text{ otherwise }\}}$. Fig. 3.12 shows the classical set of fast speed for this example.

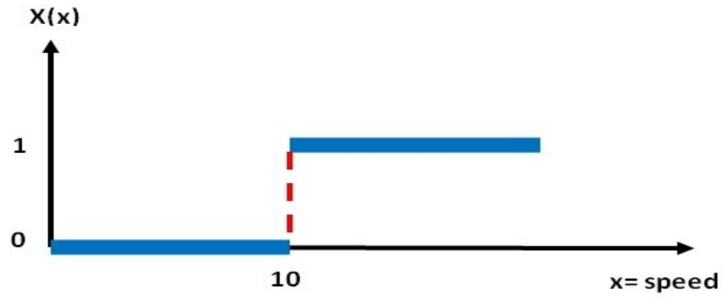


Figure 3. 12 Classical set of fast speed.

On the other hand, ***fast*** could not be defined as ordinary subset of [0, ∞) due to some justifications. How we can characterize the blimp velocity *, if it is 10.2 m/s or 12 and 15 m/s,* as a ***fast*** with different degrees. Hence, the fuzzy sets should be on ***0*** to ***1*** scale; thus, the membership should be an element in [0, 1]. In more precisely, if the speed is ***x*** then the compatibility of ***x*** with ***"is fast"*** is a matter of degree and it is totally relies on how we understand the ***"fast"*** concept or our knowledge about it. Fig. 3.13 illustrates the fuzzy set theory for the fast speed concept. the membership function **μ** which is called alternatively the characteristic function is defined on the interval [0, 1]; thus we can clearly conclude from the this figure that:

- *If the blimp speed is 8 m/s; then* $\mu_{fast}(8)=0$.
- *If the blimp speed is 10 m/s; then* $\mu_{fast}(10)=0.5$.
- *If the blimp speed is 12 m/s; then* $\mu_{fast}(12)=1$.
- *If the blimp speed is 13 m/s; then* $\mu_{fast}(13)=1$.

These previous statements are **fuzzy sets** and they are mathematical expressions which are defined the set of ***fast speed*** and give the ability to evaluate the expression ***"Blimp velocity is fast".*** Moreover, the Fuzzy Sets theory have the ability to provide framework in order to represent our vague knowledge [102] and the Fuzzy Logic will provide the framework to manipulate and to draw the inferences from this vague knowledge [103].

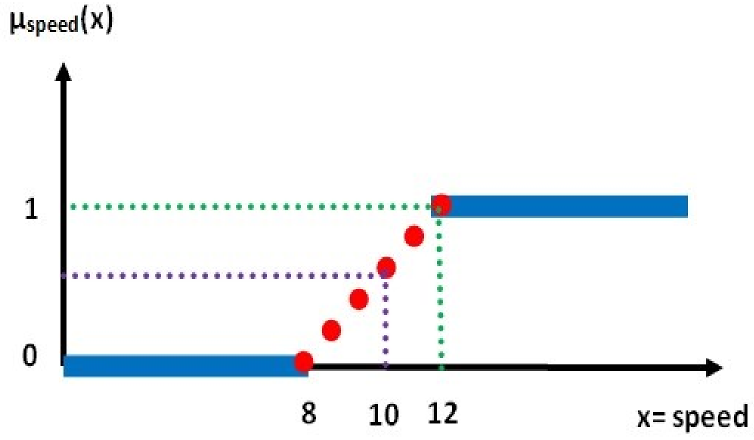


Figure 3. 13 Fuzzy set of fast speed.

As it is known, the fuzzy logic is a powerful and strong problem-solving methodology for several applications in control and information processing; more simplicity, it does resemble decision making from approximate data. In addition, it has not only the ability to provide definite conclusions from the vague information, but also every application can potentially realize some of the fuzzy logic benefits such as performance and lower cost. Therefore, there are some important and basic definitions which are important to understand the concept of the fuzzy sets, logic and control [108-113].

Definition 1 (Fuzzy Set) [102, 103, 114]***:*** Let us consider ***X*** denotes an universal set, ***A*** is the fuzzy subset of ***X*** and it is described by membership function μ_A***: X→ [0, 1]***. This means that for any element ***x*** of ***X*** there is a $\mu_A(x)$ in the interval [0, 1] with $\mu_A(x)$ represents the grade of the membership function of element ***x*** in fuzzy set ***A***. ***A*** can be determined by ***A= {(x, μA(x))| x*** ∈ ***X}***. If ***X={x₁, x₂, x₃, x₄, ..., xₙ }*** is an finite set and ***A*** is a fuzzy set in ***X*** then $A = \int \mu_A(x)/x$.

Definition 2 (Membership Functions) [102, 103, 114]: They are defined as the quantities to which linguistic value does belong to a variable. They have many shape and they can be used to define the meaning of a linguistic value such as triangular, bell, trapezoidal and Gaussian. However, the most well known and used shape is the trapezoidal membership function due to fact that it has some good characteristics such as widely used and less consumption time comparing to gaussian membership function.

Definition 2 (Height) [102, 103, 114]***:*** The height ***h(A)=*** $\boldsymbol{sup_{x \in X} \mu_A(x)}$ is the height of the fuzzy set ***A*** which is the maximum membership value that obtained by any element in that set.

Definition 3 (Normal) [102, 103, 114]***:*** The fuzzy set ***A*** will be normal when ***h(A)=1*** and in case it is ***h(A)<1*** it could be called subnormal.

Definition 4 (Convexity) [102, 103, 114]: Let us consider that $x_1, x_2, x_3 \in X, x_3 \geq x_2 \geq x_1 \rightarrow \mu_A(x_2) \geq \min(\mu_A(x_1), \mu_A(x_3))$ this is called the convex of fuzzy set.

Definition 5 (Fuzzy Number) [102, 103, 114]: The fuzzy number is a fuzzy set of real line with a convex, normal and piecewise continuous membership function of bounded support.

Definition 6 (Fuzzy Point) [102, 103, 114]: Let us consider ***A*** is a fuzzy number; in case that the ***supp(A)=*** $\boldsymbol{x_0}$; thus ***A*** is called fuzzy point.

Definition 7 (Crossover Point) [102, 103, 114]: Let us consider ***A*** is a fuzzy number; in case that the $\boldsymbol{\mu_A(x)=0.5}$; thus ***x*** is called crossover point.

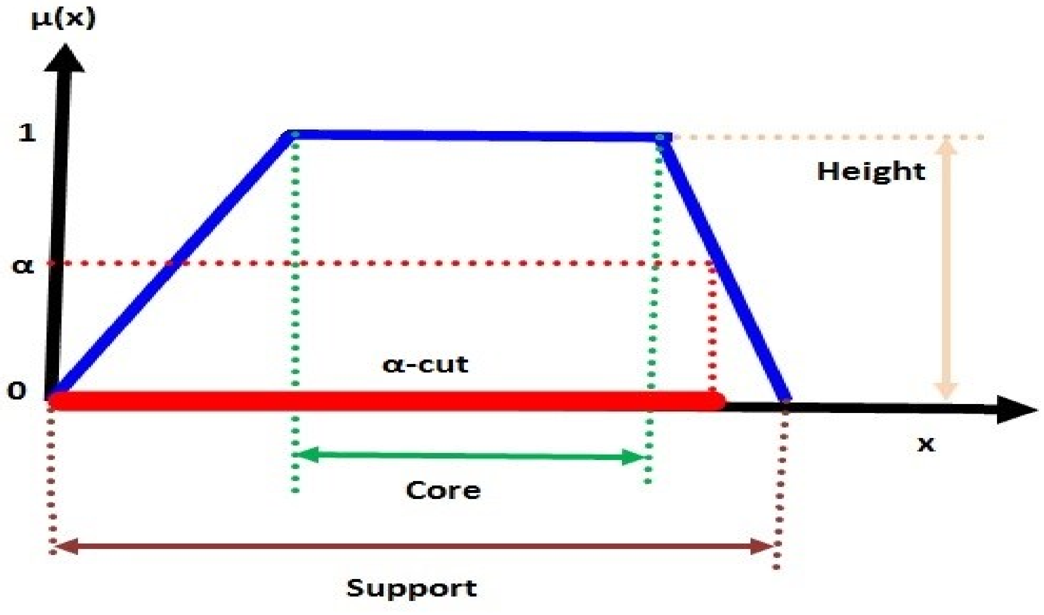


Figure 3. 14 The height, core, support, α- cut for fuzzy set A.

Definition 8 (Extension Principle) [102, 103, 114]: We need to mention here that the extension principle is very important in fuzzy set theory. It has ability to provide the general method to extend the non-fuzzy mathematical concepts to cope with fuzzy quantities; then it may allow the extension of a mapping ***f*** from points in ***X*** to fuzzy subsets of ***X*** as the following:

$$f(A) = f\left(\frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \cdots + \frac{\mu_n}{x_n}\right) = \frac{\mu_1}{f(x_1)} + \frac{\mu_2}{f(x_2)} + \cdots + \frac{\mu_n}{f(x_n)} \qquad 3.1$$

Definition 9 (Operations on Fuzzy Sets) [102, 103, 114]: There are three basic operations for fuzzy sets [102]. First, the complement operation $\bar{A}$ of the fuzzy set **A** with respect to the universal set ***X*** as it is shown in Fig. 3.15 , and it is defined as $(\forall x \in X)\left(\mu_{\bar{A}}(x)\right) = 1 - \mu_A(x)$. The second operation is the intersection $\{A \cap B\}$ for fuzzy sets ***A*** and ***B*** in the universal set ***X*** and it is defined as $\mu_{(A\cap B)}(x) = \min[\mu_A(x), \mu_B(x)]$. The third operation is the union $\{A \cup B\}$ for fuzzy sets ***A*** and ***B*** in the universal set ***X*** and it is defined as $\mu_{(A\cup B)}(x) = \max[\mu_A(x), \mu_B(x)]$. The intersection and union operations are shown in Fig. 3.16.

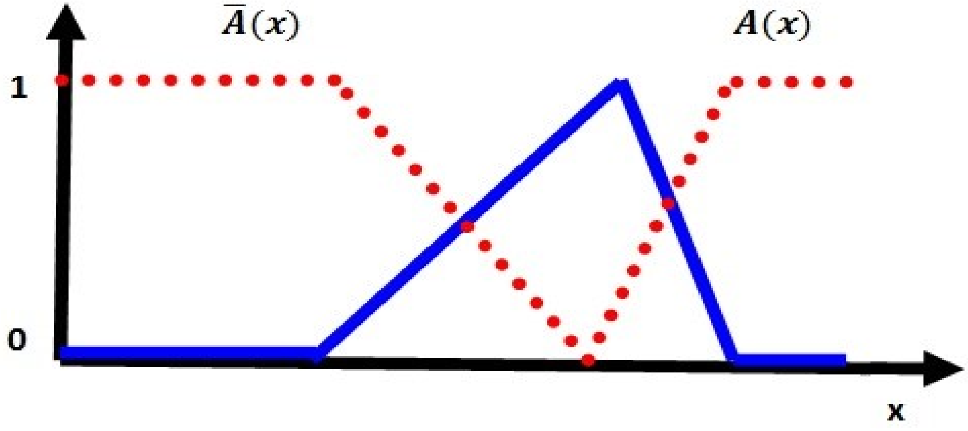


Figure 3. 15 **Fuzzy complement.**

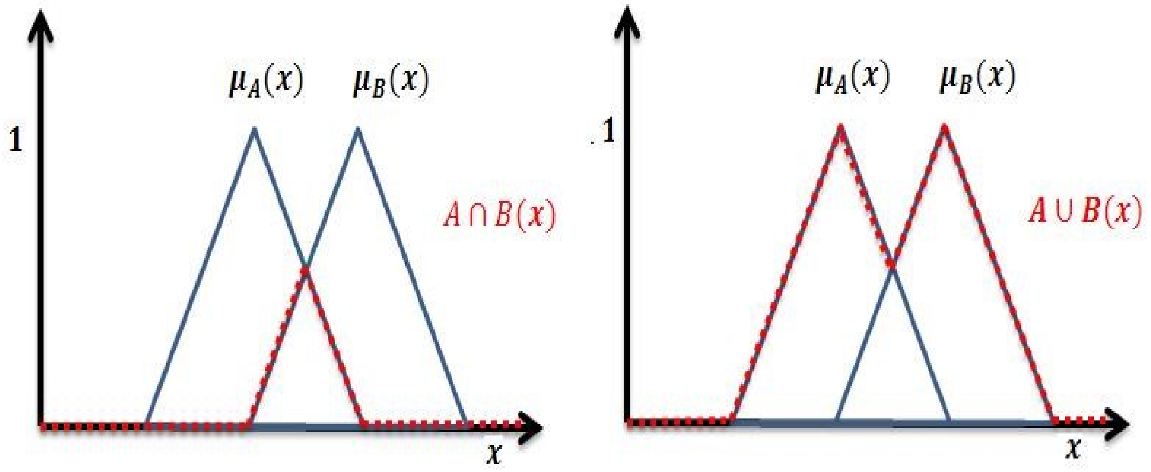


Figure 3. 16 **The Intersection operation (left) and Union operator (right).**

Definition 10 (The concept of Linguistic Variables): The linguistic variables were defined by Zadah [114, 115, 116] as the variables in which the values are sentences or words in natural language. In our research, the speed of the blimp robot takes some values from the set of terms such as ***Fast, Low, and Medium***. In accordance to that, the ***speed*** of the blimp is called a linguistic variable and the terms ***{Fast, Low, and Medium}*** are called linguistic values of the linguistic variables.

Definition 11 (α-cut and strong α -cut) [102, 103, 114]: If the fuzzy set ***A*** is defined on ***X*** and a number ***α ϵ [0, 1],*** the ***α-cut (*** A^{α} ***)*** and the strong ***α-cut (*** $A^{\alpha+}$ ***)*** can be called the crisp sets and they are given by:

$$A^{\alpha} = \{x|A(x) \geq \alpha\} \qquad 3.2$$

$$A^{\alpha+} = \{x|A(x) > \alpha\} \qquad 3.3$$

Definition 12 (Support) [102, 103, 114]: If we have set ***X*** and ***A*** is the fuzzy subset of ***X***, then support of ***A*** which is denoted as ***Supp(A)*** is the crisp subset of ***X*** that has all elements with non-zero membership functions grades in ***A***. The support of ***A*** might be same as strong ***α–cut*** of ***A*** for ***α =0***.

$$Supp(A) = \{x \in X | \mu_A(x) > 0\} \qquad 3.4$$

Definition 13 (Core) [102, 103, 114]: Lets ***A*** be a fuzzy subset of ***X***, the core of ***A***{***Core(A)}*** is the crisp subset of ***X*** whose all elements have one membership grades in ***A***. The support of ***A*** is the same as the ***α–cut*** of ***A*** for ***α =1***.

$$Core(A) = \{x \in X | \mu_A(x) = 1\} \qquad 3.5$$

***Definition 14 (Triangular* Fuzzy Number)** [218]: The fuzzy set could be called triangular fuzzy number if its membership function has the following form:

$$\mu_A(x) = \begin{cases} 0 & , x < a \\ \dfrac{x-a}{b-a} & , a \le x \le b \\ \dfrac{c-x}{c-b} & , b \le x \le c \\ 0 & , x > c \end{cases} \qquad 3.6$$

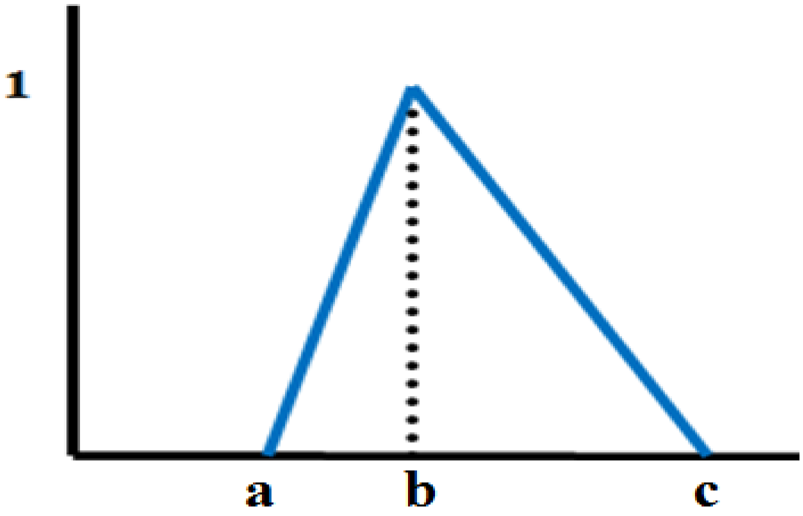


Figure 3. 17 Triangular fuzzy number.

Definition 15 (Trapezoidal Fuzzy Number) [102, 103, 114]: The fuzzy set might be called trapezoidal fuzzy, when membership function has following form:

$$\mu_{A_{ij}}(x_j) = \begin{cases} \dfrac{x_j - a_{ij}}{b_{ij} - a_{ij}} & , if\ a_{ij} \le x_j \le b_{ij} \\ 1 & , if\ b_{ij} \le x_j \le c_{ij} \\ \dfrac{d_{ij} - x_j}{d_{ij} - c_{ij}} & , if\ c_{ij} \le x_j \le d_{ij} \\ 0 & , otherwise \end{cases} \quad 3.7$$

Where **i**th is the rules and **j**th is the input variables. $a_{ij} \le b_{ij} \le c_{ij} \le d_{ij}$ are known as the breakpoints of the membership functions.

Definition 16 (Multi-dimensional) [102, 103, 114]: In order to define the fuzzy sets with more than one variable, we need to extend the fuzzy sets to have higher dimensional membership functions. This is called multi-dimensional fuzzy sets or fuzzy relations. If ***R*** is multi-dimensional fuzzy subsets of $X_1 \times X_2 \times \ldots . \times X_n$, it can be defined as:

$$R = \int_{X_1 \times \ldots . \times X_n} \mu_R(x_1, \ldots, x_n)/(x_1, \ldots, x_n) \quad 3.8a$$

Where $(X_1, \ldots, X_n)$ are continuous universes. However, if $(X_1, \ldots, X_n)$ are discrete universes then:

$$R = \sum_{X_1 \times \ldots . \times X_n} \mu_R(x_1, \ldots, x_n)/(x_1, \ldots, x_n) \quad 3.8b$$

Such a fuzzy relation can represent an association or correlation between elements of the product space. For example, $\mu_R(x_1, x_2, \ldots, x_n)/(x_1, x_2, \ldots x_n)$ is interpreted as the degree of membership of $(x_1, x_2, \ldots ., x_n)$ in ***R***. A simple example of the fuzzy relation on $X_1 = \{2, 3, 4\}$ and $X_2 = \{2, 3, 4\}$ is called ***"approximately equal"*** and it can be defined as:

$R = 1/(2,2) + 1/(3,3) + 1/(4,4) + 0.5/(2,3) + 0.5/(3,2) + 0.5/(3,4) + 0.5/(4,3)+$
$0.1/(2,4) + 0.1/(4,2).$

The membership function $\boldsymbol{\mu_R}$ of this relation can be described by:

$$\mu_R(x_1, x_2) = \begin{Bmatrix} 1 & x_1 = x_2 \\ 0.5 & |x_1 - x_2| = 1 \\ 0.1 & |x_1 - x_2| = 2 \end{Bmatrix}$$

$$R = \begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array}\left(\begin{array}{ccc} 2 & 3 & 4 \\ 1 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 \\ 0.1 & 0.5 & 1 \end{array}\right)$$

Definition 17 (The Intersection) [102, 103, 114]: The intersection for two fuzzy relations ***R*** and ***S*** on ***X*Y*** can be defined by:

$$\mu_{R\cap S}(x,y) = \min(\mu_R(x,y), \mu_S(x,y)) \qquad 3.9$$

Definition 18 (The Union) [102, 103, 114]: The Union for two fuzzy relations ***R*** and S on ***X*Y*** can be defined by:

$$\mu_{R\cup S}(x,y) = \max(\mu_R(x,y), \mu_S(x,y)) \qquad 3.10$$

For example, let us consider that the speed ***V*** of the blimp robot has three linguistic values or membership functions ***{ Fast, Medium, Low}*** with fuzzy subset ***A_1, A_2, A_3*** , respectively. The trapezoidal membership functions on interval **[0, 10]** are the reasonable expression of these fuzzy subsets as it is illustrating in Fig. 3.18.

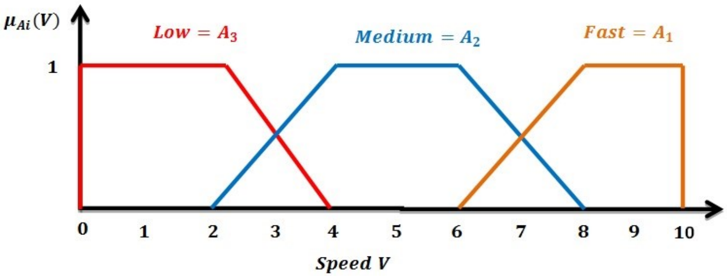


Figure 3. 18 The trapezoidal membership functions for the speed.

The trapezoidal for the previous fuzzy subsets can be formulated the following:

$$\mu_{A1}(V) = \begin{cases} 0, & V \le 6 \\ 1, & 8 \le V \le 10 \\ \dfrac{V-6}{2}, & 6 \le V \le 8 \end{cases}$$

$$\mu_{A2}(V) = \begin{cases} 0, & V \le 2 \; or \; V \ge 8 \\ \dfrac{V-2}{2}, & 2 < V < 4 \\ 1, & 4 \le V \le 6 \\ 1 - \dfrac{V-6}{2}, & 6 \le V \le 8 \end{cases}$$

$$\mu_{A3}(V) = \begin{cases} 1, & V \leq 2 \\ 0, & 4 \leq V \\ 1 - \dfrac{V-2}{2}, & 2 \leq V \leq 4 \end{cases}$$

Now, we can find that the complement $\boldsymbol{\mu_{\overline{A2}}}$ is given by the following:

$$\mu_{\overline{A2}}(V) = \begin{cases} 1, & V \leq 2 \; or \; V \geq 8 \\ 1 - \dfrac{V-2}{2}, & 2 < V < 4 \\ 0, & 4 \leq V \leq 6 \\ \dfrac{V-6}{2}, & 6 \leq V \leq 8 \end{cases}$$

It is clearly that $\boldsymbol{\mu_{\overline{A2}}} = \boldsymbol{\mu_{A1}} \cup \boldsymbol{\mu_{A3}}$ as it is illustrating in Fig. 3.19. Additionally, the **Medium** is means **not Fast *AND* not Low** $\boldsymbol{\mu_{A2}} = \boldsymbol{\mu_{\overline{A1}}} \cap \boldsymbol{\mu_{\overline{A3}}}$ as it is shown in Fig. 3.20.

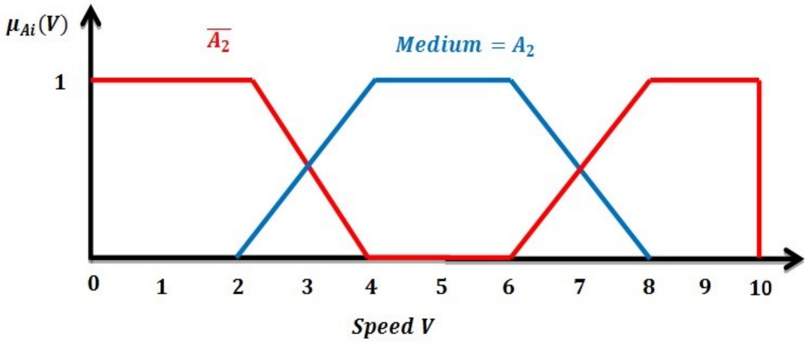


Figure 3. 19 **The trapezoidal membership functions for $\mu_{\overline{A2}} = \mu_{A1} \cup \mu_{A3}$.**

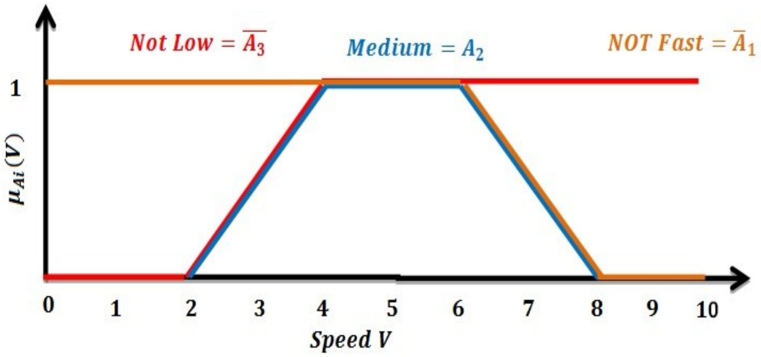


Figure 3. 20 **The trapezoidal membership functions for $\mu_{A2} = \mu_{\overline{A1}} \cap \mu_{\overline{A3}}$**

The question comes to mind now what are the definitions that might be chosen to implement the intersection, union or complement in the fuzzy set theory?. Schweizer and Sklar [117] have developed a Triangular norms (**T-norms**) and Triangular conorms (**T-conorms**) in statistical metric spaces to be used as fuzzy set intersection and union operators [118, 119, 120].

Definition 19 (Triangular Norms (T-norms)) [117]: Lets ***A*** and ***B*** are fuzzy subsets of a universal set **U**; then the truth evaluation of ***" A and B"*** is expressed as an element of the interval [0,1]. In order to extend this map we might use this expression: $\mu_{A\cap B}(u) = min\{\mu_A(u), \mu_B(u)\}$ where $u \in U$. Now, if ***A*** and ***B*** are defined on different universe of discourse **U** and **V**; then $\mu_{A\cap B}(u,v) = min\{\mu_A(u), \mu_B(v)\}$ and $(u,v) \in U \times V$. If we consider that $T: [0,1] \times [0,1] \rightarrow [0,1]$ is a function that transforms the membership functions of two fuzzy sets ***{A and B}*** to membership function of the intersection between them; then $T[\mu_A(u), \mu_B(u)] = \mu_{A\cap B}(u)$ and this **T-norms** will be used as a possible connectives for fuzzy conjunction **"and"**. The **T** function can be valid for the intersection if it is satisfy some axioms and properties. First, boundary condition $\{T(0,0) = 0; T(a,1) = T(1,a) = a\}$. Second, the value of the conjunction will be decreased when the value of at least one proposition increased $T(a,c) \leq T(b,d)$ when $a \leq b, and\ c \leq d$. Third, associativity which means that the order of the conjunction is not important $T\big(a, T(b,c)\big) = T(T(a,b),c)$. Finally, the **T-Norms** should be satisfy the commutativity which means the conjunction is not based on the order of a and b $T(a,b) = T(b,a)$.

Definition 20 (Triangular Conorms (T-conorms)) [117]: If **T** is a **T-norm**; then its ***n*** dual **t-conorm** ***S*** is given by: $S(a,b) = n(T(n(a), n(b)))$. ***S*** is ***n-dual*** of some **T-norm** with respect to some strong negations ***n***.

3.4.3. Fuzzy Implication Functions

Practically, the description of the causal link between the multi domains ***W***, ***U***, and ***V*** is not always available [121]. However, this description exists in terms of fuzzy linguistic rules of the form: *IF X is **A**$_n$ and Y is **B**$_n$ THEN Z is **C**$_n$.* Where "***X*** is ***A$_n$***", "***Y*** is ***B$_n$***" are the fuzzy propositions and they are called the premises, "***Z*** is ***C$_n$***" is fuzzy proposition and it is called the consequence. The fuzzy relation for the rule can be denoted as $(\mathrm{A_i} \cap \mathrm{B_i}) \rightarrow \mathrm{C_i}, \forall_{\mathrm{i}}= 1, \dots, \mathrm{n}$. Now, in order to combine the fuzzy premise propositions and to combine the fuzzy consequence proposition, the **T-norms** is the most suitable operator. However, to combine the premise and the consequence for the fuzzy rule the implication function ***I*** have been used. The modeling of such relation is always depending on the use of fuzzy logic implication function. The implication function is represented as the following:

$$\mu_{(A\cap B)\to C}(u,v,w) = I\left(T\big(\mu_A(u),\mu_B(v)\big),\mu_C(w)\right) = I(T(a,b),c). \qquad 3.11$$

One of the well-known implication function is ***QL-implication*** which is defined as $I_{QL}(a,b) = S(n(a),T(a,b))$ [122]. In fuzzy set theory, the ***QL-implication*** is used to represent the linguistic rules ***IF-THEN-ELSE*** [122]. For example: **IF *X is A* THEN *Y is B* ELSE IF *X is not A* THEN *Y is C*** this should be represented as : $A \to B = (A \cap B) \cup (\bar{A} \cap C)$. When **C** is " undefined" then $A \to B = (A \cap B)$ which is known as Mamdani implication in fuzzy control [123, 124].

3.4.4. Fuzzy Reasoning

This process is defined as the process in which the new fuzzy facts can be concluded from existing facts and causal links between them as it is shown in Fig. 3.21. There are two well-known fuzzy inference reasoning procedure. First, the Compositional Rule of Inference (CRI) in which the fuzzy relation is used to represent explicitly the connection between the fuzzy propositions. Therefore, consider that the fuzzy relation ***R*** links between two universe discourse ***U*** and ***W***, then if fuzzy subset ***A*** is subset of ***U*** it might be conclude that the fuzzy subset ***C*** is subset of ***W***.

$$\begin{array}{c} C = A \circ R \\ \underline{IF\ \ A'} \\ C^{*} = A' \circ R \end{array}$$

The second and important process is the Generalized Modus Ponens (GMP) which uses **IF-THEN** rules which implicitly represents the fuzzy relation when the causal link between the domains ***U*** and ***W*** is not completely known. The single antecedent inference with fuzzy input and fuzzy output can be given as the following:

$$\begin{array}{ll} Rule: \boldsymbol{IF}\ X\ is\ A\ \boldsymbol{THEN}\ \ Y\ is\ C & \\ \underline{Observation: \qquad\qquad} & \underline{X\ is\ A'} \\ Conclusion: & Y\ is\ C^{*} \end{array}$$

$$\begin{array}{ll} Rule: & A \rightarrow C \\ \underline{Observation:} & \underline{A'} \\ Conclusion: & C^* \end{array}$$

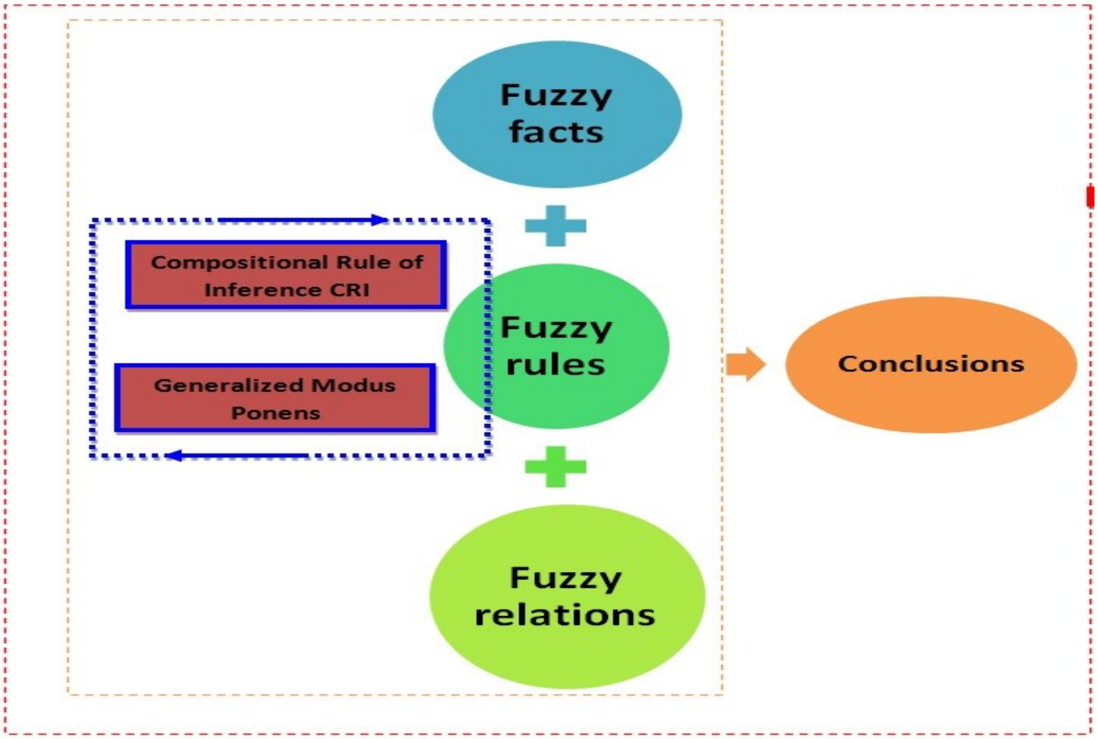


Figure 3. 21 **Fuzzy reasoning process.**

There are two types for reasoning: **the Mamdani type approach** and **the Sugeno approach** [125, 126]. In Mamdani approach [125], the relationship between the fuzzy variables ***A*** and ***C*** is expressed as $\mu_{A\times C}(u,w) = \min\{\mu_A(u), \mu_C(w)\}$. Now, given ***A***$^{'}$ then ***C**** can be computed by using **GMP** as the following:

$$\mu_{C^*}(w) = \bigvee_u \mathrm{C}(\mu_{A'}(u), \mu_R(u,w))\ \forall w \in W \qquad 3.12$$

If ***C*** is taken as ***min*** in Mamdani approach as it is illustrating in Fig. 3.22, then:

$$\mu_{C^*}(w) = \bigvee_u min(\mu_{A'}(u), min(\mu_A(u), \mu_C(w)))\ \forall w \in W \qquad 3.13$$

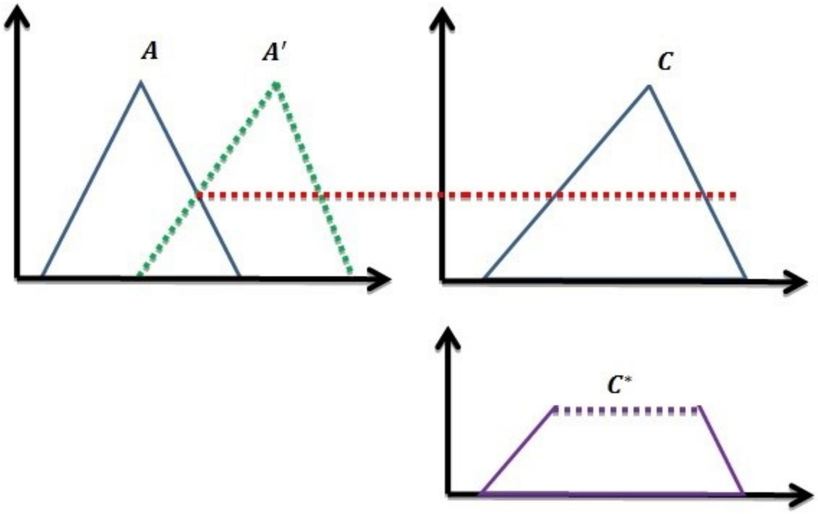


Figure 3. 22 Fuzzy inference using Mamdani approach.

In case that there is a singleton observation $\boldsymbol{u_o}$ and $\boldsymbol{\mu_{A'}(u_o) = 1}$ as it is shown in Fig. 3.23***;*** then***:***

$$\mu_{C^*}(w) = min(\mu_A(u_o), \mu_C(w)\forall w \in W \qquad 3.14$$

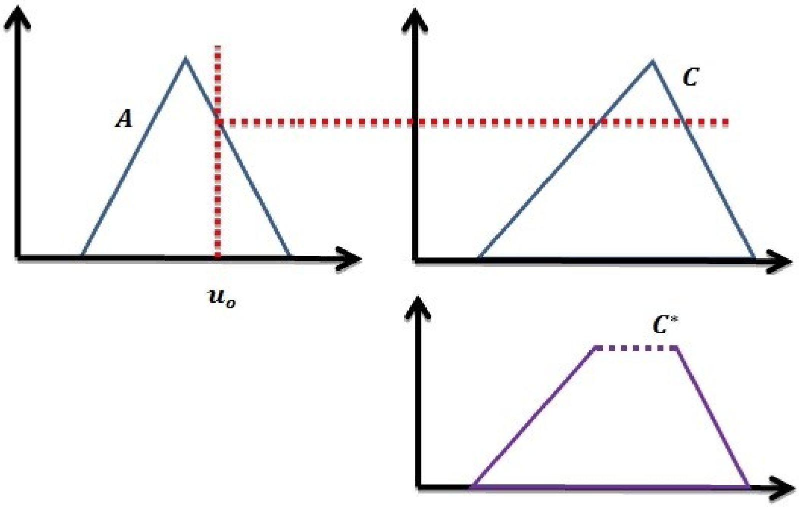


Figure 3. 23 Fuzzy inference using Mamdani approach for singleton observation.

Now in case there are two antecedent as it is shown in Fig. 3.24, the Mamdani inference is obtained as the following:

$$\mu_{C^*}(w) = \bigvee_u \bigvee_v [(\mu_A(u) \wedge \mu_{A'}(u)) \wedge (\mu_B(v) \wedge \mu_{B'}(v)) \wedge \mu_C(w)] \, \forall w \in W \qquad 3.15$$

And if it has singleton observation then it can be rewritten as the following:

$$\mu_{C^*}(w) = min(\min(\mu_A(u_o), \mu_B(v_o)), \mu_C(w))\forall w \in W \qquad 3.16$$

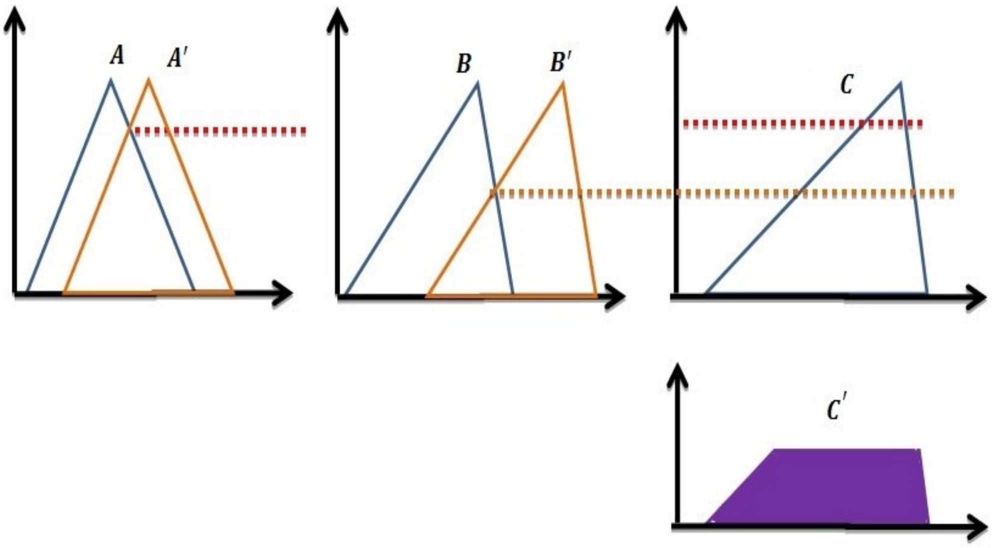


Figure 3. 24 Fuzzy inference using Mamdani for singleton observation two antecedent.

3.4.5. Fuzzy Rule -Based System

The fuzzy system which is called the fuzzy rule based system is composed to four main stages: **Fuzzifier, Rule base, Inference Engine, and Defuzzifier** as it is shown in Fig. 3.25.

1. ***The fuzzification stage*** is defined as the process to convert the crisp value input $\boldsymbol{u_0}$ to a fuzzy point $\boldsymbol{A}^{'}$ which is defined as the following:

$$\mu_{A'}(u) = \begin{cases} 1, if\ u = u_o \\ 0, otherwise \end{cases}$$

2. ***The rule-base***: we have shown in the previous section how to represent the single rule by using fuzzy relation and how to infer from only one fuzzy rule. Therefore, we will show now and discuss the general case when there are more than one rule. Let us consider that we have a fuzzy system which is described by the fuzzy rules as the following:

$$\begin{array}{llll}
\mathbf{r_1}: & \textbf{IF } X \text{ } is \text{ } A_1 & \textbf{AND } Y \text{ } is \text{ } B_1 & \textbf{THEN} \quad Z \text{ } is \text{ } C_1 \\
 & \quad\ldots\ldots & \quad\ldots. & \quad\ldots \\
\mathbf{r_k}: & \textbf{IF } X \text{ } is \text{ } A_k & \textbf{AND } Y \text{ } is \text{ } B_k & \textbf{THEN} \quad Z \text{ } is \text{ } C_k \\
 & \quad\ldots. & \quad\ldots. & \quad\ldots \\
\mathbf{r_n}: & \textbf{IF } X \text{ } is \text{ } A_n & \textbf{AND } Y \text{ } is \text{ } B_n & \textbf{THEN} \quad Z \text{ } is \text{ } C_n
\end{array}$$

Therefore, the translation of this rule base into a fuzzy relation is done by constructing the fuzzy relation ***R****n* for each rule ***r****k* ; and then combine these relation into single fuzzy relation ***R***. This process for combining the fuzzy rules into a fuzzy relation is called aggregation. This aggregation relies on the type of the relation used to represent the rule in the rule base system. In case of Mamdani approach ***R****k* is represented by using conjunction operator; then the aggregation is done by using disjunction operator.

$$R = \bigcup_k R_k \tag{3.17}$$

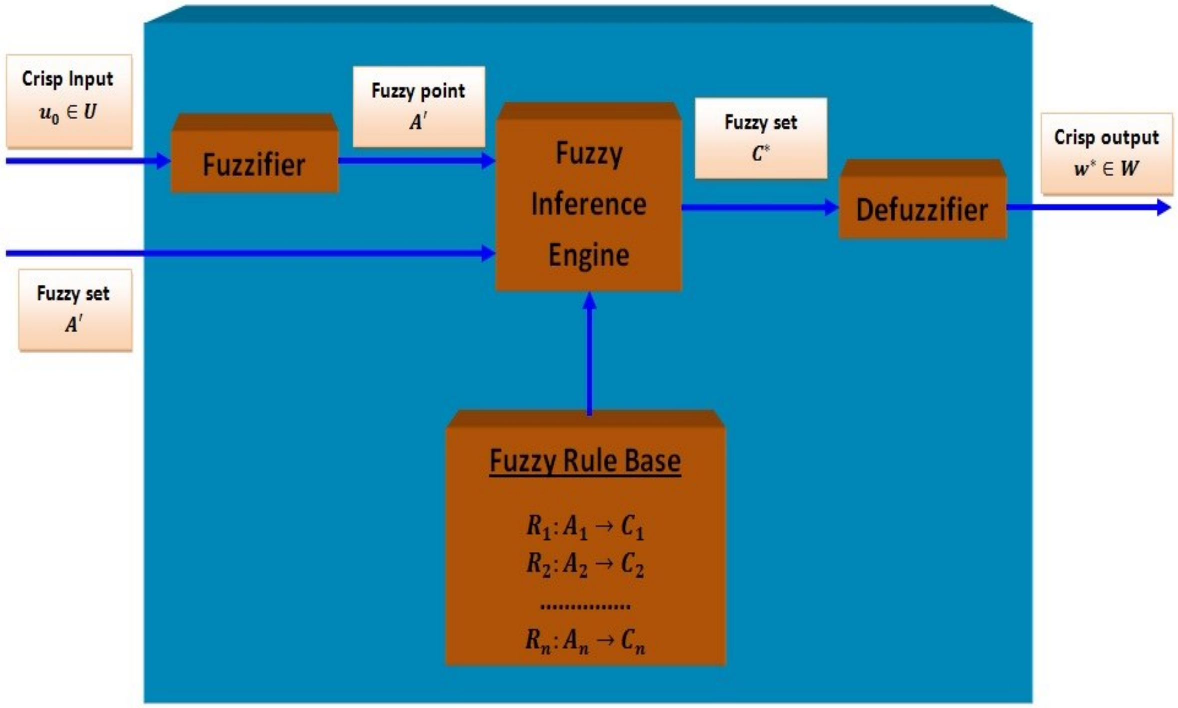


Figure 3. 25 Fuzzy rule-based system.

3. ***Inference engine***: we will cover and discuss the case when the rule is represented by Mamdani (conjunction relation). In this case there are two important approaches for the inference:
   - ***Approach 1:*** Infer and combine which is give by the following

$$C^* = \bigcup_{k=1,\ldots,n} A'o\,(A \to C) = \bigcup_{k=1,\ldots,n} A'o\,R_k \qquad 3.18$$

- ***Approach 2:*** Combine and Infer:

$$C^* = A'o \bigcup_{k=1,\ldots,n} (A_k \to C_k) = A'o\,R \qquad 3.19$$

In Mamdani approach the combine and infer is same as infer and combine and this result is true with both singleton and fuzzy inputs [127, 128].

$$\bigcup_{k=1,\ldots,n} A'^{\circ}R_k = A'^{\circ}(\bigcup_{k=1,\ldots,n} R_k) \qquad 3.20$$

4. ***Defuzzification stage:*** It is defined as the process which converts the fuzzy set obtained from inference mechanism process into crisp value. This process is very important when the crisp values are needed instead of fuzzy set in some applications especially in fuzzy control. There are many methods for defuzzification as it is described in Fig. 3.26, but the most often methods used are the following:

- ***Center of Gravity***: The crisp value is obtained by the following:

$$w^* = \frac{\sum_i \mu_{C^*}(w_i) w_i}{\sum_i \mu_{C^*}(w_i)} \qquad 3.21$$

- ***First of Maxima***: the defuzzified value of the fuzzy set is the smallest maximizing element:

$$w^* = \min\{w_i | \mu_{C^*}(w_i) = \max(\mu_{C^*}(w_i))\} \qquad 3.22$$

- ***Middle of Maxima:*** in this method the value set is calculated as the following:

$$w^* = \frac{\sum_{i=1}^{N} w_i}{N} \qquad 3.23$$

$$\mu_{C^*}(w_i) = Max_i(\mu_{C^*}(w_i)) \qquad 3.24$$

where N is the number of the elements of C*, ($\mu_{C^*}(w_i)$ is the maximal.

There are three important cases are covering in this study about the inference system with Mamdani approach: two rules with single antecedent, two rules with singleton observation, and two rules with two observations. In Mamdani approach and in case the inference has two rules and single antecedent case; then the behavior of the fuzzy system could be described as the following:

$$\begin{array}{lllll} \mathbf{r_1}: & \textbf{IF} & X\ is\ A_1 & \textbf{THEN} & Z\ is\ C_1 \\ \mathbf{r_2}: & \textbf{IF} & X\ is\ A_2 & \textbf{THEN} & Z\ is\ C_2 \end{array}$$

If we consider that the fuzzy input is $A^{'}$ then:

$$\begin{array}{lr} \mathbf{r_1}: & A_1 \rightarrow\ \ C_1 \\ \mathbf{r_2}: & A_2 \rightarrow\ \ C_2 \\ Observation: & A' \\ \hline Conclusion: & C^* = combine(C_1^*, C_2^*) \end{array}$$

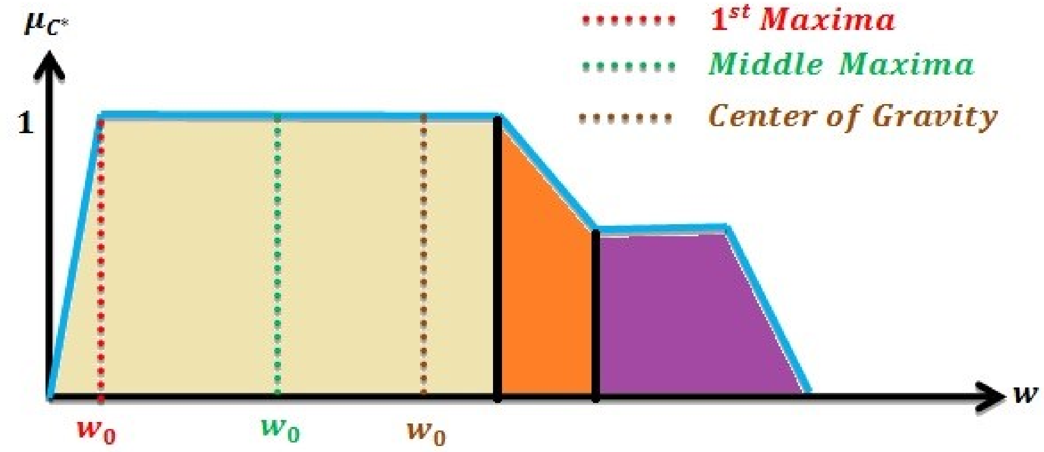


Figure 3. 26 Defuzzification methods.

The firing strength of the rules $\boldsymbol{\alpha_1}$ and $\boldsymbol{\alpha_2}$ are calculated as the following:

$$\alpha_1 = \bigvee_u (\mu_{A1}(u) \wedge \mu_{A'}(u)) \qquad 3.25$$

$$\alpha_2 = \bigvee_u (\mu_{A2}(u) \wedge \mu_{A'}(u)) \qquad 3.26$$

For the two rules, each of them has its own output which is given as the following:

$$\mu_{C_1^*}(w) = \big(\alpha_1 \wedge \mu_{C1}(w)\big)\ \forall w \qquad 3.27$$

$$\mu_{C_2^*}(w) = \big(\alpha_2 \wedge \mu_{C2}(w)\big)\ \forall w \qquad 3.28$$

The overall system is obtained by the **Max** operator as the following:

$$\mu_{C^*}(w) = \mu_{C_1^*}(w) \vee \mu_{C_2^*}(w) \qquad 3.29$$

Additionally, in Mamdani approach and in case the inference has two rules and singleton observation as it is illustrating in Fig. 3.27; then the behavior of the fuzzy system might be described by ***max-min*** composition with Mamdani implication as the following:

$$\begin{array}{ll}
\mathbf{r_1}: & A_1 \rightarrow \quad C_1 \\
\mathbf{r_2}: & A_2 \rightarrow \quad C_2 \\
Observation: & u_o \\
\hline
Conclusion: & C^* = combine(C_1^*, C_2^*)
\end{array}$$

The firing strength of the rules for crisp input $\boldsymbol{u_o}$ and the conclusions could be given as:

$$\alpha_1 = \mu_{A1}(u_o) \tag{3.30}$$

$$\alpha_2 = \mu_{A2}(u_o) \tag{3.31}$$

$$\mu_{C_1^*}(w) = \left(\alpha_1 \wedge \mu_{C1}(w)\right) \forall w \tag{3.32}$$

$$\mu_{C_2^*}(w) = \left(\alpha_2 \wedge \mu_{C2}(w)\right) \forall w \tag{3.33}$$

The overall system is obtained by the **Max** operator as the following:

$$\mu_{C^*}(w) = \mu_{C_1^*}(w) \vee \mu_{C_2^*}(w) \tag{3.34}$$

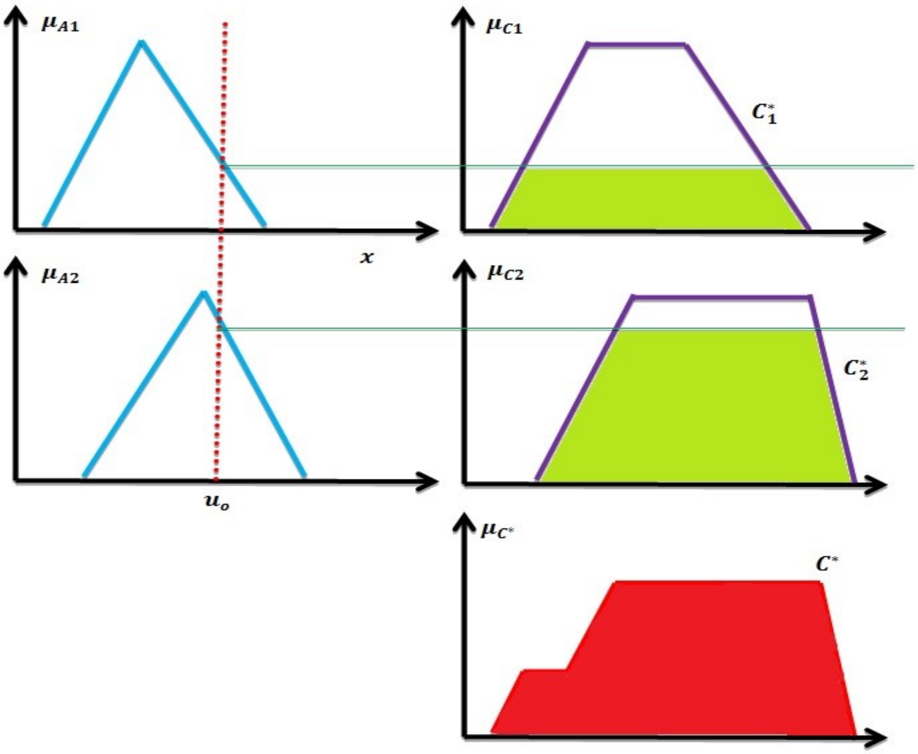


Figure 3. 27 **Fuzzy inference with crisp input using Mamdani approach.**

Finally, if the inference system describes by two rules and two antecedent with crisp inputs $\boldsymbol{u_o}, \boldsymbol{v_o}$ and considers that ***max-min*** compose with Mamdani implication; then the behavior of the fuzzy system can be described as the following:

$$\begin{array}{llllllll}
\mathbf{r_1} & \mathbf{IF} & X\ is\ A_1 & \mathbf{AND} & Y\ is\ B_1 & \mathbf{THEN} & Z\ is\ C_1 \\
\mathbf{r_2} & \mathbf{IF} & X\ is\ A_2 & \mathbf{AND} & Y\ is\ B_2 & \mathbf{THEN} & Z\ is\ C_2 \\
\multicolumn{7}{l}{Observation:\quad u_o, v_o} \\
\hline
\multicolumn{7}{l}{Conclusion: \qquad\qquad C^* = combine(C_1^*, C_2^*)}
\end{array}$$

The firing strength of the rules for crisp input crisp inputs $\boldsymbol{u_o}, \boldsymbol{v_o}$ and the conclusions could be given as:

$$\alpha_1 = \mu_{A1}(u_o) \wedge \mu_{B1}(v_o) \tag{3.35}$$

$$\alpha_2 = \mu_{A2}(u_o) \wedge \mu_{B2}(v_o) \tag{3.36}$$

$$\mu_{C_1^*}(w) = \big(\alpha_1 \wedge \mu_{C1}(w)\big)\ \forall w \tag{3.37}$$

$$\mu_{C_2^*}(w) = \big(\alpha_2 \wedge \mu_{C2}(w)\big)\ \forall w \tag{3.38}$$

The overall system is obtained by the **Max** operator as it is shown in Fig. 3.28, and as the following:

$$\mu_{C^*}(w) = \mu_{C_1^*}(w) \vee \mu_{C_2^*}(w) \tag{3.39}$$

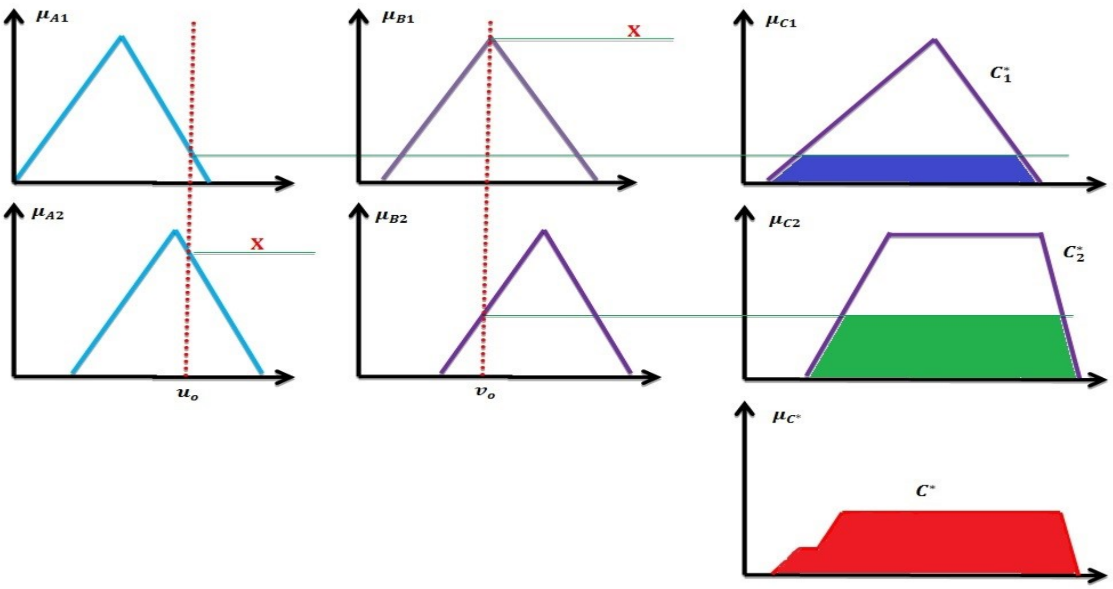


Figure 3. 28 Fuzzy inference with two crisp input using Mamdani approach.

3.4.6. A simple example

After we have introduced the concepts of fuzzy sets, logic and control, we aim to present here how the fuzzy control works in blimp system. In order to illustrate this a fuzzy control for blimp robot will be discussed in details to control the blimp's altitude. The error in the blimp height is the input to the controller whose output is the voltage of the main propellers. The altitude error is the difference between the ***desired height (Z_d)*** and ***current height (Z_c)***. The change in altitude error will be used to indicate if the robot approaches close to the desired height or flying away. The controller output here is the propellers voltage. The second input for this controller is the current vertical velocity. Therefore, Fig. 3.29 illustrates the fuzzy altitude behavior.

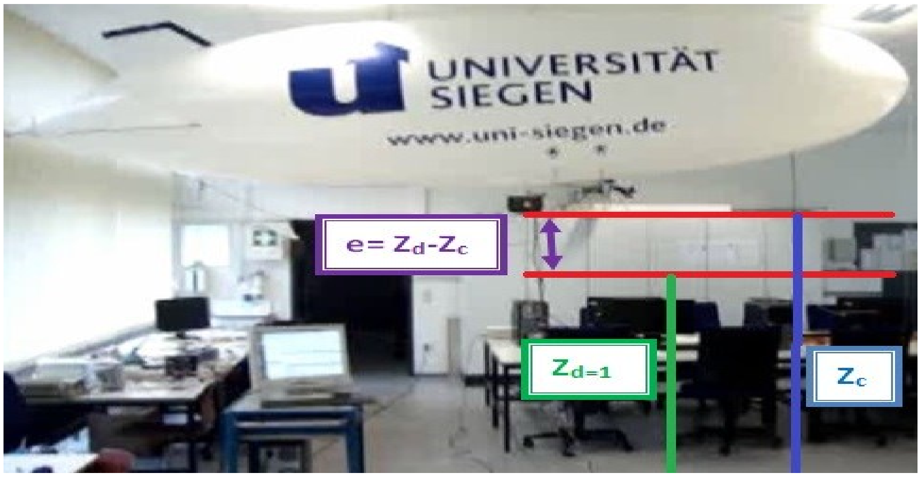


Figure 3. 29 Behavior of the fuzzy altitude control.

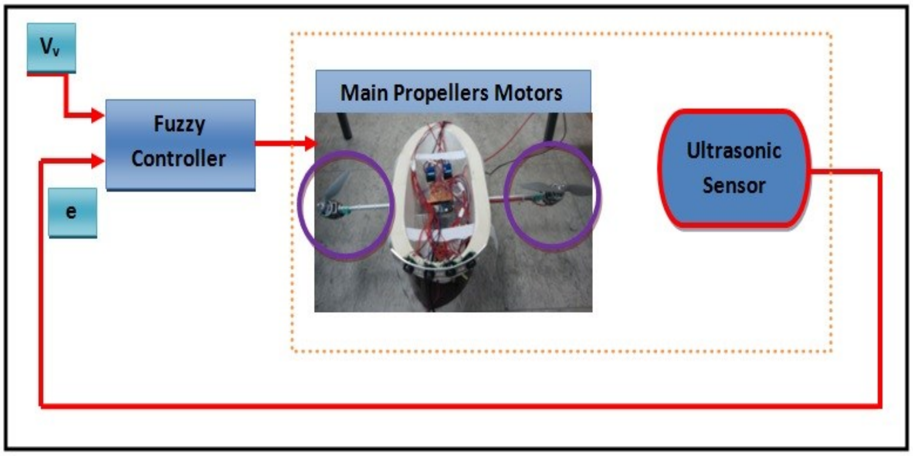


Figure 3. 30 Fuzzy control for height.

The fuzzy control as it is illustrating in Fig. 3.30 will treat the altitude as overlapping ranges. The fuzzy knowledge based rules ***IF-THEN*** have been used to calculate and then to command the microcontroller how much voltage (pulse width module) it needs to increase or decrease the motor speed. As a result, the voltage of the motors will be adjusted continuously to keep the blimp altitude at the desired value. The rules and the membership functions could be identified by the expert, simulations studies and empirical data. The membership functions that are identifying the two inputs values and the output values is shown in Fig. 3.31.

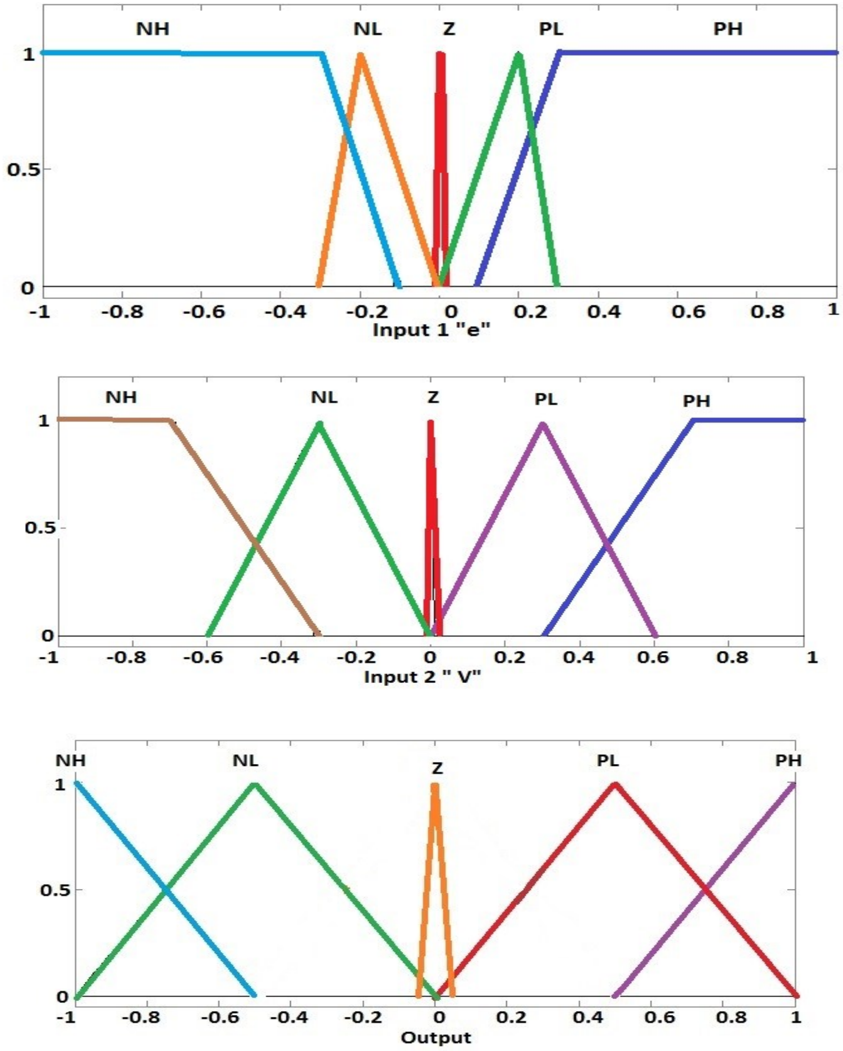


Figure 3. 31 Membership Functions.

		e				
		NH	**NL**	**Z**	**PL**	**PH**
V_v	**NH**	*PH*	*PH*	*Z*	*Z*	*NL*
	NL	*PH*	*PL*	*Z*	*Z*	*NL*
	Z	*PH*	*PL*	*Z*	*Z*	*NH*
	PL	*PL*	*Z*	*Z*	*NL*	*NH*
	PH	*Z*	*Z*	*Z*	*NH*	*NH*

Table 3. 1 Fuzzy rules.

The fuzzy knowledge-rule-base for the altitude for this example is shown in Table 3.1 and it has ***25*** rules. For simplicity, we will define in the following five simple rules:

IF {e is NH} AND {Vv is NH} THEN {Voltage of main propellers is PH}: $\boldsymbol{A_1 \& B_1 \rightarrow C_1}$.

IF {e is NL} AND {Vv is NL} THEN {Voltage of main propellers is PL}: $\boldsymbol{A_2 \& B_2 \rightarrow C_2}$.

IF {e is Z} AND {Vv is Z} THEN {Voltage of main propellers is Z}: $\boldsymbol{A_3 \& B_3 \rightarrow C_3}$.

IF {e is PL} AND {Vv is PL} THEN {Voltage of main propellers is NL}: $\boldsymbol{A_4 \& B_4 \rightarrow C_4}$.

IF {e is PH} AND {Vv is PH} THEN {Voltage of main propellers is NH}: $\boldsymbol{A_5 \& B_5 \rightarrow C_5}$.

As it was described previously, the fuzzy control is similar to the classical control. It starts by taking an input value; then performs some calculations in order to generate the output value. The first step is the fuzzification stage; thus the crisp input will be translated to fuzzy value. Let us take the error in the altitude is ***0.2***. The fuzzy crisp are ***NL*** and ***NH*** with ***0.7*** membership value for each. The second input is ***0.8***; then the fuzzy crisp and its membership value are ***PH*** with ***1***. The second stage is the Inference in which the entire set of rules is evaluated and fired to compute the two antecedent rule conclusion as the following:

$$\begin{array}{rl} Rule: & (A \cap B) \rightarrow C \\ Observation: & A', B' \\ \hline Conclusion: & C^{*} \end{array}$$

We have to notice here that ***A***$'$ and ***B***$'$ are not necessarily to be the same as ***A*** and ***B***.

In membership domain $\boldsymbol{C^{*}} = (\boldsymbol{A'} \cap \boldsymbol{B'})\boldsymbol{o}\big((\boldsymbol{A} \cap \boldsymbol{B}) \rightarrow \boldsymbol{C}\big)$ and the ***GMP*** translates into which known as ***Max-min*** composition with ***"min"*** as implication function ***I***, or more precisely as Mamdani type fuzzy inference system:

$$\mu_{C^*}(w) = \bigvee_{u\in U}\bigvee_{v\in V} C(T(\mu_{A'}(u),\mu_{B'}(v)), I(T(\mu_A(u),\mu_B(v)),\mu_C(w))) \qquad 3.39$$

The last stage is the defuzzification which combines the crisp values and the membership degrees together in a method such as **Center of Gravity** to produce the crisp value **-0.811** volts for the motor speed controller as it is shown in Fig. 3.32.

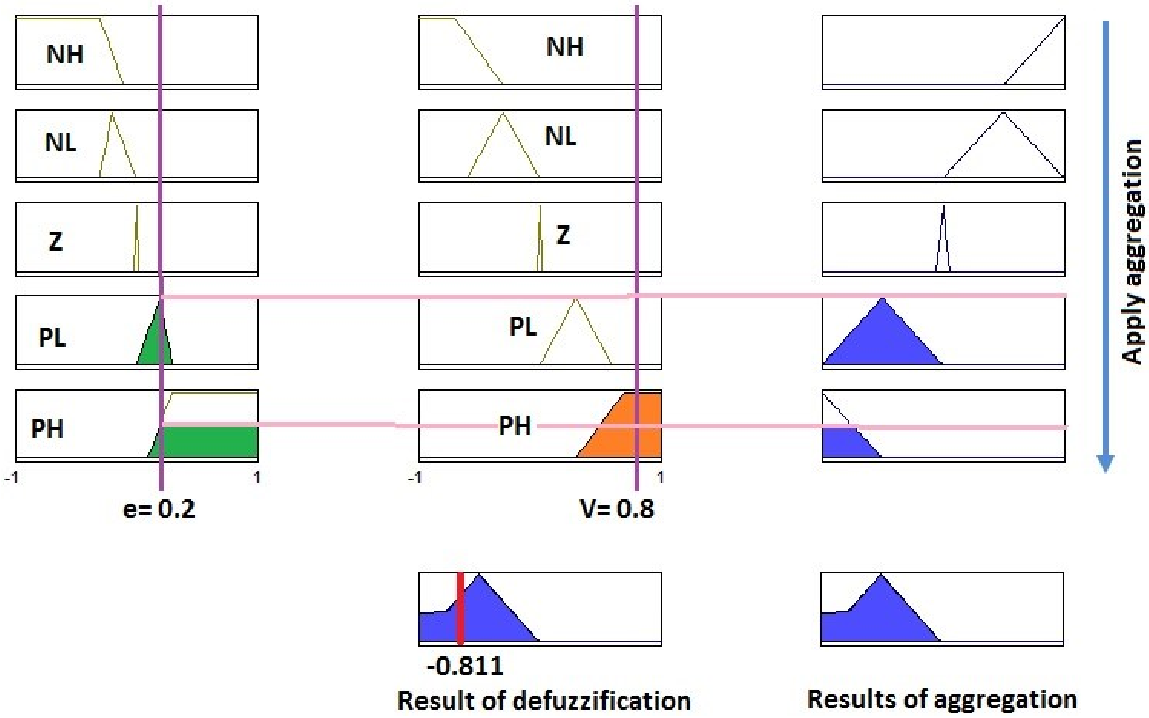


Figure 3. 32 The fuzzy inference process.

3.4.2. Fuzzy Measures and Evidence Theory:

The fuzzy measure theory takes into account several special parameters: ***the plausibility of measures***, ***the belief in these measures***, ***fuzzy membership functions***, and ***classical probability measures***. In such theory, even though the conditions are precise, but the insufficient information about an element alone makes it difficult to determine which special measure should be used. Thus, Dempster–Shafer introduced a theory which allows a combination from several sources to reach a degree of belief by taking all the available evidence [129, 130]. For instance, given a universal set **Ω** and set of all its crisp subsets or power set **P(Ω)**, then the fuzzy measure **f(A)** , which indicates the certainty degree that an element of **Ω** belongs to a certain crisp set **A**, is given by:

$$\mathrm{f(A)}: \mathrm{P}(\Omega) \rightarrow [0,1], \textit{where } \mathrm{f}(\phi) = 0 \ \& \ \mathrm{f}(\Omega) = 1 \qquad 3.40$$

In addition, the evidence theory based on two dual non additive measures: *belief measures* **(Bel)** and *plausibility measures* **(Pl).** Given a finite universal set **Ω** and a subset **A** ∈ **P(Ω)** then:

$$\mathrm{Bel}: \mathrm{P}(\Omega) \rightarrow [0,1] \qquad 3.41$$

$$\mathrm{Bel}(\phi) = 0 \qquad 3.42$$

$$\mathrm{Bel}(\Omega) = 1 \qquad 3.43$$

$$\mathrm{Bel(A)} + \mathrm{Bel}(\bar{\mathrm{A}}) \leq 1 \qquad 3.44$$

$$\mathrm{Pl(A)} = 1 - \mathrm{Bel}(\bar{\mathrm{A}}) \qquad 3.45$$

$$\mathrm{Bel(A)} = 1 - \mathrm{Pl}(\bar{\mathrm{A}}) \qquad 3.46$$

Therefore, these *Belief* and *Plausibility* measures can conveniently be defined by a basic probability assignment function $\mathrm{m}: \mathrm{P}(\Omega) \rightarrow [0,1]$.

$$\sum_{\mathrm{A} \in \mathrm{P}(\Omega)} \mathrm{m(A)} = 1, \mathrm{m}(\phi) = 0 \qquad 3.47$$

The basic probability assignment is not required that $\mathrm{m}(\Omega) = 1$ and there is no required relationship between $\mathrm{m(A)}$ and $\mathrm{m}(\bar{\mathrm{A}})$ [129]. However, the *belief* and *plausibility* measures are uniquely determined for all set $\mathrm{A} \in \mathrm{P}(\Omega)$ by the following formulas [129, 130]:

$$\mathrm{Bel(A)} = \sum_{\mathrm{B|B \subseteq A}} \mathrm{m(B)} \qquad 3.48$$

$$\mathrm{Pl(A)} = \sum_{\mathrm{B|A \cap B \neq \phi}} \mathrm{m(B)} \qquad 3.49$$

$$\mathrm{Pl(A)} \geq \mathrm{Bel(A)} \qquad 3.50$$

where $\mathrm{m}(\mathrm{B})$ is the evidence degree or the belief that there is an element belongs to the set **A** alone, and $\mathrm{Bel}(\mathrm{A})$ is the belief that an element belongs to **A** and also to the various special subsets **A**. However, the $\mathrm{Pl}(\mathrm{A})$ is the belief that an element belongs to set **A** or to any various special subsets **A.** Also, it is an extra evidence related to sets that overlap with **A**. Moreover, every set of $\mathrm{A} \in \mathrm{P}(\Omega)$ in which $\mathrm{m}(\mathrm{A}) \geq 0$ is usually called a focal element $\boldsymbol{\mathcal{F}}$ on which the available evidence is focused. The total ignorance is expressed in terms of the basic assignment by $\mathrm{m}(\Omega) = 1$ and $\mathrm{m}(\mathrm{A}) = 0$ for all $\mathrm{A} \neq \Omega$. Thus, although we know that the element is in the universal set, but we have no evidence about its location in any subset of $\boldsymbol{\Omega}$.

3.4.3. Possibilities Theory:

Recently, the fuzzy sets theory has been developed in several directions and applications including the linguistics, logic, pattern recognition, decision-making as well as the system theory. In fact, the common relation between all of these applications depends on the fuzzy restrictions in which the fuzzy relation plays as an elastic constraint on the values that might be assigned to a variable [131]. On the other hand, the possibility theory, which is a mathematical notion and an alternative to probability theory, can deal with the uncertainties in any system and handle of incomplete information. Even though it has much similarity to probability theory since it is based on set-functions, but still the difference between them is that the possibility theory presented to use possibility and necessity measures [132]. For example, the expression ***It is not probable that "I am not student"*** is **parallel** to say ***It is probable that I am student***, whilst the statement ***It is not possible that "I am not student"*** is **not parallel** to say ***It is possible that I am student***. Indeed, it has a stronger and more powerful meaning, namely: ***It is necessary that I am student***. Therefore, the theory is not only strongly linked to fuzzy systems in its mathematics, but also in its semantics [52] and it describes the degrees of possibility as numbers which generally stand for upper probability bounds. These possibility degrees were introduced in terms of ease of attainment, uncertainty notion and flexible constraints [133, 134]. Therefore, by developing frequents view of possibility and a bridge between possibility theory and statistical science, then, possibility degrees can offer a simple approach to imprecise these statistics, in terms of upper bounds of frequency [135-142].

For simplicity, assume there is a set **U**, and then the possibility distribution is the characteristic function of a subset **E** of **U**. It will model the situation in which all is known about **x** is that it cannot be exist outside **E**, thus, **(x(u) = 1 if x є E or x(u) = 0 otherwise)**. The conclusion here is that this type of possibility distribution stats that **x** is in between values **a** and **b**, and then **E** = **[a, b]**. The possibility distribution is capable of modeling several kinds of imprecise and vague information. Therefore, it can describe the imprecise information by several intervals with degree of confidence or many levels of confidence. The possibility distribution can generally represent a finite family of confidence subsets ***{ A₁, A₂, ..., Aₘ}*** in which each confidence subset **A** is attached a positive confidence level $\boldsymbol{\lambda_i}$, then, the possibilities distribution $\boldsymbol{\pi_x(x)}$ is given by:

$$\pi_x(x) = \begin{cases} 1, & \text{if } u \in A_1 \\ \min(1 - \lambda_i), & \text{otherwise} \end{cases} \qquad 3.51$$

From a mathematical viewpoint, the information which is modeled by $\boldsymbol{\pi_x(x)}$ can be nested as random set, and thus; $\boldsymbol{\pi_x(x)}$ is the function of a random set which might be described as belief function and viewed as random interval ***I*** where $\boldsymbol{\pi_x(x) = P(u \in I)}$ [143-146]. In fact, the importance of such theory appeared clearly due to the fact that there are some decisions based on possibilistic rather than probabilistic. Therefore, the fuzzy restrictions could be explained and designed as a possibility distribution **if and only if** the fuzzy membership functions play such role in possibility distribution histograms. Thus, fuzzy variables link with the possibility distribution in the same way as a random variable do with a probability distribution. Hence, the possibilities histograms might be transferred to fuzzy logic membership functions without any change since both of them have the same mathematical descriptions. As the possibilities theory deals only with evidence in where the focal elements are overlapping, it is always better to collect the possibilities data empirically, and then analyze them in order to design the possibilities histograms. The use of the interval statistics sets with their empirical random sets could be lead to develop an approach that might be used to construct the possibility distribution histogram [147]. The most important concepts in this approach are the possibility **(Pos)** which is a special belief measures and the necessity **(Nec)** which is a special plausibility measures. For all subsets $\mathbf{A}$, $\mathbf{B} \in \mathrm{P}(\Omega)$, then [147]:

$$\mathrm{Nec}(A) = 1 - \mathrm{Pos}(\overline{A}) \qquad 3.52$$

$$\mathrm{Nec}(A) + \mathrm{Nec}(\overline{A}) \leq 1 \qquad 3.53$$

$$\mathrm{Pos}(A) + \mathrm{Pos}(\overline{A}) \geq 1 \qquad 3.54$$

$$\mathrm{Pos}(A) \geq \mathrm{Nec}(A) \qquad 3.55$$

$$\mathrm{Nec}(A \cap B) = \min[\mathrm{Nec}(A), \mathrm{Nec}(B)] \qquad 3.56$$

$$\mathrm{Pos}(A \cup B) = \max[\mathrm{Pos}(A), \mathrm{Pos}(B)] \qquad 3.57$$

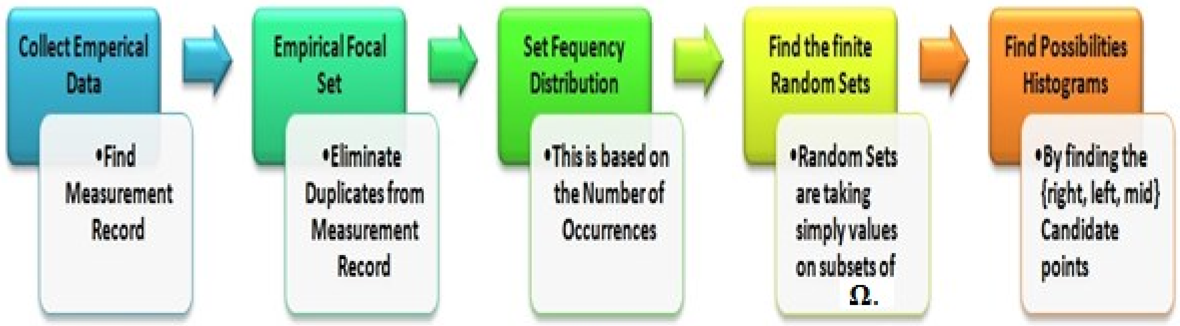
Furthermore, from (3.30 and 3.31) as a special case:

$$\min[\mathrm{Nec}(A), \mathrm{Nec}(\overline{A})] = 0 \qquad 3.58$$

$$\max[\mathrm{Pos}(A), \mathrm{Pos}(\overline{A})] = 1 \qquad 3.59$$

3.4.4. Possibility Approach

In such procedure, the empirical measurement needs a consistency data with semantic aspects of possibility theory and the concept of set statistics requires non-disjoints data. Thus, the frequency data generated from the empirical observations will lead to design the possibilities histograms which will be transferred to fuzzy membership functions. The general steps in order to design the possibilities histograms are shown in Fig. 3.33. Therefore, there are some important definitions to model the fuzzy sets based on possibility distributions theory [147-150].



Collect Emperical Data
- Find Measurement Record

Empirical Focal Set
- Eliminate Duplicates from Measurement Record

Set Fequency Distribution
- This is based on the Number of Occurrences

Find the finite Random Sets
- Random Sets are taking simply values on subsets of Ω.

Find Possibilities Histograms
- By finding the {right, left, mid} Candidate points

Figure 3. 33 Overview process to design possibilities histograms.

Definition 1 [147-150]***:*** In order to derive a possibility distribution from an empirical data, it is indispensable to demonstrate the general measuring record subsets $\mathbf{U_s} \subseteq \mathbf{\Omega}$ as defined below:

$$\langle U_s \rangle = \langle U_1, U_2, \dots \dots ., U_i \rangle, where\ 1 \le s \le i \qquad 3.60$$

These subsets will be collected for several times and recorded. Where ***s*** is counter and ***i*** is the number of the element subsets.

Definition 2 [147-150]***:*** Because the aforementioned data are empirical data and they are unknown, we could not know what and/or how they should be, it is always necessary to eliminate the duplicates from this general measurement record $\mathbf{U_s}$, then derived the Empirical Focal Set $\boldsymbol{\mathcal{F}}^{\mathbf{E}}$ which is given by:

$$\mathcal{F}^E = \{U_1, U_2, \dots \dots ., U_j\} \qquad 3.61$$

$where\ 1 \le s \le j, j \le i, \forall U_s \in \mathcal{F}^E.$

Definition 3 [147-150]***:*** As many empirical data there are a weight value or number of occurrences of $\mathbf{U_s}$ in the general measurement record, this is given by the following:

$$C_s = C(U_s) \qquad 3.62$$

Definition 4 [147-150]***:*** The set frequency distributions for the data are a function $\mathbf{m^E}: \boldsymbol{\mathcal{F}}^{\mathbf{E}} \longrightarrow [\mathbf{0}, \mathbf{1}]$ which means the number of occurrence for the subset over the total number of the occurrences and it is given by:

$$m^E(U_s) = \frac{C_s}{\sum_{U_s \in \mathcal{F}^E} C_s} \qquad 3.63$$

Definition 5 [147-150]***:*** It is known that the mathematical form of the random sets is very complicated. However, in this case of study, we assume them as a finite set, then they will take simply values on subsets of . As a result, the finite random set **S** is given by:

$$S = \{\langle U_s, \zeta_s \rangle : \zeta_s > 0\} \quad 3.64$$

Where $\boldsymbol{\zeta}_{\mathbf{s}}$ is the evidence function $\zeta_s = \zeta(U_s)$.

Definition 6 [147-150]***:*** The possibility distribution can be described as possibility histogram which can be obtained as follows:

$$\pi(\omega) = \sum_{U_s \ni \omega} m_s^E = \frac{\sum_{U_s \ni \omega} C_s}{M} \quad 3.65$$

Where M is the number of the subsets.

Definition 7 [147-150]***:*** By assuming each observed subset $\mathbf{U_s} \in \boldsymbol{\mathcal{F}}^{\mathbf{E}}$ is closed interval and symbolized by its statistics left endpoints $\mathbf{l_s}$ and right endpoints $\mathbf{r_s}$. Thus, the $\mathbf{U_s} = [\mathbf{l_s}, \mathbf{r_s}]$ and the vectors of endpoints are given by:

$$\overrightarrow{E^l} = \langle l_1, l_2, \dots, l_N \rangle \quad 3.66$$

$$\overrightarrow{E^r} = \langle r_1, r_2, \dots, r_N \rangle \quad 3.67$$

$$\vec{E} = \langle l_1, l_2, \dots, l_N, r_N, r_{N-1}, \dots, r_1 \rangle \quad 3.68$$

The linear order for the vectors of endpoints $\vec{E}$ is

$$l_1 \leq l_2 \leq \cdots \leq l_N \leq r_N \leq r_{N-1} \leq \cdots \leq r_1 \quad 3.69$$

Definition 8 [147-150]***:*** Assume a random interval $\mathbf{U_s} \in \boldsymbol{\mathcal{F}}(\mathbf{U})$ and $\mathbf{E^l} \coloneqq \{\mathbf{e^l_{k^l}}\}$,$\mathbf{E^r} \coloneqq \{\mathbf{e^r_{k^r}}\}$, $\mathbf{E} \coloneqq \{\mathbf{e_k}\}$ are the sets of left endpoints, right endpoints, and total endpoints $\overrightarrow{\mathbf{E^l}}$**,** $\overrightarrow{\mathbf{E^r}}$ **and** $\vec{\mathbf{E}}$ respectively, where $\forall \mathbf{e^l_{k^l}} \in \mathbf{E^l}$**,** $\forall \mathbf{e^r_{k^r}} \in \mathbf{E^r}$**,** $\forall \mathbf{e_k} \in \mathbf{E}$**,** $\mathbf{1 \leq k^l \leq Q^l \coloneqq |E^l|}$**,** $\mathbf{1 \leq k^r \leq Q^r \coloneqq |E^r|}$**,** $\mathbf{1 \leq k \leq Q \coloneqq |E|}$**,** $\mathbf{G_k} = \{\mathbf{e_k}, \mathbf{e_{k+1}}\}$ for $\mathbf{1 \leq k \leq Q - 1}$, Thus, $\mathbf{E^l \cup E^r = E}$ **and** $\mathbf{Q^l + Q^r \geq Q}$**.** Where ***k*** is the counter of ***E*** onto ***Q. Q*** is the number of elements***. Note*** when we use ***r*** or ***l*** it means for left or right side***.***

Definition 9 [147-150]***:*** Even though the possibilistic histograms are similar to ordinary stochastic histograms, but they are really performed from overlapping interval observations; thus they are ruled by the mathematics of random sets. If **π** is a possibilistic histogram, which means fuzzy intervals, then definitely the form of this possibilities histogram relies on both: the support and the core of the measurement record sets. Moreover, $\mathbf{C}(\boldsymbol{\pi}) = \left[\mathbf{e}^{\mathbf{l}}_{\mathbf{Q}^{\mathbf{l}}}, \mathbf{e}^{\mathbf{r}}_{\mathbf{q}^{\mathbf{r}}}\right]$ is the core of the possibility distribution and $\mathbf{supp}(\boldsymbol{\pi}) = \left[\mathbf{e}^{\mathbf{l}}_{\mathbf{l}}, \mathbf{e}^{\mathbf{r}}_{\mathbf{l}}\right] = \cup_{\mathbf{k=1}}^{\mathbf{Q-l}} \mathbf{G}_{\mathbf{k}}$ is the support of the possibility distribution and the **π** is given by:

$$\pi\left(\left[-\infty, \mathrm{e}^{\mathrm{l}}_{\mathrm{l}}\right)\right) = \pi\left(\left(\mathrm{e}^{\mathrm{r}}_{\mathrm{l}}, -\infty\right]\right) = 0 \qquad 3.70$$

Definitions 10 [147-150]***:*** Although the formers of possibilistic histograms are collections of closed segments $\boldsymbol{T_k}$ of different length, but they are not discrete points in stochastic histograms. Thus, instead of the normal interpolation, it is better to represent set of candidate point from the segments and then a continuous curve **π** will fit them. Assume the possibilistic histogram as:

$$\pi = \{T_k\} = \{[e_k, \pi(e_k)]\} \qquad 3.71$$

The right ***r*** and left ***l*** endpoints of each of the $\mathbf{T_k}$ given by

$$t_k^l \coloneqq \langle e_k, \pi(e_k)\rangle \qquad 3.72$$

$$t_k^r \coloneqq \langle e_{k+1}, \pi(e_k)\rangle \qquad 3.73$$

Moreover, the following formula is using for the midpoints of each of the $\mathbf{T_k}$:

$$h_k \coloneqq \langle \frac{e_k + e_{k+1}}{2}, \pi(e_k)\rangle \qquad 3.74$$

The midpoints of the core:

$$c \coloneqq \langle \frac{l_N + r_N}{2}, 1\rangle \qquad 3.75$$

The endpoints of the support are $l \coloneqq \langle l_1, 0\rangle$ and $r \coloneqq \langle r_1, 0\rangle$.

The set of all interval mid and end points to which a continuous curve may be fitted is $\boldsymbol{K}' \coloneqq \left\{\boldsymbol{t}_k^l, \boldsymbol{t}_k^r, \boldsymbol{h}_k\right\}$. In addition, the set of all interval mid and end points to which a continuous curve actually will be fitted is any subset $\boldsymbol{K} \in \boldsymbol{K}'$. Finally, the set of all the points to which the curve will be fitted is $\{\boldsymbol{c}, \boldsymbol{l}, \boldsymbol{r}\} \cup \boldsymbol{K} \subseteq \boldsymbol{\pi}$. More details could be found in Appendix D.

For example let us consider that we have done an experiment and collect the observed data. These data categorized into two main intervals ***A₁ = {a, b}*** and ***A₂= {c, d}.*** If each intervals observed once that means ***m(A₁)=m(A₂)=0.5***. The interval $\boldsymbol{K}' = \left\{\boldsymbol{s_1}, \boldsymbol{t_1^l}, \boldsymbol{t_2^r}, \boldsymbol{t_3^r}, \boldsymbol{s_2}\right\}$. There are three ways to find the triangular histograms parameters are shown in Fig. 3.34. First, $\boldsymbol{H_1} = \{\boldsymbol{s_1}, \boldsymbol{c_2}, \boldsymbol{s_2}\} \cup \left\{\boldsymbol{s_1}, \boldsymbol{t_1^l}, \boldsymbol{t_2^l}, \boldsymbol{t_2^r}, \boldsymbol{t_3^r}, \boldsymbol{s_2}\right\}$ that is the gray line. Second, $\boldsymbol{H_2} = \{\boldsymbol{s_1}, \boldsymbol{c_2}, \boldsymbol{s_2}\} \cup \{\boldsymbol{s_1}, \boldsymbol{t_1^r}, \boldsymbol{c_2}, \boldsymbol{s_2}\}$ which is the yellow line. Finally, $\boldsymbol{H_3} = \{\boldsymbol{s_1}, \boldsymbol{c_2}, \boldsymbol{s_2}\}$ that is the green color. We should note that the possibilities histograms are similar to the fuzzy membership functions. Both of them have the core, support and the outputs take values between 0 and 1. In addition, both of them have the same mathematical descriptions, and then, someone can convert the histograms to fuzzy membership functions without any changes as it is shown in Fig. 3.35.

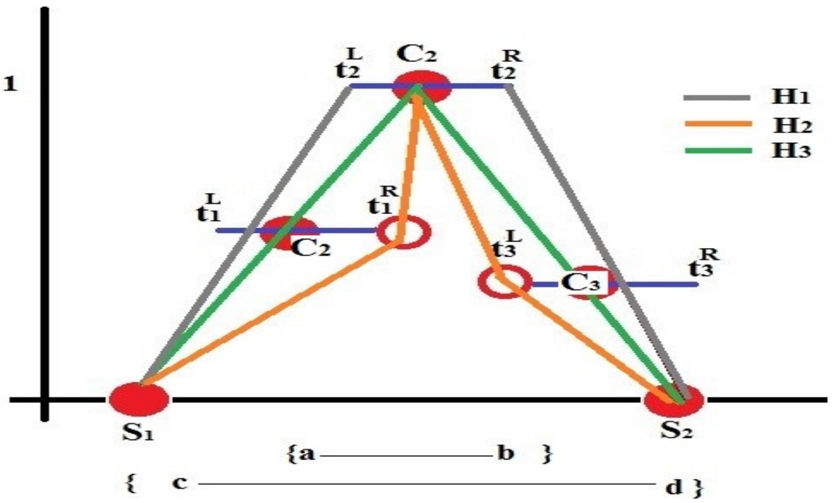
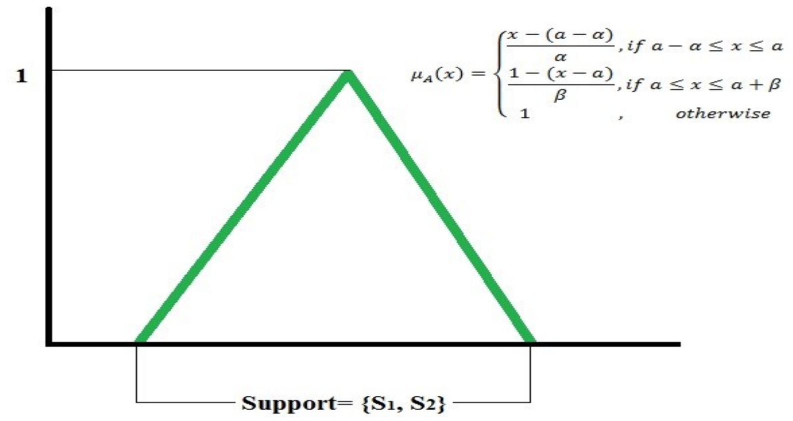


Figure 3. 34 **Triangular possibilities histograms.**



$$\mu_A(x) = \begin{cases} \frac{x-(a-\alpha)}{\alpha}, if\ a-\alpha \le x \le a \\ \frac{1-(x-a)}{\beta}, if\ a \le x \le a+\beta \\ 1 \quad , \quad otherwise \end{cases}$$

Figure 3. 35 **Triangular fuzzy membership functions.**

Moreover, in case of designing the trapezoidal histograms, which is better than triangular and can be optimized by using bacterial algorithm, the empirical data should be grouped into four main categories ***{S1, S_2, S_3, S_4}.*** Then, the interval $\boldsymbol{K'} = \{\boldsymbol{s_1}, \boldsymbol{t_1^l}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{t_2^r}, \boldsymbol{s_2}\}$. There are three ways also to find the triangular histograms parameters are shown in Fig. 3.36. $\boldsymbol{H_1} = \{\boldsymbol{s_1}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{s_2}\} \cup \{\boldsymbol{s_1}, \boldsymbol{t_1^r}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{t_2^l}, \boldsymbol{s_2}\}$ $\boldsymbol{H_2} = \{\boldsymbol{s_1}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{s_2}\} \cup \{\boldsymbol{s_1}, \boldsymbol{t_1^l}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{c_2}, \boldsymbol{s_2}\}$, $\boldsymbol{H_3} = \{\boldsymbol{s_1}, \boldsymbol{t_1^l}, \boldsymbol{s_3}, \boldsymbol{s_4}, \boldsymbol{t_2^r}, \boldsymbol{s_2}\}$. The final fuzzy membership function is illustrating in Fig. 3.37.

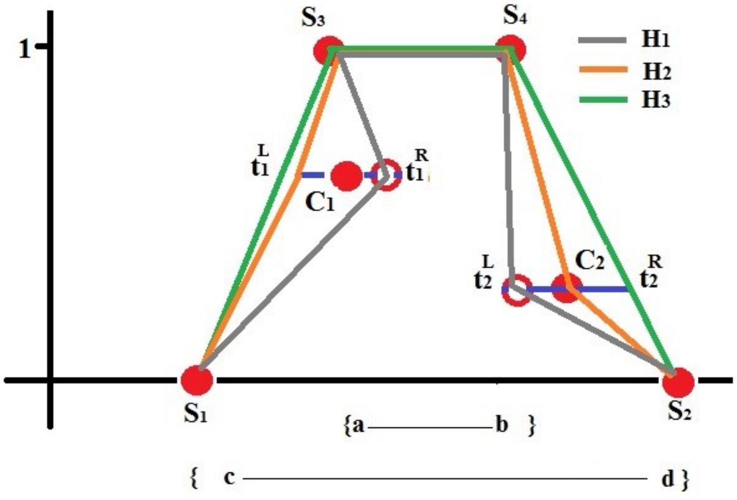


Figure 3. 36 **Trapezoidal possibilities histograms.**

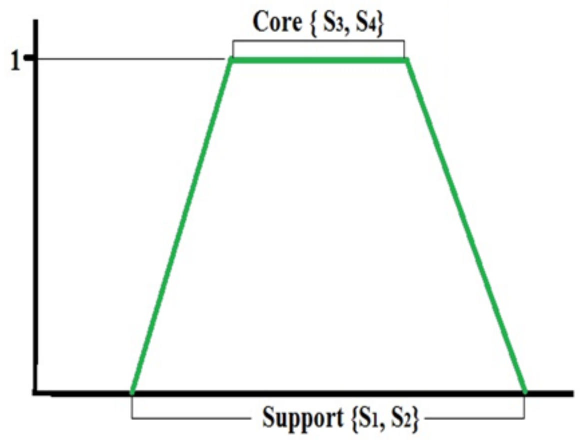


Figure 3. 37 **Trapezoidal fuzzy membership functions.**

As it was mentioned previously that the membership functions are trapezoidal and could be designed by possibilities histograms. These kinds of membership functions are general enough and widely used besides they need less computational time compare to others functions like Gaussian membership functions. The bacterial algorithm [151] could be implemented in order to optimize these membership functions and to extract the rules based knowledge. This algorithm capable of changing and tuning more than one membership functions simultaneously. In addition, it can eliminate ineffective rules. It can tune the breakpoints of the membership functions as the following [151]:

$$\begin{gathered} m_{ij} = 2d_{ij} - 2a_{ij} + c_{ij} - b_{ij} \\ n_{ij} = c_{ij}^2 + 2c_{ij}d_{ij} + 2d_{ij}^2 - 2a_{ij}^2 - 2a_{ij}b_{ij} - b_{ij}^2 \\ F_{ij} = \left(n_{ij} - m_{ij}^2 - c_{ij}m_{ij}\right)/\left(2m_{ij} + c_{ij} - b_{ij}\right) \\ H_{ij} = F_{ij} + m_{ij}3 \end{gathered} \qquad 3.76$$

Where $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ are the breakpoints of the trapezoidal membership functions and F_{ij} and H_{ij} are the new breakpoints for left and right sides for **i**$^{\mathbf{th}}$ rule and **j**$^{\mathbf{th}}$ input variables. An example for bacterial algorithm to optimize the trapezoidal membership functions is illustrating in Fig. 3.38a and fig.3.38b. As it is shown in Fig. b that breakpoints of the membership functions could be shifted to right or to the left depend on the **δ** which is the shifted values for both left and right side and it describes as: **δ_L=F-a, δ_R=H-d.**

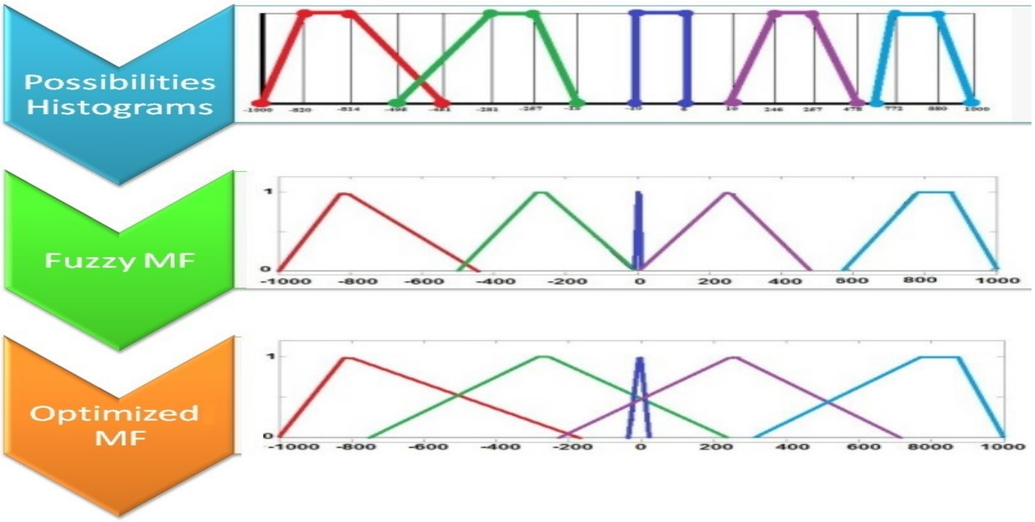


Figure 3. 38a Optimized the trapezoidal fuzzy membership functions.

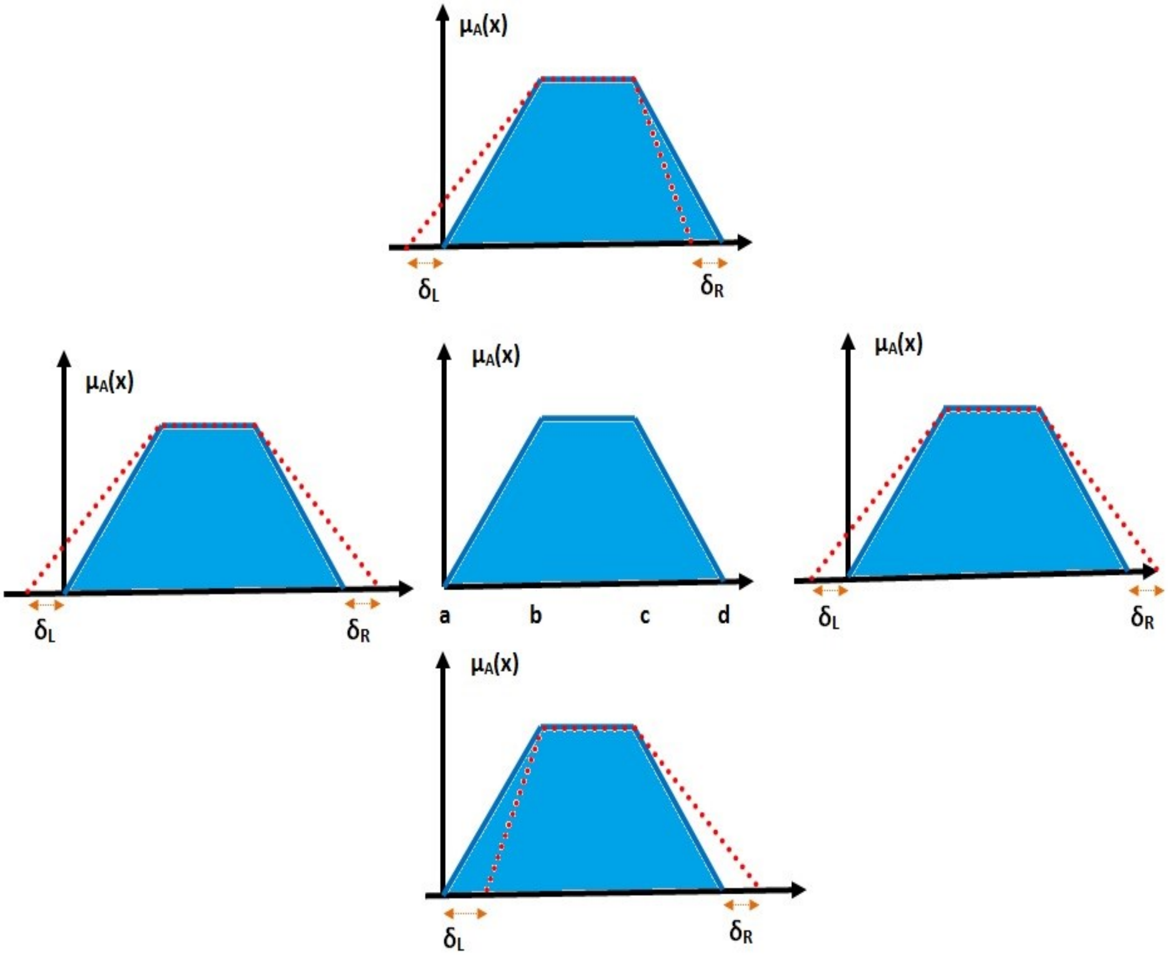


Figure 3.38b Optimized the trapezoidal fuzzy membership functions.

Regarding to Eq. 3.7 which describes the trapezoidal membership functions for **j**th input variables and **i**th rules, if we assume that the fuzzy controller has two inputs and one output with number of rules **m**, then the encoded chromosome for the fuzzy rules is shown in Fig. 3.39. The rule here is : **IF *x* is μ_{Aji} AND *y* is μ_{Aji} THEN *z* is μ_{Aj}**. In this example the rule was R_2 (j = 2) and (i= 1, 2 are input linguistic variables). Hence, the R_2 is : **IF x is μ_{A21} AND y is μ_{A22} THEN z is μ_{A2}**. The antecedents are **IF x is μ_{A21}** and **y is μ_{A22}** and the consequent is **z is μ_{A2}**.

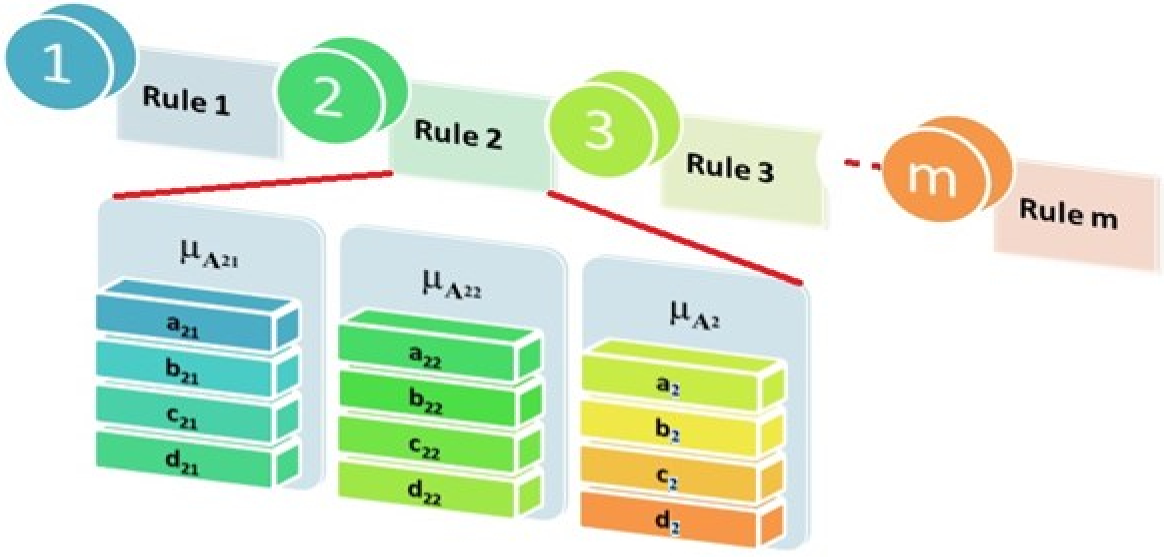


Figure 3. 39 **Fuzzy rules chromosome.**

The algorithm starts by selecting random bacterium. In this work, the encoded chromosome of the membership functions was not randomly initialized as it was describe in [151] because they were already build by possibilities histograms. Then, the mutation process starts by changing the parameters of the chromosome parts. Thus, the mutation process will not only optimize the membership functions, but also the structure of the rule base will be optimized and the ineffective rules will be deleted. Therefore, several conditions should take into account in order to delete the ineffective rules. First, if the membership functions are narrow; then the rule will be deleted. Second, if membership functions belong to same variable; the rule will be deleted for one of them. As a result, the new membership function will be recalculated as it was describe in Eq. 3.76. Details could be found in Appendix D.

3.5. Communications

In this section, the choice for a suitable communication method between the blimp and the ground control station is made. In order to come to this decision, various communication methods are discussed. The Institute of Electrical and Electronics Engineers (IEEE) has designed and developed several standards which are responsible for ensuring compliance with legislation and compatibility between several products manufacturers. The mobile communication methods which are known as the cellular networks have experienced a fast and dramatic growth in both number of users and the extent of services [152, 153]. In fact, the range of 2G and 3G networks is unlimited due to the fact that they consist of countless linked smaller networks. However, these networks practically are not covering the remote areas. In regarding to the data rate, 2G is designed for speech purposes; and thus the data rate is slightly insufficient to support image transmission [154]. Even though the 3G is better suited for such purpose, but the costs are high. Besides that, the commercial satellite constellations apply a similar approach and the satellite communication is very applicable in remote areas. However, one of the most drawbacks of satellite communication is also the high cost [155]. One of the most widely standards introduced by IEEE is known as the Wireless Local Area Network (WLAN). WLAN is often used to create wireless networks which are suitable and applicable for robots researches. Despite the high data rates, the WLAN range is limited to 100m [156, 157]. In addition, the Bluetooth standard operates in the unlicensed 2.4 GHz frequency band [158]. It was designed for low power and short range connection between two computers or portable devices [159]. Actually, at this short range, it is capable of offering data rate up to 1 Mbps [160] which is suitable for small ad-hoc networks in swarm applications [161].

Using a WiFi (IEEE 802.11g/b) wireless network system, the GCS will communicate with the blimp robot via the internet protocol TCP/IP. There are two protocol types within TCP/IP, namely the Transmission Control Program (TCP) and the User Datagram Protocol (UDP). TCP does not only guarantee the reliable and in-order delivery of sender to receiver data, but also it distinguishes data for multiple applications that run on the same host. On the other hand, UDP does not provide a guarantee for the reliability and ordering because datagram might arrive out of order. Therefore, UDP is more efficient for time-sensitive purposes [162]. Despite TCP being more robust, but it can be optimized in wired networks. In addition, one of the most common problems in TCP is the lost packet. Because in this case the application can get at the next coming packets if and only if the retransmitted copy of the lost packet is received. This is really a problem in robot application since the robot needs to work at real time [163]. Therefore, the most suitable protocol for this application is UDP.

The hybrid Peer to Peer has been adopted and used as it is illustrating in Fig. 3.40. This is because we have small clients and we have used Wireless-G Router in this project. It was found that the port numbers 6110 and 6111 are not use. Consequently, there are not any conflicts to use them in this kind of application [164, 165]. The choice of the IP addresses was based on Internet Engineering Task Force (IETF) and Address Allocation for Private Internets (AAPI), IP address range 192.168.0.0 to 192.168.255.255 is for private-use within a Local Area Network (LAN))[166].

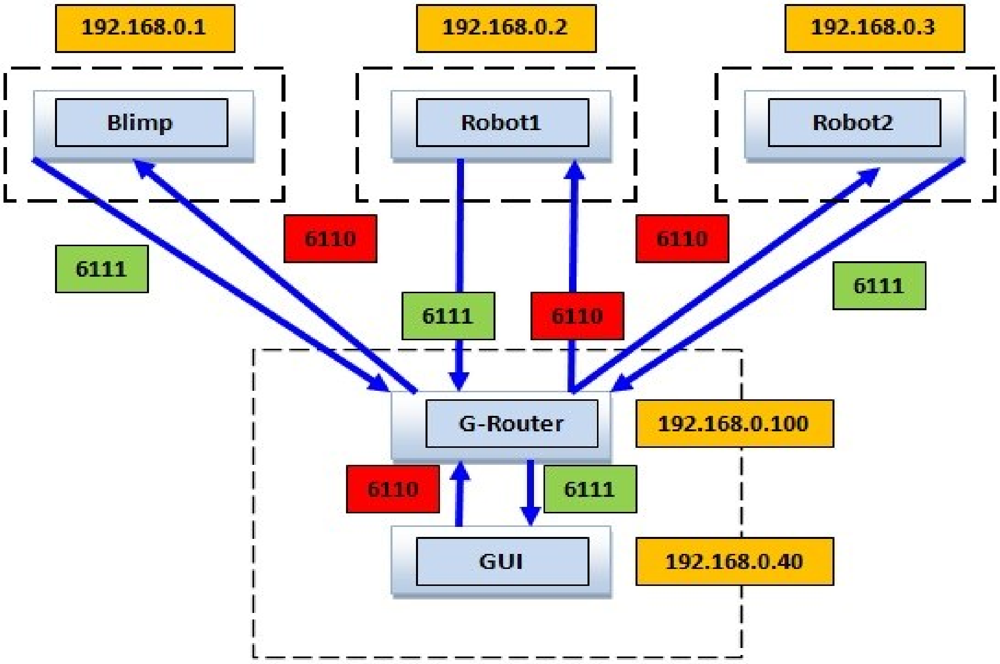


Figure 3. 40 **The hybrid P2P model.**

3.6. Wireless Sensor Networks Tracking

The WSNs consists several number of low-cost and low-power sensors which deployed in an environment monitor this region. Fig. 3.41 shows an example of **WSNs** where sensors are deployed to monitor some region and detect an intruder (ground and/or aerial target) if it passes through this region. Target localization in **WSNs** could be considered as one of the most important aspects in this kind of sensors because they can provide efficient information which enables to track the target. As it was mentioned previously, it is difficult to estimate target altitude with only single sensor, so there is a need to fuse the information from two or more wireless sensor sources. To fuse these data the maximum likelihood estimator combined with particle swarm optimization could be used to estimate the location of the target. Depending on all rows sent by different sensors, the fusion center use maximum likelihood estimator to estimate the location of the target. Moreover, the efficiency of maximum likelihood estimator is compared with the weighted average algorithm and the Cramer-Rao lower Bound (CRLB). In this thesis, we consider a 3D multi target localization problem where targets are flying within the range of the sensor network [167]. On the other hand, sensors are assumed to be ground sensors where they are deployed above the ground surface plane. Moreover, sensors modulate their decisions using OOK technique which has been proposed recently for WSNs because it is a power efficient modulation technique and enables sensors to use censoring scheme (send/no send).

3.6.1. System Model and Problem Formulation

A number of **K** sensors are deployed in 2D region to collect observations and get local decisions. Sensors keep taking observations for **T** time intervals and process it to get a local binary decision about the existence of targets. Then, this decision is transmitted to the fusion center. The fusion center receives all decisions sent from all sensors, and estimates the targets location based on maximum likelihood estimation theory. The observation for a local sensor **k** due to the $\mathbf{i}^{\text{-th}}$ target, $\mathbf{S}_{\mathbf{k},\mathbf{i}}$, depends on the received power emitted by (or reflected from) the target corrupted by additive white Gaussian noise. $\mathbf{S}_{\mathbf{k},\mathbf{i}}$ can be formulated as:

$$S_{k,i} = \left(\sum_{i=1}^{N} a_{k,i}\right) + n_{k,t} \qquad 3.77$$

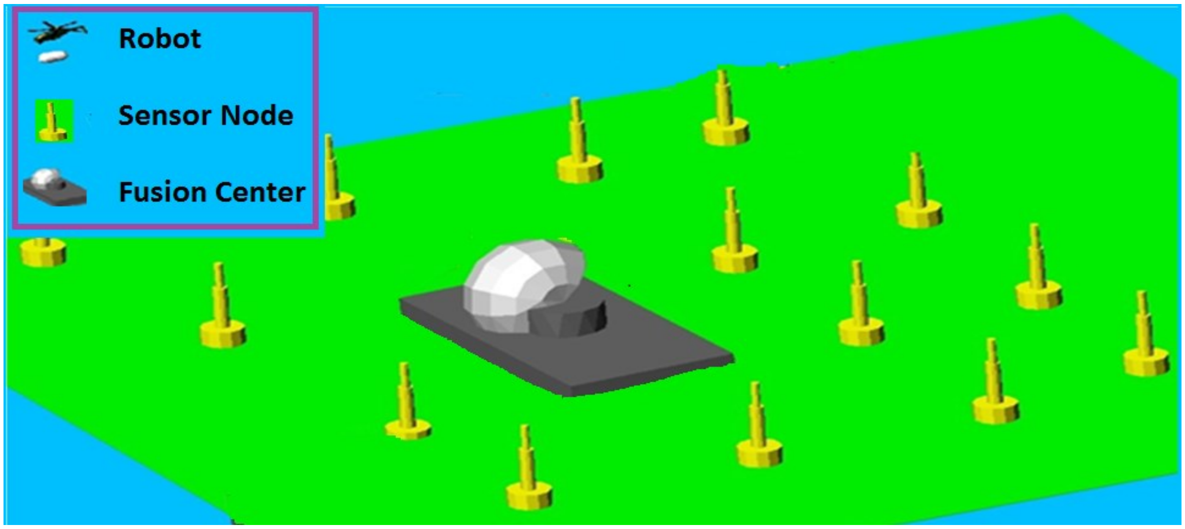


Figure 3. 41 Vehicles cross area under WSNs surveillance.

Where $\boldsymbol{a_{k,i}}$ is a constant depending on the amount of received power at **k**$^{\text{-th}}$ sensor due to **i**$^{\text{-th}}$ target, and $\mathbf{n_{k,t}}$ is the noise at **k**$^{\text{-th}}$ sensor during the **t**$^{\text{-th}}$ time frame, and ***N*** is the total number of targets with in the area and needed to be localized. $\mathbf{n_{k,t}}$ is assumed to be identical independent distribution (i.i.d) with respect to time and sensor and follows standard Gaussian distribution **N (0, $\boldsymbol{\sigma_n^2}$)**. According to the isotropic signal intensity attenuation model, $\mathbf{a_{k,i}}$ can be written as:

$$a_{k,i} = \sqrt{P_0\left(\frac{d_0}{d_{k,i}}\right)^n} \qquad 3.78$$

Where $\mathbf{P_0}$ is the signal power and it can be measured or obtained at a certain distance or reference $\boldsymbol{d_0}$ (we set $\mathrm{d_0 = 1m}$), $\mathrm{d_{k,i}}$ is the distance between the **k**$^{\text{-th}}$ sensor and the **i**$^{\text{-th}}$ target, and n is the path loss exponent which depends on the environment.

$$d_{k,i} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} \qquad 3.79$$

While $(\mathbf{x_i}, \mathbf{y_i}, \mathbf{z_i})$ are the Cartesian coordinate of the **i**$^{\text{-th}}$ target and $(\mathrm{x_k}, \mathrm{y_k}, \mathrm{z_k} = 0)$ are the Cartesian coordinate of the **k**$^{\text{-th}}$ sensor. The sensors are deployed on the ground surface (z_K=0). Sending a decision in each time interval (number of **T** decisions as a total) consumes a lot of power from the sensors. Since the WSN is subject to power constraint, we will assume that the sensor process the **T** observations to get one reliable decision to be sent at the end of the time observation window. Considering this algorithm a considerable amount of power is saved. The target is assumed to be stationary during the **T** time intervals.

3.6.2. Processing at Sensors

After taking **T** observations, each sensor sends 1 or 0 (no send) based on the average of the **T** observations. Assuming i.i.d observations with respect to time, the local decision rule can be formulated as:

$$S_{k,i} = \frac{1}{T}\sum_{t=1}^{T}\left[\left(\sum_{i=1}^{N} a_{k,i}\right) + n_{k,t}\right] \underset{u_k = 1}{\overset{u_k = 0}{\substack{\leq \\ >}}} \tau_k \qquad 3.80$$

$$P_k(u_k = 1|S_k) = Q\left(\frac{\tau_k - \left(\sum_{i=1}^{N} a_{k,i}\right)}{\sigma_k}\right) \qquad 3.81$$

$$P_k(u_k = 0|S_k) = 1 - Q\left(\frac{\tau_k - \left(\sum_{i=1}^{N} a_{k,i}\right)}{\sigma_k}\right) \qquad 3.82$$

where τ_k is a quantization threshold, u_k is the local decision of the k-th sensor, σ_k is the variance of S_k, and $Q(.)$ is the Q-function which is the complementary cumulative distribution function (CDF) of the Gaussian distribution. For example, assume we have number of sensors **k=1**, and target **i=1** and observations has done for **T=3** times. The signal $\boldsymbol{a_{1,1}}$ is received from the target are: 4, 5, 7, then:

$$S_{1,1} = \frac{1}{3}\sum_{t=1}^{3}\left[\left(\sum_{i=1}^{1} a_{1,1}\right) + n_{1,t}\right] = \frac{1}{3}\sum_{t=1}^{3}[(4+5+7) + n_{1,t}]$$

if $n_{1,t}$= 0.5, 0.5, 0.3 for t=1,2,3 then:

$S_{1,1} = \frac{1}{3}[(16) + 0.5 + 0.5 + 0.3] = 5.77$. This observation value will be compared to τ_k in order to decide if the local decision is 1 or 0.

3.6.3. Processing at the Fusion Center

The channel between sensors and the fusion center is assumed to be error free channel. Once the fusion center receives all sent decisions, the estimation process of the parameter:

$\boldsymbol{\theta} = [\mathbf{x_1},\ \mathbf{y_1}, \mathbf{x_2},\ \mathbf{y_2}, \dots, \mathbf{x_N},\ \mathbf{y_N}]^{\mathbf{T}}$ is carried out. Other parameters, i.e., $\boldsymbol{\tau_k}$, $\boldsymbol{\sigma_k}$, **N**, are assumed to be known at fusion center. However, efficient algorithms and estimators can be used to estimate the targets number. The likelihood estimator can be written as:

$$p(\boldsymbol{U}|\boldsymbol{\theta}) = \prod_{k=1}^{K}\left([P_k(u_k = 1|S_k)]^{(u_k)}[P(u_k = 0|S_k)]^{(1-u_k)}\right) \quad 3.83$$

Taking the ***ln*** for both sides, the log likelihood estimator is obtained as:

$$\ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big) = \sum_{k=1}^{K} u_k\, ln\big(P_k(u_k = 1|S_k)\big) +\ (1 - u_k)\, ln\big(P_k(u_k = 0|S_k)\big) \quad 3.84$$

Assume

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)\\ &= \sum_{k=1}^{K} u_k ln\left(Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right)\\ &\quad +\ (1 - u_k)\, ln\left(1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right)\end{aligned} \quad 3.85$$

Where $A_k(\boldsymbol{\theta}) = \sum_{i=1}^{N} a_{k,i}$.

The maximum likelihood estimator is given as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big) \quad 3.86$$

Any efficient optimization method can be used to find θ that maximizes $\mathcal{L}$(θ), i.e., particle swarm optimization.

3.6.4. Cramer-Rao Lower Bound (CRLB)

The variance of unbiased estimator θ̂ of parameter θ can be bounded as the reciprocal of the Fisher information matrix [168, 169]. The mean square error might be compared to the Cramer value in order to measure the efficiency of the estimator.

$$E\left[\big(\widehat{\boldsymbol{\theta}}(\boldsymbol{U}) - \boldsymbol{\theta}\big)\big(\widehat{\boldsymbol{\theta}}(\boldsymbol{U}) - \boldsymbol{\theta}\big)^{H}\right] \geq \frac{1}{\boldsymbol{J}} \quad 3.87$$

The Fisher information matrix can be formulated as $(3\mathrm{N} \times 3\mathrm{N})$ matrix [168, 169]:

$$\boldsymbol{J} = \boldsymbol{E}\{[-\boldsymbol{\nabla_\theta}\boldsymbol{\nabla_\theta^T}\, ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)]\} \quad 3.88$$

For simplicity, we choose the number of targets to be 2 (N=2) in the later examples. Hence, J becomes 6X6 matrix. For this case, **J** can be formulated as:

$$\begin{bmatrix}
\frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial x_1^2}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_1 \partial y_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_1 \partial z_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_1 \partial x_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_1 \partial y_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_1 \partial z_2} \\
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_1 \partial x_1} & \frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial y_1^2}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_1 \partial z_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_1 \partial x_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_1 \partial y_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_1 \partial z_2} \\
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_1 \partial x_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_1 \partial y_1} & \frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial z_1^2}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_1 \partial x_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_1 \partial y_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_1 \partial z_2} \\
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_2 \partial x_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_2 \partial y_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_2 \partial z_1} & \frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial x_2^2}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_2 \partial y_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_2 \partial z_2} \\
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_2 \partial x_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial yz_2 \partial y_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_2 \partial z_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_2 \partial x_2} & \frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial y_2^2}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_2 \partial z_2} \\
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_2 \partial x_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_2 \partial y_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_2 \partial z_1} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_2 \partial x_2} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_2 \partial y_2} & \frac{\boldsymbol{\partial^2 \mathcal{L}(\theta)}}{\boldsymbol{\partial z_2^2}}
\end{bmatrix}$$

The main diagonal entries of ***J*** are:

$$-E\left[\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2}\right] = \sum_{k=1}^{K}\left\{\left(\frac{\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)}\right) + \left(\frac{\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)\right\} \qquad 3.89$$

$$-E\left[\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2}\right] = \sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2 q_k$$

where $\mathfrak{B}_{k,i} = \frac{P_0 d_0^n\, n^2}{8\pi\sigma_k^2}\left(d_{k,i}\right)^{-(n+4)}$, $\mathcal{A}_k = e^{\frac{-\left(\tau_k - A_k(\theta)\right)^2}{\sigma_k^2}}$ and $q_k = \left(\frac{1}{Q\left(\frac{\tau_k - A_k(\theta)}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\theta)}{\sigma_k}\right)\right]}\right)$.

By symmetry:

$$-E\left[\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial y_i^2}\right] = \sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(y_k - y_i)^2 q_k \qquad 3.90$$

$$-E\left[\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial z_i^2}\right] = \sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(z_k - z_i)^2 q_k \qquad 3.91$$

The off diagonal entries of J are:

$$-E\left[\frac{\partial^2 ln\left(p(\boldsymbol{U}|\boldsymbol{\theta})\right)}{\partial x_i \partial y_j}\right] = \sum_{k=1}^{K} \sqrt{\mathfrak{B}_{k,i}\mathfrak{B}_{k,j}}\; \mathcal{A}_k(x_k - x_i)\left(y_k - y_j\right) q_k \qquad 3.92$$

$$-E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial x_i \partial z_j}\right] = \sum_{k=1}^{K}\sqrt{\mathfrak{B}_{k,i}\mathfrak{B}_{k,j}}\;\mathcal{A}_k(x_k - x_i)\big(z_k - z_j\big)\,q_k \qquad 3.93$$

$$-E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial y_i \partial z_j}\right] = \sum_{k=1}^{K}\sqrt{\mathfrak{B}_{k,i}\mathfrak{B}_{k,j}}\mathcal{A}_k(y_k - y_i)\big(z_k - z_j\big)\,q_k \qquad 3.94$$

After evaluating J, the inverse of the matrix can be directly obtained using the MatLab command; ***inv(J)***. However, the main diagonal entries of the inverse matrix ***K*** are of our interest since they act the main square errors (MSEs).

$$\boldsymbol{K} = inv(\boldsymbol{J}) = \begin{bmatrix} \mathbf{MSE_{x1}} & K_{(1,2)} & K_{(1,3)} & K_{(1,4)} & K_{(1,5)} & K_{(1,6)} \\ K_{(2,1)} & \mathbf{MSE_{y1}} & K_{(2,3)} & K_{(2,4)} & K_{(2,5)} & K_{(2,6)} \\ K_{(3,1)} & K_{(3,2)} & \mathbf{MSE_{z1}} & K_{(3,4)} & K_{(3,5)} & K_{(3,6)} \\ K_{(4,1)} & K_{(4,2)} & K_{(4,3)} & \mathbf{MSE_{x2}} & K_{(4,5)} & K_{(4,6)} \\ K_{(5,1)} & K_{(5,2)} & K_{(5,3)} & K_{(5,4)} & \mathbf{MSE_{y2}} & K_{(5,6)} \\ K_{(6,1)} & K_{(6,2)} & K_{(6,3)} & K_{(6,4)} & K_{(6,5)} & \mathbf{MSE_{z2}} \end{bmatrix}$$

For example the following MSEs are obtained when the number of sensors is 625.

$$\boldsymbol{J} = \begin{bmatrix} \mathbf{0.2979} & 0.0047 & 0.0181 & -0.0091 & -0.0764 & -0.1002 \\ 0.0047 & \mathbf{0.2979} & 0.0181 & -0.0764 & -0.0091 & -0.1002 \\ 0.0181 & 0.0181 & \mathbf{0.7583} & -0.1002 & -0.1002 & -0.0751 \\ -0.0091 & -0.0764 & -0.1002 & \mathbf{0.2979} & 0.0047 & 0.0181 \\ -0.0764 & -0.0091 & -0.1002 & 0.0047 & \mathbf{0.2979} & 0.0181 \\ -0.1002 & -0.1002 & -0.0751 & 0.0181 & 0.0181 & \mathbf{0.7583} \end{bmatrix}$$

$$\boldsymbol{K} = inv(\boldsymbol{J}) = \begin{bmatrix} \mathbf{3.7722} & 0.1713 & 0.1039 & 0.1477 & 0.9744 & 0.5043 \\ 0.1713 & \mathbf{3.7722} & 0.1039 & 0.9744 & 0.1477 & 0.5043 \\ 0.1039 & 0.1039 & \mathbf{1.4616} & 0.5043 & 0.5043 & 0.1481 \\ 0.1477 & 0.9744 & 0.5043 & \mathbf{3.7722} & 0.1713 & 0.1039 \\ 0.9744 & 0.1477 & 0.5043 & 0.1713 & \mathbf{3.7722} & 0.1039 \\ 0.5043 & 0.5043 & 0.1481 & 0.1039 & 0.1039 & \mathbf{1.4616} \end{bmatrix}$$

$\mathbf{MSE_{x1}} = 3.7722$, $\mathbf{MSE_{y1}} = 3.7722$, $\mathbf{MSE_{z1}} = 1.4616$, $\mathbf{MSE_{x2}} = 3.7722$, $\mathbf{MSE_{y2}} = 3.7722$ and $\mathbf{MSE_{z2}} = 1.4616$.

After estimating the location of the targets, the distances between them can be found. For example, a top view and 3D view of the WSNs with the two targets and the estimated locations are shown in Fig. 3.42 and Fig. 3.43, respectively. The real locations of the two targets are shown as red circles while the estimated locations are shown as black circles. The obtained estimated locations of target 1 and 2 are, respectively; (36.2553, 38.4682, 3.3205) and (-37.0593, -37.9490, 1.5082). Thus, after estimating the locations of the two targets, the estimated distance between them in 3D space can be calculated. For the last example the real distance was 113.1415 meters while the estimated distance is 105.9146 meters.

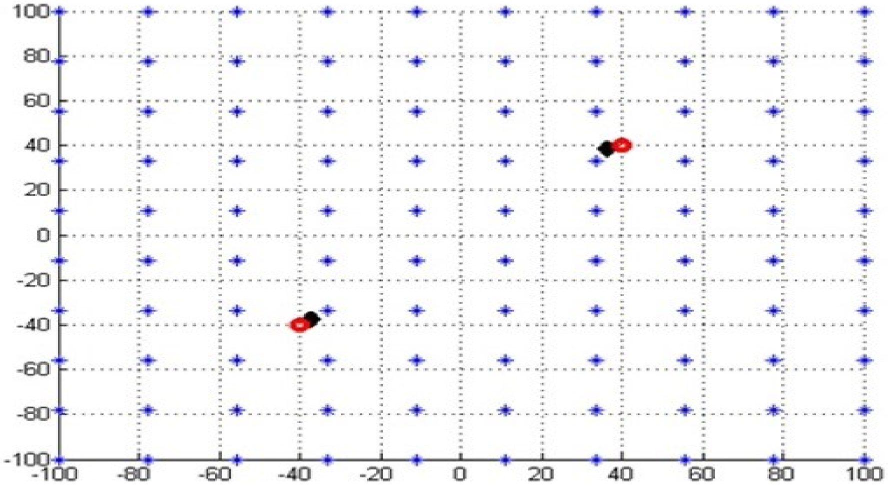


Figure 3. 42 2D top view show of the two targets and the WSNs.

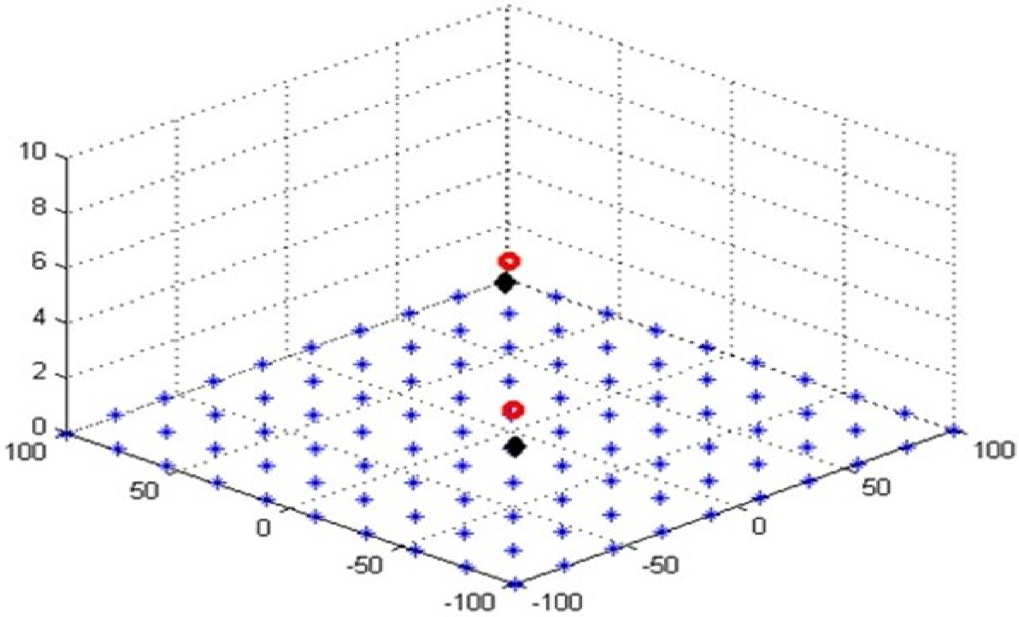


Figure 3. 43 3D show of the 2 targets and the WSNs.

The Fig. 3.44 shows the two targets (blimp robot and the helicopter) when they are moving from position to another in the environment. The objective here is to estimate the locations of the targets and to estimate the distances between them. The three real and known locations of the two targets are shown in Table 3.2. In addition, group one, group two, and group three represent the three locations for the two targets {(L1, M1), (L2, M2), (L3, M3)}. In order to estimate the locations and to study the effect of increasing the number of the sensors we have used different number of sensors which are deployed in the environment. For each location **L** and **M,** we changed the number of the sensors. The number of the sensors are 5, 8, 11, 14, and 17 square sensors (25, 64, 121, 196, and 289). Because there are random noises and the practical swarm optimization method is an iteration process, there are 50 different estimations for each location as it is shown in Fig. 3. 45.

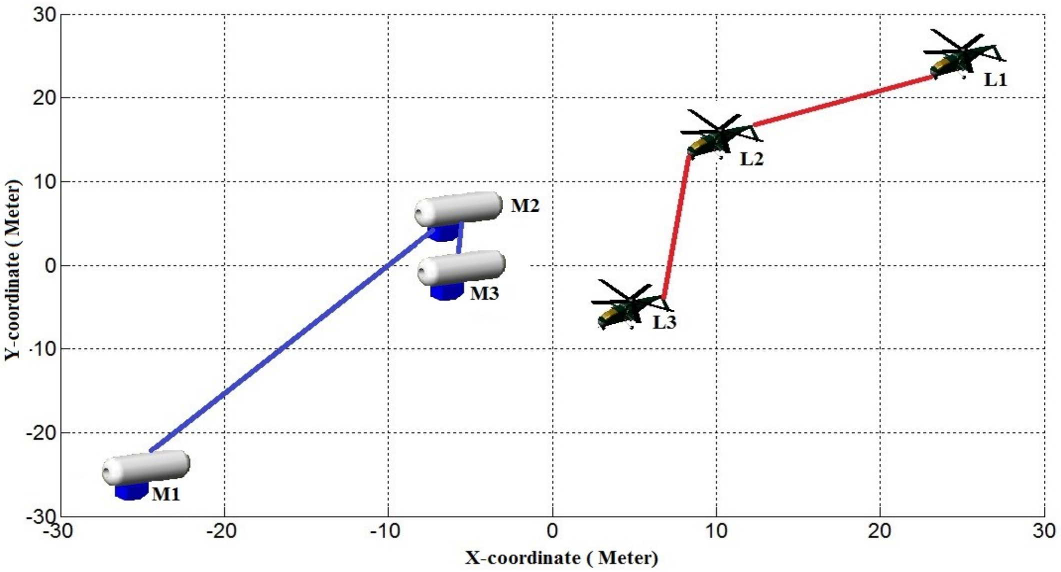


Figure 3. 44 **Blimp robot and helicopter robots targets.**

	Group One	Group Two	Group Three
Target 1 Locations	L1 (25,25)	L2 (10,15)	L3 (5,- 5)
Target 2 Locations	M1 (-25,-25)	M2 (-5,5)	M3 (5,-0)

Table 3. 2 **The real locations of the two targets.**

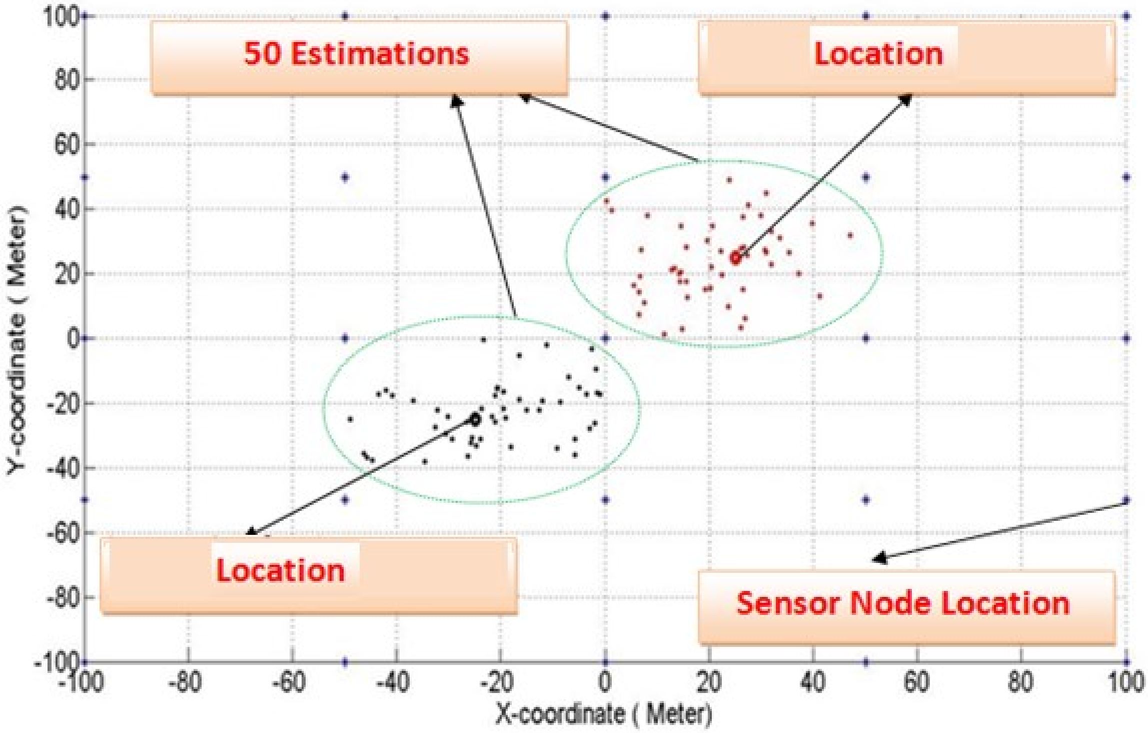


Figure 3. 45 **2D top view describes the important points.**

A top view of the **WSNs** with the two targets and the 50 estimated locations are shown in the Fig. 3.46 to Fig. 3.60. The big black and red points are the real locations for the blimp and helicopter target, respectively. The small dots (points) resemble the estimated positions obtained from the estimator. The process in order to estimate both of them done with different number of sensors. It is obviously clear that when the number of the sensors increases, the estimation of the locations will increase. However, comparing the Cramer Rao Lower Bound **(CRLB)** with the Mean Square Error **(MSE)** leads to measure the efficiency of the maximum likelihood estimator. Then, find the minimum number of sensors that are needed in order to get accurate locations when the MSE achieves or approaches the CRLB.

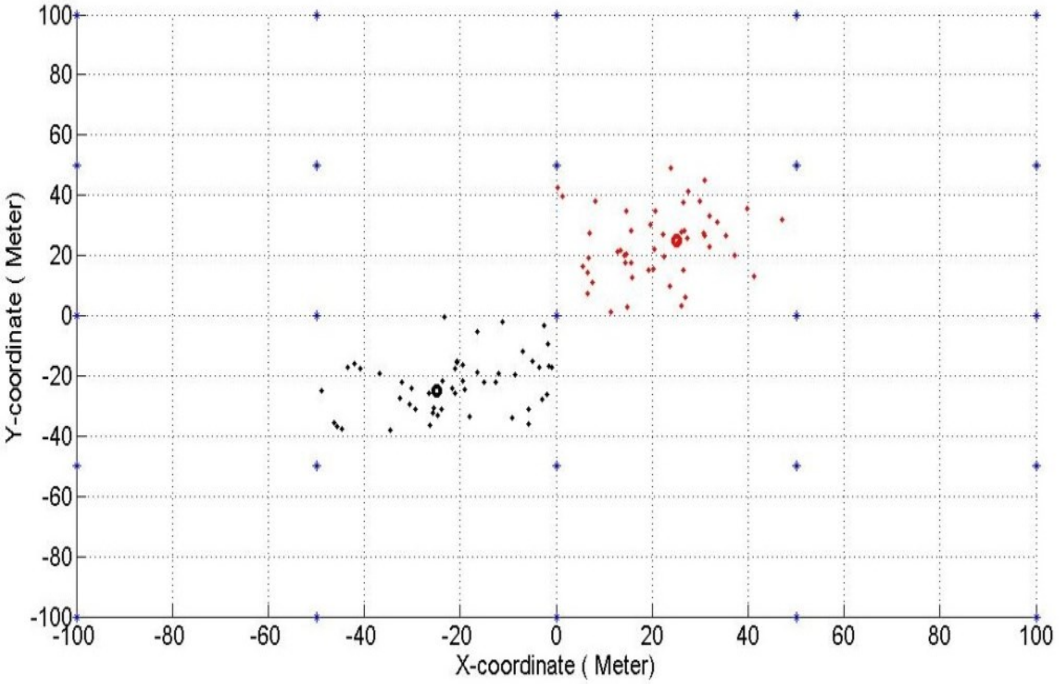


Figure 3. 46 2D top view shows the 2 targets and 5 sensors for location 1.

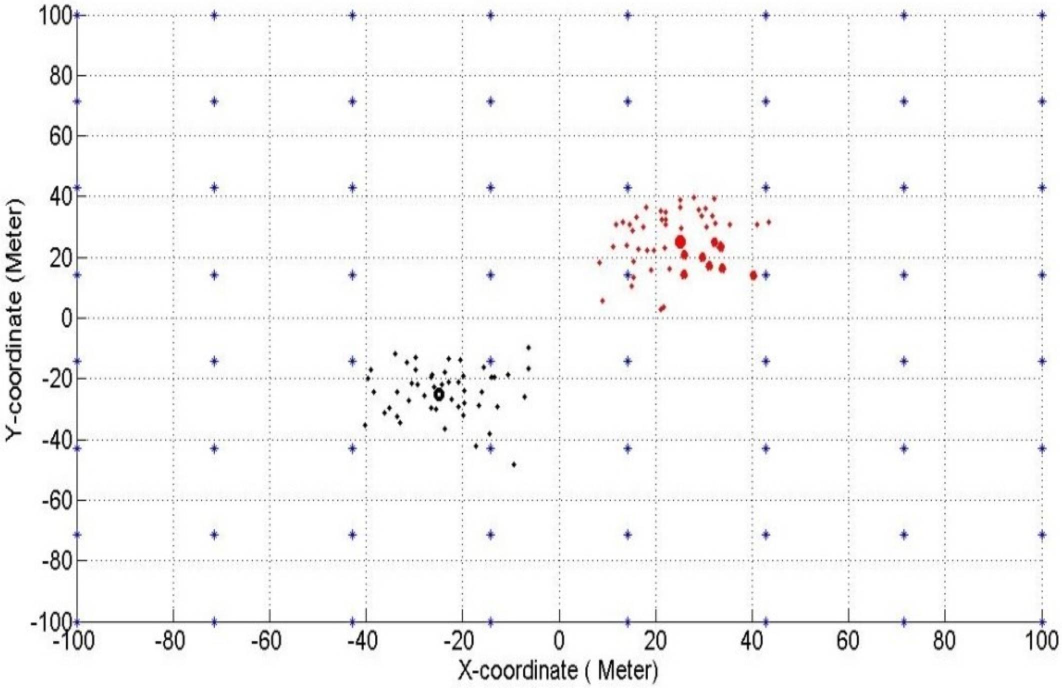


Figure 3. 47 2D top view shows the 2 targets and 8 sensors for location 1.

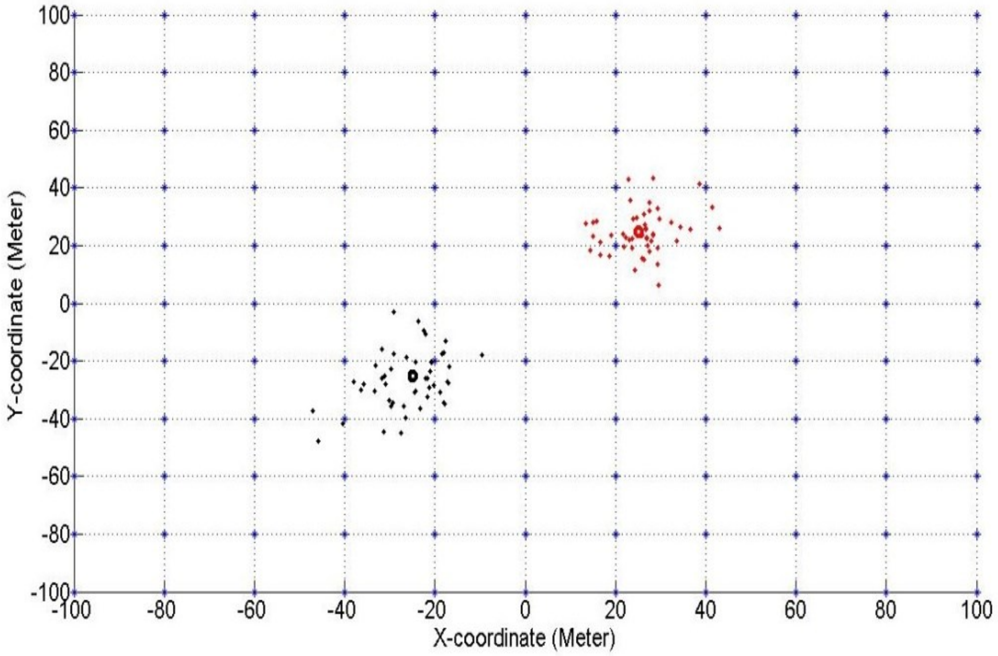


Figure 3. 48 2D top view shows the 2 targets and 11 sensors for location 1.

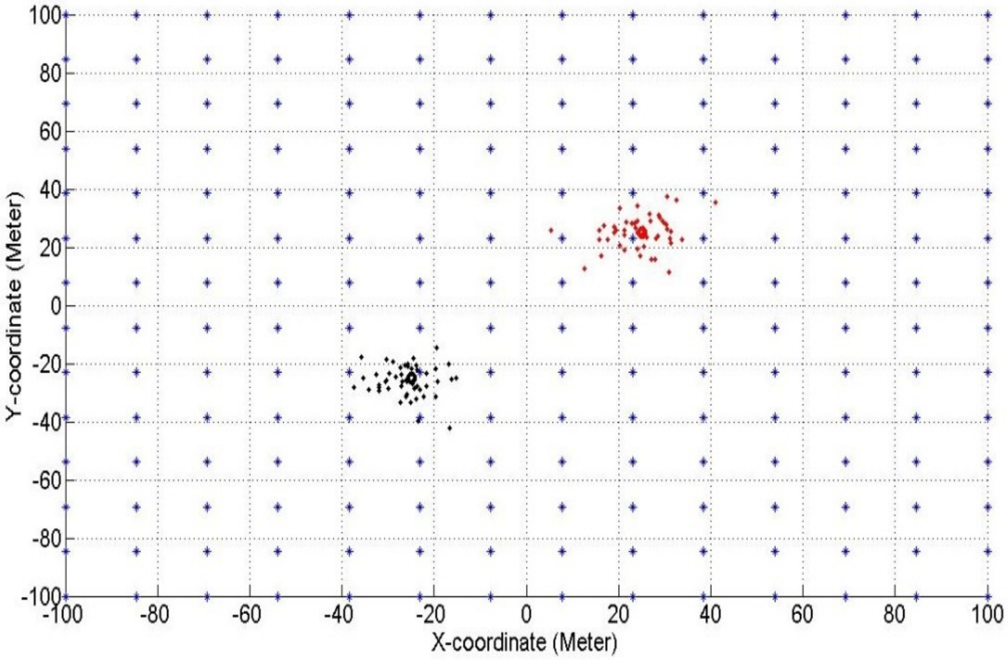


Figure 3. 49 2D top view shows the 2 targets and 14 sensors for location 1.

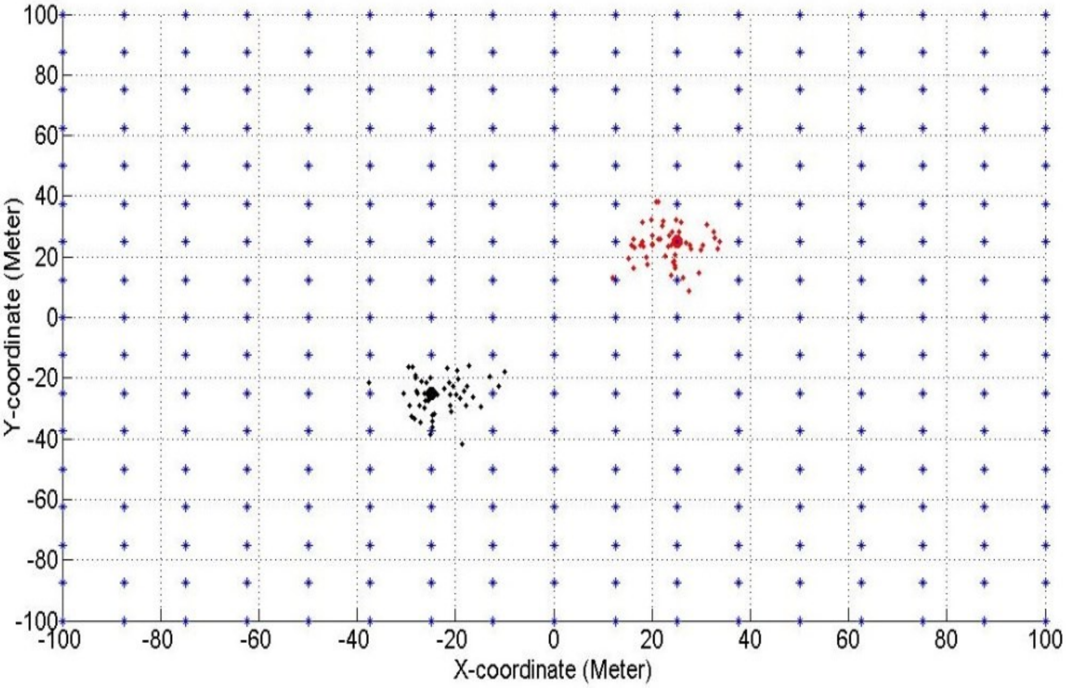


Figure 3. 50 2D top view shows the 2 targets and 17 sensors for location 1.

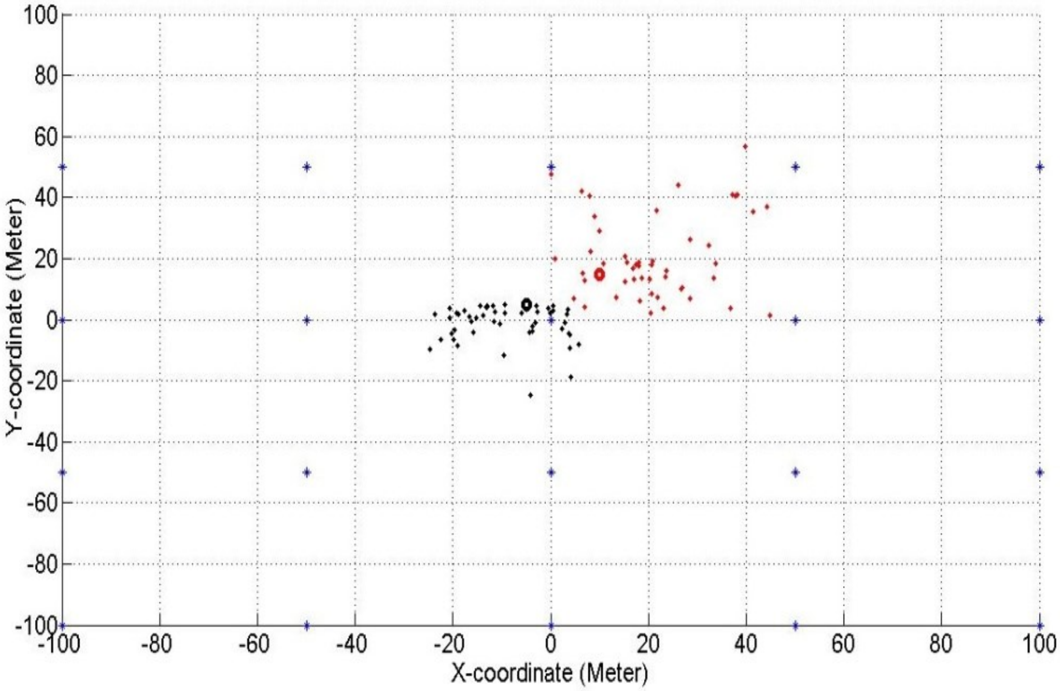


Figure 3. 51 2D top view shows the 2 targets and 5 sensors for location 2.

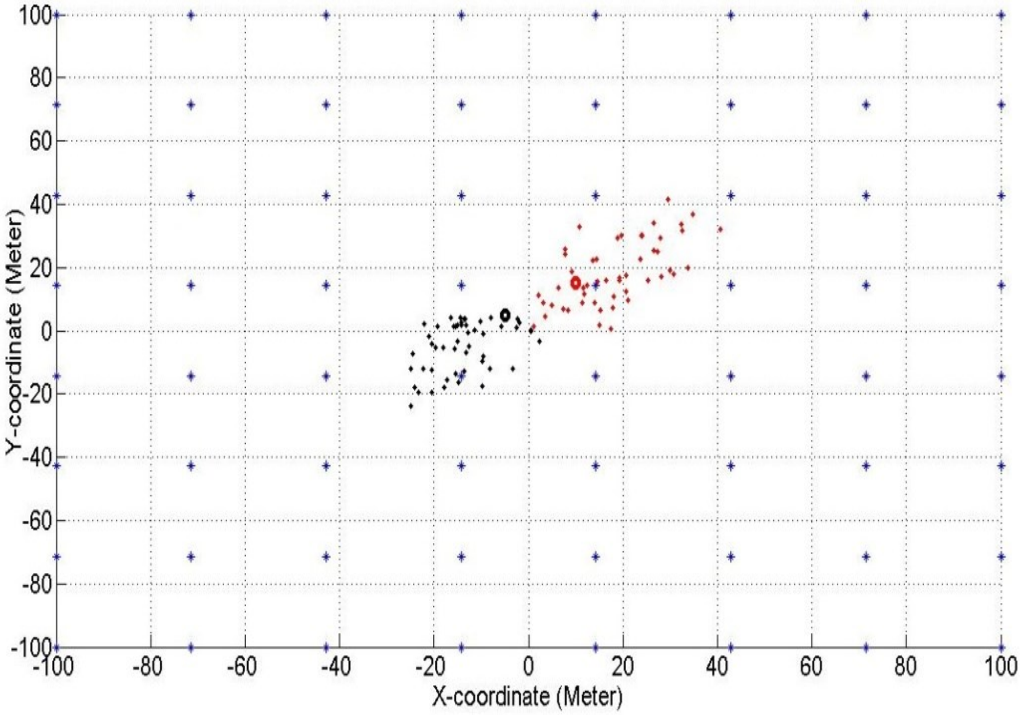


Figure 3. 52 2D top view shows the 2 targets and 8 sensors for location 2.

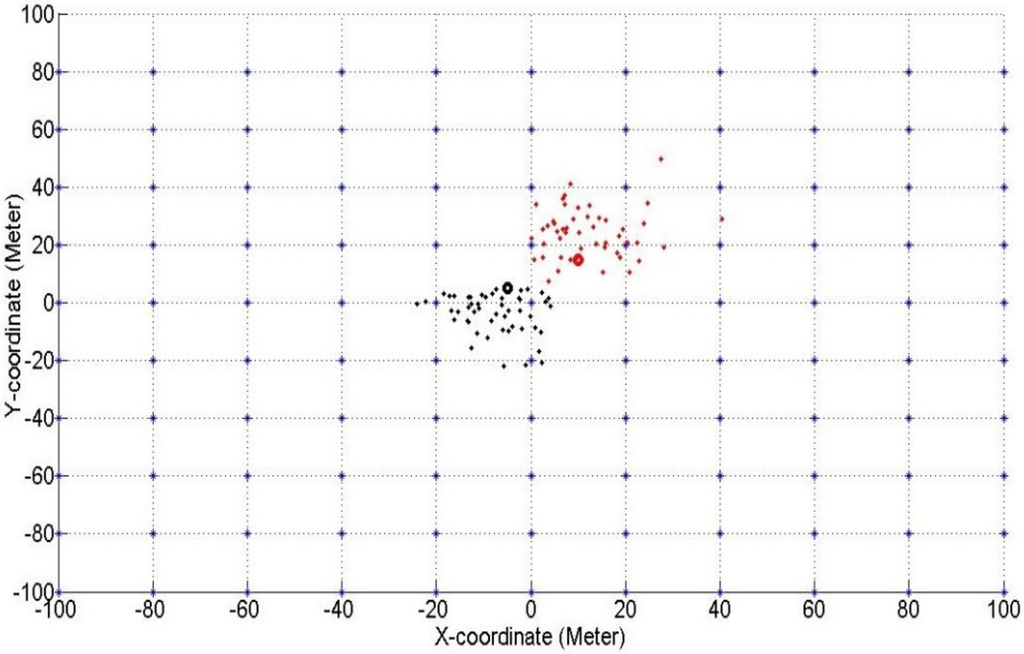


Figure 3. 53 2D top view shows the 2 targets and 11 sensors for location 2.

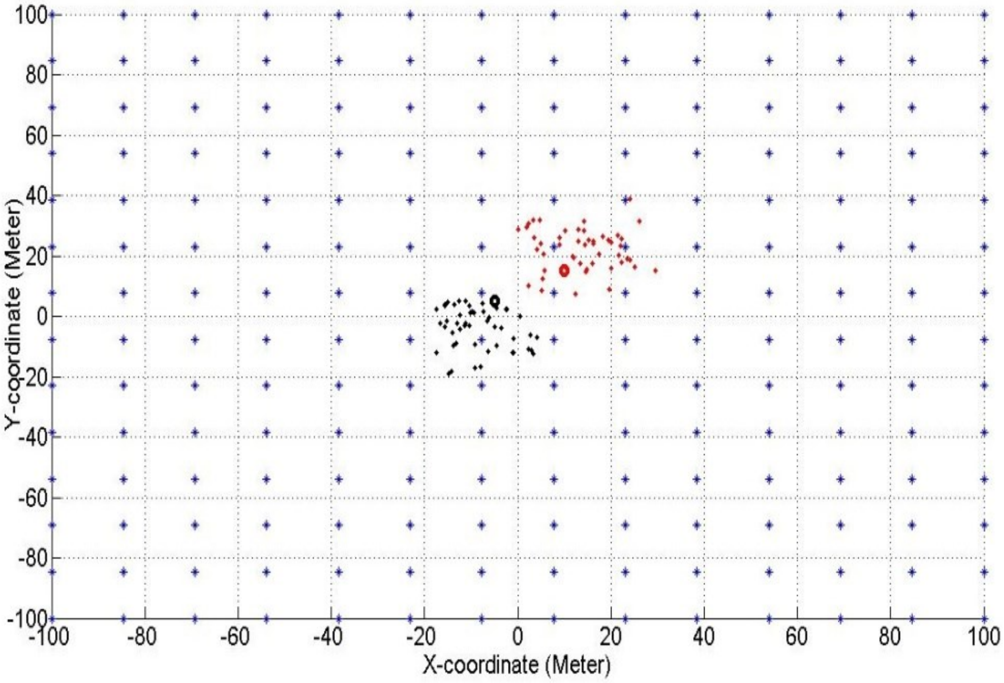


Figure 3. 54 2D top view shows the 2 targets and 14 sensors for location 2.

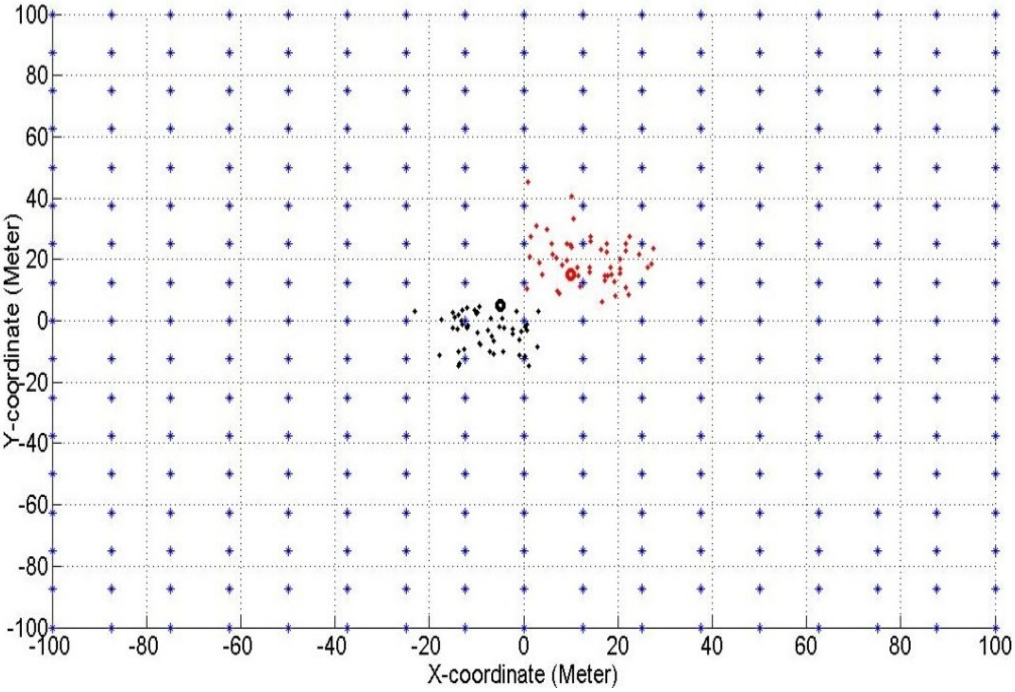


Figure 3. 55 2D top view shows the 2 targets and 17 sensors for location 2.

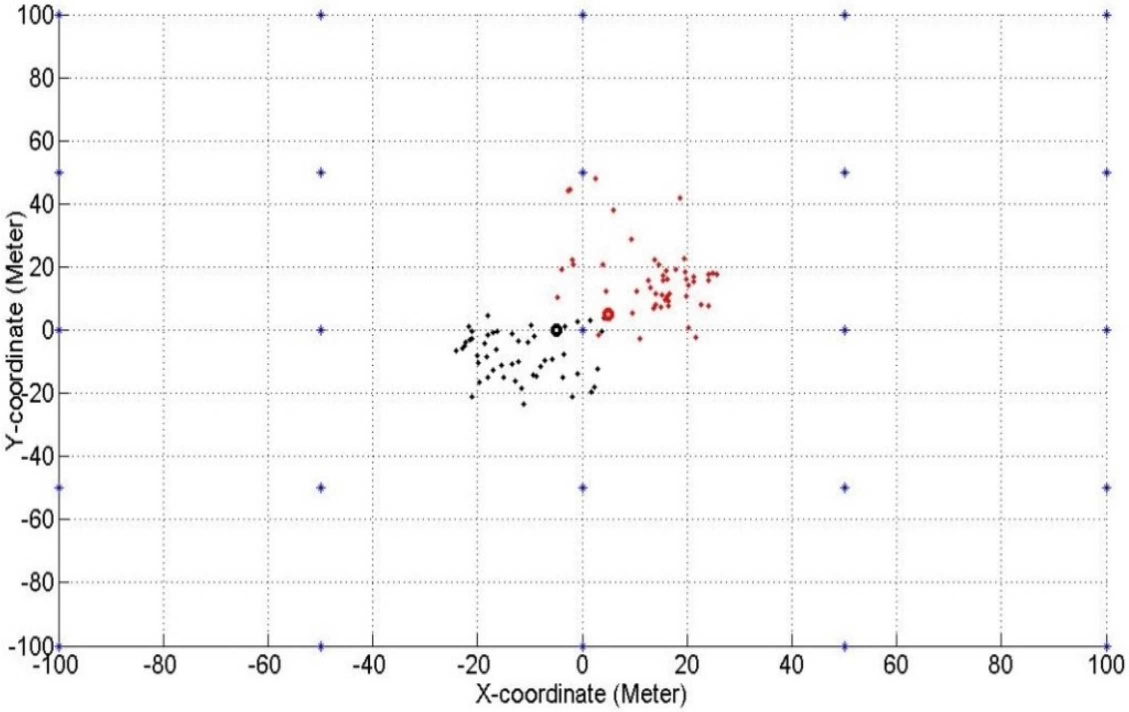


Figure 3. 56 2D top view shows the 2 targets and 5 sensors for location 3.

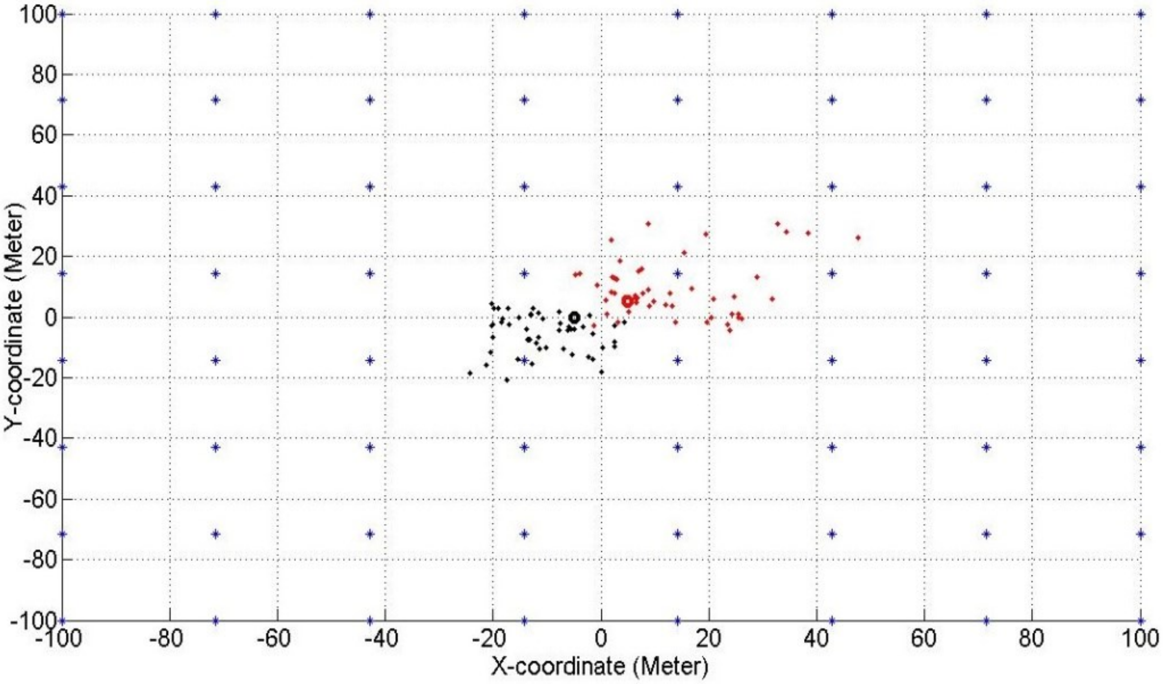


Figure 3. 57 2D top view shows the 2 targets and 8 sensors for location 3.

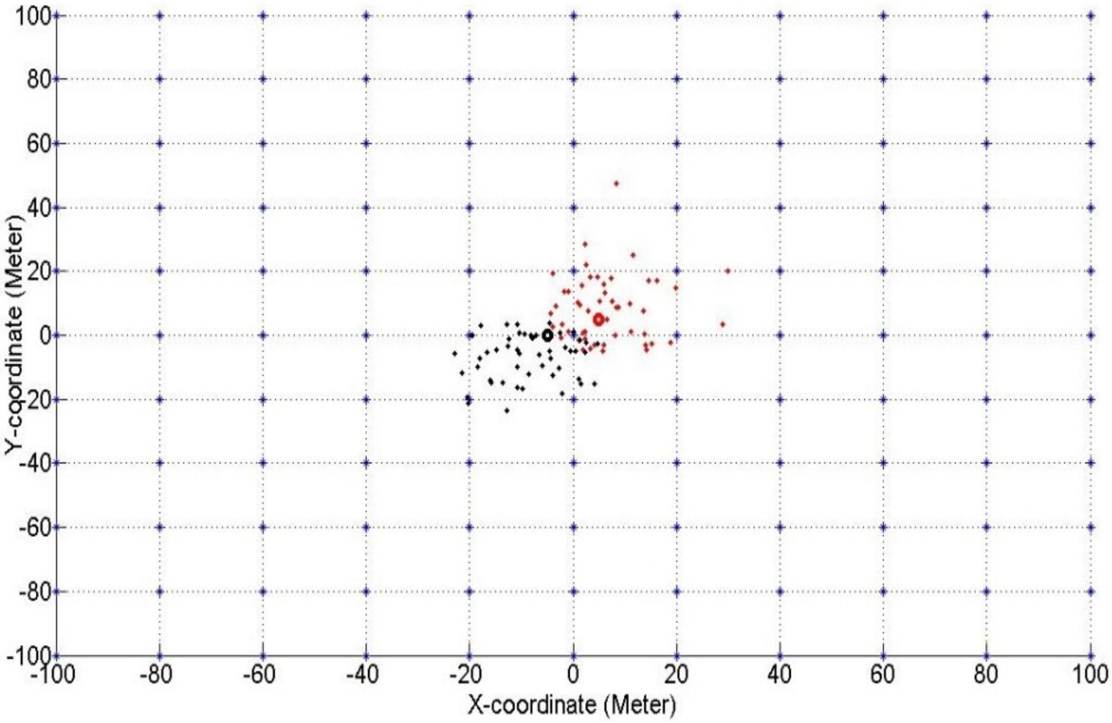


Figure 3. 58 2D top view shows the 2 targets and 11 sensors for location 3.

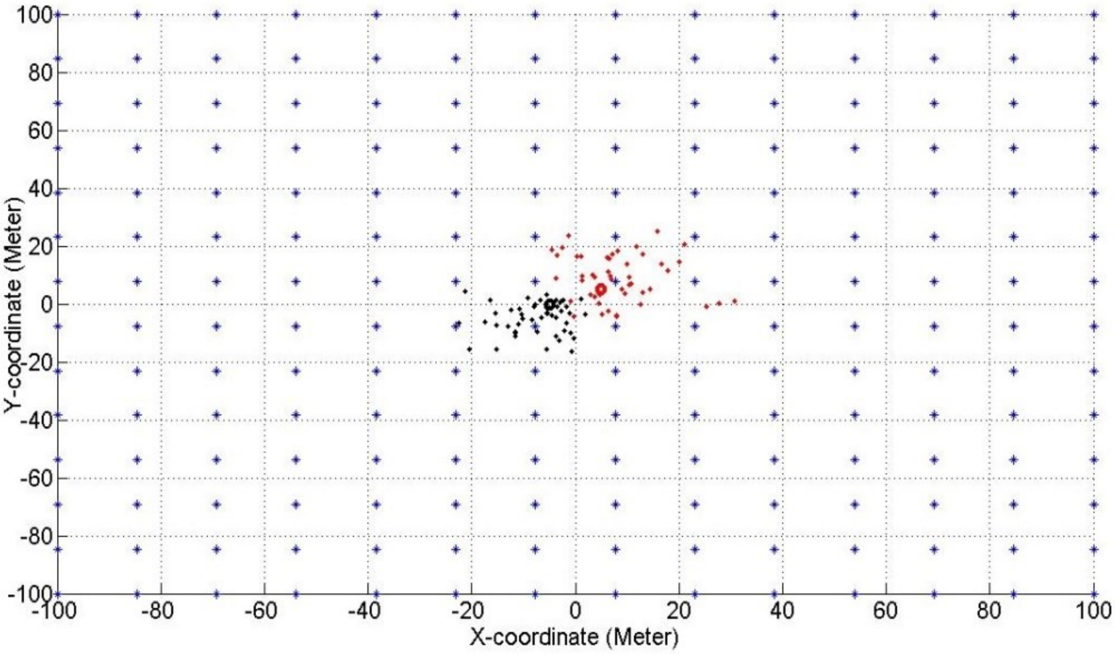


Figure 3. 59 2D top view shows the 2 targets and 14 sensors for location 3.

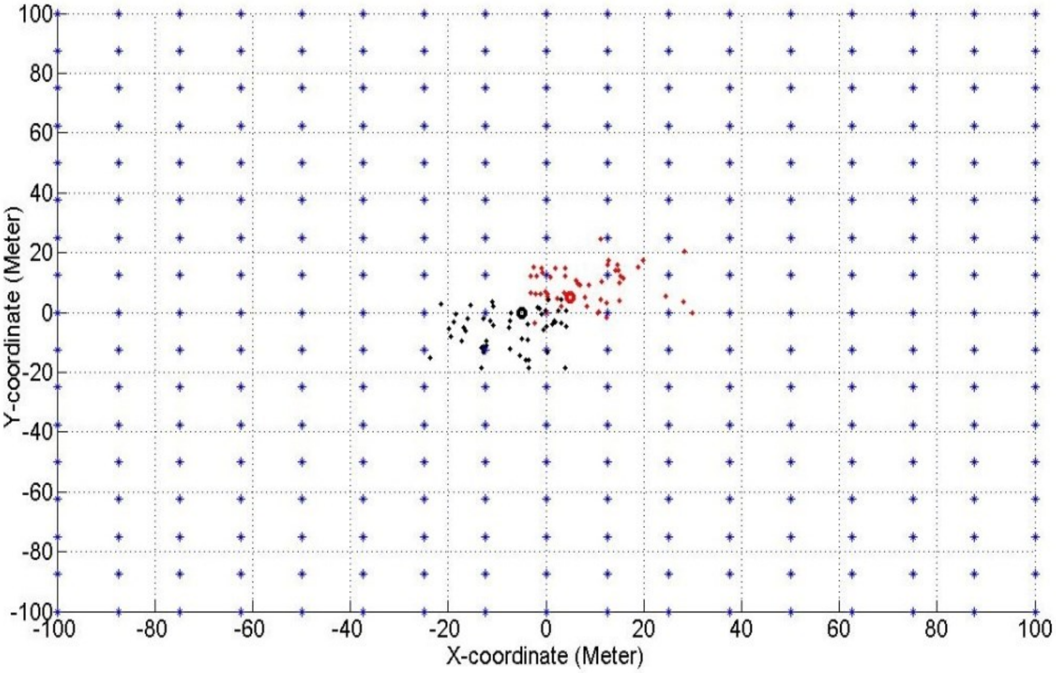


Figure 3. 60 2D top view shows the 2 targets and 17 sensors for location 3.

3.7 Conclusion

In this chapter, we introduced the cooperative perception issues in the robot system and the target tracking problems; then, discussed the fuzzy sets models based on the combination between the possibilities distributions and fuzzy logic. The process to design and optimize the fuzzy knowledge base was presented. Then, the communication protocols between the blimp robot and other robots in the system was discussed. Finally, the wireless sensor network which can be used in indoor or outdoor environments to track and localize the multi objects (aerial or ground).

Chapter 4 Computer Vision for Blimp Robot

4.1. Introduction

This chapter summarizes the most image processing and computer vision approaches and methods which have been used during this thesis. First, different approaches and methods are introduced and studied in order to find relevant visual features and matching useful for object detection, identifying and tracking in the image sequences. Then, the tracking method is used to estimate a projection model in the camera image space. Finally, implement the robust embedded visual algorithms on-board the blimp to give accurate and fast information for the blimp control system which can be used in some applications like visual object tracking as well as navigation. Indeed, the dramatic increases in the robots applications have made the computer vision an important key in several applications such as control systems. The computer vision recently proved being more powerful in robot applications since it uses non-intrusive and low cost sensors which makes the robot more intelligent. Nowadays, many designers or developers start using light-weight cameras mounted on the robot for several purposes such as stream the image sequences to the ground station, use the obtained images in autonomous and navigation tasks, and use images for servoing system applications.

Perhaps the complexity degree of the robots applications plays a significant role to classify the computer vision. The computer vision could be classified as a visual tracking if the robot analyzes the visual data in order to identify and follow interest points on the images. On the other hand, the visual navigation uses the visual data to determine objects or obstacles positions; thus, the suitable and safe path could be found. Needless to mention the map-building application that makes the visual navigation a very common way especially for cooperative systems in unstructured environment. The visual servoing technique uses the data that are extracted from visual sensor to feedback the control system in order to increase the capabilities of the blimp system.

4.2. Image Based Visual Tracking

Image processing is widely used to find the image characteristics and to recognize interest points (keypoints) for any objects in the environment. These features could be points, edges or even more complex objects such as ground robots or obstacles. Such features might be used as references for several applications like visual servoing tasks and visual navigation applications. The most features used as reference are interest points that have the most important characteristics for any objects. The most and well known detector used in order to extract robustness corners very quickly based on the eigenvalues of the autocorrelation matrix is the Harris corner detector [170]. The Harris corner detector is very famous and known as a most popular interest point detector because its ability to be strong invariance for both rotation and scale as well as it is robust against noise and illumination [171]. However, because the features and interests points are going to be tracked and found in image sequences for real applications; the robustness of this detector (Harris corner) is not enough and sufficient for this kind of purposes. Hence, the good features to track have to be selected to ensure the stability of the tracking process [172]. Therefore, the tracking features issue might be solved with several approaches or methods and the most popular technique for that is the Lucas Kanade Technique (LKT) which has demonstrated a good speed performance and good stability for any small changes [173]. Despite LKT technique successfully applying and showing good robustness for image scaling, rotation and illumination changes, it is expensive time computing [173].

4.2.1. Optical Flow and Appearance Tracking

The optical flow means to track the specific features in an image across multiple frames, and then determine the speed and direction of movement of objects. Therefore, we need to calculate the motion between two frames that take time **t** and **t+ Δt**. Given a point $\mathrm{P} = (\mathrm{x}, \mathrm{y}, \mathrm{t})$ in the image at time **t** with intensity $\mathrm{I}(\mathrm{x}, \mathrm{y}, \mathrm{t})$ which moves by $\Delta\mathrm{x}, \Delta\mathrm{y}, \Delta\mathrm{t}$. Then, the following image constraint can be given as[173, 174]:

$$\mathrm{I}(\mathrm{x}, \mathrm{y}, \mathrm{t}) = \mathrm{I}(\mathrm{x} + \Delta\mathrm{x}, \mathrm{y} + \Delta\mathrm{y}, \mathrm{t} + \Delta\mathrm{t}) \qquad 4.1$$

Considering small movement, the image constraint with Taylor series expansion can be calculated as the following:

$$\mathrm{I}(\mathrm{x}+\Delta\mathrm{x},\mathrm{y}+\Delta\mathrm{y},\mathrm{t}+\Delta\mathrm{t}) = \mathrm{I}(\mathrm{x},\mathrm{y},\mathrm{t}) + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\Delta\mathrm{x} + \frac{\partial \mathrm{I}}{\partial \mathrm{y}}\Delta\mathrm{y} + \frac{\partial \mathrm{I}}{\partial \mathrm{t}}\Delta\mathrm{t} + HOT \qquad 4.2$$

where $\frac{\partial \mathrm{I}}{\partial \mathrm{x}}, \frac{\partial \mathrm{I}}{\partial \mathrm{y}}, \frac{\partial \mathrm{I}}{\partial \mathrm{t}}$ are considered to be the image derivatives at this point $\mathbf{P}$, **HOT** is the Higher Order Terms which are ignored:

$$\frac{\partial \mathrm{I}}{\partial \mathrm{x}}\Delta\mathrm{x} + \frac{\partial \mathrm{I}}{\partial \mathrm{y}}\Delta\mathrm{y} + \frac{\partial \mathrm{I}}{\partial \mathrm{t}}\Delta\mathrm{t} = 0 \qquad 4.3$$

Then

$$\frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{v_x} + \frac{\partial \mathrm{I}}{\partial \mathrm{y}}\mathrm{v_y} + \frac{\partial \mathrm{I}}{\partial \mathrm{t}} = 0 \qquad 4.4$$

$$\mathrm{I_x v_x} + \mathrm{I_y v_y} = -\mathrm{I_t} \qquad 4.5$$

Where $\mathbf{v_x}, \mathbf{v_y}$ are the **x** and **y** velocities component, respectively. The Eq. 4.5 is called the aperture problem of the optical flow and it couldn't be solved directly. Therefore, in order to find the optical flow and resolve the optical flow, the **LKT** might be applied on small windows $\mathrm{m} \times \mathrm{m}$ around the $\mathrm{P} = (\mathrm{x},\mathrm{y})$.

$$\begin{gathered} \mathrm{I_{x1}v_x} + \mathrm{I_{y1}v_y} = -\mathrm{I_{t1}} \\ \mathrm{I_{x2}v_x} + \mathrm{I_{y2}v_y} = -\mathrm{I_{t2}} \\ \vdots \\ \mathrm{I_{xn-1}v_x} + \mathrm{I_{yn-1}v_y} = -\mathrm{I_{tn-1}} \end{gathered} \qquad 4.6$$

$$\mathrm{I_{xn}v_x} + \mathrm{I_{yn}v_y} = -\mathrm{I_{tn}}$$

The above formula is over determined since it has more **n** equations for the two unknowns. It could be rewritten in a form of matrix **Ax=b**.

$$\mathrm{A} = \begin{bmatrix} \mathrm{I_{x1}} & \mathrm{I_{y1}} \\ \vdots & \vdots \\ \mathrm{I_{xn}} & \mathrm{I_{yn}} \end{bmatrix}, \mathrm{x} = \begin{bmatrix} \mathrm{v_x} \\ \mathrm{v_y} \end{bmatrix}, \mathrm{b} = \begin{bmatrix} -\mathrm{I_{t1}} \\ \vdots \\ -\mathrm{I_{tn}} \end{bmatrix} \qquad 4.7$$

Now to solve the optical flow system the least square method could be implemented.

$$A^{T}Ax = A^{T}b \; \mathbf{OR} \; x = (A^{T}A)^{-1}A^{T}b \qquad 4.8$$

The sequence of images for the implemented optical flow and **LKT** is illustrating in Fig. 4.1a and Fig. 4.1b. In these figures we can recognize that the **LKT** can detect the moving object during the mission. In Fig. 4.1a it can detect the blimp robot as well as the person because this algorithm has ability to detect any moving object. Fig. 4.1b shows the algorithm detects and recognizes the ground robot. One advantage of this algorithm is its ability to detect the moving object and to predict the direction and velocity of the moving object. However, in order to detect only a certain target we need to improve the algorithm to distinguish and to separate the moving target from others. This can be done by implementing another technique based on color or shape which will add more time and makes it hard for real time applications.

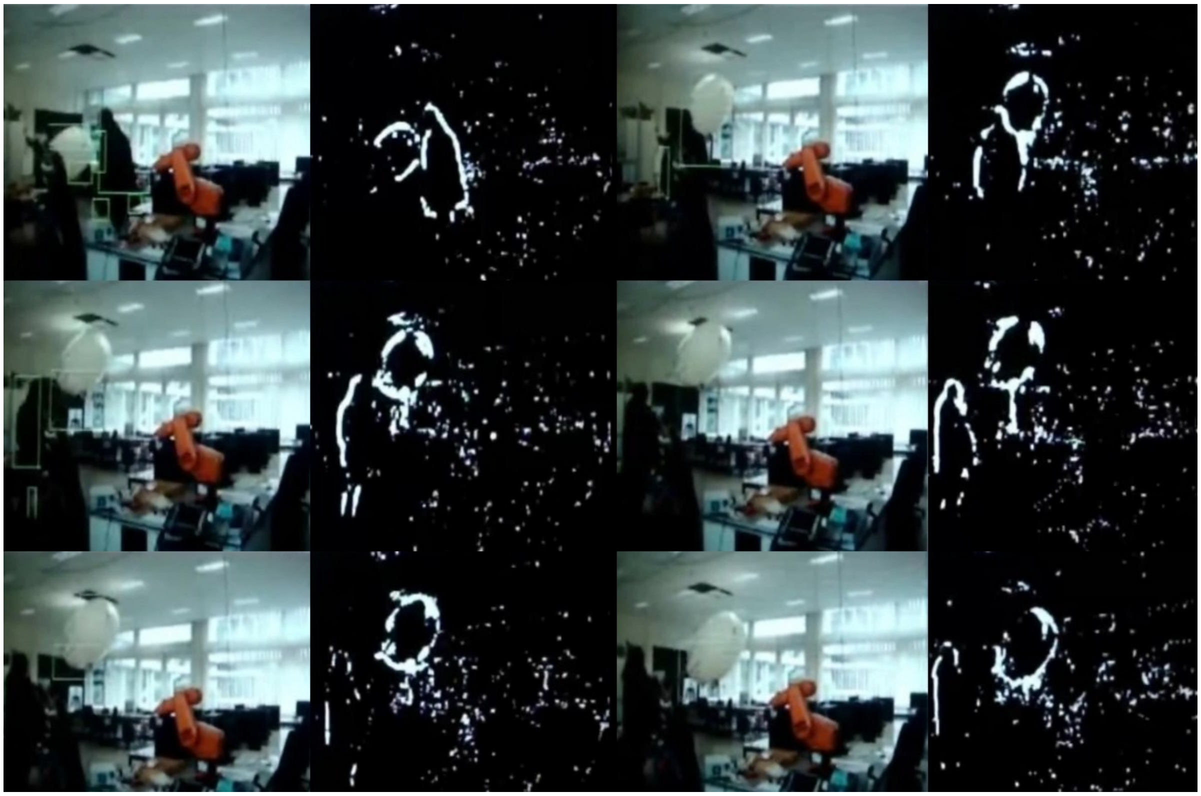


Figure 4. 1a **The optical flow sequence of images to detect blimp robot.**

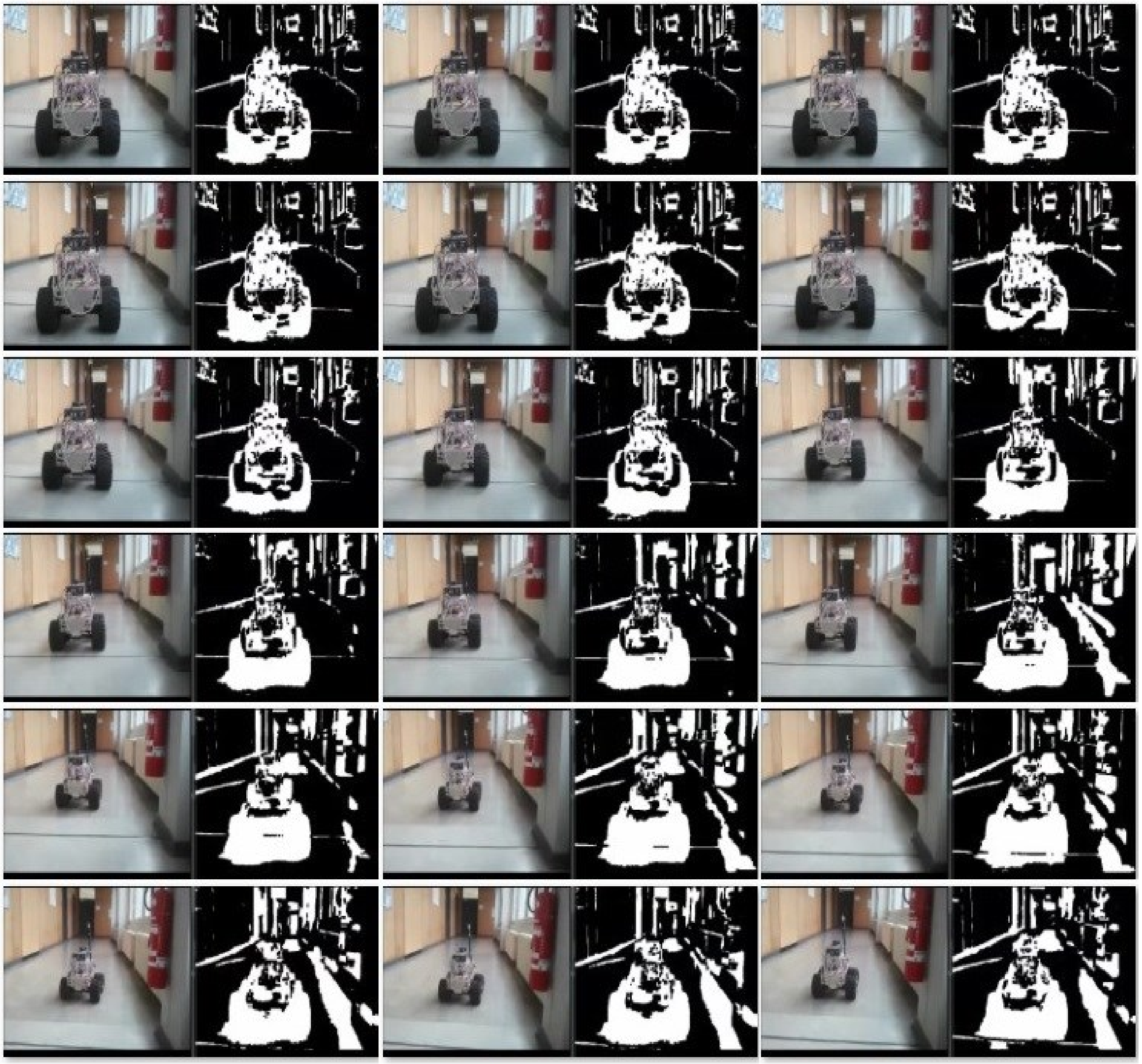


Figure 4.1b The optical flow sequence of images to detect ground robot.

4.2.2. Feature Descriptors and Tracking

Feature description defines as a computer vision process that can compute and identify the interest points in the image. The performance of the overall algorithm depends on the feature detector and the descriptors. One of the most desirable property for a detector is the repeatability which is the ability to detect same features in two different images of the same scene. This structure has to be rich enough in terms of local information contents which have to be robust under local and/or global perturbations and have to be a reliable recognition. Consequently, it is important for the features that are extracted from the training images (reference images) to be detectable even under changes in scale, rotations and translations as well as illumination variations. There are several feature descriptors suitable for visual matching and tracking. The most widely used descriptors are Scale Invariant Feature Transform (SIFT) and Speeded UP Robust Feature algorithm (SURF) [175, 176]. The SIFT detector is considered to be one of the most used algorithms for interest point detection and matching and it was developed to be used for object recognition. The SIFT could identify targets under partial occlusion and among clutter because it is ability to work and identify targets with scaling, orientation and illumination changes [175]. To ensure scale invariance it extracts features invariant to scale and rotation using Gaussian Difference (GD) of the images in different scales. In addition, in order to fulfill and realize invariance to rotation each feature is assigned by more orientations relies on the image gradient directions. The result of this process will be a descriptor which provides an efficient tool to represent an interest point. The calculation of the feature is considerable a computational cost and the feature depends on the nature of the task (whether this task needs to be done fast or accurate). On other hand, the SURF detector is based on the Hessian matrix which has a good performance in computation time and high accuracy. Given a point $\mathbf{s}(\mathbf{x},\mathbf{y})$ in an integral image I, the Hessian matrix $\mathbf{H}(\mathbf{s},\boldsymbol{\sigma})$ in $\mathbf{s}$ at scale $\boldsymbol{\sigma}$ is defined as the follows [176]:

$$\mathrm{H}(s,\sigma) = \begin{bmatrix} L_{xx}(s,\sigma) & L_{xy}(s,\sigma) \\ L_{xy}(s,\sigma) & L_{yy}(s,\sigma) \end{bmatrix} \quad 4.9$$

Where $\mathbf{L}(\mathbf{s},\boldsymbol{\sigma})$ is the convolution of the Gaussian second order derivative with the image $\mathbf{I}$ in point $\mathbf{s}$. This second-order derivatives matrix used for the image intensities in order to define the corner and to delete the offset of the gradient images. Moreover, the SURF algorithm extracts features from an image which can be tracked over multiple views and it can be used for object recognition. It has two advantages over SIFT as the author claim [176]. First, since the processing and computational time of SURF algorithm is less than that of SIFT(SURF is 3 time faster than SIFT), interest point detection of SURF algorithm could be used for real time processing. Second, SURF has been found more robust than SIFT if we consider the different image transformations [176]. The comparison between SIFT and SURF in order to detect one object (blimp robot and ground robot) in single image are shown in Fig. 4.2 a, b, c and Fig. 4.3 a, b, c. It is clearly seen that the time for computing features in SURF is about 2.6 times faster than SIFT. Also, the number of keypoints or interests points (red dots) in SURF is more than in SIFT for same image. The captured video from the blimp robot is shown in Fig. 4.4. This figure is shown the comparison between the SIFT and SURF algorithm for detecting the moving ground robot target. The number of the keypoints (interest points) for SURF is higher than for SIFT algorithm as it is illustrating in Fig. 4.6. In addition, the comparison time for detect keypoints between SIFT and SURF is shown in Fig. 4.5 which indicates that the SURF is much faster than the SIFT to detect the same object in the sequence of images.

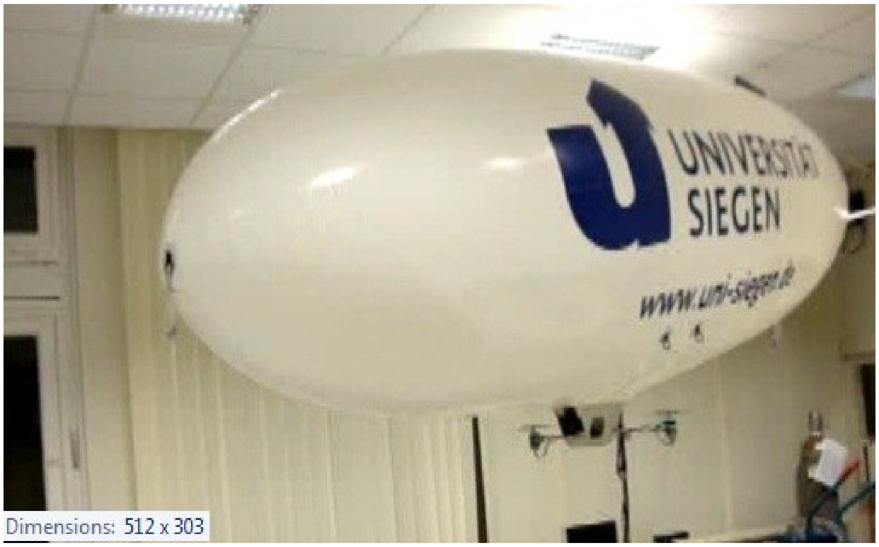


Figure 4. 2a The object test pictures1.

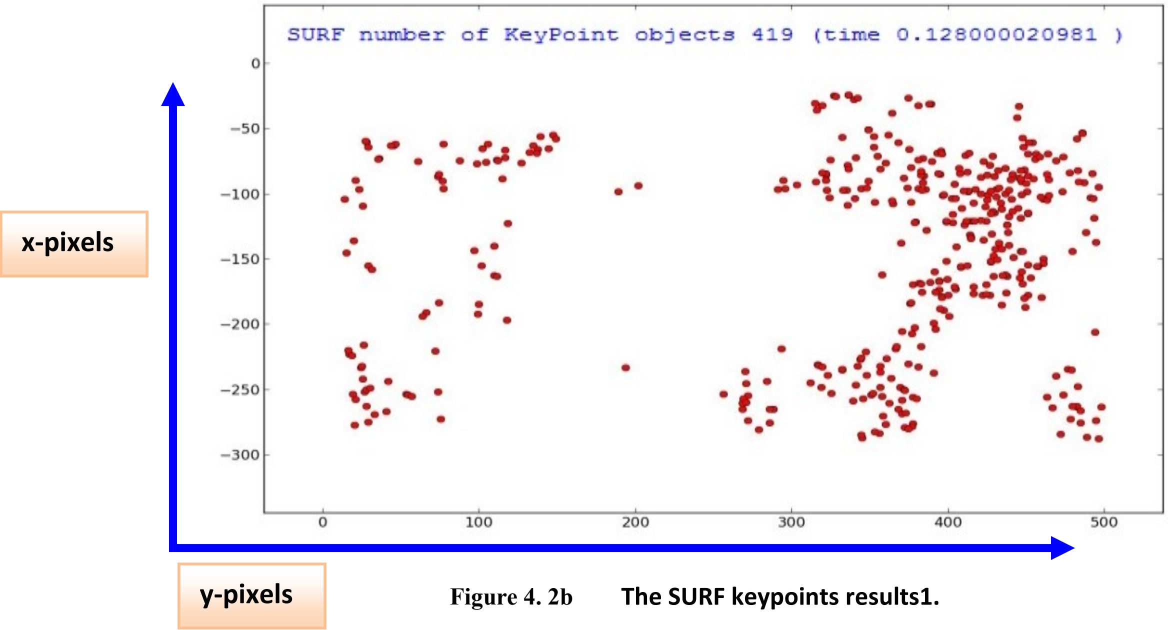


Figure 4. 2b The SURF keypoints results1.

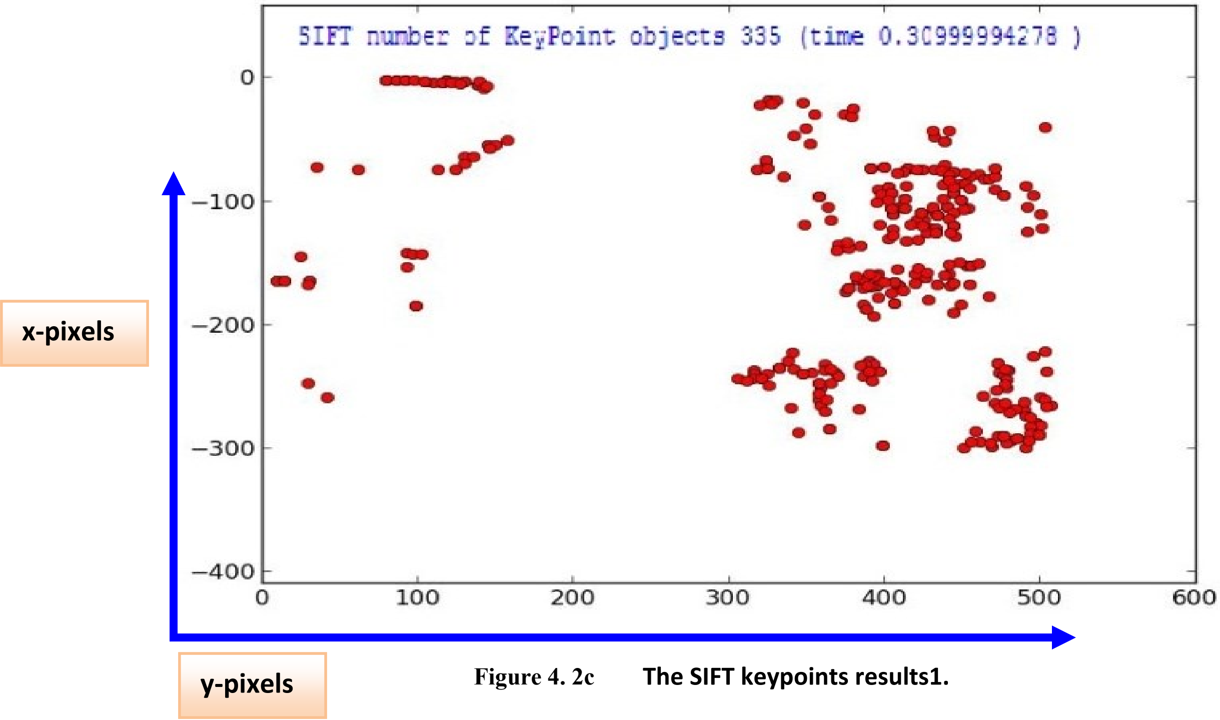


Figure 4. 2c The SIFT keypoints results1.

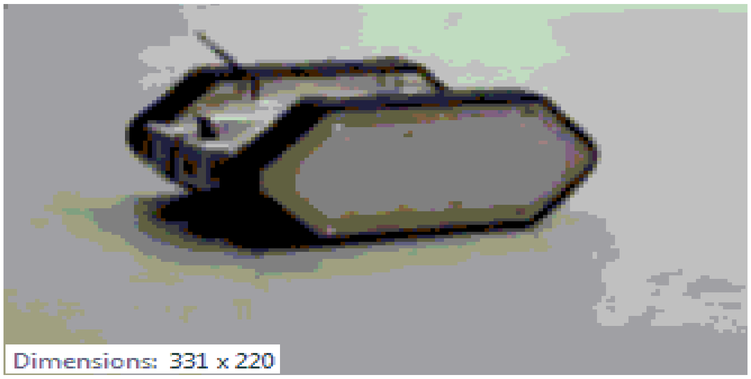


Figure 4. 3a The object test pictures2.

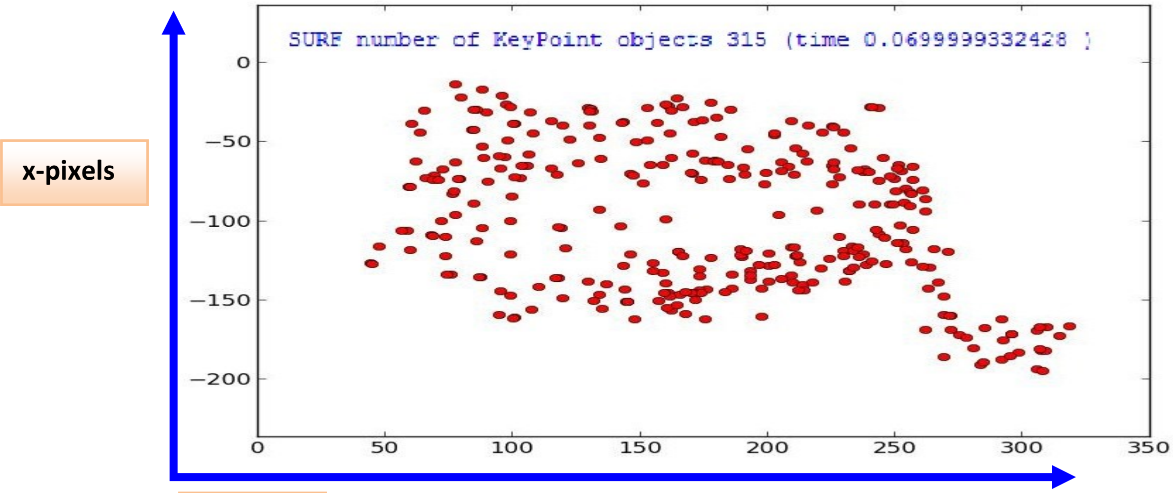


Figure 4. 3b The SURF keypoints results 2.

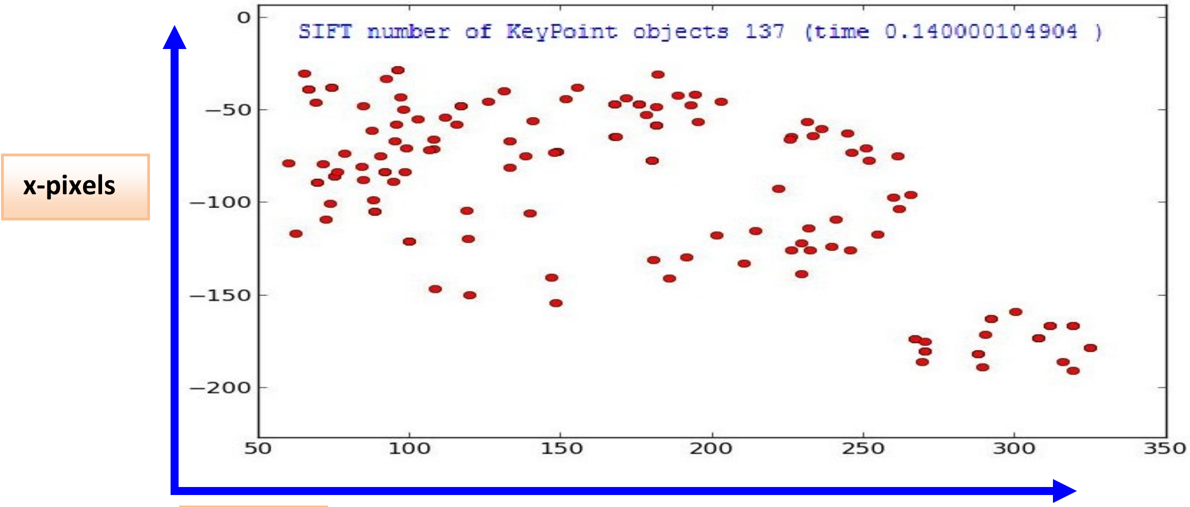


Figure 4. 3c The SIFT keypoints results2.

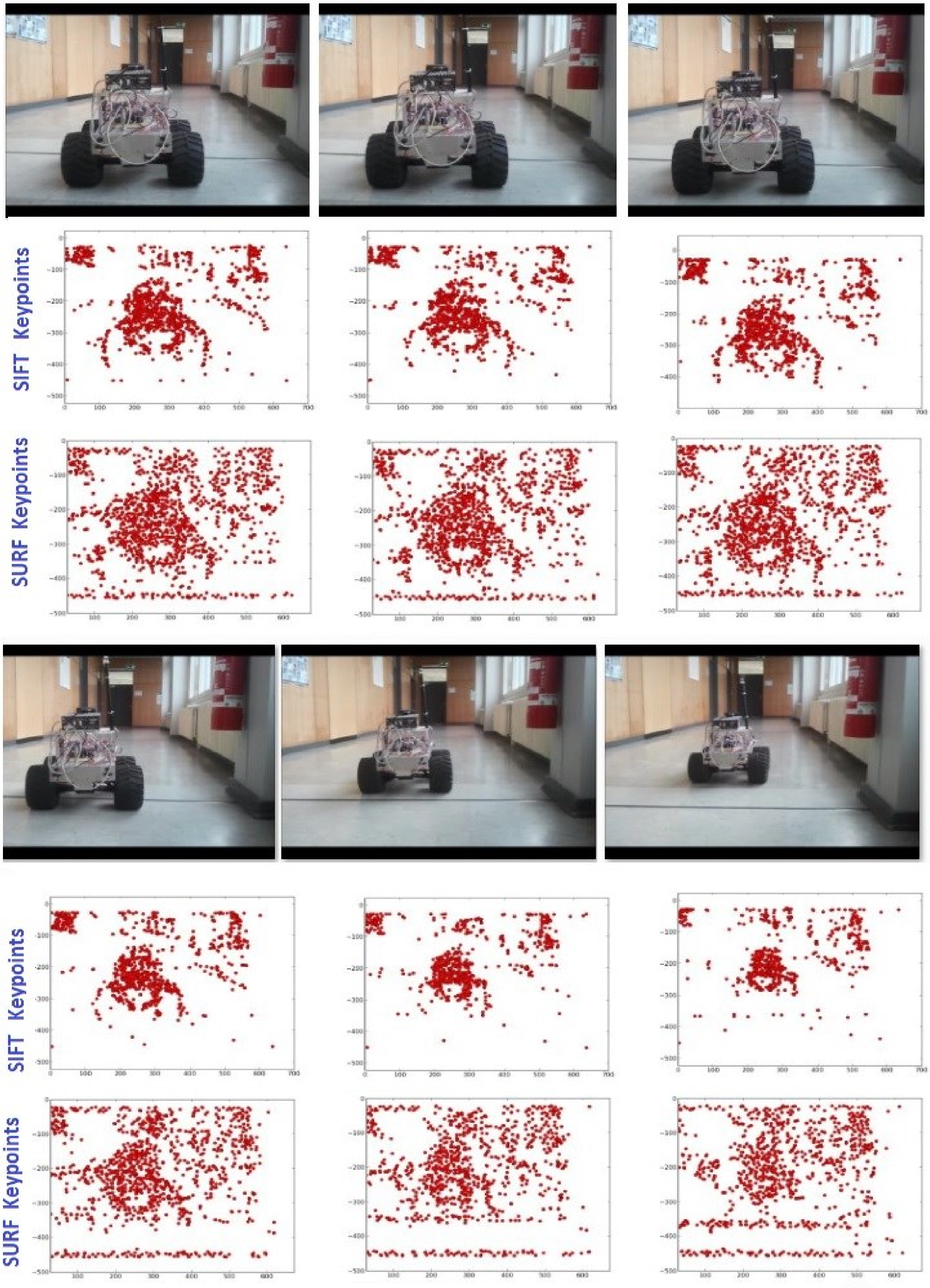


Figure 4. 4 The Comparison sequence of images between SIFT and SURF.

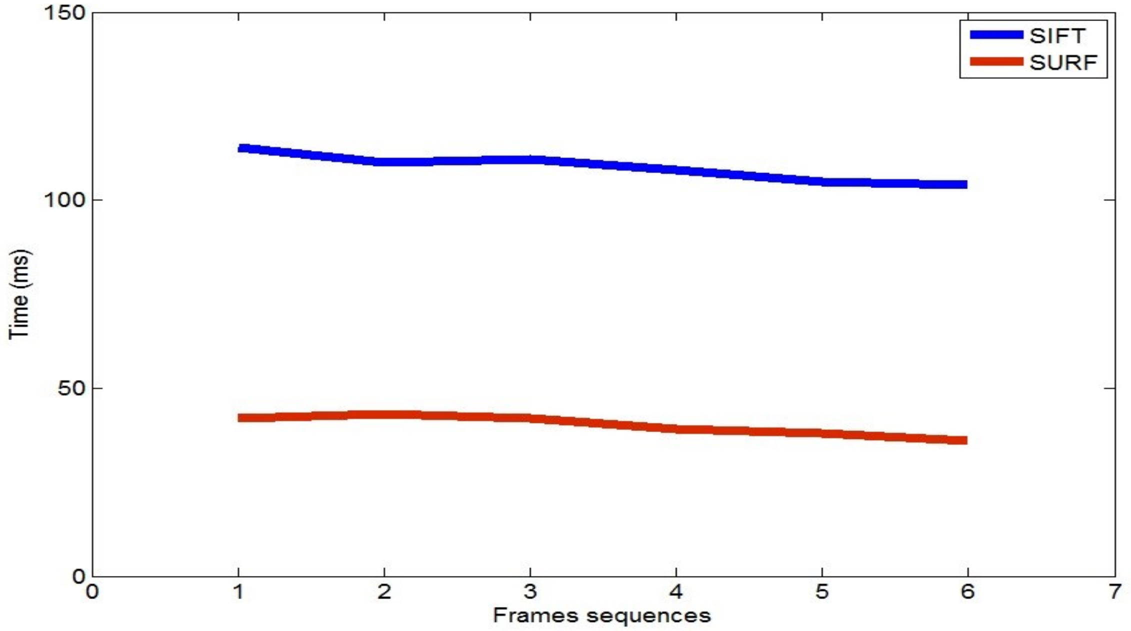


Figure 4. 5 **Comparison time for detect keypoints between SIFT and SURF.**

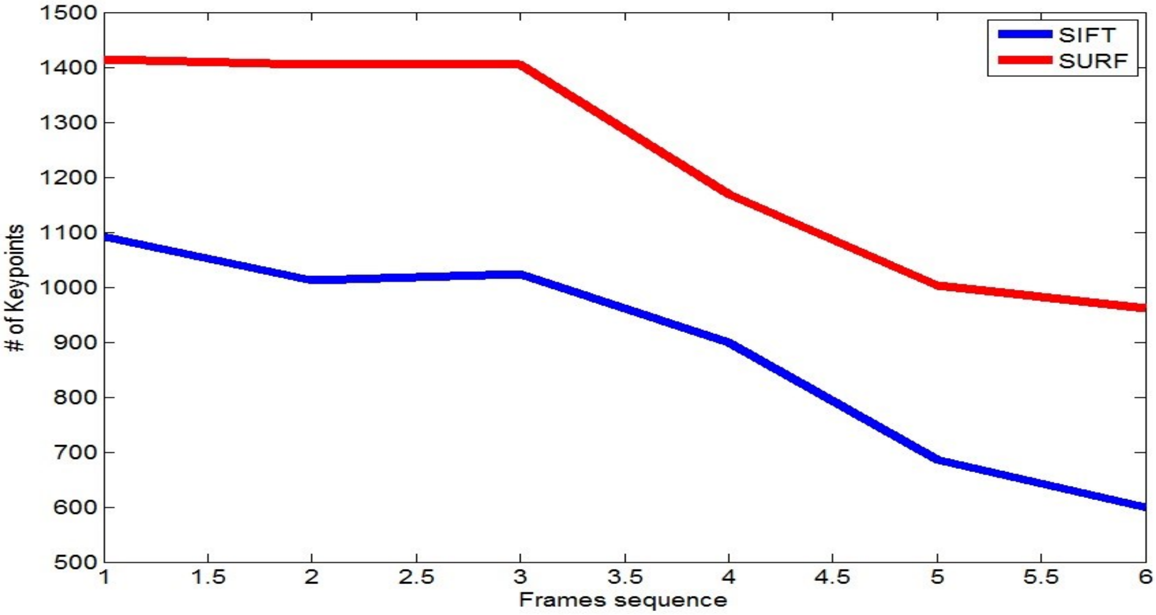


Figure 4. 6 **Comparison the Keypoints number between SIFT and SURF.**

The SURF algorithm has three main steps. The first step is to select the interest points or the keypoints that characterize distinctive regions in the image. Then, to build feature vectors for each interest point based on the window or neighboring pixels. Theses interest points and feature vectors are found and computed for both the captured sequences and reference images. Next and last step is the matching operation with homography which are conducted and implemented to retrieve the interest points. The algorithm make a distance comparison for the feature vectors in both images (reference and captured images). Fig. 4.7 and Fig. 4.8 show the results of SURF feature points matching. These points are SURF points which were detected by the algorithm and the lines are the connected between any two matched points in the figure. If the ratio of the corresponding points in the area has a certain threshold, the set of transformation can be acceptable. Then, the movement parameters could be re-solved through the least square method by using all the corresponding points that were determined by the transformation. If the ratio of corresponding points could not meet the requirements, it is necessary to re-select another set of points and calculate their corresponding motion parameters. In a case that the ratios of the corresponding points are lower than the threshold for all the movement parameters then the two frames cannot match. The SURF algorithm, as it was described before, is summarized in Table 4.1.

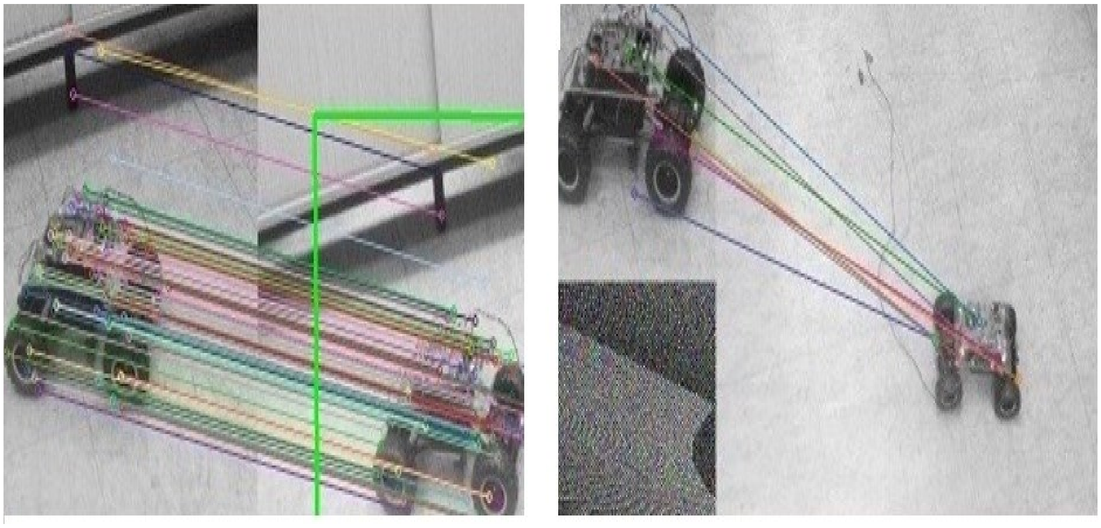


Figure 4. 7 A result of SURF feature points matching.

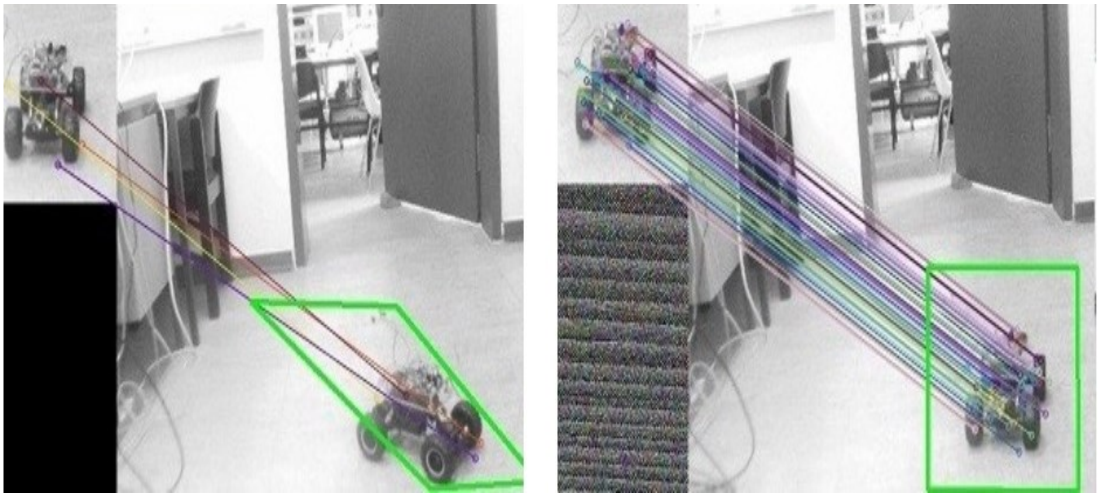


Figure 4. 8 A result of SURF feature points matching.

4.2.3. Robust Matching and Homography

The homography is defined as the relation between any two images of the same plane surface and it is the computation of the camera rotation and transition motion between these two images. A set of matched interest points between two images are frequently used to calculate geometrical transformation models like allowing for transformations and homographies. These points have several errors like the measurement of point position that occurs due to the Gaussian distribution and the outliers to the Gaussian error distribution that are mismatched points.

Inputs: Reference and captured images

❖ **Determine the interest points:**

1. Based on the determinant value of the Hessian Matrix and use of Haar-wavelet.
2. Use of integral images for computation the convolution response of the Haar-wavelet with the image.
3. Integral images speed up the computational time.
4. Compute convolution responses on x and y directions to achieve rotation invariance.

❖ **Construct a distinctive features vector for each interesting point:**

1. The components of the features vector are extracted using a sliding orientation window around the interesting point.
2. Examine different scale for the sliding window to achieve scale invariance.
3. Determine the dominant orientation.

❖ **Match the features vectors of the image against those of the reference image:**

1. Use the metric to compute and find the distance between these feature vectors. (Euclidean distance is used).
2. We consider that two feature vectors matches if the distance is less than 0.7 times compare to the second nearest neighbor.
3. Store interest points of the target images that have been found a match in the reference image.

❖ **Localize the robot:**

1. Based on the locations of the store interest points that were matched.

Output: The location of the target robot

Table 4. 1 SURF algorithm.

In statistics, the outliers are defined as the observations that are numerically differ and distant from the data [177]. These errors should occur randomly in any distribution, but they are often indicative either of measurement error. Then, it is needed a robust estimator that can be able to cope with these outliers and capable to filter the total set of matched points to find, detect and then delete the uncorrected matches. The Random Sample Consensus **RANSAC** algorithm might be used and implemented on SURF to discard and eliminate the outliers from the matched points [178]. It has ability to deal with large proportion of outliers. The algorithm starts by selecting a random subset of the original data points to obtain the model. This could be done by using a closed form solution according to the desired projective transformation. The previous step is repeated for fixed number of times, in each time will produce a homography model. When the maximum trials is reached, the robust estimation algorithm returns the projection model with largest number of inliers. This method is employed by the robust homography estimation matrix H. Assume a 2D projective transformation with set points x_i

that is the features in the reference image and a matches set points x_i' in the target image, compute the 3×3 homography estimation matrix $\mathbf{H}$ as given by:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{31} \\ h_{21} & h_{22} & h_{32} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad 4.10$$

$$x_i' = H.x_i \qquad 4.11$$

where $(x, y, 1)^T$ are the homogenous coordinates of the features in the reference image, $(x', y', 1)^T$ are the homogenous coordinates of the features for the target image, and H is robust homography estimation matrix. Since each point to point correspondences give cause to independent equations in **H**, the minimal solution needs four correspondences. The **RANSAC** algorithm is described in Table 4.2 and it shows the general steps to obtain a robust transformation.

```
Giving: matched points set $\mathbf{x_i = (x_i, y_i) \leftrightarrow x_i' = (x_i', y_i') for\ i = 1 \ldots n}$
   Define s = minimum points set.
   Define p = probability of free outliers.
   Define t = distance threshold.
   Define ε = initial probability.
   Define Consensus = number of minimum inliers.
   maximum number of samples $N = log(1 - p)/log(1 - (1 - \varepsilon)^s)$
While $\mathbf{N} > \boldsymbol{Trials}$ do
  Randomly select s pairs of matched points.
  Calculate the minimal solution for the model.
  Inliers =0
  for i=0 to n do
      $d_t^2 = d(x_i', H_{x_i})^2 + d(x_i, H^{-1}x_i')^2$
      if $\mathbf{d_t} < \boldsymbol{t}$ then
     inliers = inliers+1
     end if
 end for
 if  inliers > Consensus then
     calculate final projective transformation
     Consensus = inliers
 end if
 recalculate $\mathbf{\varepsilon = 1 - (inliers/n)}$
 recalculate N
 Trials = Trials +1
end while
```

Table 4. 2 RANSAC algorithm [178].

4.2.4. Inverse Perspective Mapping

In fact, the perspective effect must be taken into account when processing images in order to weight each pixel according to its information content. Because of this perspective effect, the captured images have some distortion. In order to deal with such problem the Inverse Perspective Mapping (IPM) has been introduced [179]. It can help to eliminate the perspective effects from the image, then remap the image into new **2D** domain in which the data among all pixels are distributed in homogeneously manner. This technique gives ability to an undistorted top-down view which is known as the bird's eye view as it is shown in Fig. 4.9. The top-left side shows the **SURF** algorithm detects a ground robot, whilst the top right side illustrates the **SURF+IPM**. The down-left side and down right side show the **SURF** and **SURF+IPM** detects 3 DOF Helicopter system, respectively.

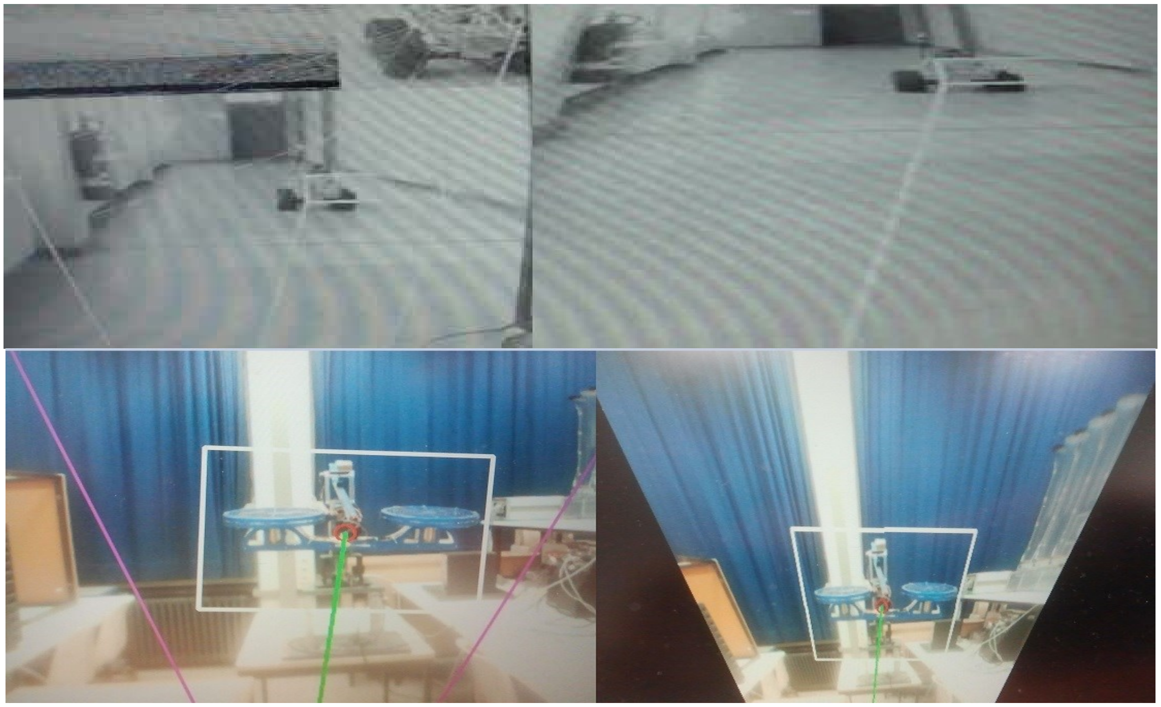


Figure 4. 9 **IPM wrap results.**

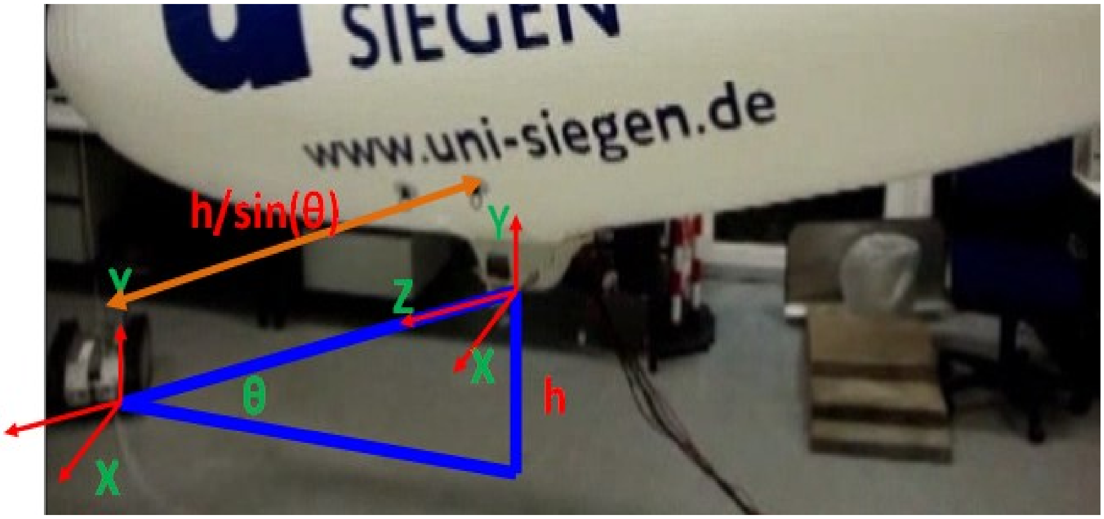


Figure 4. 10 **Camera Coordinate with respect to target.**

To better understanding this technique, let us consider the world frame **(X, Y, Z)** and the projection on the image plane is **(u, v)**. Then, the rotation will be around the x-axes and the translation on the z-axes as it is shown in Fig. 4.10. Thus, the mapping can be expressed as the following.

$$[u, v, 1]^T = KTR(x, y, z, 1)^T \qquad 4.12$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 4.13$$

$$T_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h/\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 4.14$$

$$K = \begin{bmatrix} a_x & s & u_o & 0 \\ 0 & a_y & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad 4.15$$

where **R, T, K** are the rotation, translation, and camera parameters matrices, respectively. **h** is the height of the camera, **α_x= f*m_x, α_y= f*m_y**, Here **f** is the camera focal length, **m_x, m_y** are the scale factors, **s** is skew coefficient between the x and y axis (often it is zero), **u_o, v_o** are the principal points. According to [180], the projection Matrix **P** is given by:

$$P = KTR = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{bmatrix} \qquad 4.16$$

So the Eq. 4.12 which used to map each pixel on the image plane to a top-down view can be re-written as:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \qquad 4.17$$

In order to solve Eq. 4.17 the OpenCV provides **GetTransformationMatrix()** function which can be used to find the transformation matrix and to save the results in a lookup table as it is described in **Appendix F**. We now have a position of the object in the transformed image in pixels. In order to convert it into meters, the automatic calibration of the bird's eye view projection system can be used [181]. We need our camera intrinsics and extrinsic matrices which can be obtained from the calibration method. The chessboard or any object can be placed on the ground floor to obtain a ground plane projection image. Then, remap the image into new domain which is called the bird or top eye view. The algorithm and procedure of the automatic calibration is shown in the Table 4.3. Moreover, this estimation is valid because of the linearity strength of the **IPM** algorithm. Hence, the value changes linearity as the real world distance changes, e.g. if r = 70 pixels and D= 1m, thus, when r becomes = 35 pixels leads that D= 50cm**.** This algorithms and its steps in details can be found in Appendix F.

Algorithm 3 : bird's eye view

A. Place the camera in front of object in a fixed distance a long flat floor.
B. Read the intrinsic and distortion models for the camera.
C. Find a known object on the image by 4 points.
D. Enter the found points into cvWarpPerspective(), perspective_matrix, GetPerspectiveTransform() to compute the homography matrix H.
E. This method will give us the r (distance in pixels) and we already know that the objectis far for fixed distance D.
F. Then we can find {every x number of pixels is = number in meters).
G. Do (A-F) many times and different distances and make average for the scale Factor M.
H. Since the homography matrix, the height parameter set and M have been found, we draw line from center of the object to the projection point of the camera. This line has float value changing as the program is running.
I. Find r in pixels, then, convert it to D= M*r in metric.

Table 4. 3 Bird's eye view algorithm.

Finally, the sequence of images for detecting an object by implemented **SURF** and **IPM** are shown in Fig. 4.11 and Fig. 4.12. Fig. 4.11 shows the algorithm detects a moving object and converts the image to bird's eye view in order to estimate the distances between the vision sensor and the target. In addition, Fig. 4.12 shows the algorithm detects the ground robot target and it has the ability to distinguish between the target and any other ground robot in the environment. The algorithm could detect the target for different distances and angles with high performance.

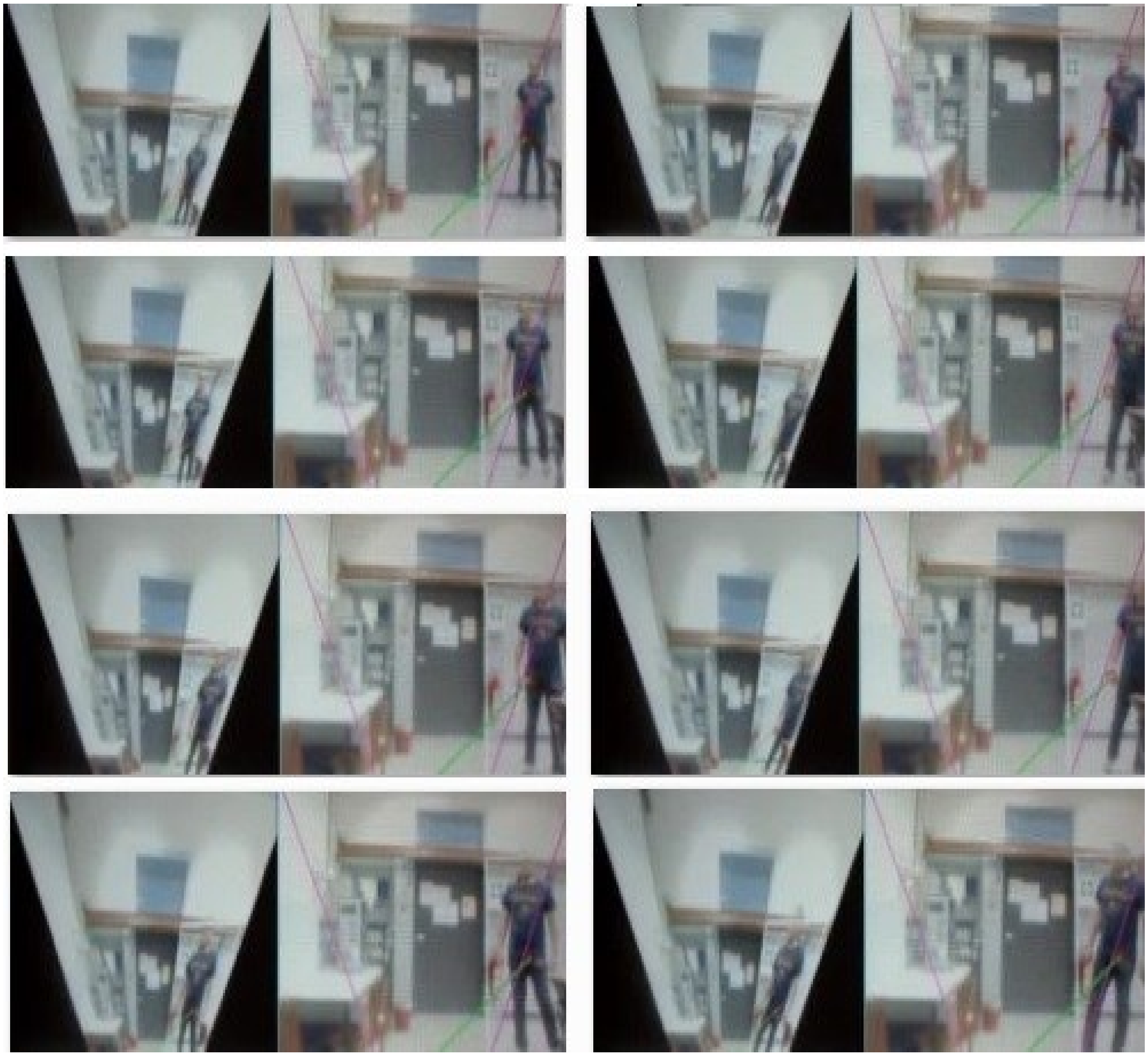


Figure 4. 11 IPM sequence of images 1.

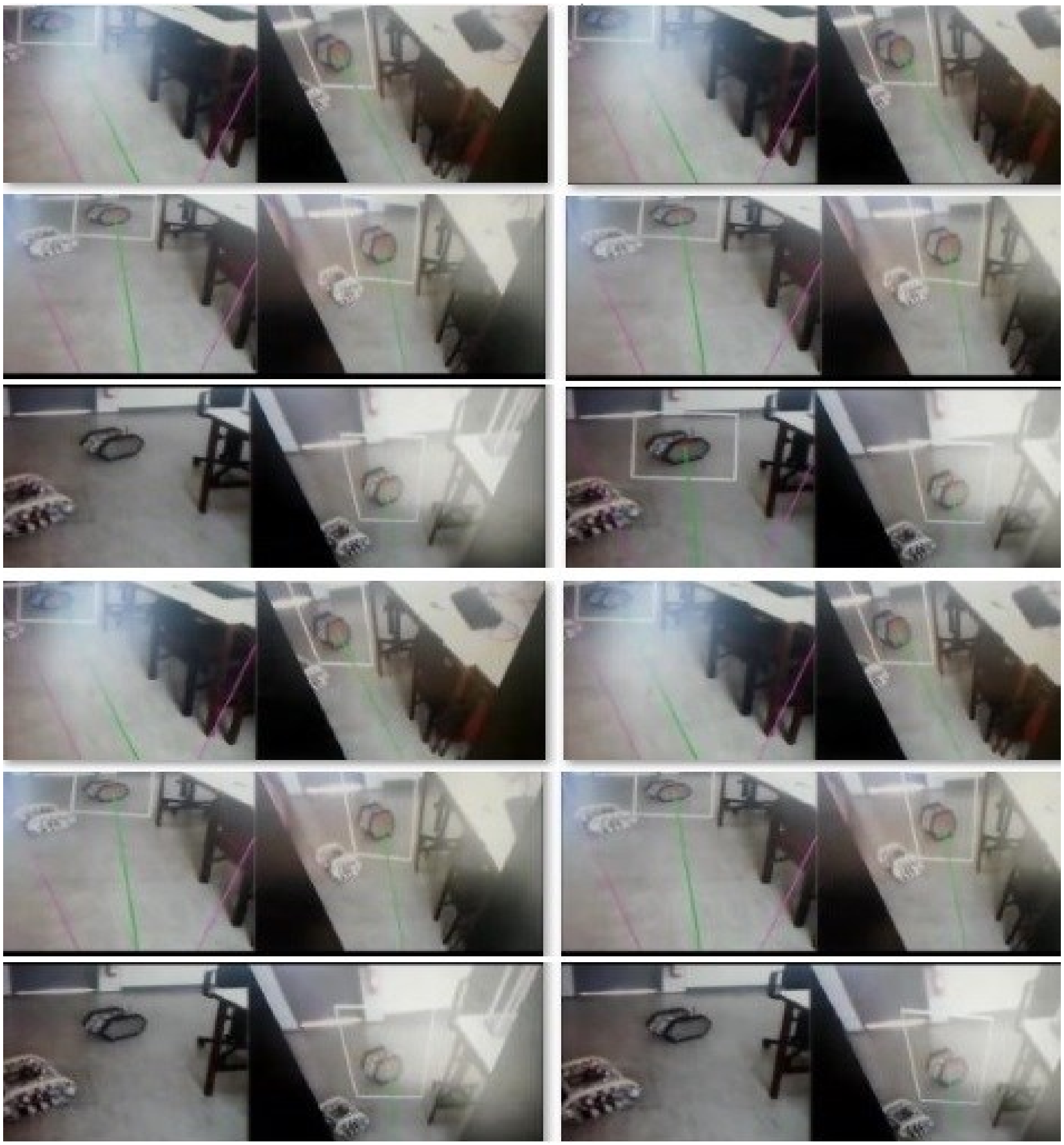


Figure 4. 12 IPM sequence of images 2.

4.2.5. Fuzzy Image Processing

The Fuzzy Image Processing **(FIP)** is defined as the collection of all approaches that can process the images with their segments values as fuzzy sets based on Integral Images [182]. This technique has three main steps: fuzzification, fuzzy inference system {that contains the membership values and rules}, and defuzzification as it is illustrating in Fig. 4. 13.

Moreover, this process (fuzzy image processing) is based on image analysis rather than on the input values as it is in the normal fuzzy logic. More precisely, before applying the fuzzy logic on any image we need to convert it or transformed it to another domain which is called perception domain [182]. The characteristics of the images will be described as linguistic variables and map them to membership functions. Thus; the images will be transferred from gray level plane to membership plane in the fuzzification process. Then, the fuzzy inference system will map the input to fuzzy output by using fuzzy rules based knowledge that is expressed through fuzzy operators. It is important to mention here that this step is very important if we are seeking to modify the membership values as it has been described in details in Chapter three. In addition, some fuzzy techniques can do the modification process such as fuzzy clustering, fuzzy rule-based approach and fuzzy integration approach. Finally, the last step is the defuzzification in which the crisp output will be calculated based on center of gravity method. Hence, the fuzzy operator generates the useful data and maps the fuzzy values to visualize them.

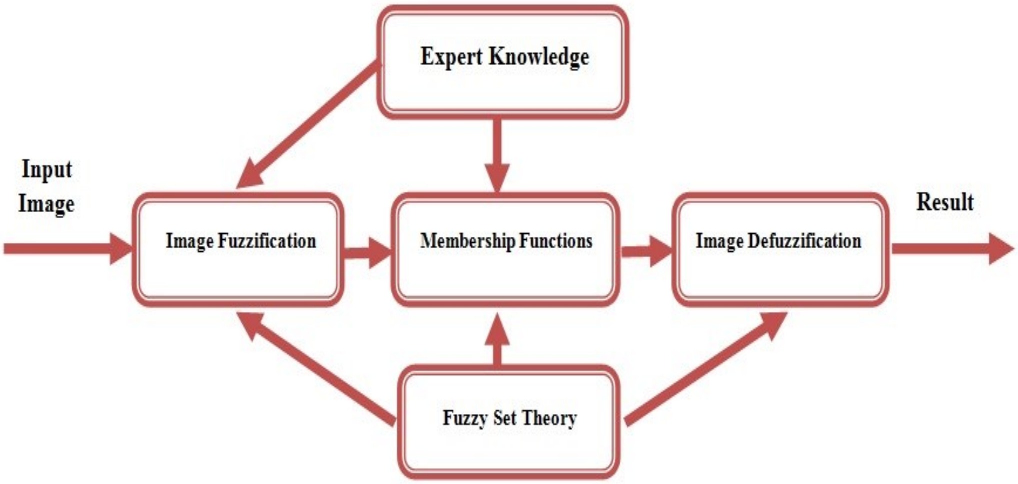


Figure 4. 13 The general structure of fuzzy image processing.

4.2.5.1. Applications of Fuzzy Image Processing

Because the fuzzy logic represents a good mathematical framework regards to deal with uncertainty in any information, it becomes widely used in image processing to manipulate uncertainty in image brightness, edges and other features. There are many applications that can gain the benefits from using fuzzy image processing such as agriculture, medicine and robotics. The combination between fuzzy image processing with genetic algorithm or neural networks can produce more robust results. Additionally, they can be used widely in path planning purposes and navigation for autonomous ground robots. In addition, it could be implemented in object recognition in soccer robots applications in order to detect the ball and determine the its location with respect to the robot position.

4.2.5.2. Fuzzy Edge Detection

Edge detection is one of the most important algorithms in vision system and image processing and plays an important role in the higher level processing [183]. The edge is usually defined as the object border [184]. There are several algorithms for edge detection that are implemented to find and to detect this edge such as Sobel, Prewitt and Robert [185]. Because threshold value is usually empirically found, the algorithm has a certain possibility to lose few edges. The Canny algorithm which successfully solves and optimizes the edge detection problem [186] also has some limitations such as the detection of edge pixels might be inaccurate. On other words, the changes in the threshold values might increase the edge detection rate, but unfortunately the accuracy might decrease. Therefore, because of these limitation beside the noise, the Canny method that might be not able to give an efficient and satisfactory results [187]. Therefore, several studies and researches have been implemented to use fuzzy logic in edge detection, but all of them as it comes to our knowledge are simulations algorithms [188-195]. However, in this research, the fuzzy image processing was implemented on on-baord blimp for real time applications. The task starts detecting the edge points of the image by taking every pixel with its eight neighbors as it is shown in Fig. 4.14. This whole structure is tuned to function as a contrast enhancing filter to segment the taken images. The filter slides over the image row by row (**i** is the row and **j** is the column indices) as it is shown in Fig. 4.15. Then, each pixel is processed with fuzzy rules base in order to consider the pixel as edge or not (under consideration as **black B, white W or edge E**) as it is illustrating in Fig. 4.16. The collections of all these edges points are taken as the final edges of the image. The whole image processing algorithm is shown in Fig. 4.17a, Fig 4.17b, and Fig. 4.17c. These figures show the main steps of the fuzzy edge detection. It starts by converting the taken image to gray scale level. Then, it finds the edges on the image. Finally, there are two sub-steps in order to accomplish the whole process. First step defines a set of contours to address the target robot based on the shape and color. Then, the similar procedure has been done to define the obstacle which is depending only on the square features. We defined a set of contours to address the obstacle based on square features as it is shown in Fig. 4.17c. The fuzzy edge detection algorithm has nine inputs with two linguistic variables for each (**W: White, B: Black**) and one output which has three linguistic variables (**W: White , B: Black, E: Edge**) as it is illustrating in Fig. 4.18 and Fig. 4.19. Detailed information about this work could be found in [98].

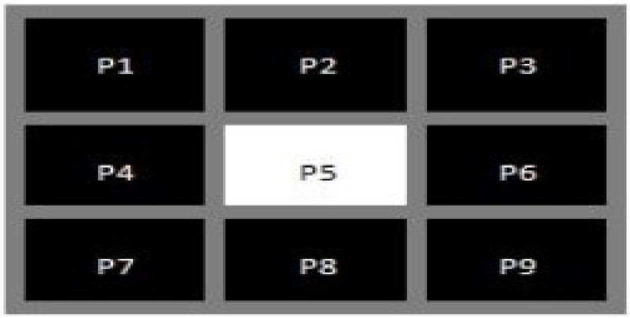


Figure 4. 14 **The 3*3 fuzzy filter mask.**

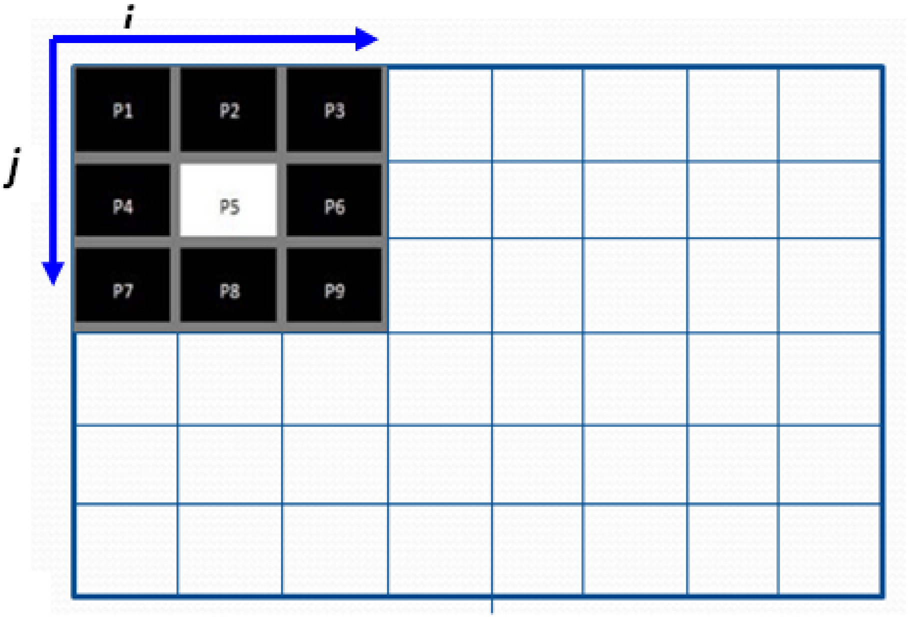


Figure 4. 15 **Slide the mask on each pixel row by row.**

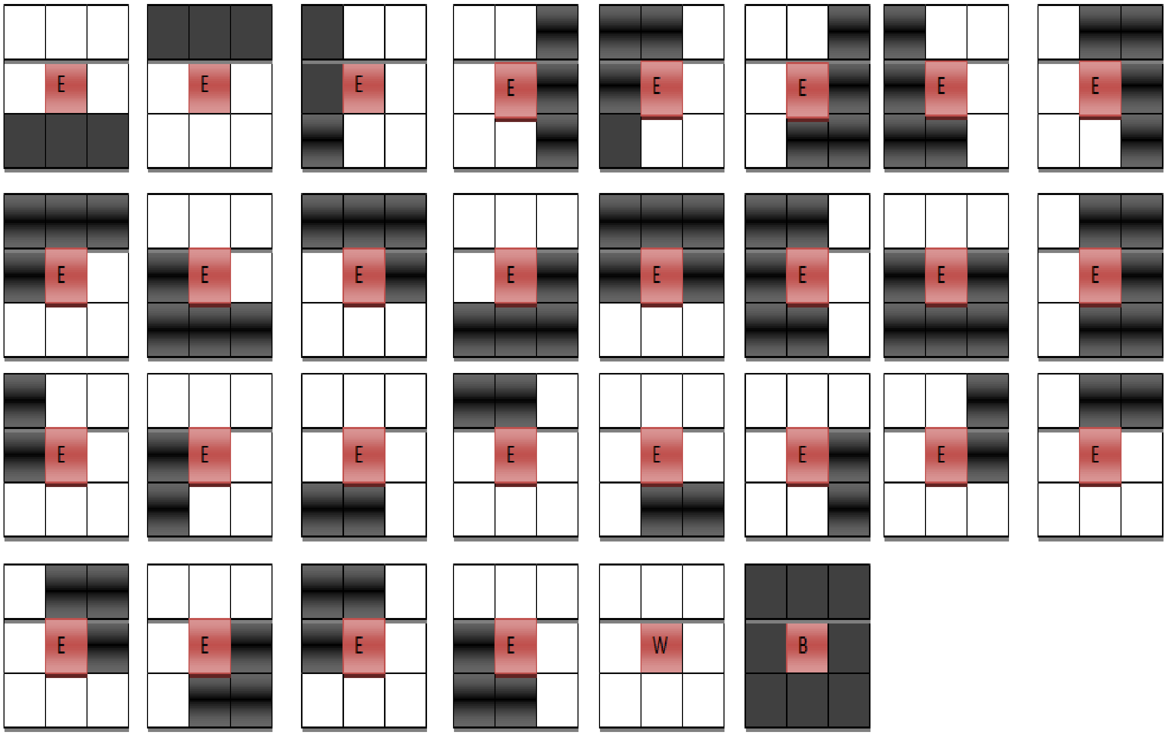
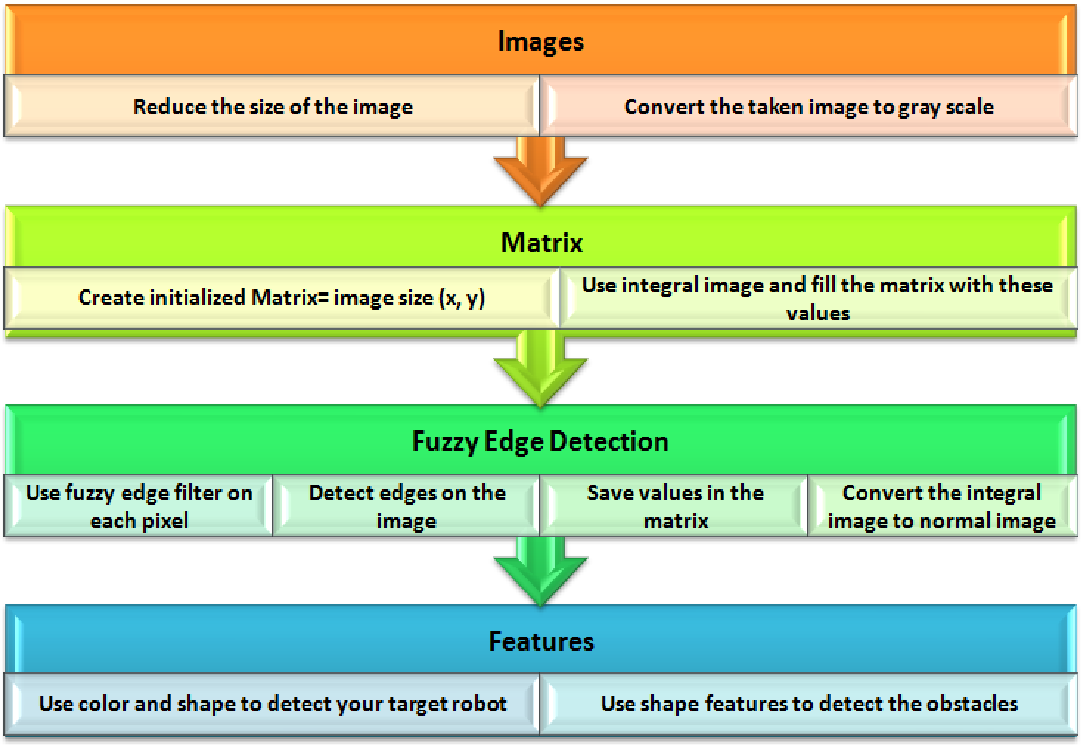


Figure 4. 16 Fuzzy Rules Matrix.



Images

Reduce the size of the image

Convert the taken image to gray scale

Matrix

Create initialized Matrix= image size (x, y)

Use integral image and fill the matrix with these values

Fuzzy Edge Detection

Use fuzzy edge filter on each pixel

Detect edges on the image

Save values in the matrix

Convert the integral image to normal image

Features

Use color and shape to detect your target robot

Use shape features to detect the obstacles

Figure 4. 17a Image processing algorithm steps.

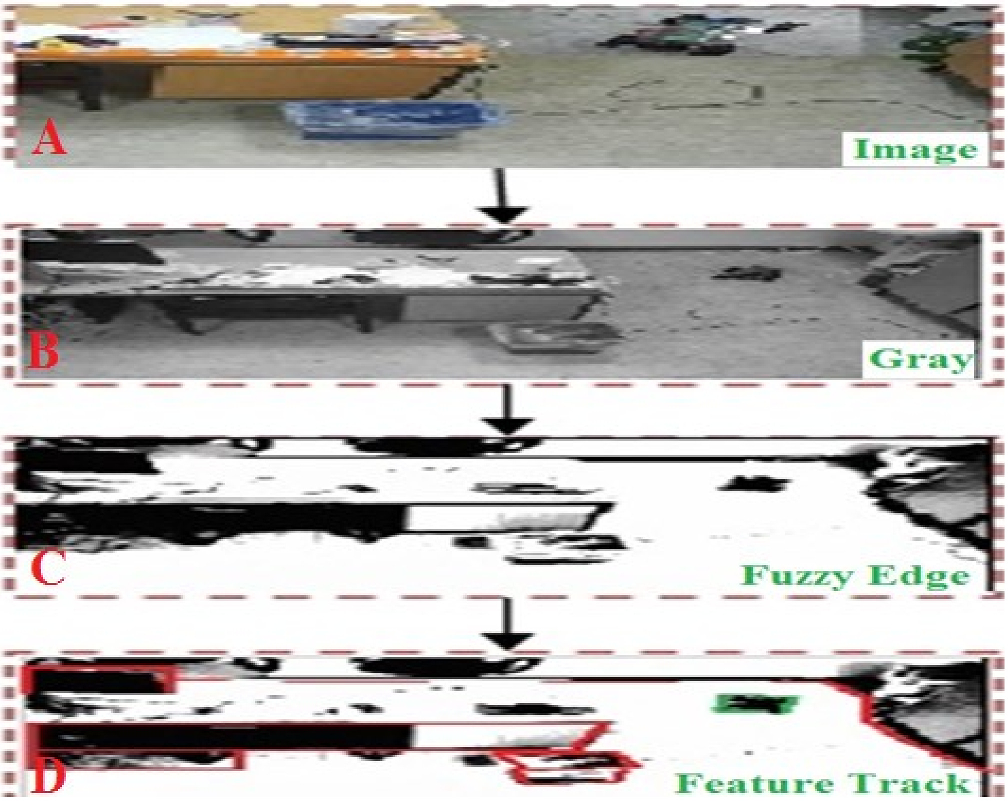


Figure 4. 18b The result of Image processing algorithm .

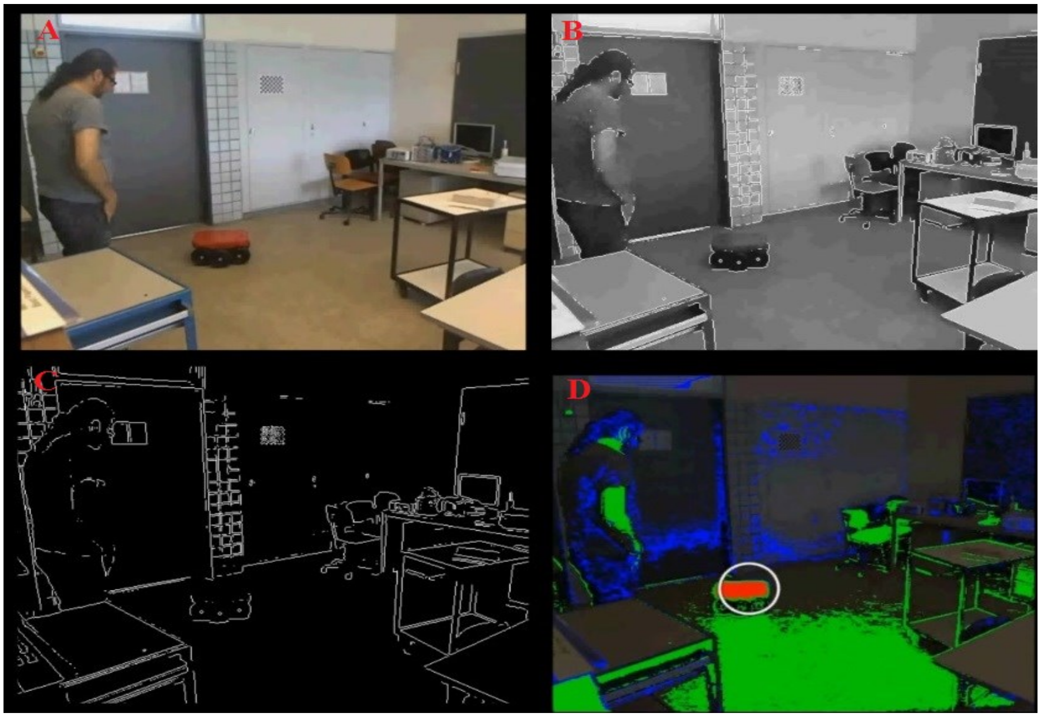


Figure 4. 19c The result of Image processing algorithm.

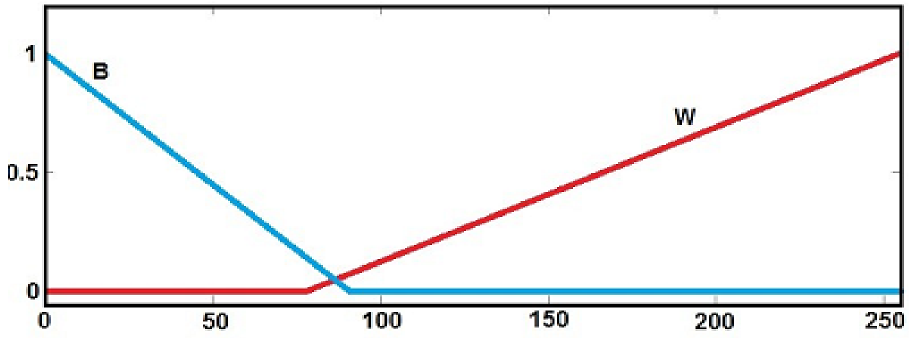


Figure 4. 20 The membership functions for Inputs.

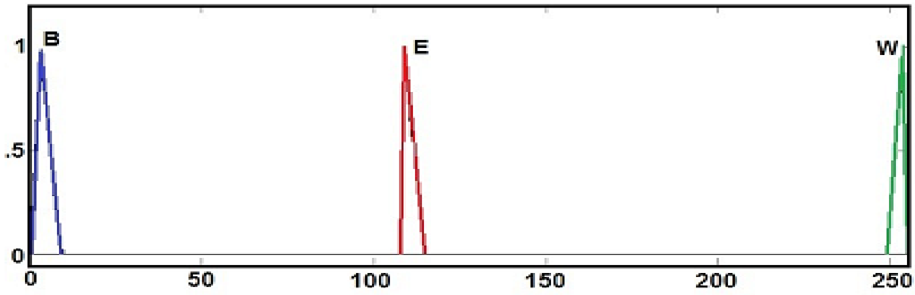


Figure 4. 21 Output Membership functions.

4.2.5.3. Fuzzy Clustering

The clustering in general is defined as a method of grouping objects into clusters where objects in same cluster are similar to each other. This method might be achieved by different algorithms and could be used widely in several applications such as pattern recognition and image analysis [197-200]. There are two types of clustering methods; namely a soft clustering and hard clustering. However, choosing of the clustering method and preferring one cluster over another is somehow difficult or unpredictable since these methods are basically based on imperial results or mathematical reasons. In hard clustering each of these objects belong to a certain cluster or not (1 or 0), but in the soft clustering or which is known as fuzzy clustering [201] each object has a certain degree of membership function which means that objects might belong to another cluster with a certain value belongs to interval [0,1]. Therefore, the segmentation which might be obtained by implementing the clustering will play a significant role for the robot vision system to partition the image into set of disjoint regions with uniform and homogeneous attributes. The fuzzy clustering method is one of the key factors for image segmentation and it has been successfully implemented [202,203]. Among all of them, the fuzzy c-means algorithm is the most popular one that is used in image segmentation [204]. It is not only more robust for ambiguity, but also able to identify the objects effectively [204]. The algorithm starts producing optimal partition **c** by minimizing the weighted within group sum of squared error objective function $\mathbf{J_{FCM}}$ [205]:

$$J_{FCM} = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^q d^2(x_k, v_i) \qquad 4.18$$

where ***X= {x₁, x₂, x₃,…,xₙ}*** $\subseteq \mathrm{R}^{\mathrm{p}}$ is the data set in the ***p*** vector, ***q*** is a weighting exponent***,*** $\boldsymbol{u_{ik}}$ is the degree of membership of $\boldsymbol{x_k}$ in the $\boldsymbol{i^{th}}$ cluster***, n*** is items number, ***c*** is the clusters, ***d*** is the distance between object $\boldsymbol{x_k}$ and cluster $\boldsymbol{v_i}$ ***,*** $\boldsymbol{v_i}$ is the prototype cluster center ***i***. The algorithm to solve the ***J*** is shown in Table 4.4. The results for implemented fuzzy c-means onto images are illustrating in Fig. 4.20 and Fig. 4.21. In these figures it is clear that the algorithm could make the segmentation and edges on the images easily with pretty good results. However, the main problem we have found with fuzzy-c means is that it tends to run slow since it's actually doing more work. Each point is evaluated with more operations as it needs to do a full inverse-distance weighting.

```
Start       Set Values for $\delta$, c and q .
            Initialize fuzzy matrix U = [$u_{ik}$].
            Set loop counter m=0.

Calculate the c cluster center $\{v_i^m\}$ with $U^m$
            $v_i^m = \frac{\sum_{k=1}^{n}(u_{ik}^m)^q x_k}{\sum_{k=1}^{n}(u_{ik}^m)^q}$
Calculate $I_k = \{i|1 \le i \le c, d_{ik} = \|x_k - v_i\| = 0\}$
Calculate the membership function $U^{b+1}$.
IF  $I_k = \Phi$
            $u_{ik}^{m+1} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{q-1}}}$
ELSE   $u_{ik}^{m+1} = 0$
K=k+1
END
          IF $\|U^m - U^{m+1}\| < \delta$
              Stop
          Else m=m+1
END
```

Table 4. 4 **Fuzzy c-means algorithm [181].**

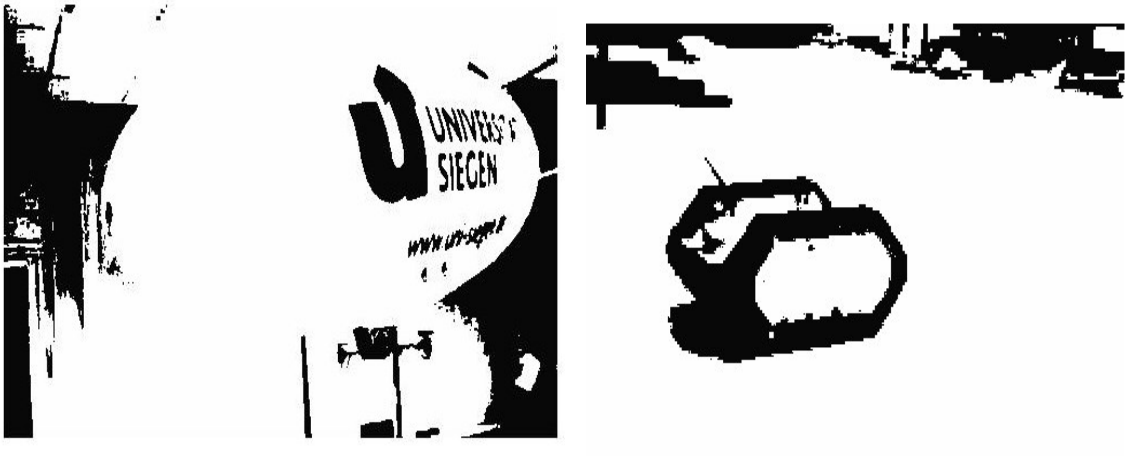


Figure 4. 22 **Results for the Fuzzy C-Means algorithm.**

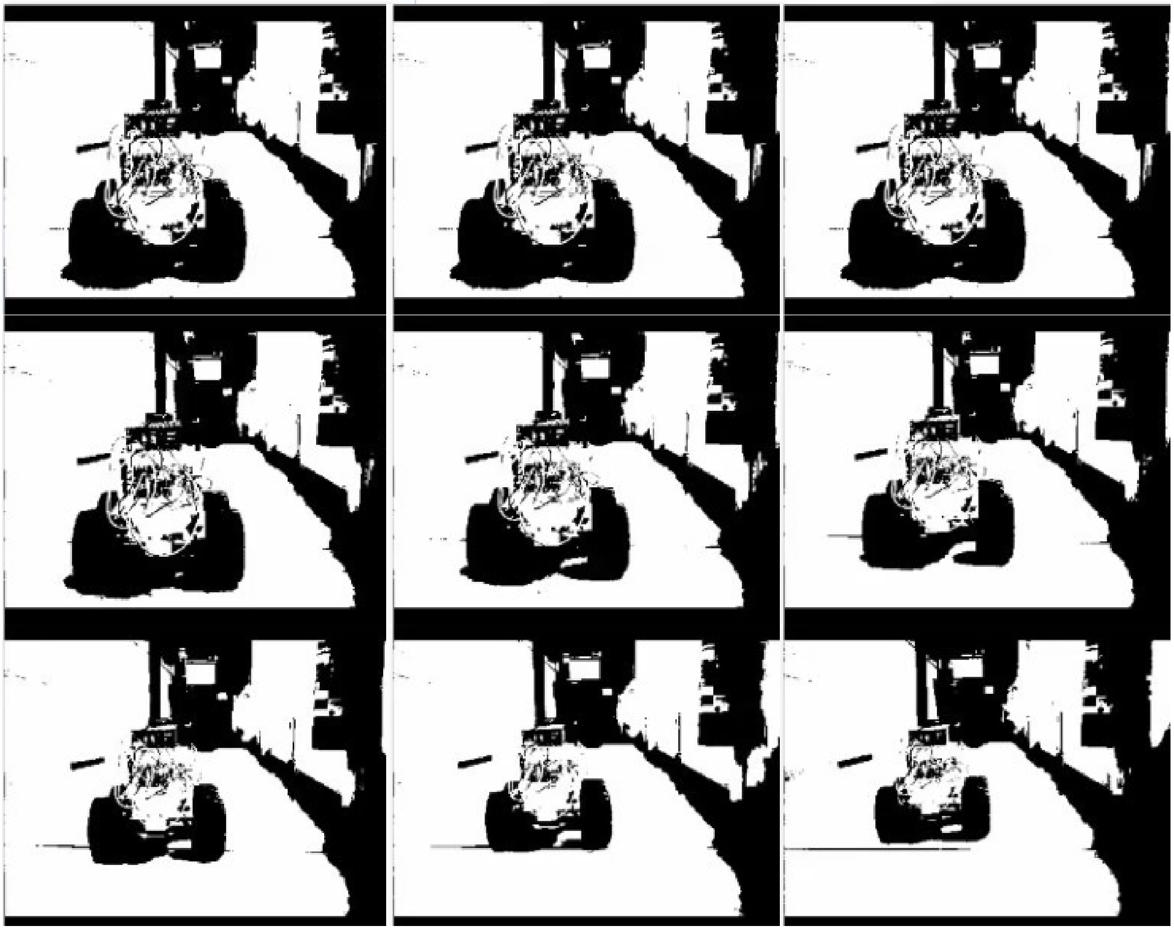


Figure 4. 23 **Results for the Fuzzy C-Means algorithm.**

4.3. Blimp's Embedded Computer Vision

In this section, we present the computer vision methods which have been used in order to make the blimp robot able to follow a ground robot, help the ground robot to navigate and explore an unknown environment and give the blimp ability to follow 3D aerial objects [206-209].

4.3.1. Detect and Follow Ground Robot

In order to develop a robust embedded visual system to follow a ground robot target by the indoor blimp robot, the previous approaches have been studied by considering some important criteria such as performance, repeatability, accuracy and speed. It is known that a higher resolution of the image should therefore only be wasted, since the image would have to be downscaled anyway. Thus, after many experiments it has been found that the execution or running time depends on the size of the image. Therefore, it was found that 168x264 pixels is the best choice because it is more accurate and fast as shown in Fig. 4.22. Then, the algorithm localize the target and find the position of the target center projection point in (x, y) axis. The position could be normalized to inputs **(x_n, y_n)** and given as the following:

$$x_n = \left\{\begin{matrix} \dfrac{x-132}{131} & ,x \geq 132 \\ -\dfrac{132-x}{132} & ,x < 132 \end{matrix}\right\} \qquad 4.19$$

$$y_n = \left\{\begin{matrix} \dfrac{y-84}{83} & ,y \geq 84 \\ -\dfrac{84-y}{84} & ,y < 84 \end{matrix}\right\} \qquad 4.20$$

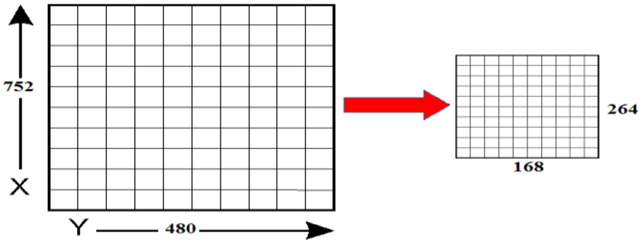


Figure 4. 24 **The image size.**

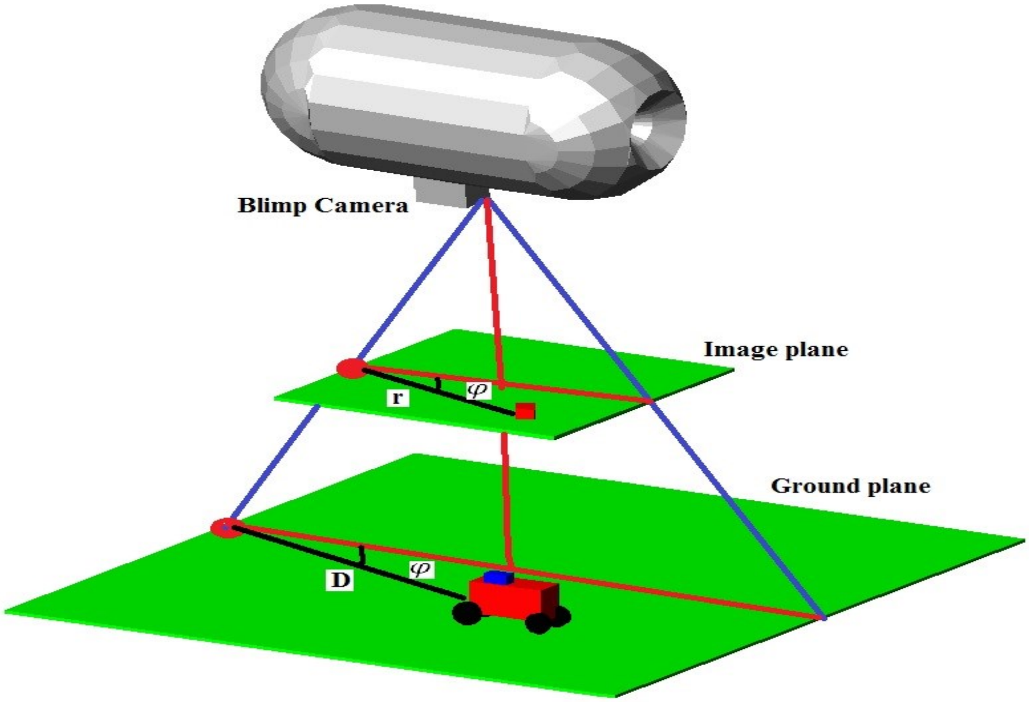


Figure 4. 25 **Coordinate of the ground robot related to Camera**

The coordinate of the ground robot related to the camera position is shown in Fig. 4.23. As it is shown, the target robot projection point in the image plane and the projection of the camera point are determined in order to find the related position of the target with respect to the blimp. After the SURF algorithm detects the target, a corresponding box was drawn around the keypoints (interest points). Then, a red circle was drawn on the center of the target object. The target position is defined related to the blimp position in the image plane and this is not a global localization. Then, target distance and orientation are converted to polar coordinate (r, ϕ) as it is shown in Fig. 4.24, the polar coordinate between the blimp projection point and the center of the target is defined as the following:

$$\varphi = \tan^{-1}\left(\frac{x_n}{y_n}\right) \qquad 4.21$$

$$r = \sqrt{x^2 + y^2} \qquad 4.22$$

Where **r** is the distance between the target centre projection point and blimp. ϕ is the normalized angle between the target centre and blimp's path projection. d is the distance obtained by implemented **IPM** algorithm.

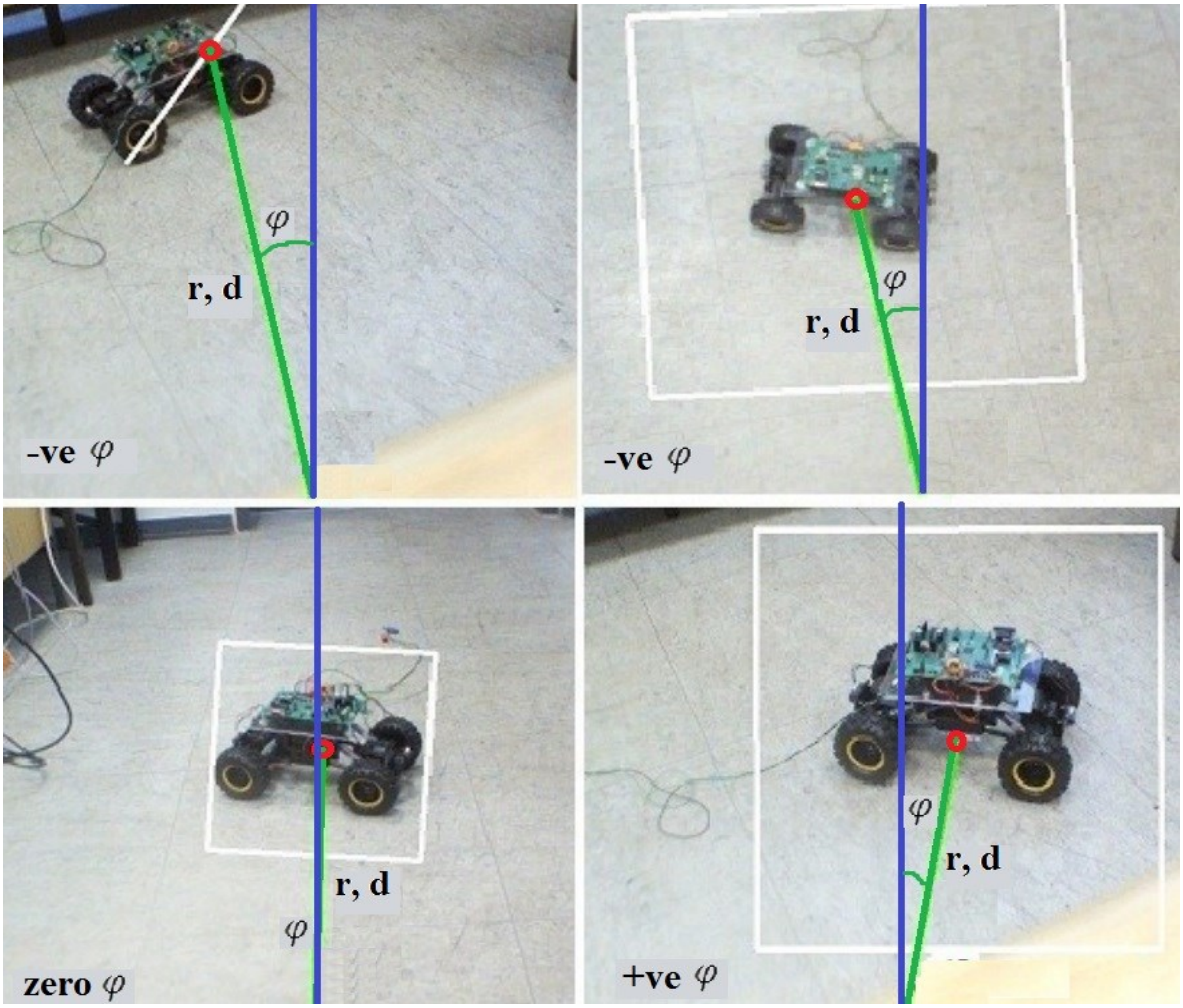


Figure 4. 26 **Target location Information.**

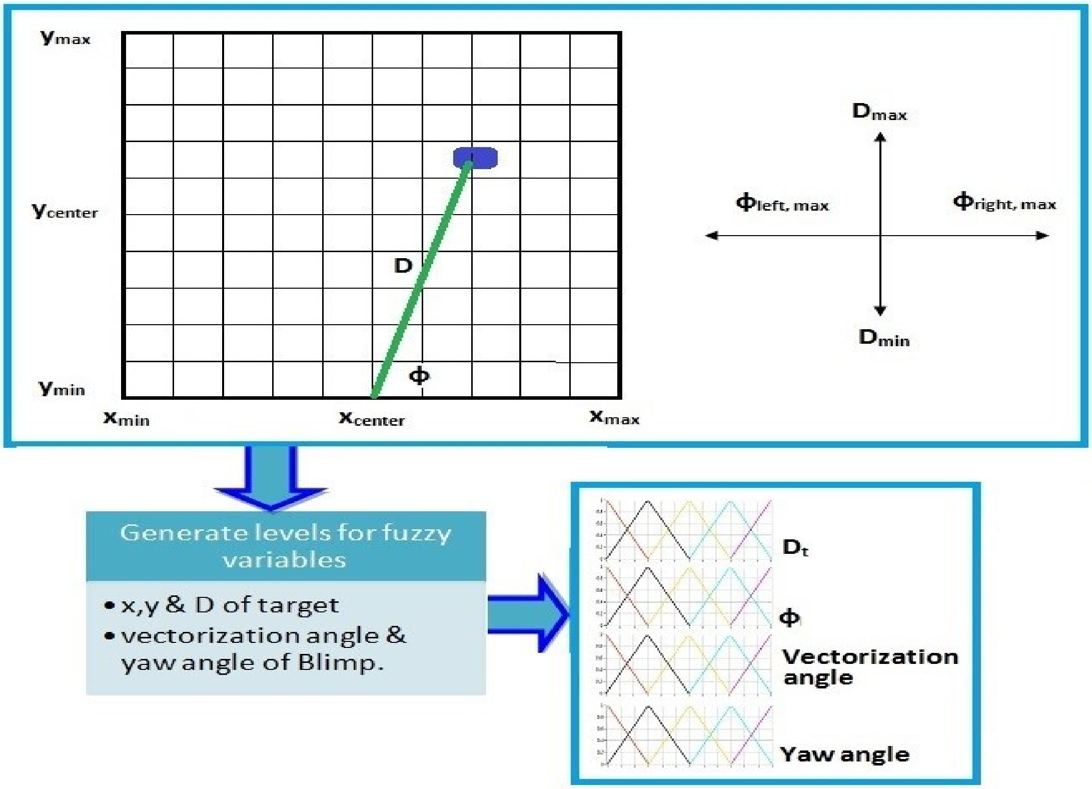


Figure 4. 27 The Servoing algorithm.

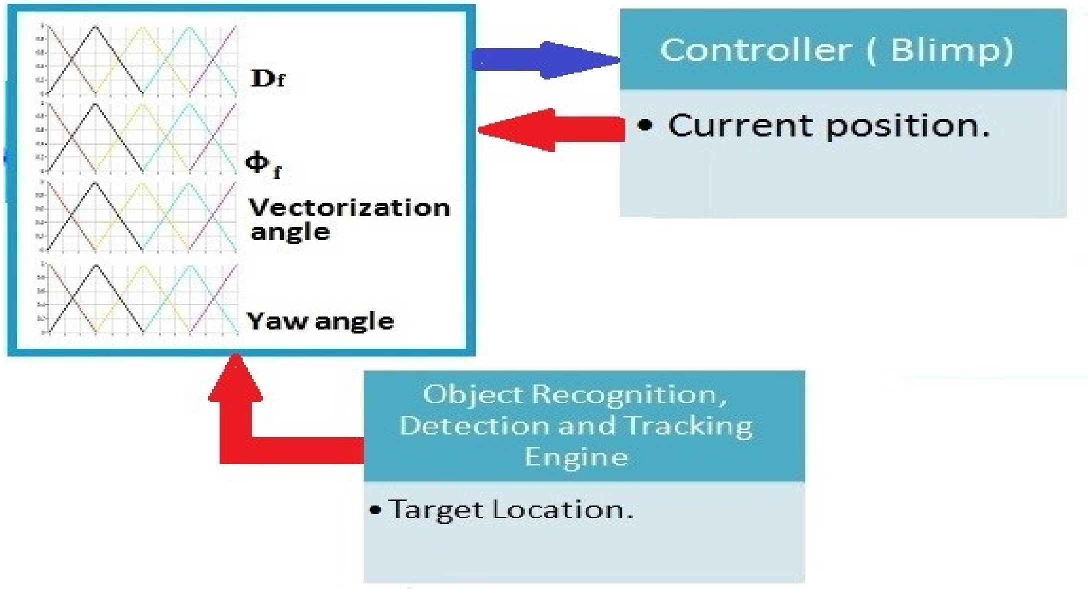


Figure 4. 28 The overall visual fuzzy Servoing algorithm.

The visual fuzzy servoing algorithm is illustrating in Fig. 4.25 and Fig. 4.26. The object recognition, detection and tracking engine will detect and localize the target with respect to blimp position; then a fuzzy sets model will correct the prediction vision information (**D_t, ϕ**). Then the final information (**D_f, $ϕ_f$**) will feedback the blimp controllers in order to control the vectorization and yaw angles. Note that we can use **D** which obtained from **IPM** or **r** that is obtained from **SURF** directly. The rules and fuzzy sets for this servoing control could be found in Appendix D.

In order to verify the proposed vision system, some experiments of the complete system were conducted. During all of these tests and experiments the blimp robot was flying at an known and a certain height and altitude. We assume that the target is already in the view of the on-board camera. The target could be detected, identified and tracked by the vision system. Based on the vision data, the vectorization angle for the main propellers and the yaw angle were controlled to follow the target and keep it in a certain position in the image. Fig. 4.28 and Fig. 4.30 show the estimated distances between target projection point and blimp in two experiments. We observe that blimp can track target and keep it at a certain distance. The estimated angle between target projection point and image's centre results are shown in Fig. 4.27 and Fig. 4.29.

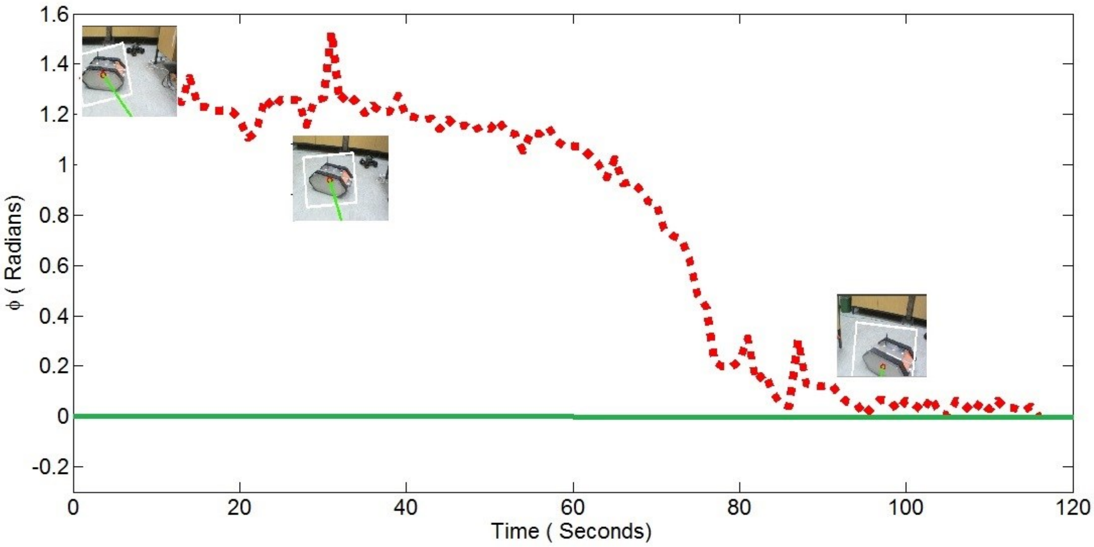


Figure 4. 29 **Angle between target, blimp and image's centre.**

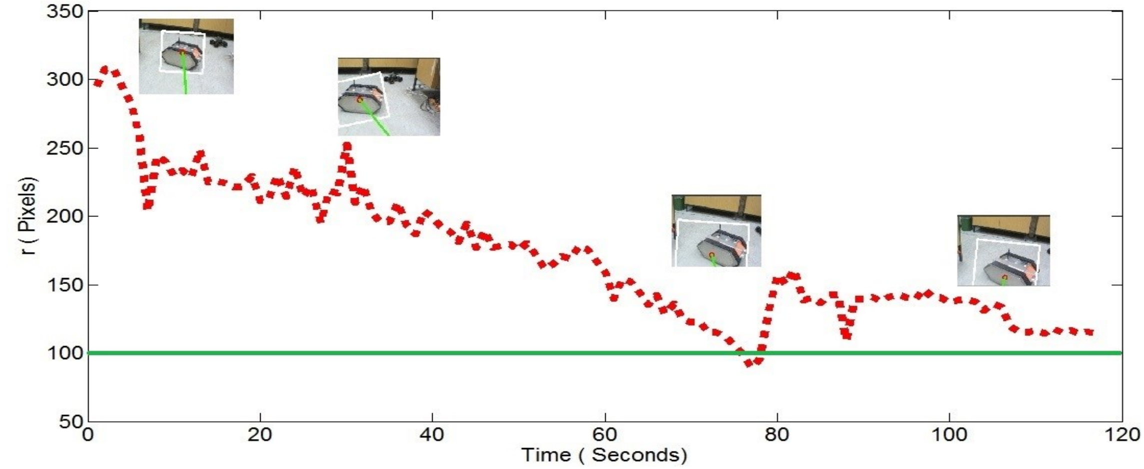


Figure 4. 30 Estimated distance between target and blimp.

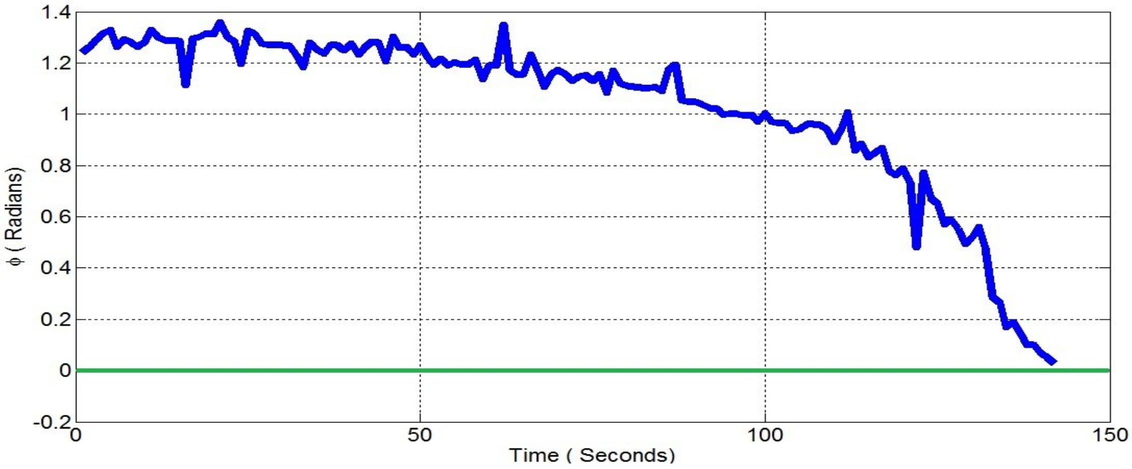


Figure 4. 31 Angle between target, blimp and image's centre.

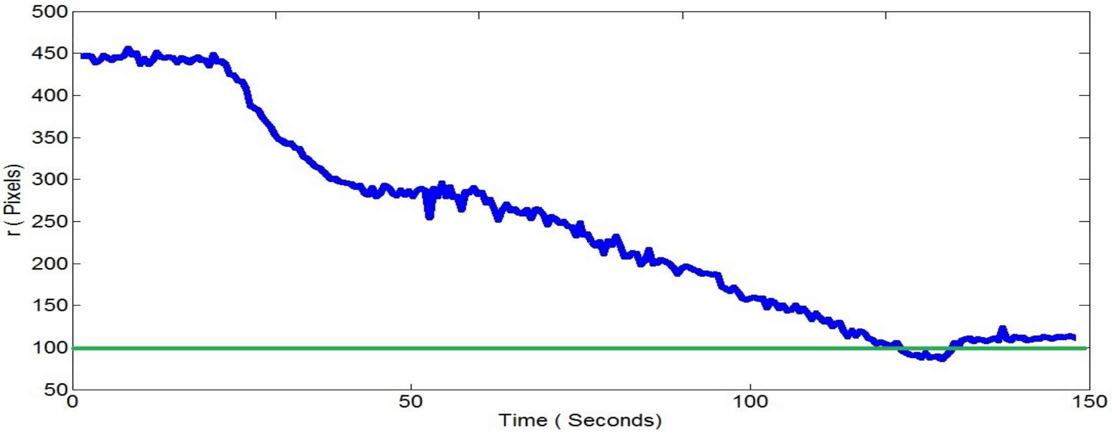


Figure 4. 32 Estimated distance between target and blimp.

4.3.2. Ground Robot Navigation

As we described before, the blimp and ground robots have received and gained more attention in the robot research due to the fact that they are strong and powerful potential in the explorations and surveillance missions . There are many applications in where the ground robot might navigate and explore an area without any previous knowledge of the surrounded environment. In this section, a system composed by a blimp and ground robots is presented to cooperate and share visual information to address the mission requirements. Additionally, the projections and coordinate system have been introduced and they are very important to use the camera obtained images in order to identify, detect and localize the mission ground robot and the obstacles in the environment. The objective here is to obtain the relative distance or position between the ground robot and obstacles in the environment. The coordinate frame for this system is illustrating in Fig. 4.31. For this purpose, the pinhole camera model has been used. It is defined as the relationship between a **3D** point in real frame world with respect to its **2D** corresponding projection point on the image plane. There are two sets of parameters are required and recovered to reconstruct the pinhole model. First, the parameters that the locate and orientate the camera reference frame respecting to the world frame and they are called extrinsic camera parameters. Then, the intrinsic parameters that can link the coordination in pixels in the image frame with coordination in camera frame.

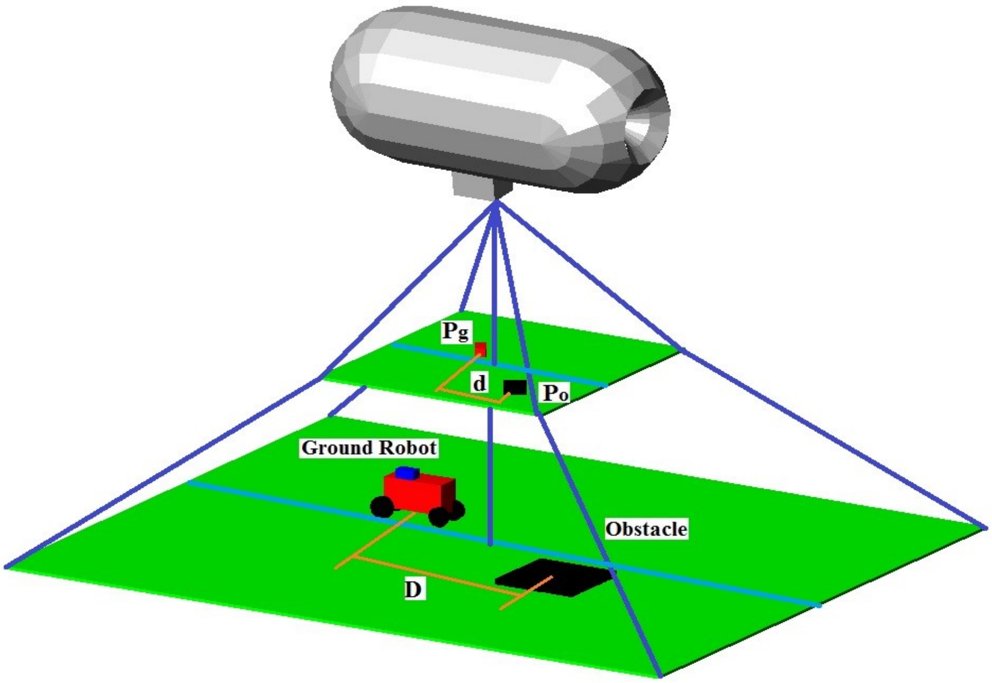


Figure 4. 33 **Coordinate frame of the system.**

The ideal pinhole camera model as it was illustrating have capability to link between a 3D point **(X, Y, Z)^T** and the corresponding **2D** projection point **(u, v)** as it is illustrating in Fig. 4.32. The projection of a **3D (X, Y, Z)^T** at **(u, v)^T** can be given as:

$$x = -(x_{pf} - o_x) * s_x \qquad 4.23$$

$$y = -(y_{pf} - o_y) * s_y \qquad 4.24$$

Where **s_x, s_y** are considered to be the effective size of the pixels **and (o_x, o_y)** are the coordinate of the principal point.

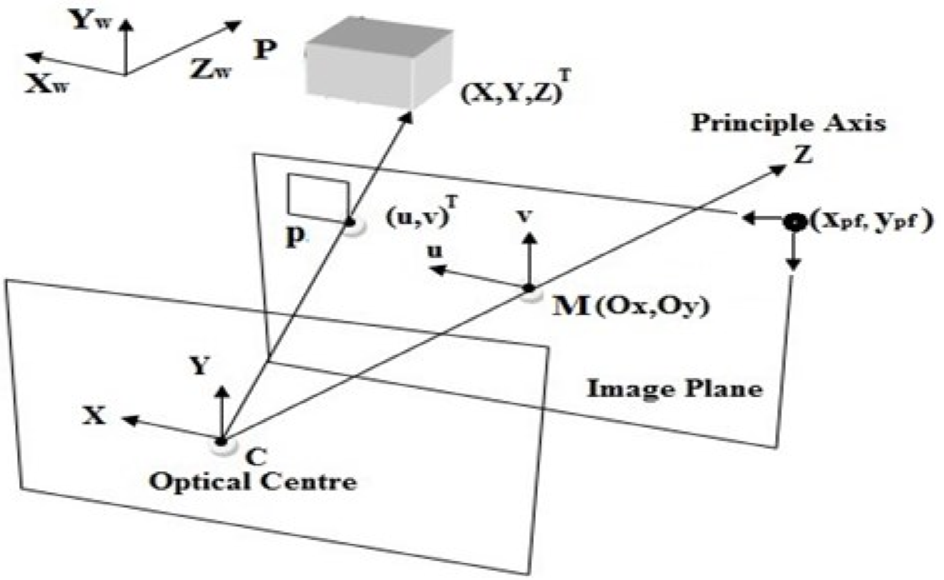


Figure 4. 34 The perspective projection model.

The most right-top coordinates here is the **(x_{pf}, y_{pf}).** Then, considering the use of the following matrix to transfer between the image plane coordinates to pixel coordinates:

$$\begin{bmatrix} x_{pf} \\ y_{pf} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-1}{s_x} & 0 & o_x \\ 0 & \frac{-1}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad 4.25$$

Now, the intrinsic camera parameters can be given by this matrix:

$$M_{in} = \begin{bmatrix} \frac{-f}{s_x} & 0 & o_x \\ 0 & \frac{-f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \qquad 4.26$$

Where **f** is the focal length. The extrinsic camera parameters matrix is given by the following:

$$M_{ex} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix} \qquad 4.27$$

Where **f** is the focal length of the camera , **R** and **T** are the that identify the rotation matrix and translation vector between the camera reference frame (known) and the world reference frame (known). $\boldsymbol{R_i^T}$ is the **i**$^{\text{th}}$ row of the rotation matrix.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad 4.28$$

By implementing and using homogeneous coordinates, the projection matrix can be given as:

$$\begin{bmatrix} x_h \\ y_h \\ w \end{bmatrix} = M_{in}M_{ex}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_P\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \qquad 4.29$$

Note that **M$_{\mathbf{p}}$** is final matrix that comes from combining the intrinsic matrix and the extrinsic matrix and it is well known as the projection matrix. The relationship between 3D point and its 2D projection point are linear transformation from space **$(X_w,Y_w, Z_w,1)^T$** to the plane **$(x_h, y_h, w)^T$**. Therefore, in the pixel coordinates we can conclude that **(x_{pf}, y_{pf})** are **$(x_h/w, y_h/h)$.** Therefore, by implementing the homogenization the pixel coordinates are given in the following:

$$x_{pf} = \frac{x_h}{w} = \frac{m_{11}X_W + m_{12}Y_W + m_{13}Z_W + m_{14}}{m_{31}X_W + m_{32}Y_W + m_{33}Z_W + m_{34}} \qquad 4.30$$

$$y_{pf} = \frac{y_h}{w} = \frac{m_{21}X_W + m_{22}Y_W + m_{23}Z_W + m_{24}}{m_{31}X_W + m_{32}Y_W + m_{33}Z_W + m_{34}} \qquad 4.31$$

Now, seeking to find the perspective camera model we need simply to assume that **$o_x= o_y =0$** and **$s_x=s_y=1$**:

$$M_p = \begin{bmatrix} -fr_{11} & -fr_{12} & -fr_{13} & fR_1^TT \\ -fr_{21} & -fr_{22} & -fr_{23} & fR_2^TT \\ r_{31} & r_{32} & r_{33} & -R_3^TT \end{bmatrix} \qquad 4.32$$

where $R_i^T = [r_{i1}, r_{i2}, r_{i3}]$, in order to check and verify the validity and correctness of the previous matrix:

$$M_pP_w = \begin{bmatrix} -fR_1^T & fR_1^TT \\ -fR_2^T & fR_2^TT \\ R_3^T & -R_3^TT \end{bmatrix}\begin{bmatrix} P_w \\ 1 \end{bmatrix} = \begin{bmatrix} -fR_1^T(P_w - T) \\ -fR_2^T(P_w - T) \\ R_3^T(P_w - T) \end{bmatrix} \qquad 4.33$$

Chapter: Computer Vision for Blimp Robot

Where $\boldsymbol{R_i^T}$ is the **i**th row of the rotation matrix.

After homogenization that **(u, v)** are point on image plane with any value of **(x_{pf}, y_{pf}),** and based on the relation between the coordinates of point **P** in real world (**P_w**) and camera coordinates (**P_c**) we get:

$$u = -f\frac{R_1^T(P_w - T)}{R_3^T(P_w - T)} = -f\frac{X_c}{Z_c} \qquad 4.34$$

$$v = -f\frac{R_2^T(P_w - T)}{R_3^T(P_w - T)} = -f\frac{Y_c}{Z_c} \qquad 4.35$$

Let us now consider and define that ***I(x, y)*** is the image plane frame. The coordinates of most left-bottom and most right-top points are (0, 0) to (263,167), respectively. The pose which is known as the (location and orientation) for the target ground robot is **P_g = (x_g, y_g, θ_g)** and the obstacle position is **P_o = (x_o, y_o, θ_o)** in the image plane. Both the locations and the orientations are related to the projection point of blimp camera in the image plane as shown in Fig. 4.26 and Fig. 4.27 and they are given by the following:

$$\theta_g = \tan^{-1}\left(\frac{x_g}{y_g}\right) \qquad 4.36$$

$$\theta_o = \tan^{-1}\left(\frac{x_o}{y_o}\right) \qquad 4.37$$

$$d_g = \sqrt{x_g^2 + y_g^2} \qquad 4.38$$

$$d_o = \sqrt{x_o^2 + y_o^2} \qquad 4.39$$

$$d_g^o = \sqrt{\left(x_o - x_g\right)^2 + \left(y_o - y_g\right)^2} \qquad 4.40$$

$$\theta_g^o = \tan^{-1}\left(\frac{x_o - x_g}{y_o - y_g}\right) \qquad 4.41$$

Where $\mathbf{d_g}$ is the distance of the ground robot center projection point with respect to blimp's camera projection point, $\mathbf{d_o}$ is the distance of obstacle projection point with respect to the blimp's camera projection point. Also, $\boldsymbol{\theta_g}$ is the angle between the robot projection and blimp's path projection, $\boldsymbol{\theta_o}$ considers to be the angle between the obstacle projection point and blimp's path projection. $\boldsymbol{d_g^o}$ and $\boldsymbol{\theta_g^o}$ are the position and orientation of the robot with respect to the obstacle, respectively, in the image frame as shown in Fig. 4. 33.

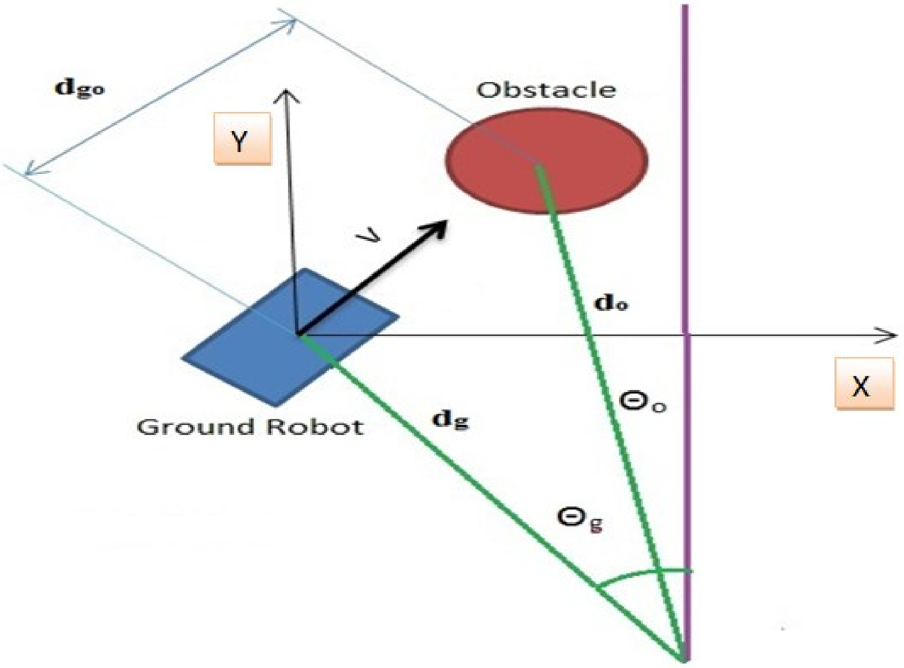


Figure 4. 35 **Position of obstacle related to ground robot.**

Based on Eq. 4.34 and Eq. 4.35, the metric distance ***D*** between the ground robot and the obstacle is obtained as the following:

$$D = \left|-\frac{H_b . d_g^o}{f}\right| \qquad 4.42$$

Where ***H_b*** is the blimp altitude. Note that the operation here is mathematical multiplication.

4.3.3. Blimp Robot Follows 3-D Aerial Object

In this section, we present a visual servoing system in order to follow a 3-Dimentional **(3D)** aerial object by blimp robot. The proposed vision is designed by using SURF on the on-board navigation system to localize the 3D aerial target. In addition, the IPM has been used to allow to remove the perspective effect from the image and to remap the image into a new 2-D domain where the information content is homogeneously distributed among all pixels. Finally, we have made the correspondence relation between the location on the image plane and the location on real work field as it is shown in Fig. 4.34.

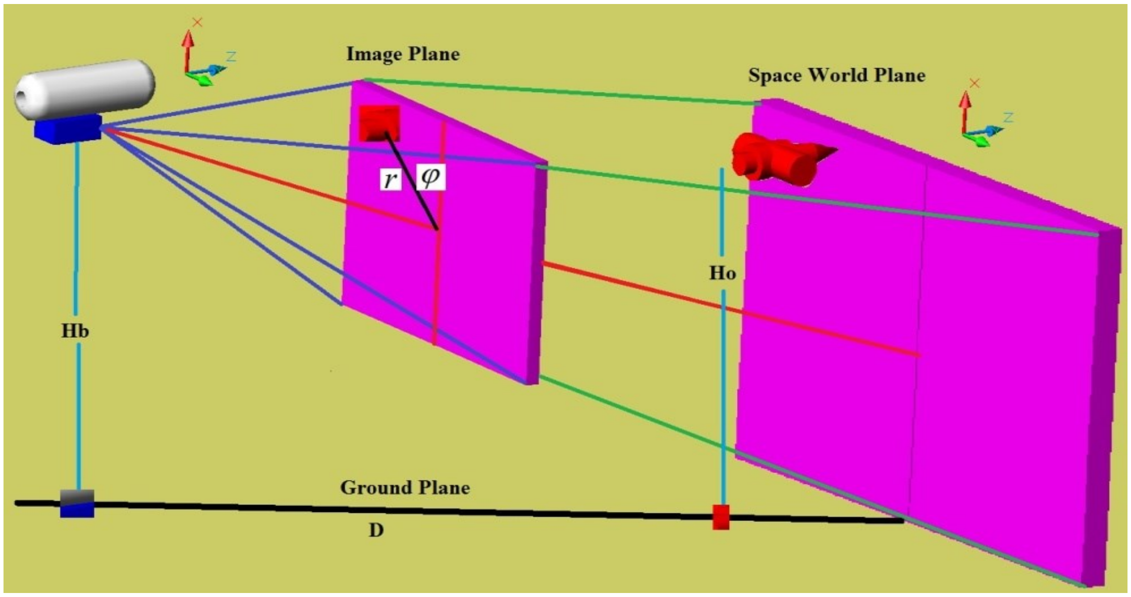


Figure 4. 36 **Coordinate of the flying object related to Camera.**

Considering a flying object at a certain altitude $\mathbf{H_o}$ on the world space. The blimp robot has a certain known altitude $\mathbf{H_b}$ and far from the flying object with a distance **D** in ground plane as shown in Fig 4. 35. The **3D** object projection on the image plane defined by its center point in $\mathbf{(x_o, y_o)}$ plane with polar coordinate ($\mathbf{r, \phi}$).

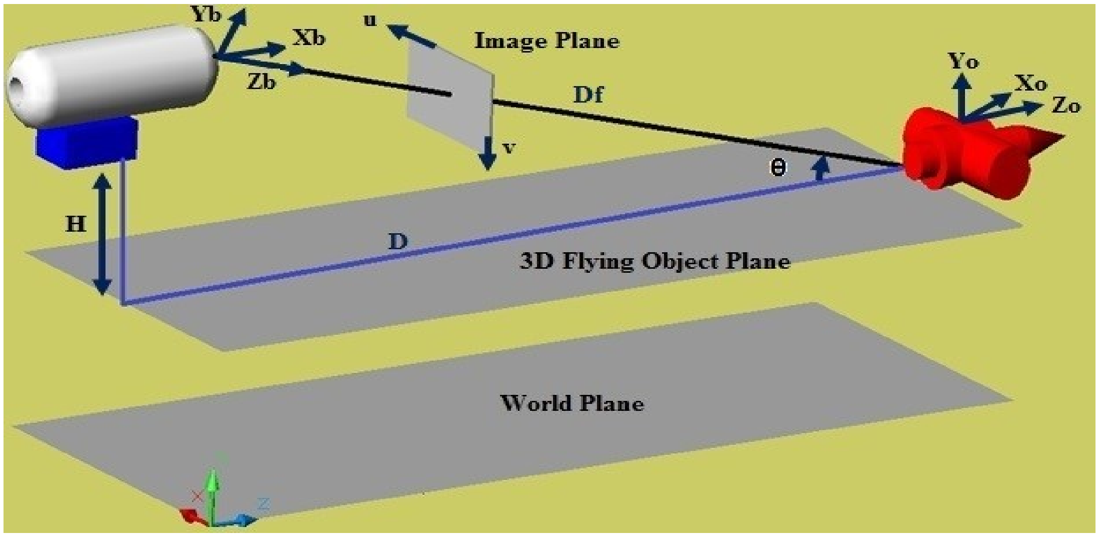


Figure 4. 37 **IPM transformation between the two objects**

In fact, the perspective view of the captured image has some distortions than the actual shape of the space in ($\mathbf{X_w, Y_w, Z_w}$) world coordinate. The angle of view and the distances of the object from the camera contribute to associate different information to each pixel of the image. Hence, this image needs to go through a pre-processing step to remedy this distortion by using the transformation technique IPM. The distance between the blimp robot and the object on the ground plane is **D** which is very important here to feedback to the controller system. By assuming that the two object fly in parallel planes, the projection of **D** line has the same length on the **3D** object plane which is also **D**. As it is shown in Fig. 4.35, the difference between the two planes is **H** and the rotation value which is the translation along the camera optical axis is **θ**. However, the use of single camera which is mounted on the gondola does not provide the depth information about this object because the non linearity between the object position in the image plane and its position in the **3D** object plane as well as in the real world plane. The transforming of the image could remove the non linearity of these distances which can be obtained by implemented **IPM**.

To verify the proposed vision system, several experiments of the complete system were conducted. During these experiments the blimp robot was flying at a certain altitude as well as the 3D objects move in the 3D object plane with constant altitude. In addition, the 3D object is static at a certain altitude and fixed distance from the blimp in some tests. The target would be localized, detected and identified and tracked by the blimp vision system. Based on the vision data, the vectorization angle for the main propellers and the yaw angle were controlled to follow the target and kept it in a certain position in the image as well as in a certain distance from the blimp. Fig. 4.36 is illustrating the angle between the blimp and the target obtained from 2 tests (data 1 and data 2). When the blimp detects the target, the control will change the yaw angle for the blimp robot in order to get the target at the center line of the image. In test 1 the target was fixed and the blimp robot changes the orientation to get the target at the center of the image. However, in test 2 the target was moving and changing the direction over time that is explaining why the angle between the target and the center line of the image was zero at 97 seconds then when the target moves away the controller succeeds to make it at the center line of the image at approximately 190 seconds.

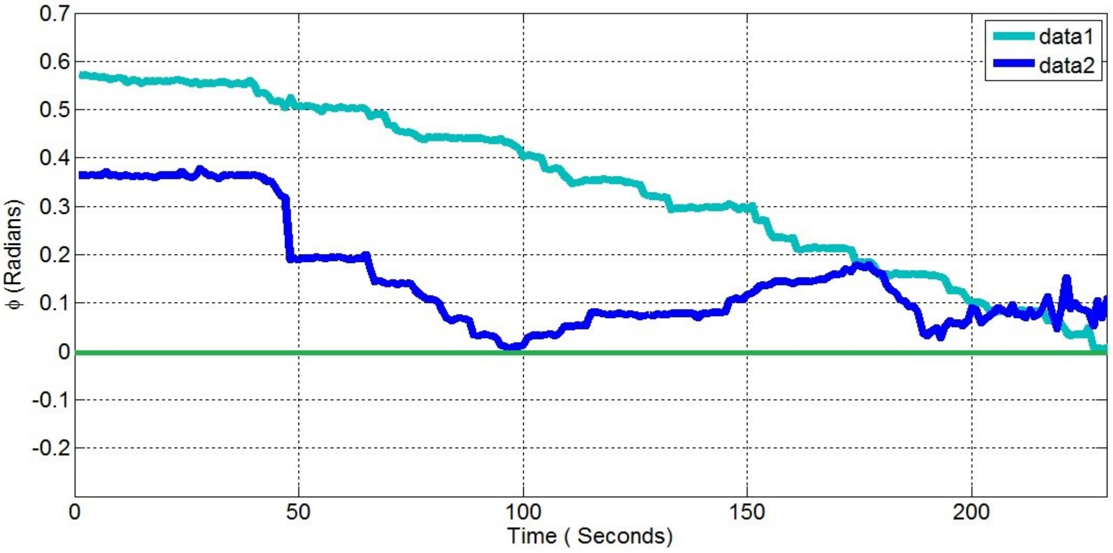


Figure 4. 38 **Angle between target, blimp and image's centre.**

Fig. 4.37 and Fig. 4.38 show the blimp behavior in order to control the yaw angle and the vectorization angle to detect the target, follow, and get it at the center line of the image. Fig. 4.37 shows the angle between the blimp and target. It is obvious that when the blimp robot detects the target, it will change the orientation in order to put the target at the center line of the image (zero angle). The distance between the blimp and the object is illustrating in Fig. 4.38. It is clearly that when the blimp robot detects that target, it will fly toward the target in order to keep it at a certain distance. We assume in this experiment that the target is static at a certain position from the blimp. The blimp vision system detects and flies toward the target, then, the engine will stop and the blimp stop moving at a certain distance which is here 130 cm.

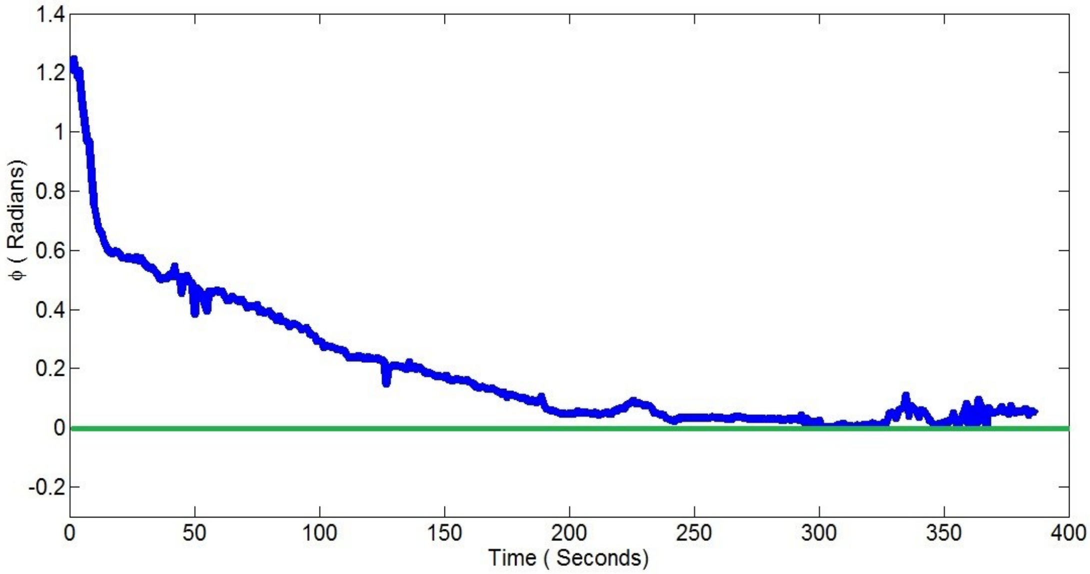


Figure 4. 39 **Angle between target, blimp and image's centre.**

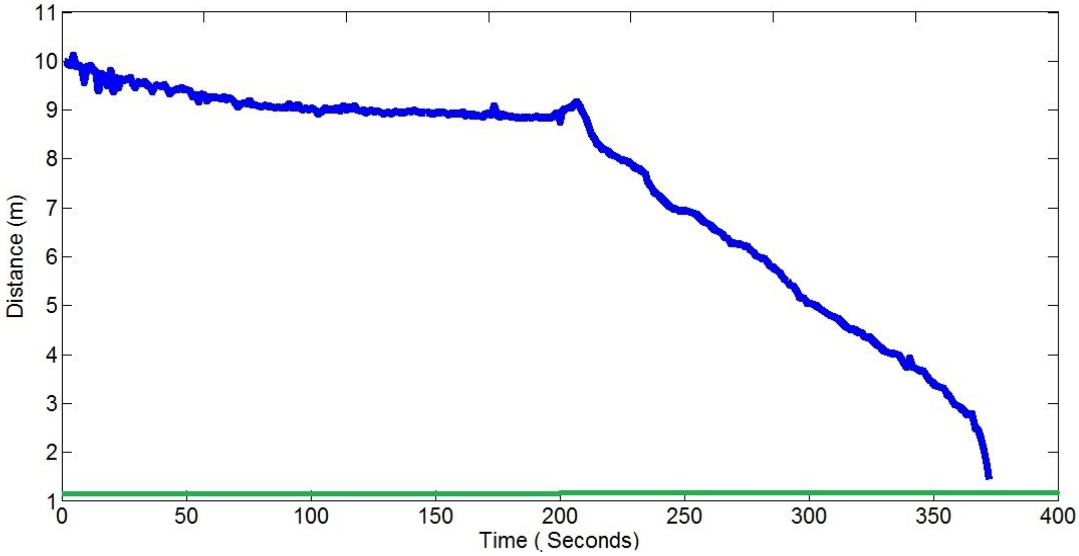


Figure 4. 40 **Distance between target and blimp.**

4.4 Conclusions

This chapter has presented the visual processing techniques and algorithms used on the development of the thesis. These techniques include visual algorithms for features detection, tracking and localizing based on images features based on SURF algorithm by considering some criteria such as speed, robust against noise and good detection in scale or rotation invariants. These features might be used for blimp robot control (servoing visual control) as well as to guide a ground robot to navigate safety in the unknown environments. By implementing IPM algorithm we could obtain the distances between the blimp robot and the target and these information will provide the servoing control with inputs in order to control the behavior of the blimp robot during the missions.

Chapter 5 Experiments and Results

In this chapter, the results of designing the blimp robot, computer vision approaches, intelligent fuzzy sets model, fuzzy controllers as well as the multi-target tracking results are discussed. The experiments are performed to test the capability of the approaches that have been described in the previous chapters. The stability issues for the visual control (servoing control) has been studied well and could be considered as solved in this thesis. This is due to the fact that the prediction vision information has been corrected by fuzzy sets and the embedded vision system plays a significant role in order to obtain these data to feed the visual control. In addition, the vision algorithm has been designed by using new features descriptors (SURF and IPM). In addition, in order to increase the robustness of this vision algorithm the RANSAC algorithm was implemented.

Because the performance, high robustness and reliability are generally required in high level controls, it thereby requires new developments on computer vision techniques and control algorithms to fulfill several specifications such as to improve the blimp robot capabilities during missions. The blimp overall control system architecture is shown in Fig. 5.1. The flight control of the blimp robot divided into three main parts to control the main behaviors of the blimp robot (altitude, avoid obstacles as well as detect and follow target in the environment). More details about the fuzzy controllers including the initial membership functions, the optimized membership functions, and the optimized rules for each controller could be found in Appendix D.

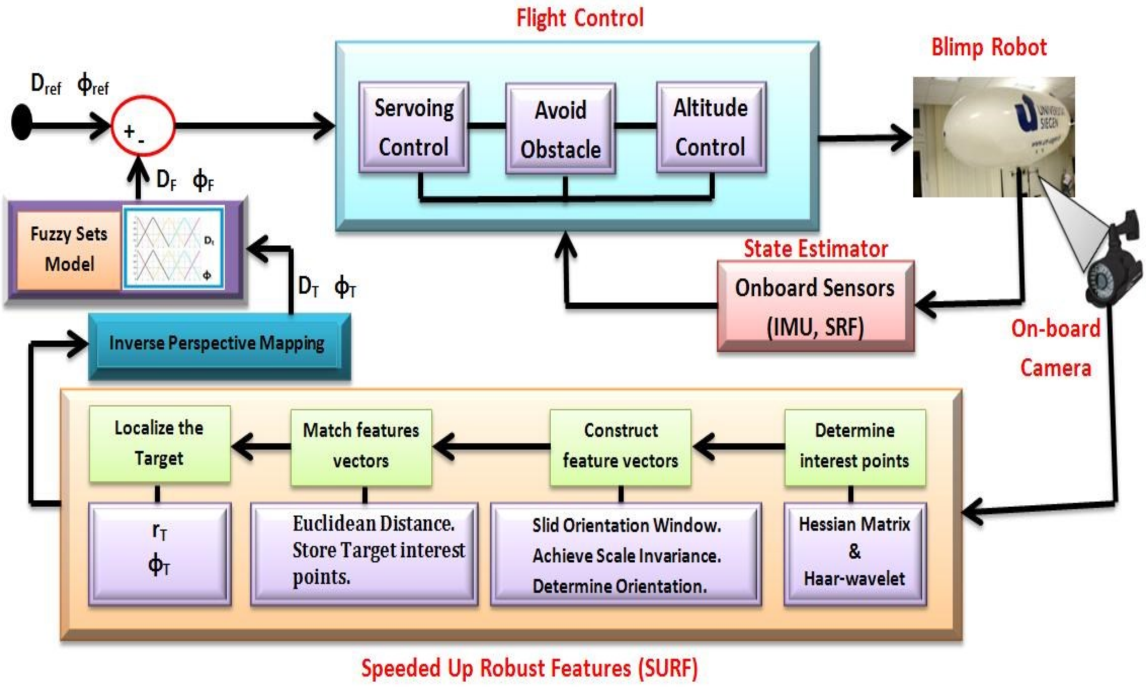

Flight Control
Blimp Robot
D_{ref} ϕ_{ref}
+
-
Servoing Control
Avoid Obstacle
Altitude Control
D_F ϕ_F
Fuzzy Sets Model
D_T ϕ_T
State Estimator
Onboard Sensors (IMU, SRF)
On-board Camera
Inverse Perspective Mapping
Localize the Target
Match features vectors
Construct feature vectors
Determine interest points
r_T ϕ_T
Euclidean Distance. Store Target interest points.
Slid Orientation Window. Achieve Scale Invariance. Determine Orientation.
Hessian Matrix & Haar-wavelet
Speeded Up Robust Features (SURF)

Figure 5. 1 Overall blimp control system architecture.

5.1 Fuzzy Set Model for Blimp's Navigation

The control system for the autonomous blimp was designed basically to keep the blimp navigating with maintaining the desired horizontal speed, avoid obstacles with reference altitude. During the navigation if the blimp detects an obstacle the control system attempts to change the angle of the propellers main motors [210].

5.1.1 Fuzzy Sets Model for the Ultrasonic Sensors

Due to its operation in three dimensions and non-linearity of propellers action, the navigation blimp software provided limited autonomous operation of the blimp. The whole blimp structure is demonstrated in Fig. 5.2. Indeed, if the blimp navigates from position ***A*** forward to position ***B***, the distance between the blimp and the objects is not accurate since the **SRF02** sensors have some limitations and drawbacks. In addition, the blimp navigates with view angle, so for more accurate distances between the blimp and any obstacle an optimization process is needed. Even though such sensors are able to give required information with fast response time, but they have some limitations due to some characteristics that might affect their performance such as angular uncertainty **β**, false reflection, and radial error **Ɛ**. While the **β** may happen due to the view angle between the sensor and the obstacle, the radial error comes from the beam width of the sensor.

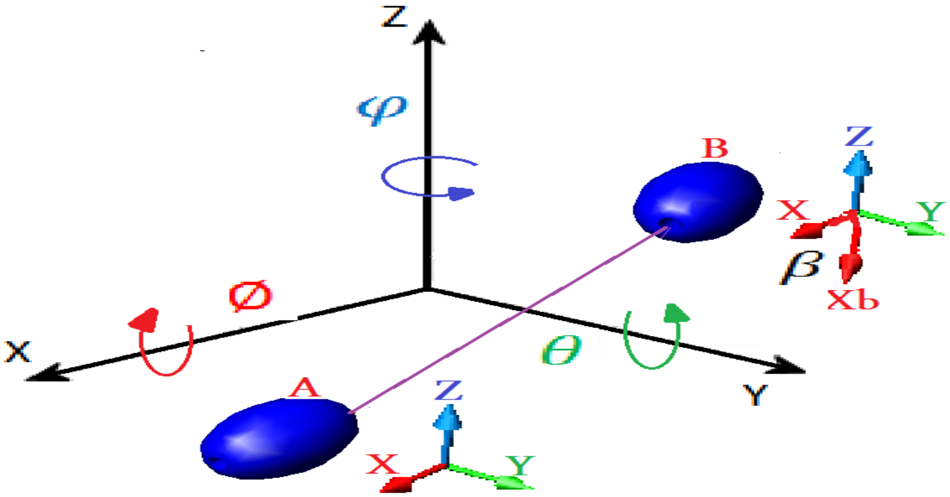


Figure 5. 2 **Blimp structure and coordinate.**

Several experiments had been done to test the behaviors and the characteristics of **SRF02** sensors. Fig. 5.3 **A, B** and **C** show the effect of the radial error and the range of view with different distances. The study aims to find the effect of the blimp's view angle and the distance between the blimp and object. First, the experiments are designed and implemented in the **RST** department to collect the sensors information and data for several distances and angles; then analyzed these data to find and design possibility histograms and then transferred them to propose a fuzzy knowledge base for the system (fuzzy membership function). Four ultrasonic sensors have been mounted on the metal-light arc on front of the gondola. The front sensors ***S1, S2, S3, S4*** provide the distance readings ***D1, D2, D3, D4***. By studying the relation between ***[{S1,S2}, {S2,S3}, {S3,S4}]*** and the view angle **β=[0°, 22.5°],** it is clearly obvious that the smallest distance reading is the one with the smallest angle view. For simplicity, we will study the relation between S1 and S2 to obtain the possibilities histograms. Then, the possibilities distributions results can be generalized for any two adjacent sensors {(S2, S3), (S3, S4)}. Fig. 5.4

shows the three possibilities between the two sensors. The three possibilities of view angel are summarized in Table 5.1.

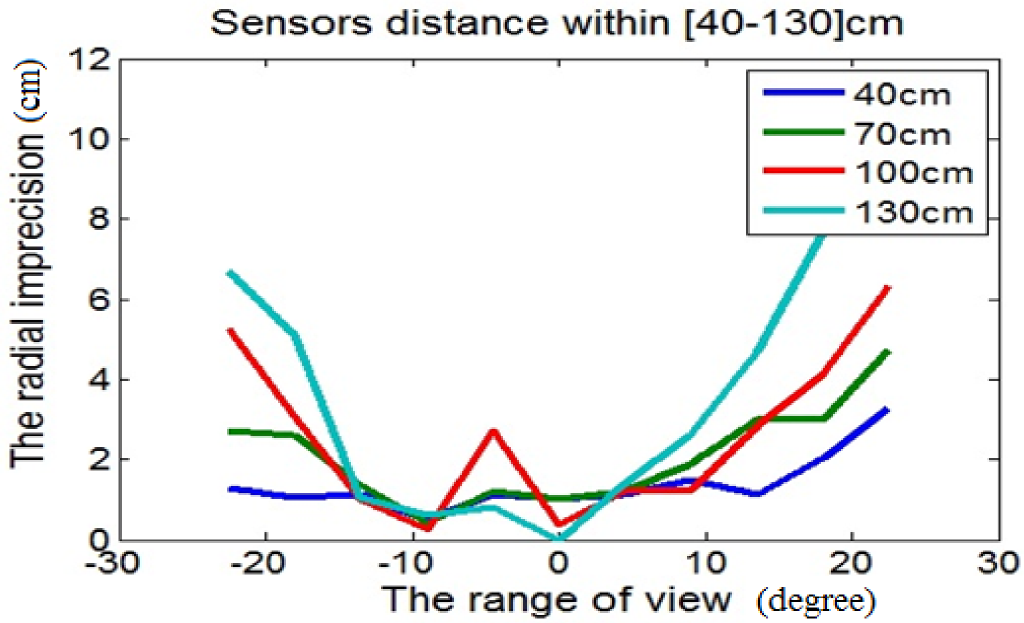


Figure 5. 3a Radial imprecision obtained from various view angle and distances A.

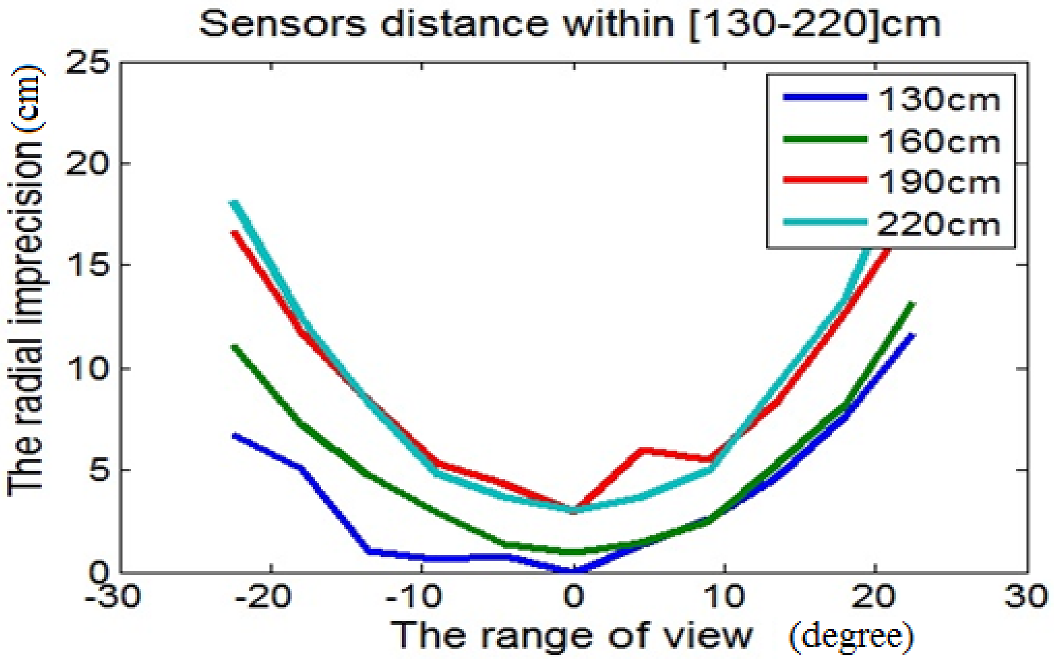


Figure 5.3B Radial imprecision obtained from various view angle and distances B.

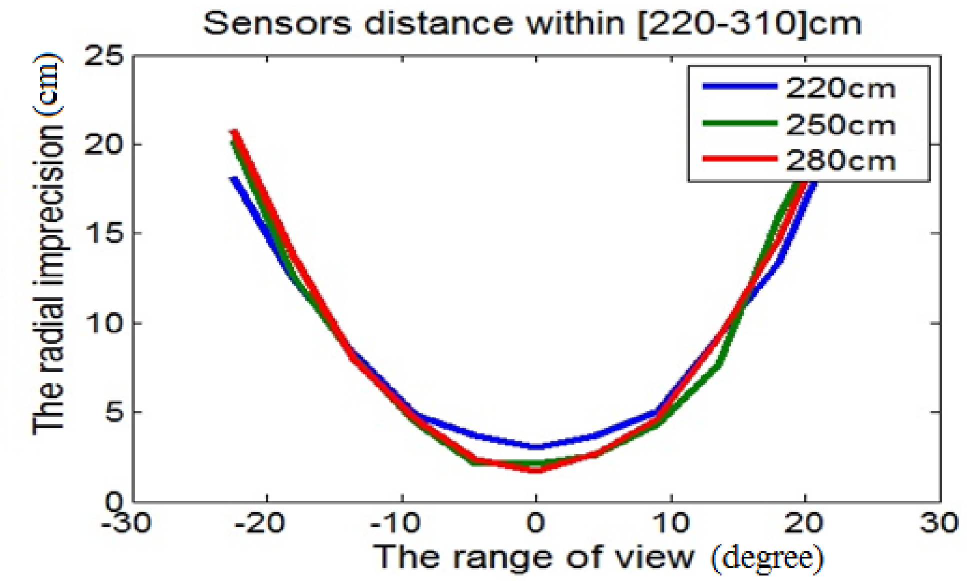


Figure 5.3C Radial imprecision obtained from various view angle and distances C.

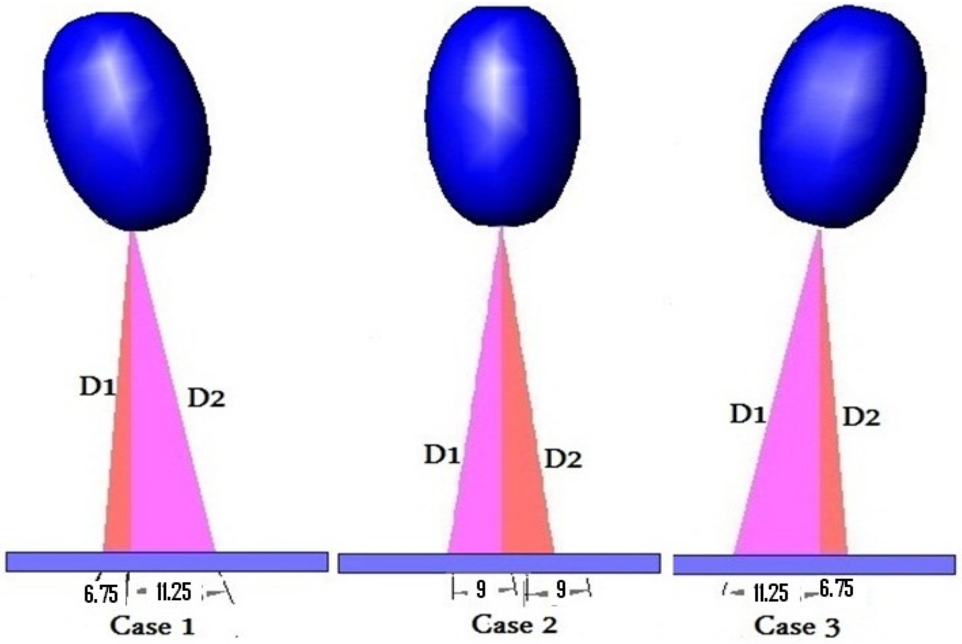


Figure 5. 4 The possibility cases between any adjacent two sensors.

Possibilities	Shortest Distance	View Angle
1	D_1	$B=\beta_1-6.75$
2	$D_1=D_2$	$B=\beta_1-9$, $B=\beta_2+9$
3	D_2	$B=\beta_2+6.75$

Table 5.1 Possibilities between two sensors.

Now, based on the collected data and then the possibilities cases analysis, we might easy represent **β** by possibility distributions **π_D(β)** as they given and summarized in Table 5.2. Table 5.3 demonstrated the support and core parameters for the new possibilities histograms or more precisely the trapezoidal membership functions. Then, three membership functions were modeled as shown in Fig. 5.5, Fig. 5.6 and Fig. 5.7.

Case	$\Pi_D=0$	$\Pi_D=1$	$\Pi_D=]0, 1[$
1	$(\beta_{1,}\infty) \cup (\beta_1-11.25, -\infty)$	$(\beta_1-6.75, \beta_1-9)$	$(\beta_{1,} \beta_1-6.75) \cup (\beta_1-9_, \beta_1-11.25)$
2	$(\beta_1-6.75_,+\infty) \cup (\beta_1-11.25, -\infty)$	(β_1-9) or (β_2+9)	$(\beta_1-9_, \beta_1-11.25) \cup (\beta_1-6.75_, \beta_1-9)$
3	$(\beta_{2,}-\infty) \cup (\beta_2+11.25, \infty)$	$(\beta_2+6.75, \beta_2+9)$	$(\beta_{2,} \beta_2+6.75) \cup (\beta_2+9_, \beta_2+11.25)$

Table 5.2 The possibilities distribution of uncertainty in the view angle.

Support A	Core A	μ_β
$(\beta_{1,} \beta_1-11.25)$	$(\beta_1-6.75, \beta_1-9)$	{0, -6.75, -9, -11.25}
$(\beta_1-6.75_, \beta_1-11.25)$	(β_1-9, β_1-9)	{-6.75, -9, -9, -11.25}
$(\beta_{2,} \beta_2+11.25)$	$(\beta_2+6.75, \beta_2+9)$	{11.25, 9, 6.75, 0}

Table 5.3 Support, Core and membership function for the three cases.

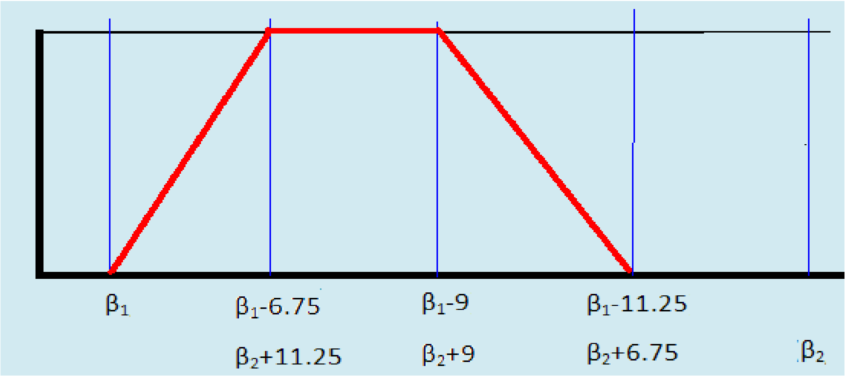


Figure 5. 5 Fuzzy membership functions case 1

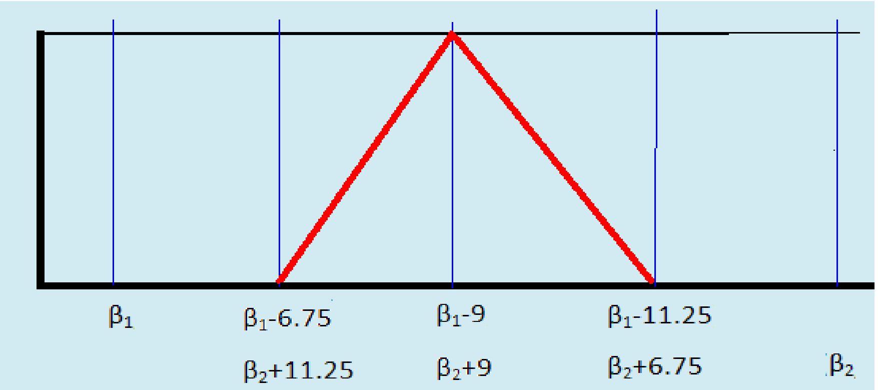


Figure 5. 6 **Fuzzy membership functions case 2.**

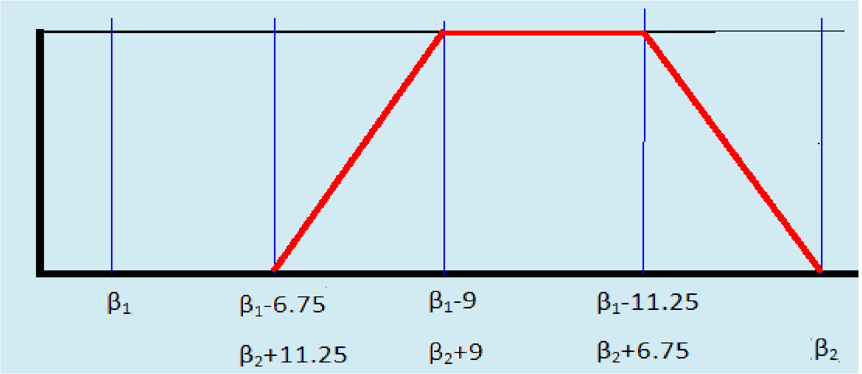


Figure 5. 7 **Fuzzy membership functions case 3.**

Therefore, to find the shortest distance **(SD)** we should take into account two points. First, we must rotate the reading distance **(RD)** from sensors to the original axis coordinate **(x, y, and z)**. Second, depend on the range of view the final shortest distance must take into account the radial error ε. However, the real distance belongs to an interval within a known possibility by using $SD_x = [\varepsilon \pm RD]\cos(|\beta|)$ and $SD_y = [\varepsilon \pm RD]\sin(|\beta|)$. However, the experiments data shows the values of these errors related to the range of view. As a result, these errors can be reduced and modeled by fuzzy sets and possibility distributions as it is shown in Table 5.4. The vectors of endpoints, cores and support for possibilities distribution are shown in Table 5.5.

$\mathbf{A_i}$	$\mathbf{S_i}$
$\boldsymbol{A_1 = \langle[0,1],[1,3],[-1,4]\rangle}$	$\boldsymbol{S_1 = \{[0,1] = 0.25,[1,3] = 0.25,[0,3] = 0.25,[-1,4] = 0.25\}}$
$\boldsymbol{A_2 = \langle[-1,4],[0,5],[2,7],[3,10]\rangle}$	$\boldsymbol{S_2 = \{[-1,4] = 0.25,[0,5] = 0.25,[2,7] = 0.25,[3,10] = 0.25\}}$
$\boldsymbol{A_3 = \langle[3,10],[4,11],[5,11]\rangle}$	$\boldsymbol{S_3 = \{[3,10] = 1/3,[4,11] = 1/3,[5,11] = 1/3\}}$

Table 5.4 Frequency data analysis.

E_i^L	E_i^R	$C_i(\pi)$	$Supp_i(\pi)$
$\{-1,0,0,1\}$	$\{1,3,3,4\}$	$[1,1]$	$[\{-1,0\},\{0,1\},\{1,1\},\{1,3\},\{3,4\}]$
$\{-1,0,2,3\}$	$\{4,5,7,10\}$	$[3,4]$	$[\{-1,0\},\{0,2\},\{2,3\},\{3,4\},\{4,5\},\{5,7\},\{7,10\}]$
$\{3,4,5\}$	$\{10,11,11\}$	$[5,10]$	$[\{3,4\},\{4,5\},\{5,10\},\{10,1\}]$

Table 5.5 Possibilities information analysis.

The histograms are shown in Fig. 5.8a which can be converted to fuzzy membership functions as it is shown in Fig. 5.8b. Therefore, the fuzzy sets model can correct and optimize the sensors reading as it is shown in Fig. 5.9.

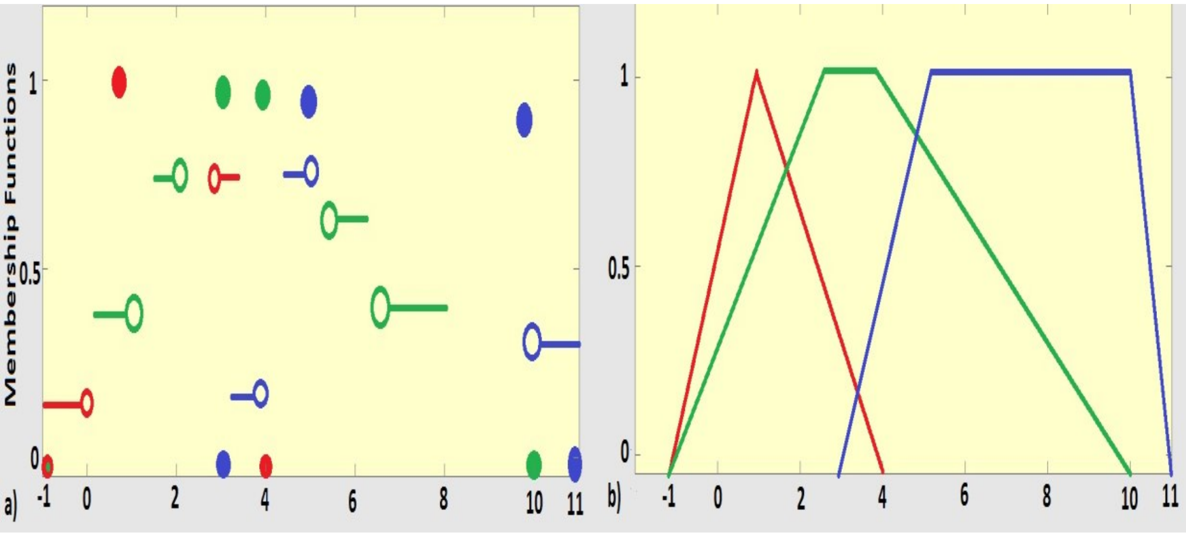


Figure 5. 8 Membership functions for redial imprecision.

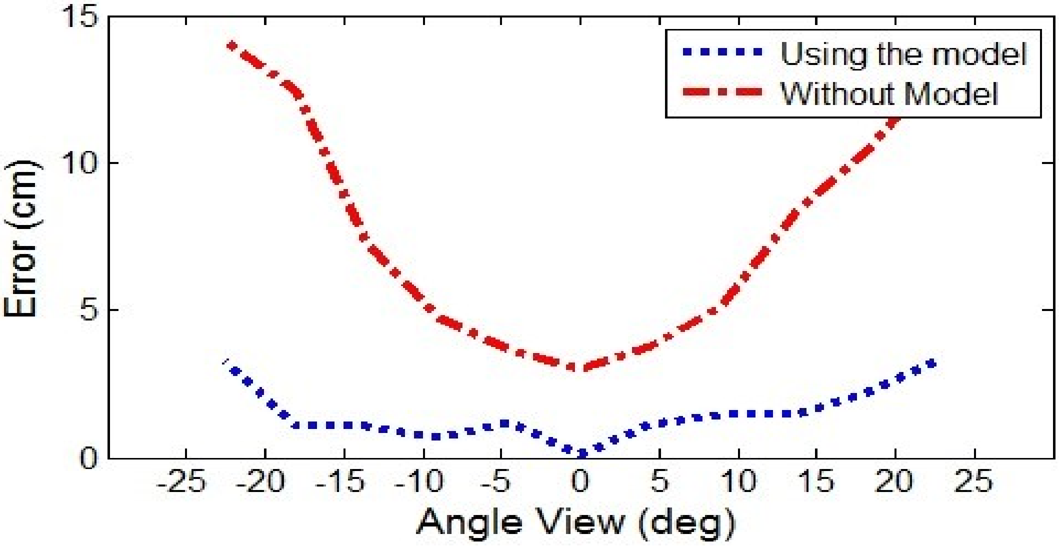


Figure 5. 9 Fuzzy Set model Outputs.

5.1.2 Fuzzy Logic Navigation Controller

For this blimp system, a second-layer fuzzy logic controller has been designed and implemented. The first layer fuzzy controls are using the fuzzy sets model to find the shortest distances between the blimp robot and any objects in the space as it was described in the previous section. The second layer actually uses the outputs of the sub-controllers as the inputs of the combined controllers which generate the main behaviors to ensure safe autonomous navigation of the blimp. The most important behavior of a robot is the obstacle avoidance. The object of such controller is to keep the blimp navigates with a safe distance from any frontal obstacles or targets. The collision avoidance control have ability and capable to control main propellers motors and then the blimp actions to change the direction when the blimp sensors detects an obstacle at a critical known distance. The controller has two inputs and one output as it is illustrating in Fig. 5.10. The first input is the error (the difference between the required avoidance distance $\mathbf{X_{ref}}$ and the shortest distance $\mathbf{X_{shortest}}$), the second input is the horizontal velocity $\mathbf{V_x}$ and the output is the angle of the main propellers.

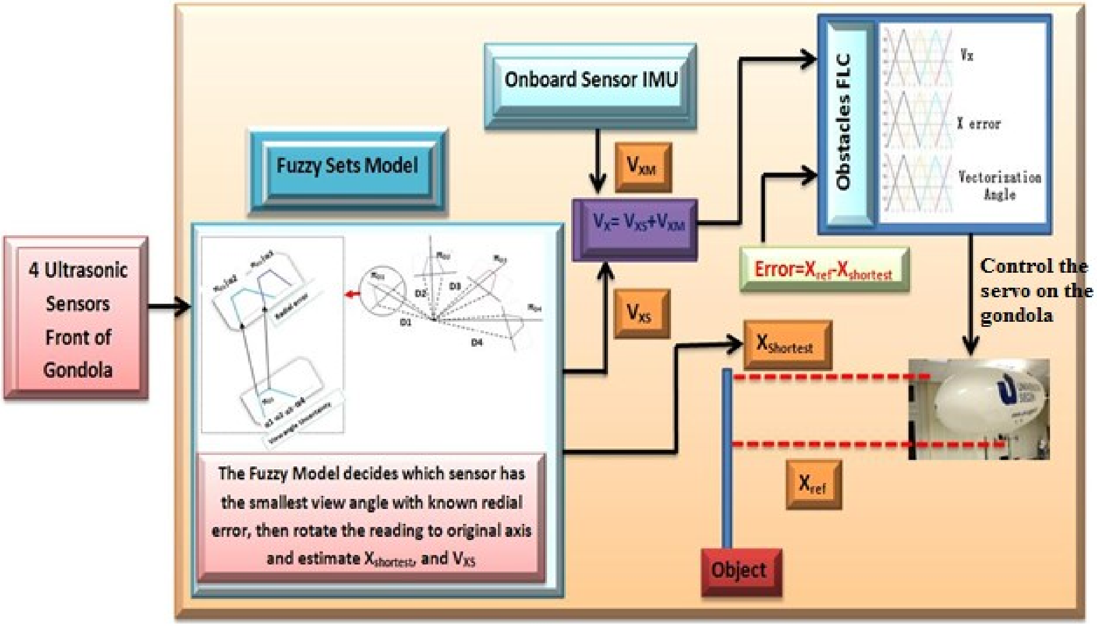

4 Ultrasonic Sensors Front of Gondola
Fuzzy Sets Model
Onboard Sensor IMU
V_{XM}
$V_X = V_{XS} + V_{XM}$
V_{XS}
Obstacles FLC
Vx
X error
Vectorization Angle
Error=X_{ref}-$X_{shortest}$
Control the servo on the gondola
$X_{Shortest}$
X_{ref}
Object
D1
D2
D3
D4
The Fuzzy Model decides which sensor has the smallest view angle with known redial error, then rotate the reading to original axis and estimate $X_{shortest}$, and V_{XS}

Figure 5. 10 Fuzzy control for avoid obstacles.

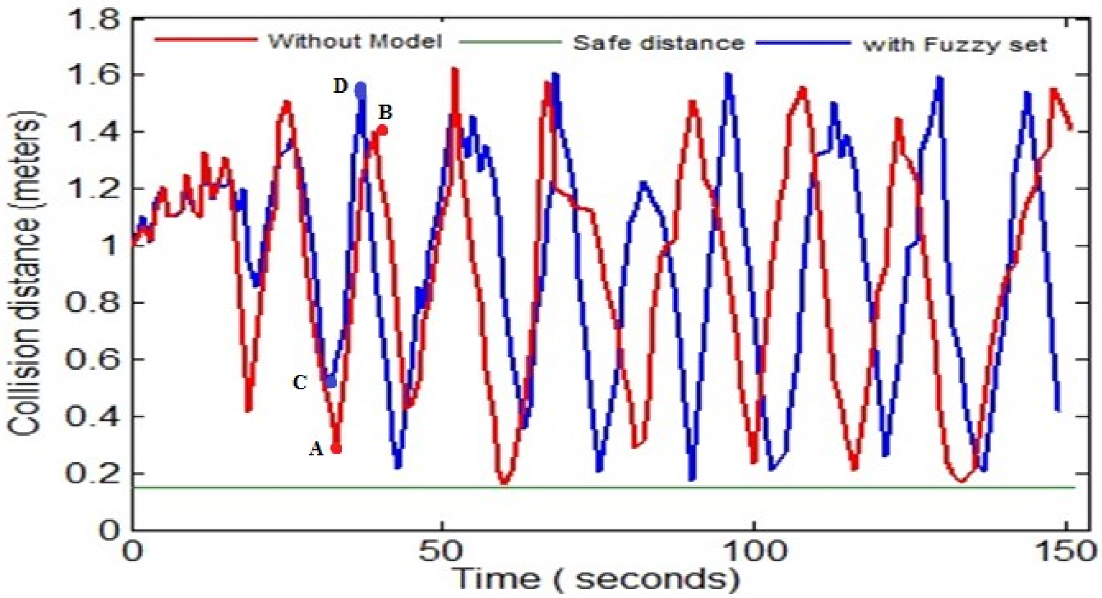


Figure 5. 11 Behavior of the fuzzy collision avoidance controller.

Fig. 5.11 shows the behavior of the blimp with constant horizontal velocity in two different tests using the fuzzy sets model or without fuzzy sets model. We need to mention here that the conditions for both tests were the same. The blimp was flying toward the wall in the **RST** Lab. When the blimp robot obtains or reaches to close and small distance (point A and C) from the wall which is here 18 cm (the green line), the horizontal velocity or horizontal speed will be reduced by the fuzzy control. The fuzzy control aims to change the vectorization angle so the blimp flies backward (point B and D) to stay away from the obstacle (wall) and so on. The altitude fuzzy controller in the blimp had two inputs: altitude error and current vertical velocity $\mathbf{V_Z}$ as it is shown in Fig. 5.12. For any amount of changes in the height or altitude errors that leads to be an indicator that whether the blimp is approaching the reference height or it is flying (or landing) away. The fuzzy controller output here is the voltage of main two propellers that is mounted on gondola.

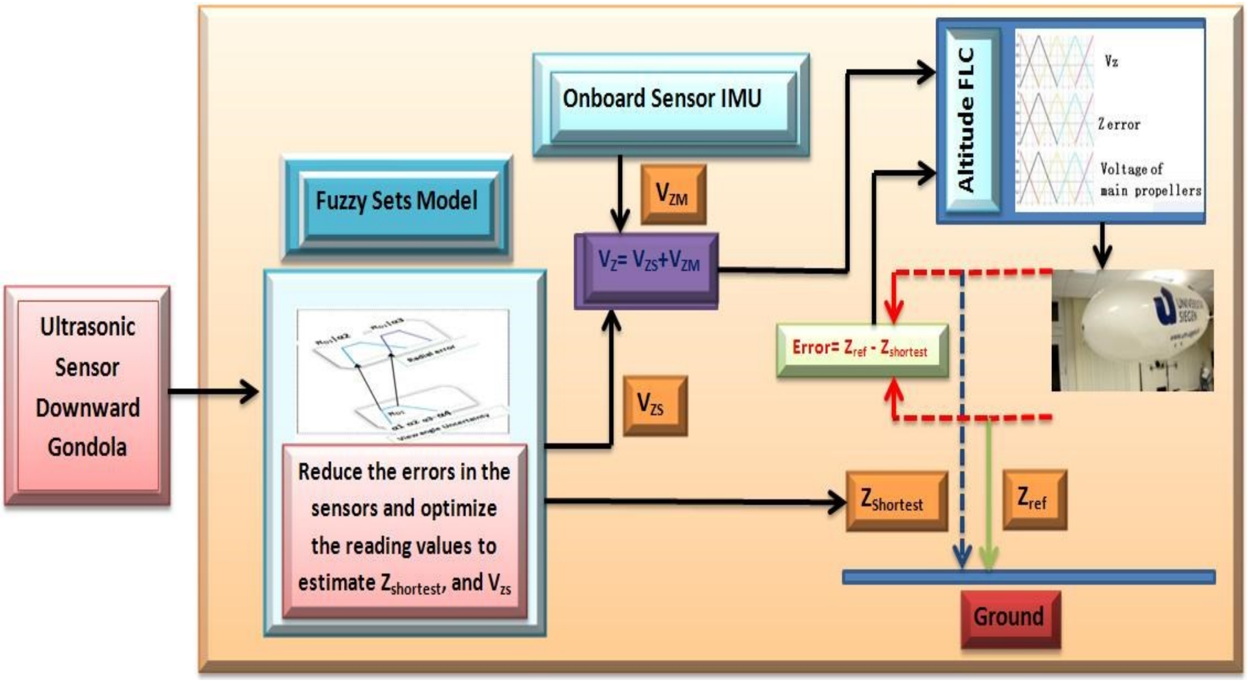

Ultrasonic Sensor Downward Gondola
Fuzzy Sets Model
Radial error
View angle Uncertainty
Reduce the errors in the sensors and optimize the reading values to estimate $Z_{shortest}$, and V_{zs}
Onboard Sensor IMU
V_{ZM}
$V_Z= V_{ZS}+V_{ZM}$
V_{ZS}
Altitude FLC
Vz
Zerror
Voltage of main propellers
Error= Z_{ref} - $Z_{shortest}$
$Z_{Shortest}$
Z_{ref}
Ground

Figure 5. 12 **The fuzzy altitude controller.**

Therefore, several tests and experiments were carried out in the **RST Labs** to check the altitude fuzzy controller. Whilst the blimp robot starts flying from 0m (the ground) to the reference altitude which varies depending on the designer's choice, the results of these tests led to good results and performance with different circumstances. Fig. 5.13, Fig. 5.14 and Fig 5.15 show the fuzzy altitude control behaviors for the blimp in three experiments with slight oscillations and deviations from the required altitude 100cm. In test 1 and test 2 the blimp was flying from the ground to reach 100cm then it detected the target and flew toward the target which explains the deviation and the oscillations in these figures. In addition, it reached the desired height for both tests at 9 seconds and 17 seconds, respectively. However, in test 3 the blimp robot detects the target before it reaches 100cm and it starts flying toward the target and during the mission it reached the desired altitude after 37 seconds.

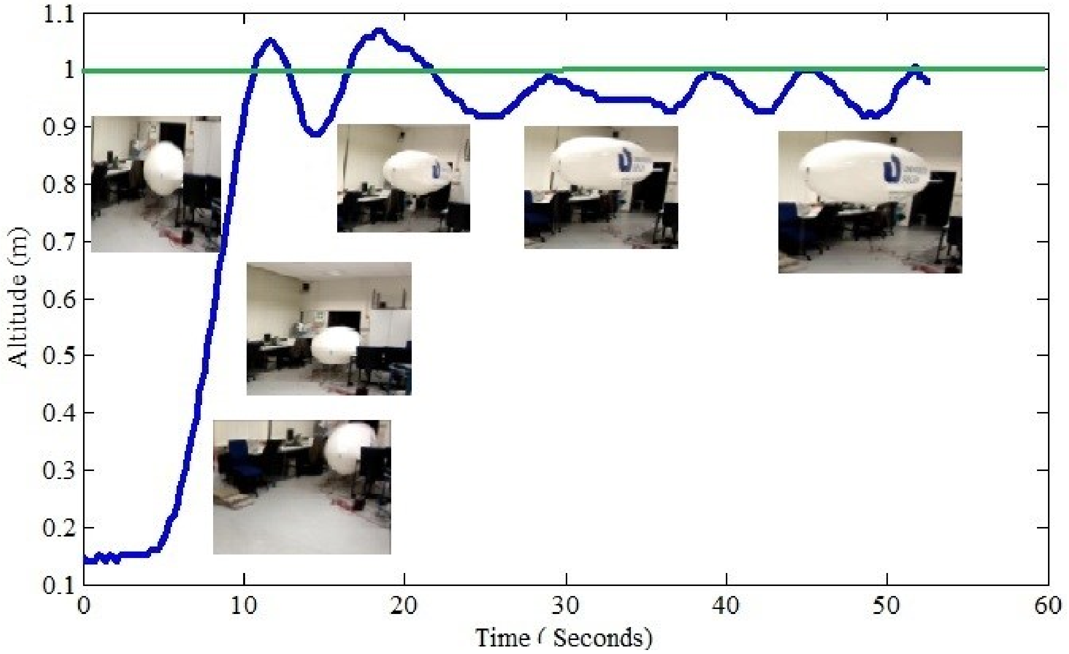


Figure 5. 13 Behavior of fuzzy altitude controller Test1.

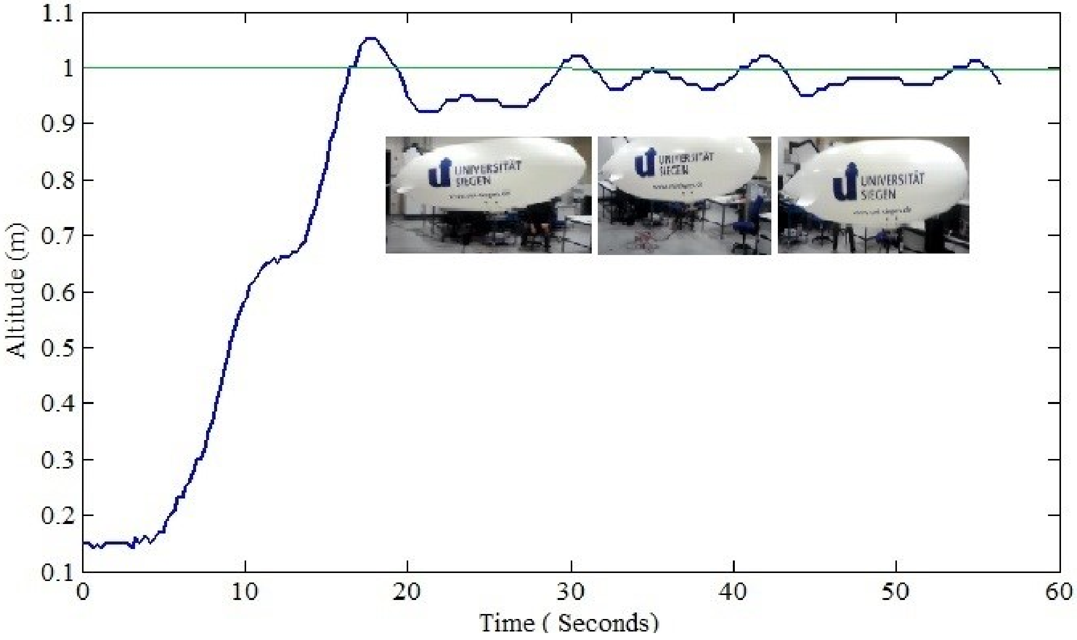


Figure 5. 14 Behavior of fuzzy altitude controller Test2.

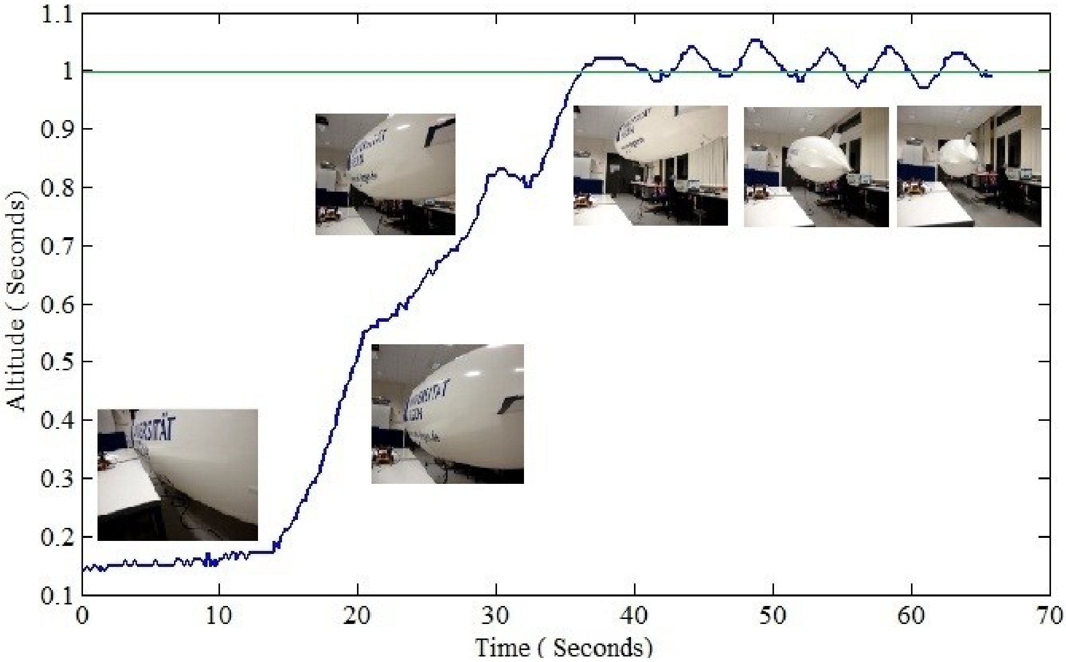


Figure 5. 15 **Behavior of fuzzy altitude controller Test3.**

5.2 Visual Servoing Controller

The visual references on the image plane to directly command the blimp robot to continuously follow aerial object moving target or ground robot target. These controllers use the principles of visual servoing theory. The visual servoing was divided into four main parts. The first one is the SURF detection and localization algorithm which is presented in chapter 4. The second, is the fuzzy sets model to correct the prediction visual information of the SURF outputs, then, provide them to low level control inputs. Finally, visual control which was employed by fuzzy control in order to control the vectorization angle of main propellers and the yaw angle for the blimp robot.

5.2.1 Fuzzy sets model for following Ground Robot.

As it was described in the chapter four, vision system had been designed to follow ground robot target by the blimp. The control strategy here are depending on the visual data which were obtained from the vision sensor to achieve high performance behaviors and to approach the main goals of the blimp by finishing the navigation tasks and missions successfully. The vision feedback was optimized also by the fuzzy sets model to correct the prediction position information, then it is integrated with the blimp's flight control system to guide it to follow the ground robot as it is shown in Fig. 5.16.

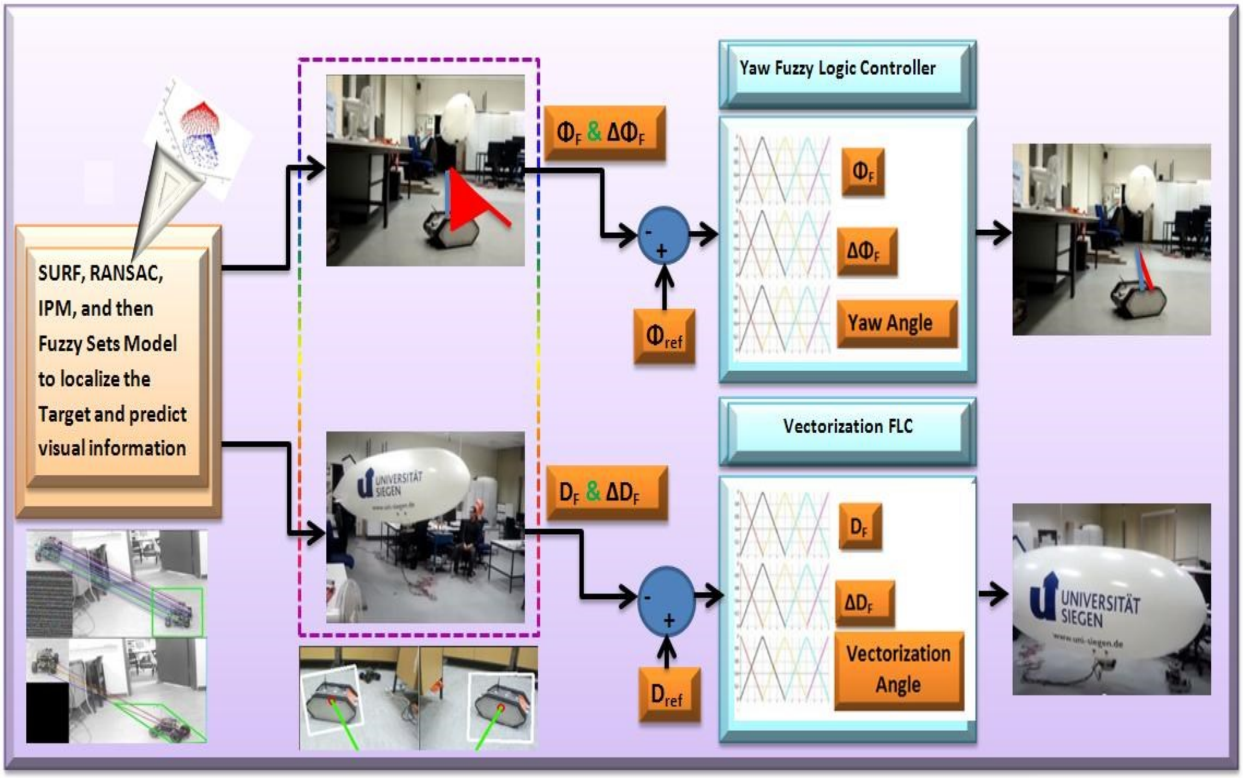

SURF, RANSAC, IPM, and then Fuzzy Sets Model to localize the Target and predict visual information
Φ_F & $\Delta\Phi_F$
Φ_{ref}
Yaw Fuzzy Logic Controller
Φ_F
$\Delta\Phi_F$
Yaw Angle
D_F & ΔD_F
D_{ref}
Vectorization FLC
D_F
ΔD_F
Vectorization Angle
-
+
-
+

Figure 5. 16 **Fuzzy visual servoing controller.**

Therefore, in order to correct the prediction position information from **SURF** algorithm, a fuzzy sets model was needed. Eight tastes and experiments were carried out on the **RST Labs** to study the effect of the distance between target and the camera as well as the angle between both of them as it is shown in Fig. 5.17. Therefore, we collected the general measuring data obtained from the camera for different distances and angles **(r, ϕ).** Then, these information was analyzed to propose possibilities histograms. These histograms will be transferred to fuzzy membership functions and fuzzy sets model. The distance takes range **[20,168]** in pixels and the angle **[-1.571, 1.571]** in radians. In each case the polar coordinate information were recorded. A summary of these experiments and the errors are given in Table 5.6. Each reading denotes by **(r, ϕ).** For simplicity, if we assume that the **r** value is **120** pixels, then we can observe that the target has a distance between **(r+0)** and **(r+2)** or **(r-2)** and **(r+1)**. As we have described previously, the frequency distributions of these data were categorized into four main groups as it is summarized in Table 5.7. The data were categorized based on the range of the errors in pixels. For example, the distances range within [20-40] and [40-60] have approximately the same intervals errors [-5,0] and [-4,0], respectively .Hence, the measurement record set $\boldsymbol{A}$ and random set **S** have been found. Because these data are fuzzy intervals, we could find the core and support of them as it is summarized in Table 5.8.

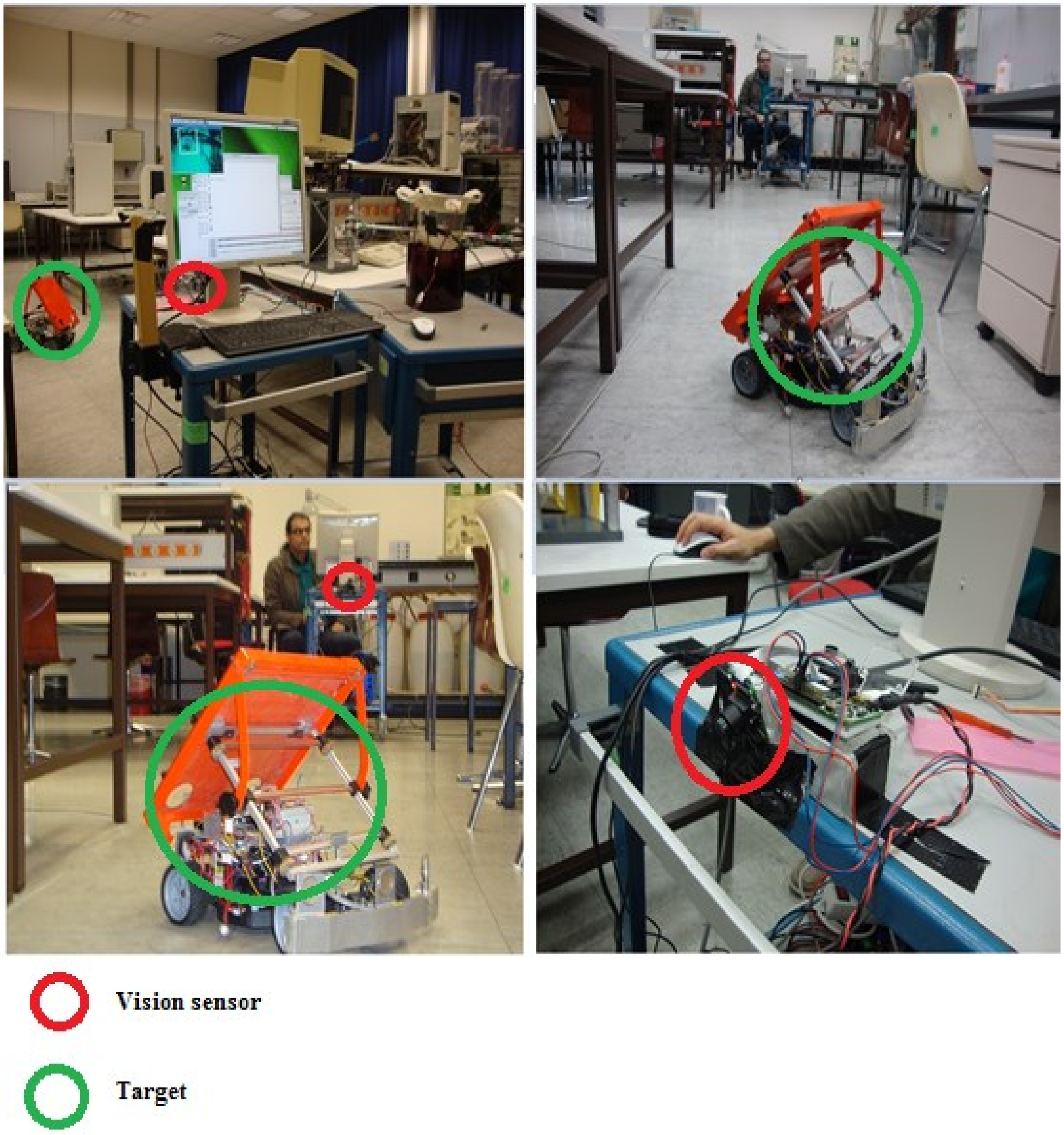


Figure 5. 17 **Experiment to study the camera behaviors.**

r (Pixels)	φ	Error ξ (Pixels)
[20-40]	[-1.571, 1.571]	[-5,0]
[40-60]	[-1.571, 1.571]	[-4,0]
[60-80]	[-1.571, 1.571]	[-3,1]
[80-100]	[-1.571, 1.571]	[-3,1]
[100-120]	[-1.571, 1.571]	[-2,1]
[120-140]	[-1.571, 1.571]	[0,2]
[140-160]	[-1.571, 1.571]	[1,3]
[160-168]	[-1.571, 1.571]	[1,4]

Table 5.6 Error analysis.

Data	A_i	S_i
[20-60]	<[-5,0],[-4,0]>	{[-5,0]=0.5, [-4,0]=0.5}
[60-100]	<[-3,1],[-3,1]>	{[-3,1]=0.5, [-3,1]=0.5}
[100-140]	<[-2,1],[0,2]>	{[-2,1]=0.5, [0,2]=0.5}
[140-168]	<[1,3],[1,4]>	{[1,3]=0.5, [1,4]=0.5}

Table 5.7 Frequency data analysis

E_i^L	E_i^R	$C_i(\pi)$	$Supp_i(\pi)$
{-5,-4}	{0,0}	{-4,0}	{(-5,-4), (-4,0)}
{-3,-3}	{1,1}	{-3,1}	{(-3,1)}
{-2,0}	{1,2}	{0,1}	{(-2,0), (0,1), (1,2)}
{1,1}	{3,4}	{1,3}	{(1,3), (3,4)}

Table 5.8 Possibilities data analysis

The possibilities histograms for the data as shown in Fig. 5.18-top can be transferred to fuzzy membership functions as shown in Fig. 5.18-down without any changes.

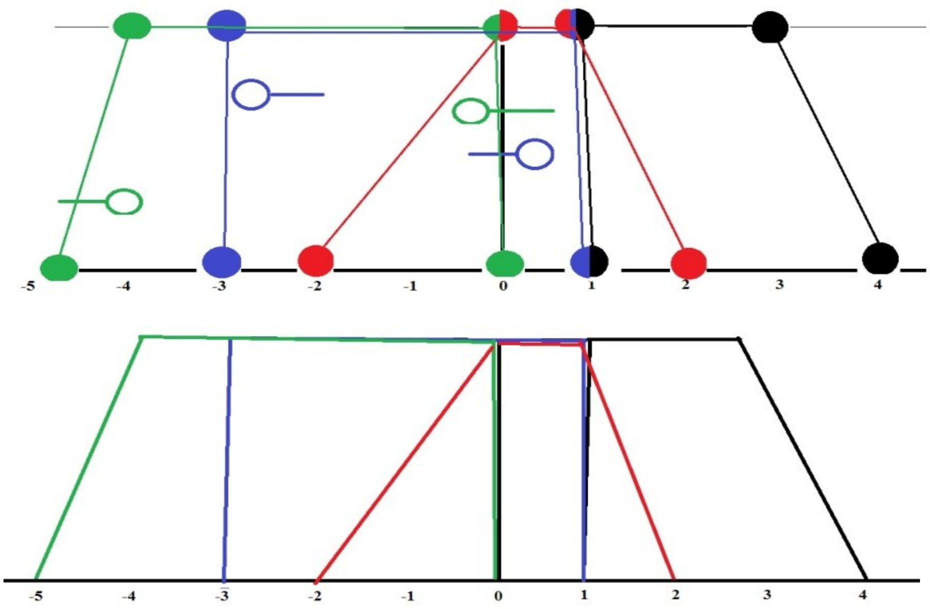


Figure 5. 18 **Possibilities distribution and membership functions.**

The experimental results in Fig. 5.19 to Fig. 5.22 indicate that the fuzzy sets model could correct the prediction position information for the **SURF** algorithm slightly. However, the model is pretty good in optimization when the values of prediction are outliers and have no sense. Fig. 5.19 was obtained from 500 frames (images) to find 500 readings for the distance **r** (in pixels). The distance here was fixed distance which is the mean value. Fig. 5.20 and Fig. 5.21 indicate that the fuzzy sets model could correct the predicted values of the distance **r** while the target was moving away from the vision sensor. **Range 1** is the minimum value in pixels that the vision sensor can read and **Range 2** is the maximum value. Fig. 5.22 is illustrating the difference between **SURF** algorithm output and the **SURF** optimized output by fuzzy sets model that aims to correct the distance and angle values which are obtained from the **SURF**. It is clearly obvious that the fuzzy sets model could correct the information especially when they are outliers as it is shown in point A, B, and C in this figure.

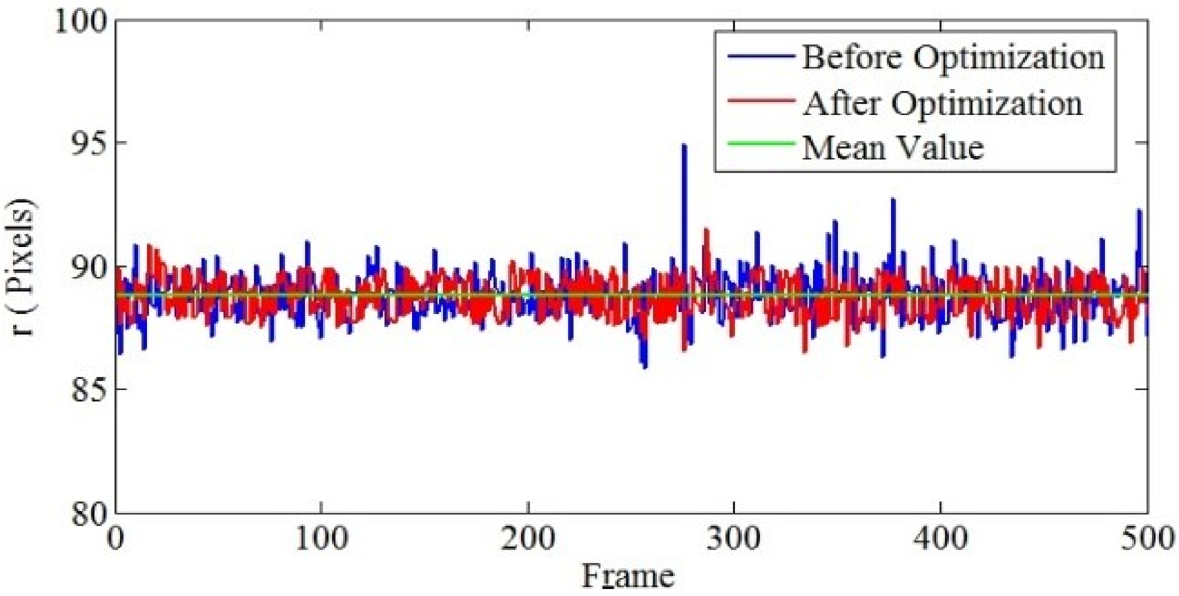


Figure 5. 19 Test results for fuzzy set model for fixed distance.

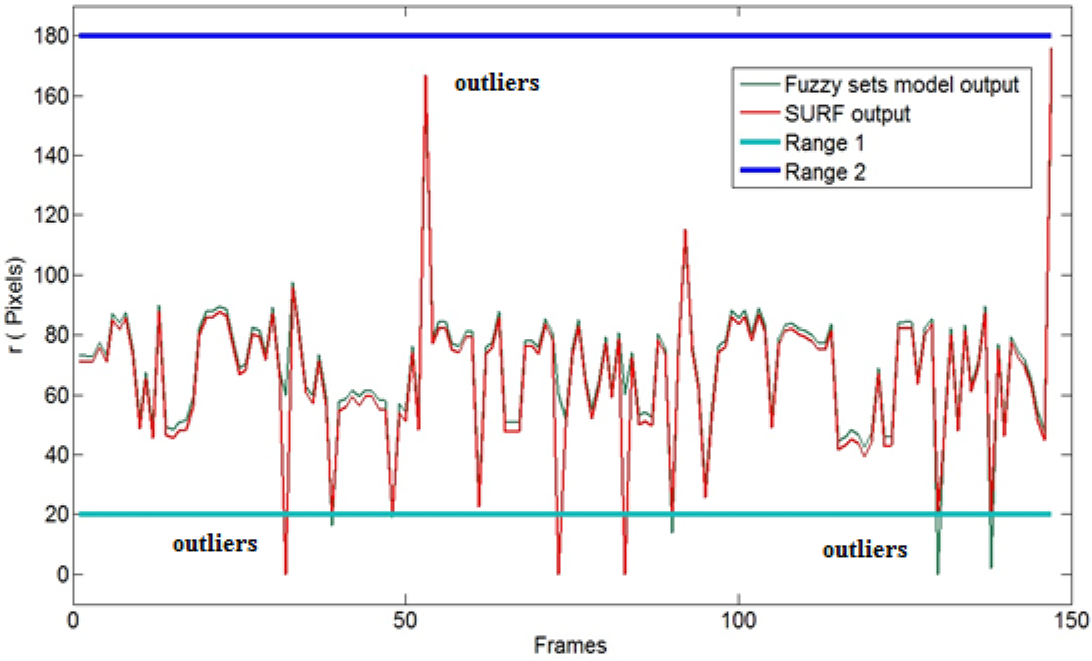


Figure 5. 20 Test 1 results for fuzzy set model for different distances (r).

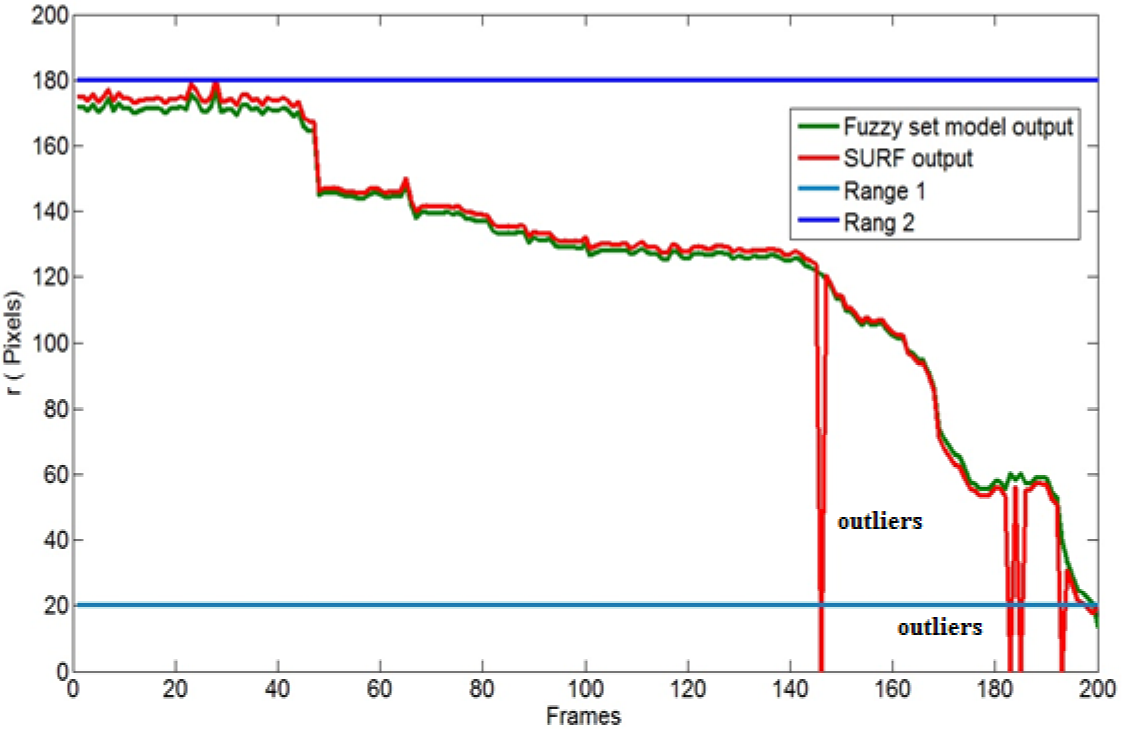


Figure 5. 21 Test 2 results for fuzzy set model for different distances (r).

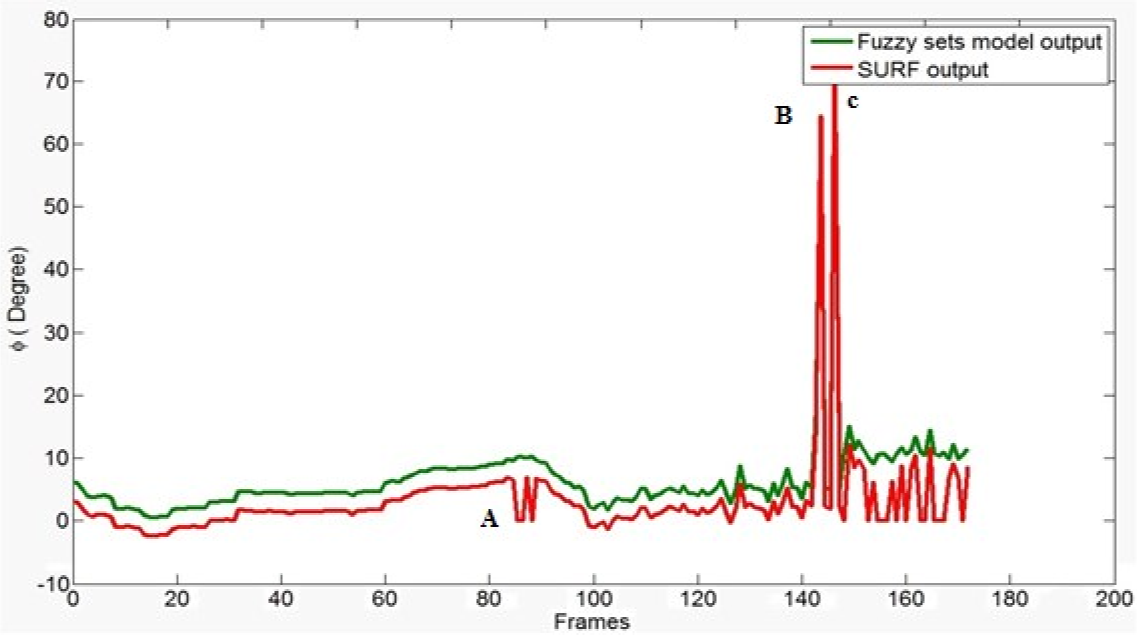


Figure 5. 22 Test results for fuzzy set model for different angles.

In order to verify the proposed vision system, several experiments of the complete system were conducted. Some examples of the target tracking are shown in Fig. 5.23, Fig. 5.24, and Fig. 5.25. After the vision algorithm detects the target and estimates the inputs for the control, the controller could change the vectorization angle depending on the distance between the target projection point and the blimp projection point on the image plane. In Fig. 5.23 and Fig. 5.24 when the blimp detects and identifies the target which is away from the blimp 100 pixels (approximately 200cm) on the image coordinate, the blimp will change the vectorization angle and moves toward the target to reduce the distance between both of them. When the distance becomes 50 pixels (approximately 100cm), the blimp robot will stop and the vectorization angle will get 90 degree. Fig. 5.25 is illustrating the test when the blimp flies to the target which is away 200 pixels (400cm) and stops when it reaches a distance about 200cm. The green lines in these figures show the desired distance that we want the blimp to achieve when it gets close to the target and the red lines are the distance measured by the vision sensor. The angle between the target projection point, blimp and the image's centre is shown in Fig. 5.26. The target was moving in the environment and the blimp robot tries keeping the target at the center of the image which have been done approximately after 15 seconds from the starting of the mission.

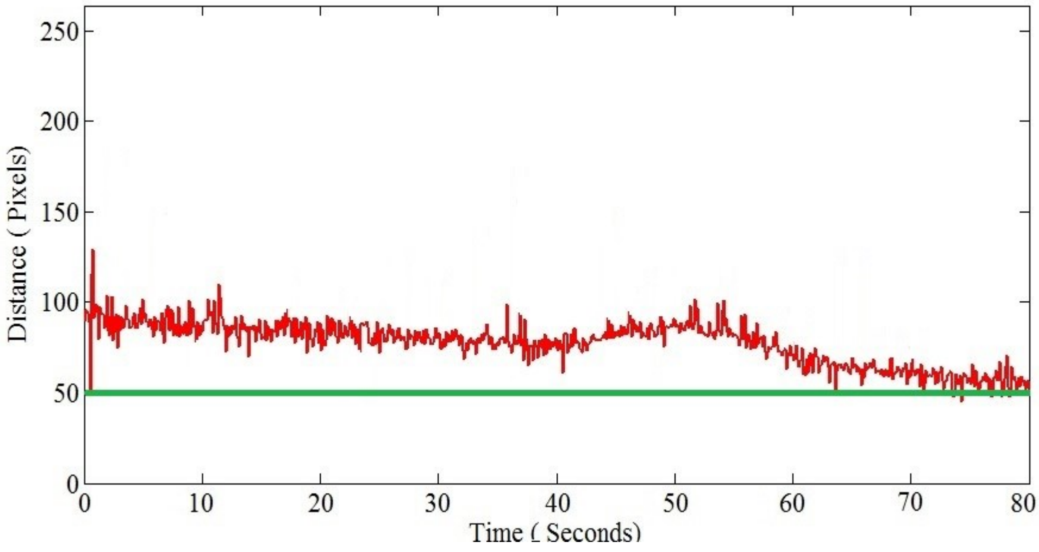


Figure 5. 23 **Distance between target and blimp.**

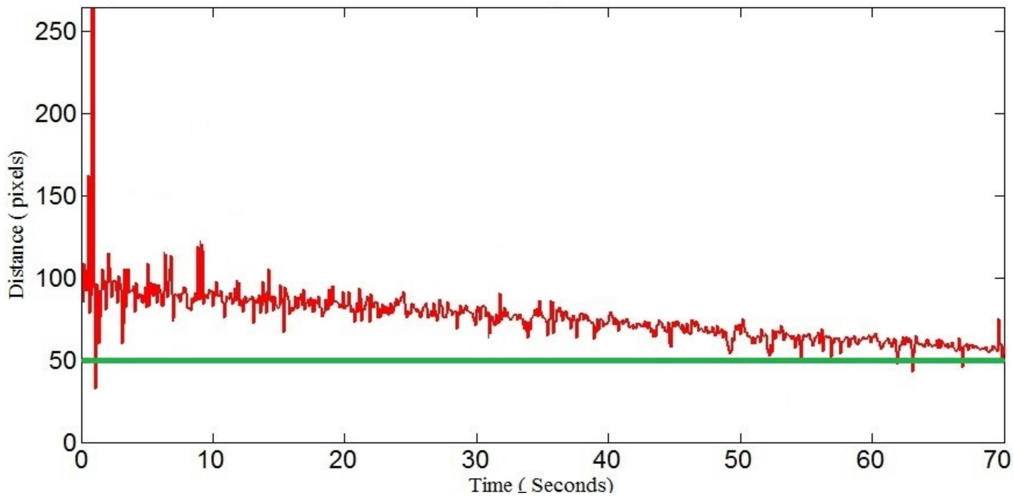


Figure 5. 24 Distance between target and blimp.

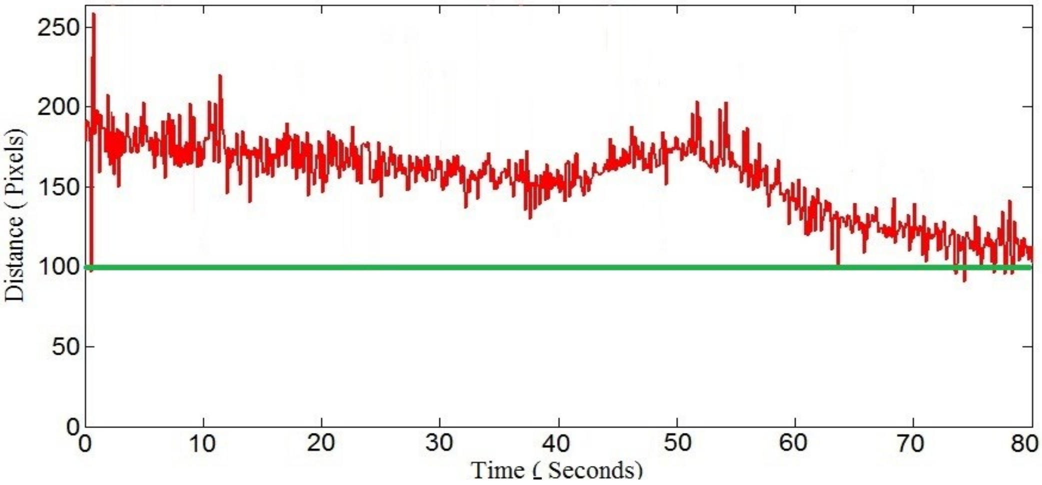


Figure 5. 25 Distance between target and blimp.

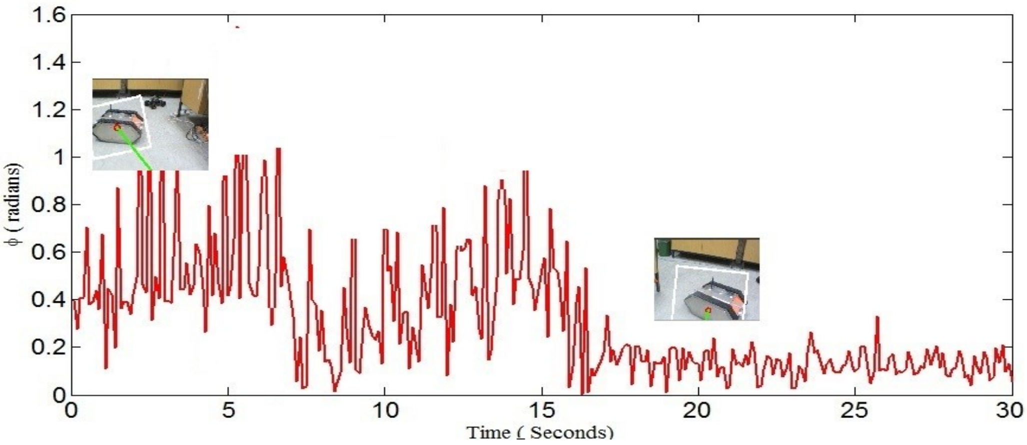


Figure 5. 26 Angle between target, blimp and image's centre

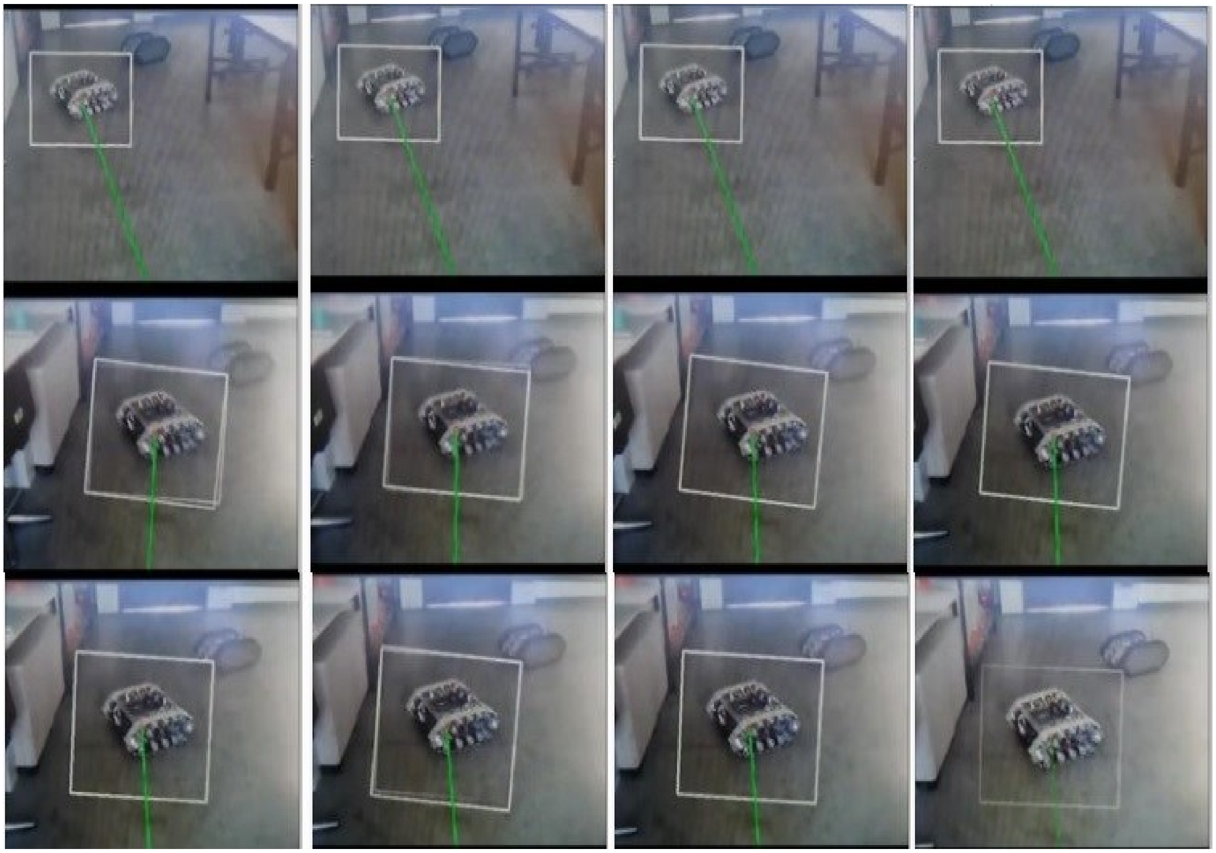


Figure 5. 27 The sequence of images to detect ground robot.

The sequence of images is taken from multi videos while the blimp robot detects, follows or lands on the ground robot are shown in Fig. 5 .27- Fig. 5. 31. Fig. 5. 27 shows the behavior of the **SURF** algorithm in order to identify, detect the ground robot target. Then, it uses homography to draw a box on the target object in order to find the center of the ground robot and the projection point of the camera on image plane then find the distances. It is clear obvious that the algorithm is stable and robust even the target distance and orientation are changing.

In Fig. 5. 28 and Fig. 5. 29 the sequence of images is taken during the mission for the blimp robot to identify and track the target. When the blimp detects the target, it will change the orientation (Yaw angle) to keep the ground target at the center of the image plane then the controller will change the vectorization angle in order to fly toward the target. It is clear from this sequence of images that the blimp succeeds reaching the target and keeps it in a certain distance.

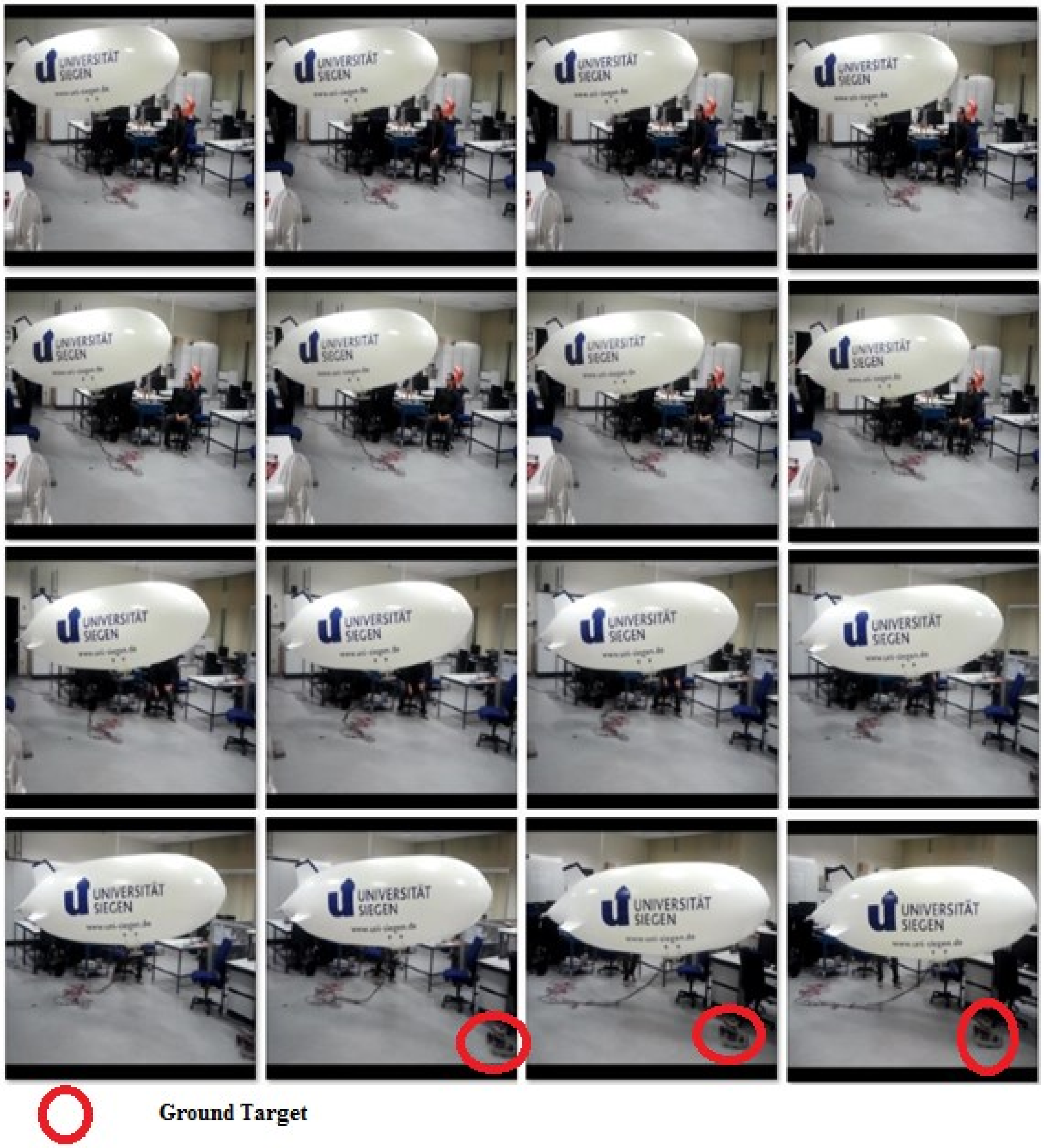


Figure 5. 28 **Blimp move to the detected target image sequences.**

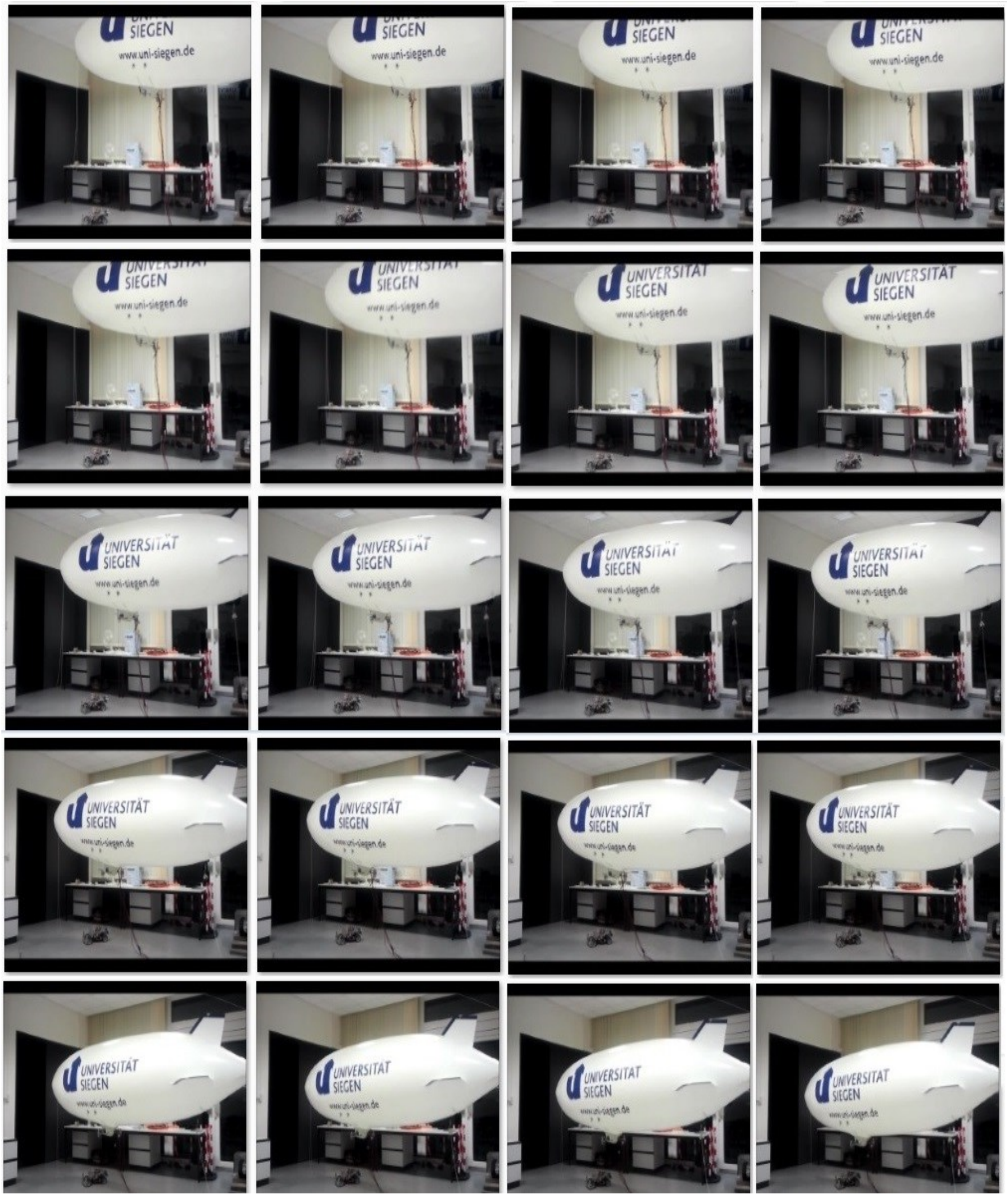


Figure 5. 29 **Blimp robot detect ground robot.**

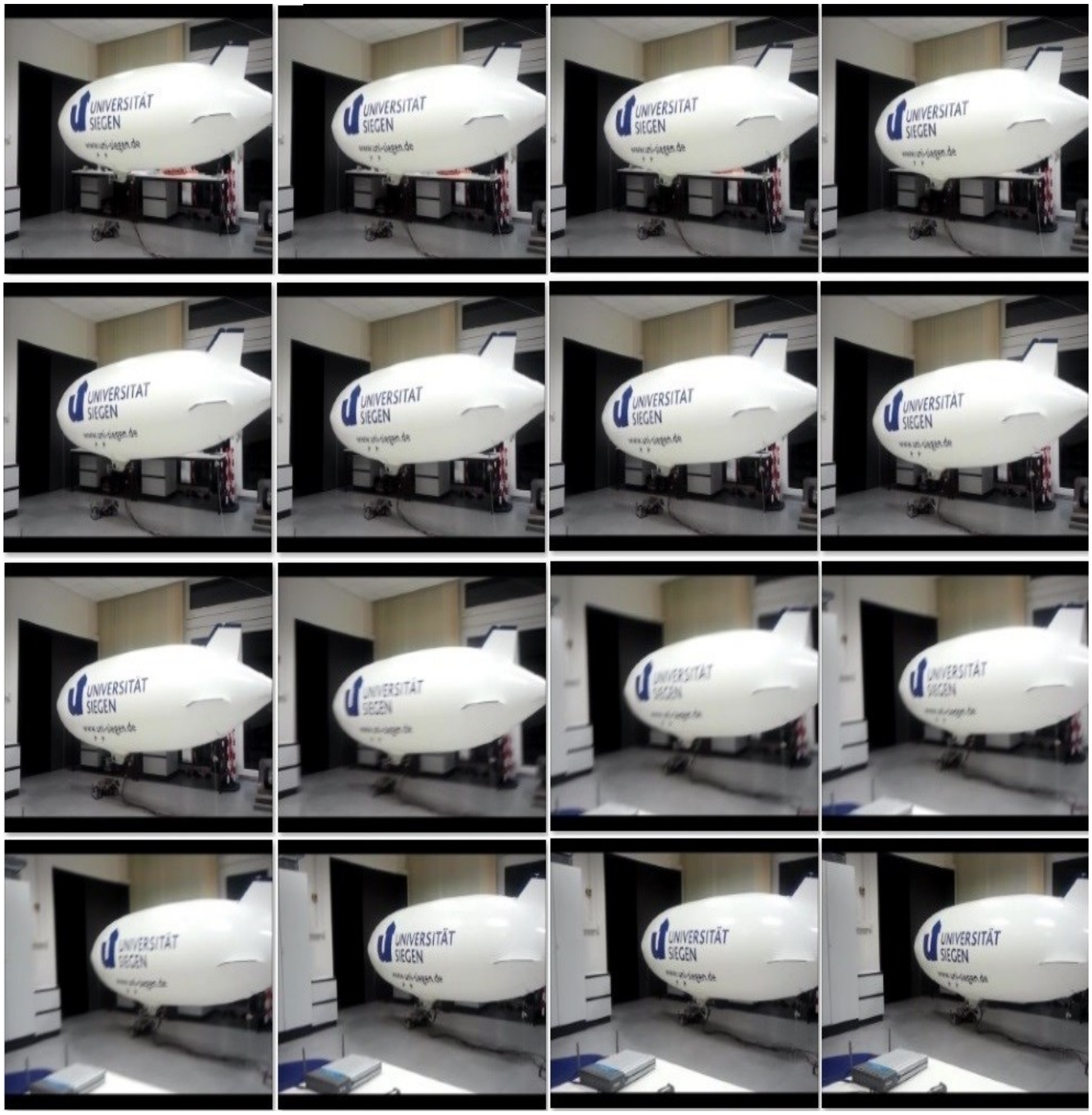


Figure 5. 30 **Blimp robot lands on the ground robot.**

In addition, Fig. 5.30 shows the behavior of the blimp robot for the landing mission. When the blimp robot controllers guide it to follow and reach the target, the landing will start and the blimp robot lands over the target as it is obvious shown in this figure. In Fig. 5. 31, the sequence of images is taken during the mission for the blimp robot to identify and track the target. When the blimp detects the target, it will change the orientation (Yaw angle) to make the ground target at the center of the image plane by using the visual Yaw fuzzy servoing controller. It is clear from the sequence of images that the blimp succeeds to change the orientation (Yaw angle) after it reaches its desired altitude.

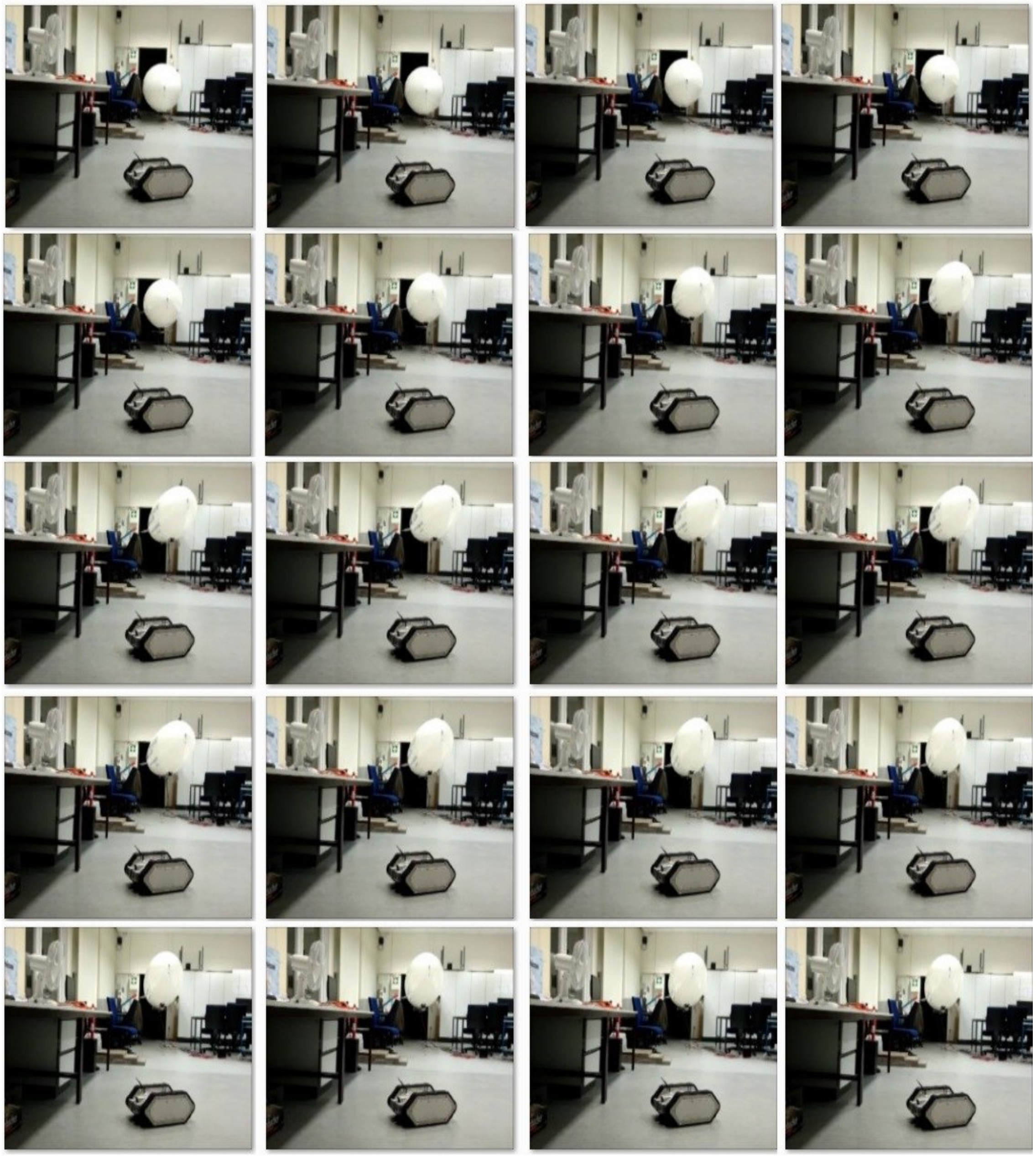


Figure 5. 31 Sequences image for changing Yaw angle for the blimp.

5.2.2 Visual Fuzzy Control for Blimp to Follow 3D Aerial Object

The visual servoing system in order to follow a 3-D aerial object by blimp robot was presented. First, the fuzzy Yaw controller has been presented to the blimp robot with two inputs and one output. The first input for this control is the angle computed value that is the differed between the image center and the object center in the projection plane. The second input is the difference between the last two computed value (last two frames or last 2 images). The output of this controller is the voltage command for the rear motor for changing the heading position of the blimp. Six experiments were done to study the effect of the estimated angles on the rear voltage. The angle here is between the center of the image and the center of the projection point of the object on the image. The general measuring data were collected for different angles, then we analyzed these data to propose fuzzy sets model by using frequency and possibilities distribution. In each case the voltage information was recorded as well as the angle values. A summary of these experiments are given in Table 5.9. Then, the mathematical study and analyzing have been done on the obtained data based on the frequency distributions of these information, then they are categorized into five main groups as it is summarized in Table 5.10.

Angle	Rear Voltage	The Vector	The Sets
[-90,-80]	[-1000, -820]	<-1000, -820>	{[-1000, -820]}
[-80,-55]	[-814, -451]	<-814, -451>	{[-814, -451]}
[-55,-30]	[-495, -257]	<-495, -257>	{[-495, -257]}
[-30,-5]	[-281, -15]	<-281, -15>	{[-281, -15]}
[-5,5]	[-10, 6]	<-10, 6>	{[-10, 6]}
[5,30]	[10, 246]	<10, 246>	{[10, 246]}
[30,55]	[257, 478]	<257, 478>	{[257, 478]}
[55,80]	[575, 772]	<575, 772>	{[575, 772]}
[80,90]	[880, 1000]	<880, 1000>	{[880, 1000]}

Table 5.9 The empirical data.

Data	A_i	S_i
[-90,-55]	<[-1000, -820], [-814, -451]>	{[-1000, -820] = 0.5, [-814, -451]= 0.5}
[-55,-5]	<[-495, -257], [-281, -15]>	{[-495, -257] = 0.5, [-281, -15]= 0.5}
[-5, 5]	<[-10, 6]>	{[-10, 6]=1}
[5, 55]	<[10, 246], [257, 478] >	{[10, 246] = 0.5, [257, 478]= 0.5 }
[55, 90]	< [575, 772], [880, 1000]>	{ [575, 772] = 0.5, [880, 1000] = 0.5}

Table 5.10 Frequency Distribution of the data.

E^L	E^R	Core	Support
[-1000, -814]	[-820, -451]	[-814, -820]	{(-1000, -814), (-814, -820), (-820, -451)}
[-495, -281]	[-257, -15]	[-257, -281]	{(-495, -281), (-281, -257), (-257, -15)}
[-10, 6]	[-10, 6]	[-10, 6]	{(-10,6)}
[10, 257]	[246, 478]	[246, 257]	{(10, 257), (246,257), (246, 478)}
[575, 880]	[772, 1000]	[772,880]	{(575, 880), (772,880), (772, 1000)}

Table 5.11 Histograms parameters.

The possibilities histograms for the data as given in Table 5.11 and shown in Fig. 5.32 can be transferred to fuzzy membership functions as shown in Fig. 5.33.

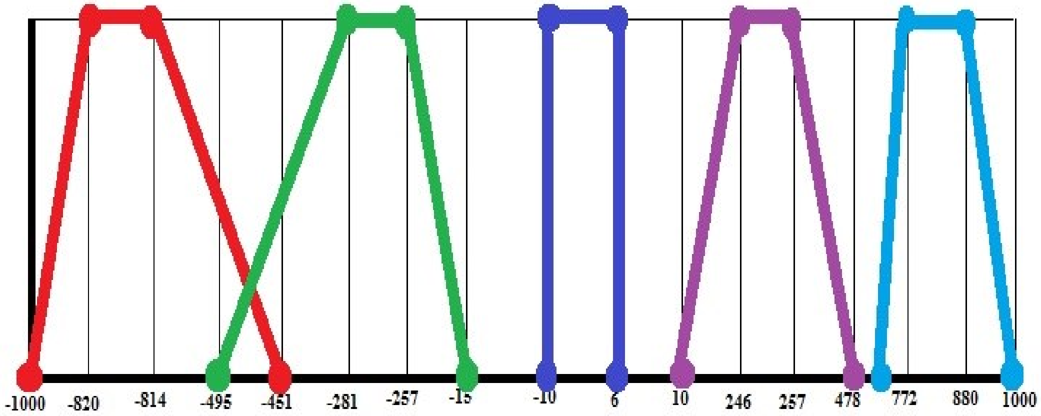


Figure 5. 32 **The possibilities histograms for the yaw output.**

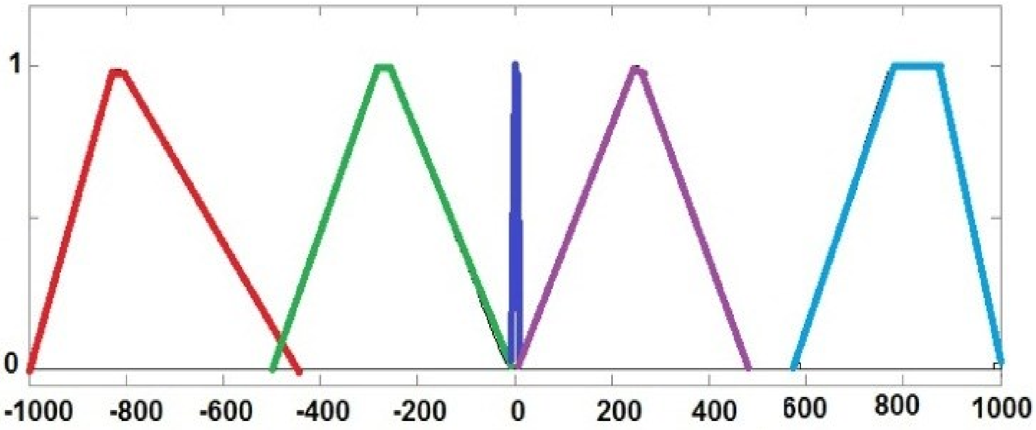


Figure 5. 33 **Fuzzy membership Functions.**

The optimized and final membership functions are shown in Fig. 5.34. The optimization was done by using bacterial algorithm which will change the breakpoints for the left and right side of the trapezoidal membership functions. For example, the red membership function in Fig.

5.33 has four breakpoints [-1000, -820, -814, -451] and in Fig. 5.34 they became [-1000, -820, -814, -185].

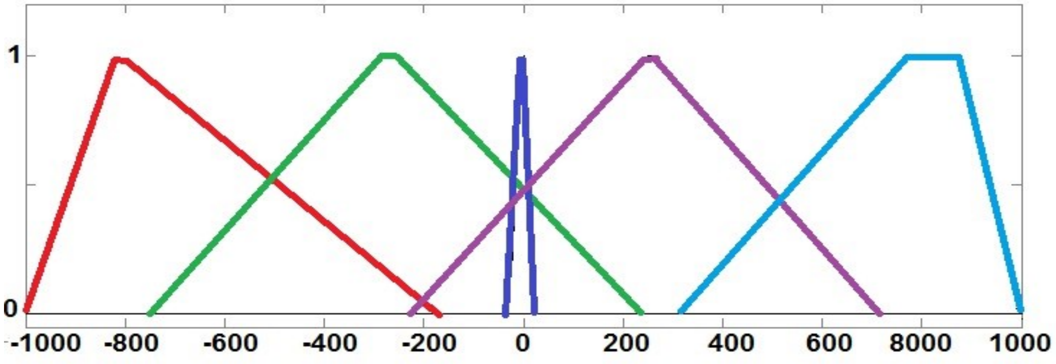


Figure 5. 34 Optimized membership functions for the output.

Second controller is the vectorization fuzzy controller. For this controller, the first input is the estimated distances between the blimp and the 3D object target. The second input is the difference between the last 2 computed values or last two measures. Then, the output is the vectorization angle for the main propellers. Same procedures were done here in order to design the fuzzy membership functions for the vectorization angle by studying the effect of the distances. The empirical data, the analysis as well as the histograms parameters are given in Tables 5.12, 5.13, and 5.14. The possibilities histograms, fuzzy membership functions as well as the optimized membership are shown in Fig. 5.35, 5.36 and 5.37, respectively.

First input (Distance)	Vectorization Voltage	The Vector	The Sets
[120,160]	[-232,-98]	<-232,-98>	{[-232,-98]}
[160,200]	[-114,5]	<-114,5>	{[-114,5]}
[200,240]	[-17, 30]	<-17, 30>	{[-17, 30]}
[240,280]	[0, 123]	<0, 123>	{[0, 123]}
[280,320]	[100, 204]	<100, 204>	{[100, 204]}
[320,360]	[228, 317]	<228, 317>	{[228, 317]}
[360,400]	[325, 435]	<325, 435>	{[325, 435]}
[400,440]	[474, 552]	<474, 552>	{[474, 552]}
[440,480]	[664, 720]	<664, 720>	{[664, 720]}

Table 5.12 The empirical data.

Data	A_i	S_i
[120,200]	<[-232,-98], [-114,5]>	{[-232,-98] = 0.5, [-114,5]=0.5}
[200,240]	<[-17, 30]>	{[-17, 30] = 1}
[240,320]	<[0, 123], [100, 204] >	{[0, 123] = 0.5, [100, 204]= 0.5}
[320, 400]	<[228, 317], [325, 435]>	{[228, 317]= 0.5, [325, 435] = 0.5}
[400, 480]	<[474, 552], [664, 720]>	{[474, 552] = 0.5, [664, 720] = 0.5}

Table 5.13 Frequency distribution of the data.

E_L	E_R	Core	Support
[-232, -114]	[-98, 5]	[-98,-114]	{(-232, -114), (-98,-114), (-98, 5)}
[-17, 30]	[-17, 30]	[-17, 30]	{(-17, 30), (-17, 30), (-17, 30)}
[0, 100]	[123, 204]	[123, 100]	{(0, 100), (123, 100), (123, 204)}
[228, 325]	[317, 435]	[317,325]	{(228, 325), (317,325), (317, 435)}
[474, 664]	[552, 720]	[552, 664]	{(474, 664), (552, 664), (552, 720)}

Table 5.14 Histograms parameters.

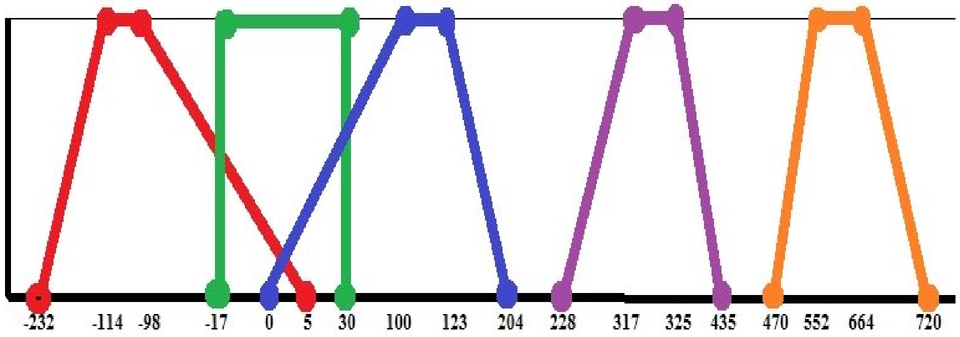


Figure 5. 35 The possibilities histograms for the vectorization output.

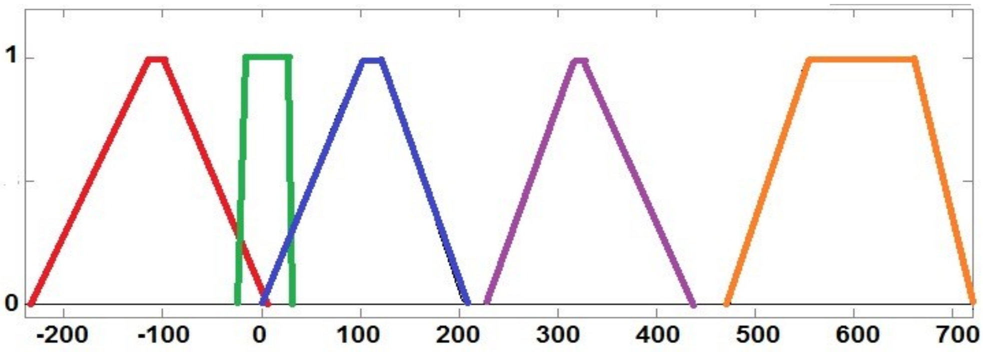


Figure 5. 36 The membership functions for output.

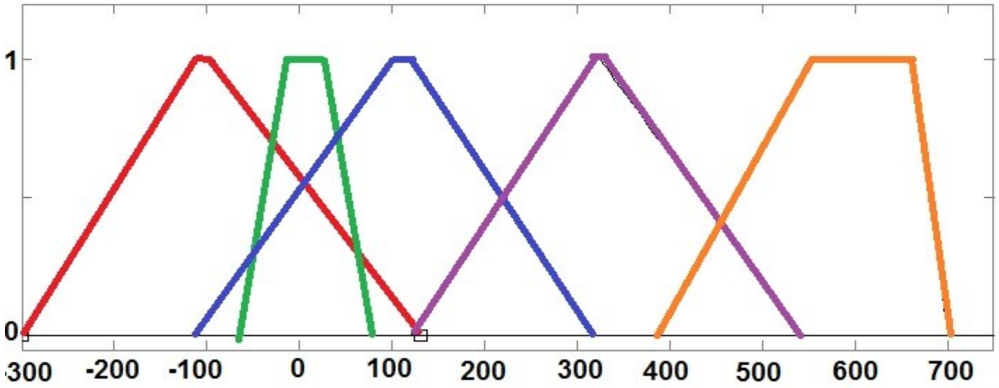


Figure 5. 37 The Final output membership functions.

The blimp vision system detects and flies following the target. Then, the engine will stop and the blimp stop moving at a certain distance. The distance between the blimp and the target is shown in Fig. 5.38 and Fig. 5.40. When the blimp detects the target it will fly toward the target to reduce the distance from 450cm to 100cm and 120cm as it is shown in Fig. 5.38 and Fig. 5.40, respectively. Fig. 5.39 and Fig. 5.41 show the blimp behavior in order to control the yaw angle and keep the object at the center of the image. Also, the sequence of images for the experiments is shown in Fig. 5.42. During this sequence of images the blimp robot could identify and detect the 3D object which is here another gondola fixed at a certain distance (Right side). When the blimp detects the target, it will change the orientation (Yaw angle) to keep the 3D object target at the center of the image plane then the controller will change the vectorization angle in order to fly toward the target. It is clear from this figure that the blimp succeeds to reach the target and to keep it in a certain distance. On the Top-Left side, 3D aerial object (Gondola) was moving from place to another while the blimp robot keeps to detect and to fly toward this object during the mission. Fig. 5.43 is illustrating the Gondola (3D aerial object) in case it is mounted and fixed at a known height and certain distance or in case it is carrying by hands to check the validity of the algorithm and the controller when the object is moving.

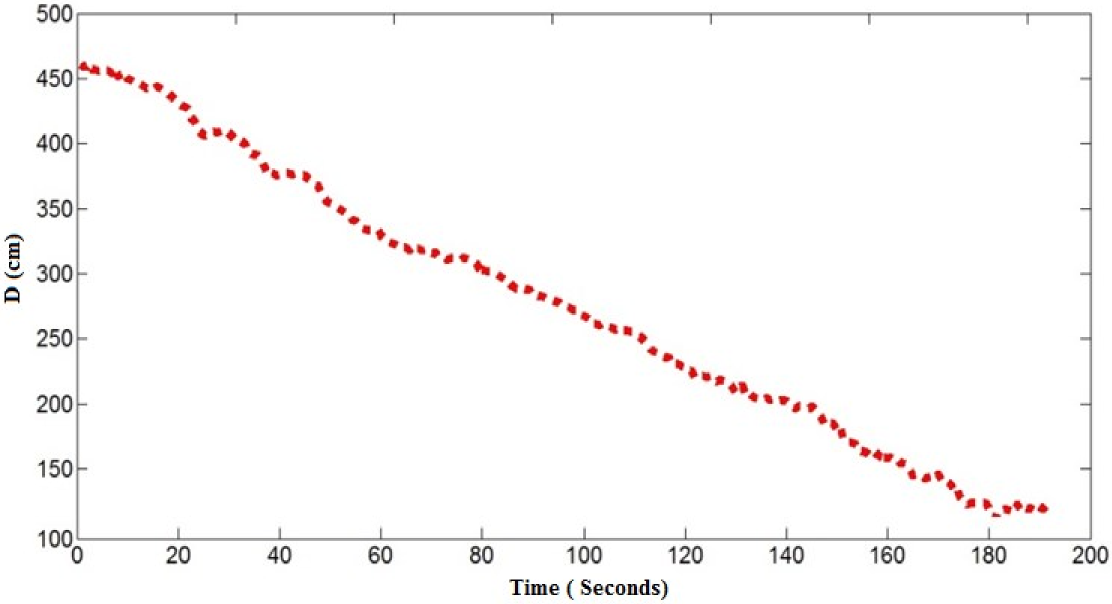


Figure 5. 38 **Estimated distances between the blimp and the target.**

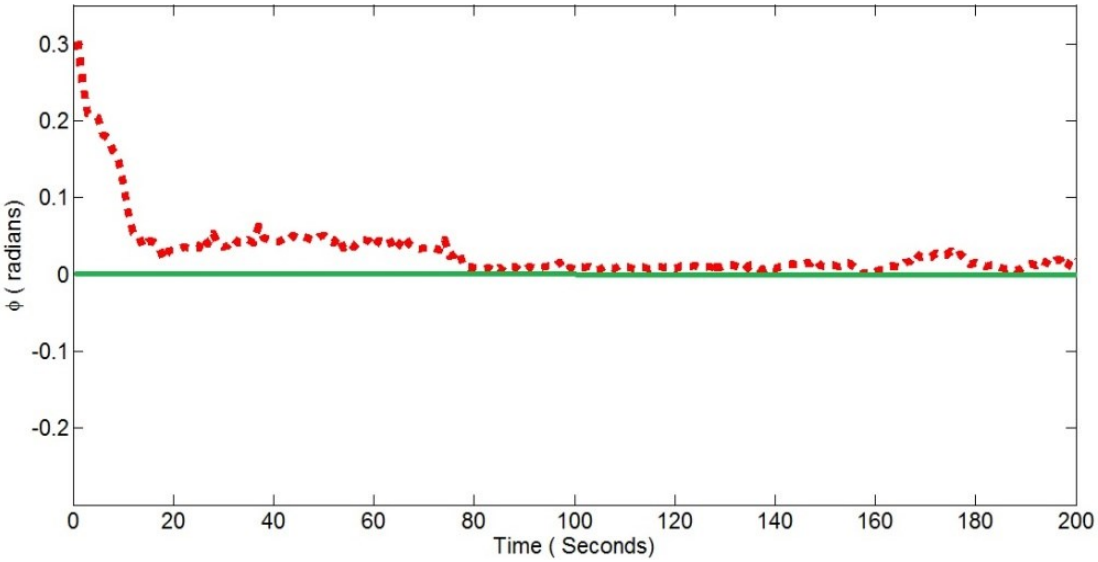


Figure 5. 39 **Angle between target, blimp and image's centre.**

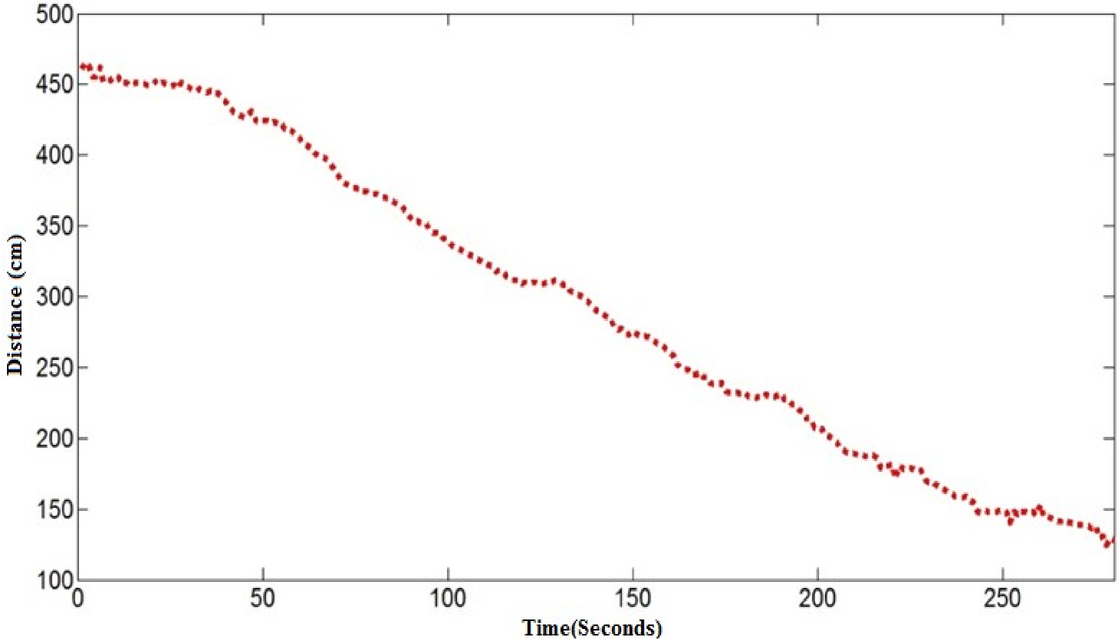


Figure 5. 40 **Estimated distances between the blimp and the target.**

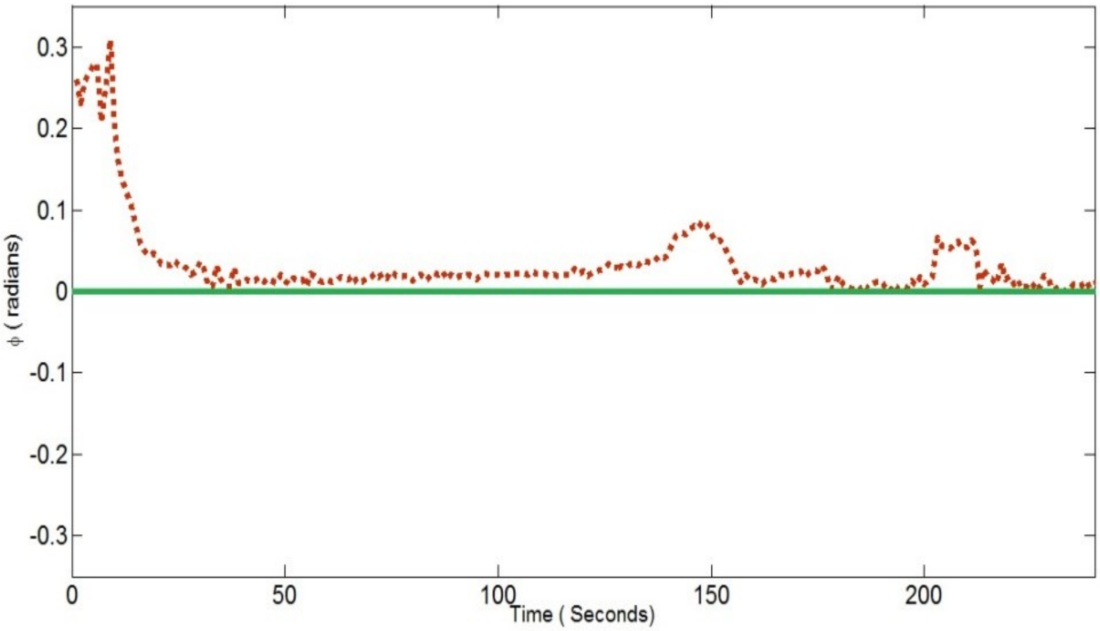


Figure 5. 41 **Angle between target, blimp and image's centre.**

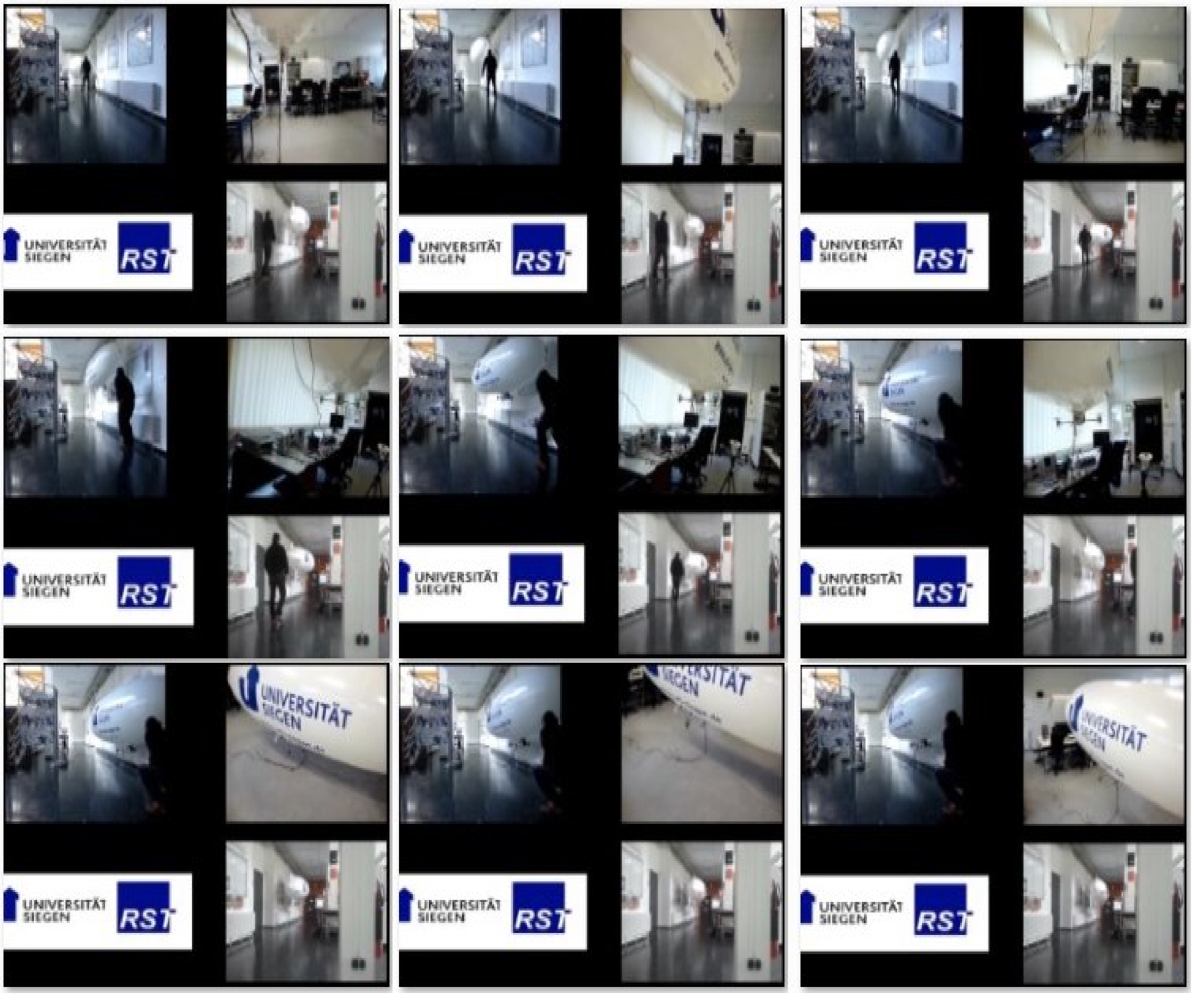


Figure 5. 42 Sequences images of the experiment.

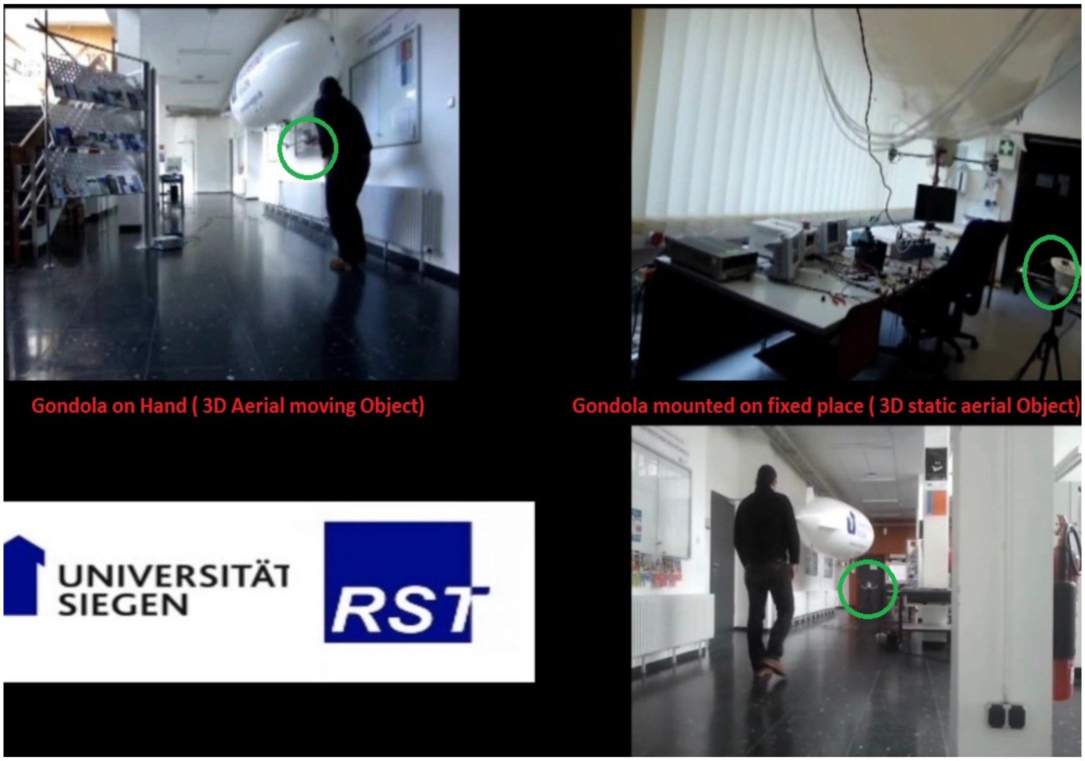

Gondola on Hand (3D Aerial moving Object)
Gondola mounted on fixed place (3D static aerial Object)
UNIVERSITÄT SIEGEN
RST

Figure 5. 43 3D aerial object (fixed in right side, moves in left side).

5.3 Localize the multi targets

Target localization is one of the most important aspects in robot researches. Since it is not possible to estimate target height (in case the targets are aerial robots) with single sensor due to observability problem, we combined the information from two or more sensors. In order to study the effect of number of sensors and minimize the number to localize the robots targets in the environment, the number of sensors was changed from 5 to 17 square sensors as it was described in Chapter 3. As it is shown in Fig. 5.44- Fig. 5.48, whilst the number of sensors increased, the estimated locations are increased. In Fig. 5.44 the two targets have real locations that are plotted by red color for 3 different locations. In this estimation process the number of sensors was 25 sensors (5 square sensors). For target 1 the locations are {(25, 25),(5, 5),(10, 15)} whilst the estimated locations are the green lines which are {(26.07, 27.74),(4.631, 12.23),(6.997, 12.93)}. For target 2 the locations are {(-25, -25),(-5, 0),(-5, 5)} whilst the estimated locations are the green lines which are {(-26.47, -25.82),(-3.407, 1.327),(-2.891, 4.507)}. It is clear that with this number of sensors the estimations are pretty good but have the problem to estimate the locations with high performance.

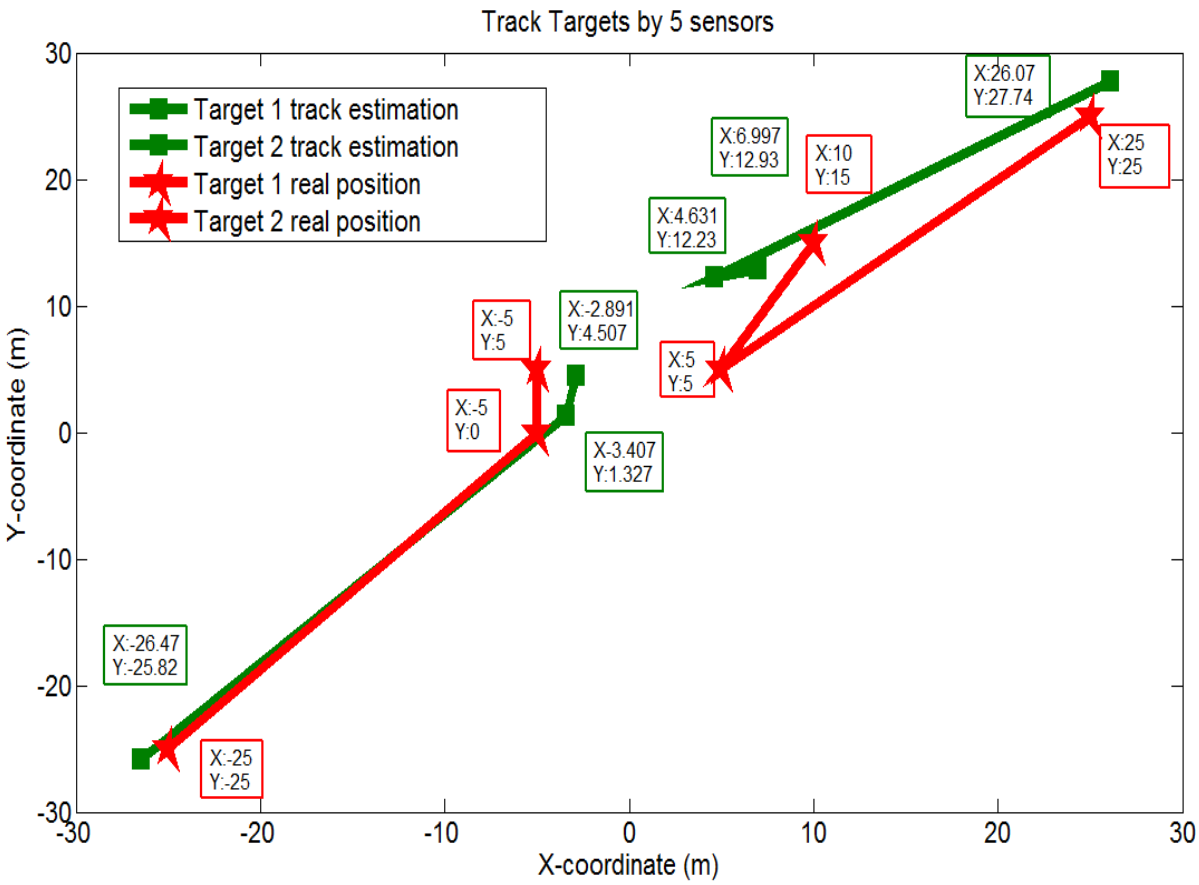


Figure 5. 44 Track two targets by 5 sensors.

In Fig. 5.45 the two targets have real locations that are plotted by red color for 3 different locations. In this estimation process the number of sensors was 64 sensors (8 square sensors). For target 1 the locations are {(25, 25),(5, 5),(10, 15)} whilst the estimated locations are the green lines which are {(25.39, 29.54),(6.598, 4.715),(11.5, 13.42)}. For example, the error in target 1 location is {0.39, 0.54} for location one. For target 2, the locations are {(-25, -25),(-5, 0),(-5, 5)} whilst the estimated locations are pretty good. It is clear that with this number of sensors the estimations are pretty good and better than using 5 square sensors but still there are some errors in the estimations as it could be seen from the figure.

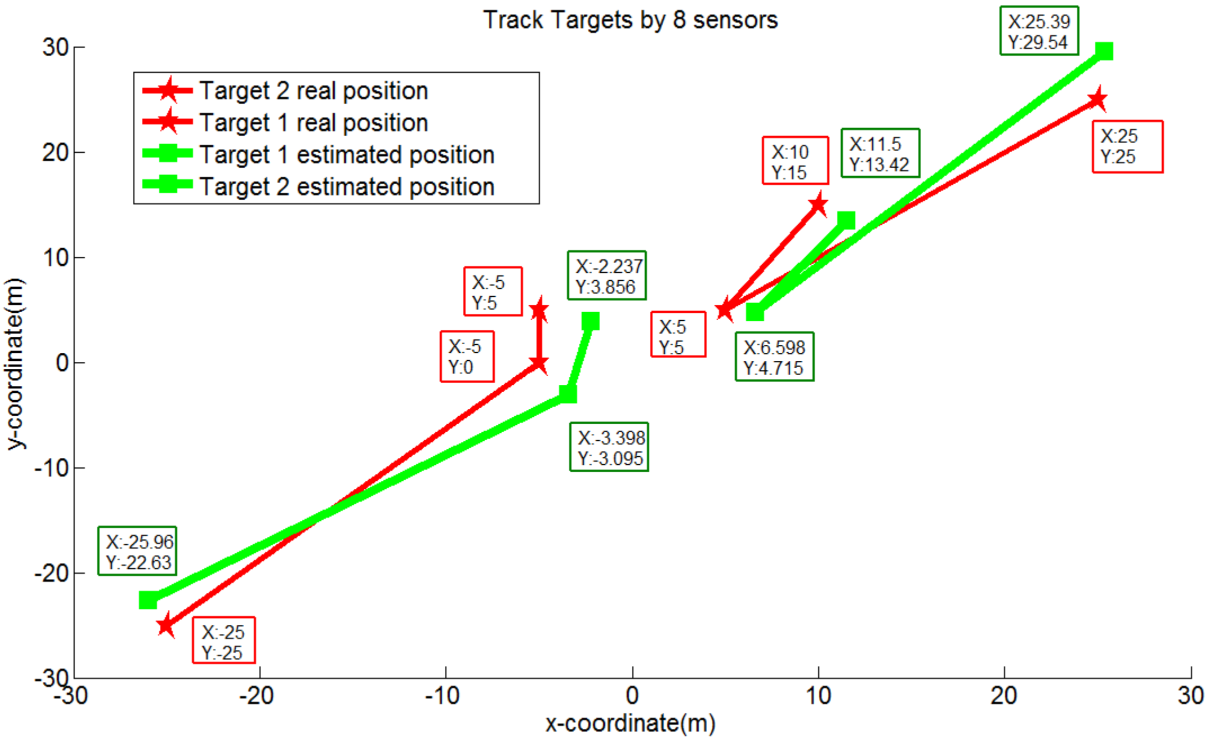


Figure 5. 45 **Track two targets by 8 sensors.**

In Fig. 5.46 the estimation process has been done on the same 3 different locations for each target. The difference here is that the Likelihood estimation has been processed by using 11 square sensors (121 sensors were deployed on the ground). It is clear from this figure that the estimation is pretty good especially in finding the locations for each target in location 1. However, the estimator still has some slight errors in estimating all locations with high performance and efficiency. It can obviously notice that the location 3 for the target 2 supposed to be at (-5, 5), but the estimator has found that the location is (-7.27, 4.6). Moreover, Fig. 5.47 is illustrating the estimated locations when we increased the number of sensors to 14 square sensors (196 sensors). The estimated locations were pretty better than before and the efficiency of the maximum likelihood estimator is increased. However, the third location for target 1 (10, 15) has been estimated to be (5.792, 15.18). In addition, in order to increase the efficiency for estimating the two targets the number of sensors that have been deployed on the ground have been increased to 17 square sensors (289 sensors) as it is shown in Fig. 5.48. It has been found that the estimated locations are pretty much better with this number of sensors and we can achieve pretty good locations.

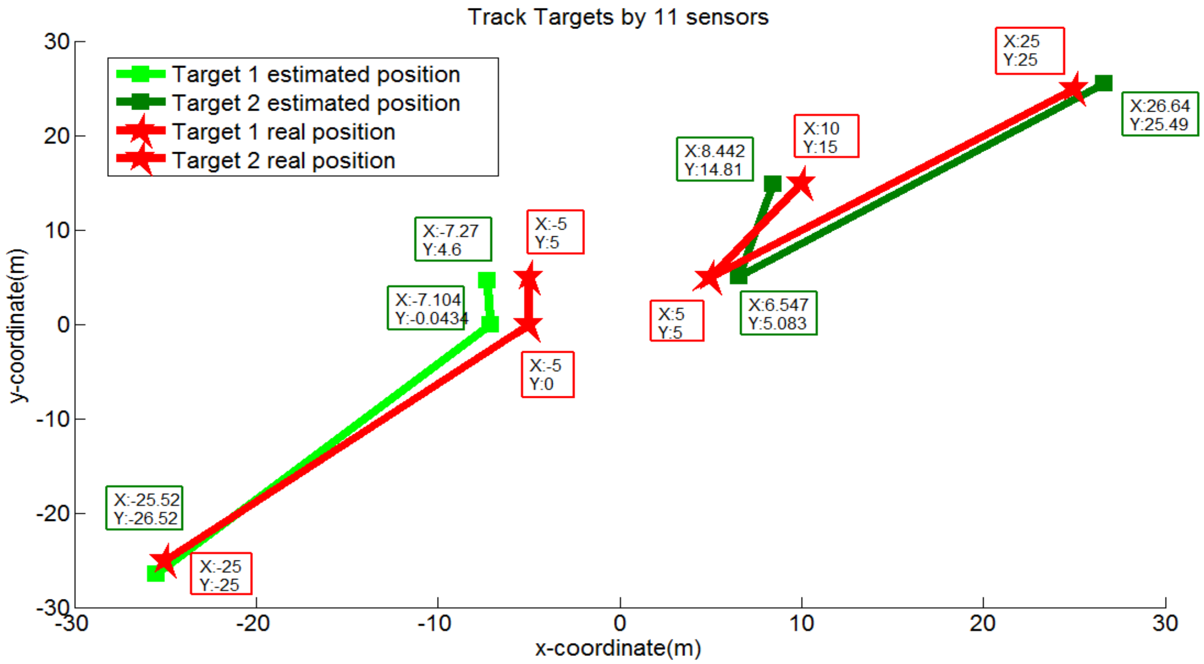


Figure 5. 46 Track two targets by 11 sensors.

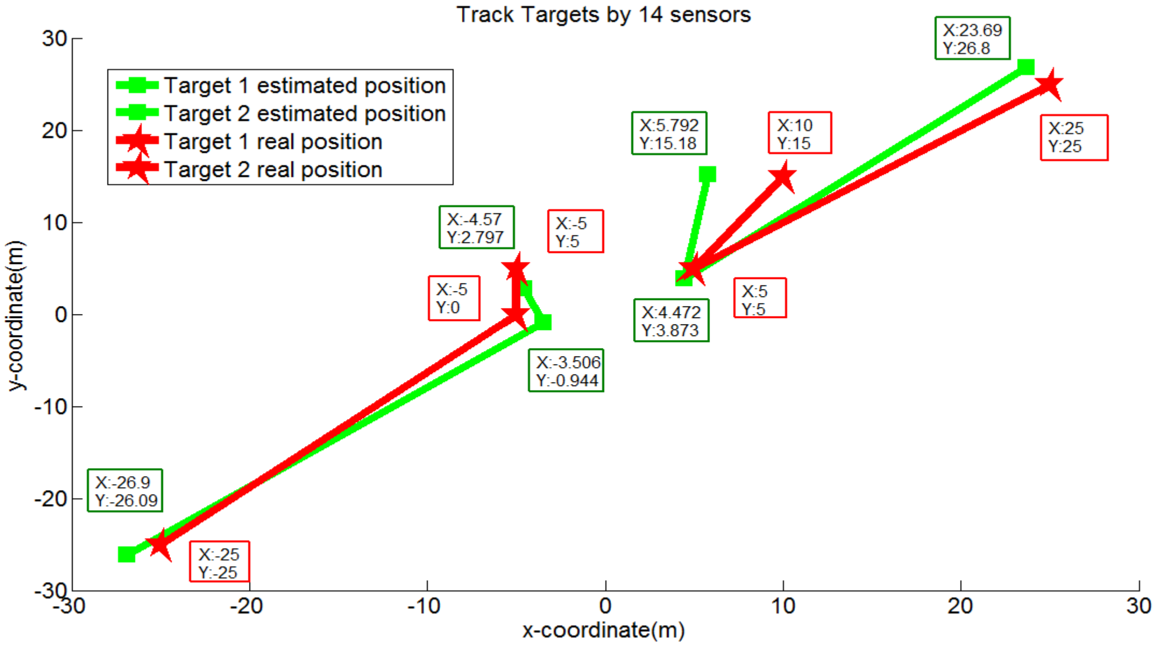


Figure 5. 47 Track two targets by 14 sensors.

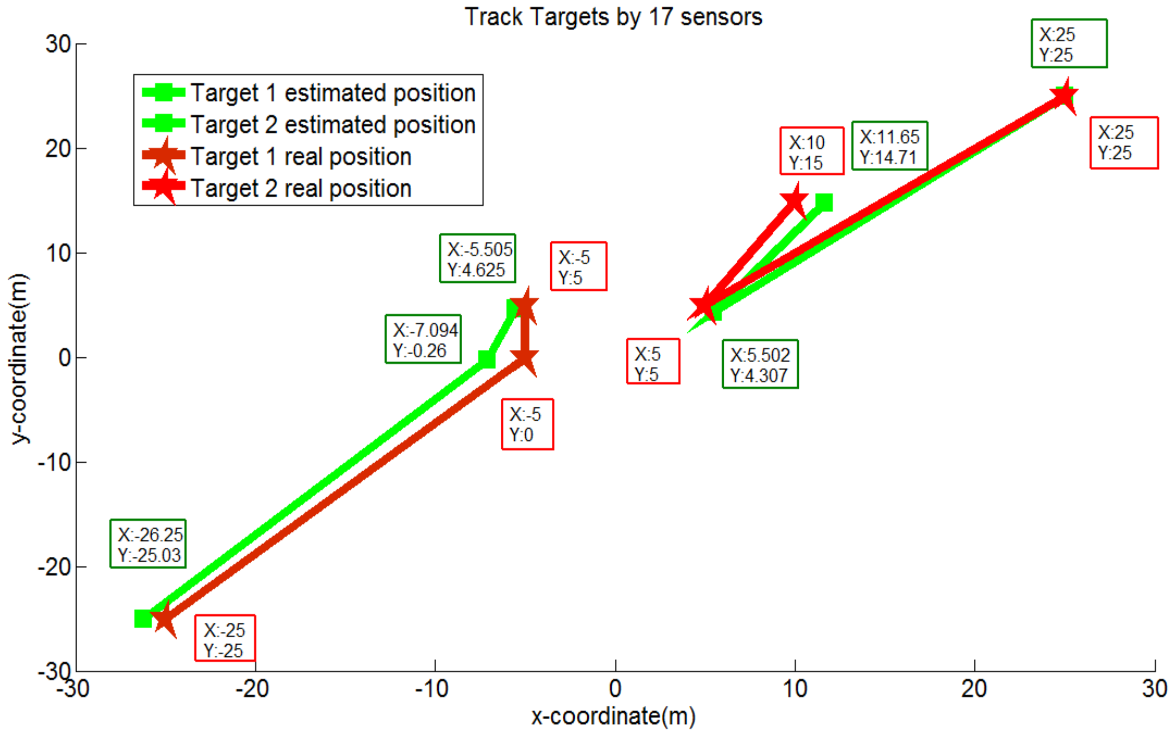


Figure 5. 48 **Track two targets by 17 sensors.**

Moreover, Increasing the number of sensors will increase the efficiency of the estimator as it has been found in the previous figures. Hence, it is better to look for another process that can help us to measure the efficiency of the estimator and to find the optimal number of the sensors that can be deployed on the ground in order to find the locations for these targets. Therefore, the mean square error (**MSE**) of the Cartesian coordinates of each target location for different number of sensors has been calculated and compared to the **CRLB**. The mean square error in **x, y,** and **z** coordinates of target 1 has been plotted vs. the square root of total number of sensors in Fig.5.49, Fig.5.50, and Fig.5.51, respectively. On the other hand, the **MSE** in **x, y,** and **z** coordinates of target 2 has been plotted vs. the square root of total number of sensors in Fig.5.52, Fig.5.53, and Fig.5.54, respectively. From these figures, one can observe that the performance of the likelihood estimator approaches the **CRLB** for large **WSNs**. The locations of the 2 targets are assumed to be known **T1** (40, 40, 4) and **T2** (-40, -40, 3). To maximize the maximum likelihood function, particle swarm optimization (**PSO**) method is used. The obtained estimated locations of target 1 and 2 are, respectively; **(36.2553, 38.4682, 3.3205)** and **(-37.0593, -37.9490, 1.5082)**. Since the estimated distance between the two targets is a matter of interest in some applications, thus, after estimating the locations of the 2 targets, the estimated distance can be easily found. For the last example the real distance is **113.1415** meters while the estimated distance is **105.9146** meters. Moreover, the Maximum Likelihood (**ML**) is a method used to estimate the location of target. As it is known that any estimating method like **ML** has certain precision, so there is some error in determining the locations. This error can be characterized by **MSE**. The **MSE** for **ML** and **CRBL** have been compared and it was found that they are approximately equal at **30** square root sensors (**900** sensors) which indicate that adding more than this number of sensors to the environment will not increase the performance of the **ML** estimator.

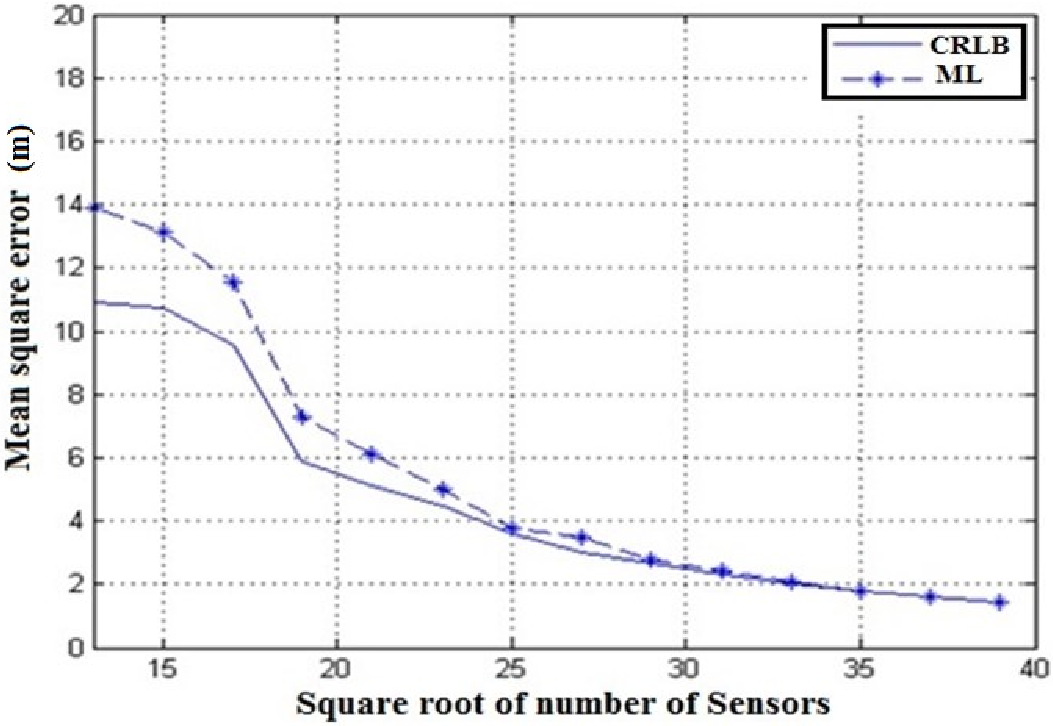


Figure 5. 49 Mean square error of x-Cartesian of target 1.

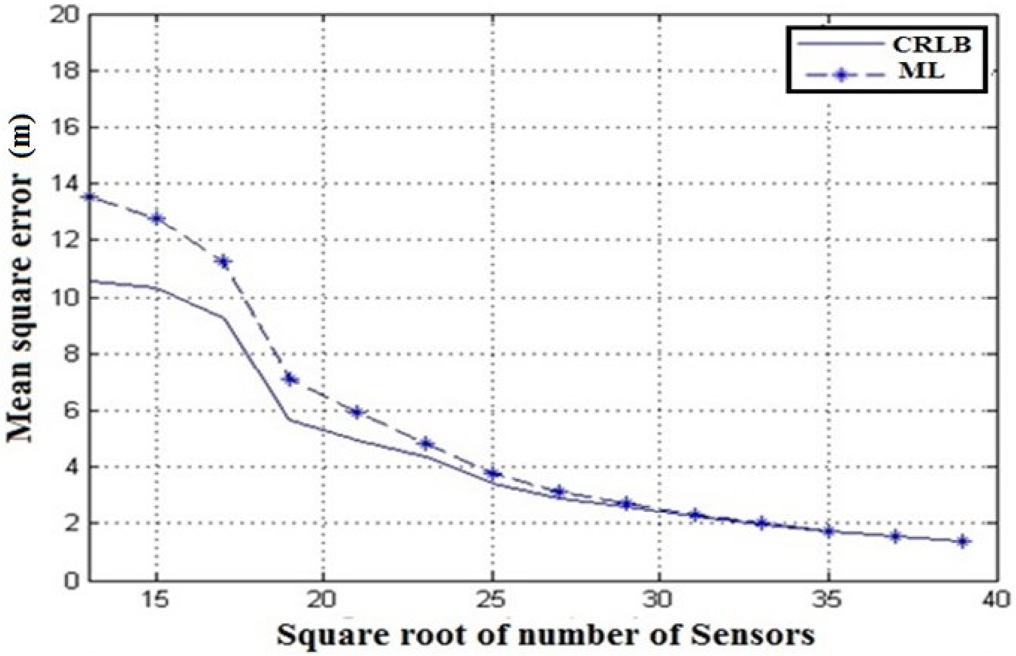


Figure 5. 50 Mean square error of y-Cartesian of target 1.

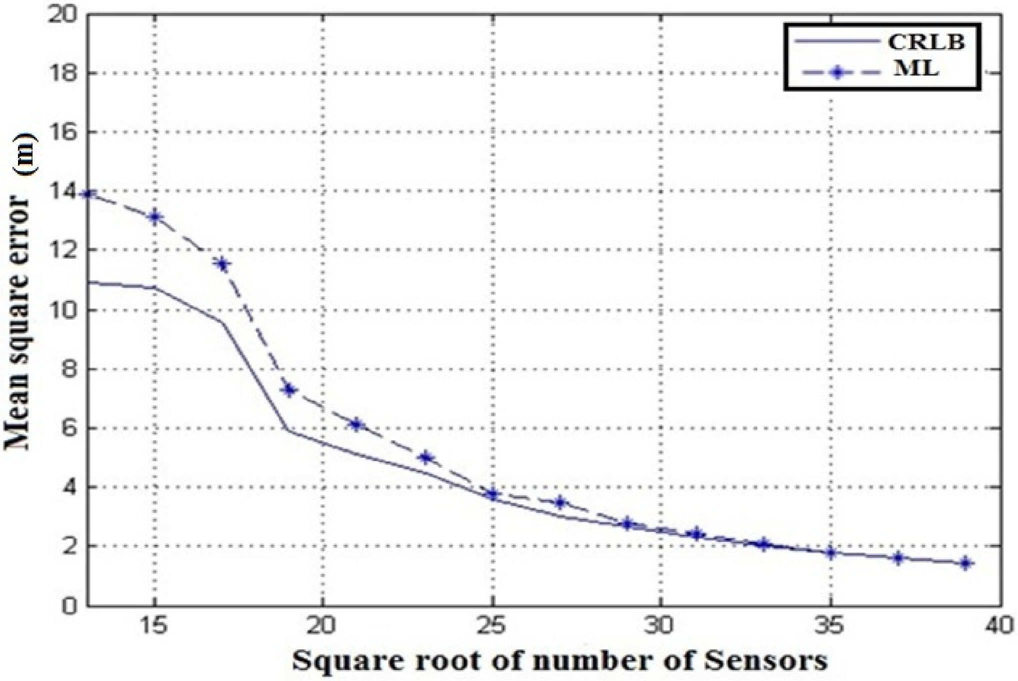


Figure 5. 51 Mean square error of z-Cartesian of target 1.

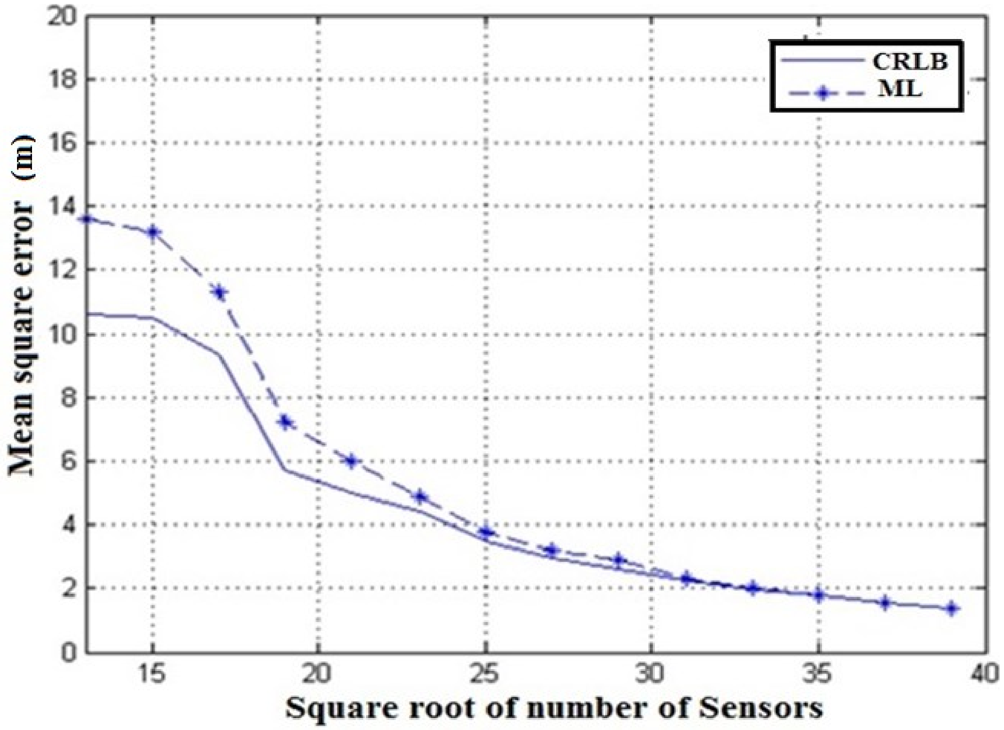


Figure 5. 52 Mean square error of x-Cartesian of target 2.

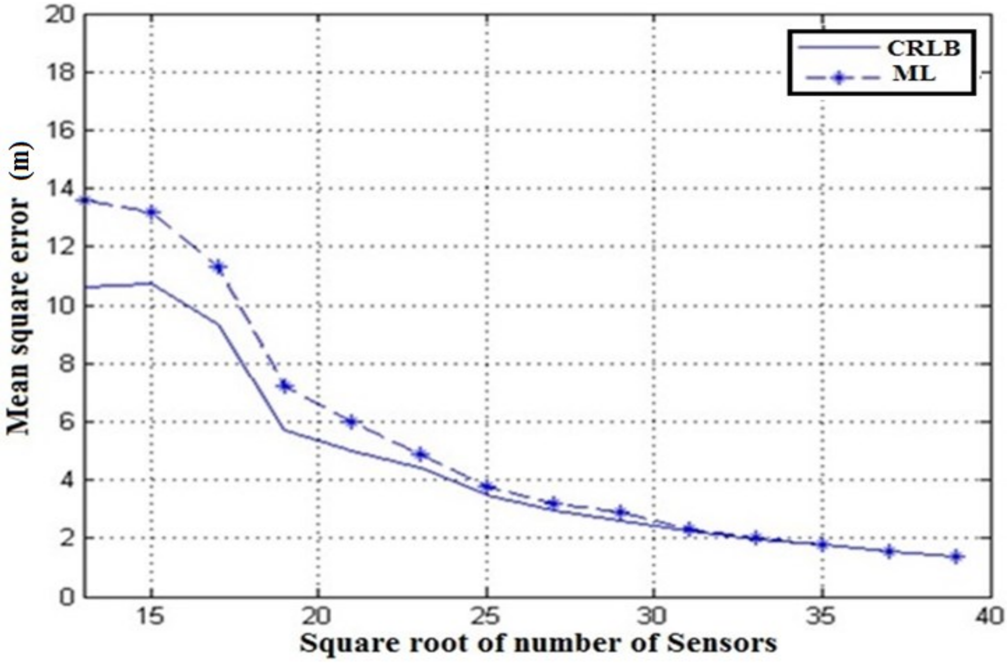


Figure 5. 53 Mean square error of y-Cartesian of target 2.

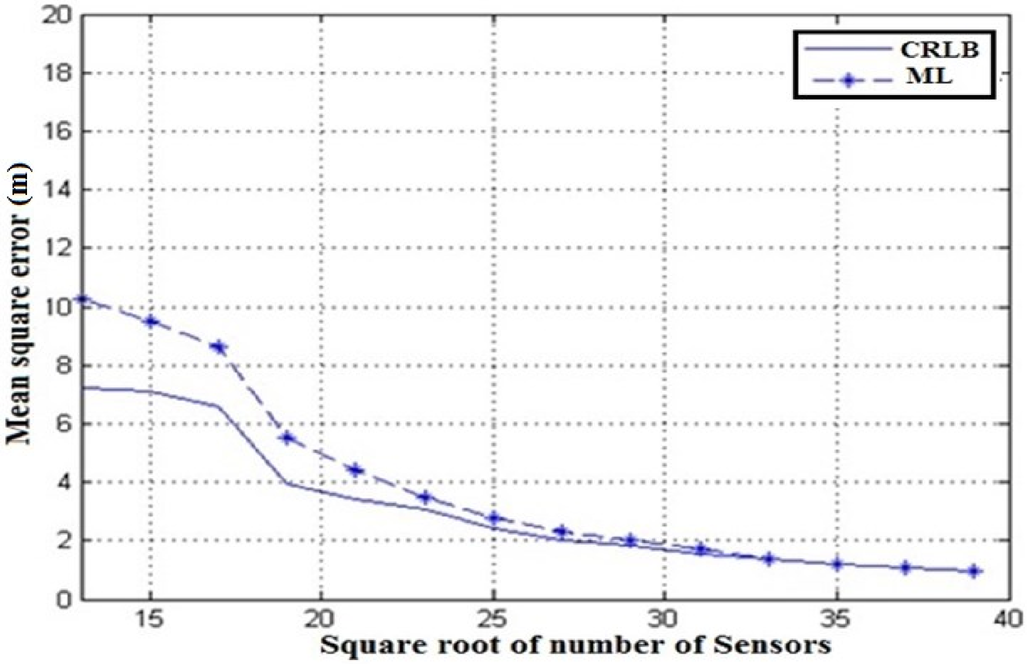


Figure 5. 54 **Mean square error of z-Cartesian of target 2.**

5.4 Conclusion

In this chapter, the design of fuzzy sets models in order to control the main behaviors of the blimp robot such as avoid obstacles, navigate in a certain altitude and also detect, localize and track targets are discussed. We have designed several fuzzy sets models in order to reduce the noise and drawbacks in the system's sensors. Then, multi- fuzzy controllers have been designed and implemented on the blimp robot to improve its ability to navigate in the environment and to finish the missions. Therefore, the combination between possibilities theory and fuzzy logic theory has been presented in order to design such models. They help us to solve the most important problem in the fuzzy methodology which is how to design the fuzzy knowledge base. Then, the bacterial algorithm was implemented on the fuzzy rules base in order to tune the membership functions and also to tune the rules. Many tests have been carried out to check the validity of the proposed controller and they show that the controller is robust and successful in all missions. On the other hand, the behavior of the **SURF** algorithm in order to identify, detect the target have been shown that it is robust and fast. Then, to validate the vision tracking as well as the intelligent control systems many experiments are done using blimp robot and ground robot in the indoor environment in our department Labs. The experiment results show the robustly, validity and applicability of the presented algorithms for the system. In addition, tracking 3D object aerial also success and show good results. Finally, in order to detect and localize multi-targets WSNs has been presented. The maximum likelihood estimator has shown good results and performance in order to detect the targets in the environment.

Chapter 6 Discussion and Future Work

In this last chapter, we present a summary of the accomplished work, highlighting its more relevant contributions of this dissertation. Then, we finish the chapter presenting a discussion towards the future directions and possibilities for this dissertation project. In this thesis, we decomposed the problem into four sub-problems: blimp localization, target tracking, cooperation, and finally the localization and tracking of the whole system based on WSNs. We addressed several key challenges for single target-based moving object tracking. First, how to separate the target (which can be aerial or ground target) from the motions of external objects, and how to detect and track complex moving objects. For this purpose, the computer vision methods algorithms for tracking is described, and the design of a fuzzy sets model and possibilities distribution filters to handle the noise and uncertainty of sensor inputs is presented. The uncertainty of target position estimation has been reduced using fuzzy sets model that are able to correct the prediction position information in the computer vision system. Then, the tracking strategy which is based on WSNs is presented to provide the information about the whole system and it is sufficient for such purposes. Unlike other works and researches, in this work we addressed the problem of estimating locations of different targets in 3D environment using maximum likelihood estimator.

In chapter one, the motivation and research work scope regarding to the problem and its solutions are discussed. In chapter two, the literature reviews to identify what has been done before was presented. Then, it discussed the vision technology regarding to the target detection and tracking. In addition, the WSNs technology and its advantages for localization and tracking had been presented. Moreover, development of a cooperation between blimp and ground robots based on fuzzy logic and image processing is formulized in chapter three. Furthermore, intelligent control as well as navigation algorithms are designed to take the best to accomplish the tasks and added more challenges to the blimp. Therefore, the blimp system coped with these challenges and uncertainties in its location by using artificial system which could deal with noises and drawbacks. Then, we approached the challenge of setting up the autonomous blimp system. Thereto, we presented solutions for multi-problems: designed a blimp system including appropriate hardware devices based on embedded system, introduced artificial intelligent methods for navigation and control, and presented artificial algorithms to deal with uncertainties, drawbacks and correct the sensors information. We would like to mention that we solved the most important problem in fuzzy control which is how to design the membership functions and the fuzzy base rules. We combined possibilities distribution theory, fuzzy sets and bacterial algorithm, then, designed the membership functions and the generalized modus ponens (Fuzzy rules base) as well as tuning the rules and delete ineffective rule.

Moreover, we presented a solution for single robot-based tracking problem using flying robot (Blimp Robot). For this purpose, we presented the tracking, localization and object detection method using Speeded Up Robust Features SURF algorithm, then, the fuzzy sets models was implemented to handle the noise and uncertainty of the vision sensor. The Inverse Perspective Mapping IPM is also implemented in the blimp vision system in order to obtain the top down bird's eye view. This helped to find the linearity relation between the metric distances in real world and pixels distances in the image plane. Then, this tracking method is used to estimate a projection model in the camera image space. The proposed methods have been implemented and tested in various indoor environments at **RST Labs**.

6.1 Future Works

Having stated what has been accomplished in this thesis, it is time to refer the future steps for this work. Perhaps the best starting point is to refer the possibilities for scalability and extendibility. About scalability, it was interesting to design blimp robot, then, design the navigation algorithms and fuzzy controllers systems. Furthermore, develop, implement and test the vision tracking and localization algorithms by using a team architecture using real robots (blimp and ground robot). The combination between possibilities distribution and fuzzy sets makes the system more robust and reliable.

However, about extendibility, despite of many researches in the tracking, navigations and localization, the problem is not fully solved yet. The main problems and issues are solved for indoor environment. For outdoor environment is still poses many challenges, furthermore the problem of illumination is still required more research works.

There is also a need for a high performance and storage memory (SDRAM and NAND flash memory). This could be done by mounting another core device which could be used for large maps in case of study the simultaneous localization and mapping SLAM. Another camera with higher field of view and resolution can also be used for further investigation. Therefore, the **SURF** algorithm could be used in **SLAM.** Then, the targets can be localized based on matching between images which are obtained from aerial and from the target. As a result, we can reconstruct the 3D map for the surrounded environment.

Another important step towards the future is implementing another approach for **WSNs**. For localization multi targets based on **WSNs**, rather than using **OOK** to modulate the decisions, it might use another approach based on probability of the detecting sensors between [0, 1]. It could be used fuzzy logic to give each sensor's decision a degree of membership functions. This could increase the computational time, but it deserves a try as the estimations might be increased. The clustering and degrees of membership functions obtained from fuzzy might be used to find if the target are closing or going far from the fusion center.

After stated the discussion and the future work, the last thing worth to include here is a quote from Lotfi Zadeh, the “Creator of Fuzzy Logic”.

" I never dreamed that Fuzzy Logic would become a worldwide phenomenon. My expectations were much more modest."

- Interview with Lotfi Zadeh, Creator of Fuzzy Logic, 1994

Appendix A

OpenEmbedded and Gumstix Boards

The products in the Gumstix Overo are divided into two parts. First, the computer-on-module (Gumstix Overo Air COM) which is a tiny and small size similar to the small battery. Second, Summit Overo Board which provides the connectors needed for I/O functions. Despite the small size, the combination between them will perform like a full Linux computer and can be re-programmed to perform and to obtain several and variety functions in any applications area such as access control, location tracking, and robotics. There is a 27-pins cable connector mounted on the top side of Gumstix to mount Caspa camera board [211, 212].

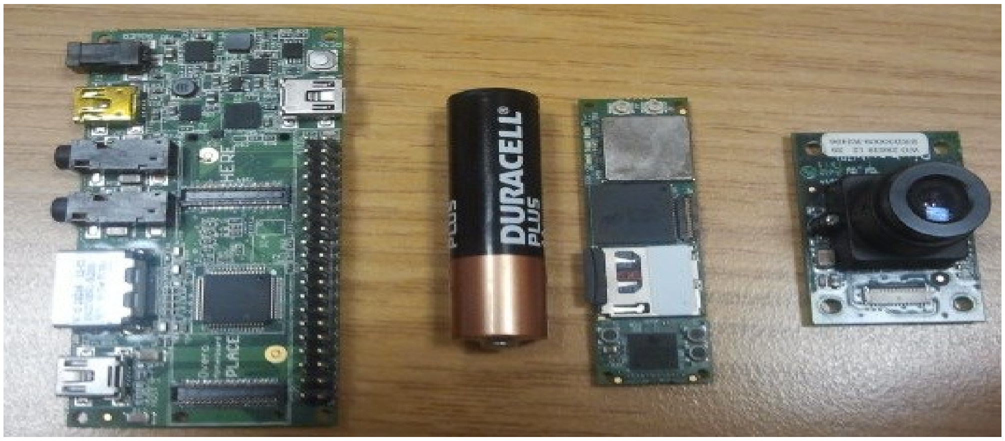


Figure A-A1 **Summit board, Gumstix Overo, Caspa Camera (left-right).**

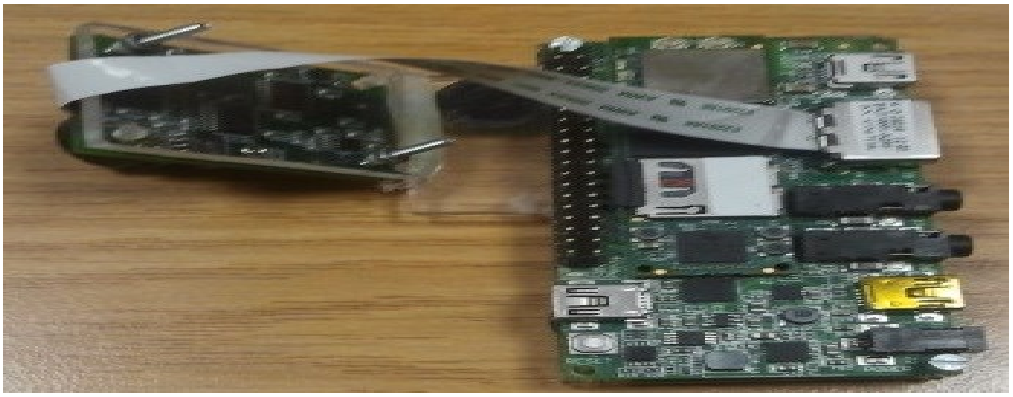
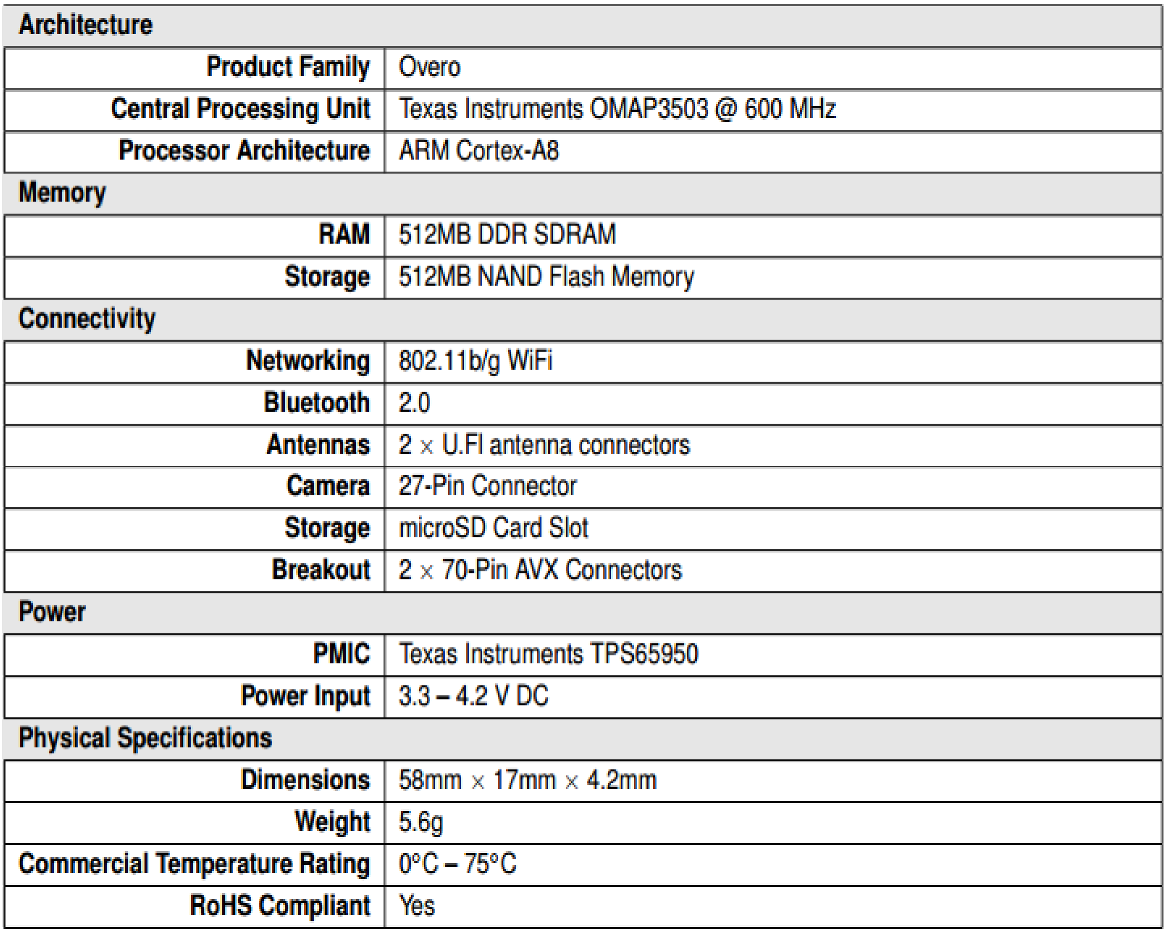


Figure A-A2 Caspa camera mounted to Gumstix Overo Air.

The main features of an Overo Air includes the following [211, 212]:



Architecture	
Product Family	Overo
Central Processing Unit	Texas Instruments OMAP3503 @ 600 MHz
Processor Architecture	ARM Cortex-A8
Memory	
RAM	512MB DDR SDRAM
Storage	512MB NAND Flash Memory
Connectivity	
Networking	802.11b/g WiFi
Bluetooth	2.0
Antennas	2 × U.FI antenna connectors
Camera	27-Pin Connector
Storage	microSD Card Slot
Breakout	2 × 70-Pin AVX Connectors
Power	
PMIC	Texas Instruments TPS65950
Power Input	3.3 – 4.2 V DC
Physical Specifications	
Dimensions	58mm × 17mm × 4.2mm
Weight	5.6g
Commercial Temperature Rating	0°C – 75°C
RoHS Compliant	Yes

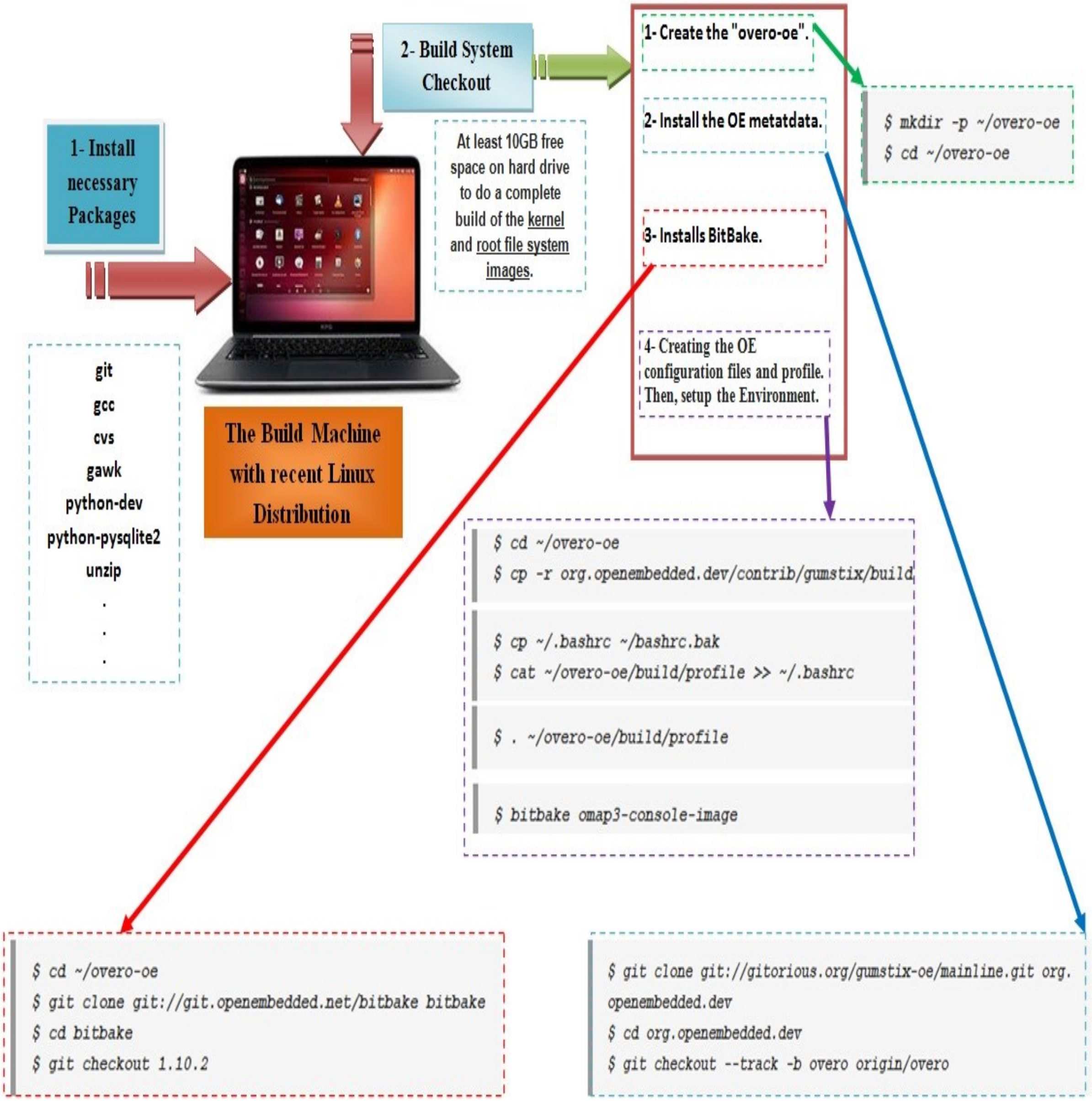
Figure A-A3 The specifications of the Gumstix Overo Air.

Test the reliability using US MIL-STD-810F Testing

The **MIL-STD-810F** tests series have been done on Overo which are approved that it could be used by all departments and agencies of the United States Department of Defense (DoD). These tests results proved the robustness of the dual { 70-pins and the 27-pins connectors} of the Overo COM mechanical design [212]. The Gumstix tests performed by simulated vibration for the equivalent of 1,642,500 kilometers. This test is simulated on a “Diesel Engine Road Vehicle” that moved on an assortment of road surfaces [212].

OpenEmbedded Development Environment

It is better to give the brief description about **OpenEmbedded** and **BitBake** [213, 214]. OpenEmbedded is the build framework for embedded Linux and it offers a cross-compile environment which can allow designers to create a complete Linux distributions for embedded systems. Indeed, the Gumstix Company had implemented the OpenEmbedded build environment for developers to program in Linux or to develop their embedded applications. Without any charge or license fee, we can use the OpenEmbedded packages management tools to prebuilt hundreds binaries of open-source software packages. However, in order to modify the kernel, we need to access to an OpenEmbedded build environment. Therefore, to start our own environment we need to follow the steps and the instructions in the BitBake tutorial using these OpenEmbedded instructions. However, the Gumstix produces pre-built images for several Linux distributions systems for users to run on their systems. These can be downloaded for the Gumstix hardware includes **Yocto** project, **Angstrom**, **Ubuntu** and **Android** [211, 212]. Fig. A-A4 shows the process to build the recent Linux distribution for Gumstix Overo Air. It starts by installing the necessary packages on a host **PC** computer which has the most recent Linux distributions. Then, following the tutorials from the OpenEmbedded and BitBake to create the root files system images [213, 214].


1- Install necessary Packages
git
gcc
cvs
gawk
python-dev
python-pysqlite2
unzip
.
.
.
The Build Machine with recent Linux Distribution
2- Build System Checkout
At least 10GB free space on hard drive to do a complete build of the kernel and root file system images.
1- Create the "overo-oe".
2- Install the OE metatdata.
3- Installs BitBake.
4- Creating the OE configuration files and profile. Then, setup the Environment.

```
$ mkdir -p ~/overo-oe
$ cd ~/overo-oe
```

```
$ cd ~/overo-oe
$ cp -r org.openembedded.dev/contrib/gumstix/build

$ cp ~/.bashrc ~/bashrc.bak
$ cat ~/overo-oe/build/profile >> ~/.bashrc

$ . ~/overo-oe/build/profile

$ bitbake omap3-console-image
```

```
$ cd ~/overo-oe
$ git clone git://git.openembedded.net/bitbake bitbake
$ cd bitbake
$ git checkout 1.10.2
```

```
$ git clone git://gitorious.org/gumstix-oe/mainline.git org.
openembedded.dev
$ cd org.openembedded.dev
$ git checkout --track -b overo origin/overo
```

Figure A-A4 The process to build recent Linux to Gumstix Overo Air.

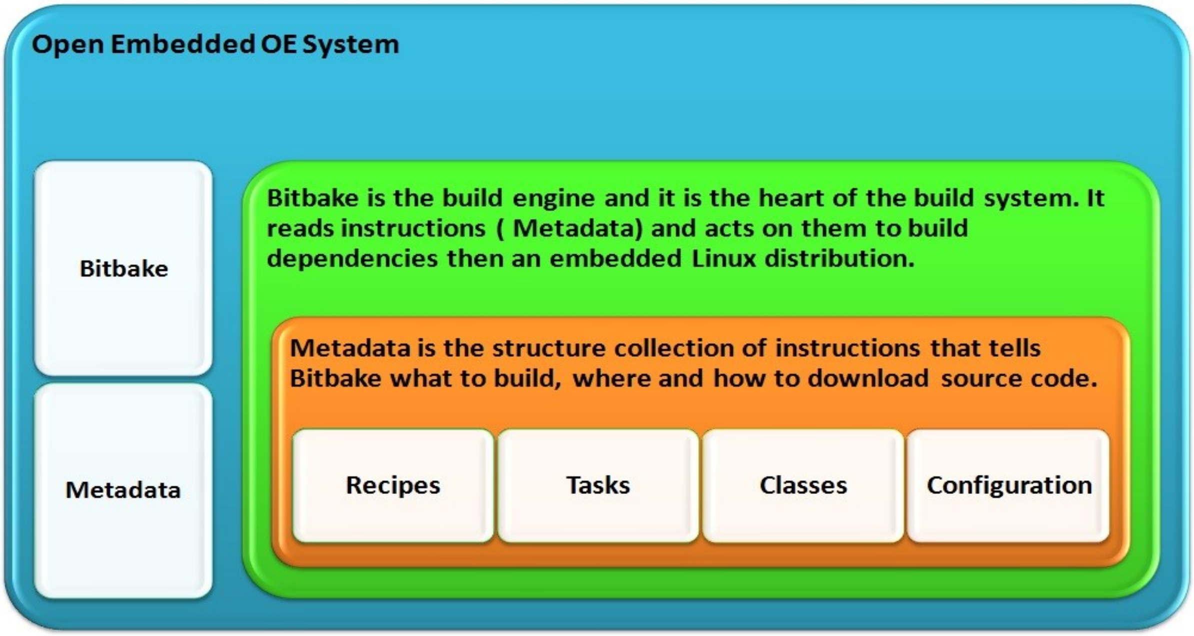

Open Embedded OE System
Bitbake
Metadata
Bitbake is the build engine and it is the heart of the build system. It reads instructions (Metadata) and acts on them to build dependencies then an embedded Linux distribution.
Metadata is the structure collection of instructions that tells Bitbake what to build, where and how to download source code.
Recipes
Tasks
Classes
Configuration

Figure A-A5 The open Embedded System.

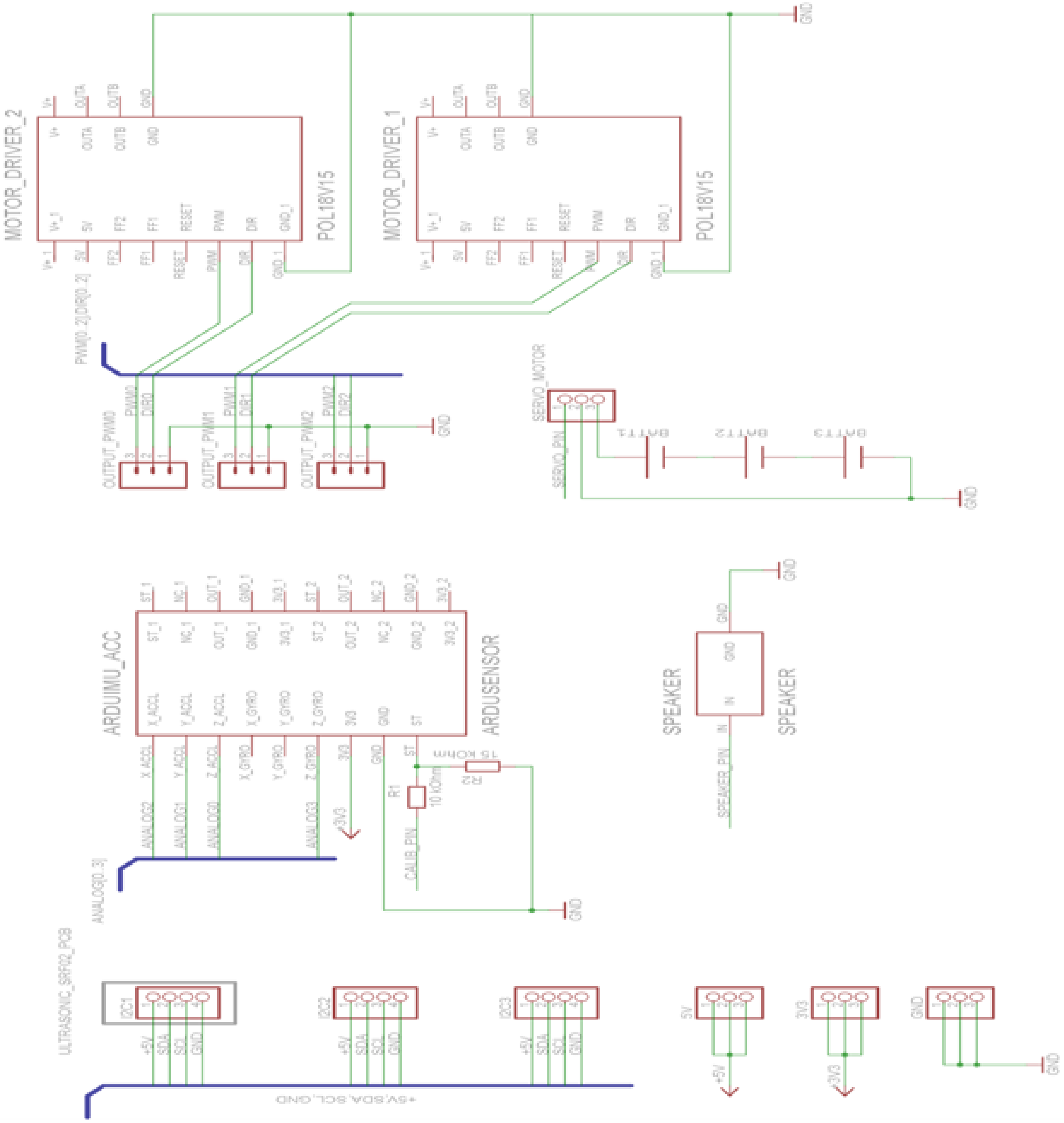


Figure A-A6 The Sketch diagram for Hardware.

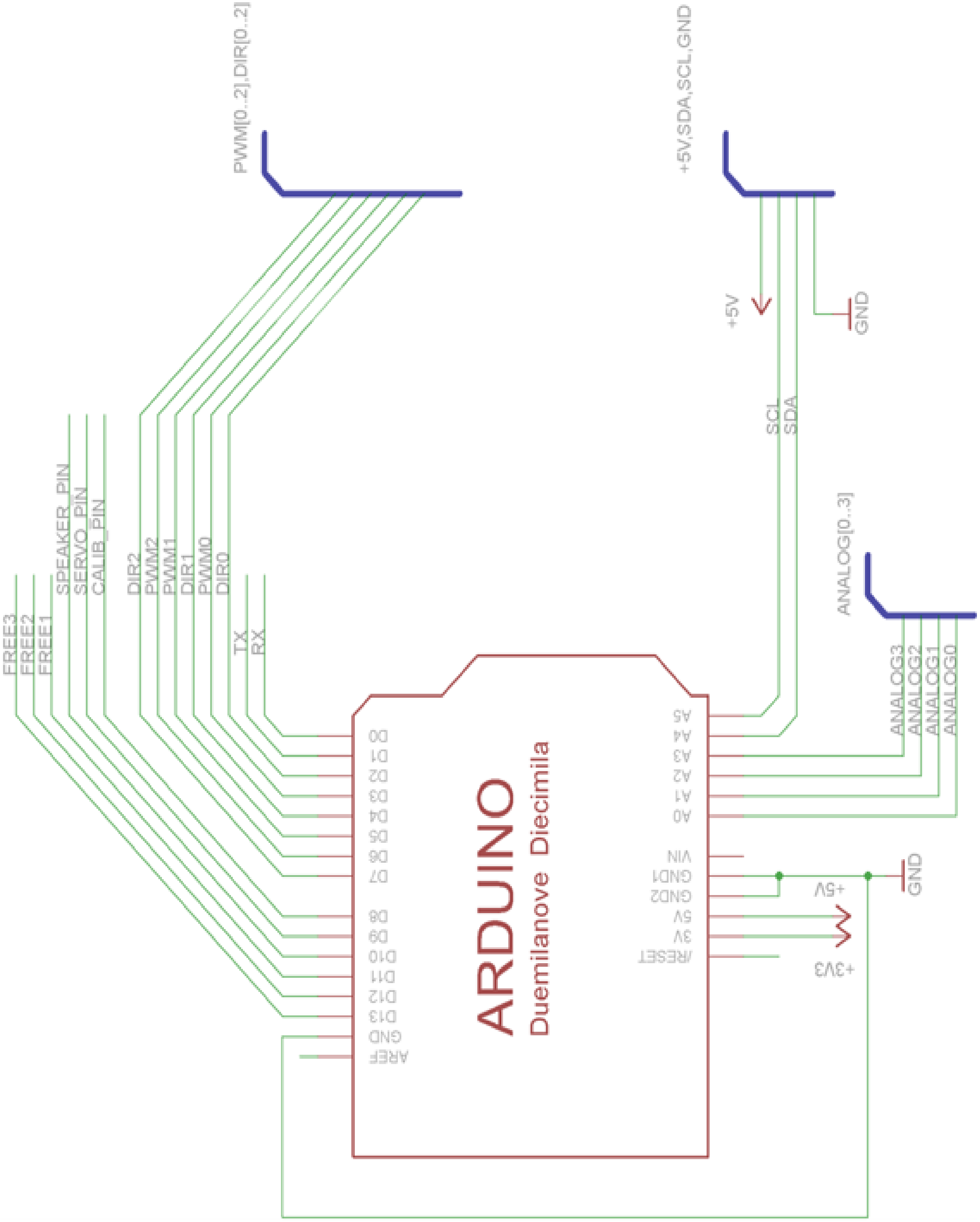


Figure A-A7 The Arduino pins.

Appendix B

A. Shneiderman Eight Golden Rules of Interface Design [215].

These are the eight rules which were obtained from the text designing the Graphical User Interface by Ben Shneiderman. These principles are derived from experience and applicable in most interactive systems.

1. Strive for consistency

The consistent sequences of action are required in similar situations; identical terminology is used in menus, and help screens; and consistent commands could be employed throughout.

2. Enable frequent users to use shortcuts.

Because the frequency of use increases, we might need to obtain some desires and objects for users such decreasing interactions number and increasing the interaction pace. Therefore, the function keys and the hidden commands are very helpful to any expert user.

3. Offer informative feedback.

In frequent and minor actions, the response should be modest; whilst for infrequent and major actions, the response is more substantial.

4. Design dialog to yield closure.

The actions sequences should be organized into groups for some essential purposes such as giving the operators the satisfaction of accomplishment, a sense of relief, and a clear indication to prepare for the next group of actions.

5. Offer simple error handling.

Because the user shall have serious errors, the designer of the system should take into account these errors so the system could have ability to be able to detect the errors and mistakes, and then offer method to handle these errors.

6. Permit easy reversal of actions.

This step is done to encourage the exploration of unfamiliar options. The units of reversibility may be a single action or a complete group of actions.

7. Support internal locus of control.

Because the operators always prefer the desire to be in charge with the system they work with and then that the system could easily respond to actions. Thus, we design a system that makes operators the initiators of actions rather than the responders.

8. Reduce short-term memory load.

displays must be kept simple with multiple page displays.

B. Jacob Nielsen 10 Usability Heuristics [216]

These are ten general principles for user interface design.

1. Visibility of system status

The system should always keep operator informed about what is going on.

2. Match between system and the real world

The system should show the information in familiar phrases to the user and make information appear in logical order.

3. User control and freedom

Since the operator might choose some functions by mistake, the designer should take into account an emergency exit during the design.

4. Consistency and standards

Operators might be not have to wonder if there is different words, or actions. Thus, follow platform conventions will help to do this.

5. Error prevention

Check the error then present then to the users.

6. Recognition rather than recall

The operator should not have to remember all information from one part to another. So instructions of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency

The system should be easy and available to use for both inexperienced and experienced operators.

8. Aesthetic and minimalist design

Dialogues don’t contain information that is irrelevant or rarely needed.

9. Help users recognize, diagnose, and recover from errors

Error messages must be expressed in plain language to indicate the problem, and suggest a solution.

10. Help and documentation

It is very necessary to provide a help and documentation that is easy to search and not be too large.

C. The Gestalt Laws of Grouping [217]

The Gestalt laws includes several laws such as proximity, similarity, good continuation, closure, , common fate, symmetry, figure-ground, and periodicity.

Appendix C

$$\boldsymbol{J} = \boldsymbol{E}\left\{\left[-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\nabla}_{\boldsymbol{\theta}}^{T}\, ln\left(p(\boldsymbol{U}|\boldsymbol{\theta})\right)\right]\right\} \text{[168, 169]}$$

$$= -\boldsymbol{E}\begin{bmatrix}
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1^2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial y_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial z_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial x_2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial y_2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial z_2} & \cdots & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial x_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial y_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_1\partial z_N} \\
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1\partial x_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1^2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1\partial z_1} & & \cdots & & & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1\partial x_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1\partial y_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_1\partial z_N} \\
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1\partial x_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1\partial y_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1^2} & & \cdots & & & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1\partial x_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1\partial y_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_1\partial z_N} \\
& \vdots & & & \ddots & & & & \vdots & \\
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x\partial x_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_N\partial y_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_N\partial z_1} & & \cdots & & & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_N^2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_N\partial y_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_N\partial z_N} \\
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_N\partial x_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial yz_N\partial y_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_N\partial z_1} & & \cdots & & & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_N\partial x_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_N^2} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_N\partial z_N} \\
\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N\partial x_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N\partial y_1} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N\partial z_1} & & \cdots & & & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N\partial x_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N\partial y_N} & \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_N^2}
\end{bmatrix}$$

$$\boldsymbol{E}\left[\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2}\right] = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\sum_{u_k=\{0,1\}}\frac{\boldsymbol{-1}}{P(u_k|S_k)}\left[\frac{\partial P(u_k|S_k)}{\partial x_i}\right]^{\boldsymbol{2}} + \frac{\partial^2 P(u_k|S_k)}{\partial x_i^2}$$

$$\sum_{k=1}^{K}\sum_{u_k=\{0,1\}}\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2} = \sum_{k=1}^{K}\frac{\partial^2 P(u_k=0|S_k)}{\partial x_i^2} + \frac{\partial^2 P(u_k=1|S_k)}{\partial x_i^2}$$

$$= \sum_{k=1}^{K}\frac{\partial^2}{\partial x_i^2}\left(P(u_k=0|S_k) + P(u_k=1|S_k)\right)$$

$$= \sum_{k=1}^{K}\frac{\partial^2}{\partial x_i^2}(1)$$

Then the element (1, 1) is

$$\boldsymbol{E}\left[\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2}\right] = \sum_{k=1}^{K}\frac{-1}{P(u_k=0|S_k)}\left[\frac{\partial P(u_k=0|S_k)}{\partial x_i}\right]^2 + \frac{-1}{P(u_k=1|S_k)}\left[\frac{\partial P(u_k=1|S_k)}{\partial x_i}\right]^2$$

Chain rule

$$\frac{\partial P(u_k=1|S_k)}{\partial x_i} = \frac{\partial P(u_k=1|S_k)}{\partial A_k(\boldsymbol{\theta})}\times\frac{\partial A_k(\boldsymbol{\theta})}{\partial x_i}$$

$$\frac{\partial A_k(\boldsymbol{\theta})}{\partial x_i} = \frac{\sqrt{P_0 d_0^n}\, n}{2}\left(d_{k,i}\right)^{-\left(\frac{n}{2}+2\right)}\left(x_k - x_{,i}\right)$$

$$\frac{\partial P(u_k=1|S_k)}{\partial x_i} = \frac{\sqrt{P_0 d_0^n}\, n}{2\,\sigma_k\sqrt{2\pi}}\left(d_{k,i}\right)^{-\left(\frac{n}{2}+2\right)}(x_k - x_i)\times e^{-\frac{\left(\tau_k - A_k(\boldsymbol{\theta})\right)^2}{2\sigma_k^2}}$$

$$\left[\frac{\partial P(u_k=1|S_k)}{\partial x_i}\right]^2 = \frac{P_0 d_0^n\, n^2}{8\pi\sigma_k^2}\left(d_{k,i}\right)^{-(n+4)}(x_k - x_i)^2\times e^{\frac{-\left(\tau_k - A_k(\boldsymbol{\theta})\right)^2}{\sigma_k^2}}$$

$$= \mathfrak{B}_{k,i}\mathcal{A}_k\left(x_k - x_{t,i}\right)^2$$

$$\left[\frac{\partial P(u_k = 0|S_k)}{\partial x_i}\right]^2 = -\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2$$

where $\mathfrak{B}_{k,i} = \frac{P_0 d_0^n\, n^2}{8\pi\sigma_k^2}\left(d_{k,i}\right)^{-(n+4)}$, $\mathcal{A}_k = e^{\frac{-\left(\tau_k - A_k(\boldsymbol{\theta})\right)^2}{\sigma_k^2}}$.

$$\boldsymbol{E}\left[\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial x_i^2}\right] = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\left\{\left(\frac{-\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)}\right) - \left(\frac{\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)\right\}$$

$$= -\sum_{k=1}^{K}\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)^2\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

By symmetry:

$$-\boldsymbol{E}\left[\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial y_i^2}\right] = \sum_{k=1}^{K}\mathfrak{B}_{k,i}\mathcal{A}_k(y_k - y_i)^2\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

$$-\boldsymbol{E}\left[\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial z_i^2}\right] = \sum_{k=1}^{K}\mathfrak{B}_{k,i}\mathcal{A}_k(z_k - z_i)^2\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

Elements (1, 2) and (2, 1) are the same.

$$\boldsymbol{E}\left[\frac{\partial^2 ln\left(p(\boldsymbol{U}|\boldsymbol{\theta})\right)}{\partial x_i \partial y_i}\right] = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\sum_{u_k=\{0,1\}}\frac{\boldsymbol{-1}}{P(u_k|S_k)}\left(\frac{\partial P(u_k|S_k)}{\partial x_i}\frac{\partial P(u_k|S_k)}{\partial y_i}\right) + \frac{\partial^2 P(u_k|S_k)}{\partial x_i \partial y_i}$$

$$\sum_{u_k=\{0,1\}}\frac{\partial^2 P(u_k|S_k)}{\partial x_i \partial y_i} = \frac{\partial^2}{\partial x_i \partial y_i}\left(\sum_{u_k=\{0,1\}} P(u_k|S_k)\right) = \frac{\partial^2}{\partial x_i \partial y_i}(1) = 0$$

⇨

$$\boldsymbol{E}\left[\frac{\partial^2 ln\left(p(\boldsymbol{U}|\boldsymbol{\theta})\right)}{\partial x_i \partial y_i}\right] = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\sum_{u_k=\{0,1\}}\frac{\boldsymbol{-1}}{P(u_k|S_k)}\left(\frac{\partial P(u_k|S_k)}{\partial x_i}\frac{\partial P(u_k|S_k)}{\partial y_i}\right)$$

$$= \sum_{k=1}^{K}\frac{-1}{P(u_k = 0|S_k)}\left(\frac{\partial P(u_k = 0|S_k)}{\partial x_i}\frac{\partial P(u_k = 0|S_k)}{\partial y_i}\right) + \frac{-1}{P(u_k = 1|S_k)}\left(\frac{\partial P(u_k = 1|S_k)}{\partial x_i}\frac{\partial P(u_k = 1|S_k)}{\partial y_i}\right)$$

$$= -\sum_{k=1}^{K}\mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)(y_k - y_i)\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

By symmetry

$$E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial x_i \partial z_i}\right] = -\sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)(z_k - z_i)\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

$$E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial y_i \partial z_i}\right] = -\sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(y_k - y_i)(z_k - z_i)\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

$$E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial x_i \partial z_i}\right] = -\sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(x_k - x_i)(z_k - z_i)\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} + \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

$$E\left[\frac{\partial^2 ln\big(p(\boldsymbol{U}|\boldsymbol{\theta})\big)}{\partial y_i \partial z_i}\right] = -\sum_{k=1}^{K} \mathfrak{B}_{k,i}\mathcal{A}_k(y_k - y_i)(z_k - z_i)\left(\frac{1}{Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)} - \frac{1}{\left[1 - Q\left(\frac{\tau_k - A_k(\boldsymbol{\theta})}{\sigma_k}\right)\right]}\right)$$

Other elements can be easily determined from these elements by exploiting the symmetry of the matrix.

Appendix D

D1: Servoing Fuzzy Control

A1- Follow Target Fuzzy Control :

This controller has two inputs and one output. The membership functions for inputs and output for the vectorization angle controller are shown in Fig. W-1, Fig. W-2 and Fig. W-3, respectively. The first input is the distance between the blimp and the target projection point. It has seven linguistic variables (Nor: Normal, VS: very small, S: small, VM: very medium, M: medium, B: Big, VB: very big). The second input is the difference between the last two distances with five linguistic variables (VS: very small, S: small, Equ: equal, B: big, VB: very big). The output is the vectorization angle command with seven linguistic variables (Z: zero, VS: very small, S: small, M: Medium, VM: very medium, B: big, VB: very big). The initial rules were 35 rules as they are shown in the Table W.1.

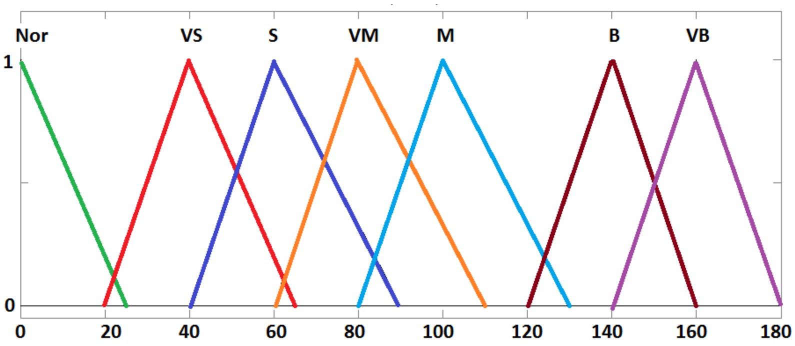


Fig. W-1. The membership functions for input1.

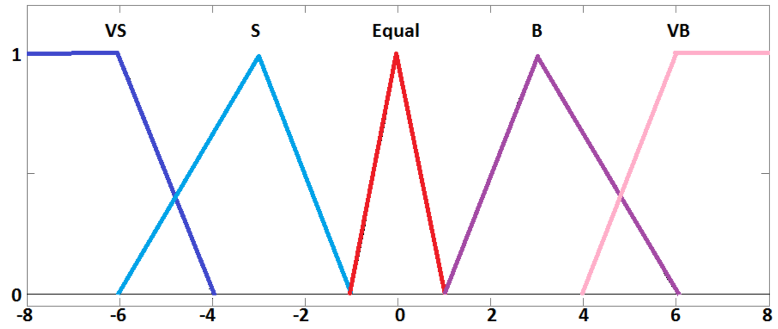


Fig. W-2. The membership functions for input2.

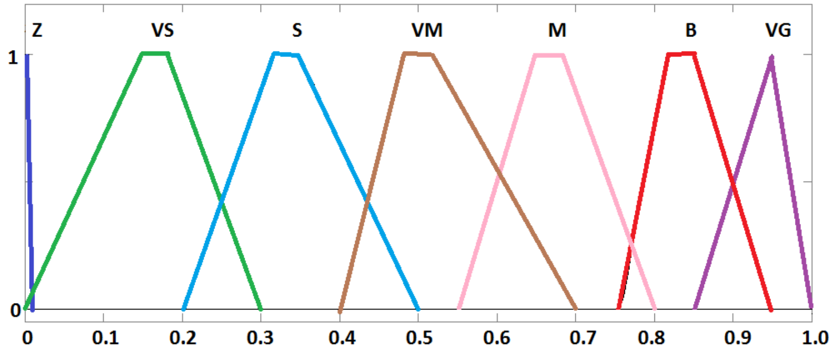


Fig. W-3. The membership functions for Output.

Table. W-1. The Initial Fuzzy Rule Base.

		In1						
		Nor	*VS*	*S*	*VM*	*M*	*B*	*VB*
In2	*VS*	Z	Z	Z	VS	VM	M	B
	S	Z	Z	VS	VS	VM	B	B
	Equal	Z	Z	VS	VS	VM	B	VB
	B	Z	VS	VS	S	VM	B	VB
	VB	Z	VS	S	S	VM	VB	VB

A2- Optimized Follow Target Fuzzy Control :

The optimized membership functions for inputs and output for the vectorization angle controller are shown in Fig. W-4, Fig. W-5 and Fig. W-6, respectively. The optimized rules were 25 rules as they are shown in the Table W-2.

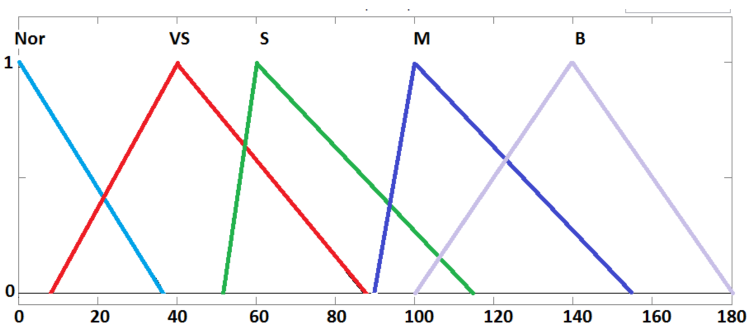


Fig. W-4. The membership functions for input1

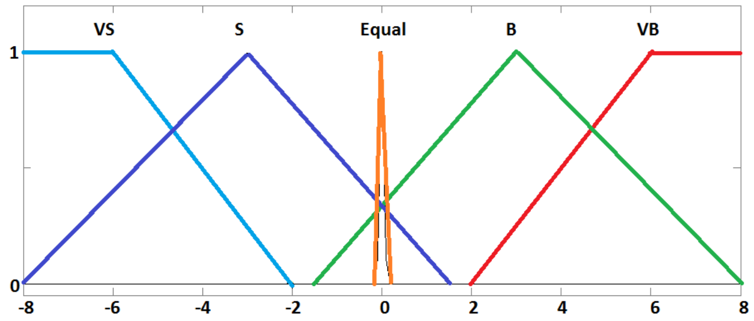


Fig. W-5. The membership functions for input2.

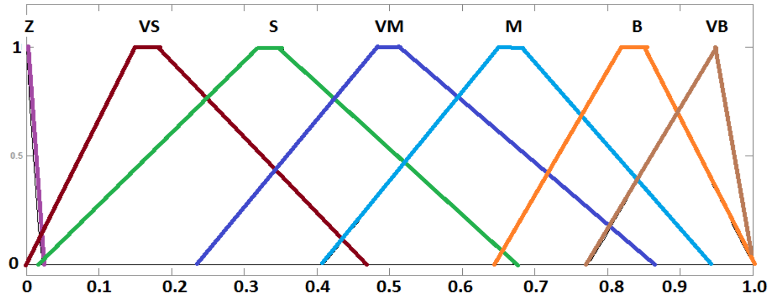


Fig. W-6. The membership functions for Output.

Table. W-2. The Optimized Fuzzy Rule Base.

		In1						
In2		*Nor*	*VS*	*S*	*VM*	*M*	*B*	*VB*
	VS	Z	Z	VS	X	VM	B	X
	S	Z	Z	VS	X	B	B	X
	Equal	Z	VS	VS	X	B	VB	X

	B	Z	VS	VS	X	B	VB	X
	VB	Z	VS	VM	X	B	VB	X

It was found that the best fitness function value for 1000 generations = 0.5206. The fitness function is the absolute errors.

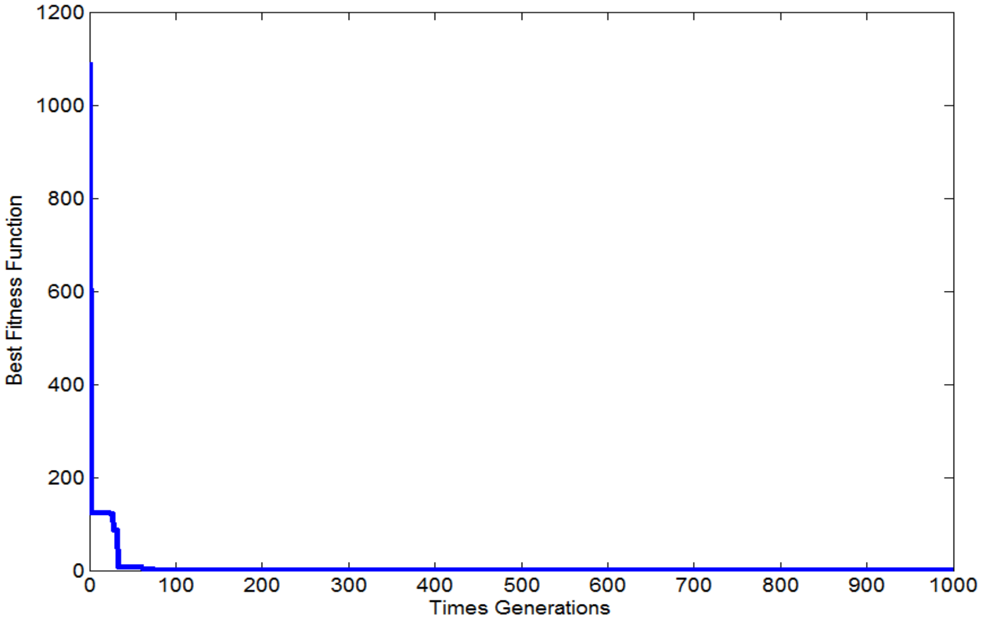


Fig. W-7 Fitness function value.

B1- Yaw Target Fuzzy Control :

The membership functions for inputs and output for the yaw angle controller are shown in Fig. W-8, Fig. W-9 and Fig. W-10, respectively.. The first input is the angle between the target and the centre line of the image. It has seven linguistic variables (VL: very left, L: left, SL: small left, Z: zero, SR small right, R: right, VR: very right). The second input is the difference between the last two measurements with seven linguistic variables (VN: very negative, N: negative, SN: small negative, Z: zero, SP: small positive, P: positive, VP: very positive). The output is the yaw command with nine linguistic variables (VVL: very very left, VL: very left, L: left, SL: small left, Z: zero, SR: small right, R: right, VR: very right, VVR: very very right). The initial rules were 49 rules as they are shown in the Table W-3.

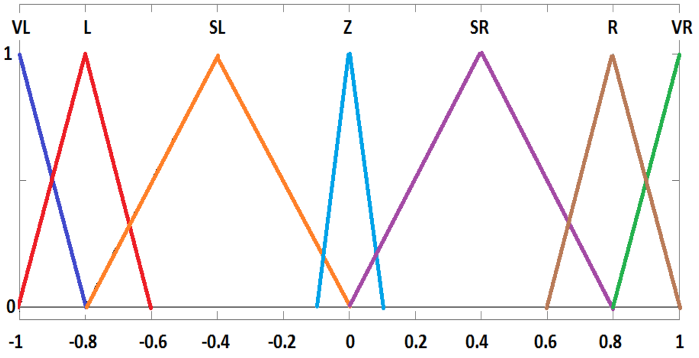


Fig. W-8. The membership functions for input1.

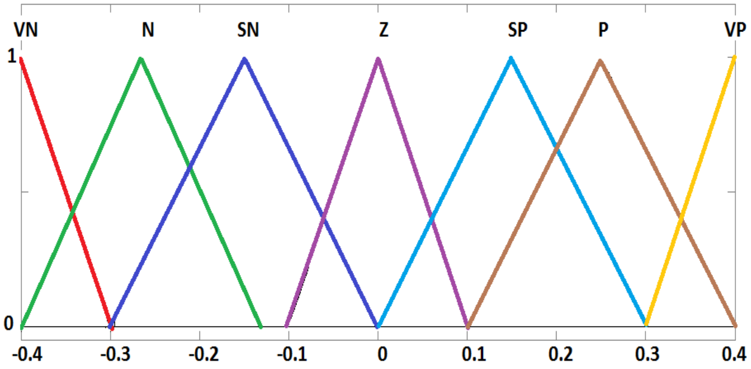


Fig. W-9. The membership functions for input2.

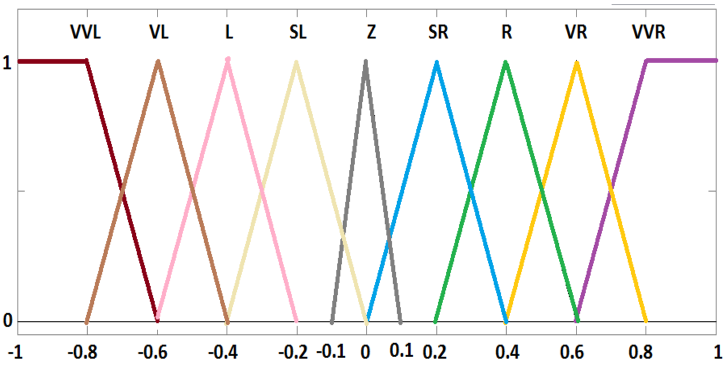


Fig. W-10. The membership functions for Output.

Table. W-3. The Initial Fuzzy Rule Base.

		In1						
In2		*VL*	*L*	*SL*	*Z*	*SR*	*R*	*VR*
	VN	VVL	VVL	L	Z	SR	R	VVR
	N	VVL	VL	SL	Z	SR	R	VVR
	SN	VVL	VL	SL	Z	SR	R	VVR
	Z	VVL	VL	SL	Z	SR	R	VVR
	SP	VVL	VL	SL	Z	SR	R	VVR
	P	VVL	L	SL	Z	R	VR	VVR
	VP	VVL	L	SL	Z	R	VR	VVR

B2- Optimized Yaw Target Fuzzy Control :

The optimized membership functions for inputs and output for the yaw angle controller are shown in Fig. W-11, Fig. W-12 and Fig. W-13, respectively. The optimized rules were 35 rules as they are shown in the Table W-4.

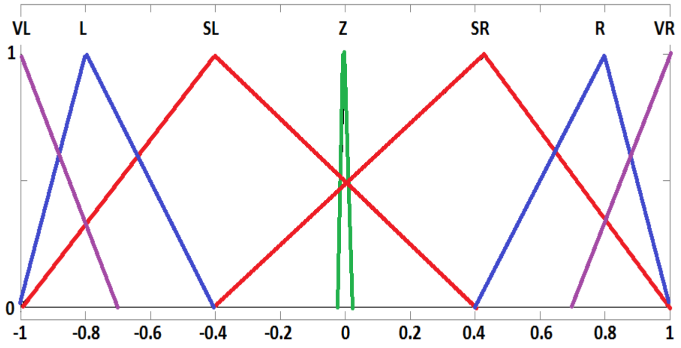


Fig. W-11. The optimized membership functions for Input1.

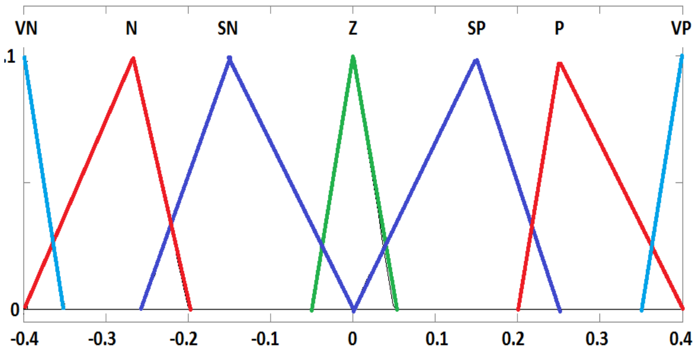


Fig. W-12. The optimized membership functions for Input2.

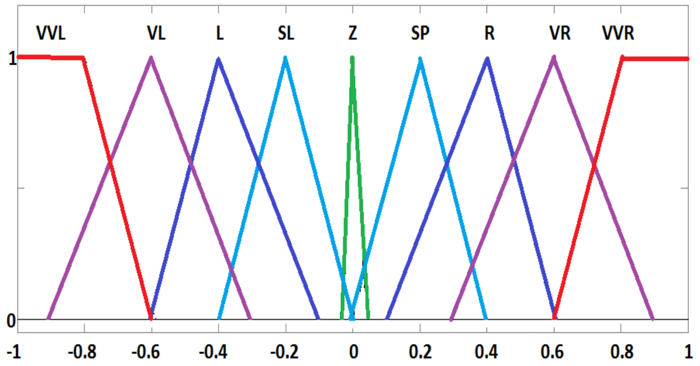


Fig. W-13. The optimized membership functions for Output.

Table. W-4. The Optimized Fuzzy Rule Base.

		In1						
		VL	***L***	***SL***	***Z***	***SR***	***R***	***VR***
In2	***VN***	**VVL**	**X**	**L**	**Z**	**SR**	**X**	**VVR**
	N	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**
	SN	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**
	Z	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**
	SP	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**
	P	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**
	VP	**VVL**	**X**	**SL**	**Z**	**SR**	**X**	**VVR**

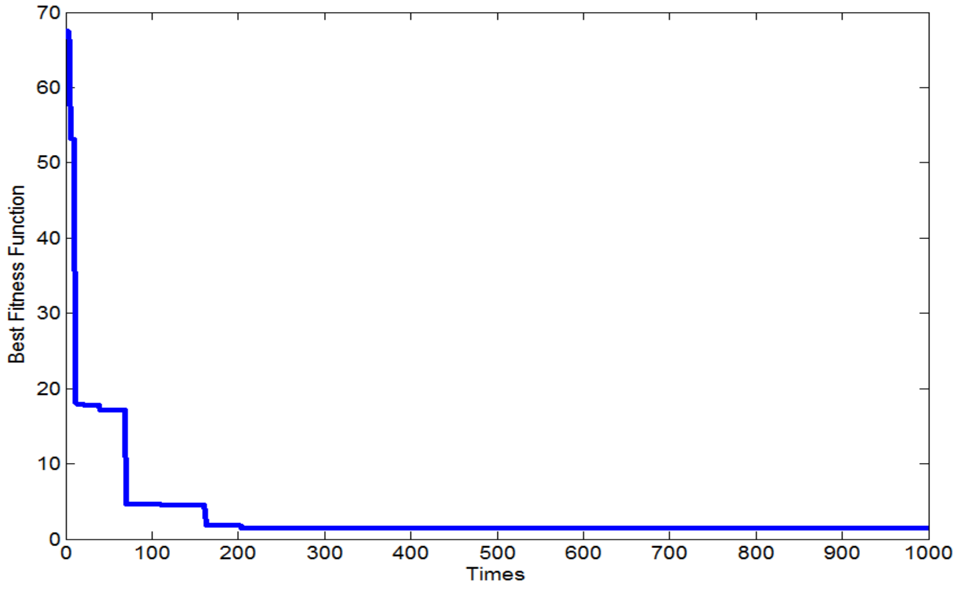


Fig. W-14 Fitness function value.

The fitness function is the absolute error and it was found that the best fitness function value for 1000 generations = 1.4742.

C1- Follow 3D Aerial Target Fuzzy Control :

This controller has two inputs and one output. The membership functions for inputs and output for the vectorization angle controller are shown in Fig. W-15, Fig. W-16 and Fig. W-17, respectively. The first input is the distance between the blimp and the target in metric after we obtain the distances based on **IPM**. It has 6 linguistic variables (Z: Zero, VC: very close, C: close, M: medium, F: far, VF: very far). The second input is the difference between the last two distances with 5 linguistic variables (VL: very low, L: low, M: medium, H: high, VH: very high). The output is the vectorization angle command with four linguistic variables (Z: zero, NOR: normal, PH: positive high, PHH: positive high high). The rules were 30 rules as they are shown in the Table W-5.

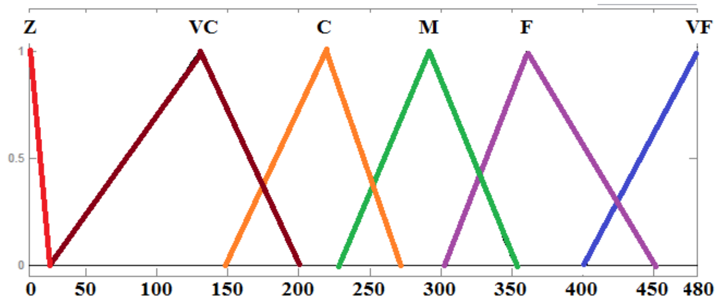


Fig. W-15. The membership functions for Input1.

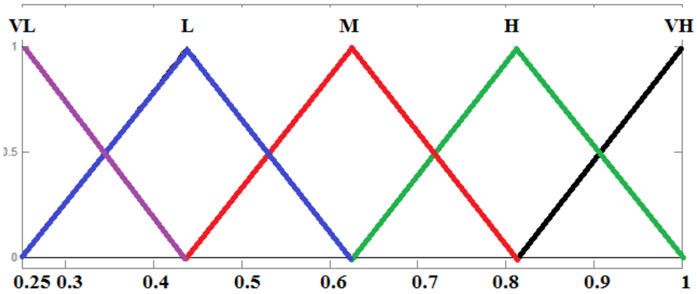


Fig. W-16. The membership functions for Input2.

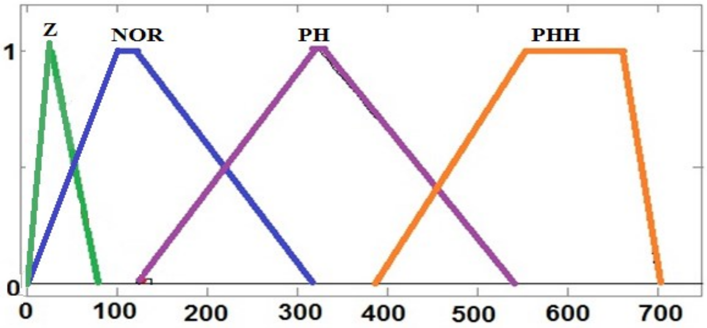


Fig. W-17. The membership functions for Output.

Table. W-5. The Fuzzy Rule Base

		In1					
		Z	*VC*	*C*	*M*	*F*	*VF*
In2	*VL*	Z	Z	NOR	PH	PH	PHH
	L	Z	Z	NOR	PH	PH	PHH
	M	Z	NOR	NOR	PH	PH	PHH
	H	Z	NOR	NOR	PH	PH	PHH
	VH	Z	NOR	NOR	PH	PHH	PHH

D1- Yaw 3D Aerial Target Fuzzy Control:

The membership functions for inputs and output for the yaw angle controller are shown in Fig. W-18, Fig. W-19 and Fig. W-20, respectively. The first input is the angle between the target, centre line of the image. It has five linguistic variables (NH: negative high, N: negative, Z: zero, P: positive, PH: positive high). The second input is the difference between the last two measures angles with five linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). The output is the yaw command with 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). The rules were 25 rules as they are shown in the Table W-6.

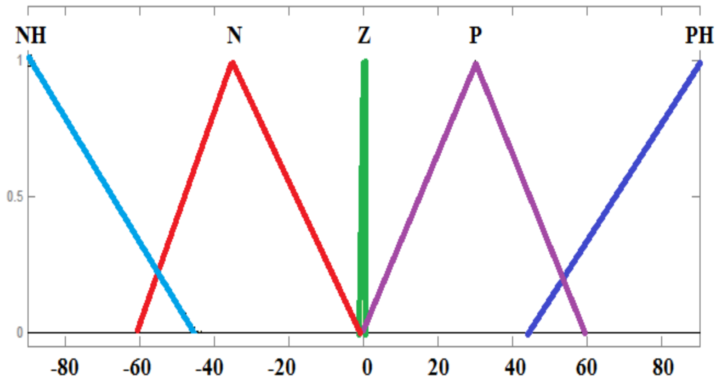


Fig. W-18. The optimized membership functions for Input1.

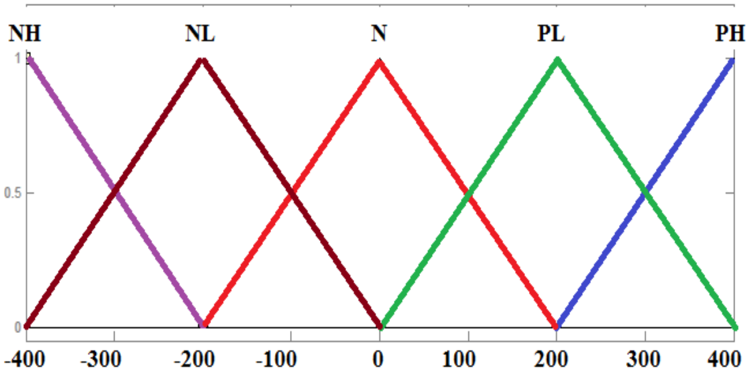


Fig. W-19. The membership functions for Input2.

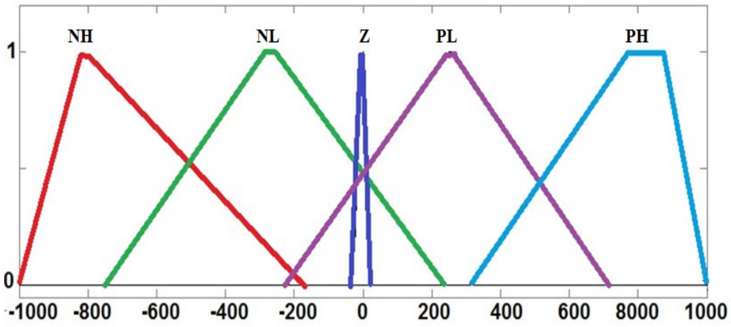


Fig. W-20. The optimized membership functions for Output.

Table. W-6. The Fuzzy Rule Base

		In1				
		NH	***N***	***Z***	***P***	***PH***
In2	***NH***	**NH**	**NL**	**Z**	**PL**	**PH**
	NL	**NH**	**NL**	**Z**	**PL**	**PH**
	Z	**NH**	**NL**	**Z**	**PL**	**PH**
	PL	**NL**	**NL**	**Z**	**PL**	**PH**
	PH	**NL**	**NL**	**Z**	**PL**	**PH**

D2: Fuzzy Control for the main behaviors

A1- Collision Avoidance Controller:

The control has two inputs. First, the error which describes the difference between the required avoidance distance and the shortest distance and it has 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). The second input is the horizontal velocity with 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). Also, it has one output which is the vectorization angle with 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). The membership functions for inputs and output for the controller are shown in Fig. W-21, Fig. W-22 and Fig. W-23, respectively. The initial rules were 25 rules as they are shown in the Table W-7.

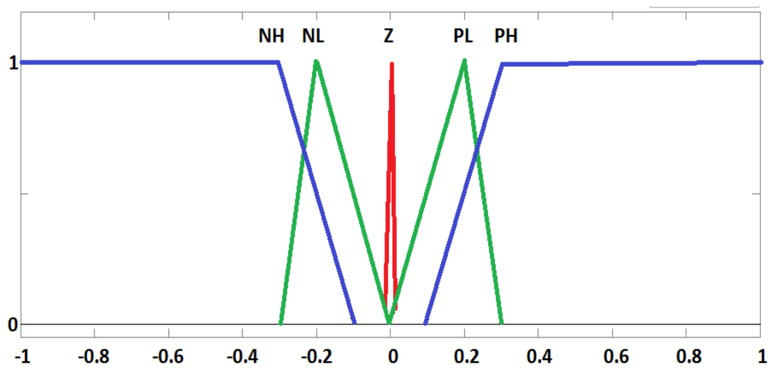


Fig. W-21. The membership functions for Input 1.

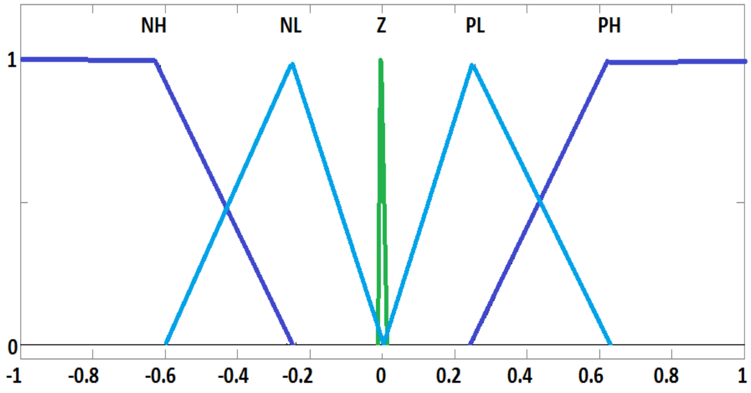


Fig. W-22. The membership functions for Input 2.

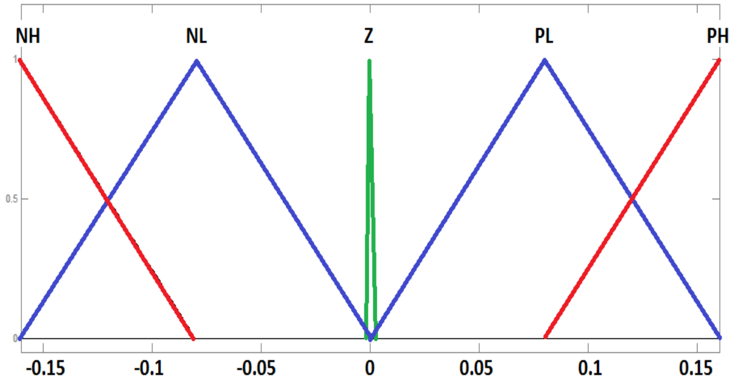


Fig. W-23. The membership functions for Output.

Table. W-7. The Initial Fuzzy Rule Base.

		In 1				
		NH	NL	Z	PL	PH
In 2	NH	PH	PL	PH	PH	PH
	NL	PH	PL	P	PH	PH
	Z	PL	Z	Z	PL	PH
	PL	PL	Z	NL	Z	PL
	PH	Z	Z	NH	N	Z

A2- Optimized Collision avoidance Controller:

The Optimized membership functions for inputs and output for the controller are shown in Fig. W-24, Fig. W-25 and Fig. W-26, respectively. The optimized rules were 25 rules as they are shown in the Table W-8.

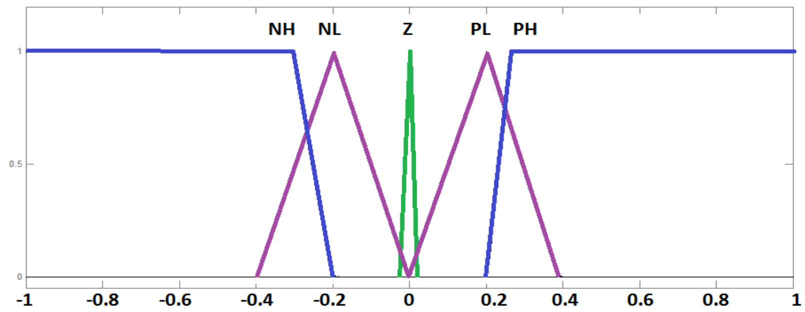


Fig. W-24. The optimized membership functions for Input1.

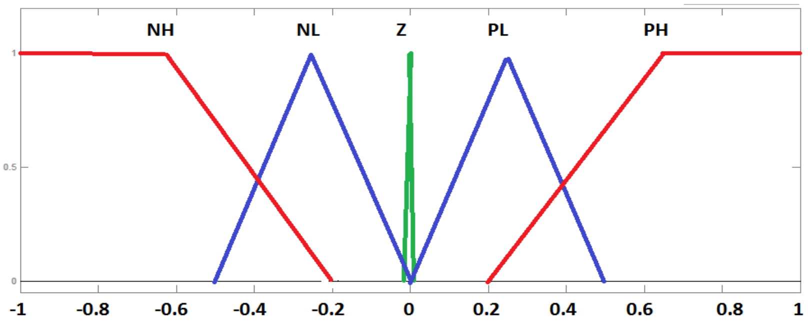


Fig. W-25. The optimized membership functions for Input2.

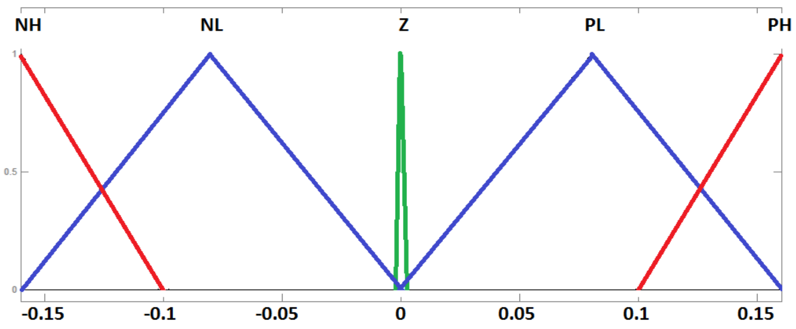


Fig. W-26. The optimized membership functions for Output.

Table. W-8. The Optimized Fuzzy Rule Base.

		In 1				
		NH	NL	Z	PL	PH
In 2	NH	PH	PL	PH	PH	PH
	NL	PH	PL	P	PH	PH
	Z	PL	Z	Z	PL	PH
	PL	PL	Z	NL	Z	PL
	PH	Z	Z	NH	N	Z

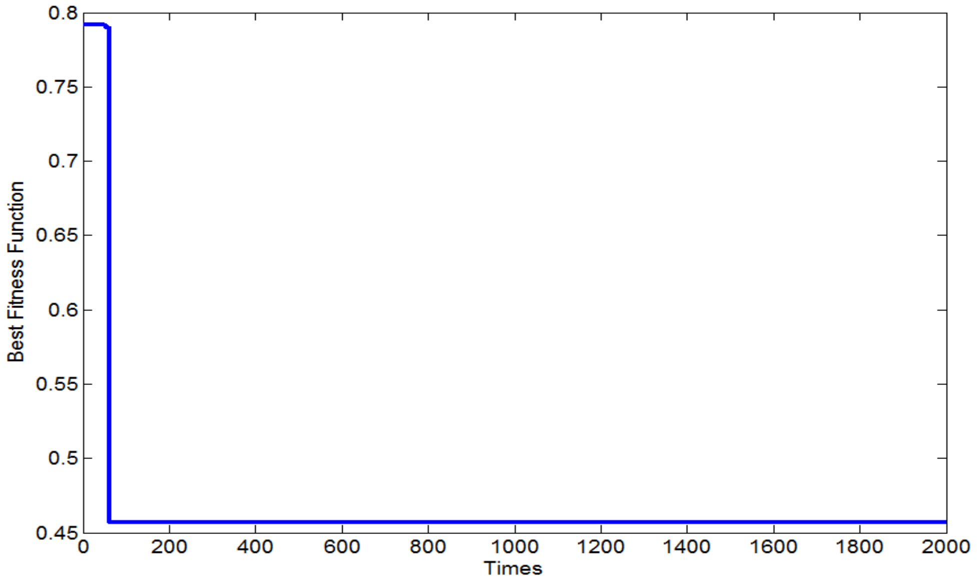


Fig. W-27 Fitness function value.

The fitness function is the absolute errors. It was found that the best fitness function value for 2000 generations = 0.4569.

B1- Altitude Controller:

The altitude controller which has two inputs: Input 1 is the altitude error with 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high) and the Input 2 is the current vertical velocity with 5 linguistic variables (NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high). The altitude error is the difference between the desired altitude and the current shortest altitude. The controller output is the voltage of main propellers with 5 linguistic variables NH: negative high, NL: negative low, Z: zero, PL: positive low, PH: positive high. We should note that few rules have zeros values to avoid the continuous motor action. The membership functions for inputs and output for the vectorization angle controller are shown in Fig. W-28, Fig. W-29 and Fig. W-30, respectively. The initial rules were 25 rules as they are shown in the Table W-9.

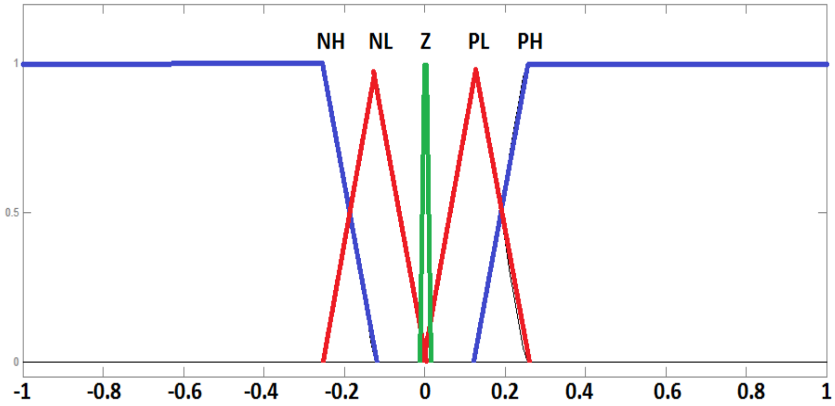


Fig. W-28. The membership functions for Input 1.

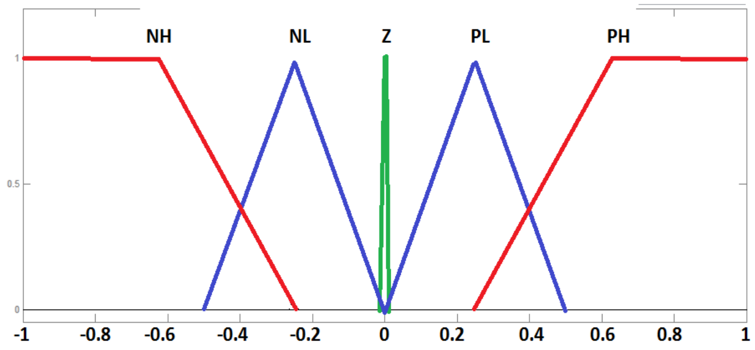


Fig. W-29. The membership functions for Input 2.

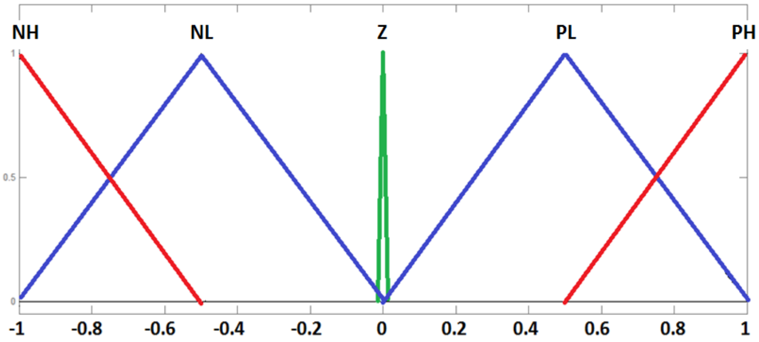


Fig. W-30. The membership functions for Output.

Table. W-9. The Initial Fuzzy Rule Base.

		In 1				
		NH	NL	Z	PL	PH
In 2	NH	PH	PH	Z	Z	Z
	NL	PH	PL	Z	Z	NL
	Z	PH	Z	Z	Z	NH
	PL	PL	Z	Z	NL	NH
	PH	Z	Z	Z	NH	NH

B2- Optimized Altitude Controller:

The Optimized membership functions for inputs and output for the vectorization angle controller are shown in Fig. W-31, Fig. W-32 and Fig. W-33, respectively. The optimized rules were 25 rules as they are shown in the Table W-10.

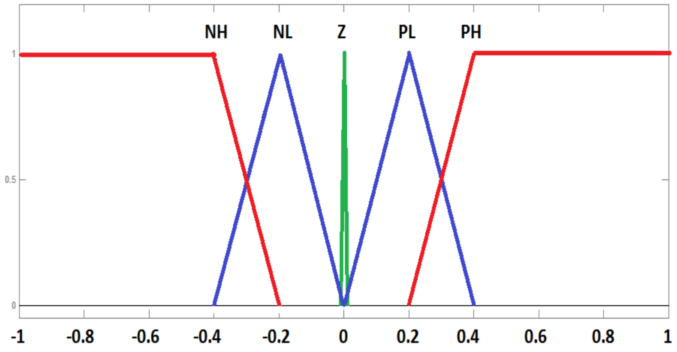


Fig. W-31. The optimized membership functions for Input 1.

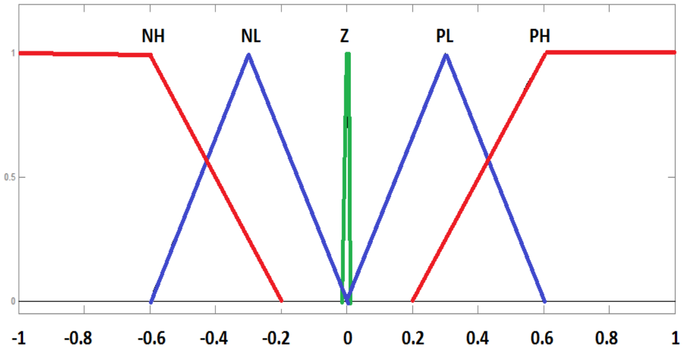


Fig. W-32. The optimized membership functions for Input 2.

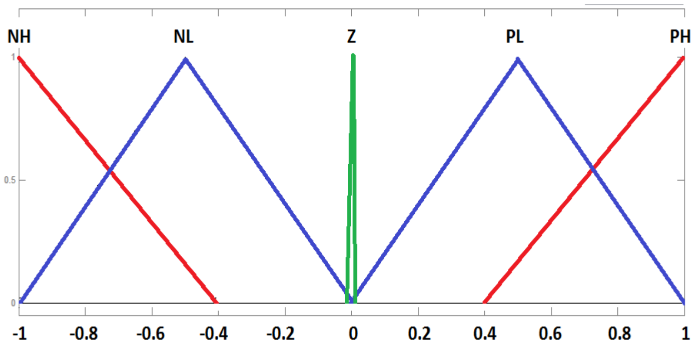


Fig. W-33. The optimized membership functions for Output.

Table. W-10. The Optimized Fuzzy Rule Base.

		In 1				
		NH	NL	Z	PL	PH
In 2	NH	PH	PH	Z	Z	Z
	NL	PH	PL	Z	Z	NL
	Z	PH	Z	Z	Z	NH
	PL	PL	Z	Z	NL	NH
	PH	Z	Z	Z	NH	NH

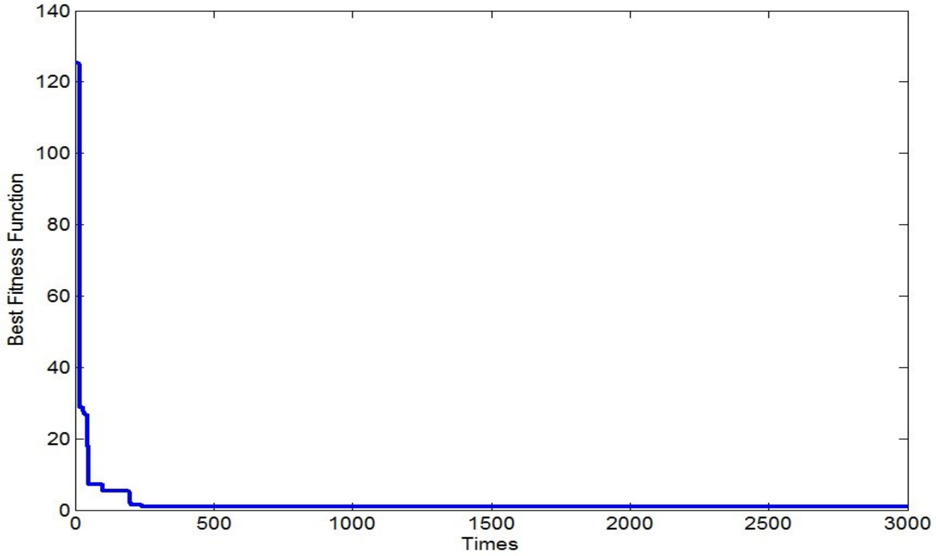


Fig. W-34 Fitness function value.

The fitness function is the absolute errors. It was found that the best fitness function value for 2000 generations = 0.9102.

D3: Possibilities Histograms Design:

Example

Data	Intervals
10	[1,12]
20	[3,14]
30	[0,20]
40	[7,8]

By assuming that we have the previous data table which have been collected experimentally. Thus, the measurement data record can be obtained as the following:

$\widehat{A_J}$**=<[1,12],[3,14],[0,20],[7,8]>, M=4**, and **F={[1,12],[3,14],[0,20],[7,8]}, N=4**. The Set of frequency distribution: **m(A$_1$)= m(A$_2$)= m(A$_3$)= m(A$_4$)= 0.25**. **S= {<[1,12],025>, <[3,14],025>, <[0,20],025>, <[7,8],025>}.** Then, we need to do a reverse order statistics of left and right points. Thus, 1, 3, 0, 7 becomes 0< 1< 3< 7 and 12, 14, 20, 8 becomes 8< 12< 14< 20. In addition, the vector of end points are:

$\overrightarrow{\boldsymbol{E^L}} < 0{,}1{,}3{,}7 >$

$\overrightarrow{\boldsymbol{E^R}} < 8, 12, 14, 20 >$

There are two important conditions to check if the possibilities distribution of these data are fuzzy intervals:

1- $\boldsymbol{L_N = max(L_J) \leq R_N = min(R_J)}$**.** This condition is valid because $7 \leq 8$.

2- The joint linear order on $\widehat{\boldsymbol{E}}$ and its focal is consistent? on other words does $0 \leq 1 \leq 3 \leq 7 \leq 8 \leq 12 \leq 14 \leq 20$? this condition is valid too. Thus, it is consistent.

Then, the possibilities distributions are fuzzy intervals. Therefore, the core of the possibilities is [7, 8] and the support = [0, 20]. $t_k^L = \{< 0, 0.25 >, < 1, 0.25 >, < 3, 0.25 >, < 7, 1 >\}$. $t_k^R = \{< 8, 1 >, < 12, 0.25 >, < 14, 0.25 >, < 20, 0.25 >\}$.

$Cmid = \{< 7.5, 1 >\}, S_1 =< 0, 0 > , S_2 =< 20, 0 >.$

The Measurement records from an interval measuring device for this example are shown in Fig. D3-1 (Top) and the random interval and order statistics are shown in same figure in the bottom. However, the possibilistic histograms is shown in Fig. D3-23. The main steps in order to optimize the fuzzy rule base by bacterial algorithm are shown in Fig. D3-3.

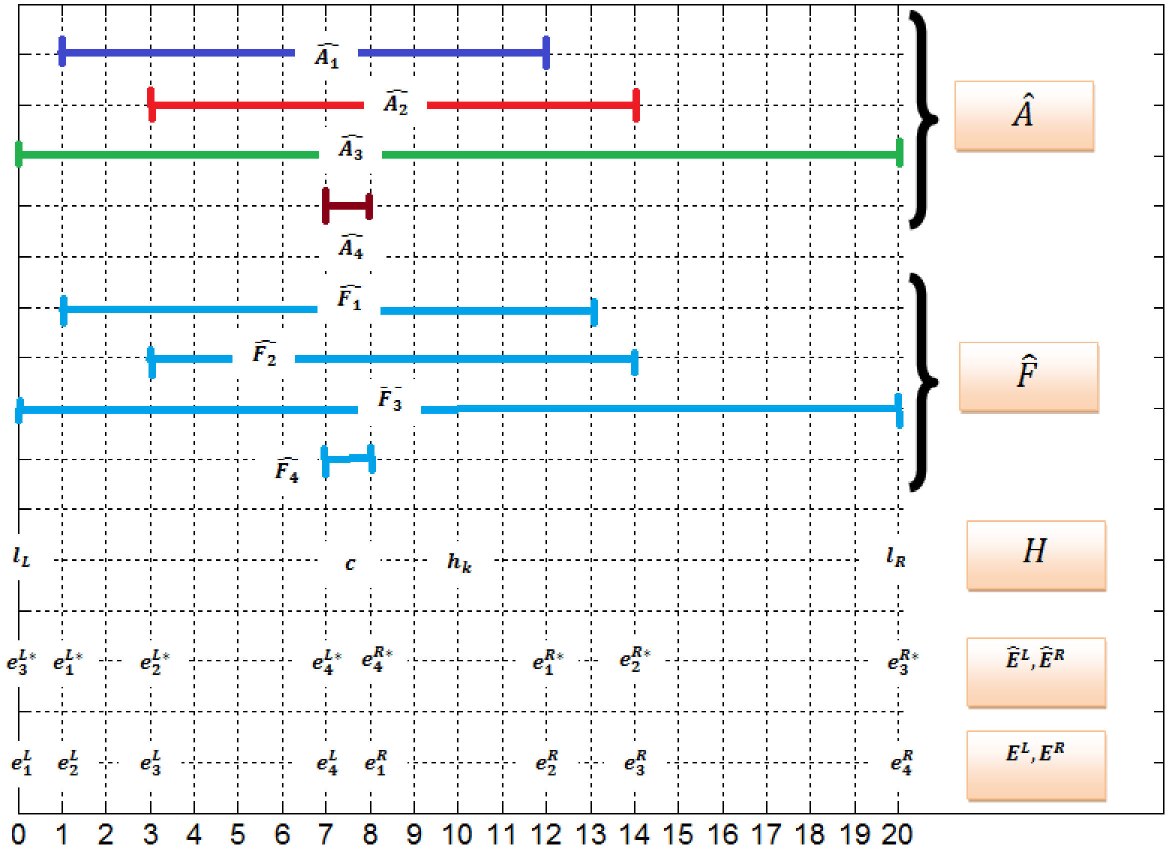


Fig. D3-1 Measurement record and random interval and order statistics.

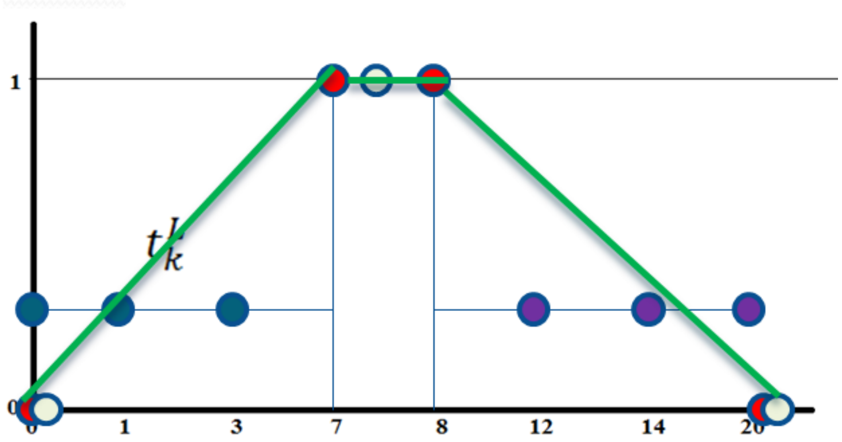


Fig.D3-2 The possibilities Histogram for the previous example.

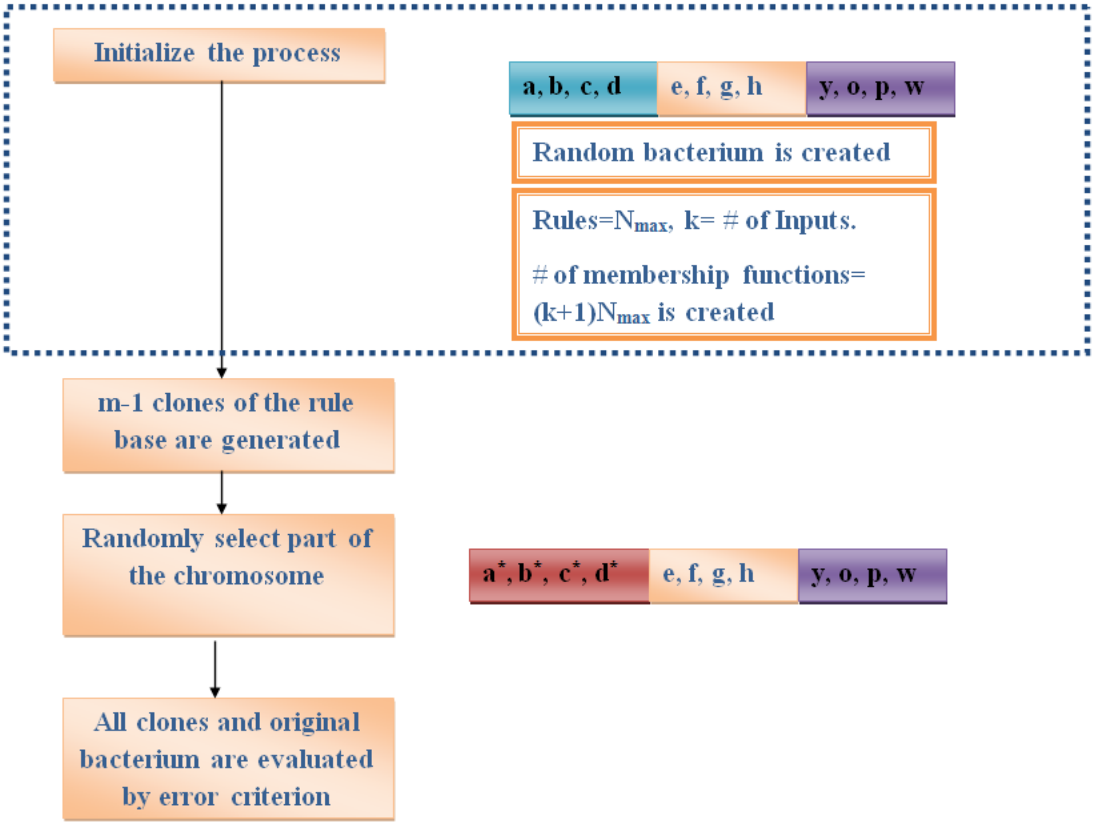

Initialize the process
a, b, c, d
e, f, g, h
y, o, p, w
Random bacterium is created
Rules=N_{max}, k= # of Inputs.
# of membership functions=
(k+1)N_{max} is created
m-1 clones of the rule base are generated
Randomly select part of the chromosome
a*, b*, c*, d*
e, f, g, h
y, o, p, w
All clones and original bacterium are evaluated by error criterion

Fig. D3-3 Flow chart for main bacterial algorithm steps.

In order to optimize the structure of the rule base there are two important operations help to do this step [151]. First, Annihilation: If the membership functions are too narrow, the rule must be deleted. The evaluation criterion is [151]:

$$l_i \left(\frac{a_j + b_j + c_j + d_j}{4} \right) \geq \beta * l_j$$

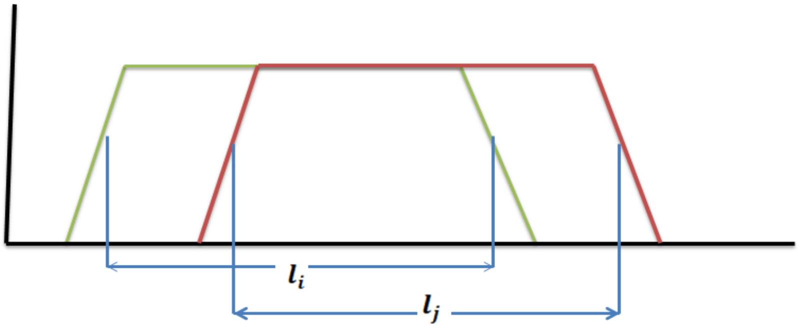


Fig. D3-4 The Annihilation.

Where ***l**$_i$**, l**$_j$**, β*** are two medians of two membership functions as well as the Annihilation parameter, respectively. Second, Fusion : If the difference between the medians of two membership functions is too small, the rule and the membership function will be removed [151] based on the following formula [151]:

$$\left(\frac{l_i}{l_j} - 1 \right) < \gamma \text{ and } |f| < \gamma$$

Where ***f*** is the fusion value which is the difference between the center of the medians and ***ϒ*** is the fusion parameter.

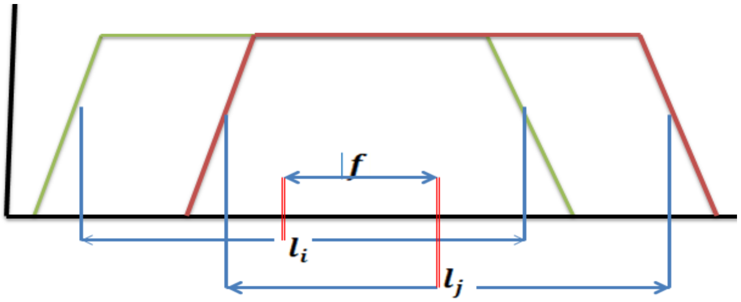


Fig. D3-5 The Fusion.

Appendix E:

Ground Station for Blimp and Robots

The Graphical User Interface (GUI) programming deals with graphical objects called widgets. Looking at the window in the GUI of the system as it is shown in Fig. E-1 it consists of menu, buttons, labels, text fields, and sliders. Each of them is referred to as a widget. All of these elements are precisely arranged and ordered into elements (panels). The panels are ordered into the main elements called Frames. Every new window that the user is able to open is actually a frame that consists of all other elements inside it.

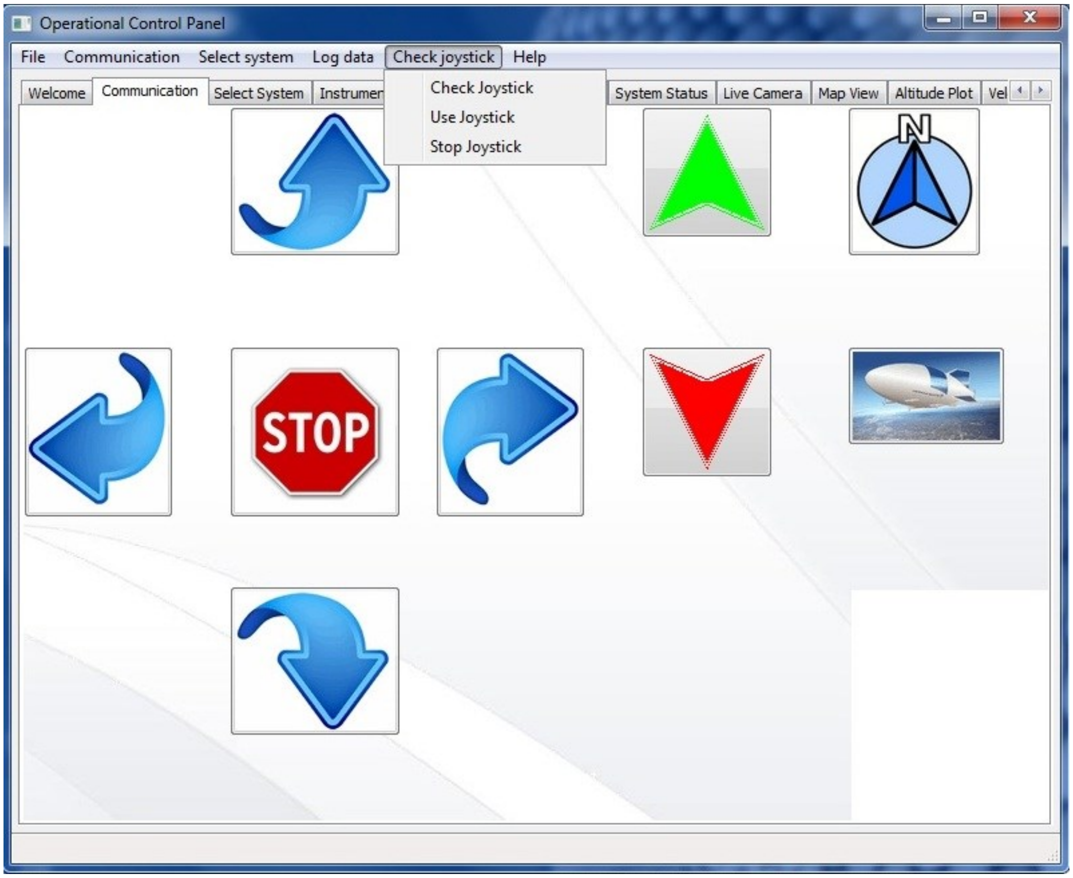
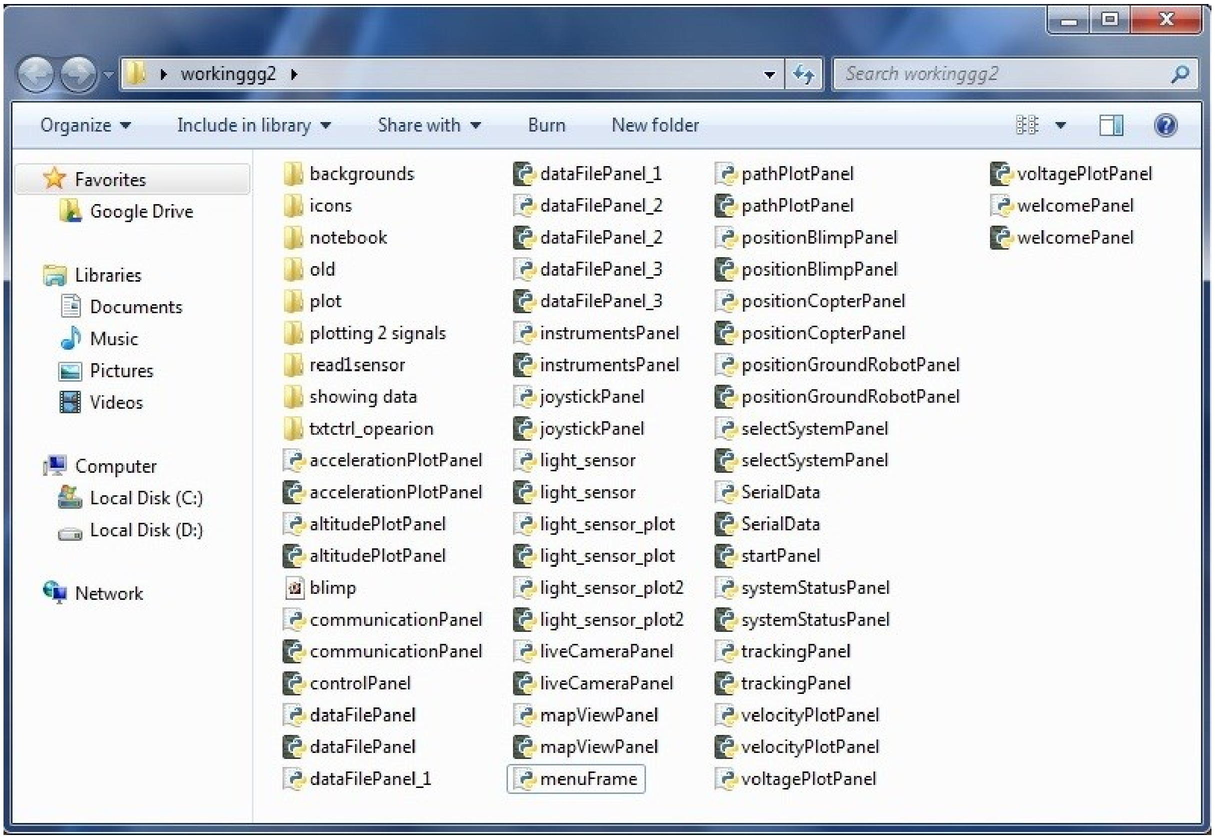


Fig. E-1. The complete Ground Station.

The program consists of 26 Python files (source codes) which are related to each other as it is shown in Fig. E-2. The Images of the buttons and the backgrounds of the different frames are organized in folders, whose paths are described in the source code of the files for proper functioning.



backgrounds
icons
notebook
old
plot
plotting 2 signals
read1sensor
showing data
txtctrl_opearion
accelerationPlotPanel
accelerationPlotPanel
altitudePlotPanel
altitudePlotPanel
blimp
communicationPanel
communicationPanel
controlPanel
dataFilePanel
dataFilePanel
dataFilePanel_1
dataFilePanel_1
dataFilePanel_2
dataFilePanel_2
dataFilePanel_3
dataFilePanel_3
instrumentsPanel
instrumentsPanel
joystickPanel
joystickPanel
light_sensor
light_sensor
light_sensor_plot
light_sensor_plot
light_sensor_plot2
light_sensor_plot2
liveCameraPanel
liveCameraPanel
mapViewPanel
mapViewPanel
menuFrame
pathPlotPanel
pathPlotPanel
positionBlimpPanel
positionBlimpPanel
positionCopterPanel
positionCopterPanel
positionGroundRobotPanel
positionGroundRobotPanel
selectSystemPanel
selectSystemPanel
SerialData
SerialData
startPanel
systemStatusPanel
systemStatusPanel
trackingPanel
trackingPanel
velocityPlotPanel
velocityPlotPanel
voltagePlotPanel
voltagePlotPanel
welcomePanel
welcomePanel

Fig. E-2 Complete Ground Station source files.

The main file for starting the GUI is *menuFrame*. The packages that have to be installed for the proper running of the program are:

- *python-2.7.3* Windows, Linux, Mac etc. Python installer
- *wxPython2.8-win32-unicode-2.8.12.1-py27*. The wx library
- *pyserial 2.6 /Python Serial Port Extension/* - access for the serial port.
- *PIL – 1.1.7* - Python Imaging Library
- *matplotlib-1.2.0.win32-py2.7* - python 2D plotting library
- *numpy-1.6.2.win32-py2.7* - array-processing package

- *PyQt-Py2.7-x86-gpl-4.9.4-1* – GUI classes

The following source code taken from the file "*dataFilePanel.py*" presents the real implementation of the *GridSizer* as it is shown in Fig. E-3.

file: dataFilePanel.py , line: 29 ÷ 42

```
gs = wx.GridSizer(4, 2, 4, 4)

gs.AddMany( [

        (dataFilePanel.blimpPic, 0),

        (wx.StaticText(self, label = "\nSave Blimp Data")),

        (dataFilePanel.blimpPic2, 0),

        (wx.StaticText(self, label = "\nPlot Blimp Data")),

        (dataFilePanel.groundRob1, 0),

        (wx.StaticText(self, label = "\nSave Ground Robot Data")),

        (dataFilePanel.groundRob2, 0),

        (wx.StaticText(self, label = "\nPlot Ground Robot Data")),

        ])

self.SetSizer(gs)
```

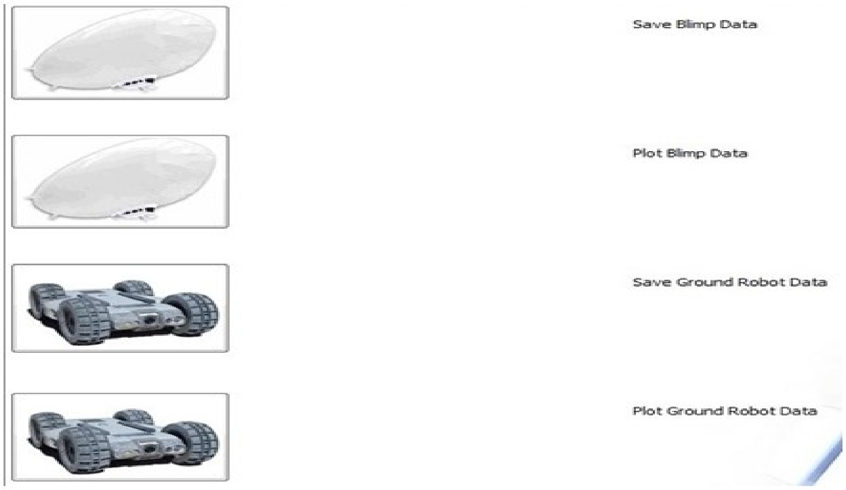


Fig. E-3 The grid sizer.

On the top side, under the menu section of the GUI are placed several choices that the user can select for specific task such as to control either blimp robot or another robot in the system. The GUI comprises of multiple panels – each driven by a separate Python script running in the background – that provide the user ability to switch back. Notebooks are an effective GUI approach, as they allow the user to select the desired view from several options at any time with an instinctive button click. ***wxPython*** supports this feature by supplying a ***wxNotebook*** widget.

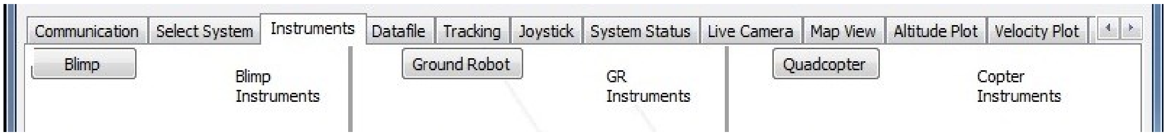
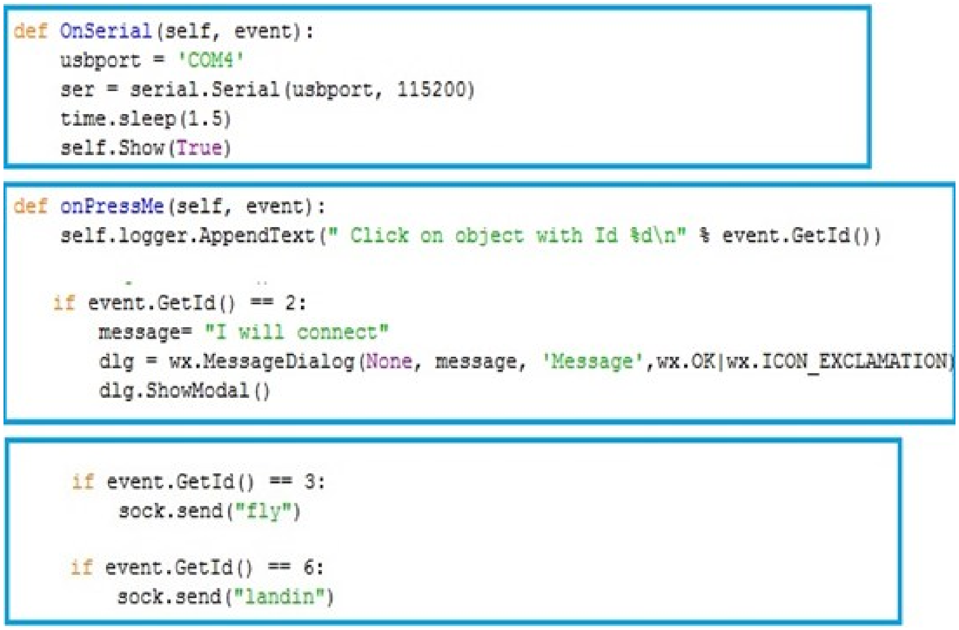


Fig. E-4 Visualization of the Notebook method.

We note here that the *events* parts are essential section of the GUI application as it is shown in Fig. E-5.



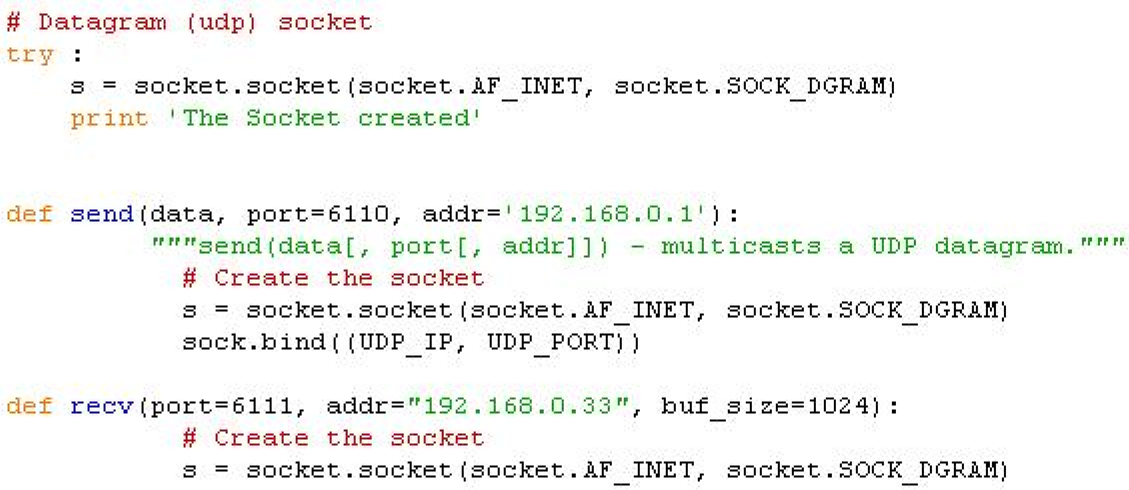
```
def OnSerial(self, event):
    usbport = 'COM4'
    ser = serial.Serial(usbport, 115200)
    time.sleep(1.5)
    self.Show(True)
```

```
def onPressMe(self, event):
    self.logger.AppendText(" Click on object with Id %d\n" % event.GetId())

   if event.GetId() == 2:
       message= "I will connect"
       dlg = wx.MessageDialog(None, message, 'Message',wx.OK|wx.ICON_EXCLAMATION)
       dlg.ShowModal()
```

```
    if event.GetId() == 3:
        sock.send("fly")

    if event.GetId() == 6:
        sock.send("landin")
```

Fig. E-5 Events Application.

To ensure the fast communication between the blimp client (Gumstix Overo Air) and the Ground Station a UDP protocol was chosen.



```
# Datagram (udp) socket
try :
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    print 'The Socket created'


def send(data, port=6110, addr='192.168.0.1'):
         """send(data[, port[, addr]]) - multicasts a UDP datagram."""
           # Create the socket
           s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
           sock.bind((UDP_IP, UDP_PORT))

def recv(port=6111, addr="192.168.0.33", buf_size=1024):
           # Create the socket
           s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

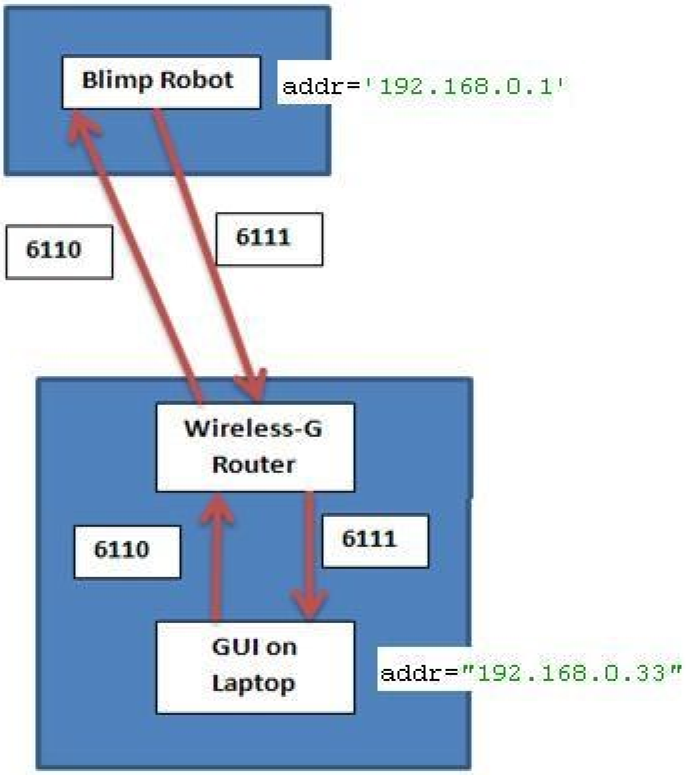


Fig. E-6 The communication between blimp robot and GUI.

The port numbers 6110 and 6111 are chosen for sending and receiving data between the blimp client and GUI, therefore should not exist any conflicts. In addition, the IP address are 192.168.0.0 and 192.168.0.33 within a Local Area Network. When the operator clicks on the button representing the desired device to be controlled (blimp or ground robot), the GUI connects with the corresponding IP-address. After the connection between the ground station and the target robot is established, the chosen robot will wait for the operator commands such as navigation, landing, and stop as it is shown in Fig. E-7 and Fig. E-8.

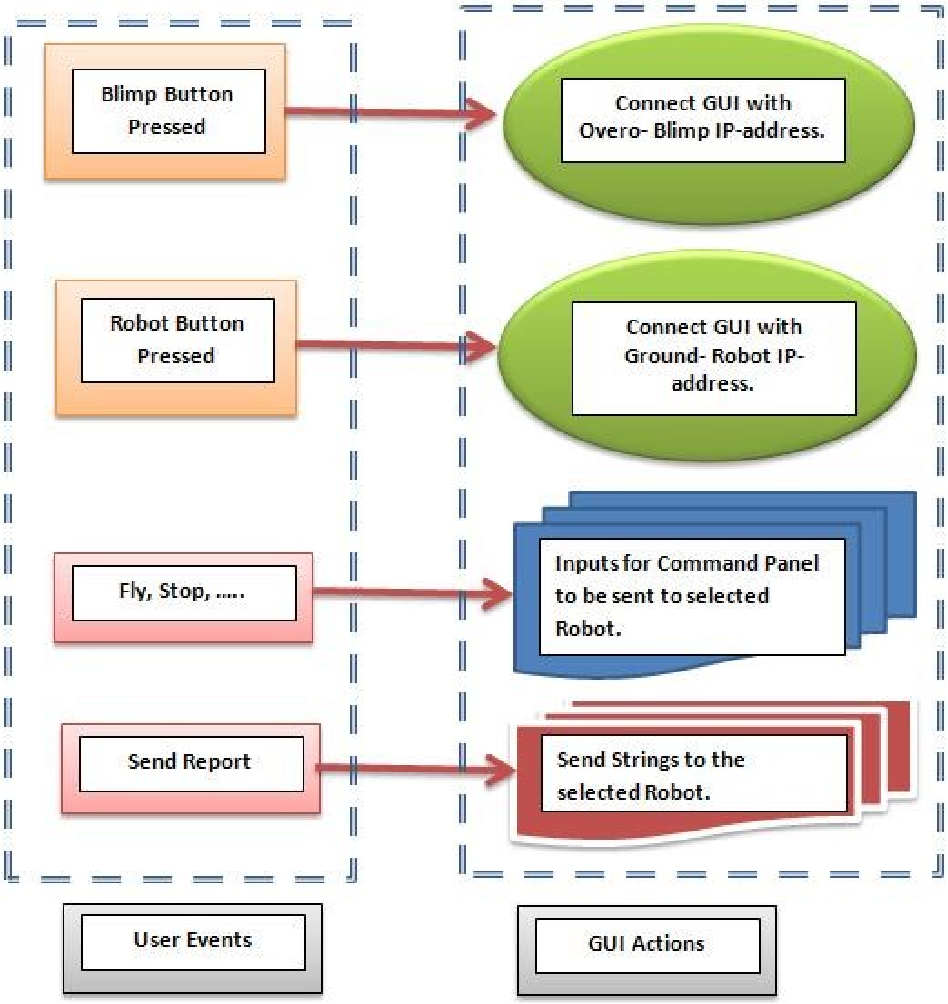

Blimp Button Pressed
Connect GUI with Overo- Blimp IP-address.
Robot Button Pressed
Connect GUI with Ground- Robot IP-address.
Fly, Stop, ...
Inputs for Command Panel to be sent to selected Robot.
Send Report
Send Strings to the selected Robot.
User Events
GUI Actions

Fig. E-7 The Events Diagram.

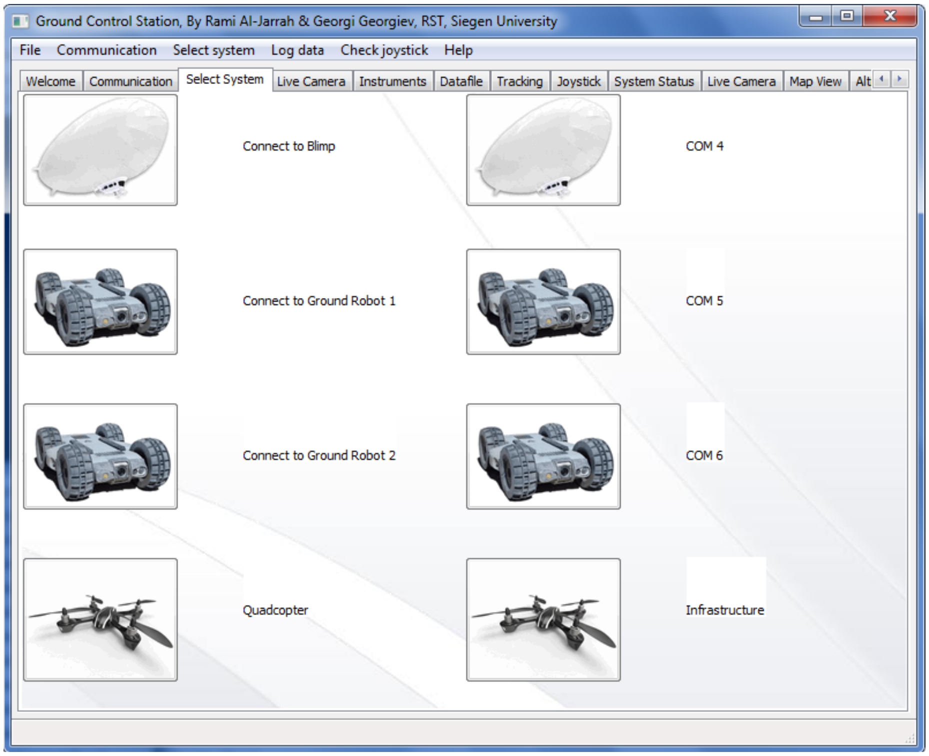


Fig. E-8 The panel to select the target robot.

The Blimp client on its side must send the desired information or commands through the UDP sockets to the GUI. The program contains two threads. The first one is the **I2C_thread** to interface through I2C with the slave Arduino. This is in case that the user requires the aerodynamic data from the sensors that are read by Arduino microcontroller. The I2C is initialized at 400kH.

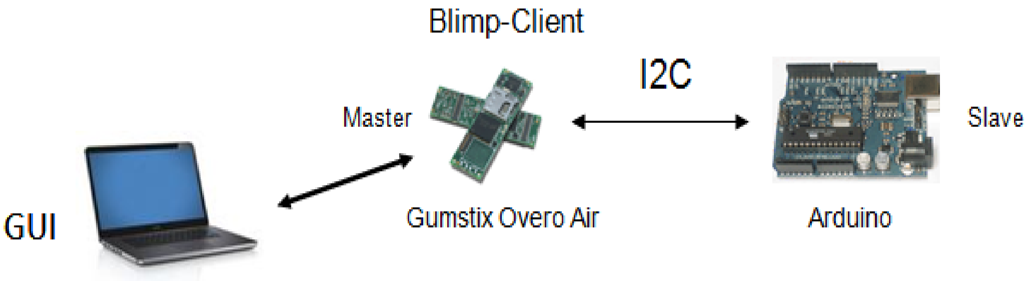


Fig. E-9 The I2C protocol.

Second, the **receive_rcv thread** which is responsible to receive strings through sockets from the GCS and send data back. After receiving the desired command, depending on the user's choice, different integer values are send to the blimp client. The blimp client then sends back to the GUI the data corresponding to that value. For instance integer 1 means the GUI will display the altitude information and integer 3 is the velocity information.

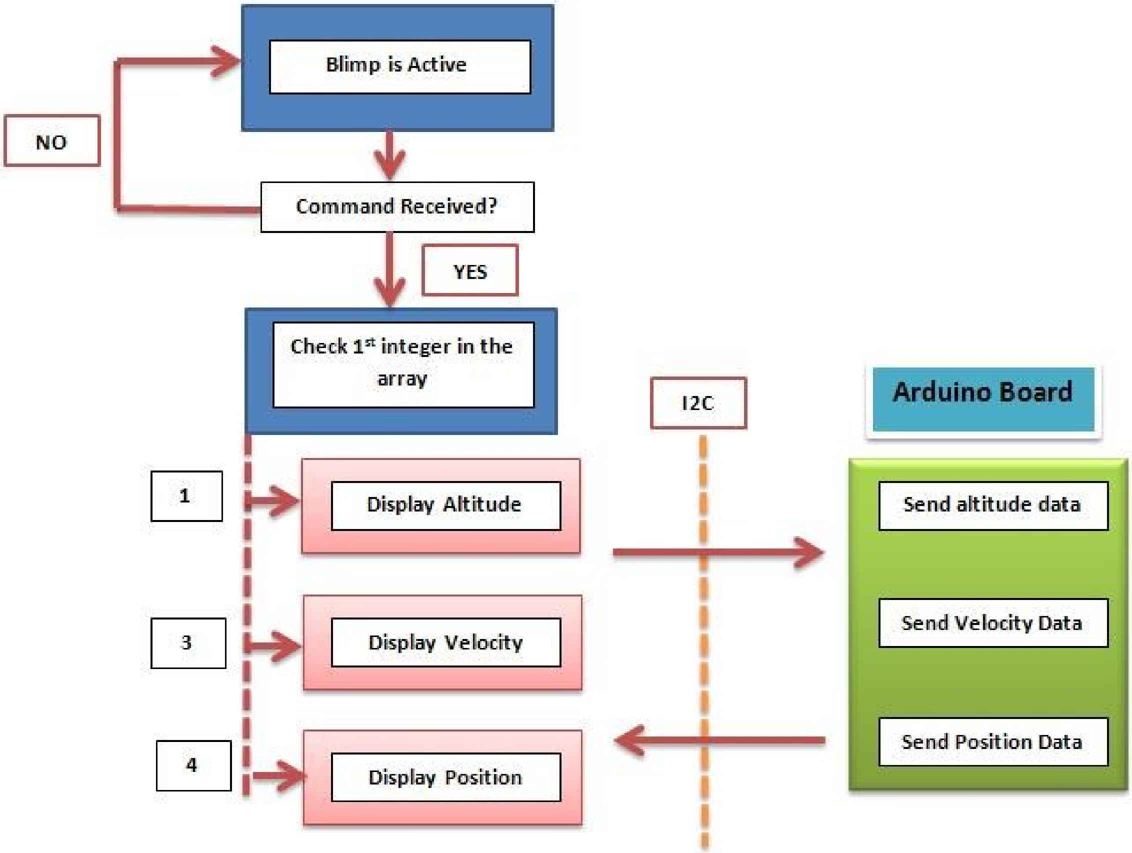
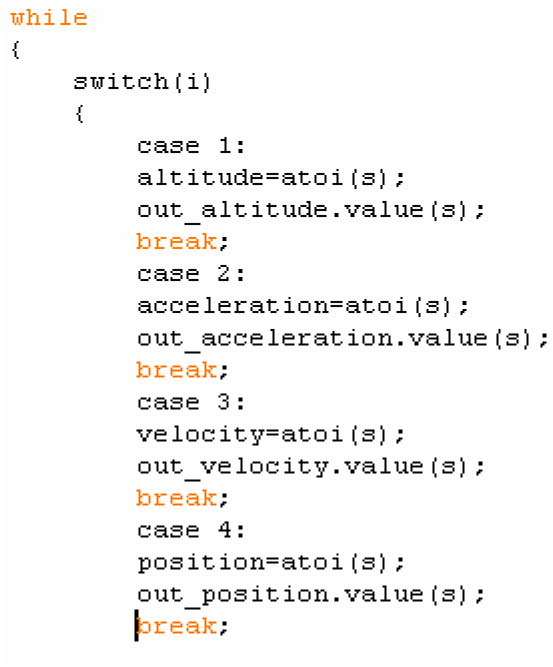


Fig. E-10 Blimp client checks the integer value.

The sample code that represents the Fig. E-10 is the following switch loop where the variable ***i*** has the integer values of the desired data.



```
while
{
    switch(i)
    {
        case 1:
        altitude=atoi(s);
        out_altitude.value(s);
        break;
        case 2:
        acceleration=atoi(s);
        out_acceleration.value(s);
        break;
        case 3:
        velocity=atoi(s);
        out_velocity.value(s);
        break;
        case 4:
        position=atoi(s);
        out_position.value(s);
        break;
```

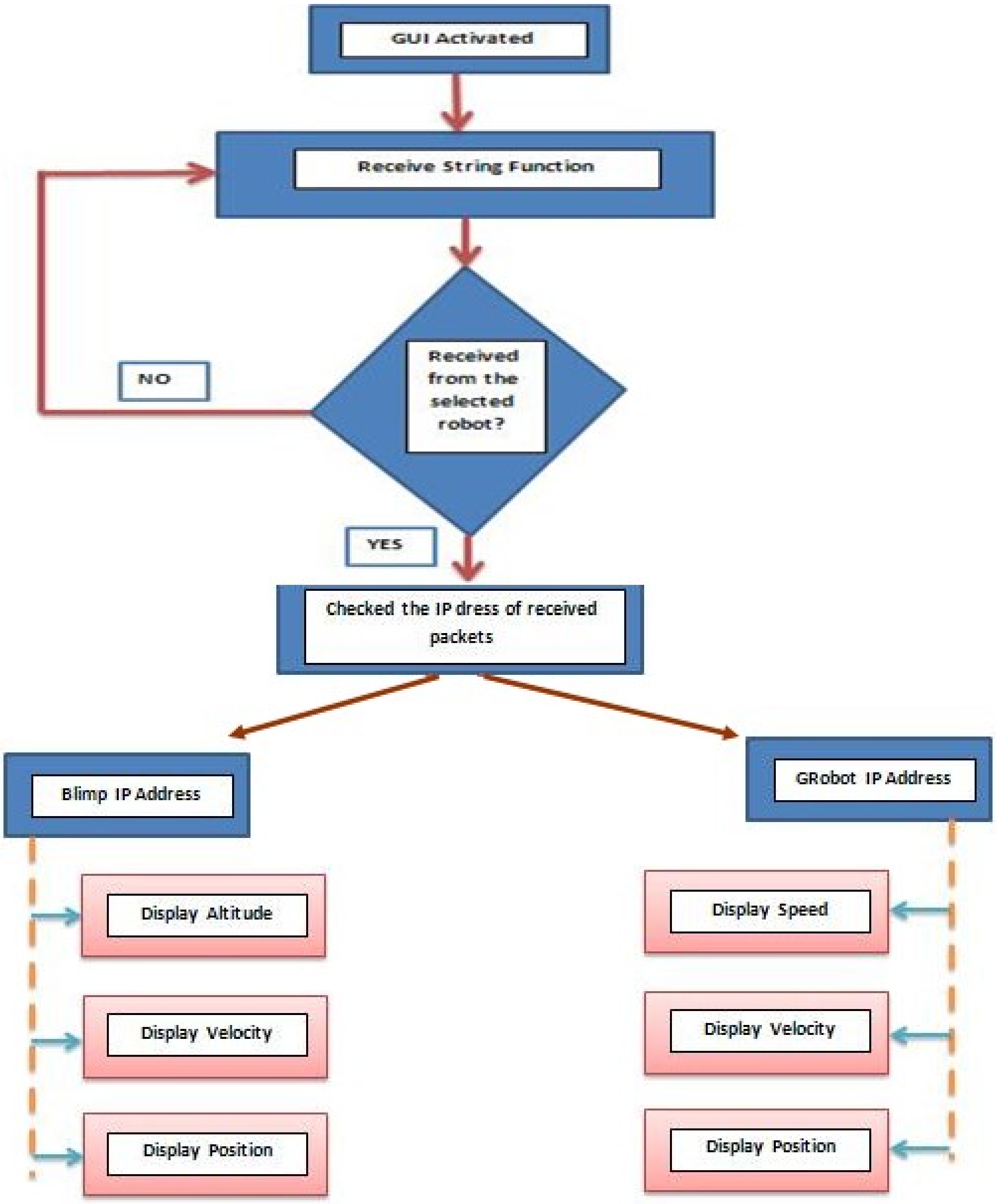


Fig. E-11 The GUI server.

Appendix F:

Sample of Vision Algorithms

F.1. IPM algorithm

The main processes for the IPM are:

1- Generate the IPM characteristics matrix which is developed from the height of the camera from the ground and angle of the camera that is mounted on Gondola.

2- Change the transform image to IPM transform view.

3- Find the distances between the camera and a certain object, then find the linear relation between the pixel in image plane and the real world frame.

The overall system flow chat for the IPM is shown in Fig. F-1. The right side of the flow chat which is responsible for the first frame operations will be done only once time on the first frame.

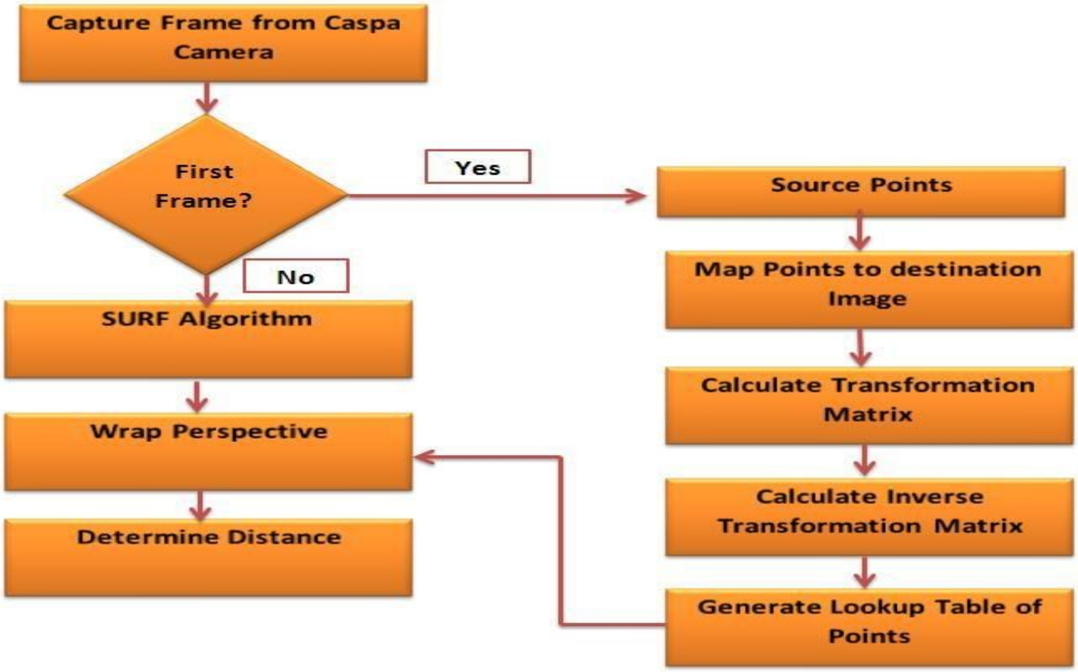
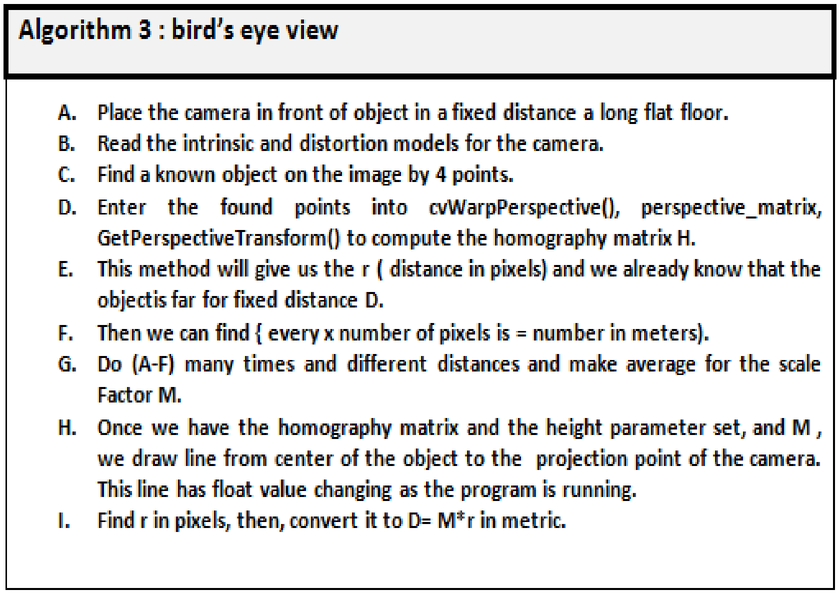

Capture Frame from Caspa Camera
First Frame?
Yes
No
Source Points
Map Points to destination Image
Calculate Transformation Matrix
Calculate Inverse Transformation Matrix
Generate Lookup Table of Points
SURF Algorithm
Wrap Perspective
Determine Distance

Fig. F-1 The system flow chat for IPM.

The process starts by capturing frames by using the Caspa camera that mounted on the blimp robot. When the blimp starts the missions, the algorithm will check the first frame from the video and will start generating many values for the transformation matrix. The process in right side starts by finding 4 points from the source image and stores them in an array. Therefore, these values could be used in order to find the transformation matrix **(RKT)** which is very essential for **IPM**. Since the mathematical for the **IPM** are complex and need manually solved, the **OpenCV** provides us with a function that deals with this matrix {**cvGetPerspectiveTransform()** or **cvGeneratePerspectiveTransform**() }. This function uses the array from previous step to find and generate the desired matrix which is here the transformation matrix in a form of lookup table. The transformation matrix is 3*3 homography matrix and could translate the points from the image plane to world real plane. On other word, this matrix shall map any point **(u, v)** in the image plane to its corresponding point in real word **(X_W, Y_W, Z_W).** On the left side, when the blimp robot detects the target robot based on **SURF** algorithm, the wrap of the images starts by using the function **cvWrapPerspective().** This function will allow the points to map from perspective form to another and it will use the transform matrix in order to generate the destination image with bird's eye view.



Algorithm 3 : bird's eye view

A. Place the camera in front of object in a fixed distance a long flat floor.
B. Read the intrinsic and distortion models for the camera.
C. Find a known object on the image by 4 points.
D. Enter the found points into cvWarpPerspective(), perspective_matrix, GetPerspectiveTransform() to compute the homography matrix H.
E. This method will give us the r (distance in pixels) and we already know that the objectis far for fixed distance D.
F. Then we can find { every x number of pixels is = number in meters).
G. Do (A-F) many times and different distances and make average for the scale Factor M.
H. Once we have the homography matrix and the height parameter set, and M , we draw line from center of the object to the projection point of the camera. This line has float value changing as the program is running.
I. Find r in pixels, then, convert it to D= M*r in metric.

Fig. F-2 The bird's eye view.

In order to calculate the distance between the camera position (Blimp) and the ground robot target, we need to find the linear relation between the metric and pixels by finding a linear relation between them or a scale factor that could convert the pixel to metric distances. This factor could be estimated for a Caspa camera by placing an object in front of the blimp robot (which is here static) or we can mounted the camera on a known position with respect to the object. Thus, measuring the distances in pixels for many tests to find the scale factor for each. Then, taking the average of these factor as it is described in Fig. F-2. The sample code which shows how we can obtain bird's eye view by using OpenCV cab ne found in[181].

Bibliography

[1] T. Xu, X. Zhang and Y. Lu, " Onboard controlling system design of unmanned airship," International Conference on Electronic & Mechanical Engineering and Information Technology, proceedings of the IEEE, vol. 6, pp. 3028- 3031, 2011.

[2] A. Elfes, A. Bueno, and M. Bergerman, "Robotic airships for exploration of planetary bodies with an atmosphere: autonomy challenges," Autonomous Robots Journal, vol. 14, pp. 147-164, 2003.

[3] W. W. Kitt, A. Chekima, and J. Dhargam, F. Wong, and T. A. Tabet, "Soft computing control system of an unmanned airship," Iraq Electrical and Electronic Engineering Journal, vol. 1, pp. 22-27, 2010.

[4] L. Zong-jian and P. Xiao-dong, "Low altitude photogrammetry of unmanned airships," in Proceedings of IEEE International Conference on Information Engineering and Computer Science, pp. 1-4, 2009.

[5] A. Moutinho and J. R. Azinheira, "Stability and robustness analysis of the AURORA airship control system using dynamic inversion," in Proceedings IEEE of International Conference on Robotics and Automation, pp. 2265-2270, 2005.

[6] A. Farinlli, L. Iocchi and D. Nardi, " Multirobot systems: a classification focused on coordination," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 5, pp. 2015 –2028, 2004.

[7] X. Jai and M. Q.-H. Meng," A survey and analysis of task allocation algorithms in multi-robot systems," Proceeding of the IEEE on International Conference on Robotics and Boimimetics, pp. 2280-2285, 2013.

[8] N. Melzer, "Human rights implications of the usage of drones and unmanned robots in warefare," European Union, Printed in Belgium, ISBN: 978-92-823-4390-6, 2013.

[9] B. Browning, and M. Veloso, "Real time, adaptive color based robot vision," Intelligent Robots and Systems, Proceeding of the IEEE/RSJ international Conference on Intelligent Robots and systems, pp. 3871-3876, 2005.

[10] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," IEEE Transactions on Robotics, vol. 24, no. 2, pp. 331–340, 2008.

[11] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer," Real-time vision-based relative aircraft navigation," Journal of Aerospace Computing, Information and Communications, vol. 4, pp. 707–738, 2007.

[12] L. Feng, L. Kai-Yew, C. Ben and L. Tong, "Development of a vision-based ground target detection and tracking system for a small unmanned helicopter," Science in China Series F: Information Sciences, vol. 52, no. 11, pp. 2201–2215, 2009.

[13] A. Korodi, A. Codrean, L. Banita, and C. Volosencu," Aspects regarding the object following control procedure for wheeled mobile robots," WSEAS Transactions on System Control, vol. 3, no. 6, pp. 537–546, 2008.

[14] A. Betser, P. A. Vela, G. Pryor, and A. Tannenbaum, "Flying in formation using a pursuit guidance algorithm," Proceeding of the IEEE on American Control Conference, vol. 7, pp. 5085 – 5090, 2005.

[15] A. Savkin and H. Teimoori, "Bearings-only guidance of an autonomous vehicle following a moving target with a smaller minimum turning radius," Proceeding of the IEE In 47th IEEE Conference on Decision and Control, pp. 4239 –4243, 2008.

[16] T. Fukao, T. Kanzawa and K. Osuka, "Tracking control of an Aerial blimp robot based on image information," 16th IEEE International Conference on Control Applications part of IEEE Multi-Conference on Systems and Control, Singapore, pp. 874-879, 2007.

[17] R. Canals, A. Roussel, J. Famechon, and S. Treuillet, "A bi-processor oriented vision-based target tracking system," IEEE Transactions on Industrial Electronics, vol. 49, no. 2, pp. 500–506, 2002.

[18] L. Mejias, P. Campoy, S. Saripalli and G. S. Sukhatme, " A visual servoing approach for tracking features in urban areas using an autonomous helicopter," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2503-2508, 2006.

[19] W. M. Hu, T. N. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," IEEE Transactions on Systems., Man, Cybernetics, Part C, Applications and Review, vol. 34, no. 3, pp. 334–352, 2004.

[20] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," IEEE Transactions on Industrial Electronics, vol. 56, no. 5, pp. 1401–1410, 2009.

[21] D. Xu, L. Han, M. Tan, and Y. F. Li, "Ceiling-based visual positioning for an indoor mobile robot with monocular vision," IEEE Transactions on Industrial Electronics, vol. 56, no. 5, pp. 1617–1628, 2009.

[22] D. Lowe, "Distinctive image features from scale invariant keypoints," International journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[23] H. Bay, A. Ess, T. Tuyelaars, and L. V. Gool, "Speeded up robust features," Journal of Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359. 2008.

[24] M. Mikolajczyk, and C. Smid, "A performance evaluation of local descriptors," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp.1615-1630, 2005.

[25] D. Nister, O. Naroditsky and J. Bergen," Visual odometry for ground vehicle applications," Journal of field Robotics, vol. 23, no. 1, pp. 3-20, 2006.

[26] F. Bonin-Font, A. Ortiz and G. Oliver," Visual navigation for mobile robots: A survey," Journal of Intelligent and Robotic Systems, vol. 53, pp. 263-296, 2008.

[27] S. Ishikawa, H. Kuwamoto, and S. Ozawa," Visual navigation of an autonomous vehicle using white line recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pp. 743 –749, 2002.

[28] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view sequenced route representation," IEEE International Conference on Robotics and Automation, vol. 1, pp. 83 –88, 1996.

[29] N. X. Dao, B.J. You and S. R. Oh," Visual navigation for indoor mobile robots using a single camera," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1992–1997, 2005.

[30] A. Cherubini and F. Chaumette," Visual navigation with obstacle avoidance," In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1593 –1598, 2011.

[31] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun," Principles of Robot Motion: Theory, Algorithms, and Implementations," MIT Press: Boston, MA, USA, 2005.

[32] A. Chu, M. Blackmore, and R. G. Oholendt," Novel concept for stratospheric communications and surveillance: star light, "Published by American Institute of Aeronautics and Astronautics AIAA Balloon System Conference, pp.1-14, 2007.

[33] W. Li, T. Zhang and K. Kuhnlenz," Vision-guided autonomous quadrotor in an air-ground multi-robot system," In Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, pp. 2980-2985, 2011.

[34] R. Rao, V. Kumar and C. Taylor," Visual servoing of a UGV from a UAV using differential flatness," In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, vol. 1, pp. 743–748, 2003.

[35] M. B. Moseley, B. P. Grocholsky, C. Cheung., S. Singh, “Integrated long-range UAV/UGV collaborative target tracking,” Proceeding on SPIE at Unmanned Systems Technology XI Conference, vol. 7332, 2009.

[36] A. Elfes, M. Bergerman, J.H. Carvalho, E.C. Paiva, J.J. Ramos, S.S Bueno, “Air-ground robotic ensembles for cooperative applications: Concepts and preliminary results,” Proceedings of 2nd International Conference on Field and Service Robotics, Pittsburgh, PA, USA , pp. 75–80, 1999.

[37] C. Phan and H. Liu,” A cooperative UAV/UGV platform for wildfire detection and fighting,” Proceedings on IEEE of the of 7th International Conference on System Simulation and Scientific Computing, Beijing, China, pp. 494–498, 2008.

[38] Y. Lin, J. Hyyppa, T. Rosnell, A. Jaakkola and E. Honkavaara,” Development of a UAV-MMS collaborative aerial-to-ground remote sensing system- A preparatory field validation,” IEEE Journal of selected topics in Applied Earth Observations and Remote Sensing, vol. 6, no. 4, pp. 1893-1898, 2013.

[39] S. Saripalli and G. S. Sukhatme, " Landing a helicopter on a moving target," in Proceedings of IEEE International Conference on Robotics and Automation, Rome, Italy, pp. 2030-2035, 2007.

[40] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel and L. Eck, " Image based visual servo control of the translation kinematics of a quadrotor aerial vehicle," IEEE Transactions on Robotics, vol. 25, no. 3, pp. 743-749, 2009.

[41] H. Choi and Y. Kim, " UAV guidance using a monocular-vision sensor for aerial target tracking ," Journal of Control Engineering Practice, vol. 22, pp. 10-19, 2014.

[42] W. Zhu, Z. Xu, B. Li and X. Ye, " Research on the observability off bearings-only tracking for moving target in constant acceleration based on multiple sonar sensors," Proceeding on IEEE, International Symposium on Information Science and Engineering, pp. 3-6, 2012.

[43] Y. Watanabe, E. N. Johnson and A. J. Calise," Vision-based approach to obstacle avoidance," Published by American Institute of Aeronautics and Astronautics AIAA, Guidance, Navigation, and Control Conference, San Francisco, California, 2005.

[44] C. Cao and N. Hovakimyan, " Vision-Based Aerial Tracking using Intelligent Excitation," Proceeding on IEEE of the American Control Conference, vol. 7, pp. 5091-5096, 2005.

[45] V. Stepanyan and N. Hovakimyan, " A Guidance Law for Visual Tracking of a Maneuvering Target," Proceeding on IEEE of American Control Conference, Minneapolis, MN, pp. 2850-2855, 2006.

[46] V. Stepanyan and N. Hovakimyan, " Adaptive Disturbance Rejection Controller for Visual Tracking of a Maneuvering Target," Journal of Guidance, Control and Dynamics, vol. 30, no. 4, pp. 1090-1106, 2007.

[47] I. F. Mondragon, P. Campoy, M.A. Olivares-Mendez and C. Martinez, " 3D Object following based on visual information for unmanned aerial vehicles," Proceeding on IEEE of the Robotics Symposium, IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications, Bogota, 2011.

[48] M.A. Olivares-Mendez, I.F. Mondragon, P. Campoy, L. Mejias and C. Martinez, " Aerial Object Following Using Fuzzy Servoing," In proceeding of 1st Workshop on Research, Development and Education on Unmanned Aerial Systems RED-UAS, Seville, Spain, pp. 61-70, 2011.

[49] A. Behrad, A. Shahrokni, and S.A. Motamedi," A robust vision-based moving target detection and tracking system," In the Proceeding of Image and Vision Computing Conference, University of Otago, Dunedin, New Zealand, November 2001.

[50] I. Cohen and G. Medioni," Detecting and tracking objects in video surveillance," In Proceeding of the IEEE of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 319-325, 1999.

[51] G. L. Foresti and C. Micheloni, " A robust feature tracker for active surveillance of outdoor scenes," Journal of Electronic Letters on Computer Vision and Image Analysis, vol. 1, no. 1, pp. 21-34, 2003.

[52] J. Kang, I. Cohen, and G. Medioni," Continuous multi-views tracking using tensor voting," In Proceedings of the IEEE Workshop on Motion and Video Computing, pp. 181-186, Orlando, Florida, 2002.

[53] H. Koyasu, J. Miura, and Y. Shirai," Real time omnidirectional stereo for obstacle detection and tracking in dynamic environments," In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 31-36, Maui, Hawaii, 2001.

[54] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, " Moving target classification and tracking from real-time video," In Proceeding of the IEEE Workshop on Applications of Computer Vision, pp. 8-14, Princeton NJ, 1998.

[55] D. Murray and A. Basu," Motion tracking with an active camera," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, no. 5, pp.449-459, Current Version 2002.

[56] L. J. Guibas, "Sensing, tracking, and reasoning with relations," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 73-85, 2002.

[57] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and V. Lesser," Distributed sensor network for real time tracking," In Proceedings of the 5th International Conference on Autonomous Agents, pp. 417-424, 2001.

[58] D. Li, K. D. Wong, Y. H. Hu, and A.M. Sayeed," Detection, classification and tracking of targets in distributed sensor networks," IEEE Signal Processing Magazine, vol.19, no. 2, pp. 17-29, 2002.

[59] J. Moore, T. Keiser, R. Brooks, S. Phoha, D. Friedlander, J. Koch, and N. Jacobson," Tracking targets with self-organizing distributed ground sensors," In Proceedings of IEEE Aerospace Conference, vol. 5, pp. 2113-2123, 2003.

[60] F. Zhao, J. Shin, and J. Reich," Information-driven dynamic sensor collaboration," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 61-72, 2002.

[61] F. E. Melo, J. F. Brancalion and K. H. Kientz" Augmentation to the extended Kalman-Bucy filter for single target tracking," Proceeding in IEEE of the 13th Conference on Information fusion, pp. 1-8, 2010.

[62] T. Zhang and S. Fei, "Target tracking based on particle filter algorithm with multiple cues fusion," Proceeding in IEEE of the 26th Chinese Control and Decision Conference, pp. 1660-1665, 2014.

[63] D. A. Prasetya and T. Yasuno, " Cooperative Control of Multiple Mobile Robot Using Particle Swarm Optimization For Tracking Two Passive Target," Proceeding in IEEE of the SICE Annual Conference 2012, Akita, Japan, pp. 1751-1754, 2012.

[64] Y. Shi, B. Liang, X. Wang, W. Xu and H. Liu," Modeling and simulation of space robot visual servoing for autonomous target capturing," Proceeding in IEEE of the International Conference on Mechatronics and Automation, pp. 2275-2280, 2012.

[65] J. Hu, N. Y. Chang, J. Yang, J. J. Wang, R. G. Lossio," FPGA-based embedded visual servoing platform for quick response visual servoing," In Proceedings of the IEEE 8th Asian Control Conference, pp. 263-268, 2011.

[66] C. Copot, C. Lazar and A. Burlacu, " Predictive control of nonlinear visual servoing systems using image moments," In Proceedings of the IEEE/IET Journal of Control Theory and Applications, IET, pp. 1486-1496, 2012.

[67] Y. Fang, X. Liu and X. Zhang," Adaptive active visual servoing of nonholonomic mobile robots," IEEE Transactions on Industrial Electronics, vol. 59, no. 1, pp. 486-497, 2011.

[68] K. Maeda, M. Minami, A. Yanou, H. Matsumoto, F. Yu and S. Hou," Frequency response experiments of 3-D pose full tracking visual servoing with eye-vergence hand-eye robot system," Proceeding in IEEE of the SICE Annual Conference, Akita, Japan, pp. 101-107, 2012.

[69] B. Jung," Cooperative target tracking using mobile robots," A Dissertation for Doctor of Philosophy, Computer Science, University of Southern California, pp. 14, 2005.

[70] C. Tomasi and T. Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Shape and Motion from Image Streams, Carnegie Mellon University, Pittsburgh, PA, 1991.

[71] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor," Indoor localization and visualization using a huma-operated backpack system," In Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation, pp. 1-10, 2010.

[72] H. Liu, S. Chen, Y. Zhang and J. Chen," Modeling radar tracking features and low observability motion planning for UCAV," In Proceedings of the IEEE 4th International Conference on Intelligent Human- Machine Systems and Cybernetics, vol. 2, pp. 162-166, 2012.

[73] D. B. Lucas and T. Kanade," An iterative image registration technique with an application to stereo vision," In Proceedings of the Imaging Understanding Workshop, pp. 121-130, 1981.

[74] P. Cmko and D. Capson," Biomimetic measurement of optical flow and centroid for visual-servo control of hover flight," In Proceedings of IEEE International Conference on Instrumentation and Measurement Technology, pp. 348-353, 2012.

[75] S. Denman, C. Fookes and S. Sridharam," Improved simultaneous computation of motion detection and optical flow for object," In Proceedings of the IEEE International Conference on Digital Image Computing: Techniques and Applications, pp. 175-182, 2009.

[76] C.-H. Chen, and Y.-P. Chan," Real time multi-target visual tracking based on velocity segmentation technique," Proceeding in IEEE of the 33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 2813-2817, 2007.

[77] C.-M. Huang and M.-H. Hung," Target motion compensation with optical flow clustering during visual tracking," In Proceedings of the IEEE of the 11th International Conference on Networking, Sensing and Control, pp. 96-101, 2014.

[78] V. Kumar, D. Rus and G. S. Sukhatme, “Networked Robots," Springer Handbook of Robotics, Part E, Ch. 41. Networked Robots, pp. 943–958, 2008.

[79] P. González, W. Burgard, R. Sanz, and J.L. Fernández," Developing a low cost autonomous indoor blimp," Journal of Physical agents, vol. 3, no. 1, 2009.

[80] M. Falahpour, H. Moradi, H.-H. Refai, M. Atiquzzaman, R. Rubin, and P. G. LoPresti, " Performance comparison of classic and fuzzy logic controllers for communication airships ," Proceeding in IEEE of the Digital Avionics Systems Conference, pp. 4.A.6-1-4.A.6-8, 2009.

[81] Y. YANG, J. WU, and W. Zheng, " Trajectory tracking for an autonomous airship using fuzzy adaptive sliding mode control," Journal of Zhejiang University-SCIENCE C (Computers & Electronics), vol. 13, no. 7, pp. 534-543, 2012.

[82] J. Cliff, " In Support of an Independent Possibility Theory," In: Foundations and Applications of Possibility Theory. Edited by G. de Cooman, D. Raun and E.E. Kerre, World Scientific, Singapore, pp. 152-164, 1995.

[83] W. A. Abdulhafiz and A. Khamis," Bayesian approach to multisensor data fusion with pre and post filtering," 10th IEEE International Conference on Networking, Sensing and Control, pp. 373-378 , 2013.

[84] R. Niu, P. K. Vershney, Q. Cheng," Distribution detection in a large wireless sensor network," Journal of Information Fusion, vol. 7, pp. 380-394, 2006.

[85] Z. Chair and P.K. Varshney, “Optimal Data Fusion in Multiple Sensor Detection Systems,” IEEE Transactions on Aerospace and Electronic System, vol. AES-22, no. 1, pp. 98–101, Jan. 1986.

[86] S.C.A Thomopoulos, R. Viswanathany, and D.C. Bougouliasz,“ Optimal Decision Fusion in Multiple Sensor Systems,” IEEE Trans. on Aerospace and Electronic System, vol. AES- 23, no. 5, pp. 644–653, 1987.

[87] M. A. Al-Jarrah, N. K. Al-Ababneh, M. M. Al-Ibrahim and R. Al-Jarrah," Cooperative OFDM for semi distributed detection in wireless sensor networks, AEU- International Journal of Electronics and Communications, vol. 68, no. 10, pp. 1022- 1029, 2014.

[88] S. Kumar, F. Zhao, and D. Shepherd, “Special issue on collaborative signal and information processing in microsensor networks,” IEEE Signal Processing Magazine, vol. 19, pp. 13-14, 2002.

[89] K. Yao, R.E. Hudson, C.W. Reed, D. Chen, and F. Lorenzelli, “Blind beamforming on a randomly distributed sensor array system,” IEEE Journal on Selected Areas in Communications, vol. 16, no. 8, pp. 1555–1567, 1998.

[90] P. Purusothaman and S. Sivakumar," Fuzzy RSSI based localization in wireless sensor networks," International Journal of Engineering Research and Technology, vol. 2, no. 8, pp. 1323-1328, 2013.

[91] M. Kumar, J.-M. Souza, P.-R. Venketeshwaran, G. Kini, M.-M. Reddy," Combined circle intersection with weighted centroid method for localization of sensor nodes in wireless sensor network," International Conference on Innovations in Electrical and Electronics Engineering, pp. 268-271, 2012.

[92] J.-C. Chen, R.-E. Hudson, and K. Yao, “A Maximum Likelihood Parametric Approach to Source Localization,” Proceeding in IEEE of the International Conference on Acoustics, Speech and Signal Processing (ICASSP2001), Salt Lake City, UT, vol. 5, pp. 3013–3016, 2001.

[93] J.-C. Chen, R.-E. Hudson, and K. Yao, “Maximum- Likelihood Source Localization and Unknown Sensor Location Estimation for Wideband Signals in the Near-Field,” IEEE Transactions on Signal Processing, vol. 50, no. 8, pp. 1843–1854, 2002.

[94] D. Li, K.-D. Wong, Y.-H. Hu, and A.-M. Sayeed, “Detection, Classification, and Tracking of Targets,” IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 17–29, 2002.

[95] C. Meesookho, U. Mitra and S. Narayanan, “ On energy based acoustic source localization for sensor networks,” IEEE Transactions o Signal Processing, vol. 56, no. 1, pp. 365-377, 2008.

[96] Y. Zheng, R. Niu and P.-K. Varshney, “ Closed-form performance for location estimation based on quantized data in sensor networks,” Proceeding in IEEE of the 13th Nonference on Information Fusion, pp. 1-7, 2010.

[97] O. Ozdemir, R. Niu, and P. K. Varshney, “Channel aware target localization with quantized data in wireless sensor networks,” IEEE Trans. on signal processing, vol. 57, no. 3, pp. 1190-1202, 2009.

[98] R. Al-Jarrah, B. Kadhim, A. Shahzad and H. Roth," Towards a heterogeneous navigation team of aerial-ground robots based on fuzzy image processing," International Journal of Materials Science and Engineering, vol. 3, no. 1, appeared at International Conference on Robotics and Mechatronics(ICROM), Nottingham, UK, 2014.

[99] S. Tadokoro," Rescue robotics challenge," Proceeding in 2010 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), , pp. 92 –98, 2010.

[100] R. Al-Jarrah and H. Roth," Design blimp robot based on embedded system & software architecture with high level communication and fuzzy logic," , Proceeding in IEEE of the 9th International Symposium on Mechatronics & its Applications (ISMA13), Amman, Jordan, 2013.

[101] R. Al-Jarrah and H. Roth," Developed blimp robot based on ultrasonic sensors using possibilities distributions and fuzzy logic," Journal of Automation and Control Engineering, vol. 1, no. 2, appeared at the 5th International Conference on Computer and Automation Engineering ICCAE, Bruxelles- Belgium, 2013.

[102] L. Zadah. " Fuzzy sets", Information and Control Journal, vol. 8, pp. 338-353, 1965.

[103] L. A. Zadah, " Outline of a new approach to the analysis of complex systems and decision processes", IEEE Transactions on Systems Man Cybernetics, vol. 3, pp. 28-44, 1973.

[104] M. Sugeno and T. Yasukawa," A fuzzy logic based approach to qualitative modeling," IEEE Transactions on Fuzzy Systems, vol. 1, pp. 7-31, 1993.

[105] J. Bazdek, "Cluster validity with fuzzy sets," Journal of cybernetics, vol. 3, pp. 58-71, 1974.

[106] S.L. Chiu, "Fuzzy model identification based on cluster estimation," Journal of Intelligent and Fuzzy Systems, vol. 22, pp. 267-278, 1994.

[107] M.R. Emami, I. Turksen, and A. Goldenberg," Development of systematic methodology of fuzzy logic modeling," IEEE Transactions on Fuzzy Systems, vol. 6, pp. 346-361, 1998.

[108] K. Demirli and I.-B. Turksen" A review of implication and the generalized modus ponens," Proceeding of the 3rd IEEE conference on Fuzzy Systems, IEEE World Congress on Coputational Intelligence, pp. 1440-1445, 1994.

[109] P.-N. Vo, M. Detyniecki and B.-B. Meunier, " Gradual generalized modus ponens," Proceeding in IEEE of the International Conference on Fuzzy Systems, pp. 1-7, 2013.

[110] A.-G. Hatzimichailidis, V. G. Kaburlasos and B. K. Papadopoulos," An implication in fuzzy sets," Proceeding in IEEE of the International Conference on Fuzzy Sets, Vancouver, BC, Canada, pp. 1-6, 2006.

[111] Z.-H. Shi, Z.-T. Gong and Y.-B. Shao, " A novel similarity measurement of intuitionistic fuzzy sets induced by triangular norm," Proceeding in IEEE of the International Conference on Machine Learning and Cybernetics, vol. 1, pp. 233-237, 2012.

[112] B. Kovalerchuk," Interpretable Fuzzy Systems: Analysis of T-norm interpretability," Proceeding in IEEE of International Conference on Fuzzy Systems, pp. 1-8, 2008.

[113] R. Fuller,“ Introduction to Neuro-Fuzzy System”, Advances in intelligent and Soft Computing, vol. 2, A Springer-Verlag Company, 2000.

[114] L. A. Zadeh, " The concept of linguistic variable and its application to approximate reasoning-I", Information Sciences, vol. 8, pp. 199-249, 1975.

[115] L. A. Zadeh, " The concept of linguistic variable and its application to approximate reasoning-II", Information Sciences, vol. 8, pp. 301-357, 1975.

[116] L. A. Zadeh, " The concept of linguistic variable and its application to approximate reasoning-III", Information Sciences, vol. 9, pp. 43-80, 1975.

[117] B. Schweizer and A. Sklar, " Probabilistic metric spaces", North-Holland, Amsterdam, 1983.

[118] R.R. Yager," Connectives and quantifiers in fuzzy sets," Fuzzy Sets and Systems, vol. 40 pp. 39-75, 1991.

[119] M. Mizumoto, " Fuzzy sets and their operations," Information and Control, vol. 48, pp. 30-48, 1981.

[120] M.M. Gupta and J. Qi," Theory of t-norms and fuzzy inference methods," Fuzzy Sets and Systems, vol. 40, pp. 431-450, 1991.

[120] D. Dubios and H. Prade, “Fuzzy logics and the generalized modus ponens revisited”. Cybernetics and Systems: An International Journal, vol. 15, pp. 293-331, 1984.

[122] L.A. Zadah," PRUF-A Meaning representation language for natural languages," International Journal of Man Machine Studies, vol. 10, pp. 395-460, 1978.

[123] D. Dubios and H. Prade, " Possibility Theory: a approach to computerized processing of uncertainty," Plenum Press, New York, 1986.

[124] D. Dubios and H. Prade, " Fuzzy sets in approximate reasoning, Part I: Inference with possibility distributions," Fuzzy Sets and Systems, vol. 40, pp. 143-202, 1991.

[125] E.H. Mamdani and S. Assilian," An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man Machine Studies, vol. 7, pp. 1-13, 1975.

[126] M. Sugeno and G.T. Kang," Structure identification of fuzzy model," Fuzzy Sets and Systems, vol. 28, pp. 125-33, 1988.

[127] D. Dubois and H. Prade," Fuzzy logics and the generalized modus ponens revisited", The International Journal of Cybernetics and Systems, vol. 15, pp. 293-331, 1984.

[128] I.B. Turksen and Y. Tian," Combination of rules or their consequences in fuzzy expert systems", Fuzzy Sets and Systems, vol. 58, pp. 3-40, 1993.

[129] J. Dezert, P. Wang, and A. Tchamova, " On the validity of Dempster- Shafer theory," Proceeding in IEEE on the 15th International Conference on Information Fusion, pp. 655-660, 2012.

[130] A. P. Dempster, A. P, "Upper and lower probabilities induced by a multivalued mapping". Studies in Fuzziness and Soft Computing Journal, vol. 219, pp. 57-72, 2008.

[131] L.A. Zadeh, K. S. Fu, K. Tanaka and M. Shimura," Calculus of fuzzy restrictions, fuzzy sets and their applications to cognitive and decision processes," Academic Press, New York, pp. 1-39, 1975.

[132] D. Dubois and H. Prade," Fuzzy Sets and Systems: Theory and Applications," Academic Press, New York, 1980.

[133] L. Zadeh," Fuzzy Sets as the basis for a theory of possibility", Journal of Fuzzy Sets and Systems, vol. 1, pp. 3–28, 1978. Reprinted in Fuzzy Sets and Systems 100, pp. 9–34, 1999.

[134] C. Zappia," Non-Bayesian decision theory ante litteram: the case of G. L. S. Shackle," available online, pp. 1-44, 2012.

[135] H. Rott," The value of truth and the value of information: on Isaac Levi's epistemology," Knowledge and Inquiry: Essays on the pragmatism of Isaac Levi, appear in Erik J. Olsson (ed.), Cambridge University Press, 2004.

[136] E. Miranda and G.D Gooman," Epistemic independence in numerical possibility theory," International Journal of Approximate Reasoning, vol. 32, no. 1, pp. 23-42, 2003.

[137] E. Miranda, I. Couso, and P. Gil," Random sets as imprecise random variables," Journal of Mathematical Analysis and Applications, vol. 307, no. 1, pp. 32-47, 2005.

[138] D. Dubios," Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities," Journal of Reliable Computing, vol. 10, pp. 273-297, 2004.

[139] D. Li, C. Zhong and L. Zhang," Fuzzy c-means clustering of partially missing data sets based on statistical representation," Proceeding in IEEE of the International Conference on Fuzzy Systems and Knowledge Discovery, pp. 460-464, 2010.

[140] J. Pearl," On the Consistency Rule in Causal Inference: Axiom, Definition, Assumption, or Theorem?," In Epidemiology, vol. 21, no. 6, pp. 872-875, 2010.

[141] G. Coletti and B. Vantaggo," Comparative models ruled by possibility and necessity: A conditional world," International Journal of Approximate Reasoning, vol. 45, pp. 341-363, 2007.

[142] D. Dubois, and H. Prade," Possibility Theory, Probability Theory and Multiple-valued Logics: A Clarification", Annals of Mathematics and Artificial Intelligence, vol. 32, pp. 35–66, 2001.

[143] A. Josang," Probabilistic Logic under Uncertainty," 13th Computing the Australasian Theory Symposium (CA TS2007), Victoria, Australia and Conferences in Research and Practice in Information Technology (CRPIT), vol. 65, 2007.

[144] G. Shafer," Belief functions and possibility measures," In Bezdek J.C., Analysis of Fuzzy Information, vol. 1, Mathematics and Logic, CRC Press, Boca Raton, FL, Chapter 3, pp. 51-84, 1987.

[145] D. Dubois, H. T. Nguyen, H. Prade," Possibility theory, probability and fuzzy sets: misunderstandings, bridges and gaps," Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series, Kluwer, Dordrecht, pp. 343-438, 2000.

[146] A. A. Alola, M. Tunay and V. Alola," Analysis of possibility theory for reasoning under uncertainty," International Journal of Statistics and Probability, vol. 2, no. 2, pp. 12-23, 2013.

[147] C. Joslyn Cliff," Measurement of Possibilistic Histograms from Interval Data," International Journal of General Systems, vol. 26, no.1, pp. 9-33, republished online 2007.

[148] C. Joslyn Cliff, " Possibilistic measurement and set statistics," in Proceeding of NAFIPS, vol. 2, pp. 458-467, 1992.

[149] C. Joslyn Cliff," Empirical possibility and minimal information distortion" , In Fuzzy Logic: State of the art, editor R. Lowen, Kluwer, 1993a.

[150] C. Joslyn Cliff, " Possibilistic semantic and measurements methods in complex systems," In Proc. on the 2nd International Symposium on Uncertainty Modeling and Analysis, Bilal Ayyub editor, IEEE Computer Society Press, pp. 208-215, 1993c.

[151] J. Botzheim, C. Cabrita, L. T. Kóczy and A. E. Runao," Fuzzy rule extraction by bacterial memetic algorithms," International Journal of Intelligent Systems, vol. 24, no. 3, pp. 312-339, 2009.

[152] G. Gu and G. Peng, “The survey of GSM wireless communication system,” International Conference on Computer and Information Application (ICCIA), pp. 121 –124, 2010.

[153] T. Wypych, R. Angelo, and F. Kuester, “AirGSM: An unmanned, flying GSM cellular base station for flexible field communications,” Proceeding in IEEE of the Aerospace Conference, pp. 1 –9, 2012.

[154] L. Olavarrieta and A. Nava, “Wireless communications: a bird’s eye view of an emerging technology,” Proceeding in IEEE on the International Symposium on Communications and Information Technology ISCIT, vol. 1, pp. 541 – 546, 2004.

[155] " Satellite Signals, Internet VSAT access via satellite: Costs,” [Online]. Available at: http://www.satsig.net/ivsatcos.htm.

[156] "IEEE Global History Network, Wireless 802.11 Wi-Fi.” [Online]. Available at: http://www.ieeeghn.org/wiki/index.php/Wireless_LAN_802.11_Wi-Fi.

[157] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” Proceeding in IEEE of the 33rd IEEE Annual Conference of the Industrial Electronics Society IECON, pp. 46 –51, 2007.

[158] R. Rashid and R. Yusoff, “Bluetooth Performance Analysis in Personal Area Network (PAN),” Proceeding in IEEE of the International RF and Microwave Conference RFM, pp. 393 –397, 2006.

[159] S. Zurbes, W. Stahl, K. Matheus, and J. Haartsen, “Radio network performance of Bluetooth,” Proceeding in IEEE of the International Conference on Communications ICC., vol. 3, pp. 1563 –1567, 2000.

[160] Y. Zheng and Z. Feng, “Simplifications of the Bluetooth radio devices,” Proceedings in IEEE of the 4th International Workshop on Networked Appliances., pp. 107 –115, 2002.

[161] M. Marsan, C. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni,“ Proceedings in IEEE of the Optimizing the topology of Bluetooth wireless personal area networks,” in INFOCOM 2002. 21th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 572 – 579, 2002.

[162] " User Datagram Protocol", [Online]. Available at: http://en.wikipedia.org/wiki/Udp.

[163] "Transmission Control Protocol," 2006, [Online]. Available at: http://en.wikipedia.org/wiki/Transmission_Control_Protocol.

[164] "Internet Assigned Numbers Authority, 2006 Port Numbers, [Online]. Available at: http://www.iana.org/assignments/port-numbers.

[165] " List of TCP and UDP port numbers," available online at: http://en.wikipedia.org/wiki/Port_numbers.

[166] "The Internet Engineering Task Force - Address Allocation for Private Internets," Feb 1996, [Online]. Available at: http://www.ietf.org/rfc/rfc1918.txt.

[167] R. Al-Jarrah, M. A. Al-Jarrah and H. Roth," Multiple 3D Target Tracking in Binary Wireless Sensor Network," International Journal of Materials Science and Engineering, vol. 3, no. 1, appeared at International Conference on Robotics and Mechatronics (ICROM2014), July 6-7, Nottingham, UK, 2014.

[168] H. L. Van Trees," Detection, Estimation, and Modulation Theory," New York: John Willey & Sons, Inc., 2001.

[169] H. V. Poor," An Introduction to Signal Detection and Estimation," 2nd ed. New York: Springer-Verlag, 1994.

[170] C. G. Harris, and M. Stephens," A combined corner and edge detection," In Proceedings of the 4th Alvey Vision Conference, pp. 147-151, 1988.

[171] C. Schmid, R. Mohr, and C. Bauckhage," Evaluation of interest point detectors," International Journal of Computer Vision, Springer Verlag, vol. 37, no. 2, pp. 151–172, 2000.

[172] J. Shi, and C. Tomasi," Good features to track," Proceeding in IEEE of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94), pp. 593-600, 1994.

[173] B. D. Lucas and T. Kanade," An iterative image registration technique with an application to stereo vision," From Proceeding of Image Understanding Workshop, pp. 121-130, 1981.

[174] S. Baker, and I. Matthews," Lucas-Kanade 20 Years On: A Unifying Framework," International Journal of Computing Vision, vol. 56, no. 3, pp. 221-255, 2004.

[175] D. G. Lowe," Object recognition from local scale-invariant features," Proceedings on IEEE of the 17th International Conference on Computer Vision, vol. 2, pp. 1150 - 1157, 1999.

[176] H. Bay, T. Tuytelaars, and L. V. Gool," SURF: Speeded Up Robust Features," In Processing of the ninth European Conference on Computer Vision, pp. 404-417, 2006.

[177] A. Kaya," Outlier analysis in regression models with graphical representations," Ozean Journal of Applied Sciences, vol. 4, no. 2, pp. 121-128 , 2011.

[178] M. A. Fischer, and R. C. Bolles," Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communication of the ACM, vol. 24, no. 6, pp. 381-395, 1981.

[179] H. A. Mallot, H.H. Bulthoff, J.J. Little and S. Bohrer," Inverse Perspective Mapping simplifies optical flow computation and obstacle detection, " Biological Cybernetics, Springer-Verlag, vol. 64, pp. 177-185, 1991.

[180] R. Hartley and A. Zisserman," Multiple View Geometry in Computer Vision," 2nd edition, pp. 153-160. Cambridge University Press, U.K, pp. 153-160, 2003.

[181] G. Bradsky, and A. Kaebler," Learning OpenCV: Computer Vision with the OpenCV Library, Oreilly Media, pp. 408-412, 1st Edition, 2008.

[182] M.M. Gupta, G.K. Knopf, and P.N. Nikiforuk, 'Computer Vision with Fuzzy Edge Perception," International Symposium on Intelligent Control, Philadelphia, pp. 271 - 278, January 1987.

[183] R. Gonzalez and R. E. Wood," Digital Image Processing," Prentice Hall, Upper Saddle River, New Jersey, 3rd edition, 2007.

[184] R. Wang, L. Gao, S. Yang, and Y. Liu, "An Edge detection method by combining fuzzy logic and neural networks," Proceeding in IEEE of the 4th International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4539-4543, 2005.

[185] L. R. Liang and C. G. Looney, "Competitive fuzzy edge detection," Applied Soft Computing Journal, vol. 3, pp. 123-137, 2003.

[186] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[187] D. Marr and E. Hildreth," Theory of edge detection," Pross. Royal Soc., London, vol. 207, pp. 187-217, 1980.

[188] S. Arora and A. Kaur, "Modified edge detection technique using fuzzy inference system," International Journal of Computer Applications, vol. 44, no. 22, pp. 9-12, 2012.

[189] P. Melin, C. I. Gonzalez, I. R. Castro, O. Mendez and O. Castillo, "Edge detection method for image processing based on generalized type-2 fuzzy logic," IEEE Transactions on Fuzzy Systems, vol. PP, no. 99, pp. 1-12, 2014.

[190] C. I. Gonzalez, J. R. Castro, G. E. Martinez, P. Melin and O. Castro, "A new approach based on generalized type-2 fuzzy logic for edge detection," Proceeding in IEEE of the Joint IFSA World Congress and NAFIPS Annual Meeting, pp. 424-429, 2013.

[191] M. Alimohammadi, J. Pourdeilami and A. A. Pouyan, "Edge detection using fuzzy inference rules and first order derivation," Proceeding in IEEE of the 13th Iranian Conference on Fuzzy Systems, pp. 1-5, 2013.

[192] C. I. Gonzalez, J. R. Castro, P. Melin and O. Castro," An edge detection method based on generalized type-2 fuzzy logic," Proceeding on IEEE of the IEEE Symposium on Advanced in Type-2 Fuzzy Logic Systems, pp. 39-44, 2013.

[193] L. Gang, H. Tong-yaun, Y. He, Z. Chuan, D. Jia, and S. Lei, "Fuzzy edge detection of color remote sensing image," Proceeding in IEEE of the International Conference on Automatic Control and Artificial Intelligence, pp. 1098- 1101, 2012.

[194] D. O. Aborisade," Novel fuzzy logic based edge detection technique," International Journal of Advanced Science and Technology, vol. 29, pp. 75-81, 2011.

[195] T. Zoubir, T. Abdelouaheb," A fast edge detection using fuzzy," Proceeding in IEEE of the International Conference on Communications, Computing and Control Applications, pp. 1-5, 2011.

[196] R. Al-Jarrah, B. Kadhim, A. Shahzad, and H. Roth," Towards a Heterogeneous Navigation Team of Aerial-Ground Robots based on Fuzzy Image Processing", International Journal of Materials Science and Engineering, vol. 3, no. 1, 2015.

[197] E.-C. Vladimir," Why so many clustering algorithms," ACM SIGKDD Explorations Newsletter, vol. 4, no. 1, pp. 65–75, 2002.

[198] B. Auffarth," Clustering by a Genetic Algorithm with Biased Mutation Operator," Proceeding on IEEE of the Congress on Evolutionary Computation CEC, , pp. 1–8, 2010.

[199] E. Achtert, C. Böhm, P. Kröger," DeLiClu: boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking," Proceeding of the 10th Pacific-Asian Conference on Advances in Knowledge Discovery and Data Mining, Singapore, pp. 119–128, 2006.

[200] S. Das," Pattern Recognition using Fuzzy c-means Technique," International Journal of Energy, Information and Communications, vol. 4, no. 1, pp. 1-14, 2013.

[201] R. Nock and F. Nielsen," On Weighting Clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, pp. 1223 - 1235, 2006.

[202] J. C. Noordam, W. H. A. M. van den Broek, and L. M. C. Buydens," Geometrically guided fuzzy C-Means clustering for multivariate Image segmentation," Proceeding in IEEE of the International Conference on Pattern Recognition, vol. 1, pp. 462–465, 2000.

[203] J. K. Parker and L. O. Hall," Accelerating fuzzy c-means using an estimated subsample size," IEEE Trans. on Fuzzy Systems, vol. 22, no. 5, pp. 1229- 1244, 2014.

[204] M.-J. Tsai and H.-S. Chang," A color differentiated fuzzy c-means based image segmentation algorithm," Proceeding in IEEE of the IEEE visual Communications and Image Processing, pp. 1-5, 2012.

[205] H. L. Capitaine and C. Frélicot," A fast fuzzy c-means algorithm for color image segmentation," Published by Atlantis Press, Aix-les-Bains, France, pp. 1074-1081, 2011.

[206] R. Al-Jarrah and H. Roth," Visual Fuzzy Control for Blimp Robot to Follow 3D Aerial Object," The 8th International Conference on Neural Network and Artificial Intelligence (ICNNAI 2014), 3-6 June, 2014, Brest State Technical University, Belarus. Published in: Communications in Computer and Information Science, Springer International Publishing, Vol. 440, pp. 98-111, 2014.

[207] R. Al-Jarrah and H. Roth," Ground Robot Navigation Relies on Blimp Vision Information," 3rd International Conference on Information Computer Application (ICIC2014). February 20-21, 2014, Barcelona, Spain, 2014.

[208] R. Al-Jarrah, R. Ait Jellal and H. Roth," Blimp Based on Embedded Computer Vision and Fuzzy Control for Following Ground Vehicles," 3rd IFAC Symposium on Telematics Applications (TA 2013). November 11-13, 2013. Seoul, Korea, 2013.

[209] R. Al-Jarrah, M. A. Al-Jarrah and H. Roth," Visual Fuzzy Logic Path Planning Controller for Mobile Robots," International Journal of Enhanced Research in Science Technology and Engineering. vol. 3, no. 1, pp. 215-221, 2014.

[210] R. Al-Jarrah and H. Roth," Intelligent Tracking and Trajectory Navigation Approach for Blimp Robot," International Journal of Enhanced Research in Science Technology and Engineering. vol. 3, no. 1, pp. 198-204, 2014.

[211] "Gumstix developer center," [Online]. Available at: http://www.gumstix.org/.

[212] "Gumstix, Inc.," [Online]. Available at: https://www.gumstix.com/.

[213] "Welcome to OpenEmbedded," [Online]. Available at : http://www.openembedded.org.

[214] "BitBake User Manual," [Online]. Available at :

http://docs.openembedded.org/bitbake/html.

[215] "Schneiderman Golden Rules," [Online]. Available at:

http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html.

[216] "The 10 usability heuristics," [Online]. Available at:

http://www.nngroup.com/articles/ten-usability-heuristics/.

[217] "Gestalt laws of grouping," [Online]. Available at:

https://www.siggraph.org/education/materials/HyperVis/vision/percorg.htm.

[218] J.-S. R. Jang,C.-T. Sun and E. Mizutani " Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence", Prentice Hall, NJ 07458, 1997.