# DESIGN OF AN AUTONOMOUS NAVIGATION SYSTEM FOR A MOBILE ROBOT

By


**Andre Paul**

**Department of Bioresource Engineering**
**Macdonald Campus**
**McGill University, Montreal**


**June 2005**

**A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Science**

*Your file Votre référence*
*ISBN: 978-0-494-28649-4*
*Our file Notre référence*
*ISBN: 978-0-494-28649-4*

# ABSTRACT

Andre Paul

**M.Sc.** **Bioresource Engineering**

## Design of an autonomous navigation system for a mobile robot

An autonomous navigational system for a mobile robot was developed based on a Laser-Range-Finder-based path planning and navigational algorithms. The system was enhanced by incorporating collision avoidance algorithms using data from a sonar sensor array, and further improved by establishing two virtual regions in front of the robot for obstacle detection and avoidance. Several virtual detector bands with varying dimensions were also added to the sides of the robot to check for rotational clearance safety and to determine the direction of rotation. The autonomous navigational system was tested extensively under indoor environment. Test results showed that the system performed satisfactorily in navigating the mobile robot in three structured mazes under indoor conditions.

An artificial landmark localization algorithm was also developed to continuously record the positions of the robot whilst it was moving. The algorithm was tested on a grid layout of 6 m $\times$ 6 m. The performance of the artificial landmark localization technique was compared with odometric and inertial measurements obtained using a dead-reckoning method and a gyroscope-corrected dead-reckoning method. The artificial landmark localization method resulted in much smaller root mean square error (0.033 m) of position estimates compared to the other two methods (0.175 m and 0.135 m respectively).

# RESUMÉ

Andre Paul

**M.Sc.** **Génie des bioresources**

## Conception d'un système de navigation autonome pour robot mobile

Un système de navigation autonome pour robot mobile fut conçu à partir d'algorithmes de planification de chemin et de navigation. Ce système fut amélioré par l'inclusion d'un dispositif d'évitement de collision se fiant sur des données provenant d'une batterie de capteurs sonar, ainsi que par la création de deux régions-images virtuelles devant le robot, servant à la détection et l'évitement d'obstacles. Plusieures bandes de détecteurs de regions-images virtuelles de différentes dimensions furent ajoutés aux côtés du robot pour servir au contrôle de la zone de sureté de rotation et pour déterminer la direction de rotation. Le système de navigation fut éprouvé à plusieurs reprises à l'intérieur. Ces essais, lors desquels le robot eut à naviguer trois labyrinthes structurés, révélèrent une performance adéquate du système, à l'intérieur.

Une méthode de localisation par point caractéristique artificiel fut conçue pour continuellement enregistrer la position du robot en déplacement. L'algorithme fut éprouvé sur une grille de 6 m × 6 m. La performance de la méthode de localisation par point caractéristique artificiel fut comparée à des mesures de distance parcourue et d'inertie obtenues par des méthodes à l'estime, et à l'estime avec correction gyroscopique. La méthode de localisation par point caractéristique artificiel donna lieu a une erreur quadratique moyenne (0.033 m) largement inférieur à celles obtenues pour les deux autres méthodes, soit 0.175 m et 0.135 m, respectivement.

# ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. N. Wang for her continuous support, valuable guidance and motivation during the past two years. I will always remember her patience and generosity showed to me.

I would like to express my gratitude and appreciation to Dr. Zuoxi Zhao for his help with C++ programming, data collection and experimental tests. The input of Eric Thibault from the Department of Electrical Engineering, McGill University, for helping in the summer of 2004 with computer programming was very much appreciated as well. The Bioresource Engineering Department had indeed provided the atmosphere and support needed. Dr. Sam Satocinal and Mr. Ray Cassidy from the mechanical workshop, many thanks to them for allowing me to use the workshop when I needed to, and for providing an area for conducting indoor experiments.

I would like to thank the Public Service Ministry (PSM) in Guyana and the International Council for Canadian Studies (ICCS, Ottawa) for their financial support in the form of a Canadian Commonwealth Scholarship. The support of the Program Officer from ICCS, Diane Cyr, was greatly appreciated! I would personally like to thank the management of the Guyana Sugar Corporation, particularly Dr. Harold Davis and Mr. Norman Friday for granting me study leave to pursue this program. The financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) for the research facilities, and the Graduate Office of McGill University for a travel grant to present my work at the American Society of Agricultural Engineers (ASAE) Annual Conference (2004), were also appreciated.

Finally, I would like to thank all my colleagues for their friendship and moral support for the past two years. Most of all, I would like to sincerely thank my family for their continued support and confidence in me!

# Contribution of authors

Dr. Ning Wang was the thesis supervisor and had direct advisory input and solutions to problems as the work progressed. Dr. N. Wang also edited the entire thesis. Chapter IV of this thesis was presented as a conference paper at the 2004 Annual Conference of the American Society of Agricultural Engineers (ASAE). The current author was responsible for concepts, designs and analysis of results. Eric Thibault assisted in computer programming and data collection.

Chapters V and VI will be submitted for publication. The current author was responsible for concepts, designs, and analysis of results. Dr. Zuoxi Zhao assisted in editing Chapters IV, V, and VI. Dr. Z. Zhao also assisted in computer programming, data collection and experimental tests.

# TABLE OF CONTENTS


**ABSTRACT**..........ii
**RESUMÉ**..........iii
**ACKNOWLEDGEMENTS**..........iv
**CONTRIBUTION OF AUTHORS**..........v
**TABLE OF CONTENTS**..........vi
**LIST OF TABLES**..........ix
**LIST OF FIGURES**..........x
**NOMENCLATURE**..........xii
**I. GENERAL INTRODUCTION**..........1
**II. GENERAL OBJECTIVES**..........4
**III. LITERATURE REVIEW**..........5
3.1 INTRODUCTION..........5
3.2 AUTOMATIC GUIDANCE IN AGRICULTURE..........6
3.2.1 Importance of automatic guidance in agriculture..........6
3.2.2 Applications of automatic guidance-Precision farming...7
3.3 COMPONENTS OF AUTOMATIC GUIDANCE SYSTEM....9
3.3.1 Guidance and steering controls..........9
3.3.2 Guidance sensors for agricultural machines..........11
3.3.2.1 Leader cables..........11
3.3.2.2 Mechanical guidance..........12
3.3.2.3 Radio navigation..........13
3.3.2.4 Ultrasonic guidance..........13
3.3.2.5 Optical guidance..........14
3.3.2.6 Machine vision..........16
3.3.3 Comparison of different guidance sensors..........19
3.3.4 Position determination in an automatic guidance system..........21
3.3.4.1 Principal of operation of GPS..........22
3.3.4.2 GPS for automatic guidance..........23
3.4 AUTONOMOUS NAVIGATION..........25
3.4.1 Overview..........25
3.4.2 Obstacle detection and avoidance..........25
3.4.3 Position determination..........28

**IV. LASER RANGE FINDER-BASED AUTONOMOUS GUIDANCE SYSTEM FOR A MOBILE ROBOT**..........30
4.1 INTRODUCTION..........30
4.2 OBJECTIVES..........33
4.3 MATERIALS AND METHODS..........34
4.3.1 Experimental setup..........34
4.3.2 Hardware components..........34
4.3.2.1 The robot platform..........35

4.3.2.2 Sensors....................................................36
4.3.2.3 Sensor's coordinate system..............................39
4.3.2.4 Robotic navigation components.........................40
4.3.3 Software design..............................................42
4.3.3.1 Software structure........................................42
4.3.3.2 The coordinate transformation method.................47
4.3.3.3 Path planning concept....................................48
4.3.3.4 Design of the navigational program in a structured maze..........................................51
4.4 EXPERIMENTAL DESIGN.........................................52
4.5 RESULTS AND DISCUSSION.....................................52
4.6 CONCLUSIONS.....................................................54
4.7 REFERENCES.......................................................55

**CONNECTING TEXT**.......................................................57

**V. POSITION DETERMINATION USING LANDMARK LOCALIZATION FOR A MOBILE ROBOT**........................58
5.1 INTRODUCTION....................................................58
5.1.1 Motion measurements........................................58
5.1.2 Localization using landmarks...............................61
5.2 OBJECTIVES.........................................................63
5.3 MATERIALS AND METHODS....................................63
5.3.1 Robot platform.................................................63
5.3.2 Internal positioning sensors.................................64
5.3.2.1 Gyroscope-Principle of operation.......................64
5.3.2.2 Wheel encoders...........................................65
5.3.3 LRF and sonar sensors.......................................67
5.3.4 Software design................................................68
5.3.5 System tests....................................................71
5.3.5.1 Test 1: Accuracy of linear distance and angular displacement measurements.............................72
5.3.5.2 Test 2: Accuracy of position determination on a grid layout.........................................73
5.3.5.3 Test 3: Gyroscope performance during maximum slip.........................................74
5.3.5.4 Test 4: Landmark localization method for position estimation in a grid layout.....................75
5.4 RESULTS AND DISCUSSION.....................................77
5.4.1 Test 1: Accuracy of linear distance and angular displacement measurements........................77
5.4.2 Test 2: Accuracy of position determination on a gird layout.............................................80
5.4.3 Test 3: Gyroscope performance during maximum slip...82
5.4.4 Test 4: Landmark localization method for position estimation in a grid layout...................................83

5.5 CONCULSIONS........................................................ 87
5.6 REFERENCES........................................................ 88

**CONNECTING TEXT**........................................................ 91

**VI. AUTONOMOUS NAVIGATION USING SONAR SENSORS AND A LASER RANGE FINDER**...........92
6.1 INTRODUCTION........................................................ 92
6.2 OBJECTIVES........................................................95
6.3 MATERIALS AND METHODS........................................................96
6.3.1 Hardware design........................................................ 96
6.3.2 Software design........................................................97
6.3.2.1 Principle of the obstacle detection and avoidance algorithm........................................................98
6.3.2.2 Definition of the rotation clearance safety............99
6.3.2.3 Definition of rotation manoeuvres......................100
6.3.2.4 Design of the autonomous navigation system......... 101
6.3.3 Experimental design........................................................108
6.3.3.1 Tests for the field of view (FOV) of the sonar sensors........................................................108
6.3.3.2 Tests for position determination capabilities of a LRF for a moving robot.......................... 109
6.3.3.3 Tests for the autonomous navigation system in three maze layouts........................................................ 110
6.4 RESULTS AND DISCUSSION........................................................111
6.4.1 Tests for the FOV of the sonar sensors......................111
6.4.2 Tests for position determination capabilities of a LRF for a moving robot........................................................112
6.4.3 Tests for the autonomous navigation system in the three maze layouts........................................................114
6.5 CONCLUSIONS........................................................ 118
6.6 REFERENCES........................................................ 119

**VII. GENERAL CONCLUSIONS**........................................................ 121

GENERAL REFERENCES........................................................123

APPENDICES........................................................ 133

Appeddix A – Functional block diagram of the ADXRS300 Gyroscope.....134
Appendix B – Test results for linear distance and angular displacement accuracy........................................................ 135
Appendix C – Results for position estimation on a grid layout...............136
Appendix D – Results for LRF-based artificial landmark localization....... 137

# LIST OF TABLES


Table 3.1: Advantages and disadvantages of various guidance sensors.................................................. 19

Table 4.1: Sonar sensor position................................................39

Table 5.1: Statistical analysis of linear displacement...................... 78

Table 5.2: Statistical analysis of angular displacement.................... 80

Table 5.3: Statistical analysis of X coordinate on a grid layout.............82

Table 5.4: Statistical analysis of Y coordinate on a grid layout...........82

Table 5.5: Statistical analysis of X coordinate using landmark localization technique............................................84

Table 5.6: Statistical analysis of Y coordinate using landmark localization technique............................................85

Table 5.7: Significant difference between the position estimation methods...........................................................85

Table 5.8: Summary of RMSE between actual and measured coordinates.......................................................... 86

# LIST OF FIGURES


Figure 3.1: Guidance and steering controller.................................. 11

Figure 3.2: Laser guidance system developed by Shmulevich, et al, 1989........................................15

Figure 3.3: Principle of Differential Global Positioning System (DGPS)..................................................22

Figure 4.1: Layout of the maze...............................................34

Figure 4.2: The P3-AT Robot..................................................35

Figure 4.3: The principle of operation of a LRF............................. 37

Figure 4.4: Sonar sensors array............................................... 38

Figure 4.5: Coordinates of the LRF and sonar sensor array................39

Figure 4.6: Robot's navigation components..................................40

Figure 4.7: Structure of a software thread.................................... 43

Figure 4.8: Architecture of software design..................................44

Figure 4.9: Architecture of software design for the autonomous navigation system...................................................46

Figure 4.10: Coordinate transformation principle.............................47

Figure 4.11: Robot's path planning system....................................49

Figure 4.12: Flowchart of obstacle avoidance.................................50

Figure 4.13: Schematic of software design.................................... 51

Figure 4.14: Path of the robot and map generated by the LRF ............. 53

Figure 5.1: Dead-reckoning concept.......................................... 66

Figure 5.2: Block diagram of the software structure........................69

Figure 5.3: Block showing the Kalman Filter in the navigation system.. 71

Figure 5.4: Angular test layout...............................................73

Figure 5.5: Test set up for the landmark localization method.............75

Figure 5.6: Landmark localization technique.................................75

Figure 5.7: Percentage error between desired and measured distances for linear displacement................................. 78

Figure 5.8: Percentage error between desired and measured angles for angular displacement....79
Figure 5.9: Comparison among robot's actual position, position from the encoder, and position from encoder with gyroscope correction (units in mm).... 80
Figure 5.10: Percentage error between measured and actual distances on grid layout.... 81
Figure 5.11: Gyroscope effect.... 82
Figure 5.12: Results for landmark localization technique....84
Figure 6.1: Regions 1 and 2 for obstacle detection and avoidance.... 98
Figure 6.2: Detector bands (L and R) for rotational clearance safety.... 99
Figure 6.3: Detector bands (L1,R1 and F)....100
Figure 6.4: Schematic diagram of the autonomous navigation software..103
Figure 6.5: Flowchart of the autonomous navigation system....107
Figure 6.6: Measurement of sonar sensor sensitivity.... 108
Figure 6.7: Experimental set up for position determination of a LRF for a moving robot....109
Figure 6.8: Maze layouts....110
Figure 6.9: Sonar sensors field of view....111
Figure 6.10: Changes of X and Y coordinates when the robot moved randomly in the test layout....112
Figure 6.11: Heading angle when the robot moved randomly in the test layout....113
Figure 6.12: Path for Maze layout (A).... 114
Figure 6.13: Path for Maze layout (B)....116
Figure 6.14: Path for Maze layout (C)....117

# NOMENCLATURE

| | |
|---|---|
| AFS | Advanced farming systems |
| AGV | Automatic guided vehicle |
| AMR | Autonomous mobile robot |
| ANOVA | Analysis of variance |
| API | Application Programming Interface |
| ARIA | ActicMedia Robotics Interface Application |
| AROS | ActivMedia Robotics Operating System |
| CCD | Closed couple device |
| d | Target distance, m |
| DGPS | Differential Global Positioning System |
| DSS | Decision support systems |
| EC | Electrical conductivity |
| EGNOS | European Geostationary Navigation Overlay Service |
| FOG | Fibre optic gyroscopes |
| FOV | Field of view |
| GIS | Geographic Information Systems |
| GLONASS | Globaluaya Navigatsionnaya Sutnikovaya Sistema |
| GPS | Global Positioning Systems |
| HIMM | Histogramic-In-Motion-Mapping |
| L | Distance between the left and right encoders, m |
| LRF | Laser range-finder |
| N | Number of measurements |
| NAVSTAR | Navigation System with Time and Ranging |
| NIR | Near infrared |
| PWM | Pulse-width-modulated |
| RMS | Root mean square |
| RMSE | Root mean square error |
| $r_a$ | Distance of Pole A from the centre of the Laser range-finder, m |

| | |
|---|---|
| $r_b$ | Distance of Pole B from the centre of the Laser range-finder, m |
| s | Horizontal distance separating Pole A and Pole B, m |
| RTK | Real-time kinematic |
| SD | Standard deviation |
| SE | Standard error |
| SIP | Server Information Packet |
| $V_l$ | Left encoder velocity, m/s |
| $V_r$ | Right encoder velocity, m/s |
| WAAS | Wide Area Augmentation System |
| 3D | Three dimensional |
| x | Vertical displacement, m |
| $x_2, y_2$ | Cartesian coordinates in the global coordinate frame |
| $x_t, y_t$ | Robot's target coordinates |
| $X_1$-$Y_1$ | Local coordinate frame |
| $X_2$-$Y_2$ | Global coordinate frame |
| y | Horizontal displacement, m |
| $Y_1'$-$X_1'$ | Intermediate coordinate frame |
| $y_1, x_1$ | Cartesian coordinate of an object in the local coordinate frame |
| $y_1', x_1'$ | Cartesian coordinate of an object in the intermediate coordinate frame |
| $\theta_1$ | Heading of an object in a local coordinate frame, rad |
| $\theta_1'$ | Heading of an object in an intermediate coordinate frame, rad |
| $\theta$ | Angle between local coordinate frame and intermediate coordinate frame, rad |
| $\dot{\theta}$ | Angular velocity in the *x-y* plane, rad/s |

| | |
|---|---|
| $\theta_b$ | Angle between line of sight of Pole B and $X_1$ axis, rad |
| $\theta_a$ | Angle between line of sight of Pole A and $X_1$ axis, rad |
| $\theta_i$ | Target angle, rad |
| $\theta_r$ | Angle between $X_1$ axis and $X_2$ axis (robot's heading), rad |

# I. GENERAL INTRODUCTION

For the last five decades, researchers have been trying various methods to automatically navigate vehicles, whether in the air, land, or sea. The methods vary from aircraft, land vehicles, agricultural machines, ocean-vessels, and indoor and outdoor mobile robots. Many sensors for automatic navigation have been developed and tested extensively, and yielded positive results. These sensors are important for position estimation and orientation of the vehicle relative to some reference coordinate system. The sensors include mechanical guidance sensors, sonar sensors, laser scanners, radio beacons, closed couple device (CCD) cameras, gyroscopes, and wheel encoders. In the last ten years, Global Positioning Systems (GPS) have become very popular due to its accuracy and declining costs.

One important aspect of any navigation system is the requirement for an accurate position estimation, whether it is the position of an object relative to the vehicle (in a local coordinate frame) or that in a global coordinate frame. It is also essential that the vehicle be able to identify its position at any time instant either in global coordinate frame or relative to its local environment. Several approaches have been developed to solve the positioning challenge, including map-based navigation (*a priori*, knowledge of the environment), trajectory planning, landmark-based methods and dead-reckoning. Sensors used for the navigation system and local environment in which the vehicle operates are two of the fundamental factors influencing the accuracy and success of a navigation system. In the event that one sensor is not sufficient to achieve the accuracy needed, more than one sensor can be incorporated into the system. This has led to the fusion of sensory readings using various sensing techniques, of which the Kalman filter approach is very popular.

Dead-reckoning is a very common technique used for position estimation, especially on mobile robot applications, because of its simplicity and low costs. However, this method has its limitation, as the position uncertainty increases with time during operation. Wheel slippage and effective rolling radius are two of the main factors influencing the accuracy of this method. Hence, odometric measurements alone are not adequate to deal with positioning problem. In many instances, odometry and gyroscope readings are fused together to better predict the position of a mobile robot.

Other researchers prefer to use laser scanners and sonar sensors to achieve an estimate of position, as these methods are not dependent on wheel slip or the ground condition such as undulation. Noguchi et al. (2002) developed an automatic guidance system for an agricultural robot tractor using the fusion of Real-time kinematic GPS (RTK-GPS) and Fibre optic gyroscopes (FOG). The guidance system was capable of navigating the robot in either straight or curved paths at speeds up to 7.2 km/h. The system resulted in a root mean square (RMS) travel error of less than 20 mm.

As mentioned earlier, a variety of methods have been developed for navigational systems whereby features in the environment play a critical role. These features can either be naturally occurring such as edges, corners, plant rows, drains, and furrows, or artificial landmarks placed at known locations in the local environment. The selection of sensor(s) and the design of an appropriate algorithm that encompasses all the acquired data are two of the critical tasks influencing the development of an automated system (driver-assisted) or a completely autonomous system (driverless).

Agricultural practices have been revolutionized with the introduction of computer and electronic components. Automatic navigation of agricultural machines, including tractors and combine harvesters, have seen similar technological advancements. GPS and machine vision techniques find many applications in agricultural production. Commercial companies, including John Deere (USA), AGCO (USA), Case IH (USA), and CLAAS (Germany), are only some of the few manufacturers that provide automatic guidance systems (hardware and software) for their machines. Studies by Kaminaka et al. (1981) showed that steering accuracy decreased significantly as extra demands are placed on the operators. While there have been advancements in cab designs to improve operator comfort and conveniences, increasing field speeds, plus increasing power levels; increased implement widths have not made the task of accurate guidance any easier for the operator. These are the main reasons which drive the development of automatic guidance systems in agriculture.

# II. GENERAL OBJECTIVES

This study has the general objective of developing an autonomous navigation system for a mobile robot. The study also investigates the use of a laser range-finder (LRF) and an array of sonar sensors as the main navigational and obstacle detection sensors. The specific objectives are:

1) To develop navigation algorithms utilizing data from a laser range-finder (LRF) and a ring of eight sonar sensors
2) To enhance the navigational algorithms by incorporating obstacle detection and avoidance using data from the sonar sensors
3) To develop a LRF-based artificial landmark localization technique for position and orientation determination, and to compare the accuracy of the landmark localization technique with odometric and inertial measurements from two wheel encoders and a gyroscope, respectively.
4) To develop an advanced autonomous navigation system for obstacle detection and avoidance using defined virtual regions and detector bands in front of the robot
5) To test the advanced autonomous navigation system in three indoor mazes.

# III. LITERATURE REVIEW

## 3.1 Introduction

The automatic navigation of a wide range of vehicles has been at the heart of many commercial and research institutions. Perhaps, the most common ones are automatic pilots for aircraft and ocean vessels. Efforts have been made to automatically guide and land aircrafts using satellite navigation such as Global Positioning Systems (GPS) (Schanzer, 1992). Off-road and road vehicles have seen similar technological advancements. Lutzeler and Dickmanns (1998) reported that at the Bundeswhr University of Neubiberg (Germany), an autonomous road vehicle was developed that was capable of traveling at speeds of up to 130 km/h with the ability to change lanes during overtaking manoeuvres. Indoor and outdoor mobile robots used in various industries and the military have also been designed to operate autonomously (Schmidt and Freyberger, 1996). Autonomous vehicles tend to be employed in situations that are dangerous and hazardous to humans, so as to minimize and more so eliminate the exposure of humans to toxic and lethal substances. Autonomous systems have also been introduced into the mining industry; a 30 tonne Load-Haul-Dump truck, equipped with a reactive navigation control system, successfully achieved full speed autonomous operations in an operational underground mine (Roberts et al., 2002). Recently, intelligent service robots are becoming popular to assist people in their daily chores. This includes robot applications for hospital services, museums, office buildings and shops. Autonomous systems have also been used in planetary rovers and battlefield surveillance vehicles. As the costs for computer and electronic systems decline, machine vision and navigation systems for autonomous vehicles in agriculture and other industry are coming closer to commercialization on a larger scale.

Vehicle automation can be considered in two categories, automated systems (driver-assisted) and autonomous systems (driverless). In automated systems, an operator is responsible for monitoring the vehicle performance, and assists to operate the vehicle in difficult tasks, even driving the vehicle to the field or work site. This tends to reduce the demand and stress on the operator, improve efficiency and safety, and allow the operator to function at a higher level of performance over an extended period of time (Gerish et al., 1997). Automated systems are cheaper than autonomous systems and possess the capability to switch from an automatic mode to a manual mode. Automatic mode can also be easily integrated to an existing fleet of machines.

## 3.2 Automatic guidance in agriculture

### 3.2.1 Importance of automatic guidance in agriculture

Automatic navigation of vehicles in agriculture is no exception; agriculture today is under increasing pressure to feed a growing population with a diminishing work force. In the past, animal power and machines have been introduced unto the farm, which resulted in an increase of land that a farmer can cope with. Fitzpatrick et al. (1997) claimed that operators of agricultural machines are still one of the main limiting factors affecting agricultural production. The operators are under immense pressure to maintain steering accuracy in order to achieve a high quality of work. On the contrary, steering accuracy tends to decrease as increasing demands are placed on the operators (Van Zuydam, 1999 and Kaminaka et al., 1981). Long hours of work and repetitive tasks result in operator's fatigue, which in turn affect safety and decrease operation efficiency. Automatically guided machines will minimize stress on operators, reduce operator's work intensity, increase operation safety, and enhance efficiency. The aim, therefore, of automatic guidance is to steer the vehicle along a desired path automatically, plus be able to detect vehicle posture, and create the appropriate steering signals. Vehicle posture as defined by Kanayama and Hartman (1989) is the position and orientation of a vehicle relative to a reference frame. Another advantage of automatic guidance in agriculture is the potential for the improvement in placement of seed, fertilizers and pesticides. Nieminen and Sampo (1993) claimed that human operators using large equipment tend to overlap previous paths during certain operation, thereby applying double inputs (chemical and fertilizer) to specific sections of a field. An automatic guidance system can precisely control large equipment to minimize the amount of treatment overlap, thus reduce the farmer's expenditure.

### 3.2.2 Applications of automatic guidance - Precision farming

Agricultural practices have been revolutionized during the last ten years due to the applications of computers and electronics. The term "Precision farming" was born and defined as the pursuit of increased efficiency in the management of agriculture. It incorporates a vast amount of technologies, computing, electronics, and supports the use of vehicle positioning systems, geographic information systems (GIS), decision support systems, remote sensing and telecommunications (Blackmore, 1994 and Gibbons, 2000). However, the research on automatic guidance system, primarily, focuses on two particular aspects of precision farming, and that is the vehicle positioning and navigation components. Agricultural production has benefited tremendously from the industrial and technological eras. The industrial era brought mechanization and synthetic fertilizers, whilst the technological era brought genetic engineering and automation. The information age offered the application of technological advances to precision farming (Whelan et al., 1997).

Precision farming has facilitated the acquisition of comprehensive data on production variability in both space and time. Zhang et al. (2002) outlined an overview of precision farming in which six groups of variability that affects agricultural production were listed, namely yield variability, field variability, soil variability, crop variability, management variability and anomalous variability. A number of grain yield sensors are available commercially and classified into four groups - weight-based sensors, optical yield sensors, $\gamma$-ray sensors, and impact sensors (mass flow sensors). Similarly, soil sensors vary from near-infrared (NIR) sensors for measurement of soil organic matter and moisture contents (Hummel et al., 2001) to electromagnetic induction to measure electrical conductivity (EC) (Myers et al., 2000). When a tractor or a combine harvester, equipped with a GPS, and the soil or yield sensors, traverses the field, soil maps or yield maps can be produced. This amalgamation of yield and/or soil sensors with vehicle positioning has opened new doors and opportunities for farm managers by providing spatial variability information of yield and soil properties within a field. Traditionally, only the total yield of the field was known, but now it is possible to have more detailed information of soil and yield simultaneously in a fast manner. GIS provides a meaningful way to deal with this information as it is simply an effective way of computerizing a set of map overlays to investigate the interaction between them.

Once the yield variability is known, the farm managers will then want to treat the fields either in terms of seeds, fertilizers, spray applications or other field treatments such as cultivation. A decision support system (DSS) can be formulated using GIS and a set of economic and agronomic software models and provides the farm managers with specific information needed to make a farming decision. Treatment maps can be downloaded to a tractor so that variable treatments can be applied to the field. The glamour of a precision farming culture as mentioned by Blackmore (1994) is to apply only *what is needed, when it is needed* so that it gets used with maximum effect and minimum waste.

Some agricultural and industrial companies have developed guidance systems to suit their products. FIELDSTAR is a system developed by AGCO Corporation, Duluth, USA. It provides data logging, positioning and guidance, implement monitoring and control, as well as software to analyze data, draw maps, and create treatment maps. It also consists of other elements such as satellite navigation systems and computer-based Geographic Information Systems. FIELDSTAR uses GPS technology to precisely locate a vehicle or machine in a field. This valuable information has many management benefits, including allowing the users to monitor operations, create yield maps, vary inputs according to soil potentials and to conduct precision farming. This system allows the operators to steer the machine in parallel path at exact distances apart. The 'parallel swathing' not only provides accuracy but also boosts outputs and improves efficiency by minimizing over and under-lapping.

John Deere, Illinois, USA, also developed a precision management solution, GreenStar. GreenStar consists of three main components, GreenStar display, mobile processor, and StarFire iTC receiver. The GreenStar display is similar to a computer monitor with menu- driven commands that allow operators to program information quickly. The display area allows viewable operational data while on the go and can be mounted in a tractor, combine or sprayer depending on the operation at hand. The mobile processor is the brain of the system and logs information such as farm, field, yield, crop and positioning information to a PCMCIA data storage card. The StarFire iTC receiver is a dual frequency GPS technology with terrain compensation functionality. It is a very accurate system using signals from satellites and the John Deere differential correction network. GreenStar system also includes AutoTrac, Parallel Tracking, Field Doc™, and GreenStar Combine Yield Systems.

Case IH (Wisconsin, USA) and CLAAS (Harsewinkel, Germany) have also developed precision farming systems. Case IH have developed AFS (Advanced Farming Systems) AccuGuide™ Auto-guidance systems, together with AFS software. CLAAS have developed the CLAAS Autopilot , Laserpilot and GPS Pilot (Outback S).

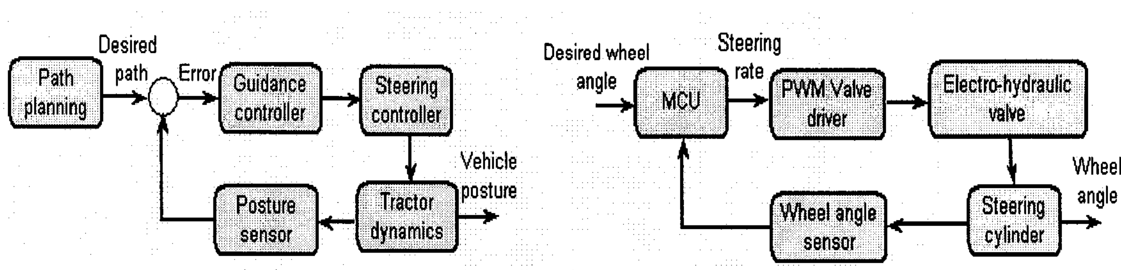
## 3.3 Components of automatic guidance system

Generally automatic vehicle control can be subdivided into three parts, position sensors, vehicle controllers, and actuators. The position or guidance sensors supply the system with the position deviation of a vehicle or implement from a desired path. The vehicle controller or guidance controller as it is often referred to, determines the appropriate steering commands once the vehicle's posture relative to a desired posture is known. The actuator or steering controller is responsible for executing the commands of the guidance controller and often comprises of hydraulic and/or electronic components.

### 3.3.1 Guidance and steering control

The automatic guidance of agricultural vehicles requires a steering controller to implement the steering signals generated by the guidance sensors. This presents a number of challenges to engineers as vehicle dynamics and electro hydraulic valve properties affect the performance of steering control (Laine, 1994). Some of these challenges include the working environment such as surface (ground) properties, weather, field conditions and equipment status. This implies that steering controllers should be able to cope with changes in ground conditions, variation in equipment operation states, traveling speed, tire cornering stiffness, and other parameters affecting steering dynamics. Researchers have developed kinematic models of the machines that they worked on. Choi et al. (1990) developed discrete-time equations to describe the motion of a tractor, and O'Connor et al. (1996) developed linearized equations of motion to represent steering. Both models were based on only geometric properties of vehicles and with the assumption that there was no sideslip of the wheels. Mass and inertial factors were also ignored. The limitation of the model was that vehicles were restricted to slow speeds only, because as the speed increased, the tire sideslip angle during turning manoeuvres also increased. These limitations made the kinematic models inadequate.

A feed-back controller based on the guidance dynamic of a tractor was developed by Stombaugh et al. (1999). The objective was to design a GPS-based automatic guidance system for a two-wheel drive test tractor (Case-IH) with a capability of driving at speeds between 16.2 km/h and 24.48 km/h (maximum practical operating speed), while maintaining an accuracy of ±0.3 m in a desired straight path. A Novatel RT-20 Kinematic Differential GPS posture sensor and a linear potentiometer wheel angle sensor mounted on a steering cylinder were used on the test tractor. The flow charts of the guidance system and steering controller are shown in Figure 3.1(a) and Figure 3.1 (b), respectively. The electro-hydraulic valve, when activated, overrode the manual steering but caused no interference when not activated. A pulse-width-modulated (PWM) signal was used to control the electro-hydraulic valve (Figure 3.1 (b)). One of their conclusions revealed that a classical model-based controller could provide automatic guidance for a two-wheel-drive tractor to within ±16 cm of the desired straight path at speeds up to 6.8 m/s.




(a) Guidance system (b) Steering controller

**Figure 3.1: Guidance and steering controller (Stombaugh et al., 1999)**

## 3.3.2 Guidance sensors for agricultural machines

A thorough review of guidance methods for agricultural machines was presented by Tillet (1991), who classified the guidance methods into six main categories, namely leader cables, mechanical guidance, optical guidance, radio navigation, ultrasonic guidance, and dead-reckoning. Dead-reckoning will be covered in Chapter V.

### 3.3.2.1 Leader cables

The use of leader cables dates back as far as 1924 (Tillett, 1991). This type of guidance is based on the detection of a magnetic field produced by small low-frequency signals, typically, 150 mA at 2 kHz. It is not widely used in agriculture, but very popular in automatic guided vehicles (AGVs) operating in factories and warehouses. Since the magnetic field is not significantly affected by soil, a cable carrying the guidance signal is buried a couple of centimeters below the concrete floor in factories or about 0.5 m under agricultural fields. The sensors mounted on the vehicle consist of a large number of turns of copper wires wound on a ferrite core. Vertical sensing coils and a balanced pair of horizontal coils were developed by Finn-Kelcey and Owen (1967) and Telle and Perdok (1979) respectively. A change in the magnetic field generated by the lead cable induced a signal in the sensing coil. The magnitude of the signal depended on the strength of the magnetic field. The vertical sensing coil consisted of a wound coil with a vertical axis that picked up the vertical components of the flux whilst a current was induced in its windings at the same frequency as the signal generated. The induced current and the flux in the coil varied as the direction of flow of current changed in the cable. On the other hand, the horizontal sensing coils consisted of a balanced pair of horizontal coils straddling the leader cable. They were placed in symmetry with respect to the cable, so that the flux at each coil was the same. When the coils are not symmetrical the flux is different in each coil. A heading signal was derived by mounting a pair of repeat coils in line with the first coils. The repeat coils can be ahead or behind the first pair of coils. Basically, the heading varied with the difference between the front and rear signals and the displacement errors varied with the sum of the two signals.

### 3.3.2.2 Mechanical guidance

Mechanical guidance systems were based primarily on two types of features - existing features and specially provided features. Existing features used included field drains, crop rows, and furrows. A feeler spring pressed against crop stems was used to activate an electro-hydraulic valve, which controlled a steering ram, achieving a steering accuracy of ±50 mm in normal conditions (Suggs et al., 1972). A furrow guidance system was developed by Hilton and Chesney (1973), and a tractor was guided within ±40 mm of a desired path. The method consisted of a furrow following arm pressed firmly against the furrow walls by means of springs. The furrow arm was connected through a series of linkages to the tractor control valves. Slopes, uneven gradient, and changes in soil or surface conditions were some of the factors that affected the reliability and accuracy of the system. Some researchers have also tried to incorporate special features in the field to assist with this type of guidance, varying from specially designed furrow and buried cables to steel rails. Widden and Blair (1972) proposed a system whereby a buried cord below the surface acted as a guidance marker. The tractor uprooted the cord through a sensing mechanism connected to a servo hydraulic valve for steering. An accuracy of ±50 mm was reported using this method.

A commercial company, KTBL (Kuratorium fur Technik und Bauwesen in der Landwirtschaft) in Dethlingen, Germany, manufactures side-guiding systems for implements working in potato fields. The use of this type of system was only limited to ridge cultivations and was not suitable for non-contact guidance systems. Two of the main problems with this system were that the corrections of the driver error were limited to about 10 cm and automatic steering on sloping ground could only be accomplished on slopes less than 6% (Keicher and Seufert, 2000)

#### 3.3.2.3 Radio navigation

Radio navigation is a very appealing method for navigational purposes, as it can cover a wide area and requires only a few beacons located at convenient spots. A radio navigation system, AG-NAV (AG-NAV, Texas, U.S.A) was designed to locate a machine in a field during spraying and fertilizer operations. The accuracy was reported to be ±0.23 m at 0.8 km range (Searcy et al., 1990). Other researchers have also used microwave reflections from passive beacons located on headlands to calculate vehicle positions by triangulation (Bonicelli and Monod, 1987). The problem with microwave systems is that they are limited to line of sight, and experience has proven that they are insufficiently accurate for automatic guidance. Navigation using earth satellites for position estimation are becoming more popular and this concept is discussed in Section 3.3.3.

#### 3.3.2.4 Ultrasonic guidance

Ultrasonic devices for guidance of machines have developed at a rapid rate due to its simplicity and low costs. Indoor mapping and navigation using ultrasonic sensors have been used extensively in mobile robot applications. Detail description of this technology is in Chapter VI. Ultrasonic sensing technology was applied differently in the past to assist in machine guidance. For example, it was used to sense a ploughed furrow, but was later abandoned due to inadequate reflection from soil (Warner and Harries, 1972). A harvesting vehicle equipped with ultrasonic device was able to detect apple tree trunks, thus assisting in machine guidance (McMahon et al., 1982), and Patterson et al. (1985) used two ultrasonic sensors straddling a row of transplants to aid a trans-planter.

In recent years, sonar sensors have been used extensively for mobile robots, which include landmark localization, obstacle detection and avoidance. Araujo and Grupen (1998) used feature localization models to identify line, corner and edge features for navigation tasks. The main objective of the work was to detect common features in indoor environments and to use those features as landmarks to recover the robots' pose when necessary. The authors successfully developed an adequate sonar-based model and configuration based on accurate feature information. Several experiments using sonar sensors to acquire 2D information have been conducted in the past, however, Akbarally and Kleeman (1995) developed a novel sonar sensor consisting of three transmitters and three receivers that could localize and classify 3D targets into 16 different naturally occurring indoor classes. The sensor was capable of localizing targets in 3D to sub-millimeter accuracy and sub-degree bearing (0.2° angle) accuracies within a range of 6 m.

### 3.3.2.5 Optical guidance

Optical guidance sensors include laser scanners, infrared sensors, visible light sensors, and image sensors. The use of laser scanners for localization and guidance is discussed in greater detail in Chapter VI.

Historically, several researchers attempted to use laser technology for machine guidance, especially in drainage machines. By using a single horizontal plane, the drainage machine was able to position a cutting tool at an accurate depth relative to any position within the line of sight of the laser source.

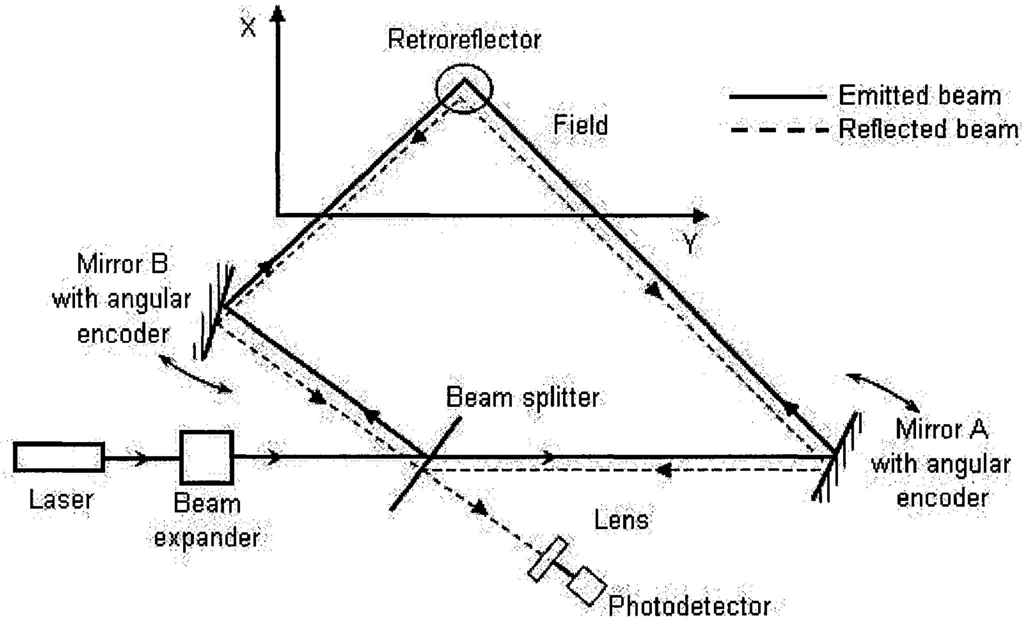



**Figure 3.2: Laser guidance system developed by Shmulevich et al., 1989**

A laser scanning method for guiding field machinery was developed by Shmulevich et al. (1989). The author used a single continuous laser, which was split into two beams and directed by two rotating mirrors as shown in Figure 3.2. A test vehicle was equipped with a retro-reflector and placed in the field. By detecting the reception of the laser beam from the two rotating mirrors the vehicle position was computed using the angles of the mirrors and the distance apart (of the mirrors) at the time the laser beam was detected.

An interesting approach was adopted by Kawamura and Namikawa (1984) using infrared sensors for machine guidance. A tractor equipped with a rotary cultivator was guided to within ±50 mm of the desired path by utilizing infrared reflectance characteristics of cultivated and uncultivated soils. Two infrared sensors were mounted ahead of the front wheels, one on either side of the boundary between cultivated and uncultivated soil. The tractor was considered on course when the detectors were on either side of the boundary and off course when both detectors were on the same side. The use of visible light is not a common approach for machine guidance as ambient light can cause interference. However, some factory use this method on indoor AGVs, which follow a retro-reflective tape stuck to the floor. It is a cheap system and routes can be changed quickly and easily. One limitation of this method is that the tape must be clean and unbroken, which eliminates its use in field operations.

### 3.3.2.6 Machine vision

Machine vision technology has been developed in agricultural sectors for various applications, including post-harvest technology, classification, sorting, precision farming, land-based remote sensing, and aerial-based remote sensing (Chen et al., 2002). This technique utilizes imaging cameras ranging from monochrome cameras that perform simple shape and size recognition to multispectral and hyperspectral imaging systems to identify materials and detect subtle and/or minor features in an object. Multispectral imaging comprises of a set of images, each being acquired at a narrow band of wavelengths. Hyperspectral imaging, on the other hand, has emerged as a powerful tool in earth remote sensing and medical diagnosis. This technique is a combination of imaging and spectroscopy to obtain both spatial and spectral information from an object, and can be used in precision farming applications, including detection of plant stress and crop infestation, agricultural product quality and safety sensing. Machine vision technology is also progressively being used in automatic guidance systems in autonomous vehicles.

In imaging systems, a camera receives light reflected from the surface of an object and then converts the light into electrical signals using charge-coupled device (CCD). Machine vision with respect to automatic guidance has allowed cultivator blades to be set much closer to plants with the advantage of greatly increasing the efficiency of weed control and hence saving time and money that may have been needed for additional spraying. Some researchers have used colored CCD cameras for automatic navigation (Buluswar and Draper, 1998, Crisman and Thorpe, 1990) whilst others continue to use monochrome cameras in land vehicle navigation (Dickmanns and Mysliwetz, 1992 and Matthies et al., 1995). Vision techniques employed in agriculture in the past have implemented algorithms that maintained the vehicle position using crops rows as a navigational aid. This approach was capable of achieving an accuracy of ±15 mm at a forward speed of 1.8 km/h using a horticultural robot (Hague and Tillett, 1996). Slaughter et al. (1999) demonstrated that it was possible to use an off-the-shelf machine vision hardware to develop a real-time guidance system for row crop cultivator. A John Deere (Model 7800) tractor with two CCD Sony (Model SSC-C370) cameras mounted directly above the centre of a pair of cultivation discs were used for a set of field experiments. A machine vision algorithm was developed to navigate the tractor based on color segmentation using a binary and stochastic pattern to differentiate crop plants from randomly located weeds. The system was able to achieve a RMS guidance error of 7 mm under low weed loads and 12 mm under high weed loads operating at travel speeds of 16 km/h.

Many imaging systems were designed using simple concepts such as the use of tracking windows within the image, the use of non-linear algorithm, and the use of row-fitting algorithms, which can withstand a noisy image. Reid and Searcy (1987) used run-length encoding on a threshold image as a means of identifying the crop row edges for the automatic guidance of an agricultural tractor, obtaining offset errors of up to 10 cm. This method provided a significant reduction in the amount of image data for analysis, which were quantized into two sets, 'light' and 'dark'. The coordinates where the data changed from light to dark or vice versa on each horizontal line were observed, indicating a transition point on the outline of the rows. A different approach was adopted by Billingsley and Schoenfisch (1997). They used linear regression in three crop row segments with a 'viewport' as a datum. A straightforward averaging technique was then implemented to determine the displacement and slope of the row from the centre of the viewport. The algorithm achieved an accuracy of ±20 mm.

An agricultural combine harvester equipped with a single monochrome camera of resolution 752 pixels (horizontal) by 582 pixels (vertical) was capable of detecting the cut and uncut edges in a maize field. The lateral positioning signal of the harvester from the crop was used as a guidance signal for the machine (Benson et al., 2003). The camera in this case was mounted on the header of the harvester. The authors mentioned that the location of the camera tended to influence the algorithm. By mounting the camera over the transition point, i.e. over the cut and uncut edges, the distortion was minimized and signal interpretation became easier. The vision system was used to determine the relative heading and orientation of the crop relative to the header. The crop heading angle and offset were then directly mapped to the orientation and distance between the crop and harvester. Finally, the offset in pixels was converted to distance units. There were, however, some large errors associated with this approach as reported by the authors. Firstly, due to the camera offset from the centre of the machine, any slight trouble and inaccuracies in the steering angle were magnified. Secondly, poor image quality produced inaccurate cut-edge parameterization and consequently induced errors in the guidance signals. This in turn resulted in errors in the desired steering angle. Thirdly, limited contrast in the image tended to inhibit the cut-edge tracking, as cut and uncut portions of the field contained plants of the same moisture and growth stage. This minimized the spectral differences between the two portions, as visible and near infrared optical filters did not increase the segmentation between the two regions of cut and uncut sections.

Chen et al. (2003) implemented a machine-vision-based guidance system on a six-row automatic rice transplanter using a Sony DCR-PC10 digital video camera. Their findings revealed that the system was capable of analyzing images of the field shoreline (concrete or soil banks) and/or rows of seedlings, coupled with the functional capability to detect the seedling rows and the end of the field. The detection accuracy of the system was reported to be 99.2%, 98.6% and 98.9% for concrete bank, soil bank and rice seedlings, respectively. Mas et al. (2002) used Blob analysis and Hough Transform to identify the crop rows in a soybean field in order to mark out the centre of the path that the tractor must follow for its automatic guidance. A CCD camera mounted on an agricultural tractor was used for the experiments. Results showed that Hough Transform could effectively overcome noise problems derived from real crop images, and Blob analysis and merging algorithms provided adequate filtering to identify crop rows.

### 3.3.3 Comparison of different guidance sensors

The table below compares the different guidance sensors described in Section 3.3.2.

**Table 3.1: Advantages and disadvantages of various guidance sensors**

| Guidance sensors | Advantages | Disadvantages |
|---|---|---|
| Leader cables | 1) Simple<br>2) Low cost<br>3) Magnetic field is not greatly affected by soil, so the cable can be buried | 1) System is not compact<br>2) It is an intrusive system that requires implementation of leader cable in the ground<br>3) Leader cables are subject to vandalism by humans and animals<br>4) Limited use in field operations |
| Mechanical guidance | 1) Low cost<br>2) Existing features such as drains, furrows and crop rows can be used as guidance aid<br>3) Gantry systems (rails) have a high degree of accuracy combined with low rolling resistance | 1) Contact system<br>2) Accuracy affected by irregular features, such as soil or surface conditions<br>3) Specially designed features can increase capital cost<br>4) Subject to mechanical failure<br>5) Performance is subjected to weather and environmental conditions |
| Radio navigation | 1) Non-contact guidance system<br>2) Can cover a wide area with few beacons<br>3) GPS systems are becoming cheaper and much more accurate | 1) Radio signals affected by trees and tall buildings<br>2) Microwave systems are limited to line of sight and insufficiently accurate<br>3) Accuracy of GPS is affected by clock errors, multipath effects, ionospheric and tropospheric conditions, inaccuracies in the receivers and |

| | | |
|---|---|---|
| | | orbital positions |
| Ultrasonic guidance | 1)Non-intrusive method<br>2) Low cost and simple<br>3) Simple interface<br>4) Easy to implement and easy to interpret sonar data<br>5) Typically accurate readings | 1) Response time is limited to the velocity of sound in air<br>2) Subject to noise interference, beam spreading and scattering<br>3) Multiple sonar signals may cross-talk creating inconsistency between some readings<br>4) Multi target reflections are difficult to model |
| Optical guidance | 1) Non-contact system<br>2) Reliable method and easy to implement<br>3) High resolution and accuracy<br>4) Versatile | 1) Target surface properties affect reflection of laser beam<br>2) Incidence angle can give rise to errors<br>3) Loss of synchronization at high data transfer rate<br>4) 3D laser scanners are expensive and have slow mapping capability in real time |
| Machine vision | 1) Can be used for several purposes<br>2) Non-contact guidance system<br>3) Low power consumption<br>4) Random pixel access allows for fast read out of small area of interest<br>5) Detection accuracy can be as high as 99%<br>6) Large amount of information is collected quickly<br>7) Potential exists for this method to be both cheap and powerful | 1) Requires additional software for image analysis<br>2) Requires landmarks – either existing or specially designed<br>3) Subject to poor image quality<br>4) Operating under natural lighting conditions<br>5) Storing and processing data |

### 3.3.4 Position determination in an automatic guidance system

GPS was developed by the American military initially for accurate positioning of military personnel. The system first became available for public uses in 1995. Navigations by satellites have resulted in two systems, the NAVSTAR GPS (Navigation System with Time and Ranging – Global Positioning System) maintained by the US Department of Defence and the US Department of Transportation, and GLONASS (Globaluaya Navigatsionnaya Sutnikovaya Sistema – Global Navigation Satellite System), which is a Russian Global Navigation Satellite System. GPS consists of a constellation of 24 satellites positioned at an altitude of about 21 726 km, which circle the earth at intervals of 12 hrs, providing complete coverage of the earth's surface. These satellites transmit very accurate timing information back to base stations on earth. The receivers of the GPS pick up the signals from several satellites that are available within range. The more numerous the satellites, the more robust the position information obtained from them. According to field tests, trees and tall buildings tend to interfere with the satellite signals. Estimates of GPS accuracy indicates that horizontal errors is within about 22 m 95% of the time, and vertical errors is within about 33 m 95% of the time (Shaw et al., 2000). However, observed accuracies suggest that performance is better than that, and is perhaps 10 m or less in some cases.

There are basically two modes that GPS can be used: single mode and differential mode. The single mode uses one receiver, which collects the timing information and processes it into position. This method has some inherent positional errors. It is, however, the cheapest and easiest. The differential mode uses two receivers, one mounted on the vehicle and the other in a fixed position. Figure 3.3 shows a differential mode of GPS. The fixed receiver appears to move because of the introduction of the randomized positional error. The mobile receiver on the vehicle then picks up this error and it is deducted from the incoming signal, thereby reducing the overall positional errors.

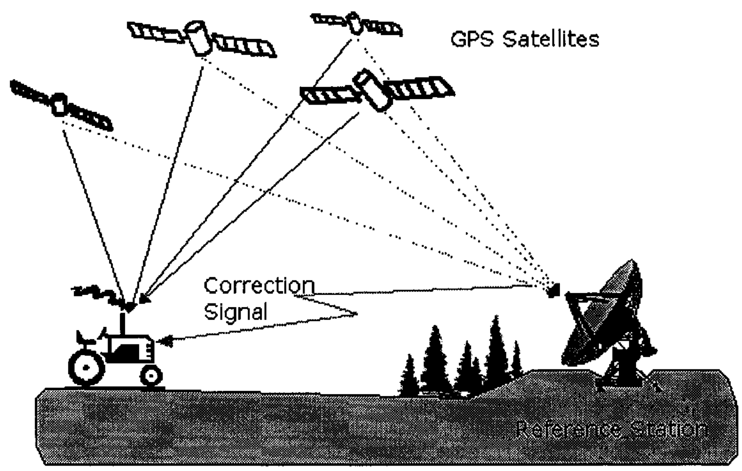



**Figure 3.3: Principle of Differential Global Positioning System (DGPS)**

### 3.3.4.1 Principle of operation of GPS

Principally, four satellites must be available to compute a 3D position. The term "trilateration" is used to define this principle and the position of the receiver is calculated from any point on the earth's surface to the satellites in view. Radio signals are broadcasted continuously by the satellites at two carrier frequencies within the L-band region of the microwave spectrum. The distance of the receivers from the satellites are related to the time that the radio signals travel from the satellites to the receivers. There are two ways in which location can be estimated using GPS, *pseudoranging* and *carrier-phase management*. *Pseudoranging* is based on time differences to estimate distances. The clocks that record the time ought to be therefore accurate, and for this reason atomic clocks are installed in satellites and advanced quartz clocks in the receivers. The GPS receivers simultaneously generate codes that match the codes produced by the satellites and as the GPS receivers receive the coded signals from satellites, they estimate the temporal displacement required to synchronize the two codes. This temporal displacement is used for an estimate of distances between the receivers and each of the satellites within range. Pseudoranging technique, however, is less accurate than other methods. Because of its simplicity, convenience and low cost, it is widely used for determining positions of fixed points. *Carrier-phase* *management,* on the other hand, is based on a detailed examination of the signals broadcasted by the satellites. GPS receivers detect either or both of the signals in the L-band and then add them to a signal generated by the receiver. This method is an application of the Doppler principle, whereby observed frequency shifts are used to derive positional information. Real-time kinematic GPS (RTK-GPS) is an advanced form of *carrier-phase management* in which GPS signal corrections are transmitted in real-time from a reference receiver at a known location to one or more remote receivers.

In summary, a few factors that influence the accuracy of GPS are:

- Clock errors
- Ionospheric and tropospheric conditions
- Multipath effects
- Inaccuracies (noise) in the receivers
- Orbital position (ephermis)

In addition to the five possible cause of errors mentioned above, designers of GPS systems intentionally added errors to create *selective availability* (SA). SA allows for full precision of the GPS system for authorized users and at the same time allows civil users access to the degraded signal. SA is as a result of deliberate errors introduced in the ephermis and the clock to increase positional errors.

### 3.3.4.2 GPS for automatic guidance

Research at Standford University led to the development of ultra precise GPS positioning system, known as Carrier Phase Differential GPS (CPDGPS). Graduate students at the university replaced the inertial guidance system of a United Airlines 737 with a CPDGPS to provide position and latitude information, and successfully land the aircraft automatically 110 times (Pervan and Parkinson, 1997). Subsequently, a research project was initiated for the development of a tractor guidance system using the CPDGPS because of its low cost, high accuracy and absence of drift/bias. O'Connor et al. (1996) developed a linear vehicle model to control a John Deere Model 7800 tractor using CPDGPS along four 50 m rows. The control method was considered accurate if the mean tracking error was less than 5 cm and the Standard Deviation (S.D) of the GPS measured tracking error of a control point from the desired trajectory was less than 10 cm. The initial results using the tractor without any implement attached achieved a mean tracking error of less than 1 cm and a S.D of approximately 2 cm. In order to find out the effects of hitched implements on the tractor, a three-shank subsoiler was hitched to the tractor and repeatedly tested in the four rows. The mean tracking error was 0.4 cm and the S.D was 4.0 cm. The soil engaging implement introduced additional tracking disturbances through the subsoiler, but at the same time minimized the tractor's lateral displacement from ground disturbances. Further work by this group explored navigational accuracy along non-linear trajectories and sloping terrain.

Stoll and Kutzbach (2000) described a guidance system using a RTK-GPS as the sole positioning sensor on a self-propelled CLAAS forage harvester. RTK-GPS offers a positional accuracy of 10 mm to 50 mm, which is considered sufficient for several agricultural operations. The performance of the system was investigated under different path shapes, ground conditions, and speeds. A SD of less than 100 mm was obtained for all the conditions tested. Similarly, Cordesses et al. (2000) also used RTK-GPS as the only guidance sensor on a combine harvester and were able to achieve 50 mm accuracy with speed ranging from 4 km/h to 10 km/h. In this approach a kinematic model of the harvester was developed and a non- linear law was incorporated for path planning capability.

Nagasaka et al. (2004) mentioned that there was a trend in Japan for increasing large paddy fields, which led them to develop an automatic six-row rice transplanter. Their system, however, used RTK-GPS for precise positioning in conjunction with fibre-optic gyroscope (FOG) sensors to measure direction. Actuators were used to control steering, engine throttle, clutch and brake. They used the FOG sensor for orientation, which was a different approach compared to the systems described earlier, where RTK-GPS was the sole sensor. An absolute rotary encoder was used to sense the steering angle, proximity sensors detected the clutch and brake positions, and electrical linear cylinders controlled the clutch and brake pedals. The maximum RMS deviation from the desired straight path was less than 120 mm at travel speeds of 2.52 km/h.

## 3.4 Autonomous navigation

### 3.4.1 Overview

Autonomous mobile robot (AMR) as defined by Hoppen et al. (1990) is a system which perceives information about its environment in order to use this information for solving a given task. The system must consist of an onboard computer to be able to conduct all computations independently, without any external intervention, and must be able to reach a target position in a known or unknown environment. There are essentially three requirements for AMR systems, *environment perception, sensor data processing, and position estimation. Environment perception* refers to the models used by the robots to represent its environment, of which two basic models are commonly used, namely a Cartesian map-based model or sensor-based model. A Cartesian map-based model requires a 2D or 3D model of the environment, while a sensor-based model requires a structural description of the environment. *Sensor data processing* is responsible for feature extraction of raw sensor data, fusion of data from various sensors, and finally generation of the environmental models. The selection of sensors for a mobile robot has significant impact on its autonomous capabilities, for this reason a variety of sensors are often used so that the inherent weaknesses of one technology can be overcome. *Position estimation* refers to the determination of the position of objects relative to the robot, that is, in a local coordinate frame as well as in a global coordinate frame. Various sensors and techniques are used for position estimation and the robot must be able to accurately locate itself in a reference frame.

### 3.4.2 Obstacle detection and avoidance

Navigational systems for mobile robots can be considered in two phases, a path-planning phase and a path-following phase. Real-time obstacle detection and avoidance are two of the key issues relating to successful motion planning of a mobile robot. Path-planning can be further subdivided into two categories as static (when the obstacles are stationary) and dynamic (when the obstacles are moving or changing shape or size). The environment further dictates the algorithms employed by a mobile robot, since the environment could be completely known, that is when the trajectory of an obstacle is known (*a priori*), or when the environment is partially known. The main problem in obstacle avoidance as highlighted by Turennout et al. (1989) is the cooperation between the sensor system and the trajectory control system. Principally, the path-planning phase encompasses all trajectory controls - trajectory control normally receives a path specification from the global path-planning algorithm. As soon as an unexpected object is detected in its path, a robot must rely on its sensors to guide it along the contour of the object. A successful algorithm for avoiding obstacles can only be implemented if the size and shape of the obstacles are known. If the robot is only equipped with distance measuring sensor, an estimate of the obstacle's position can be obtained, but not of its size and shape. Turennout et al. (1989) developed an obstacle avoidance algorithm for contour and flat surfaces using a robot called PAVLOV, equipped with four ultrasonic sensors. The robot was able to avoid convex and non-convex surfaces in indoor environments.

All mobile robots feature some forms of collision avoidance techniques, ranging from simple algorithms that detect an object and subsequently stop the robot in order to prevent a collision, to complicated algorithms that enable the robot to detour an obstacle. The complicated algorithms involve not only a detection component but also quantitative measurements concerning the obstacle's dimensions. Major research efforts have been applied in finding solutions to the problem of motion planning in known environments with largely static obstacles and to a lesser degree, dynamic obstacles. The main objectives were to determine a collision-free path from a starting point to a goal point and at the same time optimize the performance of the robot. Some of these techniques include accessibility graphs, tangent graphs, visibility graphs, retraction methods and visibility graphs (Fujimura, 1991; Hwang and Ahuja, 1992). A collision cone approach was used by Chakravarthy and Ghose (1998) as an aid of collision detection and avoidance between irregularly shaped moving objects with unknown trajectories. The collision cone concept has its root in aerospace literature and is basically a concept that provides a convenient means of determining whether any two moving objects are on a collision course. Borenstein and Koren (1989) developed and implemented a real-time collision approach called the 'virtual force field'. This concept was an integration of two other concepts known as certainty grid (for obstacle representation) and potential fields (for navigation). This combination method was suitable for navigational systems using inaccurate sensors (e.g. ultrasonic) and systems with sensor fusion algorithm. One great advantage of the method was that the motion of the robot could continue without stopping in front of the obstacles.

Two other common approaches to obstacle avoidance are the wall-follow method and edge-detection method. In the wall-follow method, the robot moves along predefined distances from a wall. When an obstacle is detected, the robot regards the obstacle as part of the wall and continues it course around the obstacle at the predefined distance. This method is less versatile and is only suitable for few specific applications. In the edge-detection method, the robot determines the vertical edges of the obstacles and afterwards steers the robot around either edge (Borenstein and Koren, 1988). The boundary of the obstacle was represented by a line connecting the two edges. One of the disadvantages of the edge-detection method was that the robot had to stop in front of an object to allow for an accurate measurement.

Montano and Asenio (1997) studied the obstacle avoidance problem and subsequently developed a real-time navigation system based on artificial potential field using data from a 3D laser sensor. Kinematic and dynamic models of the robot were explicitly created and the robot was able to follow its path and make correction to that path when an unknown obstacle appeared.

### 3.4.3 Position determination

*Position estimation*, or *spatial localization* as it is alternately called, is one of the primary requirements of any autonomous navigational tasks. The environment and the information available to a robot dictate the design of a navigation algorithm. The technique for determining the position of a robot in its workspace varies significantly with the conditions under which the robot operates and the sensors it is equipped with. Most robots are equipped with wheel encoders that can be used to obtain an estimate of the robot's position at any instant. However, wheel slippage and quantization effects affect the accuracy of the position. These errors build up as time increases and the position estimates become more uncertain. Therefore, many mobile robots use other forms of sensing methods, such as vision, range, motion measurement (inertial), or a combination of various sensors to aid the position computation process. By fusing data from two or more sensors, the weakness(es) of one sensor can be minimized and the overall results can produce a much more accurate prediction of position. Sensors used for position prediction include sonar sensors, laser scanners, gyroscopes, radio beacons, and cameras.

The position and orientation determination techniques can be broadly classified into four groups: 1) landmark-based methods; 2) dead-reckoning and trajectory integration; 3) methods using a standard reference pattern; and 4) methods using *a priori* knowledge of a world model and then matching the sensor(s) data with the model. The landmark-based method is a popular approach, whereby the robot uses its approximate location to identify the landmarks in the environment (Krotkov, 1989). The landmarks can be naturally occurring, such as corners or edges, or artificial, such as specially placed beacons with known coordinates. Once the landmarks are identified and the range/altitude is measured relative to the robot, the robot's position and orientation can be computed in an absolute reference frame. The two main disadvantages of this method are the requirements for the availability of landmarks and the reliance on the robot's ability to detect them. In the second method (trajectory integration), the position and orientation of the robot are estimated by integrating over its trajectory and dead-reckoning approach. The robot maintains an estimate of its current position and orientation and continuously updates its estimate as it moves along (Matthies and Shafer, 1987). This technique, however, is not easy to accomplish, as it requires the robot to establish a correspondence between the features detected by the sensors at the current location and those at the previous location in order to plan the trajectory. The third method (standard reference pattern) requires the placement of standard reference patterns at known positions in the robot's environment. When these patterns are detected, the robot can compute its position and pose from the known locations of the pattern and its geometry. Researchers have used different patterns and geometry, and the position estimation technique varied accordingly (Kabuk and Arenas, 1987). In the fourth method, the robot is aided in its navigational tasks by providing *a priori* information about its environment in the form of a preloaded world-map. The basic objective using this technique is to sense the environment using on-board sensors and match the sensory observations with the preloaded world-map. This approach provides an estimate of the robot's position with reduced uncertainty and allows the robot to carry out other navigational tasks.

Urdiales et al. (1999) developed a fast localization algorithm for autonomous robots in dynamic environments based on definition of very small sized landmarks. The landmarks were calculated by obtaining the coordinates of the circular depth function from a ring of sonar sensors. Finally, a pyramid structure was used to enhance and fasten the performance of the localization algorithm. Koshizen et al. (1999) applied the concept of sensor fusion of odometry and sonar sensors by the Gaussian Mixture Bayes' Technique with Regularized Expectation Maximization (GMB-REM). A Nomad200 (Nomadic Technologies, California, USA) robot was used as a test robot and the results indicated that the sensor fusion scheme minimized the robot's position error no matter how much noise was present in the sonar model.

# IV. A LASER-RANGE-FINDER-BASED AUTONOMOUS GUIDANCE SYSTEM FOR A MOBILE ROBOT

## 4.1 INTRODUCTION

With the advent of precision farming, researchers are focusing on different techniques to automate navigation of agricultural vehicles. Additionally, machines and implements are getting bigger in physical size and higher in complexity, all of which contributes to the arduous tasks bestowed on their operators. The most frequently cited reason for implementation of an automatic guidance system is the need to relieve operators from continuously making steering adjustments while maintaining the machine and implement at an acceptable level of performance.

Automatic navigation control intends to steer a vehicle in a predetermined path automatically. The guidance system must be able to detect vehicle posture and thereby create appropriate steering commands to achieve the goal. In the last several decades, a variety of guidance systems, including leader cables, mechanical-based, radio-based, ultrasonic-based, and optical-based have been implemented for agricultural vehicles (Tillet, 1991). These guidance systems were all based on different technologies, but, fundamentally, most of them use the same guidance parameters, such as heading angle and offset to control steering (Zhang et al., 1999). Heading angle is defined as the angle between the vehicle centre line and the planned path, and the offset is the displacement of the vehicle central mass off the desired path.

Chateau et al. (2000) developed a guidance system for two agricultural vehicles using a laser range-finder (LRF). A combine harvester and a windrow harvester were used as test vehicles. A correlation-based approach was used to model the sensor parameters. With respect to the combine harvester, the LRF was placed on the left side of the conveyor on the machine and the authors assumed that the crop edge was always on the same side. Two models were subsequently developed, one for the crop edges and the other for the effects of dust. Dusts were usually generated by the passage of crop into the pickup of the harvester. When this occurred, the laser beam was retro-diffused when it encountered dust particles causing inaccuracies in distance measurements of the crop. Once the position of the sensor and the maximum height of the crop edge were known, the distance of the sensor from the crop was computed. A minimum threshold distance between the LRF and the crop was set to differentiate between the presence of dust from crop. When the LRF measurements were less than the threshold distance, they were considered as dust and the corresponding measurements were discarded. When the LRF was used on the windrow harvester, a windrow model of the crop was developed. This application was much simpler than the combine harvester model, in that the windrow harvester operated in a structured environment and the amount of dust generated was significantly less.

In a recent issue of the Spring 2005 *Landwards*, published by the Institution of Agricultural Engineers (UK), an article covered an automatic navigation system, namely Autopilot, used by the German Company, CLAAS (Brunnert, 2005). CLAAS has been using the Autopilot for their combine harvesters for over 25 years. Initially, the CLAAS Laserpilot was introduced, which was a 2-dimensional laser scanner that detected swaths, crop edges and other guidance aid. Most recently, the company used GPS in conjunction with other guidance sensors for their combine harvesters. Their integrated approach is much more accurate, and continues to get cheaper in terms of receivers cost and signal broadcast fees. Another advantage of this type of system is that they are not dependent on visual guidance aid, instead uses computed tracks.

A novel navigation method for a mobile robot using a LRF and a memory-based method was developed by Adachi et al. (2003). A robot memorized sequential scanning data of the LRF in a recording run. Localization of the robot was subsequently achieved by comparing the current scan and the memorized scan sequence in an autonomous run. A histogram-based technique was used to match the current scan and the memorized scan, calculate the angle between two the scans and the angular displacement between them; to rotate the current scan by the angular displacement; to calculate the translation histograms of two scans for horizontal and vertical directions and finally to calculate the highest correlation directions using the histograms. Similarly, Mazl and Preucil (2000) built 2D maps using data from a LRF mounted on a robot. The tasks were divided into two parts. The first part dealt with preprocessing of odometry and range information to determine the position and heading of a robot in the environment. The second part updated the internal world model of features, using the existing map and observed entities. Another method used for navigation is the Bayesian Segmentation Theory, implemented by Victorino and Rives (2004) on a mobile indoor robot. The segmentation theory was used to extract and track line segments parameters from successive laser scans. Subsequently a methodology was developed to estimate the line parameters and track the distances of the objects in the indoor environment. The tracked distances were then used in a feedback loop of a sensor-based control navigation strategy.

Development of a navigation algorithm that enabled robots to build their own maps of an environment and at the same time use the maps for localization is an important step towards creating successful autonomous systems. Simultaneous mapping and localization have been used extensively. Researchers have fused information from two or more sensors to improve performance. Diosi and Kleeman (2004) presented a method for combining measurements from a LRF and an advanced sonar array, to accurately measure bearing and range information. The advanced sonar array used was capable of classifying targets into different shapes, such as right angled, concave, corners, planes, point/edge features – in a single measurement cycle. The authors successfully fused line and corners measured from the LRF and the sonar array. Laser measurements, on the other hand, were used to simplify and improve the selection of reliable sonar point features and assisted in removing multiple reflection sonar phantom features. Ma and Moore (2003) also used a 2D laser and sonar sensors for collision avoidance for mobile robots based on the Histogramic-In-Motion-Mapping (HIMM) algorithm. A total of 26 sonar sensors were mounted around the robot and a 2D laser was placed in front. Decisions pertaining to collision avoidance were based on the HIMM map plus the velocity vector of the robot in its global coordinate system. Experimental results showed that the robot operated reliably in a dynamic parking lot operating at speeds of 2.7 km/h.

In this study a LRF was used to detect and set target points in the autonomous navigation algorithm, whilst a sonar sensor array was used primarily for obstacle detection and avoidance. The navigational algorithm developed, fused the data from both sensors to effectively navigate a robot in a maze layout. Previous researchers have fused the information from one or more sensors to achieve the same tasks (i.e. for obstacle detection or feature recognition), whereas in this study the sensors were used independently for different tasks.

## 4.2 OBJECTIVES

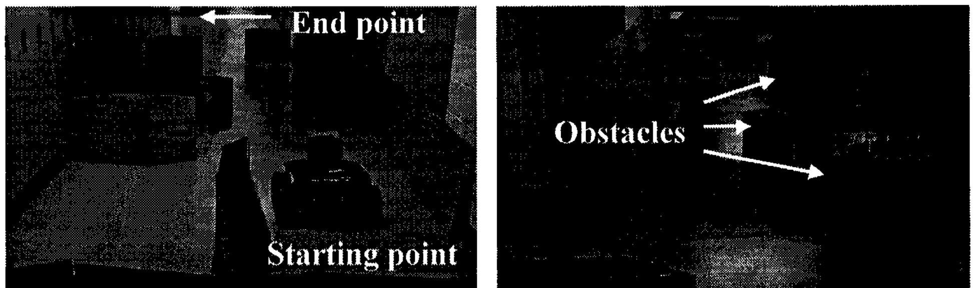
The main objective of this study was to develop an autonomous navigation system for a robot using a LRF, a sonar sensor array, a microcontroller, and an onboard computer. This study aimed at improving navigation and making the system more sensitive by using the principle of sensor fusion. The LRF was used mainly for navigation while a ring of eight sonar sensors was used for obstacle and wall detection. Finally, an algorithm was developed to fuse the data from the LRF and sonar in an appropriate sequence to effectively navigate the robot through a maze. The specific objectives included:

- To develop navigational algorithms utilizing data from a LRF and a ring of eight sonar sensors;
- To develop a LRF-based path planning method for a test robot platform;
- To enhance the navigational algorithms by incorporating obstacle detection and avoidance using data from the sonar sensors; and
- To test the navigational algorithms in an indoor maze.

# 4.3 MATERIALS AND METHODS

## 4.3.1 Experimental setup

An experimental maze was set up for testing the navigational algorithms (Figure 4.1). The maze was approximately 11 m long and 2.5 m wide.




(a) Front view (b) Rear view

**Figure 4.1: Layout of the maze**

The maze was constructed in a hallway made of concrete walls and rubber tiled floor. Some parts of the maze were bordered with lockers made of painted steel sheets. Additionally, cardboard was used in some parts to form a desired shape of the maze and to create internal routes and obstacles. The robot was initially placed at a starting point and subsequently programmed to autonomously navigate its way to the end of the maze, whilst detecting and avoiding obstacles.

## 4.3.2 Hardware components

The hardware components used in the research included a robot platform and two external navigation sensors (a ring of eight sonar sensors and a LRF).

#### 4.3.2.1 The robot platform

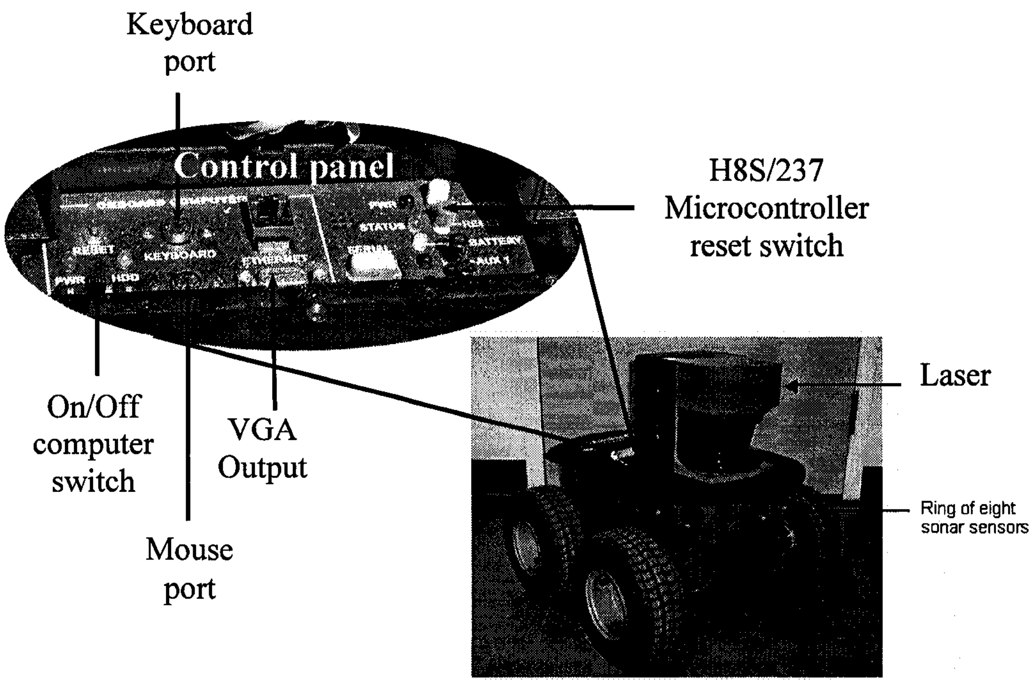



**Figure 4.2: The P3-AT Robot**

A Pioneer 3 All Terrain robot (P3-AT) developed by ActivMedia Robotics (New Hampshire, USA) was used as a test platform. The robot had a 500 mm(L) ×490 mm(W)×260 mm(H) aluminum body with a body clearance of 80 mm. The total weight of the robot was 12 kg including three 12 V batteries. It was four-wheel driven (215 mm diameter drive wheels), employing a skid steering approach for turning. In terms of mobility, it comprised of four DC wheel motors, which used a 66:1 gear ratios and contained two 100-ticks wheel encoders. The maximum translational speed that could be achieved was 4.32 km/h and the maximum traversable slope was a 40% grade. In addition to the two wheel encoders, a vibrating gyroscope chip (ADXRS300, Analog Devices, Massachusetts, USA) was also mounted on the robot. Both the wheel encoders and gyroscope were used to determine the position and orientation of the robot. A dead-reckoning method based on the two wheel encoder readings was used to compute the position and orientation of the robot. In order to compensate for wheel slippage from the skid steering and slippery ground conditions, the dead-reckoning results were fused together with the inertial measurement readings obtained from the gyroscope. A simple Kalman Filter was used to compute the final orientation and position.

The robot was equipped with an 850 MHz single-board computer (VSBC8, VersaLogic Corporation, Oregon, USA) and an 18 MHz microcontroller (H8S/237, Hitachi, Japan), which were both used for acquisition and processing of the data received from the sensors. The control panel of the robot consisted of several ports so as to create a user interface to the onboard computer. These included a mouse port, a keyboard port, a VGA output for a computer monitor, and an ethernet port. There was also a reset push button switch for the microcontroller, which stopped the program from being executed, once it was pressed. Separate LEDs on the control panel served as visual indicators for the power status of the batteries, the microcontroller and the computer.

The single-board computer in particular, allowed for autonomous navigation operations of the robot and was used for acquiring and processing LRF data in real-time. Due to a large amount of data from the LRF to be analyzed, the microcontroller alone did not have the capacity to handle it. The LRF data were analyzed every 100 ms. This time was set in the software, and could be changed as desired. The microcontroller controlled the wheel motors, and acquired data from the sonar sensors, wheel encoders, and gyroscope, and prepared data packets to be sent to the single-board computer for interpretation and analysis. The microcontroller also received instructions from the computer and sent appropriate signals to the robot's components. These instructions included heading directions, steering, and speed commands.

#### 4.3.2.2 Sensors

<u>Laser range finder</u>

A SICK LMS200 LRF (SICK, Waldkirch, Germany) was mounted on the robot for navigational purposes, to detect target locations and plan the robot's paths.

The LRF was based on the principle of "time of flight" measurement as shown in Figure 4.3.

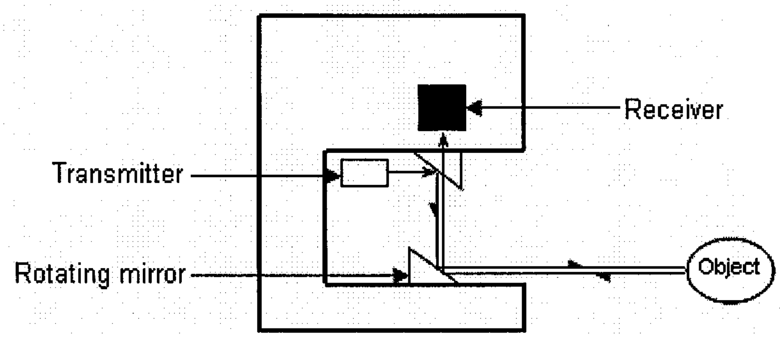



**Figure 4.3: The principle of operation of a LRF (Ye and Borenstein, 2002)**

A pulsed laser-beam was transmitted and then deflected off a rotating mirror so that a fan-shaped scan was made of the surrounding area. If an object was in the path of the laser beam, it was reflected and registered in the laser receiver. The time between the emission and reception of the pulsed laser-beam was directly proportional to the distance between the object and the laser. The angular resolution of the scan was selectable at 0.5° or 1°. In this study, the 0.5° resolution was used to produce 361 measured values in a 180° scan. This resolution was selected so as to achieve a higher sensitivity and detection accuracy in the readings to form a basis for accurate and detailed mapping capabilities. The dimension of the LRF was about 185 mm(L)×155 mm(W)×156 mm(H) with a weight of 4.5 kg. The maximum measurement range was 150 m, with a resolution of 10 mm. The LRF communicated with the single-board computer via a RS-232 serial interface with a baud rate of 38.4 kbaud. The advantages of using a LRF as a guidance sensor were: the nature of non-contact measurements, high measurement resolution, high scanning frequency (75 Hz), real-time data processing, and no requirement for illumination of target objects. Although the surface properties (color and material) of a target tend to affect the reflectance of a laser beam, the LRF was reliable up to a range of 20 m regardless of reflective properties, shape or color of an object.

Sonar sensors

A ring of eight sonar sensors was mounted on the robot. Figure 4.4 illustrates the sonar array, mounted at a height of 220 mm above ground level. The sonar had a sensitivity range between 100 mm and 7 m.

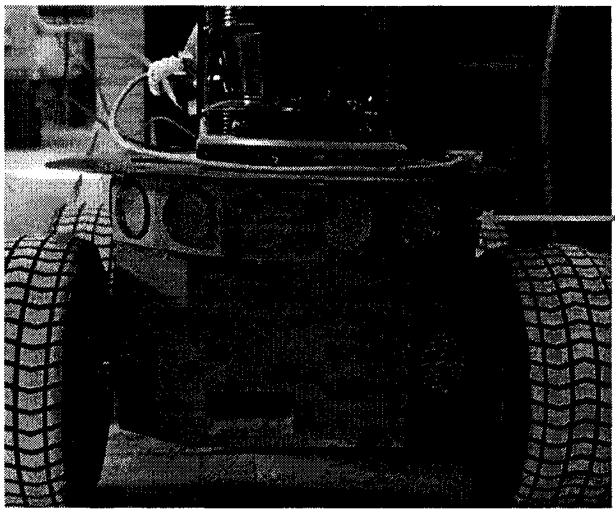



**Figure 4.4: Sonar sensors array**

The sonar sensors used were of the active type, whereby sound waves were generated and transmitted from the sensors. The sound waves were reflected when they encountered an object and received by a receiver that amplified the echo. Once the speed of the sound wave and the time taken for the signal to bounce back to the receiver were known, the distance of the object could be calculated. Each sonar sensor had its own driver electronics to enable an independent control, and a transducer for object detection and distance measurement. The eight sonar sensors were multiplexed and the acquisition rate of the array was set at 25 Hz (40 milliseconds per sonar sensor). A gain control adjuster was located under the robot's panel, which was used to control the sensitivity and range depending on the environment in which the robot was working. Low-gain settings reduced the robot's ability to detect small obstacles, therefore, was useful when operating in noisy environments, or on uneven or highly reflective surface. By increasing the sensitivity, the sonar sensors were able to detect smaller obstacles and objects at a distance further away.

### 4.3.2.3 Sensor's coordinate system

Each sonar sensor was positioned at an angle indicated in Figure 4.5. The LRF was located off the centre of the robot at an x-y coordinate of (160 mm, 7 mm). The robot's origin was defined as the centre of the robot, which was also the origin of the robot's local coordinate frame. ($Y_1$-$O_1$-$X_1$)

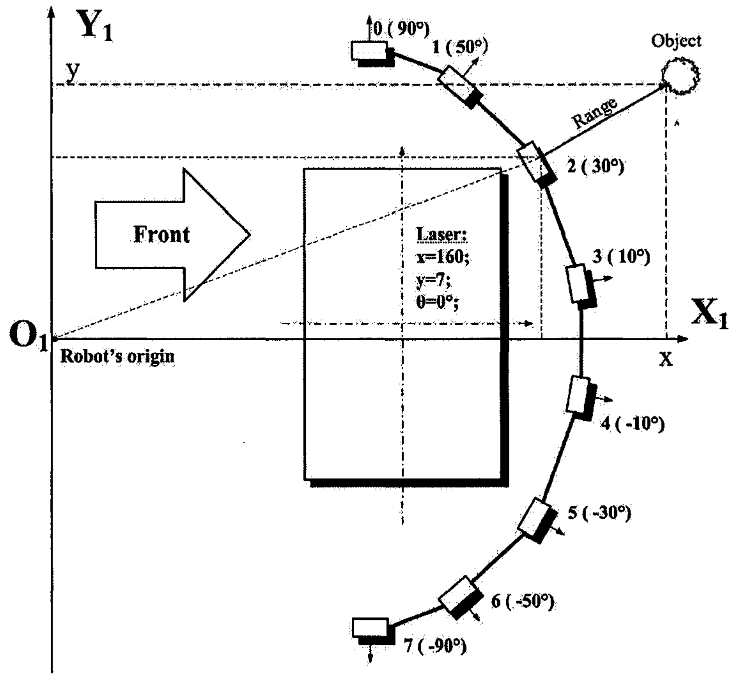



**Figure 4.5: Coordinates of the LRF and sonar sensor array**

**Table 4.1: Sonar sensor position**

| Sonar sensor | x (mm) | y (mm) | Heading (°) |
|---|---|---|---|
| 0 | 145 | 130 | 90 |
| 1 | 185 | 115 | 50 |
| 2 | 220 | 80 | 30 |
| 3 | 240 | 25 | 10 |
| 4 | 240 | -25 | -10 |
| 5 | 220 | -80 | -30 |
| 6 | 185 | -115 | -50 |
| 7 | 145 | -130 | -90 |

As shown in Figure 4.5, sonar sensors 1-6 were placed facing outwards at 20° interval, whilst sonar sensors 0 and 7 were each placed at +90° and -90° respectively. This arrangement provided a 180° coverage in front of the robot. The sonar sensors were placed in a symmetry pattern about the $X_1$ axis of the robot. Sonar numbers 0-3 were in symmetry with 4-7. Table 4.1 also shows the coordinates of the sonar sensors relative to the robot's origin $O_1$.

### 4.3.2.4 Robotic navigation components

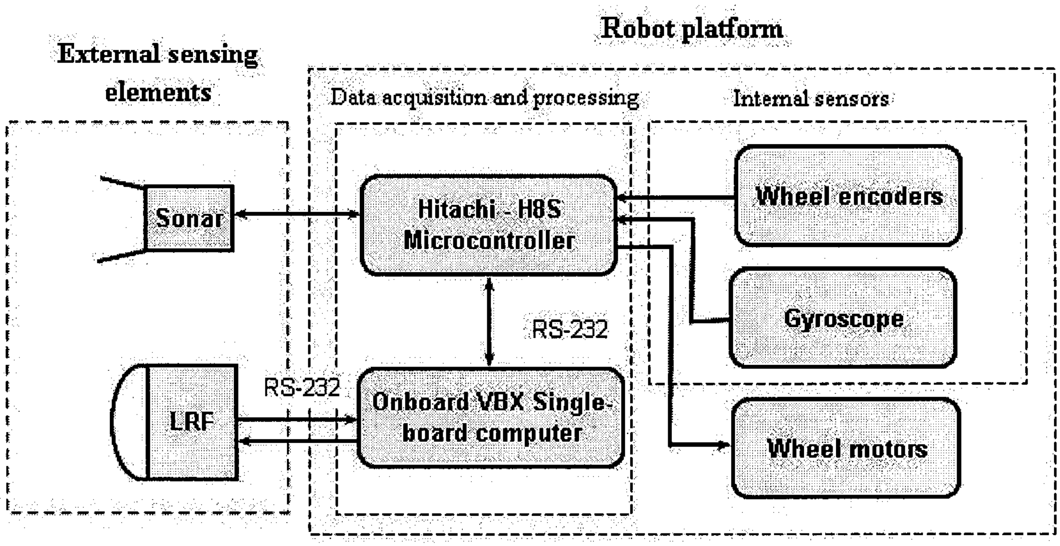



**Figure 4.6: Robot's navigation components**

As shown in Figure 4.6, the single-board computer communicated with the LRF and the microcontroller through RS-232 serial interfaces. The P3-AT robot used a client-server mobile robot control architecture. The single-board computer was the client, and the microcontroller was the server. The microcontroller was loaded with the robot's server operating software, ActivMedia Robotics Operating System (AROS, ActivMedia Robotics, New Hampshire, USA). This operating system was stored on the FLASH ROM of the microcontroller. The controller server managed all the low-level details of the robot's controls and operations, including motion, heading, and odometry. It also included firing the sonar sensors, collecting data from the sonar sensors, wheel encoders and gyroscope. The server (the microcontroller) communicated with the control client (the single-board computer) using special client-server communication packet protocols. Two basic command packets were used; one packet from the client to server and another from the server to client (Server Information Packets, SIP).

The operating system on the microcontroller had a structured command format for receiving and responding to directions from the computer for control and operation of the robot. The number of commands that could be sent to the microcontroller within a specific period depended on the baud rate and synchronicity link of the serial interface. In the design system, 115.2 kbaud was used. The AROS on the microcontroller accepted several different computer-motion commands, but of two mutually exclusive types; either independent-wheel velocity mode or platform translational/rotational mode. During independent wheel velocity mode, the robot's microcontroller tried to maintain precise wheel velocities, while in the translational/rotational mode, the microcontroller maintained both platform speed and heading. All the motion command arguments sent by the single-board computer to the microcontroller used units of millimeters or degrees, and AROS converted these units into encoder-related motion values using two separate parameters, one for translation and the other for rotation.

The user also had the flexibility of using special client commands on the single-board computer to either enable or disable any particular sonar, to change the firing sequence or to change the cycle time. In this research, however, the sonar sensors were fired in an order of 0-7 and the cycle time was set at 40 ms. The onboard gyroscope was used to compensate robot heading changes that weren't detected by the wheel encoders, such as from wheel slippage, gearbox play, wheel imbalance, or surface conditions. The microcontroller collected 10-bit gyro rate and 8-bit temperature data every 25 ms and upon request sent the data to the single-board computer in a SIP. Adjustments to the robot's heading and position were both done on the onboard computer.

### 4.3.3 Software design

#### 4.3.3.1 Software structure

The overall control for the robot platform, sensors, and other components was based on a client-server architecture. The microcontroller acted as a server to manage the low-level tasks of robot controls and operations, including motion, heading and odometry, and acquiring data from sensors, e.g. the sonar sensor array. The single-board computer was an intelligent client to implement the full gamut of robotic control strategies and tasks, such as obstacle detection and avoidance, sensor fusion, localization, features recognition, mapping, intelligent navigation, etc. depending on various applications. The communication between the client and the server was through a RS-232 connection. The lowest levels of client-server interactions included serial communications, server information packet processing, cycle timing, and a variety of accessory controls, such as sonar sensors, LRF, gyroscope, and wheel encoders.

The ActivMedia Robotics Interface Application (ARIA, ActivMedia, NH, USA) was a client-side software, written in C++, and run on the single-board computer. ARIA is a development Application Programming Interface (API) to facilitate object-oriented development of various robot control and management applications. A library was provided by the manufacturer, with a large amount of functions for data acquisition and processing, robot control and management, and other general-purpose routines. ARIA is a flexible open-source environment and highly multi-threaded. Each thread was designed to realize a task. Figure 4.7 shows a main structure of a thread.

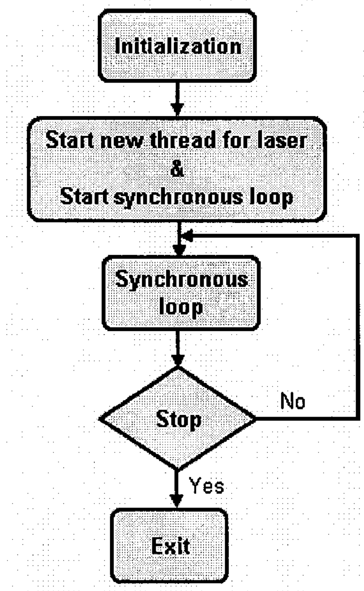



**Figure 4.7: Structure of a software thread**

To avoid the situation that multiple threads tried to access and handle the same data sources, the data sources were protected with ARIA synchronization objects, mutual exclusive objects and suspension objects. The mutual exclusive objects guaranteed that only one thread accessed a data source at a time. The mutual exclusive object locked the data source. Other threads were not allowed to access the data source and had to wait until the data source was released by the mutual exclusive thread. If no mutual exclusive object was used, ARIA granted access of the data source to the requesting thread. The suspension object worked in conjunction with the mutual exclusive object to delay the execution of a thread. This object was used to put the requesting thread to sleep whilst waiting for the mutual exclusive object to free the data source. An alternative was to have the requesting thread continuously check for a change in condition, i.e. whether the mutual exclusive object free the data source.

The heart of an ARIA thread structure was the synchronous task loop (Figure 4.7), which collected, organized, and managed the robot's operating states. It formed a convenient interface for other ARIA components as well as upper-level applications to access the robot state-reflection information for assessment, planning, and ultimately intelligent, purposeful control of the platform and its accessories. One of the main tasks of the synchronous loop was to maintain the clockwork cycles and multi-threaded rhythms of the robot's control system.

In this study, the synchronous task loop consisted of a root class, branch classes, and leaf classes (Figure 4.8). The root class acted as a main stem, which linked all the branch classes and the leaf classes. Two main branch classes were a sensor handling class and a user-task class. The sensor handling class dealt with all the leaf classes related to sonar sensors and LRF operations, while the user-task classes handled the measurements for the navigation. Several leaf classes, State Reflector, Packet Handler, Robot Locker, Robot Unlocker, and Action Handler, were attached to the root class, which administrated initialization and operations of the thread.

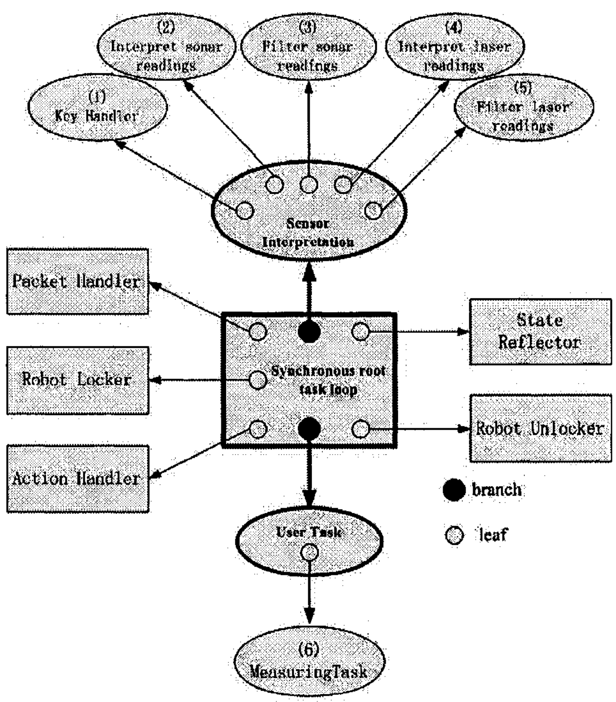


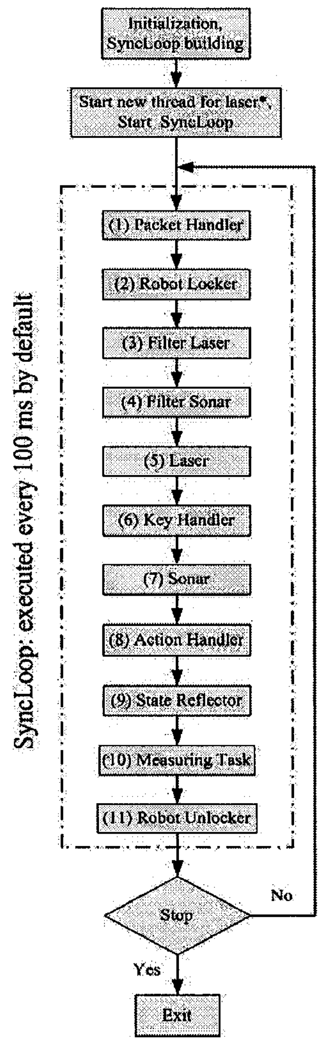

**Figure 4.8: Architecture of software design**

The state reflection referred to the distribution of the robot's operating conditions and values as extracted from the latest standard SIP. The packet handler dealt with the low-level details of constructing and sending client-command packets to the server, i.e. the microcontroller. It also received and decoded the various SIPs received from the server. The SIPs, the standard SIP or extended SIP, were sent by the server and contained information about the robot and its accessories. The standard SIPs were received every 100 ms, which had the information of the robot's current position, heading, translational and rotational speeds, and freshly accumulated sonar sensor readings. The extended SIPs contained different operating information, such as input/output port readings of the server. They were sent by the server only when they were explicitly requested by the client computer.

The action handler contained two types of action commands, direct motion commands and motion commands. The direct motion commands consisted of 1-byte simple commands, which were sent directly to the microcontroller, and were used to either enable or disable the wheel motors. The motion commands were sent by the client software, i.e. ARIA, and used to control the mobility of the robot, e.g. to set individual wheel velocity, or coordinated translational and rotational velocities, to change the robot's absolute or relative heading, to move the robot for a specific distance, and to stop the robot. To realize these operations, action classes were programmed for each operation, respectively, and executed based on priority (lowest priority goes last) in each synchronous task cycle prior to state reflection.

In the sensor handler class, six leaf functions were attached (Figure 4.8). The key handler (1) was used for interfacing the keyboard with the thread in order to be able to set and change modes of robot operations by pressing the appropriate key on the keyboard, to enter file names for data storage, and to set target coordinates for the robot using keyboard inputs. The escape key was specially granted to stop the overall program. The sonar sensor functions (2 and 3) were specifically designed for interpreting and filtering sonar sensor data. Similarly, LRF interpretation and LRF filter functions (4 and 5) were responsible for processing the laser readings.

Figure 4.9 summarizes the main thread for the automatic navigation. A total of eleven functions were included in the synchronous loop. To improve the efficiency of the program, a special thread ran in parallel with the main thread to handle the acquisition of large amount of the LRF data.




* this new thread mainly performs LRF packet handling

**Figure 4.9: Architecture of software design for the autonomous navigation system**

### 4.3.3.2 The coordinate transformation method

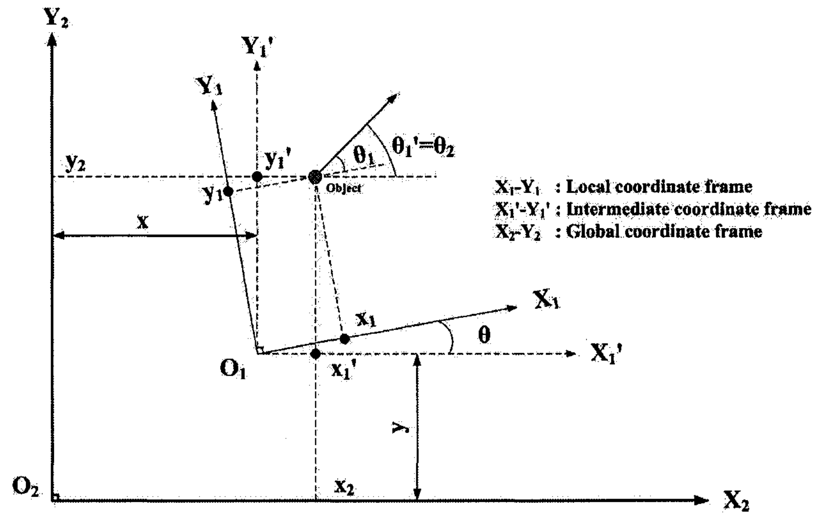



**Figure 4.10: Coordinate transformation principle**

Three coordinate frames were defined in order to determine the actual position of an object. Coordinate frame 1 - $Y_1$-$O_1$-$X_1$ was defined as the local coordinate frame; coordinate frame 2 - $Y_1'$-$O_1$-$X_1'$ was defined as the intermediate coordinate frame after the local coordinate frame has been rotated by θ; and coordinate frame 3 - $Y_2$-$O_2$-$X_2$ was defined as the global coordinate frame. The global coordinate system of the robot was defined as the origin of the robot's position (start of the maze). $\theta_1$ was the heading of the object in the local frame and $\theta_1'$ was the heading of the object in the intermediate frame ($Y_1'$-$0_1$-$X_1'$), whilst $y_1$, $x_1$ was the Cartesian coordinates of the object in the local frame and $y_1'$, $x_1'$ was the Cartesian coordinates in the intermediate frame. Note that $\theta_1'$ was equal to $\theta_2$. The following expressions were then used to solve for the position of the object in the global coordinate frame, i.e. $x_2$, $y_2$ and $\theta_2$.

$$x_2 = x + x_1' \tag{4.1}$$

$$x_2 = x + x_1 \cos(-\theta) + y_1 \sin(-\theta) \tag{4.2}$$

$$y_2 = y + y_1' \tag{4.3}$$

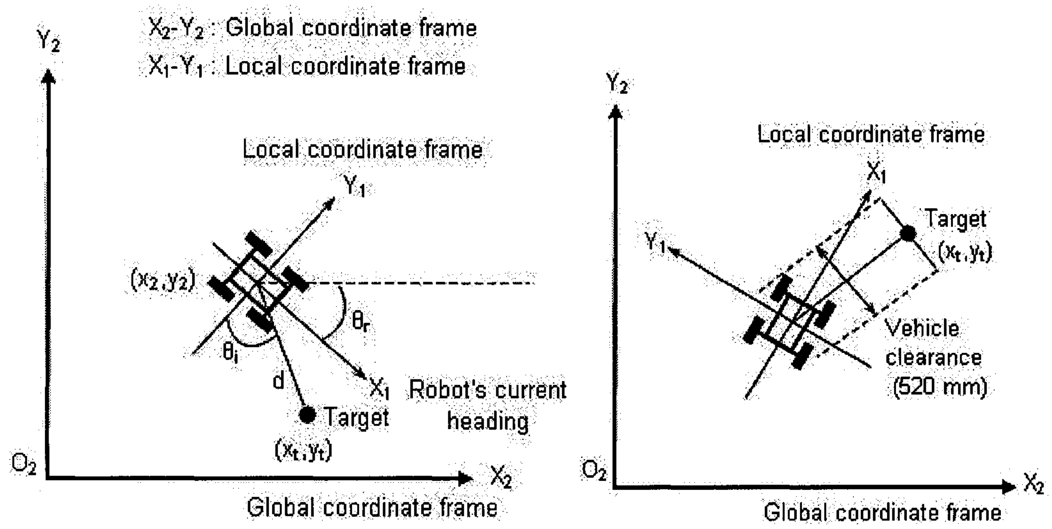
$$y_2 = y + y_1 \cos(-\theta) - x_1 \sin(-\theta) \tag{4.4}$$

$$\theta_2 = \theta_1 + \theta \tag{4.5}$$

The functions developed for the LRF and sonar sensors had the capability to locate objects in a local coordinate frame as well as in a global frame. Two types of position parameters were generated. Firstly, the sonar sensors and LRF acquired the distances of an object, and secondly, the Cartesian coordinates of the object in the local coordinate frame ($Y_l$-$O_l$-$X_l$) were calculated. Once the coordinates of the object from $Y_l$-$O_l$-$X_l$ were known, equations 4.1-4.5 were used to calculate the position of that object in the global frame. This coordinate transformation technique is shown in Figure 4.10 and is specifically applied to obstacle detection and avoidance.

### 4.3.3.3 Path planning concept

The path planning methods of the robot are illustrated in Figure 4.11. Once the LRF detected an exit with a vehicle clearance for the robot, a target coordinate was then transferred to the microcontroller, which sent a signal to the wheel motors to effect the change. An exit for the robot was defined as a point where the robot detected no obstacle and an aperture was present. The readings received from the LRF were analyzed and the point with the largest distance was then set as its target. The target coordinate $(x_t, y_t)$ was based on the robot's current position and the global coordinate frame (origin). However, the target would always have been a surface. So, compensation was made for the vehicle clearance, and a constant value of 250 mm was subtracted from the actual target distance detected by the LRF. A minimum threshold value of 200 mm was also set, for which a target must be greater than. This was incorporated to prevent the robot from setting small target distances in the case of facing a wall only a few centimeters away. If a target point was identified, but there was not enough clearance for the robot to proceed, the target point was discarded. A predetermined value of 520 mm was set as the clearance threshold. Any target point with a clearance value greater than 520 mm was considered as an exit.




**(a)** **(b)**

**Figure 4.11: Robot's path planning system**

The target distance (d) and the target angle ($\theta_i$) were obtained from the LRF data. The target angle was always relative to the robot's local coordinate frame rather than the global coordinate frame. The robot's current position ($x_2 y_2$) and heading ($\theta_r$) relative to the global coordinate frame were obtained from the navigational algorithm. The target positions ($x_t$, $y_t$) in the global coordinate frame were calculated using equations 4.6 and 4.7.

$$x_t = x_2 + d\cos(\theta_i + \theta_r - 90) \qquad (4.6)$$

$$x_t = y_2 + d\sin(\theta_i + \theta_r - 90) \qquad (4.7)$$

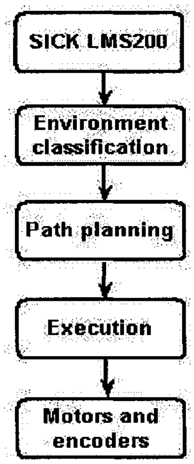



**Figure 4.12: Flowchart of obstacle avoidance**

The obstacle detection and avoidance algorithm consisted of three main tasks, namely, environment classification, path planning, and execution. The environment classification task identified all obstacles in front of the robot and calculated safe and unsafe regions in which to travel. The array of eight sonar sensors was used for this purpose, which covered a 180° range in front of the robot. Once the region classification was completed a path was planned, either around the obstacle in the event of an obstacle being present or to set a target coordinate ($x_t$, $y_t$) once an exit was identified (Section 4.3.3.4). Control signals were sent to the wheel motors to drive the robot towards the target. The flowchart of the obstacle detection and avoidance algorithm is shown in Figure 4.12.

### 4.3.3.4 Design of the navigational program in a structured maze

A schematic of the navigational algorithm developed is shown in Figure 4.13.

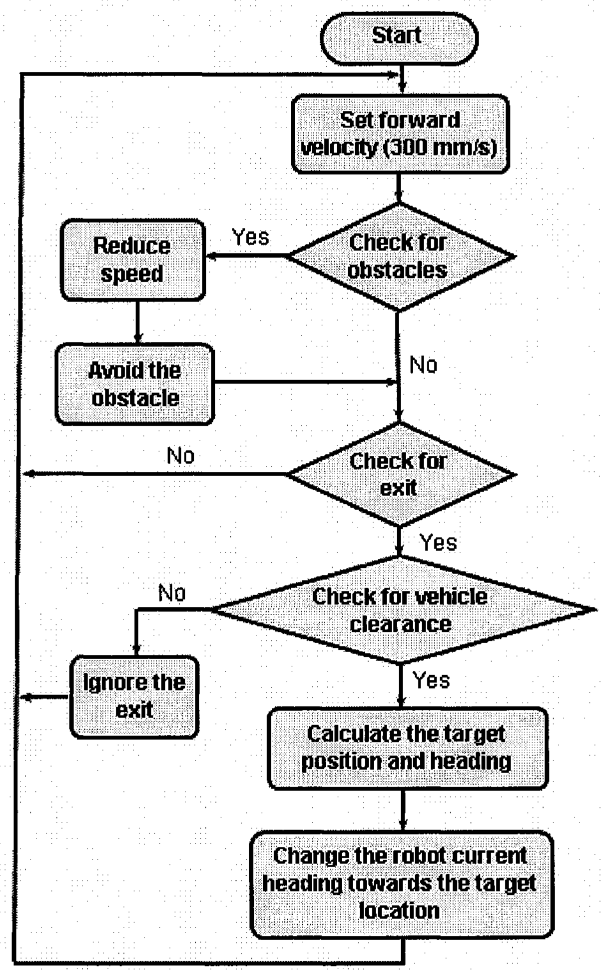



**Figure 4.13: Schematic of software design**

The forward speed of the robot was initially set at 1.08 km/h in the forward direction. As the robot moved forward it continuously scanned its environment, checking for obstacles. If an obstacle was present the robot reduced its speed and avoided the obstacle. If no obstacle was present the LRF data was used to check for an exit. If no exit was found, the robot continued in its current path. If an exit was found, the navigational algorithm checked for vehicle clearance so as to ensure that it was possible to traverse the planned path. If there was not enough clearance, the robot ignored that exit. However, if there was sufficient vehicle clearance for the robot, the robot set that exit point as its target and set its heading towards that point.

## 4.4 EXPERIMENTAL DESIGN

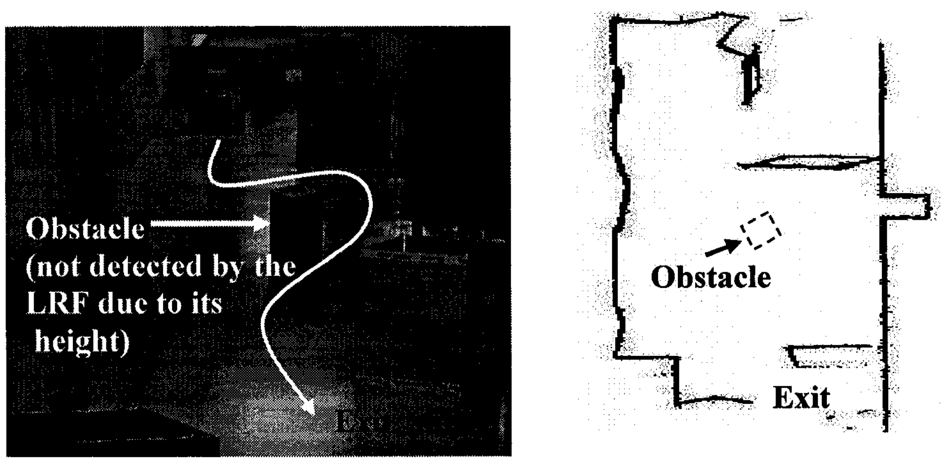
Five trials were conducted from the starting point as indicated in Figure 4.1. The robot's performance was monitored in terms of its ability to detect and avoid obstacles and walls, and at the same time be able to find an exit of the maze. The path planning concept and navigational algorithms were evaluated visually by observation.

## 4.5 RESULTS AND DISCUSSION

The robot traveled at a speed of 1.08 km/h while the LRF was scanning 180° and the sonar ring of eight was activated. The robot successfully avoided obstacles and walls, and navigated through the maze to the exit as shown in Figure 4.1. The navigational algorithm was tested five times in the maze layout, and the robot successfully arrived at the exit whilst avoiding obstacles. The sonar sensors were used to avoid obstacles and detect the walls of the maze in particular, whilst the LRF was used to scan the environment to locate possible apertures, thereby identifying target locations. Since the ring of sonar sensor was installed 80 mm below the LRF horizontal scanning plane (300 mm above the ground), the sonar sensors were able to detect obstacles that were not in the plane of the LRF. The LRF successfully scanned its environment and set target points based on threshold values that were preset in the software.

The sonar sensors arrangement as depicted in Figure 4.5 were all set at varying threshold values at which it would stop the robot in the event of it coming in close proximity to an obstacle which was equal to or less than the threshold values. These values range from 200 mm for sonar 0 and 7; 300 mm for sonar 1 and 6; 400 mm for sonar 2, 3, 4, and 5. The results showed that this arrangement worked satisfactorily.

The results revealed that special considerations need to be taken into account when using the single-board computer and the microcontroller to acquire and process real-time data from the LRF and sonar sensors. The data acquired using the LRF could not be logged to a file and analyzed simultaneously to determine target directions as it created a conflict between writing and retrieving the data from the same file. Readings from the LRF were analyzed instantaneously by the single-board computer. It must also be pointed out that although the LRF reading might have indicated a possible target direction, the algorithm developed tested the environment to determine whether there was enough clearance for the vehicle. If there were not enough clearance the robot would continue going forward.




(a) (b)

**Figure 4.14: Path of the robot and map generated by the LRF**

Figure 4.14 (a) illustrates the actual route of the robot and the map generated (Figure 4.14 (b)) by the LRF of the maze whilst traveling. The LRF readings were logged to a file and the data points collected were used to generate a map.

It was also observed that when the robot was programmed to travel at speeds of approximately 0.54 km/h and lower, the readings from the LRF remained unchanged for about 10 successive scans. Further tests need to be carried out to investigate the cause and to find a solution.

The use of sonar sensors for obstacle detection and avoidance garner lots of support since they are relatively cheap and simple to implement. However, there are some interesting effects in the use of sonar sensors, such as noise, interference, beam spreading, scattering and undesired reflections. The surface affects the angle at which the sound wave from the sonar sensors is reflected, as smooth surfaces tend to reflect the sound energy at a perpendicular direction to its surface. If the surface is irregularly shaped, the sound wave may be reflected in a direction that is out of range of the field of view of the sensors, resulting in the robot being unable to detect an object. The scattering behavior of the sound wave is completely dependent on the surface structure and this may be a disadvantage of using sonar sensors in some environment.

Another consideration that must be taken into account in any automatic guidance system is accurate position determination. The P3-AT robot used odometric measurements from two wheel encoders, coupled with readings from an onboard gyroscope to compute its position by way of a simple Kalman filter. Also, wheel slippage was not taken into account, though there was some degree of slippage using skid steering on a tiled surface. This method have incurred some errors in the path planning algorithm when targets were set, as the actual position may have been different from the computed position and orientation of the robot due to the errors mentioned above.

## 4.6 CONCLUSIONS

The algorithm developed for the autonomous navigation of a P3-AT robot worked successfully in a maze. The LRF scanned 180° range in front of the robot and was able to identify the navigational routine in the maze. The sonar sensors, on the other hand, satisfactorily provided useful data to successfully enable collision avoidance. The data from the LRF was processed in real-time and instructions were sent from the microcontroller to the wheel motors to effect the change required. The guidance system developed comprising of the fusion of data from the sonar sensors and LRF provided real-time information to the single-board computer, which in turn processed and analyzed the data to satisfactorily navigate the robot through a maze, detect obstacles and prevent collisions. A path planning method to identify target points and to generate a path were successfully developed and tested in the maze.

## 4.7 REFERENCES


Adachi, Y., H. Saito, Y. Matsumoto, and T. Ogasawara. 2003. Memory-based navigation using data sequence of laser range finder. *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation.* Vol. 1, 479-484

Brunnert, A. 2005. Automatic guidance of harvesting machinery – From mechanical sensors to Global Position Systems. *Landward* . Vol. 60 (2), 2-4

Chateau, T., C. Debain, F. Collange, L. Trassoudaine, and J. Alizon. 2000. Automatic guidance of agricultural vehicles using a laser sensor. *Computers and Electronic in Agriculture*. Vol. 8 (3), 243-257

Diosi, A., and L. Kleeman. 2004. Advanced sonar and laser range finder fusion for simultaneous localization and mapping. *Proceeding of the IEEE International Conference on Intelligent Robots and Systems.* Vol. 2, 1854-1859.

Han, S., Q. Zhang, H. Noh, and B. Shin. 2004. A dynamic performance evaluation method for DGPS receivers under linear parallel tracking applications. *Transactions of the American Society of Agricultural Engineers*. Vol. 47 (1), 321-329.

Ma, L., and K.L. Moore. 2003. Sonar and laser based HIMM map building for collision avoidance for mobile robots. *IEEE International Symposium on Intelligent Control*. 755-760.

Mazl, R., and L. Preucil. 2000. Building a 2D environment map from laser range-finder data. *Proceedings of the IEEE Intelligent Vehicles Symposium*. 290-295

Ollis, M., and A. Stenz. 1996. First results in vision-based crop line tracking. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 1, 951-956

Tillet, N.D. 1991. Automatic guidance for agricultural field machines: A review. *Journal of Agricultural Engineering Research.* Vol. 50,167-187

Victorino, A.C., and P. Rives. Bayesian segmentation of laser range scan for indoor navigation. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 3, 2731-2736

Ye, C., and J. Borenstein. 2002. Characterization of a 2D laser scanner for mobile robot obstacle negotiation. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 3, 2512-2518

Zhang, Q., J.F. Reid, and N. Noguchi. 1999. Agricultural vehicle navigation using multiple guidance sensors. *Proceedings of the International Conference on Field and Service Robotics*. Pittsburg. UILU-ENG-99-7013.

# CONNECTING TEXT

Accurate position and orientation estimates are two of the fundamental factors underlying a successful autonomous navigation system. Dead-reckoning using wheel encoders and inertial measurements from a gyroscope are two common approaches. By using more than one sensor for position estimation, the weakness(es) of one sensor modality can be minimized. Typically, a Kalman filter is used to fuse the measurements from the wheel encoders and a gyroscope to provide an accurate estimate of positions. Chapter V describes an artificial landmark localization method using a laser range-finder (LRF) for position and orientation determination. This method is compared with dead-reckoning and a dead-reckoning with gyroscope correction method.

# V. POSITION DETERMINATION USING LANDMARK LOCALIZATION FOR A MOBILE ROBOT

## 5.1 INTRODUCTION

The concept of the automatic guidance of agricultural vehicles dates back to the late fifties when leader cable guidance systems were used (Morgan, 1958). As time progressed, various sensing methods for vehicle positioning have been developed. These methods can be placed into two main categories, namely ground-based and satellite-based. Sensors can also be classified in various categories such as motion measurements (odometry and inertial), artificial landmarks (laser and radar), and localisation (sonar sensors and CCD cameras).

Ground-based navigational methods can be further subdivided into two schemes, a Cartesian map-based and a relative sensor-based method. The map-based method requires a 2D or 3D model of its operating environment in conjunction with a dead-reckoning and motion control system with real-time localisation capabilities. The sensor-based navigation method is required to convert 3D structural data into sufficient guidance commands.

### 5.1.1 Motion measurements

#### (a) Encoders

Odometry is a very common type of motion measurement that has been widely used due to its simplicity and low cost. It uses the rotation of one or more wheels that is in contact with the ground to measure vehicle positions. This method is subjected to a variety of errors, such as wheel slippage, track width of the vehicle, and variation in the wheel rolling radius. There is a tendency for large cumulative errors over time giving rise to a rapid loss in positional accuracy (Hague et al., 2000). Borenstein (1996) developed several measures to compensate or reduce systematic odometric errors for laboratory robot vehicles. Considering the nature of agricultural environments such as rough soil surface, undulating terrain, and the adhesive properties of moist soil, odometric measurements for in-field vehicle positioning can give rise to large errors. Some researchers (Hague et al., 2000) have suggested that motion measurements should not be taken from the driving wheels but from separate odometry wheels to reduce the impact of slip. This is somewhat impractical as the odometry wheels must be in contact with the ground and collinear with the driving wheels. This needs complex mechanical arrangements.

Dead-reckoning is a simple technique employed by mobile robots for localization, that integrates the rotation and translation of the contact wheel(s) to calculate the robot's Cartesian coordinates. This method, however, is unreliable because of the accumulation of errors over time as mentioned earlier, thus many researchers believe it is unsuitable for real-world applications. However, Yamauchi (1996) claimed that although dead-reckoning was insufficiently precise for low-level navigation, it provided general information with respect to the robot's position. The author applied the concept of evidence grid (a region in space divided into Cartesian grid) on a Nomad 200 robot (Nomadic Technologies, California, USA) to compensate for errors arriving from dead-reckoning due to slip. The results indicated that this technique produced more accurate position estimation. One major advantage of this method was that it withstood transient changes, such as people walking past the robot and rearrangement of obstacles.

**(b) Gyroscopes**

Gyroscopes and accelerometers are two types of inertial sensors that can be used as alternatives for vehicle positioning as oppose to wheel encoders for dead-reckoning. They have been used in various vehicle applications (Schonberg et al., 1996). There are three main types of gyroscope, spinning mass gyros, optical gyros, and vibrating gyros. Spinning mass gyros typically has a mass spinning steadily with a free moveable axis (gimbal). When the gyroscope is tilted, it causes precession, which is the motion orthogonal to the direction of the tilt sensed on the rotating mass axis, so that the angle moved is obtained. In an optical gyroscope, a laser ray is reflected continuously within an enclosure. The enclosure is allowed to rotate and the duration between the time of laser emittance to reception is calculated. Fibre-Optic gyroscope (FOG) is an example of an optical gyroscope that uses a coil of optical fibre. Vibrating gyroscopes consist of a vibrating element (vibrating resonator chip) that uses the Coriolis effect of the sensor element to sense the speed of rotation. The vibrating element causes secondary vibration orthogonal to the original vibrating direction. The rate of turn is related to the secondary vibration.

In Japan, an automated six-row rice transplanter was developed using RTK-GPS for precise positioning and FOG sensors to measure direction (Nagasaka et al., 2004). The RTK-GPS achieved 20 mm precision at 10 Hz data output rate, and the FOG sensors were used to maintain the transplanter inclination. The inclination of the transplanter due to changes in terrain affected the position of the machine, but was corrected using the FOG sensor data. FOG drift was compensated by comparing the lateral direction deviation calculated using the yaw angle and the machine speed with the deviation calculated from the GPS data.

An encoder-based navigation system is simply a dead-reckoning method, which provides a position, heading, linear and angular velocity for mobile robots. An autonomous mobile robot (AMR) navigation system using a differential encoder and a gyroscope was developed by Park et al. (1997). They used an AUTOGYRO (FOG) on a Labmate mobile robot. Their experimental results demonstrated that the position of the robot was much more precise by combining the encoder and gyroscope readings. The systematic errors arising from the differential encoder and the stochastic errors from the gyroscope were modeled in the navigation filter. An indirect Kalman filter was used instead of an extended Kalman filter, since the indirect method estimated both the errors from the encoder and gyroscope, and by augmenting both of these errors the filter was able to correct them both.

### 5.1.2 Localisation using landmarks

A landmark is a localized physical feature that a robot can sense and use to estimate its position in relation to a known map or a reference frame. The landmark-based navigation is based on detection and recognition of these features in a robot's environment in order to navigate the robot to a specific location. These features can be defined using natural or artificial landmarks. They can also be generated from a known map or geometric models. Natural landmarks are flexible, easy to use and cheap. However, it is often sparse and unstable. Artificial landmarks can be predefined and tend to reduce the complexity of the localization algorithms. Both natural and artificial landmark-based localization require high performance sensors to accurately detect them.

Landmarks can be detected using several sensing mechanisms, including mechanical sensors, sonar sensors, laser scanners, radar, and CCD cameras. These sensing methods have their pros and cons. Mechanical sensors were one of the first types of sensors used to detect plant rows, soil furrows, and artificial landmarks such as rails and buried cables, to navigate agricultural field machines. This is a cheap method but accuracy and reliability was very poor. Laser scanners and sonar sensors are also relatively cheap and easy to implement, but they suffer from noise interference, beam spreading and scattering. However, localization using laser scanners to identify artificial landmarks is a promising technique in terms of performance and cost. This technology usually involves the measurement of the bearing of artificial landmarks relative to each other. The position of a robot can be computed using two distinctive techniques: triangulation and Kalman filtering algorithm (Skewis and Lumesly, 1994; Durrant-Whyte, 1996). Visual detection of landmarks using cameras is a rapidly advancing technique (Mata et al., 2003; Hayet et al., 2003; Sala et al., 2004) due to its accuracy and reliability. Visual recognition refers to the determination of the position and orientation of a physical landmark from an image projection of that landmark. This problem poses lots of challenges in practical applications because of uncontrolled illumination, distances, and view angles to the landmarks, as well as requirement for resources to store and process the image data. The visual detection of landmarks has been based on several fundamental concepts, such as the detection and extraction of *contour segmentation, 2D line segmentation, vertical lines, intersecting lines, edges,* and *vanishing points*.

Triangulation is a well-known technique for identifying location by means of bearings from fixed points in an environment. Data from sensors generally have some degree of uncertainty and contains noise. There are a variety of ways in which the problem of position uncertainty has been dealt with by different authors. This includes using environment where the points look identical (Sugihara, 1988), environments using distinguishable landmarks (Sutherland and Thompson, 1993), and a statistical approach (Leonard and Whyte, 1991). If one considers a mobile robot that uses a map to navigate in an environment that contain landmarks, and sensors that can measure angles, the robot can employ the following algorithm to find its location:

a) identify the landmarks in its environment
b) locate the landmarks on the map
c) measure the bearing of the landmarks with respect to each other
d) calculate its position

Once a), b) and c) are accomplished successfully, the robot's position can be computed. This approach has a unique problem, when the robot and the three landmarks form a circle or they all lie on a line. In practice, however, there are two main problems, discrepancies in the angles measured and misidentified landmarks. Another built in problem is if the map itself has some errors. Betke and Gurvits (1997) tested their triangulation and linear position estimation approach for estimating position and orientation using a mobile robot equipped with a camera that pointed upwards onto a reflective ball that acted like a mirror of the surroundings. Their findings revealed that the main advantages of the linear position estimation approach against the triangulation method were a) the algorithm used linear equations to represent the positions of landmarks, b) the position estimate was very close to the actual robot's position and orientation and c) large outlying errors in some readings were found. Two outstanding features of the work were that the algorithm developed used noisy input data and the landmarks were represented by complex numbers.

In practice, cheap and reliable sensing methods are the preferred way. But there is often a trade-off between costs and accuracy. Dead-reckoning is unreliable and results in accumulated error over long period of time. The fusion of two or more sensors to predict position and orientation of a mobile robot is common, although this method requires complex algorithms and models to represent the errors created by the sensors. A practical and reliable landmark-based navigation system will help to promote in-field robots.

## 5.2 OBJECTIVES

The primary objective of this study was to develop a landmark localisation technique based on a LRF and artificial landmarks, and to develop an algorithm to improve position estimation for navigational purposes.

Specific aims included:

- To test the linear and angular displacement accuracy of the mobile robot using the onboard gyroscope and wheel encoders.
- To test the effectiveness of the combination of the gyroscope and dead-reckoning measurements using a Kalman filter algorithm in compensating errors due to slip.
- And finally, to develop a landmark localisation technique using a LRF and artificial landmarks, and to compare it against the inertial measurement methods.

## 5.3 MATERIALS AND METHODS

### 5.3.1 Robot platform

The automatic navigation system was developed for a Pioneer 3 All Terrain (P3-AT) robot (ActivMedia Robotics, New Hampshire, USA). The robot was equipped with two internal sensors, namely a gyroscope chip (ADXRS300, Analog Devices, Massachusetts, USA) and two wheel encoders used for dead-reckoning. Two external sensors included a LRF (SICK LMS200, Waldkirch, Germany) and a ring of eight sonar sensors. The body of the robot was made of aluminium with dimensions of 500 mm(L)×490 mm(W)×260 mm(H) and a body clearance of 80 mm. The total weight was 12 kg including three 12 V batteries. It was four-wheel driven (215 mm diameter drive wheels), employing a skid steering approach for turning. The wheel motors used a 66:1 gear ratios and contained two 100-ticks wheel encoders. The maximum translational speed that could be achieved was 4.32 km/h and the maximum traversable slope was 40% grade.

The P3-AT robot was also equipped with a 850 MHz single-board computer (VSBC8, VersaLogic Corporation, Oregon, USA), and a 18 MHz microcontroller (H8S/237, Hitachi, Japan). The microcontroller managed all the low-level details of the robot's control and operation, including motion, heading and odometry. The single-board computer was used to acquire and process LRF data in real-time and implement the automatic navigation algorithms.

### 5.3.2 Internal positioning sensors

A brief explanation on the principle of operation of the two internal position sensors is discussed in the following section.

#### 5.3.2.1 Gyroscope – Principle of operation

The gyroscope (ADXRS300, Analog Devices, Massachusetts, USA) used was an angular rate sensor built using proprietary MEMS®surface micro-machining process by Analog Devices. It had a dynamic range of ±300 °/s and a sensitivity of 5 mV/°/s. The bandwidth was 0.04 kHz and the temperature range was - 40°C to 85°C. The power supply required a voltage of 4.75 V and a current of 6 mA.

The principle of operation was based on vibrating gyroscopes. It consisted of two polysilicon sensing structures each containing a dither frame, which was electrostatically driven to resonance. A Coriolis force was produced when there was a rotation about the z-axis, normal to the plane of the chip. The Coriolis force displaced the sensing structures perpendicular to the vibration. At the edges of the sensing structures were a series of capacitive pickoff structures, which detected the Coriolis motion. The resulting signal was then amplified and fed to a series of demodulation stages to produce the rate signal output. The dual sensor design resisted external g-forces and vibration. The ADXRS300 also consisted of a temperature sensor for temperature coefficient calibration plus a precision voltage reference.

The purpose of the gyroscope was to compensate for large errors due to wheel slip, lost of wheel contact on the ground, and other factors like skid steering and gearbox play. The gyroscope was connected to an analog to digital (A/D) input channel of the microcontroller. Gyroscope measurements were of 10-bit integers. When the robot was stationary the measurement was centered around 512. Measurements less than 512 represented counter-clockwise direction and measurements above 512 represented clockwise direction. The measurements were dependant on the gyroscope's temperature and drift. In the event of drift, the gyro auto-calibrated itself to the centre range of 512, if the robot was stationary for one second. A stationary position meant the rotational and translational velocities were less than 1°/sec and 1 mm/sec respectively for a period of 1 s. Another condition for auto calibration to take place was that the gyro's current reading must be within 0.5% of the average readings. A block diagram of the gyro is shown in Appendix A with the resonator loop in place that was responsible for the Coriolis effect.

#### 5.3.2.2 Wheel Encoders

The P3-AT robot had four wheel motors and two wheel encoders. The two left-wheel and two right-wheels were mechanically coupled (with a belt) respectively, thus the encoders produce two distinct speeds, one for the left pair and the other for the right pair. The dead-reckoning concept used consisted of simple geometric equations using odometric data to calculate the position of the robot in comparison to its starting point.

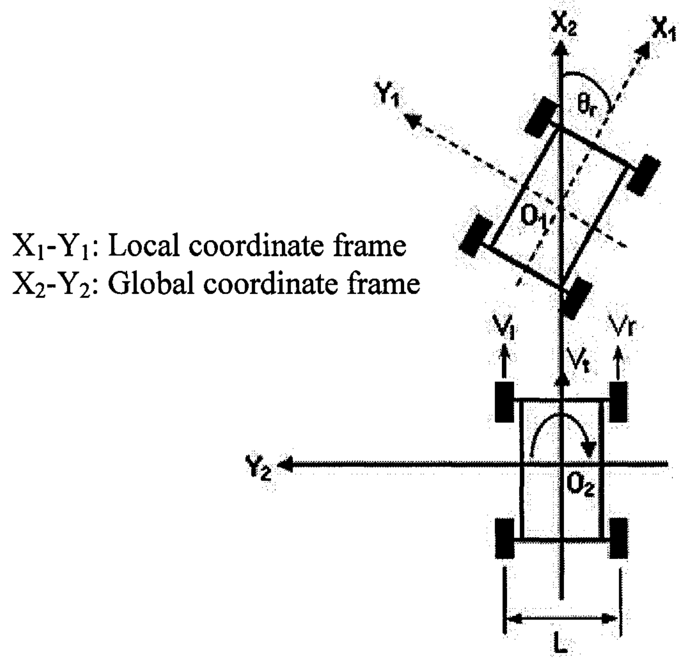



**Figure 5.1: Dead-reckoning concept**

Figure 5.1 shows the robot after undergoing a translational and rotational movement. $X_2$-$Y_2$ was considered as a global coordinate frame and $X_1$-$Y_1$ as a local coordinate frame of the robot. $V_l$ and $V_r$ represented the left and right encoder velocities respectively, whilst $V_t$ was the total translational velocity. The robot's heading was represented as an angle $\theta_r$, with an angular velocity as $\dot{\theta}$. $V_t$ was resolved into two components of velocity, a horizontal and vertical components. The Cartesian coordinates of displacement was obtained by an integration of the horizontal and vertical components of translational velocity over time. The distance measurement was computed using equations 5.1 to 5.6.

$$V_t = \frac{V_r + V_l}{2} \tag{5.1}$$

$$V_r = V_t + \frac{L\dot{\theta}_r}{2} \tag{5.2}$$

$$V_l = V_t - \frac{L\dot{\theta}_r}{2} \tag{5.3}$$

$$\dot{\theta}_r = \frac{V_r - V_l}{L} \qquad (5.4)$$

$$x = \int_0^t v_x \, dt \qquad (5.5)$$

$$y = \int_0^t v_y \, dt \qquad (5.6)$$

Where:

$V_l$ – left encoder velocity, m/s

$V_r$ – right encoder velocity, m/s

L – distance between left and right encoder, m

$\theta_r$ – angle between $X_1$ and $X_2$ axis (robot's heading), rad

$\dot{\theta}_r$ – angular velocity in the *x-y* plane, rad/s

x – vertical displacement, m

y – horizontal displacement, m

## 5.3.3. LRF and sonar sensors

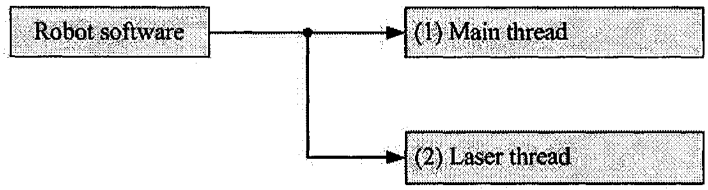
The LRF operated on the principle of "time of flight" measurements. The field of view was 180°, and the maximum measurement range was 150 m. The angular resolution was selected at 0.5° by means of the software. The statistical error of the LRF was ± 15 mm for distances from 1 m to 8 m, and ± 40 mm for distances from 8 m to 20 m. The advantages of using a LRF as a navigation and localization sensor were its high scanning frequency (75 Hz) and the transfer of measurement data to the single-board computer occurred in real-time.

A total of eight sonar sensors were mounted on the robot. The sensitivity and range of the sonar sensors were adjusted using a gain control adjuster located under the robot's panel. An appropriate setting was used depending on the environment in which the robot was working. Low-gain settings reduced the robot's ability to detect small obstacles, therefore, was useful when operating in noisy environments, or on uneven or highly reflective surface. As the sensitivity, the sonar sensors were able to detect smaller obstacles and objects at a distance further away. Each sonar sensor were multiplexed and the acquisition rate of the array was 25 Hz (40 ms per sonar array).

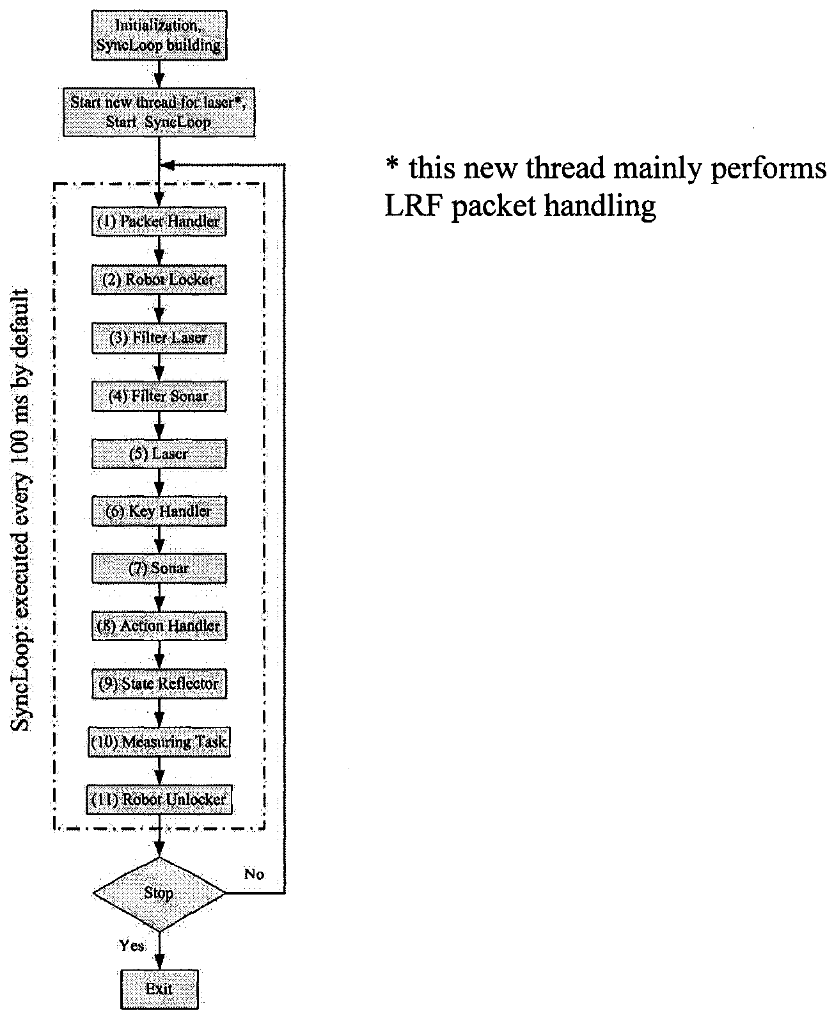
### 5.3.4 Software design

The overall software structure consisted of two parts, a low-level control operating system running on the microcontroller, and an autonomous navigation and localization program running on the single-board computer. ActivMedia Robotics Operating System (AROS) was running on the microcontroller (server) and ActivMedia Robotics Interface Application (ARIA) was running on the single-board computer (client). The microcontroller communicated with the single-board computer using special client-server communication packet protocols over a RS-232 interface. The server information packet (SIP) contained information about the sonar sensors measurement, gyroscope measurements, and odomteric measurements. ARIA was used to decode and interpret the information in the SIP and to send appropriate commands to the wheel motors for motion control.

The ARIA program developed comprised of two threads; a LRF thread, which was used primarily for receiving and processing data from the LRF and a main thread, which performed a variety of functions for planning, and intelligent, purposeful control of the robot's platform and its components (Figure 5.2 (a)). The main thread consisted of eleven functions and was executed every 100 ms (Figure 5.2 (b)). This included functions for interpreting and analyzing the encoder and gyroscope readings encoded in the SIPs from the microcontroller by the Packet handler (1). The key handler (6) was used for interfacing the keyboard with ARIA for setting and changing modes of operations of the robot by pressing an appropriate key on the keyboard. Various modes such as automatic and manual operation were developed for testing and evaluating the robot. Appropriate keys on the keyboard were defined to allow the selection of the operation mode. The keyboard also served as a means of entering file names for data storage and for setting target coordinates manually for the robot. The function action handler (8) executed the tasks of obstacle detection, obstacle avoidance and all motion such as forward, reverse, and turns. The motion commands sent by the single-board computer was used to control the mobility of the robot, e.g. to set individual wheel speeds, or coordinated translational and rotational velocities; to change the robot's absolute or relative heading; to move the robot to a specific distance or; to stop the robot.




**(a)**




**(b)**

**Figure 5.2: Block diagram of the software structure**

With respect to localization using artificial landmarks, the LRF was used to detect two landmarks (two cylindrical poles) and the appropriate equations were developed to compute the Cartesian coordinates of the robot relative to the poles, and finally these coordinates were transformed unto the global coordinate frame.

Results from the wheel encoders, gyroscope and LRF were collected every 100 ms and written to a data file for analysis.

## Kalman filter

When using more than one sensor modality to obtain position orientation, such as a gyroscope and wheel encoders, an algorithm is needed to combine the readings to produce an accurate estimation of position. Typically a Kalman filter is used for this purpose. There are two main types of Kalman filters, the extended Kalman filter (also known as the direct Kalman filter) and indirect Kalman filter. Significant work has been done in developing error models for sensors and their integration using the filters. Barshan and Durant-Whyte (1995) developed error models for inertial sensors and included them in an Extended Kalman filter for estimating the position and orientation of a moving robot vehicle. Vaganay et al. (1993) used two accelerometers and three gyroscopes, then fused accelerometric measurements of gravity and integration of differential equations (from the robot's attitude and angular velocity measured by the gyroscope) using a extended Kalman filter, resulting in a system that was very sensitive and accurate.

The extended Kalman filter (EKF) does not correct the systematic encoder errors and gyroscope errors mutually, but merely minimize their errors using independent models. The indirect Kalman filter estimates the systematic errors of the encoders and the stochastic errors of the gyroscope exclusively and is fed back into the navigation system. In this case, the filter is not in the navigation loop but in the case of the EKF the filter is in the main navigation loop. Since the filter is out of the navigation loop with respect to the indirect approach, the encoder information will be available even if the filter fails or if it not used (Park et al., 1997).

An indirect Kalman filter was used in this research. The Kalman filter was outside the main navigation loop. The encoder information was available even if the Kalman filter was not used. A block diagram is shown in Figure 5.3.

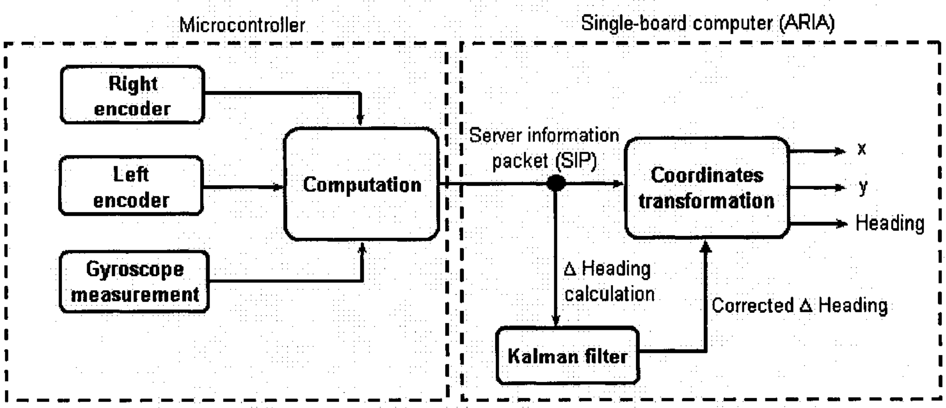



**Figure 5.3: Block diagram showing the Kalman Filter in the navigation system**

The microcontroller computed the position and orientation of the robot with data from the gyroscope and right and left wheel encoders. The computed position information were then encoded in separate SIPs, and sent to the single-board computer. The single-board computer decoded the SIPs to obtain an estimate of the robot's position and then transformed the readings into a global coordinate frame. The coordinate transformation included transformation of the heading and Cartesian coordinates from a local coordinate frame to a global coordinate frame. When desired, the software was also used to compute only the measurements from the wheel encoders, or the gyroscope, or a combination of the wheel encoders and gyroscope. Considering the latter case using a combination of the gyroscope and encoder readings, a simple Kalman filter was used. The single-board computer used the information in the relevant SIPs to calculate the final pose of the robot.

### 5.3.5 System tests

Several experiments were conducted with the P3-AT robot to study the accuracy of the dead-reckoning method and the effectiveness of the gyroscope. All experiments were carried out indoor on a concrete surface on which a grid was outlined of size 6 m × 6 m consisting of cells 1 m × 1m. With the marked grid, actual distances could be measured quickly and easily. Two cylindrical poles with diameters of 80 mm and 125 mm were used for the landmark localisation technique. In some instances the robot was programmed to automatically drive to a goal and in other cases it was manually driven to achieve an X,Y coordinate with a specific final heading.

#### 5.3.5.1 Test 1: Accuracy of linear distance and angular displacement measurements.

Specific objectives included:

- To determine the effectiveness of combining a gyroscope and dead-reckoning measurements for position estimation when a robot moved in the X direction only without any change in its Y coordinate;
- To determine the effectiveness of combining a gyroscope and dead-reckoning measurements for heading when a robot was rotated on a specified spot.

*Approach*: The robot was programmed to automatically drive from it's starting position (0,0,0 in the global coordinate frame) for desired distances of 1 m to 10 m with a 1 m increment. Distances of 1 m interval were marked on the floor. The robot's actual position at each incremental value was noted by making a mark on the ground corresponding to the centre between the two front wheels of the robot (the wheel centre was used as a benchmark). This distance was then measured and recorded. The test was conducted two times, first without using the gyroscope (dead-reckoning only), and secondly with dead-reckoning using gyroscope correction. The measured values for each method were recorded. The test was repeated three times and the average of the measurements were used for analysis.

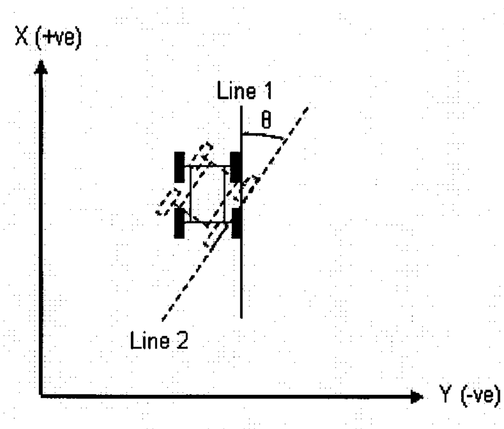



**Figure 5.4: Angular test layout**

The robot was also programmed to rotate on a spot at desired angles. In order to measure the actual angle turned by the robot, a straight line was drawn on the ground (line 1, Figure 5.4), and after the robot had rotated a second line was drawn (line 2, Figure 5.4). The angle turned (θ) was measured, as illustrated in Figure 5.4. The angular test was conducted in a similar manner as the linear displacement test with dead-reckoning only and dead-reckoning with gyroscope correction.

The percentage error was computed to evaluate the dead-reckoning and dead-reckoning with gyroscope correction methods, using the expression below:

$$\text{Percentage error} = \left| \frac{\text{Desired distance (angle)} - \text{Measured distance (angle)}}{\text{Desired distance (angle)}} \right| \times 100\%$$

### 5.3.5.2 Test 2: Accuracy of position determination on a grid layout.

Specific objective included:

- To determine the effectiveness of the combined gyroscope and dead-reckoning measurements with changes in X,Y coordinates and heading angles

*Approach*: The robot was manually driven (approximately 1.08 km/h) to selected points on the grid. In this case, the centre of the robot (from where all positions were defined) was used as a benchmark. At its final position the centre of the robot was aligned to the intersection of the grid lines. The centre of the robot was used as a benchmark instead of the centre of the front wheel so as to eliminate the need for any coordinate transformation of its position. A visible mark on the robot's centre allowed for an easy alignment of its centre. The final heading for grid locations 1 to 8 was manually set at +90° and the final heading for grid location 9 to 16 was set at -90°. This was done to maintain consistent motion pattern for all the selected points. When the robot reached its final position, the wheel encoder readings (dead-reckoning only), and the dead-reckoning with gyroscope correction readings were recorded. The percentage error between the actual position and measured positions were computed. The measured position refers to the wheel encoder measurement, and the combined wheel encoder and gyroscope measurement. The percentage error was computed as follows:

$$\text{Percentage error} = \left|\frac{\text{Actual distance} - \text{Measured distance}}{\text{Actual distance}}\right| \times 100\%$$

### 5.3.5.3 Test 3: Gyroscope performance during maximum slip.

Specific objective included:

- To evaluate the performance of the gyroscope during maximum slip (100%)

*Approach*: The robot was suspended so that the four wheels were not in contact with the ground. It was then "driven" manually (approximately 1.08 km/h) for a period of time so that sufficient data was collected and stored in a data file. The robot was "driven" in the forward, reverse, left and right turn directions. In this case, the wheel encoders could not sense that the robot was not physically moving when suspended, but the gyroscope was able to sense this condition. Three types of data were collected, namely, wheel encoder readings, gyroscope readings, and the combined gyroscope and wheel encoders reading. The data was collected every cycle of 100 ms time duration. Data from every 3 cycles were analyzed, and a total of 97 cycles were required as it was the equivalent of a 360° rotation.

### 5.3.5.4 Test 4: Landmark localization method for position determination in a grid layout

Specific objectives included:

- To develop a landmark localization technique using a LRF and artificial landmarks
- To evaluate and compare the localization technique developed in (1) with the fusion of sensory readings from the encoders and gyroscope

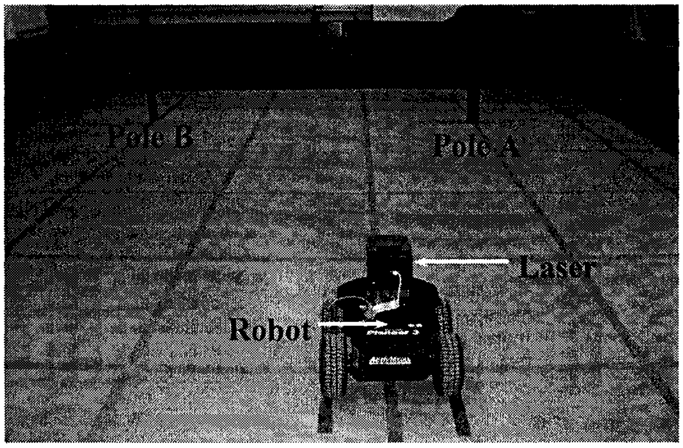



**Figure 5.5: Test set up for the landmark localization method**

*Approach*: Figure 5.5 shows the experimental setup. Two poles (used as artificial landmarks) were placed at coordinates of (4000, 2000) and (4000, -1000), respectively, with reference to the robot's global coordinate frame.

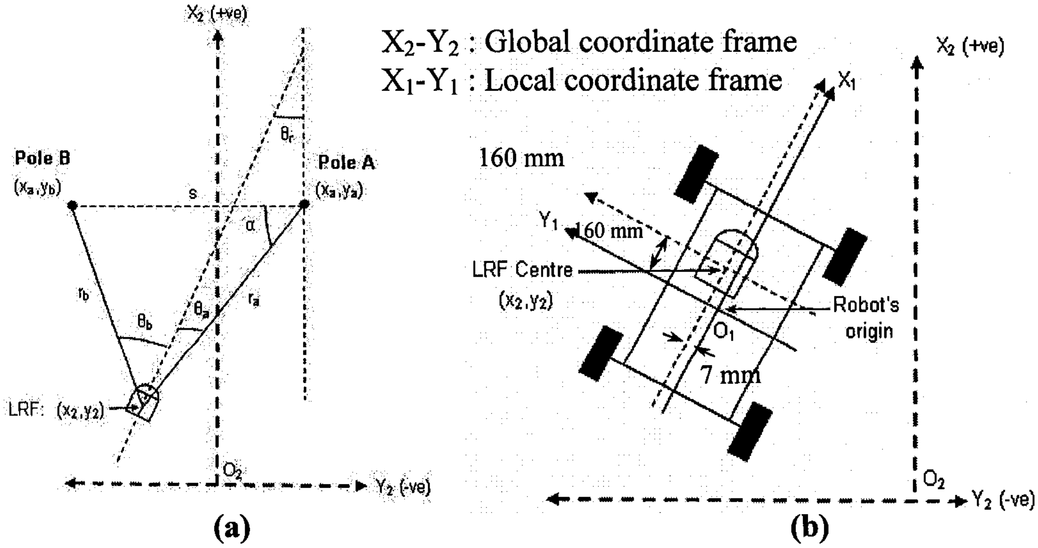



**Figure 5.6: Landmark localization technique**

The separation of the landmarks represented as $s$ was set at 3 m in the tests. The distance $s$ could be varied depending on the work space available and the environment in which the robot was working. The distances between the landmarks and the centre of the LRF were defined as $r_a$ and $r_b$ for Pole A and Pole B respectively. $\theta_a$ and $\theta_b$ were defined as the angles between $X_l$ axis of the local coordinate frame and the "line of sight" of the poles. In Figure 5.6 (a), $\theta_a$ was defined as a negative angle, and $\theta_b$ as a positive angle. Objects detected to the left were assigned a positive value and to the right negative values. The variables $r_a$, $r_b$, $\theta_a$ and $\theta_b$ were all known from the LRF data (Figure 5.6 (a)). In order to compute the position of the LRF ($x_2$, $y_2$) with respect to the poles the following equations were derived and used in the landmark localization algorithms.

$$(x_2 - x_a)^2 + (y_2 - y_b)^2 = r_b^{\,2} \tag{5.7}$$

$$(x_2 - x_a)^2 + (y_2 - y_a)^2 = r_a^{\,2} \tag{5.8}$$

$$(2y_2 - y_a - y_b) \times (y_b - y_a) = r_a^{\,2} - r_b^{\,2} \tag{5.9}$$

$$y_2 = \frac{r_a^{\,2} - r_b^{\,2}}{2(y_b - y_a)} + \frac{y_a + y_b}{2} \tag{5.10}$$

$$x_2 = x_a \pm \sqrt{r_a^{\,2} - (y_2 - y_a^{\,2})} \tag{5.11}$$

The heading angle $\theta_r$ was calculated using the following expressions:

$$\frac{\sin\alpha}{r_b} = \frac{\sin(\theta_b - \theta_a)}{s} \tag{5.12}$$

$$\sin\alpha = r_b \times \frac{\sin(\theta_b - \theta_a)}{s} \tag{5.13}$$

$$\theta_r = 180 - (90 - \theta_a + \alpha) \tag{5.14}$$

$$\theta_r = 90 + \theta_a - \alpha \tag{5.15}$$

Figure 5.6 (b) shows the position of the LRF relative to the robot's local coordinate frame. The centre of the LRF was located at 160 mm, 7 mm with respect to the local coordinate frame of the robot. The position of the centre of the LRF was subsequently transformed to the robot's origin to obtain the position of the robot in the global coordinate frame.

The robot was again driven to specific locations on the grid and the encoder, gyroscope and landmark localization system readings were all recorded and analyzed.

### Statistical analysis

An analysis of variance (ANOVA) at the 0.05 significant level was conducted to investigate if there was any significant difference between the different methods of position estimation. The error between the measured values, from encoder readings, the dead-reckoning method with gyroscope correction and the landmark localization method, and desired values were computed and subsequently used in the ANOVA test. A commercial software package, Origin 7.5 (OriginLab Corporation, Northampton, U.S.A.), was used to conduct the ANOVA.

## 5.4 RESULTS AND DISCUSSION

Results of the four tests are presented in this section. Various approaches for position estimation are compared and discussed.

### 5.4.1 Test 1: Accuracy of linear distance and angular displacement measurements.

When the robot was driven, the encoder readings were used to calculate its position and orientation. The gyroscope was also activated and changes in headings were recorded. A Kalman filter was used to incorporate the encoder values and gyroscope readings to compute the final position of the robot. The detailed results are showed in Appendix B.

Linear displacement

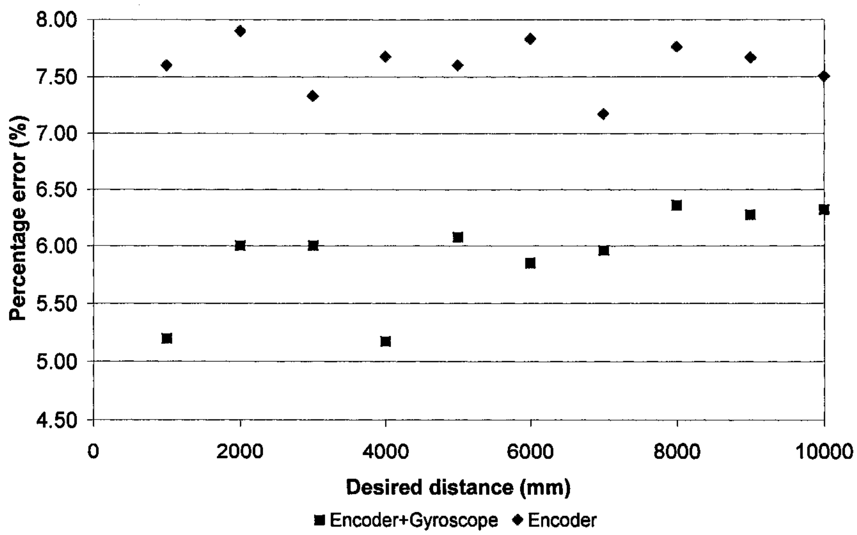



**Figure 5.7: Percentage error between desired and measured distances for linear displacement**

In Figure 5.7, the percentage errors between the desired distances and encoder readings varied from a minimum value of 7.17% to a maximum of 7.90% without gyroscope correction. The root mean square error (RMSE) between the desired distances and encoder reading was calculated as 0.47 m. The percentage error between the desired distances and the encoder readings with gyroscope correction varied from a minimum of 5.18% to a maximum of 6.36%. The RMSE was calculated as 0.38 m. Results from the ANOVA test is shown in Table 5.1.

**Table 5.1: Statistical analysis of linear displacement**

| Method | N | Mean | SD | SE | *p*-value |
|---|---|---|---|---|---|
| Encoder | 10 | 417.4 | 229.18 | 72.47 | 0.39175 |
| Encoder+Gyroscope | 10 | 333.7 | 196.08 | 62.01 | |

Note: N : Number of measurements

SD : Standard deviation

SE : Standard error

Results from the ANOVA test showed that at the 0.05 significant level, there was no significant difference between the two methods, since the *p-value* was greater than 0.05 (Table 5.1). The dead-reckoning system would normally have larger errors over a long period of time. With respect to short distances, the dead-reckoning method was reliable. Preliminary tests on the robot had indicated that the percentage errors obtained from the dead-reckoning method varied from 20% to 30% for a period of time ranging between 5-15 minutes. This was much larger than the percentage error obtained in this test and can be attributed to the fact that there was no change in the robot's heading and the experiment was not conducted over a long period of time.

Angular displacement

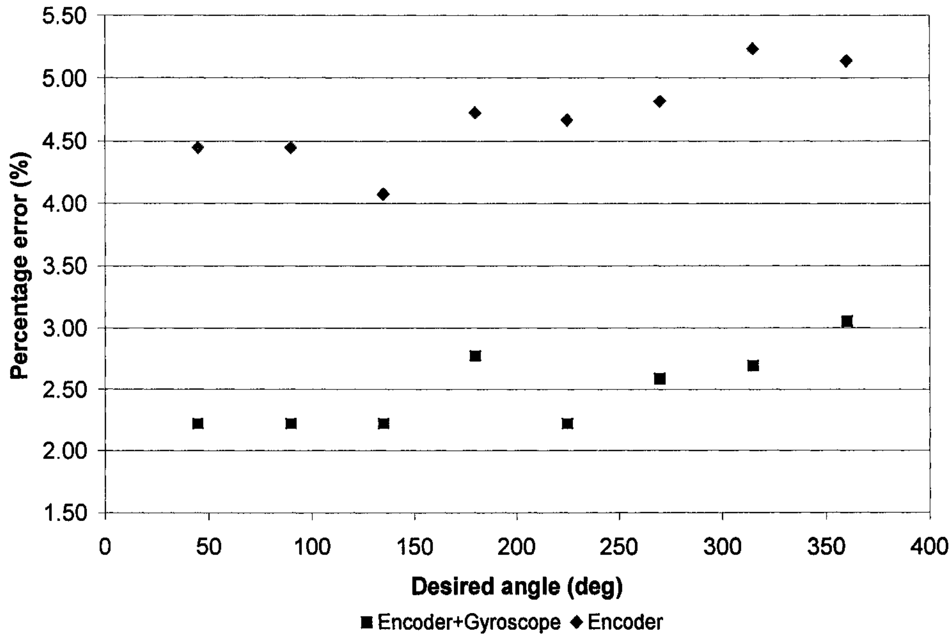



**Figure 5.8: Percentage error between desired and measured angles**

The result from the angular displacement test is shown in Figure 5.8. The RMSE between the desired angle and the measured angle using dead-reckoning with gyroscope correction was calculated to be 6.19°. However, the RMSE between the desired angle and the angle calculated from the encoder dead-reckoning approach was found to be 11.22°. Figure 5.8 illustrates that the percentage errors using the combination of the two sensors were much lower than those with just the encoder readings. Subsequently, an ANOVA was carried out to determine whether there was any significant difference between the encoder readings and encoder with gyroscope correction readings.

**Table 5.2: Statistical analysis of angular displacement**

| Method | N | Mean | SD | SE | P value |
|---|---|---|---|---|---|
| Encoder | 8 | 9.69 | 6.04 | 2.14 | 0.09573 |
| Encoder+Gyroscope | 8 | 5.31 | 3.39 | 1.20 | |

Note: N : Number of measurements

SD : Standard deviation

SE : Standard error

A *p-value* of 0.09573 was obtained from the analysis, indicating that there was no significant difference between the two methods with respect to angular displacement.

## 5.4.2 Test 2: Accuracy of position determination on a grid layout.

This experiment was conducted on a grid layout for the P3-AT robot with changes in both heading and X-Y coordinates. The robot started from position (0,0) and traveled to 19 locations. Grid position 1 to 16 were all set at a fix X coordinate value of 4000 mm and the heading was set at +90 for locations 1-8 and -90 for locations 9-16.

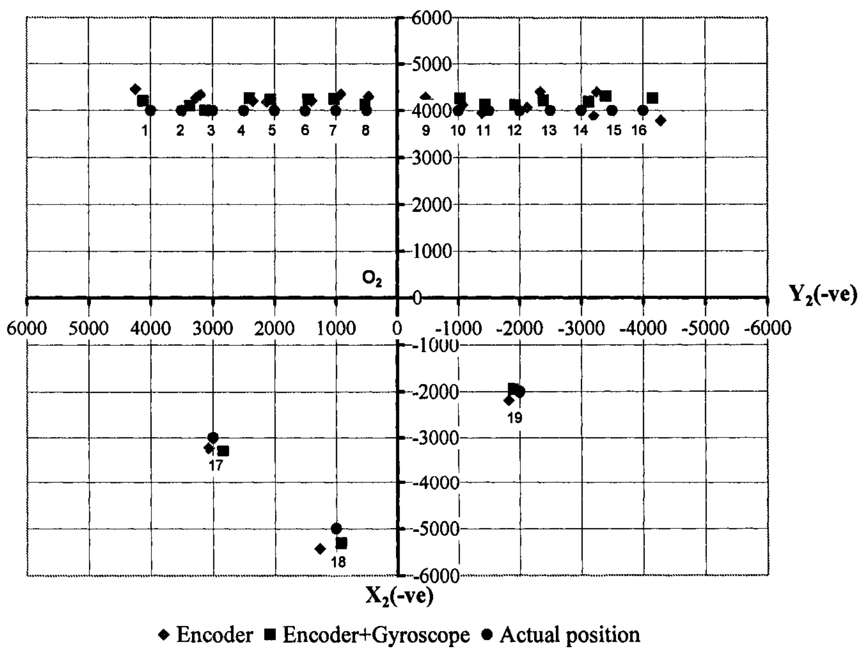



**Figure 5.9: Comparison among robot's actual position, position from the encoder, and position from encoder with gyroscope correction (units in mm)**

The test results are shown Figure 5.9, on a global coordinate frame ($X_2$,$Y_2$). The data obtained from dead-reckoning with gyroscope correction were in closer proximity to the actual position of the robot. A detail of all the results obtained is shown in Appendix C.

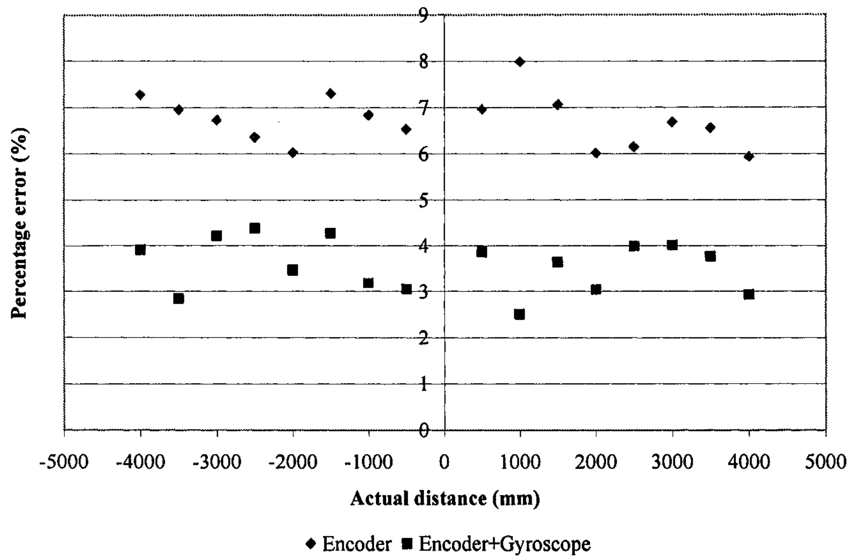



**Figure 5.10: Percentage error between measured and actual distances on grid layout (1-16)**

Figure 5.10 shows the percentage errors for the gird locations 1-16. The encoder readings when compared to the actual position of the robot had a percentage error that varied between 5.94% and 7.98% (RMSE = 1.02 m) whilst the position estimation using dead-reckoning with gyroscope correction ranged from 2.51% to 4.39% (RMSE = 0.98 m). In section 5.4.1 the RMSE for the encoder reading and dead-reckoning with gyroscope correction were computed as 0.47 m and 0.38 m respectively. An ANOVA was then carried out to ascertain whether the two methods made any significant difference of position estimation on a grid layout. On a grid layout there were changes in the X,Y coordinate as well as the heading. The results are shown in Table 5.3 and 5.4.

Table 5.3: Statistical analysis of X coordinate on a grid layout

| Method | N | Mean | SD | SE | P value |
|---|---|---|---|---|---|
| Encoder | 20 | 450.55 | 942.35 | 210.72 | 0.8570 |
| Encoder+Gyroscope | 20 | 394.59 | 925.19 | 206.88 | |

Table 5.4: Statistical analysis of Y coordinate on a grid layout

| Method | N | Mean | SD | SE | P value |
|---|---|---|---|---|---|
| Encoder | 20 | 146.08 | 83.42 | 18.65 | 0.0738 |
| Encoder+Gyroscope | 20 | 82.79 | 47.19 | 10.55 | |

Note: N : Number of measurements

SD : Standard deviation

SE : Standard error

Results from the ANOVA revealed that there no significant difference between the two methods with regards to the X and Y coordinate (*P value* > 0.05, Table 5.3 and Table 5.4).

### 5.4.3 Test 3: Gyroscope performance during maximum slip.

This test was conducted at an average rotational speed of 1.08 km/h with the wheels suspended off the ground, whilst the robot was driven manually.

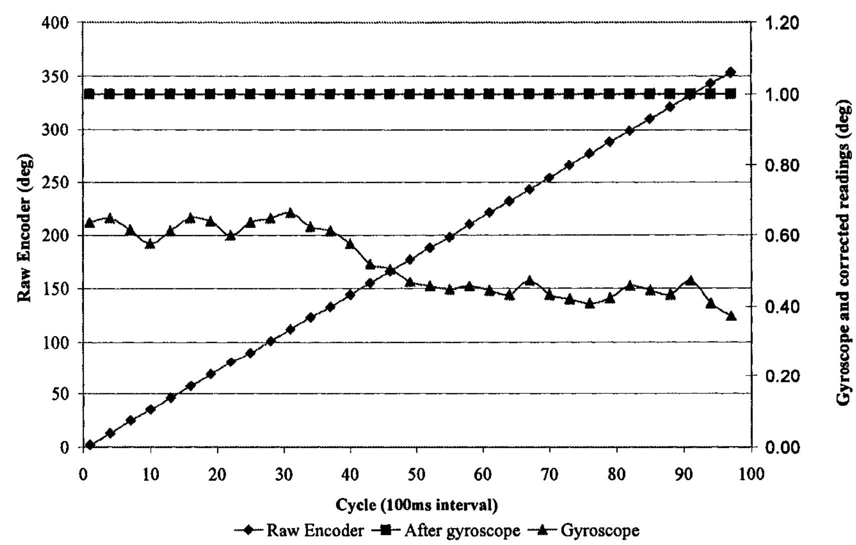



Figure 5.11: Gyroscope effect

As time increased, the encoders recorded a change in heading calculated from its dead-reckoning approach. The change in heading was quite consistent, resulting in a straight line since the rotational velocity was kept constant. The gyroscope readings fluctuated between 0.4° and 0.6° (Figure 5.11). This may have been caused by a drift due to temperature changes. Although the gyroscope had a temperature sensor to compensate for the drift, other factors such as vibration and the accuracy of the gyroscope could be attributed to this fluctuation in readings. When the encoder and gyroscope readings were fused together the resulting heading was computed as 1°. This was constant for the data examined during this time period of the experiment.

### 5.4.4 Test 4: Landmark localization method for position estimation in a grid layout

In this test the robot was manually driven to fixed locations on the grid as show in Figure 5.12. This was very much similar to the experiment described in section 5.3.3.2 with the exception that the system now comprised of a landmark localization component to ascertain position and orientation. Eighteen grid locations were selected, and the X,Y coordinates and heading angles were all recorded.

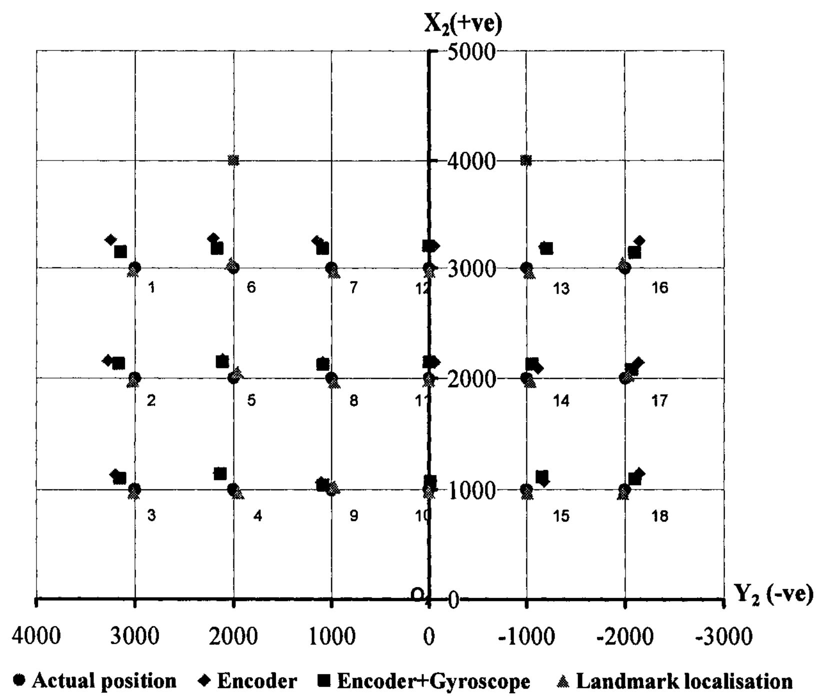



**Figure 5.12: Results for landmark localization technique (units in mm)**

Figure 5.12 illustrates the final X and Y coordinates of the robot on the grid determined by the encoder, encoder with gyroscope correction, and the landmark localization measurements, respectively. The landmark localization technique lay in the closest proximity to the actual position of the robot. The detail of all the results obtained is shown in Appendix D.

**Table 5.5: Statistical analysis of X coordinate using landmark localization technique**

| Method | N | Mean | SD | SE | *p*-value |
|---|---|---|---|---|---|
| Encoder | 18 | 163.44 | 65.05 | 15.33 | $3.75\times10^{-11}$ |
| Encoder+Gyroscope | 18 | 128.89 | 44.37 | 10.46 | |
| Landmark localization | 18 | 32.17 | 8.40 | 1.98 | |

**Table 5.6: Statistical analysis of Y coordinate using landmark localization technique**

| Method | N | Mean | SD | SE | *p*-value |
|---|---|---|---|---|---|
| Encoder | 18 | 140.44 | 65.15 | 15.36 | |
| Encoder+Gyroscope | 18 | 102.06 | 58.93 | 13.89 | $3.48\times10^{-8}$ |
| Landmark localization | 18 | 23.28 | 12.05 | 2.84 | |

Note: N : Number of measurements

SD : Standard deviation

SE : Standard error

Analysis of all the data from this test showed that there was a significant increase in the accuracy of the position computation using the landmark localization technique. The ANOVA tests demonstrated that there was a significance difference across the three methods, as the *P values* for both the X and Y coordinates were less than the 0.05 significant level (Tables 5.5 and 5.6). In addition, a "Turkey test" was carried out to investigate if there was any significant difference between means of individual methods and the results are shown in Table 5.7.

**Table 5.7: Significant difference between the position estimation methods**

| | **Encoder** | **Encoder + Gyroscope** | **Landmark localization** |
|---|---|---|---|
| **Encoder** | - | No | Yes |
| **Encoder + Gyroscope** | No | - | Yes |
| **Landmark localization** | Yes | Yes | - |

From Table 5.7 it can be seen that there was no significant difference between the encoder readings and the encoder with gyroscope corrected method. However, there were significant differences between the landmark localization method and the two other methods.

The RMSE between the actual and measured values for the landmark localization method for the X and Y coordinates were 0.033 m and 0.026 m respectively. The RMSE for the X and Y coordinates with the dead-reckoning only method was 0.175 m and 0.154 m respectively, whilst with the dead-reckoning with gyroscope correction was 0.135 m and 0.117 m respectively. This translates to within 20-35 mm accuracy for the landmark localization method with respect to the actual positions employed in this experimental layout.

**Table 5.8: Summary of RMSE between actual and measured coordinates**

| Position estimation method | RMSE/m | |
|---|---|---|
| | x | y |
| Dead-reckoning | 0.175 | 0.154 |
| Dead-reckoning with gyroscope correction | 0.135 | 0.117 |
| Artificial landmark localization | 0.033 | 0.026 |

In summary, the landmark localization method based on a LRF offered the most accurate estimation of position and orientation. The LRF was used to determine: 1) the distances and angles of the landmarks relative to the robot and 2) to compute the position and orientation of the robot in a global coordinate frame. Dead-reckoning, on the other hand, could be very reliable for short distances and environments with minimum slip. However, as time increases, the errors using only the encoders tend to accumulate. This problem is overcome by use of the localization technique, which was not affected by the conditions in which the robot was expected to work (i.e. surface undulation and slippery environments). The accuracy and resolution of the LRF (or any other localization sensor such as a CCD camera or sonar sensor) were the limiting factors that influenced the accuracy of the position estimations.

It must be noted that in this experiment the landmarks (two poles) were placed at the same X coordinate in the reference frame and the LRF on the robot had to be able to detect these two poles simultaneously in order to compute its position and orientation. If the poles were not detected, the resulting position computation produced an error. This was a setback as the motion and path of the robot would be limited and would be impractical and unsuitable for real-world applications. This disadvantage could be overcome by placing several known landmarks around the robot's working environment, thereby allowing the robot much more flexibility and path planning capabilities. The onboard gyroscope was a single axis device that detected the yaw rate. This worked satisfactorily as only heading changes were required. In events of undulating terrain, a three-axis gyroscope would be very useful, as slopes can be detected, thus speed and direction could be corrected based on information received from the gyroscope. Overturn of tractors is a very common mishap in agricultural operations resulting in loss of time, money and even harmful to the operators. In the case of an autonomous navigation system safety is of paramount importance, therefore slope detection is crucial and should be incorporated into the control system.

The use of odometric measurements for navigational purposes poses a few challenges in agricultural operations due to a number of factors. First of all, the mere nature of dead-reckoning technique requires a wheel to be in contact with the ground and any changes in the rolling circumference would induce some errors in the position of the vehicle. Considering the conditions in a typical agricultural field with high soil moisture content and the adhesive properties of soil, it is quite natural that at some times soil will stick to the tires on the wheel and in effect increase the rolling radius. The fusion therefore of sensory data seems like a feasible solution as was employed in the navigation system. A Kalman filter can be used to fuse data from the encoders and a gyroscope to improve the accuracy of the position and orientation estimate of the vehicle.

## 5.5 CONCLUSIONS

An indirect Kalman filter was used to fuse dead-reckoning and gyroscope measurements to produce an estimate of a mobile robot's position and orientation. Results from the test showed that the improvements made by the encoder readings with gyroscope correction when compared to only dead-reckoning method was not very significant as the encoder readings tend to be reliable for short distances and also slip was minimal.

A landmark localization technique for computing position and orientation of a mobile robot was successfully developed and tested in an indoor environment. Two artificial landmarks were placed at known positions with respect to the global coordinate frame of the robot. A LRF was used to detect the positions of the two landmarks. The distances of the landmarks from the centre of the LRF were determined using range information. The angles between the line of sight of the landmarks and the $X_l$ axis of the local coordinate frame were also computed. Suitable equations were derived and included in the localization algorithm to compute the position and orientation of the robot in a global coordinate frame. The results from the tests showed that the position estimation using this technique was much more accurate when compared to the dead-reckoning and dead-reckoning with gyroscope correction approaches. The RMSE between the actual positions and the computed positions for the X and Y coordinates were found to be 0.033 m and 0.026 m respectively. This RMSE was significantly less when compared to the RMSE obtained from the dead-reckoning with gyroscope correction measurements. This was a significant difference due to the fact that the landmark localization method using the LRF was not dependent on the operating environment such as slip and undulation.

## 5.6 REFERENCES


Barshan, B., and H.F. Durant-Whyte. 1994. Orientation estimate for mobile robots using gyroscopic information. *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robot and Systems (IROS '94)*. Vol. 3, 1867-1874.

Betke, M., and L. Gurvits. 1997. Mobile robot localization using landmarks. *IEEE Transaction on Robotics and Automation*. Vol. 13 (2), 251-263.

Borenstein, J., and L. Feng. 1996. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transaction on Robotics and Automation.* Vol. 12 (6), 869-880.

D'Orazio, T., F.P. Lovergine, M. Ianigro, E. Stella, and A. Distante. 1994. Mobile robot position determination using visual landmarks. *IEEE Transactions on Industrial Electronics*. Vol. 41 (6), 654-662.

Durrant-Whyte, H.F. 1996. An autonomous guided vehicle for cargo handling applications. *International Journal of Robotics Research*. Vol. 15 (5).

Hague, T., J.A. Marchant, and N.D. Tillett. 2000. Ground based sensing system for autonomous agricultural vehicles. *Computer and Electronics in Agriculture*. Vol. 25 (1/2), 11-28.

Hayet, J.B., F. Lerasle, and M. Devy. 2003. Visual landmarks detection and recognition for mobile robot navigation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2, 313-318.

Leonard, J.J., and H.F. Durant-Whyte. 1991. Simultaneous map building and localization for an autonomous mobile robot. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 3, 1442-1447.

Mata, M., J.M. Armingol, A. De La Escalera, and M.A. Salichs. 2003. Using learned visual landmarks for intelligent topological navigation of mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, 1324-1329.

Morgan, K.E. 1958. A step towards an automatic tractor. *Farm mech*. Vol. 10 (13), 440-441

Nagasaka, Y., N. Umeda, Y. Kanetai, K. Taniwaki, and Y. Sasaki. 2004. Autonomous guidance for rice transplanting using global postioning and gyroscopes. *Computer and Electronics in Agriculture*. Vol. 43 (3), 223-234.

Park, K., H. Chung, J. Choi, and J.G. Lee. 1997. Dead-reckoning navigation for an autonomous mobile robot using differential encoder and a gyroscope. *Proceedings of the IEEE International Conference on Advanced Robotics*. 441-446.

Sala, P.L., R. Sim, A. Shokoufandeh, and S.J. Dickinson. 2004. Landmark selection for vision-based navigation. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 4, 3131-3138.

Schonberg, T., M. Ojala, J. Suomela, A. Torpo, and A. Halme. 1996. Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation. *International Journal of Syst. Sc*. Vol. 27 (8), 745-752.

Skewis, T., and V. Lumesly. 1994. Experiments with a mobile robot operating in a cluttered unknown environment. *International Journal of Robotic Systems*. Vol. 11 (4), 281-300.

Sugihara, K. 1988. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics and Image Processing*. Vol. 42, 112-129.

Sutherland, K.T., and W.B. Thompson. 1993. Inexact navigation. *Proceedings of IEEE International Conference on Robotics and Automation.* Vol. 1, 1-7.

Vaganay, J., M.J. Aldon, and A. Fourinier. 1993. Mobile robot attitude estimation by fusion of inertial data. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, 277-282.

Yamauchi, B. 1996. Mobile robot localization in dynamic environments using dead-reckoning and evidence grids. *IEEE Conference on Robotics and Automation*. Vol. 2, 1401-1406

# CONNECTING TEXT

A laser range-finder (LRF) and a ring of eight sonar sensors were used as external sensors for an autonomous navigation system described in Chapter IV. The LRF was subsequently used in Chapter V as part of an artificial landmark localization method for position and orientation estimates. The position estimates from the internal sensors, including two wheel encoders and a vibrating gyroscope, were compared with an artificial-landmark localization technique. Chapter VI further investigates the reliability of the LRF for position estimates of the robot in a dynamic state. The sensitivity of the sonar sensor was also investigated and the appropriate sensitivity setting was selected to suit the robot's operating environment. An autonomous navigation system was developed using the data from the sonar sensors and the LRF. An obstacle detection and avoidance algorithm was developed based on a few virtual regions and detector bands to navigate the robot in a real-time manner. Finally, the overall system was tested in three maze layouts.

# VI. AUTONOMOUS NAVIGATION USING SONAR SENSORS AND A LASER RANGE FINDER

## 6.1 INTRODUCTION

Robots and robotic devices are becoming more prevalent in various industries and research areas worldwide. It is important to develop systems with a high degree of autonomy and capability of operating in structured and unstructured environments. In order to make this a reality, a robotic system must be versatile and intelligent, i.e. it must have a good understanding of its work space and be able to sense objects in the real-world. A range of sensors is available to interact with the real-world and gather data about its operating environment. Sonar sensors and laser range-finders (LRFs) have been used extensively as obstacle detection and avoidance sensors. Many researchers have developed navigational algorithms that include feature tracking, collision avoidance, and localization using these two sensors.

### Sonar sensors

In general, ultrasounds have several applications, for example, in the medical field, it has been used for imaging and creating maps of the human anatomy (Husey, 1975). Robotics in particular has attracted the use of sonar sensors for distance measurements and localization of object surfaces. There are two types of sonar sensors, namely active and passive. Active sonar sensors have been used primarily in communications, navigational systems and tracking, whilst the passive types have been used in surveillance. Miller (1984) successfully developed a robotic system that used sonar sensors to determine the position of a mobile robot. The algorithm developed compared the current sonar sensor readings with a preloaded map of an environment to determine the position of the robot. An assumption with the experiment was that the map of the environment used was accurate. Elfes (1989) conducted research to create real-time maps of a robot's environment using an array of sonar sensors. The resulting two-dimensional maps were subsequently used for path planning and navigation. A robust method was used to combine the range data from the sonar sensors to cope with uncertainties and errors in the data. The robot was able to achieve complex tasks and navigate in unknown and unstructured environments. Langer and Thorpe (1992) used a land vehicle (Navlab, Carnegie Mellon University, Pittsburgh, USA) equipped with sonar sensors to track features such as a railroad track and a line of parked cars. The system built a local grid map of its surroundings, which was used for obstacle detection and feature tracking. The authors mentioned that future work would enable the vehicle to detect a parking space and subsequently park autonomously. A fuzzy logic approach was developed by Beom and Cho (2000) on a mobile robot (LCAR), equipped with sonar sensors. The fuzzy logic was based on behavior schemes. The appropriate behavior was automatically selected depending on the situation in the vicinity of the robot. The situation was detected via the sonar sensors. In order to compensate for wheel slippage and dead-reckoning errors, a single rotating sonar and two cylindrical beacons were mounted on top of the robot for localization.

There are some advantages and disadvantages when using sonar sensors. One setback of using ultrasound signals in air is the limitations of its range and data update rates due to attenuation and low speed of sound. Since most research projects have used sonar systems at low vehicle speeds, this problem has a low impact on the system performance. Sonar sensors are not confused by transparent or colored surfaces, as is the case with optical sensors. If the sonar sensors do not face the reflecting surface in a normal direction, then the reflected sound wave may not be received by the sensor. This may tend to cause some problems unless the reflecting surface is rough or comprises of edges. However, in reality many outdoor surfaces have a surface roughness and feature projections that allow the sonar sensors to detect an object. This makes sonar sensors very appealing for navigational outdoor purposes, in addition to its simplicity and low cost.

### Laser sensors

Laser sensors can be divided into three categories, namely 2D LRF, 3D LRF and flash laser. A 2D LRF scans one plane only, whilst 3D LRF scans and 'nods' thereby producing a three-dimensional image of its environment. A flash laser, on the other hand, does not have any moving parts and could only provide an image of point. It is believed that a 3D LRF is the best option for mapping and obstacle negotiation. However, due to its high cost and slow mapping ability in real-time (because of relatively slow vertical scan), the 3D LRF has been placed at a disadvantage. Researchers have implemented other methods of using 2D LRF to create 3D maps. This was done by aiming the LRF in a forward and downward direction on the front end of a mobile robot, thus creating a 3D image (Borenstein and Koren, 1991).

As researchers continue to investigate the best sensor(s) for automatic navigation, the LRF seems to stand out from the rest as a single sensor that has the capability to provide the autonomous vehicle with the necessary operating characteristics. Olivier and Ozguner (1986) developed a navigational algorithm for an intelligent vehicle with a LRF. The algorithm tested on the LRF was subdivided into four parts - region classification, path planning, path smoothing and path execution, which resulted in a fast and reliable system in an area containing lots of obstacles. The region classification section identified all dangerous obstacles in front of the vehicle, the path planning section constructed a path around the obstacles and the path smoothing ensured it was possible to execute the planned path successfully. Finally, the path execution was responsible for generating and sending the appropriate signals to the wheel motors to achieve the planned path. In other areas of research, 3D LRFs have been used for conducting tasks such as wall following, door crossing and obstacle avoidance based on reactive behavior. Information from the LRF was used for robot self-localization and local 3D map building (Montano and Asensio, 1997).

Whilst research objectives have been focused on developing sensors and algorithms for navigational purposes, work has also been done to investigate the performance of LRFs. Researchers have investigated the influence of temperature, target properties, and the effects of noise on 3D LRFs. Hoffman and Krotkov (1991) reported that changes in temperature from 17°C to 27°C produced changes in a LRF range measurements up to 3 m. Langer et al. (2000) claimed that ambiguity intervals of range measurements and range accuracy of a LRF can be overcome by selecting a suitable modulation frequency. Most importantly however, a characterization study was conducted by Ye and Borenstein (2002) on a 2D LRF (SICK LMS200, Waldkirch, Germany). Their findings revealed that target surface properties and the incidence angle of the laser beam were sources of measurement errors. These errors at its maximum were 17 mm. Another source of error not related to its characteristics was loss of synchronization at its highest data transfer rate, but the authors did find a work around that problem.

The focus of this research was to design and implement an autonomous navigation system on a mobile robot using data from a LRF and ring of sonar sensors. This study differs from previous research in that no knowledge of the operating environment was required for navigation, so the robot could navigate in an unknown environment. Previous researchers have used preloaded map of an environment, with which current sensor readings were compared to obtain position and orientation of obstacles (Mazl and Preucil, 2000; Adachi et al., 2003). In this study, data from a LRF was analyzed instantaneously and the robot's position was obtained using an artificial landmark-based localization technique. The autonomous system was designed to allow the robot to detect and avoid obstacles in real-time, and to achieve a preset goal. Virtual regions and detector bands were established to determine the magnitude and direction of rotation. The system was designed with navigational efficiency being a fundamental requirement, so that a goal can be achieved in the most efficient and effective manner, without repetition of previously traversed paths. Three main robotic states were defined to allow the robot to adjust to different conditions in its operating environment.

## 6.2 OBJECTIVES

The main objective of this study was to develop an algorithm to effectively navigate a test robot through different maze layouts using a LRF and a ring of eight sonar sensors. Specific objectives included:

- To investigate the sensitivity of the sonar sensor array

- To determine the effectiveness of a landmark localization technique for position estimation in a dynamic state
- To develop an algorithm for obstacle detection and avoidance
- To develop an algorithm to determine the magnitude and direction of rotation, and side clearance safety for a mobile robot in a structured environment.
- To test the overall system in three structured maze layouts.

# 6.3 MATERIALS AND METHODS

## 6.3.1 Hardware design

A Pioneer 3 All Terrain robot (P3-AT, ActivMedia Robotics, New Hampshire, USA) was used as a test platform. The robot was equipped with a SICK LMS200 LRF (SICK, Waldkirch, Germany) and a ring of eight sonar sensors. The P3-AT robot also consisted of two wheel encoders and a vibrating gyroscope chip (ADXRS300, Analog Devices, Massachusetts, USA). Odometric measurements from the wheel encoders were fused with the gyroscope measurements to produce an enhanced position and orientation estimate using a Kalman filter.

A single-board computer and microcontroller were the central controllers for autonomous operations of the robot. The single-board computer was used primarily for collecting and processing data from the LRF and performed the automatic navigation and obstacle avoidance algorithm. The microcontroller controlled the wheel motors, acquired data from the sonar sensors, wheel encoders, gyroscope, and prepared data packets to be sent to the single-board computer for interpretation and analysis.

The LRF is a non-contact measurement system with high scanning frequency and measurement resolution making it an ideal sensor for localization, obstacle detection and path planning functions. The LRF operated on the principle of "time of flight" measurement. A laser beam was transmitted and deflected off a rotating mirror (frequency of 75 Hz). The reflected beam was registered in the receiver and distance of an object was calculated based upon the time of flight of the laser beam and the speed of light. The SICK LMS200 LRF was approximately 185 mm(L)×155 mm(W)×156 mm(H) with a weight of 4.5 kg. The resolution was 10 mm and the maximum range detected by the LRF was 150 m. The LRF scanned 180° at 0.5° or 1°. An angular resolution of 0.5° was selected for this study, as it provided higher accuracy for mapping capabilities.

A ring of eight sonar sensors was mounted in the front of the robot at a height of approximately 220 mm above the ground. They provided 180° coverage in front of the robot. Their coordinates and orientation was presented in section 4.3.2.3. The sonar sensors were multiplexed and the acquisition rate of the array was fixed at 25 Hz (40 milliseconds per sonar sensor).

### 6.3.2 Software design

The major navigational algorithm was developed using ActivMedia Robotics Interface Applications (ARIA) and ran on the single-board computer. The designed navigational software enabled the robot to navigate autonomously to a predefined goal and detect and avoid obstacles in different maze layouts.

Eight sonar sensors and a LRF were used as range devices for detecting and locating obstacles. Sonar sensors were installed at a lower height than the LRF, so that lower obstacles, which could not be detected by the LRF were detected by the sonar sensors. Due to the fact that normally a sonar sensor could only see objects right ahead of it the eight sonar sensors served as auxiliary object detection devices because of their limited detection range. On the other hand, the LRF outputted 361 data points evenly distributed over a scan range of 180° providing complete and reliable object detection and data distribution. The LRF was, therefore, used as the main obstacle detection sensor.

In the autonomous navigational algorithm, data from both sensors were used in deciding whether an area was clear or not. “Virtual regions” and “detection bands” were defined relative to the robot. If neither the LRF nor the sonar sensors “saw” any objects within these regions and detector bands, the path of the robot was identified as “clear”.

To navigate in a maze autonomously, an intelligent robot needs to identify a path towards an exit or a goal point, avoid obstacles, prevent collision on its way, and record previous paths to avoid retracing. The robot's motion consisted of both rotational and translational displacement. To ease the control, the rotational speed was controlled separately from the linear translation. When the robot needed to turn, it stopped translational movement and started to turn, until the rotational movement was completed. No translational movement occurred until the rotational speed was reduced to zero. With this design, the robot always took a straight path, which greatly simplified the motion control and map recording. For example, only two points were needed to record a path, the starting point and the end point.

### 6.3.2.1 Principle of the obstacle detection and avoidance algorithm

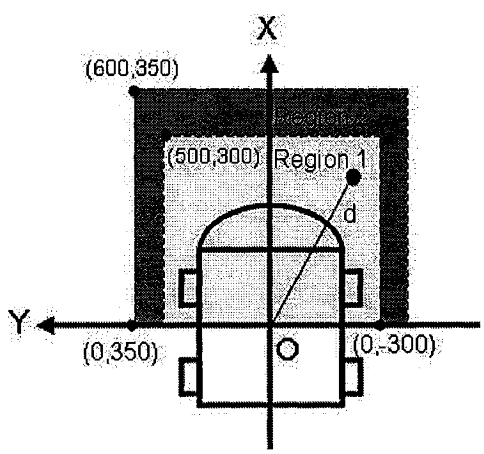



**Figure 6.1: Regions (1 & 2) for obstacle detection and avoidance**

Two main regions (Figure 6.1) in front of the robot were defined as Regions 1&2. Region 1 (inner boundary) was used to define an area in front of the robot to detect the presence of an obstacle. Region 1 was 600 mm wide and 500 mm long, whilst Region 2 (outer boundary) was 700 mm wide and 600 mm long. The boundaries were defined by virtue of the robot's dimensions and operating environment. The sizes of the maze, including the separation of walls and the distances of obstacles in relation to the walls were the factors contributing to the selection of the stated boundary conditions. Region 1 was slightly wider than the width of the P3-AT robot to allow for enough clearance. The lengths of Regions 1 and 2 were selected to cater for the length and width of the robot during turning maneuvers, in a constrained environment.

The regions were defined for both the LRF and sonar sensors. If an obstacle was detected in Region 1 either by the LRF or a sonar sensor, the translational velocity was set to zero and the robot initiated its rotational velocity either clockwise or counter-clockwise to avoid the obstacle. Region 2, on the other hand, was used to define an area that dictated an angle at which the robot rotated to avoid the obstacle detected in Region 1. Once the robot started rotating, the positions of the obstacle changed relative to the robot and the two regions. The robot rotated until the object appeared out of Region 2. The rotation angle was determined primarily, by the distant separating Regions 1 and 2. In the event that the angle of rotation was too small or too large, the region boundaries could be adjusted as desired.

### 6.3.2.2 Definition of the rotation clearance safety

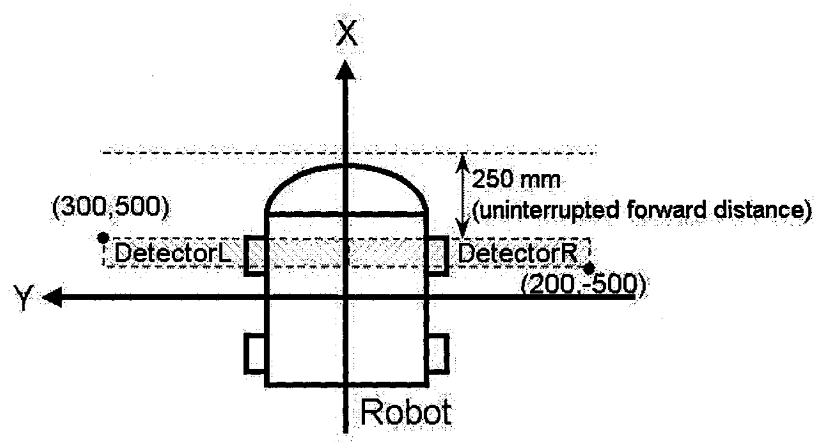



**Figure 6.2: Detector bands (L & R) for rotational clearance safety**

In order to make the robot sensitive to its environment two pair of virtual detector bands, DetectorL and DetectorR were set up. Their main purpose was to check for an uninterrupted forward clearance of 250 mm at the left and right sides of the robot during turning manoeuvres. An uninterrupted forward distance indicated that no obstacle was detected in a forward distance of 250 mm. The detector bands were 500 mm long and 100 mm wide, and centred 250 mm from the Y axis, measuring from the centre of the bands. Since the LRF was located at 160 mm, 7 mm from the robot's origin, DetectorL&R were placed in front of the LRF at 250 mm on the X axis, so that they were in the field of view of the LRF (Figure 6.2). Objects at either side of the robot tended to impede the motion of the robot during turning. Sufficient clearance on both sides of the robot was essential to compensate for the length and width of the robot when rotating. The purpose of these bands was *not* to check for direction of rotation but rather to check for safety in clearance when rotating.

### 6.3.2.3 Definition of rotation manoeuvres

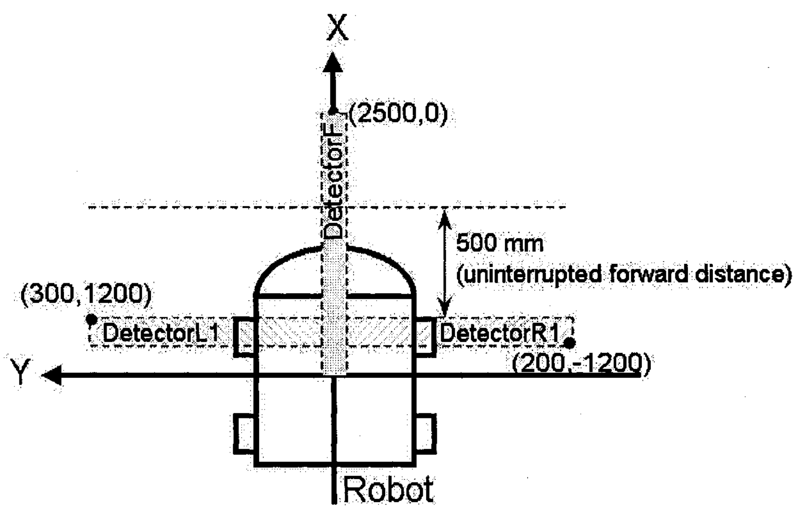



**Figure 6.3: Detector bands (L1, R1 & F)**

Three additional detector bands, DetectorL1, DetectorR1 and DetectorF were also established on the robot. These bands were much longer than DetectorR and DetectorL. DetectorL1&R1 were 1200 mm in length, 100 mm wide, and located at a point 250 mm from the Y axis, measuring from the centre of the bands. DetectorF was 2500 mm long, 100 mm wide, and symmetrical about the X axis (Figure 6.3). The main purpose of these larger defined areas was to continuously monitor and log the position of objects relative to the robot. These bands were very critical in aiding to determine the heading of the robot once an obstacle was detected. The two side bands constantly swept both sides of the robot and checked for an uninterrupted minimum forward clearance distance of 500 mm along the X axis. A minimum value of 500 mm was selected as the forward distance because that was sufficient clearance for the width of the P3-AT robot, whilst 1200 mm was chosen as the length of the bands because the width of the maze in which the robot was operating did not exceed 1.8 m. Therefore, 1200 mm was long enough to detect an uninterrupted forward distance of 500 mm in the experimental mazes. The bands dictated which direction (left or right) to turn once an obstacle was detected. For example, if an obstacle was detected at any point in time and a minimum forward clearance distance of 500 was found on the right (as detected by DetectorR1), the robot would have turned to the right. DetectorF, on the other hand, was used to check for previously traversed paths or intersection of a current path with a previous path.

Four possible scenarios therefore existed with regards to clearance: 1) clearance was found to the right, 2) clearance was found to the left, 3) clearance on both sides and, 4) no clearance on either side. In the case of scenario 1 and 2, the robot turned to either the right or left, depending where the clearance was detected. Considering scenario 3, with clearance on both sides, the algorithm computed the distance of the clearance area from the robot and selected the closest clearance to the robot as its turning direction. Scenario 4 was a special state whereby DetectorR1 and DetectorL1 did not detect any clearance that was sufficiently wide enough for the robot to traverse, given that an obstacle was detected in front of the robot. In this case, the direction of turn was not dictated by these two detector bands, instead, the direction of turn depended on the position of the obstacle relative to the robot. If the obstacle was detected to the left, the robot turned to the right, keeping the obstacle on the left, and vice versa.

#### 6.3.2.4 Design of the autonomous navigation system

The aims of the software were to enable the robot to achieve predefined goals, whilst detecting and avoiding obstacles in real-time. Obstacle detection and avoidance mechanisms were developed and implemented as virtual regions in front of the robot. Detector and rotational clearance safety bands were also established to determine rotational direction and safety threshold, respectively. Three main robotic adjustment states were developed, namely, *stop state*, *goal alignment state*, and *rotational-translation state*. The robot states were used to enable autonomous navigation in maze layout, and to cope with varying circumstances.

The flowchart of the software design for the robot navigating autonomously in the mazes is shown in Figure 6.4. The program initially started with declaration and initialization of predefined variables, which was subsequently followed by a series of calculations. These calculations included:

- The distance of objects detected in Regions1&2 (Figure 6.1). The range of an object relative to the centre of the robot was computed and stored in a variable. In the event that there was not any obstacle present, a preset value of 3000 mm was stored in the variable. This was the maximum preset range of the sonar sensors.
- The side clearance safety threshold. The distances of objects detected either by DetectorL or DetectorR were computed and stored in a variable. These two areas were continuously monitored for an uninterrupted forward distance of 250 mm.
- Direction of rotation. DetectorL1 and DetectorR1 continuously monitored both sides of the robot for a forward uninterrupted distance of 500 mm. This information was updated every cycle and logged to a variable.
- The angle of turn. Two main types of angle were declared and calculated, *goal alignment angle* and *obstacle avoidance angle*. The *goal alignment angle* refers to the angle between the robot's current heading and the position of the goal, and the *obstacle avoidance angle* refers to the angle the robot rotated to avoid an obstacle based on the principle explained in section 6.3.2.1 (Principle of obstacle detection and avoidance). The angles of objects detected in Regions1&2 were also computed. These angles were defined as the angle between the line of sight of the objects and robot's current heading.

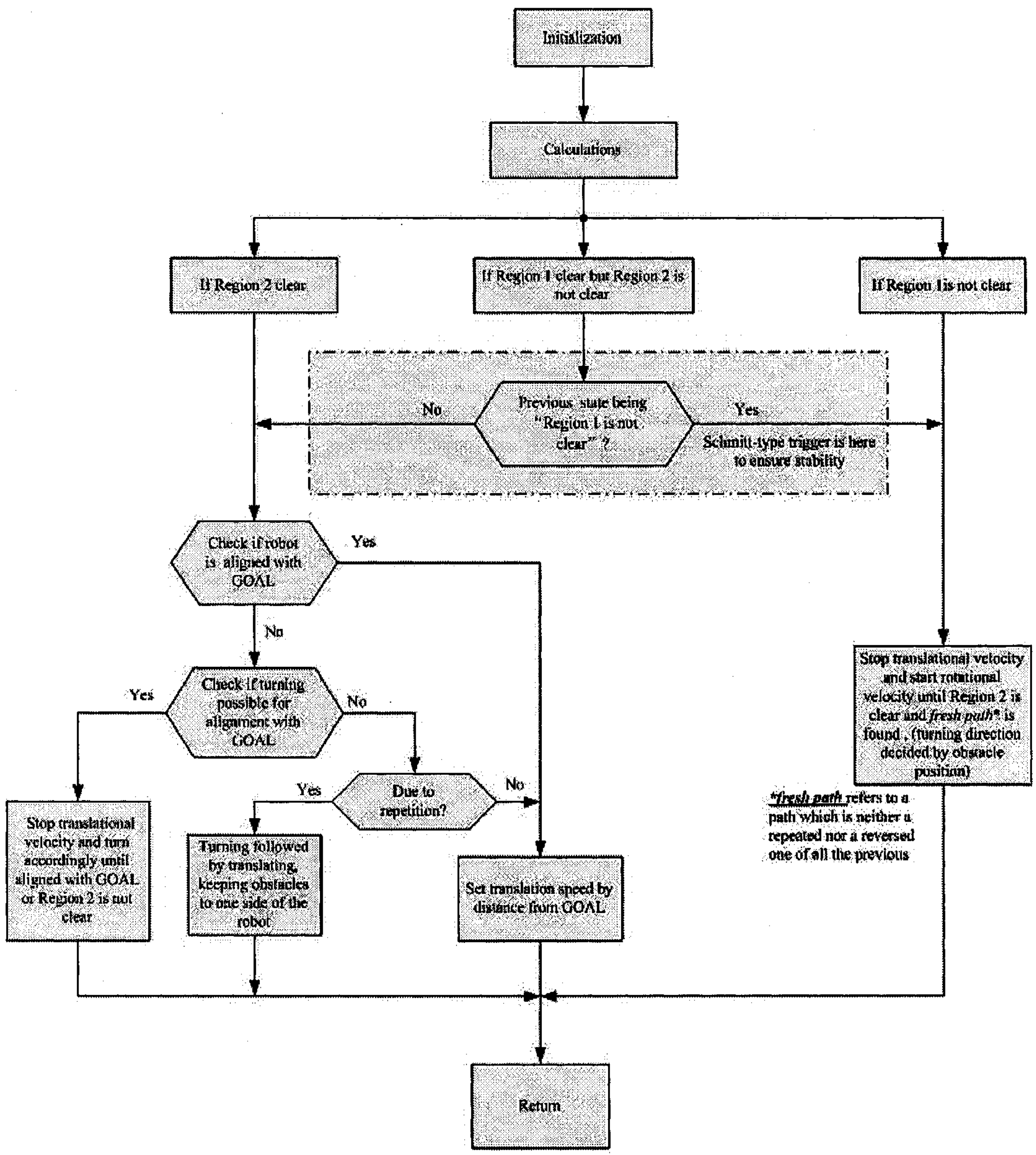



**Figure 6.4: Schematic diagram of the autonomous navigation software**

- Other computations included *boolean* expressions for intersection of paths traversed by the robot, path previously traveled or whether the current path has been repeated.

As the robot moved in its environment, the situation in the vicinity of the robot changed. The robot had to continuously make decisions based on its present situation. For this reason, three fundamental robotic *adjustment states* were defined to cope with changing situations, namely:

- *Stop state*: Translational and rotational velocity was set to zero
- *Goal Alignment state*: An appropriate heading was selected based on the position of the goal and the position of the robot. The robot rotated until it was in alignment with a final goal
- *Rotation-translational state*: Rotation followed by translation

*Goal alignment state*

Initially, the robot was programmed to travel to a predefined goal, by inputting the Cartesian coordinate of that goal relative to the global coordinate frame. The algorithm computed the *goal alignment angle*, based on the current heading of the robot in the local coordinate frame and the position of the goal. Once the *goal alignment angle* was computed, the translational velocity was set to zero and the robot rotated to the desired heading for alignment with the goal. Two outcomes were possible after alignment with the goal, either the robot successfully traveled to the goal or an obstacle was present in its path.

The robot turned either to the left or right to align itself with the goal in an efficient manner. A right or left turn was determined by the direction of the *goal alignment angle*. Clockwise angles were designated as negative angles and counter-clockwise angles as positive angles. In order for the robot to align itself with the final goal, two sets of conditions must exist:

1) Left turn: Regions1&2 were clear – ensuring no obstacles were present in these two regions; and DetectorL was clear with sufficient clearance on the left side; and the *goal alignment angle* was greater than zero (+ve).
2) Right turn: Regions1&2 were clear – no obstacles were present in these two regions; DetectorR was clear with sufficient clearance on the right side; and the *goal alignment angle* was less than zero (-ve).

If either conditions 1) or 2) were true, the software checked whether the planned path to align the robot with the goal was intersecting or a repeat of any of the previous paths traversed already. This precautionary measure was taken to minimize unnecessary motion of the robot and optimize its performance and navigational capabilities. A repetition of a previous path was not acceptable as it would be inefficient to traverse that path because the previous path obviously was not a successful path to achieve a goal. If a previous path was not intersected on this routine check, the robot set an appropriate rotational velocity followed by a translational velocity until it was in alignment with the goal. Achievement of a goal could be accomplished if there were no objects between the current position and the goal, or maybe unsuccessful if there were other objects ahead to prevent achievement of the goal.

As stated before, Region 1 was used to detect if there were any obstacles ahead of the robot in the *goal alignment state*. If an object was detected, the operating state of the robot was changed to the *stop state*.

*Stop state*

In this state, the translational velocity was set to zero, followed by a rotational velocity. The angle of rotation was determined based on the principle discussed in section 6.3.2.1 (Principle of obstacle detection and avoidance), i.e. until Region1 was clear. And the direction of rotation was based on information from DetectorsL1&R1. As DetectorL1&R1 were continuously monitoring the sides of the robot, any aperture of width greater than 500 mm was detected and logged. If DetectorL1 detected an aperture, the robot rotated to the left, and kept the obstacle to the right. As mentioned in section 6.3.2.3 (Definition of rotation manoeuvres) four possible scenarios were possible. The principle of keeping the obstacle to the left or right was part of the obstacle detection and avoidance algorithm.

*Rotation-translational state*

This state consisted of two predefined parameters, namely, a *maximum translational distance* of 1 m and a *maximum rotational angle* of 120°. Whilst the robot was keeping an obstacle to the left or right sides of the robot, the sides of the robot were monitored either by DetectorL or DetectorR, respectively. It must be noted the position of the obstacle with respect to the robot dictated which detector band was in effect. If the obstacle was kept to the left only DetectorL was activated in the navigational algorithm. If the obstacle was on the right, only DetectorR was activated. The use of the detector bands in this case for rotational clearance safety had significant implications in practice:

- It meant that the edge of a wall could be sensed
- It meant that a way out of an enclosed area could be found

When DetectorL or DetectorR detected enough safety distance, the *maximum rotational angle* was brought into effect, and the robot rotated to the left or right depending on whether the active band detected enough rotational safety distance. However, a full 120° was not achieved in practice, but rather acted as a maximum limit. As the robot rotated Region 1 checked for obstacles. Rotation was stopped when an obstacle was detected by Region 1. If no obstacle was detected the *maximum rotational angle* was the limit. Rotation was followed by a *maximum translational distance* of 1m. Again, Region1 had a higher priority, meaning, if an obstacle was detected in Region 1 before translating the full 1 m, the robot was brought to a halt. After the *maximum translational distance* was executed the robot entered a *goal alignment state*.

A detailed flowchart of the software design of the autonomous navigation system is shown in Figure 6.5. It illustrates the adjustment operating states of the robot and shows the predefined conditions to which satisfy a state.

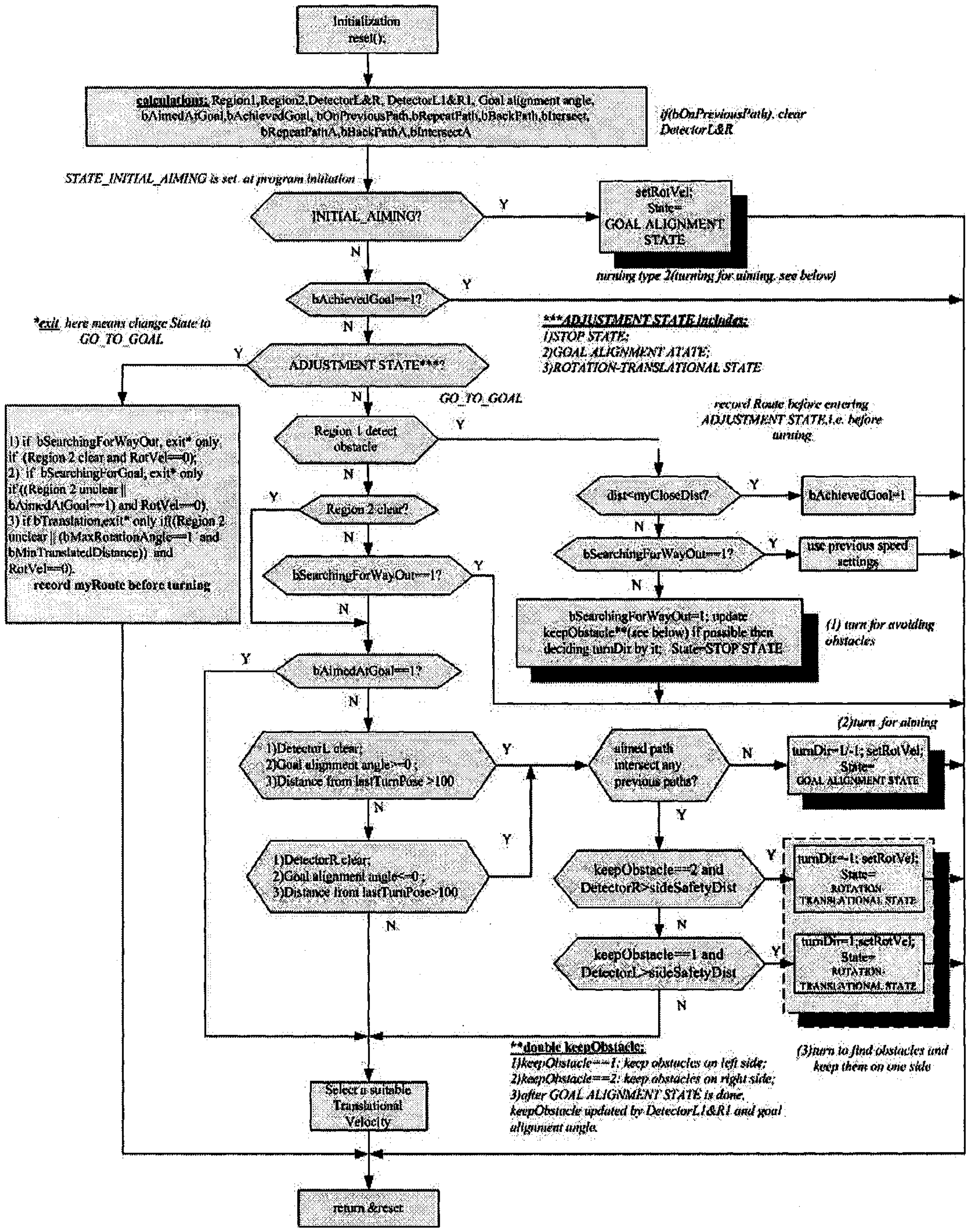



**Figure 6.5: Flowchart of the autonomous navigation system**

## 6.3.3 Experimental design

### 6.3.3.1 Tests for the field of view (FOV) of the sonar sensors

The specific objective was:

- To investigate the sensitivity of the sonar array, and to select the best sensitivity setting for the autonomous navigational system.

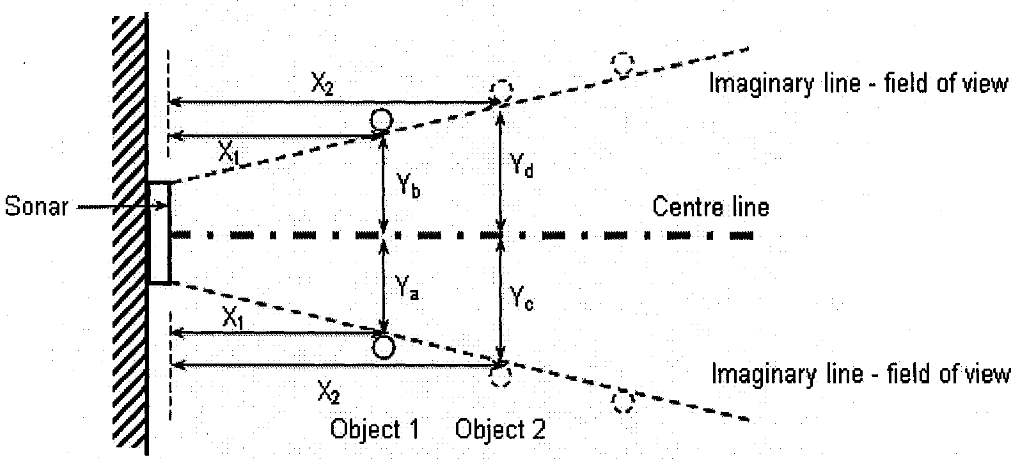



**Figure 6.6: Measurement of sonar sensor sensitivity**

Where:

$X_1$ - horizontal distance of Object 1 from sonar sensor

$X_2$ - horizontal distance of Object 2 from sonar sensor

$Y_a$ - vertical distance of Object 1 from centre line

$Y_b$ - vertical distance of Object 1 from centre line

$Y_c$ - vertical distance of Object 2 from centre line

$Y_d$ - vertical distance of Object 2 from centre line

One sonar sensor was placed flat against a wall and activated. Objects were placed at distances $X_1$ and $Y_a$ from the sensor until the sensor just about detected the object. The object was gradually shifted around by trial and error until the point at which the sonar sensor just about detected it was achieved. The object was made of wood and cylindrical in shape (5 mm in diameter). A computer program was developed to provide a visual indication as to whether the object was detected or not, as the sensor readings were displayed on the screen. This procedure was repeated for $Y_b$ as shown in Figure 6.6. The distance $X_1$ was varied from 250 mm to 3000 mm, and the corresponding Y coordinate was recorded. These measurements were repeated three times for different sensitivity settings of the sonar sensor. Two sonar sensors were tested separately and the average of the measurements were recorded. High, medium and low sensitivity settings were tested, and finally the actual FOV of the sonar sensor was generated.

#### 6.3.3.2 Tests for position determination capabilities of a LRF for a moving robot

The specific objective was:

- To investigate the reliability of the localization method described in section 5.3.4.4 (Landmark localization method for position determination on a grid layout), whilst the robot was in motion.

Section 5.3.4.4(Landmark localization method for position determination on a grid layout) described a localization approach when the robot was stationary. This test was conducted to ascertain the reliability of the localization method when the robot was in motion.

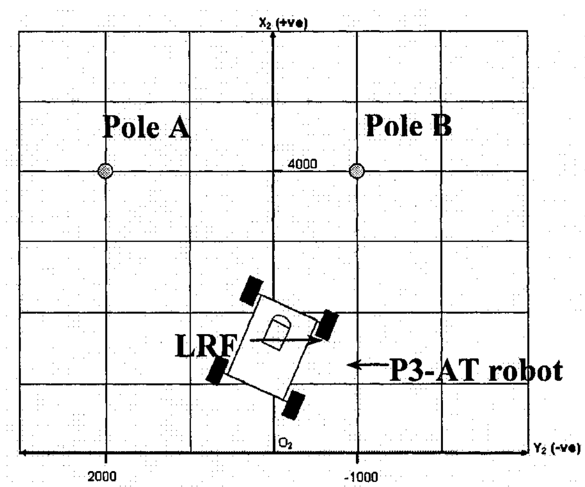
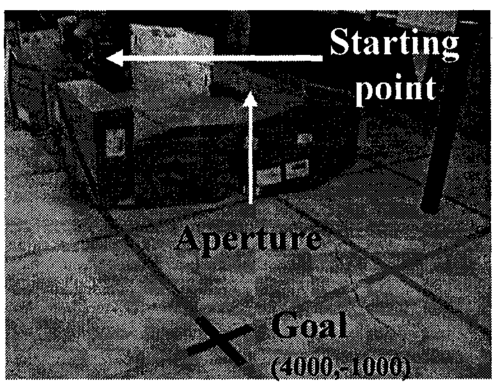



**Figure 6.7: Experimental set up for position determination of a LRF for a moving robot**

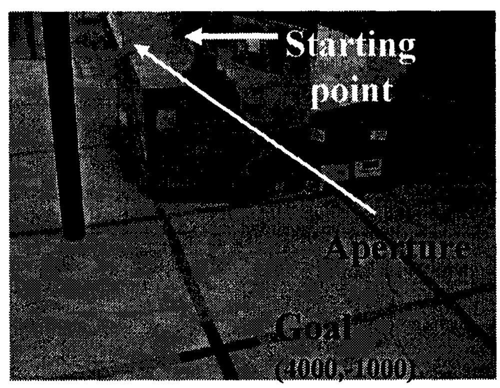
The robot was driven manually in the grid and the position and orientation of the robot were measured with the LRF, wheel encoders, and wheel encoders with gyroscope correction. The data obtain from the three methods were then analyzed and compared.

### 6.3.3.3 Tests for the autonomous navigation system in three maze layouts

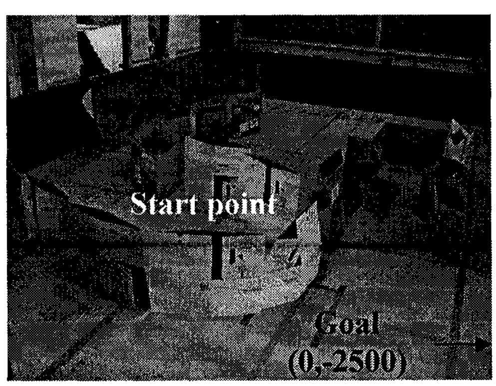
Three test layouts of the mazes are shown in Figure 6.8. The mazes were made of cardboard and painted concrete floor. The robot was initially placed at its starting position (0,0) (origin of the global coordinate frame) and its goal was set as (4000 mm, -1000 mm) for the mazes shown in Figures 6.8 (a) and (b). The goal for Figure 6.2 (C) was (0, -2500 mm). The technique developed in Chapter V, section 5.3.5.4, (Landmark localization method for position determination on a grid layout) was included in the autonomous navigation algorithm. In each of the three mazes, two artificial landmarks were placed near the goals. The navigational algorithm computed the position and orientation of the robot using the artificial landmark localization technique to compensate for errors in the dead-reckoning method.




**(A)**




**(B)**




**(C)**

**Figure 6.8: Maze layouts**

The specific objectives were:

- To test the autonomous navigation system that would enable the robot to achieve predetermined goals.
- To test the autonomous navigation system on detecting and avoiding obstacles on its way to the goal positions.

# 6.4 RESULTS AND DISCUSSIONS

## 6.4.1 Tests for the FOV of the sonar sensors

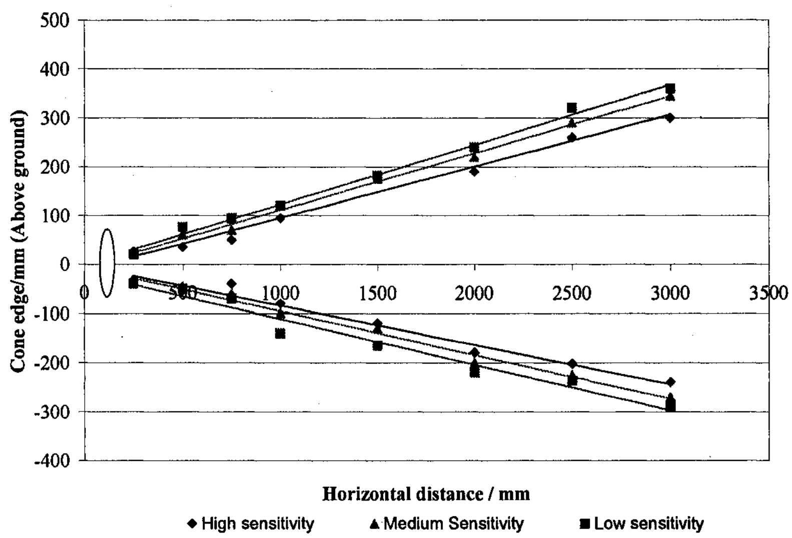
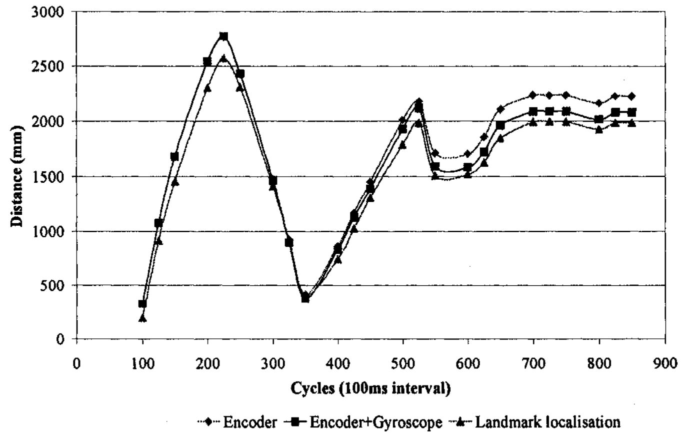


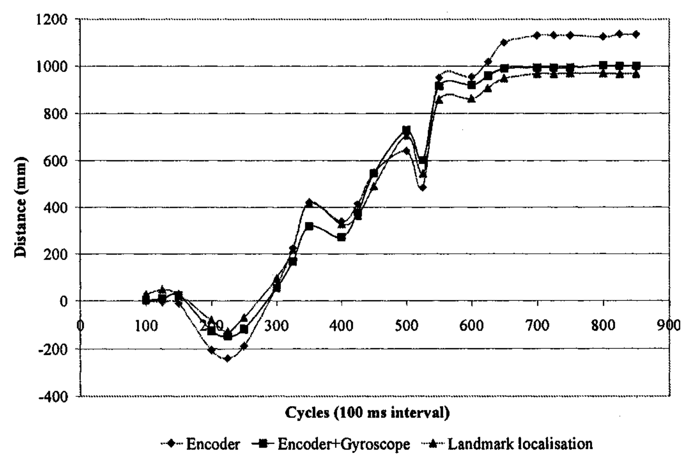

**Figure 6.9: Sonar sensors field of view**

Figure 6.9 shows the FOV of the sonar sensors measured at high, medium and low sensitivity. The FOV at low, medium and high sensitivity setting was found to be 28°, 26° and 22° respectively. At high sensitivity the FOV was narrow (22°), meaning the sonar signals were concentrated in a smaller region, thereby enabling the sonar to detect smaller objects and objects at distances further away. As the sensitivity was lowered, the sonar beam spread out forming a wider conical shape. In this case the signal strength was weaker, therefore the robot's ability to detect small obstacles was reduced. The appropriate sensitivity was selected based on the robot's operating environment. A low sonar sensitivity setting would be suitable if the robot was operating in a noisy environment, or on uneven or highly reflective surface. A medium sensitivity setting was selected in this research as the robot was operating in a painted concrete surface, which had some degree of reflection due to the presence of light and glossy surface color of the ground. It was also important that the sonar sensors detect small objects at distances up to 3000 mm (selected by software).

## 6.4.2 Tests for position determination capabilities of a LRF for a moving robot




(a)




(b)

**Figure 6.10: Changes of X and Y coordinates when the robot moved randomly in the test layout, (a) changes in X coordinates, (b) changes in Y coordinates**

Figure 6.10 shows the results obtained from the landmark localization technique as the robot moved randomly in its operating environment. In both figures, the encoder measurements, dead-reckoning with gyroscope correction measurements, and the LRF-based landmark localization measurements using the LRF are shown. The landmark localization measurements followed a similar pattern as the other two methods. An analysis of the various position estimation methods was done in section 5.4.4 (Landmark localization method for position estimation in a grid layout), in a static condition. In a static condition, the localization method was found to have a RMSE of 0.033 m and 0.021 m for the X and Y coordinates respectively. A comparison of errors between measured values and desired values using a "Turkey test" showed that there was a significant difference between the landmark localization method and the two other methods (Table 5.7)

Results for the robot's heading were also recorded and the data is shown graphically in Figure 6.11.

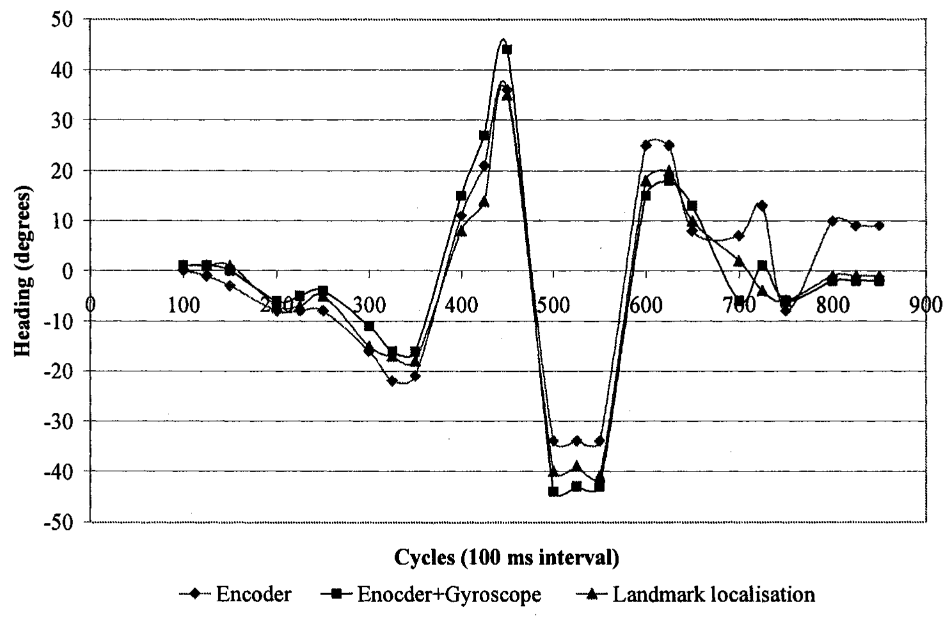
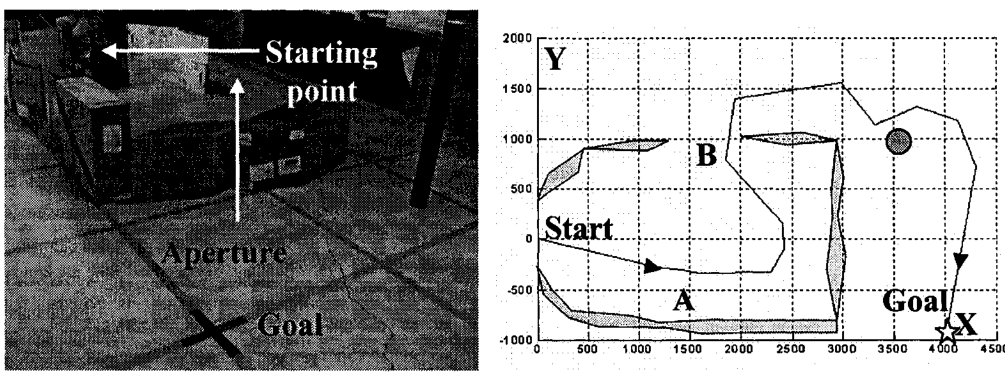



**Figure 6.11: Heading angle when the robot moved randomly in the test layout**

Position estimation using artificial landmarks and a LRF was very reliable in a dynamic condition and over long periods of time. It can be seen from the results that as time increased there was a tendency for larger differences between position estimation for the three methods. As time increases errors due to slip and gearbox play accumulates rendering the dead-reckoning method unreliable.

### 6.4.3 Tests for the autonomous navigation system in the three maze layouts

As the robot traversed a maze, its Cartesian coordinates and orientation were logged to a data file. Subsequently, maps of the robot's paths were developed using Matlab (V 7.0, Mathworks, Massachusetts, USA). The system performance was evaluated by the robot's ability to achieve a predetermined goal, whilst avoiding obstacles on its way. The system was also assessed in terms of the positional accuracy of robot at its final destination.




(a) (b)

**Figure 6.12: Path for Maze layout (A)**

Figure 6.12 shows the path taken by the robot in a maze layout. The shaded region represents the walls and obstacles in the operating environment. The goal was set at (4000 mm,-1000 mm) and the robot autonomously navigated to the goal point avoiding obstacles on its way. Its final position as obtained from the landmark localization technique was (4010 mm, -980 mm). The percentage error between the desired position and landmark localization measurement for the X and Y coordinates were 0.25% and 2% respectively. Initially, the software computed the *goal alignment angle*, which was used to establish the course of the robot. At the starting point the robot was in the *goal alignment state*. The following three conditions were true, hence the robot turned to the right and set a forward translational velocity until it reached point A:

- The *goal alignment angle* was negative
- Regions1&2 were clear
- DetectorR was clear

When the robot traveled to an approximate location at (1600 mm, 350 mm) (Point A in Figure 6.12 (b)) it detected the walls. At that instance the robot was in the *stop state.* An obstacle was detected in Region 1. The direction of rotation as defined in the algorithm was dictated by DetectorL1 and DetectorR1. In this case, DetectorL1 sensed an opening to the left, which was wide enough to accommodate the robot, hence it turned to the left. The magnitude of the left rotation was done sequentially until Region 2 was clear. Since a left turn was executed, the obstacle that was detected at that point, which was in practice the wall, was kept to the right side of the robot. Regions1&2 were used for keeping the obstacle to the desired side.

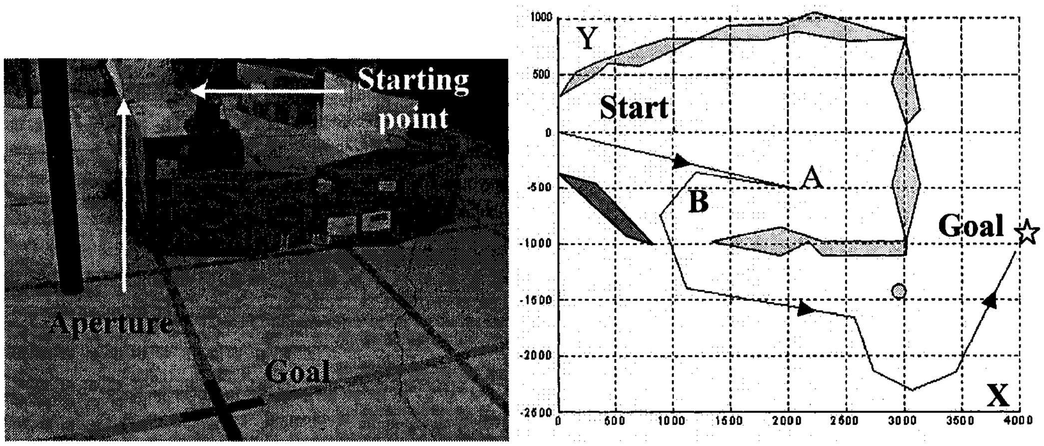
Between Point A and Point B the wall (obstacle) was kept to the right and DetectorR continuously checked for an interrupted forward rotational clearance safety distance of 250 mm. At Point B, the robot changed its operating state to *rotation-translational state* because at that point DetectorR sensed sufficient rotational clearance safety. The *maximum rotational angle* was used a limit for a right turn, followed by a *maximum translational* distance of 1 m.

After a way out of the maze was found (that is at Point B), the robot entered a goal alignment state, and the software computed the *goal alignment angle*. In that case the angle was a negative angle (the position of the goal relative to the robot was in a clockwise direction, hence a negative angle). The robot turned to the right to travel towards the goal. A right turn was selected because the following three conditions were true (point B in Figure 6.12 (b)):

- DetectorR was clear – enough clearance to the right
- The goal alignment angle was negative

- Distance from last aiming pose was greater than 150 mm – this distance was included to avoid the robot from setting its goal heading too frequently

The algorithm continuously updated the *goal alignment angle*, and monitored DetectorR and DetectorL. From Point B to the goal point the obstacle was kept to the right until the goal was achieved. As the robot approached the goal, its position and orientation was computed using the LRF-based localization method.




(a) (b)

**Figure 6.13: Path for maze layout (B)**

Figure 6.13 shows maze layout B with the path taken by the robot and the position of obstacles and walls. The goal was set at (4000 mm, -1000 mm) and the final position as measured by the landmark localization technique was (3975 mm, -1018 mm). The percentage error between the desired position and the landmark localization measurement was 0.63% and 1.8% for the X and Y coordinates respectively. The algorithm followed the same principle as discussed for the previous layout. Initially, the software computed the *goal alignment angle*, which was used to establish the course of the robot. At the starting point the robot was in the *goal alignment state*. The following three conditions were true, hence the robot rotated to the right and set a forward translational velocity until it reached point A:

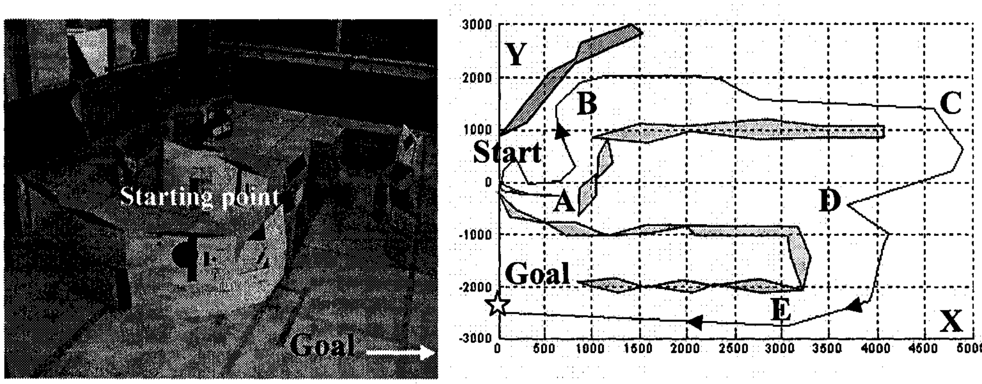
- The goal alignment angle was negative
- Regions1&2 were clear

- DetectorR was clear

At point A the robot entered a *stop state*. The translational velocity was set to zero and the robot rotated to the right because DetectorR1 sensed sufficient clearance to the right to accommodate the robot. Between Point A and B the wall was kept the left side of the robot. Since the wall was kept to the left side DetectorL was actively checking for sufficient rotational clearance safety of 250 mm. And at point B the robot entered a *rotational-translational state*. A left turn was taken because:

- DetectorL was clear
- The goal alignment angle was positive
- Distance from last aiming pose was greater than 150 mm

Once the rotational-translational state was completed at Point B the obstacles were kept to the left until the goal was achieved at (4000 mm, -1000 mm).




(a) (b)

**Figure 6.14: Path for maze layout (C)**

Figure 6.14 shows a structured environment with a goal at (0 mm, -2500 mm). The final position as recorded by the landmark localization technique was (13 mm, -2506 mm). At the starting point, the algorithm calculated the *goal alignment angle* and set that angle as the heading of the robot. There were obstacles present in the vicinity of the robot that prevented it from traveling directly to the goal as can be seen from Figure 6.14, hence Regions 1 & 2 were used to guide the vehicle out that section of the maze until it reached point A. At point A the robot did not repeat any of its previous paths as DetectorF was used to prevent any repetition. A way out of the initial part of the maze was detected by DetectorL1, as the aperture at point B was wide enough to accommodate the robot. From point B to C the robot continued to be guided by Regions 1 and 2 but was constantly checking for rotational clearance safety of 250 mm. Along points B to C Detector R was used to check for side clearance. Rotational clearance was found at point C and a right turn was selected because:

- DetectorR was clear
- Goal alignment angle was negative
- Distance from last aiming pose was greater than 150 mm.

A left turn was taken at point D because:

- DetectorL was clear
- Goal alignment angle was positive

At point E the *goal alignment angle* was computed and the robot set that angle as its heading to successfully achieve its target.

## 6.5 CONCLUSIONS

In conclusion, the sonar sensor's sensitivity was investigated and the relevant FOV was selected. A medium sensitivity setting was selected based on the operating environment.

The LRF successfully detected two artificial landmarks in a dynamic state and the algorithm developed computed the Cartesian coordinates and heading of the robot. These measurements were compared with the dead-reckoning measurements, and the dead-reckoning with gyroscope correction measurements. The results showed that as time increased there was tendency for larger discrepancies between the results.

An autonomous navigation was successfully developed and tested in three maze layouts. Several virtual regions and detector bands were established to aid in the navigations process. Two main regions (one inner boundary and one outer boundary) were established at the front of the robot to detect obstacles and determine the angle of rotation. The inner region was used to detect obstacles and outer region dictated the angle of rotation, i.e. until the obstacle appeared out of the outer region. Two rotational side clearance safety bands were developed to check for an interrupted forward clearance of 250 mm when the robot was aligning itself with a predefined goal. The direction (+ve or –ve) of the goal alignment angle was also used to determine the direction of rotation only when the robot was aligning with a goal. And finally, three additional virtual bands, two at the side and one in front, all 100 mm in width, were added. The two side bands were used to check to for a minimum forward interrupted distance of 500 mm, which was used to determine the direction of rotation (left or right) when an obstacle was present. The detector band at the front was developed to check for previously traversed or intersected paths. These predefined boundaries were incorporated in an algorithm, and using the LRF and eight sonar sensors a mobile robot successfully navigated in a structured environment.

## 6.6 REFERENCES


Beom, H.R., and H.S. Cho. 2000. Sonar-based navigation experiments on a mobile robot in indoor environments. *Proceedings of the IEEE Symposium on Intelligent Control.* 395-401.

Borenstein, J., and Y. Koren. 1991. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transaction on Robotics and Automation.* Vol. 7 (4), 535-539.

Elfes, A. 1987. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation.* Vol. 3 (3), 249-265.

Fan, Z., G.S. Pereira, and V. Kumar. 2005. Cooperative localization and tracking in distributed robot-sensor networks. *Tsinghua Science and Technology.* Vol. 10 (1), 99-101.

Hoffman, R., and E. Krotkov. 1991. Perception of rugged terrain for a walking robot: true confession and new directions. *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems.* Vol. 3, 1505-1510.

Langer, D., and C. Thorpe. 1992. Sonar based outdoor vehicle navigation and collision avoidance. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems.* Vol. 2, 1445-1450.

Langer, D., M. Mettenleiter, and C. Frohlich. 2000. Imaging laser scanner for 3-D modeling and surveying applications. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 1, 116-121.

McKinion, J.M., J.L. Willers, and J.N. Jenkins. 2004. Wireless local area networking for farm operations and farm management. 2004 ASAE Annual International Meeting, Paper number 043012.

Miller, D. 1984. Two dimensional mobile robot positioning using onboard sonar. Proceedings in Pecora IX Remote Sensing Symposium.

Montano, L., and J.R. Asensio. 1997. Real-time robot navigation in unstructured environments using a 3D laser rangefinder. *Proceedings of the IEEE International Conference on Intelligent Robot and Systems*. Vol. 2, 526-532.

Olivier, J.L., and F. Ozguner. 1986. A navigation algorithm for an intelligent vehicle with a laser rangefinder. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 3, 1145-1150.

Ye, C., and J. Borenstein. 2002. Characterization of a 2-D laser scanner for mobile robot obstacle negotiation. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 3, 2512-2518.

# VII. GENERAL CONCLUSIONS

An autonomous navigation system and an artificial landmark localization technique were successfully developed and on a mobile robot in an indoor environment. In conclusion the following were achieved:

- An autonomous navigation system was developed using data from a laser range-finder (LRF) and a ring of eight sonar sensors. The LRF was used for setting target points and to implement the autonomous navigational routine in a maze. The ring of eight sonar sensors was used primarily for obstacle detection and avoidance.
- A landmark localization technique was developed to determine the position and orientation of a mobile robot with respect to a global coordinate frame, using a LRF. Two artificial landmarks were used with known coordinates in the global coordinate frame. The LRF detected the distances and angles to the two artificial landmarks. A coordinate transformation method was developed to calculate the position of the robot in a global coordinate frame either in a static or dynamic state.
- The position of the robot was also obtained using dead-reckoning (from two wheel encoders), and inertial measurements from an onboard gyroscope. A Kalman filter algorithm fused the wheel encoder measurements with the gyroscope measurements to produce an improved estimate of position and orientation of the robot. Measurements from the dead-reckoning method, dead-reckoning with gyroscope correction, and the LRF-based landmark localization methods were compared. The RMSE with respect to the robot's actual position for the X and Y coordinates were: Dead-reckoning – 0.175 m and 0.154 m, respectively; Dead-reckoning with gyroscope correction – 0.135 m and 0.117 m, respectively; Artificial landmark localization – 0.033 m and 0.026 m, respectively. The LRF-based artificial landmark localization method offered the best estimate of positioning.
- The sensitivity of the sonar sensors was investigated and the field of view (FOV) was clearly defined for low, medium and high sensitivity settings.

The medium sensitivity setting was used for the experiments because of the requirement for detecting small objects at distances up to 3000 mm.

- An advanced autonomous navigation system was further developed using a LRF and sonar sensors. Two virtual regions in front of a robot were established to successfully prevent collision. An inner region was used to detect obstacles and an outer region to determine the angle of rotation to avoid the obstacle. Two detector bands at the left and right sides, respectively, were developed to check for apertures in the robot's environment, so as to determine the direction of rotation of the robot in the event of detecting an obstacle. Additionally, two more detector bands at both sides of the robot were established to check for rotational clearance safety to ensure no object impeded the motion of the robot during turning manoeuvres. Due to the nature of autonomous navigation of mobile robots, three robotic adjustment states were defined to ensure intelligent decision making and motion control. A *stop state*, *goal alignment state* and *rotational-translational* states were defined.
- The advanced autonomous navigation system was tested in three different maze layouts and the robot successfully accomplished its preset goal using the virtual regions and detector bands to detect and avoid obstacles in real-time.

# GENERAL REFERENCES


Adachi, Y., H. Saito, Y. Matsumoto, and T. Ogasawara. 2003. Memory-based navigation using data sequence of laser range finder. *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation.* Vol. 1, 479-484.

Akbarally, H., and L. Kleeman. 1995. A sonar sensor for accurate 3D target localization and classification. *Proceeding of the IEEE International Conference on Robotics and Automation.* Vol. 3, 3003-3008.

Araujo, E.G., and R.A. Grupen. 1998. Feature detection and identification using a sonar array. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 2, 1584-1589.

Barshan, B., and H.F. Durant-Whyte. 1994. Orientation estimate for mobile robots using gyroscopic information. *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robot and Systems (IROS '94).* Vol. 3, 1867-1874.

Benson, E.R., J.F. Reid, and Q. Zhang. 2003. Machine vision-based guidance system for agricultural grain harvesters using cut-edge detection. *Biosystems Engineering.* Vol. 86 (4), 389-398.

Beom, H.R., and H.S. Cho. 2000. Sonar-based navigation experiments on a mobile robot in indoor environments. *Proceedings of the IEEE Symposium on Intelligent Control.* 395-401.

Betke, M., and L. Gurvits. 1997. Mobile robot localization using landmarks. *IEEE Transaction on Robotics and Automation.* Vol. 13 (2), 251-263.

Billingsley, J., and M. Schoenfisch. 1997. The successful development of a vision guidance system for agriculture. *Computers and Electronics in Agriculture.* Vol. 16 (2), 147-163.

Blackmore, S. 1994. Precision farming: An introduction. *Outlook on Agriculture.* Vol. 23 (4), 275-280.

Borenstein, J., and L. Feng. 1996. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transaction on Robotics and Automation.* Vol. 12 (6), 869-880.

Borenstein, J., and Y. Koren. 1988. Obstacle avoidance with ultrasonic sensors. *IEEE Journal of Robotics and Automation.* Vol. 4 (2), 213-218.

Borenstein, J., and Y. Koren. 1989. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 19 (5), 1179-1187.

Borenstein, J., and Y. Koren. 1991. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transaction on Robotics and Automation.* Vol. 7 (4), 535-539.

Bonicelli, B., and M.O. Monod. 1987. A self propelled ploughing robot. *American Society of Agricultural Engineers*. Paper 87-1064.

Brunnert, A. 2005. Automatic guidance of harvesting machinery – From mechanical sensors to Global Position Systems. *Landward*. Vol. 60 (2), 2-4.

Buluswar, S.D., and B.A. Draper. 1998. Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*. Vol. 11, 245-256.

Chakravarthy, A., and D. Ghose. 1998. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 28 (5), 562-574.

Chateau, T., C. Debain, F. Collange, L. Trassoudaine, and J. Alizon. 2000. Automatic guidance of agricultural vehicles using a laser sensor. *Computers and Electronic in Agriculture*. Vol. 8 (3), 243-257.

Chen, B., S. Tojo, and K. Watanabe. 2003. Machine vision based guidance system for automatic rice transplanters. *Applied Engineering in Agriculture*, ASAE. Vol. 19 (1), 91-97.

Chen, Y.R., K. Chao, M.S. Kim. 2002. Machine vision technology for agricultural applications. *Computers and Electronics in Agriculture*. Vol. 36, 173-191.

Choi, C.H., D.C. Erbach, R.J. Smith. 1990. Navigational tractor guidance system. *Transations of the American Society of Agricultural Engineers*. Vol. 33 (3), 699-706.

Cordesses, L., C. Cariou, M. Berducat. 2000. Combine harvester control using Real Time Kinematic GPS. *Precision Agriculture*. Vol. 2 (2), 147-161.

Crisman, J., and C. Thorpe. 1990. Color vision for road following, vision and navigation. The Carnegie Mellon NAVLAB, Kluwer.

Dickmanns, E.D., and B.D. Mysliwetz. 1992. Recursive 3D road and relative ego-state recognition. *IEEE Transactions on Pattern analysis and Machine Intelligence*. Vol. 14 (2), 199-213.

Diosi, A., and L. Kleeman. 2004. Advanced sonar and laser range finder fusion for simultaneous localization and mapping. *Proceeding of the IEEE International Conference on Intelligent Robots and Systems.* Vol. 2, 1854-1859.

D'Orazio, T., F.P. Lovergine, M. Ianigro, E. Stella, and A. Distante. 1994. Mobile robot position determination using visual landmarks. *IEEE Transactions on Industrial Electronics*. Vol. 41 (6), 654-662.

Durrant-Whyte, H.F. 1996. An autonomous guided vehicle for cargo handling applications. *International Journal of Robotics Research.* Vol. 15 (5).

Elfes, A. 1987. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation.* Vol. 3 (3), 249-265.

Fan, Z., G.S. Pereira, and V. Kumar. 2005. Cooperative localization and tracking in distributed robot-sensor networks. *Tsinghua Science and Technology.* Vol. 10 (1), 99-101.

Finn-Kelcey, P., and V.M. Owen. 1967. Leader cable tractor guidance. *Proceedings of the Agricultural Engineering Symposium of the Institution of Agricultural Engineers.*

Fitzpatrick, K., D. Pahnos, W.V. Pype. 1997. Robot windrower is first unmanned harvester. *Industrial Robot*. Vol. 24 (5), 342-348.

Fujimura, K. 1991. Motion planning in dynamic environments. Tokyo, Japan. Springer – Verlag.

Gerrish, J.B., G.R. Fehr, G.R. Van, and D.P. Welch. 1997. Self-steering tractor guided by computer vision. *Applied Engineering in Agriculture*. Vol. 13 (5), 559-563.

Gibbons, G. 2000. Turning a farm art into science – an overview of precision farming. URL: http://www.precisionfarming.com.

Hague, T., J.A. Marchant, and N.D. Tillett. 2000. Ground based sensing system for autonomous agricultural vehicles. *Computer and Electronics in Agriculture*. Vol. 25 (1/2), 11-28.

Hague, T., and N.D. Tillett. 1996. Navigation and control of an autonomous horticultural robot. *Mechatronics.* Vol. 6 (2), 165-180.

Han, S., Q. Zhang, H. Noh, and B. Shin. 2004. A dynamic performance evaluation method for DGPS receivers under linear parallel tracking applications. *Transactions of the American Society of Agricultural Engineers*. Vol. 47 (1), 321-329.

Hayet, J.B., F. Lerasle, and M. Devy. 2003. Visual landmarks detection and recognition for mobile robot navigation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Vol. 2, 313-318.

Hilton, D.J., and A.A.W. Chestney. 1973. Low cost self steering devices for out-of-furrow ploughing. *The Agricultural Engineer*. Vol. 28, 102-106.

Hoffman, R., and E. Krotkov. 1991. Perception of rugged terrain for a walking robot: true confession and new directions. *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems.* Vol. 3, 1505-1510.

Hoppen, P., T. Knieriemen, and E. Von Puttkamer. 1990. Laser-radar based mapping and navigation for an autonomous mobile robot. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 2, 948-953.

Hummel, J.W., K.A. Sudduth, and S.E. Hollinger. 2001. Soil moisture and organic matter prediction of surface subsurface soils using a NIR sensor. *Computers and Electronics in Agriculture*. Vol. 32, 149-165.

Hwang, Y.K., and N. Ahuja. 1992. Gross motion planning – a survey. *ACM Computer Survey.* Vol. 24, 219-291.

Kabuka, M.R., and A.E. Arenas. 1987. Position verification of a mobile robot using a standard pattern. *IEEE Journal of Robotics and Automation*. Vol. 3 (6), 505-516.

Kaminaka, M.S., G.E. Rehkugler, and W.W. Gunkel. 1981. Visual monitoring in a simulated agricultural machinery operation. *Human factors*. Vol. 23 (2), 165-173.

Kanayama, Y., and B.I. Hartman. 1989. Smooth local path planning for autonomous vehicles. *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 3, 1265-1270.

Kawamura, N., and K. Namikawa. 1984. Automatic steering of tractor with rotary tiller. Research report on agricultural machinery, Kyoto University. No. 14.

Keicher, R., and H. Seufert. 2000. Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture*. Vol. 25, 169-194.

Krotkov, E. 1989. Mobile robot localization using a single image. *Proceeding of the IEEE International Conference on Robotics and Automation*. Vol. 2, 978-983.

Koshizen, T., P. Bartlett, and A. Zelinsky. 1999. Sensor fusion of odometry and sonar sensors by the Gaussian Mixture Bayes technique in mobile robot position estimation. *Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 4, 742-747.

Laine, P. 1994. Development methods of controller used in automatic guidance system. *Proceeding of X11 World Congress on Agricultural Engineering*. 1159-1166.

Langer, D., and C. Thorpe. 1992. Sonar based outdoor vehicle navigation and collision avoidance. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 2, 1445-1450.

Langer, D., M. Mettenleiter, and C. Frohlich. 2000. Imaging laser scanner for 3-D modeling and surveying applications. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, 116-121.

Leonard, J.J., and H.F. Durant-Whyte. 1991. Simultaneous map building and localization for an autonomous mobile robot. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 3, 1442-1447.

Lutzeler, M., and E.D. Dickmanns.1998. Road recognition with MarVEye, intelligent vehicles. Stuttgart. (http://www.unibw-muenchen.de/campus/LRT/LRT13/Veroeffentlichungen/index.html).

Ma, L., and K.L. Moore. 2003. Sonar and laser based HIMM map building for collision avoidance for mobile robots. *IEEE International Symposium on Intelligent Control*. 755-760.

Mas, F.R., Q. Zhang, J.F. Reid, and J.D. Will. 2002. Machine vision row crop detection using Blob Analysis and the Hough Transform. *Proceedings of the International Conference on Automation Technology for Off-Road Equipment*. 327-336.

Mata, M., J.M. Armingol, A. De La Escalera, and M.A. Salichs. 2003. Using learned visual landmarks for intelligent topological navigation of mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, 1324-1329.

Matthies, L., A. Kelly, T. Litwin, and G.T. Tharp. 1995. Obstacle detection for unmanned ground vehicles: A progress report. *Intelligent vehicles*.

Matthies, L., and S.A. Shafer. 1987. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*. Vol. 3 (3), 239-248.

Mazl, R., and L. Preucil. 2000. Building a 2D environment map from laser range-finder data. *Proceedings of the IEEE Intelligent Vehicles Symposium*. 290-295

McKinion, J.M., J.L. Willers, and J.N. Jenkins. 2004. Wireless local area networking for farm operations and farm management. 2004 ASAE Annual International Meeting, Paper number 043012.

McMahon, B.C., B.R. Tennes, and T.H. Burkhardt. 1982. Development of apple harvester microprocessor based steering control system. *American Society of Agricultural Engineers*. Paper 82 – 3038.

Miller, D. 1984. Two dimensional mobile robot positioning using onboard sonar. Proceedings in Pecora IX Remote Sensing Symposium.

Montano, L., and J.R. Asensio. 1997. Real-time robot navigation in unstructured environments using a 3D laser rangefinder. *Proceedings of the IEEE International Conference on Intelligent Robot and Systems*. Vol. 2, 526-532.

Morgan, K.E. 1958. A step towards an automatic tractor. *Farm mech.* Vol. 10 (13), 440-441.

Myers, D.B., N.R. Kitchen, R.J. Miles, and K.A. Sudduth. 2000. Estimation of a soil productivity index on claypan soils using soil electrical conductivity. *Procedings of the 5th International Conference on Precision Agriculture*. July 16-19, 2000, Bloomington, MN, USA.

Noguchi, N., M. Kise, K. Ishii, and H. Terao. 2002. Field automation using robot tractor. *Proceedings of the International Conference on Automation Technology for Off-Road Equipment*. July 26-27, 2002 Chicago, Illinois, USA, 701P0502, 398-404.

Nagasaka, Y., N. Umeda, Y. Kanetai, K. Taniwaki, and Y. Sasaki. 2004. Autonomous guidance for rice transplanting using global positioning and gyroscopes. *Computer and Electronics in Agriculture*. Vol. 43 (3), 223-234.

Nieminen, T., and M. Sampo. 1993. Unmanned vehicles for agricultural and off-highway applications. SAE Technical Paper Series NO. 93-2475.

O'Connor, M., T. Bell, G. Elkaim, and B. Parkinson. 1996. Automatic steering of farm vehicles using GPS. *Proceedings of the 3rd International Conference on Precision Agriculture*. 767-778.

Olivier, J.L., and F. Ozguner. 1986. A navigation algorithm for an intelligent vehicle with a laser rangefinder. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 3, 1145-1150.

Ollis, M., and A. Stenz. 1996. First results in vision-based crop line tracking. *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, 951-956.

Park, K., H. Chung, J. Choi, and J.G. Lee. 1997. Dead-reckoning navigation for an autonomous mobile robot using differential encoder and a gyroscope. *Proceedings of the IEEE International Conference on Advanced Robotics*. 441-446.

Patterson, R.J., B.W. Fehr, and L.P. Sheets. 1985. Electronic guidance system for a planter. *American Society of Agricultural Engineers*. Paper 85 – 1587.

Pervan, B.S., and B.W. Parkinson. 1997. Cycle ambiguity estimation for aircraft precision landing using the Global Positioning System. *Journal of Guidance Control Dynamic*. Vol. 20 (4), 681-698.

Reid, J.F., and S.W. Searcey, 1987. Automatic tractor guidance with computer vision. *Proceedings of the International Congress on Off-Highway and Powerplant.*

Roberts, J.M., E.S. Duff, and P.I. Corke. 2002. Reactive navigation and opportunistic localization for autonomous underground mining vehicles. *An International Journal of Information Sciences.* 127-146.

Sala, P.L., R. Sim, A. Shokoufandeh, and S.J. Dickinson. 2004. Landmark selection for vision-based navigation. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Vol. 4, 3131-3138.

Schanzer, G. 1992. The use of satellite navigation systems for precise applications in land, air and space environments – status, problems and trends. *Journal of Satellite-based Positioning Navigation Communication.* Vol. 1, 94-100.

Schmidt, G., and F. Freyberger. 1996. Autonome Mobile Systeme. *Information aktuell*. Springer.

Schonberg, T., M. Ojala, J. Suomela, A. Torpo, and A. Halme. 1996. Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation. *International Journal of Syst. Sc.* Vol. 27 (8), 745-752.

Searcy, S.W., J.K. Schueller, Y.H. Bae, and B.A. Stout. 1990. Measurement of agricultural field location using microwave frequency triangulation. *Computers and Electronics in Agriculture*. Vol. 4 (3), 209-223.

Shmulevich, I., G. Zeltzer, and A. Brunfield. 1989. Laser scanning method for guidance of field machinery. *Transactions of the American Society of Agricultural Engineers.* Vol. 32 (2), 425-430.

Skewis, T., and V. Lumesly. 1994. Experiments with a mobile robot operating in a cluttered unknown environment. *International Journal of Robotic Systems*. Vol. 11 (4), 281-300.

Slaughter, D.C., P. Chen, and R.G. Curley. 1999. Vision guided precision cultivation. *Precision Agriculture*. Vol. 1 (2), 199-216.

Stoll, A., and H.D. Kutzbach. 2000. Guidance of a forage harvester with GPS. *Precision Agriculture*. Vol. 2 (3), 281-291.

Stombaugh, T.S., E.R. Benson, and J.W. Hummel. 1999. Guidance control of agricultural vehicles at high field speeds. *Transactions of the American Society of Agricultural Engineers*. Vol. 42 (2), 537-544.

Suggs, C.W., B.K. Huang, and P.E. Davis. 1972. Automatic steering of field machines. *American Society of Agricultural Engineers*. Paper 72 – 122.

Sugihara, K. 1988. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics and Image Processing*. Vol. 42, 112-129.

Sutherland, K.T., and W.B. Thompson. 1993. Inexact navigation. *Proceedings of IEEE International Conference on Robotics and Automation.* Vol. 1, 1-7.

Telle, M.G., and U.D. Perdok. 1979. Field experiments with a leader cable tractor guidance. *American Society of Agricultural Engineers*. Paper 79-1069.

Tillet, N.D. 1991. Automatic guidance for agricultural field machines: A review. *Journal of Agricultural Engineering Research.* Vol. 50, 167-187.

Turennout, P., E.L. Egmond, G. Hondered, and W. Jongkind. 1989. Obstacle avoidance for a mobile robot. IEEE International Workshop on Intelligent Robots and Systems. 598-604.

Vaganay, J., M.J. Aldon, and A. Fourinier. 1993. Mobile robot attitude estimation by fusion of inertial data. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 1, 277-282.

Van Zuydam, R.P. 1999. A driver's steering aid for an agricultural implement based on an electronic map and real time kinematic DGPS. *Computers and Electronic in Agriculture.* Vol. 24 (3), 153-156.

Urdiales, C., A. Bandera, R. Ron, and F. Sandoval. 1999. Real time position estimation for mobile robots by means of sonar sensors. *Proceedings of IEEE International Conference on Robotics and Automation.* Vol. 2, 1650-1655.

Victorino, A.C., and P. Rives. 2004. Bayesian segmentation of laser range scan for indoor navigation. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems.* Vol. 3, 2731-2736

Warner, M.G.R., and G.O. Harries. 1972. An ultrasonic guidance system for driverless tractors. *Journal of Agricultural Engineering research.* Vol. 17 (1), 1-9.

Whelan, B.M., A.B. McBratney, and B.C. Boydell. 1997. The Impact of Precision griculture. *Proceedings of the ABARE Outlook Conference*, 'The Future of Cropping in NW NSW', Moree, UK, p. 5.

Widden, M.B, and J.R. Blair. 1972. A new automatic tractor guidance system. *Journal of Agricultural Engineering Research.* Vol. 17, 10-21.

Yamauchi, B. 1996. Mobile robot localization in dynamic environments using dead-reckoning and evidence grids. *IEEE Conference on Robotics and Automation.* Vol. 2, 1401-1406

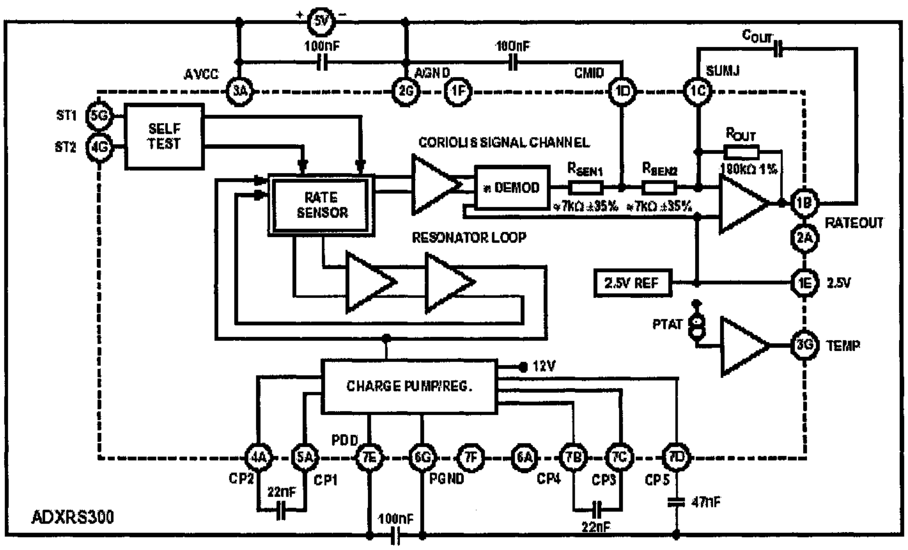
Ye, C., and J. Borenstein. 2002. Characterization of a 2D laser scanner for mobile robot obstacle negotiation. *Proceedings of the IEEE International Conference on Robotics and Automation.* Vol. 3, 2512-2518.

Zhang, N., M. Wang, and N. Wang. 2002. Precision agriculture – a worldwide overview. *Computer and Electronics in Agriculture.* Vol. 36, 113-132.

Zhang, Q., J.F. Reid, and N. Noguchi. 1999. Agricultural vehicle navigation using multiple guidance sensors. *Proceedings of the International Conference on Field and Service Robotics.* Pittsburg. UILU-ENG-99-7013.

# APPENDICES

## **Appendix A -** Functional block diagram of the ADXRS300 Gyroscope




**Functional block diagram of the gyro (Courtesy of Analog devices)**

**Appendix B** – Test results for linear distance and angular displacement accuracy

| Desired distance - DD(mm) | Measured distance (mm) | | Percentage error (%) (between DD and GE) | Percentage error (%) (between DD and E) |
|---|---|---|---|---|
| | Gyroscope+Encoder (GE) | Encoder (E) | | |
| 1000 | 948 | 924 | 5.20 | 7.60 |
| 2000 | 1880 | 1842 | 6.00 | 7.90 |
| 3000 | 2820 | 2780 | 6.00 | 7.33 |
| 4000 | 3793 | 3693 | 5.18 | 7.68 |
| 5000 | 4696 | 4620 | 6.08 | 7.60 |
| 6000 | 5649 | 5530 | 5.85 | 7.83 |
| 7000 | 6583 | 6498 | 5.96 | 7.17 |
| 8000 | 7491 | 7379 | 6.36 | 7.76 |
| 9000 | 8435 | 8310 | 6.28 | 7.67 |
| 10000 | 9368 | 9250 | 6.32 | 7.50 |
| Mean | | | 5.92 | 7.60 |

**(A) Percentage error between desired and measured distances**

| Desired angle - DA (°) | Measured angle (°) | | Percentage error (%) (between DA and GE) | Percentage error (%) (between DA and E) |
|---|---|---|---|---|
| | Gyroscope+Encoder (GE) | Encoder (E) | | |
| 45 | 44.0 | 43.0 | 2.22 | 4.44 |
| 90 | 88.0 | 86.0 | 2.22 | 4.44 |
| 135 | 132.0 | 139.5 | 2.22 | 4.07 |
| 180 | 175.0 | 171.5 | 2.78 | 4.72 |
| 225 | 220.0 | 214.5 | 2.22 | 4.67 |
| 270 | 263.0 | 257.0 | 2.59 | 4.81 |
| 315 | 306.5 | 298.5 | 2.70 | 5.24 |
| 360 | 349.0 | 341.5 | 3.06 | 5.14 |
| Mean | | | 2.50 | 4.69 |

**(B)Percentage error between desired and measured angles**

# Appendix C - Results for position estimation on grid layout

| Grid position | Coordinates | Encoder (E) | After Gyroscope (AG) | Actual (A) | Percentage error | |
|---|---|---|---|---|---|---|
| | | | | | (A and E) | (A and AG) |
| 1 | X (mm) | 4454.24 | 4205.2 | 4000 | 11.36 | 5.13 |
| | Y (mm) | 4237.54 | 4117.11 | 4000 | 5.94 | 2.93 |
| | Theta (deg) | 86.75 | 92.37 | 90 | 3.61 | 2.63 |
| 2 | X (mm) | 4267.89 | 4102.45 | 4000 | 6.70 | 2.56 |
| | Y (mm) | 3270.13 | 3367.76 | 3500 | 6.57 | 3.78 |
| | Theta (deg) | 88.34 | 92.43 | 90 | 1.84 | 2.70 |
| 3 | X (mm) | 4325.9 | 4006.01 | 4000 | 8.15 | 0.15 |
| | Y (mm) | 3200.77 | 3120.18 | 3000 | 6.69 | 4.01 |
| | Theta (deg) | 86.66 | 93.56 | 90 | 3.71 | 3.96 |
| 4 | X (mm) | 4201.88 | 4265.3 | 4000 | 5.05 | 6.63 |
| | Y (mm) | 2345.9 | 2400.24 | 2500 | 6.16 | 3.99 |
| | Theta (deg) | 87.1 | 92.57 | 90 | 3.22 | 2.86 |
| 5 | X (mm) | 4182.21 | 4233.20 | 4000 | 4.56 | 5.83 |
| | Y (mm) | 2120.14 | 2060.76 | 2000 | 6.01 | 3.04 |
| | Theta (deg) | 89.47 | 95.87 | 90 | 0.59 | 6.52 |
| 6 | X (mm) | 4200.50 | 4235.69 | 4000 | 5.01 | 5.89 |
| | Y (mm) | 1394.30 | 1445.21 | 1500 | 7.05 | 3.65 |
| | Theta (deg) | 88.59 | 92.22 | 90 | 1.57 | 2.47 |
| 7 | X (mm) | 4338.45 | 4248.12 | 4000 | 8.46 | 6.20 |
| | Y (mm) | 920.20 | 1025.05 | 1000 | 7.98 | 2.51 |
| | Theta (deg) | 84.99 | 90.00 | 90 | 5.57 | 0.00 |
| 8 | X (mm) | 4291.48 | 4125.36 | 4000 | 7.29 | 3.13 |
| | Y (mm) | 465.26 | 519.32 | 500 | 6.95 | 3.86 |
| | Theta (deg) | 87.49 | 91.23 | 90 | 2.79 | 1.37 |
| 9 | X (mm) | 4270.76 | 4231.47 | 4000 | 6.77 | 5.79 |
| | Y (mm) | 13.06 | 10.42 | 0 | - | - |
| | Theta (deg) | 84.73 | 93.28 | 90 | 5.86 | 3.64 |
| 10 | X (mm) | 4285.36 | 4130.25 | 4000 | 7.13 | 3.26 |
| | Y (mm) | -467.28 | -515.24 | -500 | 6.54 | 3.05 |
| | Theta (deg) | 94.54 | -91.28 | -90 | 205.04 | 1.42 |
| 11 | X (mm) | 4115.71 | 4256.33 | 4000 | 2.89 | 6.41 |
| | Y (mm) | -1068.4 | -1032 | -1000 | 6.84 | 3.20 |
| | Theta (deg) | -95.45 | -87.44 | -90 | 6.06 | 2.84 |
| 12 | X (mm) | 3945.37 | 4122.49 | 4000 | 1.37 | 3.06 |
| | Y (mm) | -1390.45 | -1435.78 | -1500 | 7.30 | 4.28 |
| | Theta (deg) | -95.74 | -89.46 | -90 | 6.38 | 0.60 |
| 13 | X (mm) | 4054.8 | 4119.07 | 4000 | 1.37 | 2.98 |
| | Y (mm) | -2120.59 | -1930.38 | -2000 | 6.03 | 3.48 |
| | Theta (deg) | -93.69 | -93.68 | -90 | 4.10 | 4.09 |
| 14 | X (mm) | 4397.20 | 4210.34 | 4000 | 9.93 | 5.26 |
| | Y (mm) | -2340.70 | -2390.20 | -2500 | 6.37 | 4.39 |
| | Theta (deg) | -90.22 | -92.34 | -90 | 0.24 | 2.60 |
| 15 | X (mm) | 3877.64 | 4181.95 | 4000 | 3.06 | 4.55 |
| | Y (mm) | -3201.81 | -3126.46 | -3000 | 6.73 | 4.22 |
| | Theta (deg) | -97.73 | -88.16 | -90 | 8.59 | 2.04 |
| 16 | X (mm) | 4389.90 | 4305.30 | 4000 | 9.75 | 7.63 |
| | Y (mm) | -3256.68 | -3400.23 | -3500 | 6.95 | 2.85 |
| | Theta (deg) | -89.89 | -91.24 | -90 | 0.12 | 1.38 |
| 17 | X (mm) | 3786.03 | 4266.15 | 4000 | 5.35 | 6.65 |
| | Y (mm) | -4290.97 | -4156.58 | -4000 | 7.27 | 3.91 |
| | Theta (deg) | -97.64 | -89.52 | -90 | 8.49 | 0.53 |
| 18 | X (mm) | -3226.17 | -3288.72 | -3000 | 7.54 | 9.62 |
| | Y (mm) | 3068.07 | 2835.80 | 3000 | 2.27 | 5.47 |
| | Theta (deg) | -178.00 | -171.35 | -180 | 1.11 | 4.81 |
| 19 | X (mm) | -5427.48 | -5309.86 | -5000 | 8.55 | 6.20 |
| | Y (mm) | 1268.52 | 924.90 | 1000 | 26.85 | 7.51 |
| | Theta (deg) | 92.46 | 91.33 | 90 | 2.73 | 1.48 |
| 20 | X (mm) | -2190.15 | -1951.44 | -2000 | 9.51 | 2.43 |
| | Y (mm) | -1817.47 | -1896.80 | -2000 | 9.13 | 5.16 |
| | Theta (deg) | -88.68 | -79.03 | -90 | 1.47 | 12.19 |

# Appendix D - Results for LRF-based artificial landmark localization

| Grid position | Coordinates | Actual | Encoder (E) | After gyroscope (AG) | Landmark localization (LL) | Percentage error | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | E | AG | LL |
| 1 | X (mm) | 3000 | 3260 | 3150 | 2970 | 8.67 | 5.00 | 1.00 |
| | Y (mm) | 3000 | 3242 | 3145 | 3025 | 8.07 | 4.83 | 0.83 |
| | Theta (°) | 0 | -3 | -1.5 | 0.6 | - | - | - |
| 2 | X (mm) | 2000 | 2158 | 2132 | 1970 | 7.90 | 6.60 | 1.50 |
| | Y (mm) | 3000 | 3271 | 3164 | 3022 | 9.03 | 5.47 | 0.73 |
| | Theta (°) | -12 | -11.2 | -12.5 | -11.8 | 6.67 | 4.17 | 1.67 |
| 3 | X (mm) | 1000 | 1135 | 1101 | 969 | 13.50 | 10.10 | 3.10 |
| | Y (mm) | 3000 | 3195 | 3150 | 3019 | 6.50 | 5.00 | 0.63 |
| | Theta (°) | 0 | 3 | 2 | 0.9 | - | - | - |
| 4 | X (mm) | 1000 | 1148 | 1144 | 971 | 14.80 | 14.40 | 2.90 |
| | Y (mm) | 2000 | 2144 | 2130 | 1953 | 7.20 | 6.50 | 2.35 |
| | Theta (°) | 0 | -5 | -2 | -0.7 | - | - | - |
| 5 | X (mm) | 2000 | 2167 | 2146 | 2054 | 8.35 | 7.30 | 2.70 |
| | Y (mm) | 2000 | 2105 | 2110 | 1961 | 5.25 | 5.50 | 1.95 |
| | Theta (°) | -27 | -25.1 | -28.2 | -27.8 | 7.04 | 4.44 | 2.96 |
| 6 | X (mm) | 3000 | 3275 | 3182 | 3041 | 9.17 | 6.07 | 1.37 |
| | Y (mm) | 2000 | 2204 | 2166 | 2027 | 10.20 | 8.30 | 1.35 |
| | Theta (°) | 0 | -3 | 1 | -1.8 | - | - | - |
| 7 | X (mm) | 3000 | 3255 | 3182 | 2964 | 8.50 | 6.07 | 1.20 |
| | Y (mm) | 1000 | 1140 | 1090 | 974 | 14.00 | 9.00 | 2.60 |
| | Theta (°) | 0 | 6 | -5 | -2.3 | - | - | - |
| 8 | X (mm) | 2000 | 2140 | 2122 | 1963 | 7.00 | 6.10 | 1.85 |
| | Y (mm) | 1000 | 1087 | 1084 | 972 | 8.70 | 8.40 | 2.80 |
| | Theta (°) | 0 | -3 | 3 | -1.6 | - | - | - |
| 9 | X (mm) | 1000 | 1061 | 1039 | 1026 | 6.10 | 3.90 | 2.60 |
| | Y (mm) | 1000 | 1110 | 1086 | 974 | 11.00 | 8.60 | 2.60 |
| | Theta (°) | 0 | -2 | -1.5 | -0.8 | - | - | - |
| 10 | X (mm) | 1000 | 1073 | 1073 | 980 | 7.30 | 7.30 | 2.00 |
| | Y (mm) | 0 | -21 | -15 | 4 | - | - | - |
| | Theta (°) | 0 | 2 | -1 | -1.5 | - | - | - |
| 11 | X (mm) | 2000 | 2142 | 2144 | 1980 | 7.10 | 7.20 | 1.00 |
| | Y (mm) | 0 | -57 | -2 | 3 | - | - | - |
| | Theta (°) | 0 | -1 | 6 | -0.8 | - | - | - |
| 12 | X (mm) | 3000 | 3202 | 3203 | 2967 | 6.73 | 6.77 | 1.10 |
| | Y (mm) | 0 | -53 | 1 | -6 | - | - | - |
| | Theta (°) | 0 | 2 | -1 | -0.8 | - | - | - |
| 13 | X (mm) | 3000 | 3198 | 3183 | 2960 | 6.60 | 6.10 | 1.33 |
| | Y (mm) | -1000 | -1181 | -1207 | -1026 | 18.10 | 20.70 | 2.60 |
| | Theta (°) | 0 | -1 | -1 | -0.5 | - | - | - |
| 14 | X (mm) | 2000 | 2127 | 2087 | 1971 | 6.35 | 4.35 | 1.45 |
| | Y (mm) | -1000 | -1115 | -1056 | -1037 | 11.50 | 5.60 | 3.70 |
| | Theta (°) | 13 | 14.3 | 12.1 | 13.4 | 10.00 | 6.92 | 3.08 |
| 15 | X (mm) | 1000 | 1072 | 1116 | 975 | 7.20 | 11.60 | 2.50 |
| | Y (mm) | -1000 | -1178 | -1160 | -1008 | 17.80 | 16.00 | 0.80 |
| | Theta (°) | 0 | -2 | 0 | 0 | - | - | - |
| 16 | X (mm) | 3000 | 3250 | 3145 | 3041 | 8.33 | 4.83 | 1.37 |
| | Y (mm) | -2000 | -2150 | -2100 | -1975 | 7.50 | 5.00 | 1.25 |
| | Theta (°) | 0 | -1 | -1.2 | 0.3 | - | - | - |
| 17 | X (mm) | 2000 | 2138 | 2075 | 2025 | 6.90 | 3.75 | 1.25 |
| | Y (mm) | -2000 | -2133 | -2069 | -2031 | 6.65 | 3.45 | 1.55 |
| | Theta (°) | 16 | 18.2 | 16.8 | 15.4 | 13.75 | 5.00 | 3.75 |
| 18 | X (mm) | 1000 | 1141 | 1096 | 968 | 14.10 | 9.60 | 3.20 |
| | Y (mm) | -2000 | -2142 | -2102 | -1980 | 7.10 | 5.10 | 1.00 |
| | Theta (°) | 0 | 3 | 1 | -1 | - | - | - |