Indoor Localization of a Mobile Robot Using Sensor Fusion

A thesis presented in partial fulfillment of the requirements for the degree of

Master of engineering

in

Mechatronics with Honours

at Massey University, Wellington,

New Zealand.

Jason Zhou

August 2011

Supervisor: Dr. Loulin Huang

Abstract

Reliable indoor navigation of mobile robots has been a popular research topic in recent years. GPS systems used for outdoor mobile robot navigation can not be used indoor (warehouse, hospital or other buildings) because it requires an unobstructed view of the sky. Therefore a specially designed indoor localization system for mobile robot is needed. This project aims to develop a reliable position and heading angle estimator for real time indoor localization of mobile robots. Two different techniques have been developed and each consisted of three different sensor modules based on infrared sensing, calibrated odometry and calibrated gyroscope. Integration of these three sensor modules is achieved by applying the real time Kalman filter which provides filtered and reliable information of a mobile robot's current location and orientation relative to its environment. Extensive experimental results are provided to demonstrate its improvement over conventional methods like dead reckoning. In addition, a control strategy is developed to control the mobile robot to move along the planned trajectory. The techniques developed in this project have potentials for the application for mobile robots in medical service, health care, surveillances, search and rescue in indoor environments.

Acknowledgements

I would like to acknowledge my Supervisor Dr. Loulin Huang, for his consistent support throughout the duration of my master degree. Thanks are also given to those people from the University, who has helped me a lot during the development process of this project. I would also like to express my deep gratitude to my family, especially to my parents for their constant encouragement, their tolerance and for their assistance in many ways for the successful completion of this thesis. Last but not least, I would like to thank those who always stood by me throughout my life and for their sacrifice and tolerance during the writing of this thesis.

Table of Content

List of Figures

List of Tables

1 Introduction

This section introduces the aims of this project and why indoor localization of mobile robots is important. And a brief overview of the developed indoor localization method for mobile robots is presented. Also the structure of the entire thesis is explained.

1.1 Aims

The main research aim is to develop a reliable indoor localization system for mobile robots. This research was also done to achieve an understanding of the specific topic that is covered in this project. The secondary aim of the research was to identify new research topics for continuations of research study.

1.2 Indoor localization of mobile robots

For a long time, accurate and reliable indoor localization of mobile robots have been a challenging research topic. This is due to common localization methods such as odometry which based on encoder readings have problems with accumulated errors. Other modern localization systems such as GPS (Global Positioning System) have several limitations. Most significantly, direct line of sight is required between the receiver and the satellites. Any objects obstructing this path can block the signal from a satellite making GPS suitable only for outdoor environment purposes. Therefore severely limiting its applications with mobile robots as a large proportion are designed for indoor use.

Figure 1: A mobile robot navigating in an indoor environment

Many different methods have been developed in an attempt to solve the problems of robot localization [1 - 18]. These can be classified into the following two main categories:

Relative Localization: The robot's position and orientation are determined relative to objects that are either stationary or moving in the environment. Pose coordinates are evaluated using data information provided by different on-board sensors, such as encoders, gyroscopes and accelerometers.

Absolute Localization: The robot's absolute position and orientation are evaluated using data information provided by external sensors such as visual landmarks, navigation beacons or GPS. However, such technique required expensive installation of sensors, high maintenance and computational costs.

Although the techniques described above can achieve localization for mobile robots, however they still face further physical limitations specific to the indoor environment. One of the ways to overcome this is to combine multiple sensory data from different sensor modules to provide better, reliable and accurate information of the robot's current location and orientation relative to its environment. Thus, in most mobile robot applications the relative and absolute positioning estimation methods have been employed together as one system to improve the localization performance [1 - 15].

1.3 Overview of the developed indoor localization system

In this project, a new solution for reliable indoor localization of mobile robots has been developed. The proposed localization system combines the absolute and relative position measurement obtained from three different sensor modules based on infrared sensing, calibrated odometry and calibrated gyroscope. Integration of the three sensor modules is achieved by implementing the Kalman filter technique in conjunction with a conditional algorithm created to analyze and to provide position measurement and heading angle data collected from the three different sensor modules. The Kalman filter technique is selected due to its ease of implementation and its ability to update multiple pose data continuously. Throughout the development process, two different approaches in our proposed method for localization have been developed and these can be represented by Figure 2 and Figure 3 respectively.

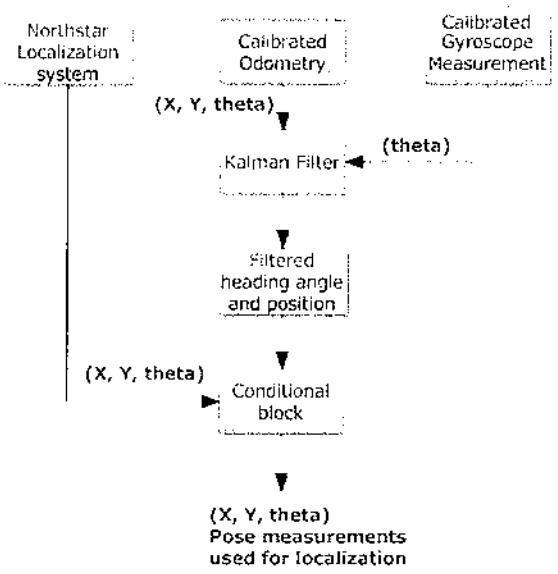

Figure 2: Overall structure of method one used for localization

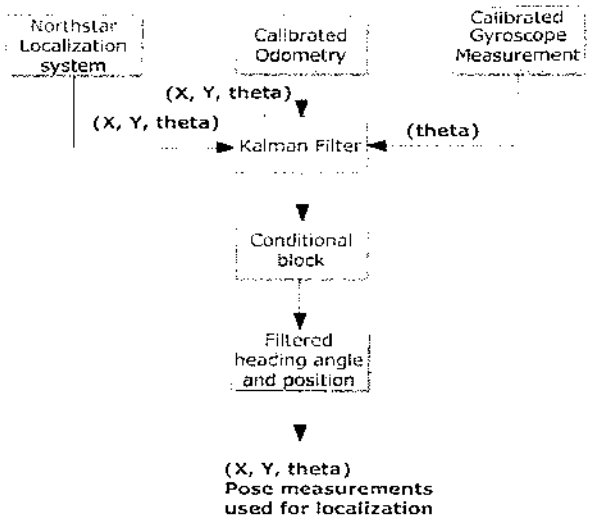

Figure 3: Overall structure of method two used for localization

Figure 2 and 3 shows the overall structure of the two different approaches of the developed indoor localization system. Essentially, each approach consists of the relative position measurement from the calibrated odometry and calibrated gyroscope and absolute position measurement from the optical beacon device known as the Northstar system [21]. The pose measurement collected from the three sensors modules are used to provide the information of the mobile robot's current location and orientation relative to its environment. The effectiveness of the two presented techniques and full description of the developed indoor localization system are explained in details in section 4.

1.4 Structure of the thesis

Section 2 presents an introduction to the field of mobile robot localization and the fundamental problems of localization. Also different localization techniques and calibration methods used to improve localization performance will be discussed and explained in details.

Section 3 presents the background of the robot model and details the software development process for robot control.

Section 4 presents testing and calibrations methods used and experiments results are explained in details.

In section 5, the conclusion of the thesis is presented and further development of the localization system and related work is discussed.

2 Literature reviews

This literature review is based on mobile robots that operate within a two dimensional plane and capable to orientate itself to any orientation in any position. The pose coordinates are in the form of (x, y, Θ). Where x and y are the Cartesian coordinates and Θ is the orientation of the robot.

2.1 Localization of mobile robots

In this section, an introduction to the field of mobile robot localization and the fundamental problems of localization is presented. Different localization techniques and calibration methods used to improve localization performance will also be discussed and explained in details.

2.1.1 Problems of localization

The term localization is the process of a mobile robot estimating its own position with respect to its surrounding environment. In order to have the mobile robot to perform useful tasks and to navigate autonomously, the position and orientation of the mobile robot needs to be accurately determined. Thus, robot localization is the key problem in providing autonomous capabilities to a mobile robot. These problems can be classified into the following three main categories as stated in [1 - 7], [25], and [34 - 37]:

- Local localization problem
- Global localization problem
- Kidnapped robot problem

In local localization, the mobile robot's initial home position is defined and the position of the robot's movement is tracked continuously as it moves from one location to another. Pose coordinates are evaluated using data information provided by different onboard sensors, such as encoders and gyroscopes. It is the easiest of the three problems mentioned above. However, the main challenge is when the mobile robot can not track its real position in the worst case and the location of the robot can not be recovered, thus position estimation error would grow without bound during the localization process.

In global localization, the mobile robot's initial home position is not defined. It is essentially a method of perceiving the correct position of the mobile robot in its surrounding environment by using external sensor devices without any prior knowledge of the initial home position. The main challenge here is that a map of the robot's surrounding environment is required simply because the estimated position by the mobile robot does not represent the environment and without a map the estimated position has no use and the robot can not adapt to the changes in the environment. And one of the disadvantages of this would be manually building a map require costly installations and maintenance. However, when a map of the environment is provided for the mobile robot, it will be able to localized itself and find out where it is.

The kidnapped robot problem is the hardest problem out of the three mentioned above. It is similar to the global localization issue in mobile robotics. The main challenge is when the mobile robot has to determine its correct position, when it is kidnapped at one position on the map and found itself on another unknown position on the map. In order to determine the correct pose of the mobile robot, the robot has to be able to recognize the unknown position. Essentially, the term kidnapping in this context means that the robot is shifted from one position to another position and movement can not be detected.

Furthermore, the local localization failure and kidnapped robot problem can be solved through global localization.

2.1.2 Localization methods

Many different solutions have been developed and implemented to tackle the problems in the field of robot localization. And these position estimation methods can be classified into the following two main categories:

- Relative positioning method
- Absolute positioning method

2.1.2.1 Relative positioning method:

The most common relative positioning method is based on odometry. It is the most basic way to estimate the robot's position by measuring the traveled distance of each wheel. Essentially, it is a method that aims at compensating odometric errors during robot navigation.

Odometry

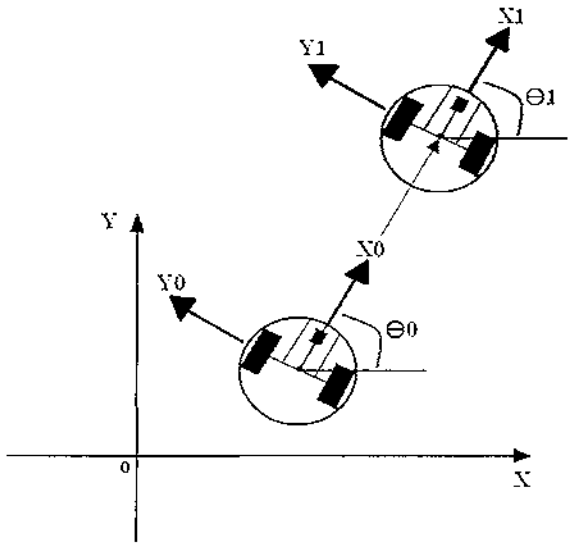

Figure 4: Body and navigation frame of a mobile robot

A typical differential drive mobile robot with two independently driven wheels has optical encoders mounted on the drive motors to count wheel revolutions. The collected wheel revolutions will be

translated into linear displacement relative to the floor. At a specified sampling interval (T), the encoders used to count the left and right wheel revolution have a number of increment (N_L) and (N_R) pulses respectively. At each time (t) the travel distance for each wheel can be computed with the following equations:

$$C_m = \frac{\pi . D_n}{n . C_e}$$

$$\Delta U_{L,t} = C_m N_{l,t}$$
$$\Delta U_{R,t} = C_m N_{r,t} \qquad (1)$$

(C_m) is the conversion factor used to translate encoder pulses into linear wheel displacement (position). (D_n) is the drive wheel diameter in (mm). (C_e) is the encoder resolution (as in pulses per revolution) and finally (n) is the gear ratio of the reduction gear between motor and the drive wheel. ($\Delta U_{L,t}$ And $\Delta U_{R,t}$) is the left and right wheel travel distance respectively.

$$\Delta P_t = \frac{(\Delta U_{R,t} + \Delta U_{L,t})}{2}$$
$$\Delta \theta_t = \frac{(\Delta U_{R,t} + \Delta U_{L,t})}{D} \qquad (2)$$

Where (ΔP_t) and ($\Delta \theta_t$) are the linear and angular displacement of the center point of the mobile robot.

Using simple kinematic equations we can compute the momentary position of the mobile robot relative to its start position as shown in Figure 4 and Figure 5. This computation technique is called odometry. It is the most basic implementation of dead reckoning used to estimate the mobile robot's relative location. It is simple, inexpensive and easy to realize in different real-time applications. However, because odometry is the integration of incremental motion information over time, it also inevitably leads to the accumulation of errors. Thus, odometry is only suitable for the measurement of short distances. Additionally, it can not provide reliable position information for distances over as little as 10 meters [1 - 3].

The mobile robot's kinematic model can be represented by the following diagram:

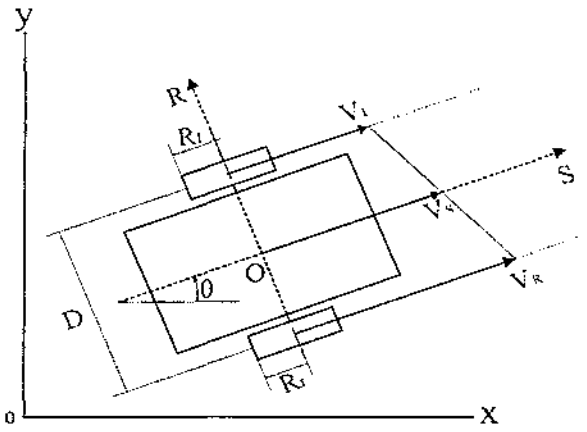

Figure 5: The P3DX mobile robot kinematics in the x – y plane

Odometry of the mobile robot is computed base on the integration of the following kinematic equations:

$$X_{t+1} = X_t - V_S dt.\sin.\theta_{t+1} \tag{3}$$

$$Y_{t+1} = Y_t + V_S dt.\cos.\theta_{t+1} \tag{4}$$

$$V_S = \frac{Wr.Rr + W_L.R_L}{2} \rightarrow \frac{V_R + V_L}{2} \tag{5}$$

$$\theta_{t+1} = \theta_t + \dot{\theta}\, dt \tag{6}$$

$$\dot{\theta} = W, \quad W = \frac{Wr.Rr - W_L.R_L}{D} \rightarrow \frac{V_R - V_L}{D} \tag{7}$$

The angular velocities of left and right wheels are represented by (W_L) and (W_R) respectively. Parameter (D) is the distance between the left and right wheel of the mobile robot. (V_L) and (V_R) are the translational velocities of the left and right wheels. Parameter (θ) is the angle of orientation between the x-axis and the robot's axle (D) as shown in Figure 5 and $(\dot{\theta})$ is the yaw rate of the robot in the $x-y$ plane. The reference home position (0, 0) of the robot is defined to face in the positive y axis direction and coordinates (X_t) and (Y_t) is computed at each time step (t) relative to the defined start location.

The main sources of concern on odometry errors would be orientation errors such as wheel slippage and the wheelbase error. This is because once these errors occur; the orientation errors will grow without bound and cause large position errors in the distance travelled by the mobile robot. Apart from these two examples of odometry error, there are several others and all of these sources of odometry error can be summarized into the following two main categories [1 - 5]:

Systematic errors:

- Misalignment of wheels
- Uncertainty about the effective wheelbase
- Limited encoder resolution
- Limited encoder sampling rate
- Unequal wheel diameters

Non-systematic errors:

- Travel over uneven floor such as carpet.
- Travel over unexpected objects such as bumps, ramps or crack floor
- Wheel slippage due to the following:
  - Unexpected external forces apply onto the robot
  - Slippery floor surface
  - Over accelerating
  - Skidding (over turning)
  - Castor wheel contact with the floor

As discussed earlier, the advantages in using the odometry method are that the onboard encoders are easy to use and inexpensive. And if the encoder errors have been calibrated, using the calibrated odometry method can provide relatively accurate information. However, the main problem here is that the non-systematic errors can appear unexpectedly. Error due to wheel slippage and uneven floor is accumulated without bounds. Thus, in most mobile robot localization systems, absolute positioning methods have been implemented to reduce the effects of odometry errors and to provide better and more reliable position estimation.

Calibration methods to improve Odometry

Many different calibration methods have been proposed and implemented mostly by using Kalman filters to improve the odometry method for robot localization [1, 3 - 7, 16 - 19].

J. Borenstein and L. Feng presented the "UmBmark" test method. It stated that inequality of wheel diameters and the uncertainty of the wheelbase are the primary Odometry error sources. The systematic errors of the robot can be modeled and corrected. The proposed method is aimed at correcting odometry errors on differential drive robots. The robot is configured to travel around a rectangular shape from two different starting points twice. One is to move forward counterclockwise and the other one is to move forward clockwise. At the end of each trail, the distance travelled from the start position is measured and the odometry errors are removed and replaced with corrected odometry parameters accordingly. This is the simplest calibration method available and no other sensors are required during the calibration process. However, in this method none of the non-systematic errors has been considered. This could results incorrect odometry readings caused by different error factors such as, wheel slippage, uneven floor and cracked floor. Another disadvantage is that the start position and the final position of the mobile robot must be identical [2].

The same authors introduce a terminology called the "Gyrodometry". This method combines the data obtained from an external gyroscope with calibrated odometry pose data to reduce the Odometry errors caused by non-systematic errors. Such as when the mobile robot travelled over an uneven floor, it relies on the calibrated odometry position measurement most of the time and substituted the data from a gyro when odometry and gyro data differ substantially. This method has an effective improvement in reducing non-systematic errors caused by uneven floor [3].

Similarly in [1], another odometry calibration method used for localization is proposed. Essentially, the heading angle parameter computed by the odometry method is substituted with the measurement of the gyroscope. And the Kalman filter method is used to allow integration of these two sensor modules. The method also introduces two calibration parameters (β) and (α). The first parameter (β) is used to adjust the variations of the axle length between the wheels. And second parameter (α) is the proportionality factor of the actual speed measured by a tachometer and the speed obtained by encoders. Additionally, they've conducted two different experiments in both

indoor and outdoor environment. Experiment results show an improvement of at least one order of magnitude in accuracy compared to the un-calibrated and unfiltered in the case of using the odometry method.

Another odometry calibration method called the "PC-method" is developed to improve both systematic and non-systematic odometry errors. This method employs three essential steps. The first and second step focuses on the compensation of the systematic errors by modeling the odometry error model and estimating the error parameters. The final step deals with the non-systematic errors. The calibration process involved in estimating the robot's position as it moves along a path using different positioning methods. The obtained results are compared and the odometry model is calibrated. The PC-Method is more accurate than the UmBmark calibration method as it is not an end point localization method and it utilizes all trajectory points for precise error estimation while the UmBmark method uses end point errors only. The main disadvantage of this method is that additional sensors are required for position estimation and initial and finial positions of the mobile robot must coincide similar to the UmBmark method [7].

2.1.2.2 Absolute positioning methods:

The absolute positioning methods rely on evaluating the robot's absolute position and orientation using data information provided by external sensors, such as active or passive visual landmarks, laser scanner, navigation beacons, GPS (Global Positioning System) or map matching. Each of these absolute positioning methods required expensive installation of sensors, high maintenance costs and computational costs. And more importantly, none of these localization systems are particularly easy to realize in most real time robot applications. However, the absolute positioning methods are far more reliable than the local positioning methods. They can localize a mobile robot without any prior knowledge about its position and can cope with situations in which the robot can not recover its location when it has lost track of its position. More importantly, the error growth from measurements of external sensors is substantially less in the case of absolute positioning method. This is because the position of the mobile robot is determined by external sensors and the accuracy of such measurement is usually time and location independent. The noisy data from the encoder measurement is not included and so there is no accumulation of error with distance travelled and time [1 - 9], [22 - 39].

Landmark and beacon based navigation system

As mentioned above, one of the absolute localization methods is based on navigation beacons or landmarks, these beacons and landmarks are usually placed in known locations, they are used to emit signals into the environment and when the mobile robot receives these signals from the allocated beacons and landmarks, it can estimate its current position in the environment.

The landmark system can only localize the position of the mobile robot when there are visible landmarks to the robot's sensors in the environment and such approach is usually quite expensive to set up and modification to the environment is required.

GPS is an example of beacon based navigation system as satellites are used to send out signals and these signals are received on Earth. But one of the major problems of such system is that one cannot keep track of the mobile robot for small distances, this is due to the accuracy of the received position signal is within in 1 to 10 meters from the real position and it requires a clear view of the sky in order to locate its satellites. So such system is not suitable for indoor environments.

In [8], three different indoor optical sensing techniques have been compared and presented as solutions to the problems of object recognition and reliable indoor localization. The three technologies studied included the Visual Pattern Recognition (ViPR), Visual Simultaneous Localization and Mapping (vSLAM), and the optical beacon based localization system (Northstar). It presents the ViPR system as a vision based module that can recognize objects using a single camera; this system is also reliable to changes in light, partial occlusions, rotation and affine distortions in the environment. And in order to adapt changes in the environment, the mobile robot must be able to build a map using its sensors and utilize the constructed map to localize itself. This technique is known as SLAM. The vSLAM system presented uses laser range finders similar to the SLAM method but localization and mapping is achieved using only one camera and the technique is significantly cheaper.

Similarly in [9], three different absolute positioning methods have been examined for acquisition of indoor 2 dimensional ground true data by using an external localization system. The three techniques included the Camera based system using a discretized field, the Northstar localization system and the Stargazer (Hagisonic) system. The presented Camera based technique consists of

three cameras each placed at a different angle in the room to record the positions of robots at 1 second intervals. And by post-processing the images for the location of the robots relative to the labeled floor grids in the images, the position information can be calculated. However, accuracy in pose data and theta dimensions are not accurate, especially in theta. This is mainly due to the fact that the position accuracy varied with the size and position of the mobile robot captured in the image. The presented Stargazer localization system consisted of an active IR detector (Infra-Red camera) placed on the mobile robot and passive landmarks installed on the ceiling. The robot moves with the active detector and each landmark installed on the ceiling have a fixed and known position. Position information is reported in real time and image processing is conducted and processed by the system itself. This system provides accurate pose information, theta dimensions and data frequency relative to each landmark.

Finally, out of the five different absolute positioning methods studied in [8 – 9], the Northstar system is presented as an inexpensive, easy to realize and a reliable and robust solution that uses optical beacons (Projector) in the indoor environment to estimate the pose of the mobile robot using an optical sensor (Detector). More importantly, this system is based on detecting surface reflection of the patterns (two IR spots) rather than direct detection of the emitter, so that the line of sight limitation is minimized. Further explanation on the operation principle and advantages of the Northstar localization system are presented in the next page.

Northstar Localization system

The Northstar localization system is a beacon based optical solution to autonomous indoor navigation of mobile robots that combines a sensor, a processor and an infra-red projector to provide position and orientation information in real-time. A Northstar localization system consisted of the following two components:

- Northstar detector
- Northstar projector

The Northstar system can be configured in the following two different ways:

Navigation configuration: In this configuration the projector is fixed in the environment and the detector is mobile.

Assert tracking configuration: In this configuration the projector is mobile and the detector is fixed in the environment.

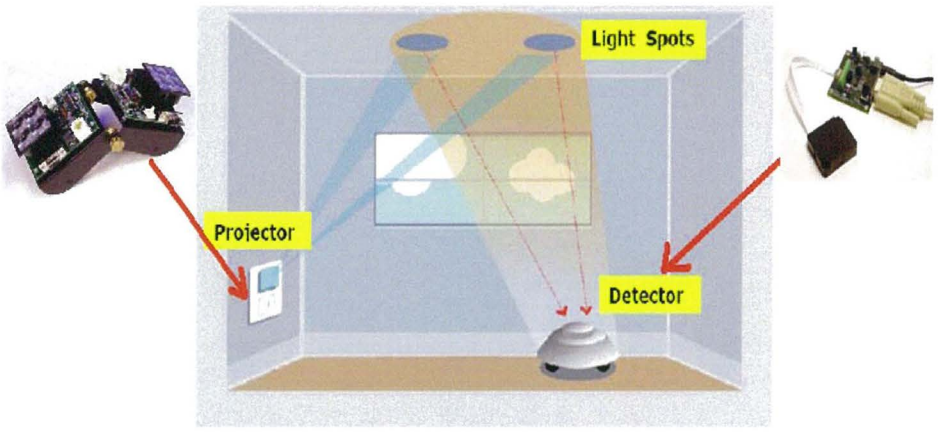

Figure 6: The Northstar localization system configured for navigation

Our research aim is to develop an indoor localization system for mobile robots. Therefore, only the navigation configuration of the Northstar system is used for this purpose.

Operation principle of the Northstar system

The projector consisted of two platforms containing four LED lights that provide a mark-based localization system where two unique infrared beams are projected as beacons onto the ceiling surface modulated at different frequencies as shown in Figure 6. The detector is an optical sensor which uses triangulation to determine the current position, heading direction relative to the unique infrared spot pairs in the surrounding environment at all times. The output of the Northstar localization system is a string consisted of six values, these included the number of spots, x, y, theta and the intensity value for each of the two platforms of the projector. Additionally, Northstar Detectors can monitor up to 20 frequencies at one time and come preprogrammed from the factory to automatically recognize 20 standard frequency ID codes. Room ID codes are used to define spot pairs that can be used by the detector to distinguish between specific regions. The particular frequency which a particular projector beam is transmitted at is set by using the rotary switch on the circuit board.

One of the key advantages of Northstar over other beacon or landmark based localization system is due to its ease of installation of its beacons and it doesn't required modifications to the environment. The active beacon can be easily installed by simply plugging the projector into a wall outlet.

Detector coordinates

This coordinate system is used to detect spot positions and its origin (0,0) is positioned at the center of the detector. Here the detector's field of view is represented by a 65,536 X 65,536 square grid providing x and y data in the ranges of {-32,768 to +32,768}. These units are dimensionless. Detector coordinates are illustrated below in Figure 7.

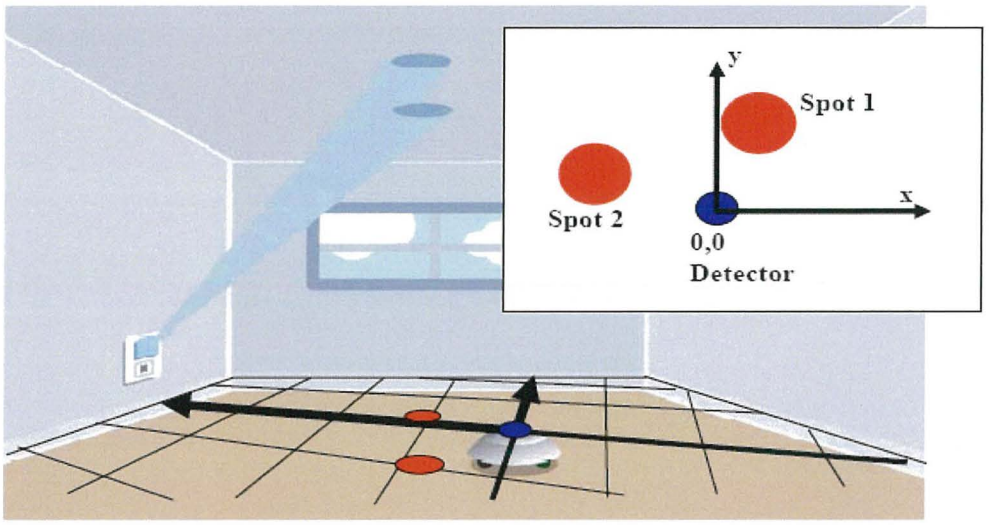

Figure 7: Detector Coordinates illustration

Room Coordinates

This system is used to calculate the detector's position and heading and is fixed to a pre-defined pair of spots. Room coordinates are illustrated below in Figure 8.

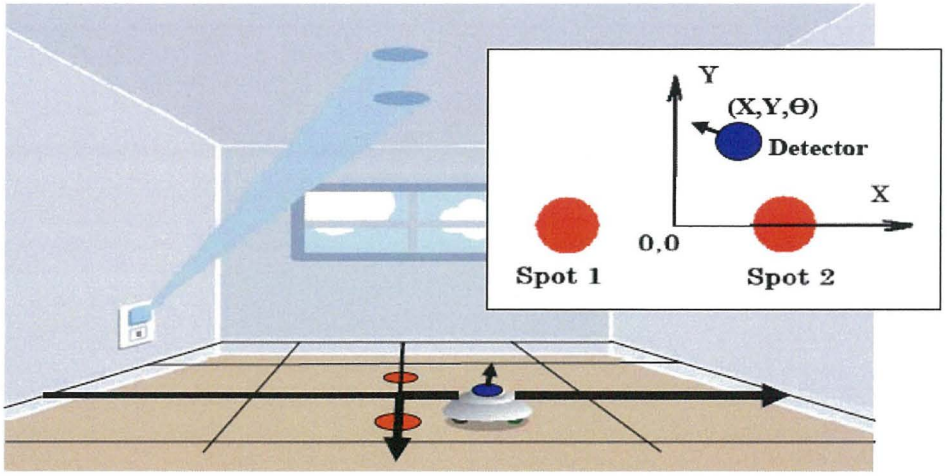

Figure 8: Room Coordinates illustration

Expanding the detector's field of view

It is possible to enlarge the detector's field of view to increase the area of localization for the Northstar system simply by installing two or more projectors. The general procedure to achieve this is shown in Figure 9.

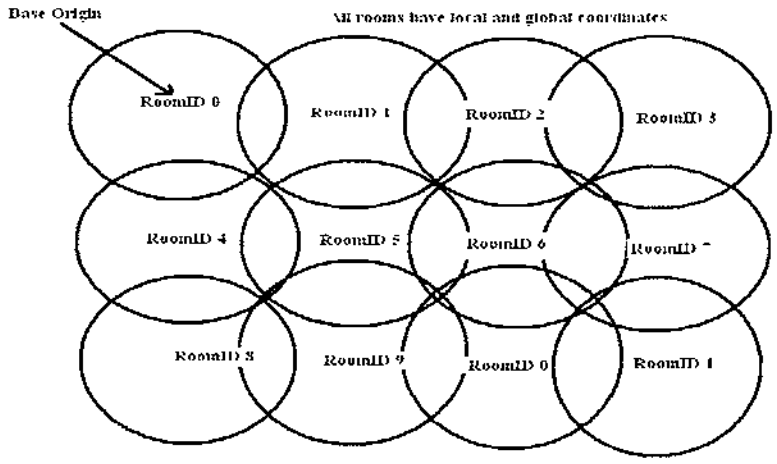

Figure 9: Increasing the field of view by using multiple projectors

When the mobile robot has travelled beyond the optimum field of view in one room, the Northstar system will switch to another room. A translation and rotation is required based on the orientation of each room's coordinate frame with another. And the detector is preprogrammed to automatically recognize 10 standard Room ID codes, so if more that 10 Room ID are needed then placement of each projector needs to be carefully considered to avoid putting rooms with the same IDs in too close proximity. In this project only one projector is used as the provided floor area of the laboratory for testing is restricted to 4m by 4m.

Experiments with the Pioneer P3DX mobile robot shows that localization using the Northstar system with one projector can provide position accuracy of 5cm in x, y and 2 - 5 degrees in heading angle within the optimum field of view which is approximately two meters. Throughout testing, results show the Northstar measurement is slightly skewed when the robot has travelled past two meters from the reference home position (0, 0) and the Northstar system becomes fairly inaccurate. However, once the robot has travelled back into the optimum field of view, the Northstar system was fairly accurate compared to the collected ground truth data. Nevertheless, the accuracy of results could also have been significantly affected by the type of non-flat surface of the ceiling in the lab provided by our university. Further explanation on the test methods and obtained test results are presented in section 4.

2.1.3 Calibration methods to improve speed estimation from encoder readings

In most real time motion control applications, a classical method for speed is through direct differentiation on the obtained position signals, typically the output of an optical incremental encoder. This method is also called Euler approximation. In practice, differentiation operation is performed by taking the difference between the positions in terms of the number of pulses of neighboring samples divided by a sampling time. Normally the sampling time is very small (at milliseconds levels) and the division operation has an effect of amplification on noises inherent in the position signals. In addition, low resolution and imperfections of the encoder contribute to a large measurement noises in the position measurement. Even in a high resolution encoder, a quantization error of maximally half an encoder count is inevitable in the position measurement [10].

Different speed estimation algorithms are also investigated and compared in [13]. It pointed out that the least squares speed estimator eliminates the effect of imperfect measurements, and the TSE and BDE estimators have a better response during the speed's transient period. None of the speed estimators discussed in [13] is suitable for a system that has a large dynamic range of speeds.

Many methods have been proposed to address the shortcomings of the classical method. In [12], Taylor series expansion (TSE) and sliding window method are combined for speed estimation. In the sliding window method, the speed regulator's electrical current sampling time which is always less than the speed sampling time is used for the speed estimation. This allows minimum position data to be used in the speed estimation. Simulation results show that this method has a little improvement over other methods like Taylor Series Expansion (TSE), least square method and second and third order Backward Difference Expansions (BDE) . No experimental results are reported and errors caused by encoder imperfections are not considered.

In [14], it is found that by tracking backward some steps in the position signals, the speed estimation error in very low speed region can be reduced. The number of backward steps depends on the resolution of the encoder and the speed. The relative accuracy of speed estimation is inversely proportional to the number of backward steps. Following this method, at the high speed region more encoder pulses will be generated for a given time period, and at the low speed region,

more steps have to be traced back for the same number of pulses. However, longer time delay resulted is undesirable in real time applications. An empirical time stamping concept is also introduced in [14] in which a high resolution clock is used to capture and store the encoder events consisting of pulses count and the corresponding time instants. By increasing the time span covered by the stored encoder events, the oscillations caused by encoder imperfection measurements are reduced. Considering that the amount of events is limited by the storage capacity, a so called skip option is introduced to skip a fixed number of events in between the chosen stored events. This skip option also performs a low pass filtering function with a spatial cut off frequency that is dependent on the momentary speed. Some other methods, mainly based on Kalman filter, have been proposed to reduce both stochastic errors and unknown system parameter errors [16, 17, 18, and 21]. In [16], a second order sliding mode with Kalman filter is used to detect and control speed and position of a Brushless DC motor with a low resolution encoder in very low speeds. A Kalman filter is used to estimate the position, speed and load torque from the motor current and the second order sliding mode outputs. It has been found that the methods based on Kalman filter perform better than classical methods and their modified versions.

From the fact that sampling time has error amplification effect on speed estimation, in this project we have proposed a simple approach to adjust sampling time considering the magnitude of actual speed. Though the actual speed at a sampling instant is not known, this does not cause much problem as its estimated value is available from the estimation at the previous sample. A formula for sampling time adjustment is proposed and presented in section 4. The accuracy of the obtained results is evaluated against the readings of an external optical tachometer.

3 The robot model and software development process

This section presents the background of the robot model and details the software development process for robot control.

3.1 Background on the robot model

A description on the serial communications of the robot model used with the developed localization system is presented in this section.

This project would not be possible without a mobile robot. There are many different types of mobile robots capable of moving around and fulfilling a task in a given environment. Mobile robots are not fixed to one physical location and they can access places that are unreachable by human that thus they can do things that human can not physically do them self. For example, a mobile robot with self navigation capabilities can do jobs such as delivering medicines in a hospital which requires close contact with the infected patients and therefore reduce the risk of facing medical staff.

The Pioneer 3-dx ActivMedia mobile robot used in this project as shown in Figure 10 is a differential drive robot. It includes an embedded computer; two independently drive wheels, a castor wheel for stability, battery power, position speed encoders, integrated sensors and accessory like a high resolution camera. The robot is controlled by the onboard microcontroller and the robot server software.

Figure 10: The Pioneer3-DX mobile robot used in this project

The mobile robot P3DX operates as the server in a client-server environment. The server is embodied in the operating system software call ARCOS (Advanced Robotics Control and Operations) embedded on the robot's microcontroller. It manages the low level tasks of robot control and operation. The robot server does not perform robotic tasks, but it is executed by the client instead.

The client software provides the high level control operation and it must be running on a computer connected to the robot's microcontroller. This can be the robot's onboard computer connected to the robot's microcontroller via serial communication. This serial communication between microcontroller and the client can be established through the RS-232 port. Or alternatively, a remote networked computer can be used to act as the client and the robot's onboard computer is used to run the server program. To establish such communication link between the two computers, TCP/IP control protocols and wireless network are used to exchange information with onboard computer of the mobile robot.

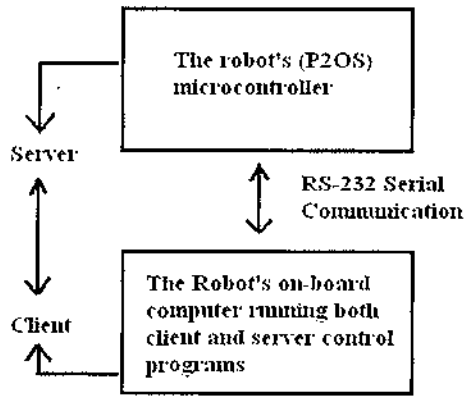

Figure 11: Controlling the robot from the on-board computer via an RS-232 serial cable

Figure 11 shows the client and server control program are both running on the robot's on-board computer. The client program can request and send services from the server program through the RS-232 serial cable.

There are four types of serial connection from Robot server to PC client and they can be listed as follow:

1. Robot tethered to an off-board PC
2. Robot communicating wirelessly with off-board PC via radio modem
3. Robot carrying a tethered laptop
4. Robot with an embedded onboard computer

In this project the wireless option 2 is used to remotely control and communicate with the mobile robot via a wireless network adaptor. The overall structure of wireless communication with the robot is presented in Figure 12.

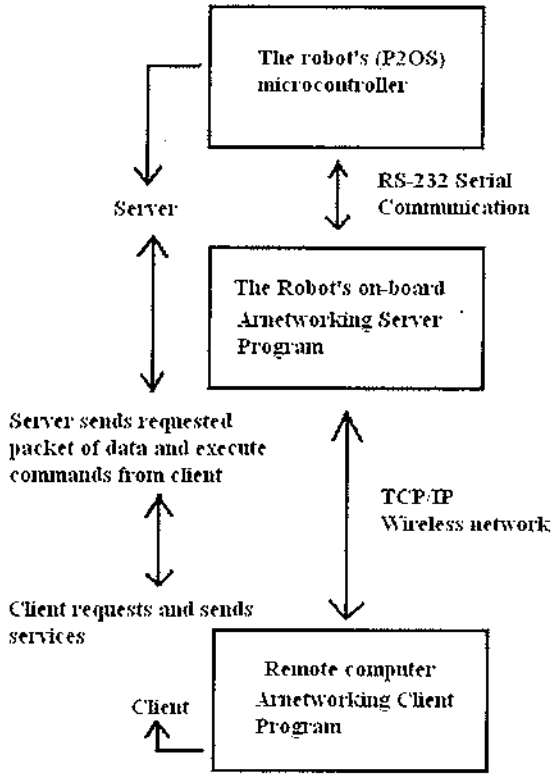

Figure 12: Controlling the robot remotely using the ArNetworking library

Figure 12 shows the mobile robot is controlled remotely by using the ArNetworking library via a remote networked computer with a wireless network adaptor installed. The ArNetworking library is used to write the server and client programs. It is a more complete framework than the available ArSocket library (socket programming) written in C++ for sending updates to client and receiving commands and other sensor data from it. ArNetworking includes many services already available for the server to use, this including the robot's status and position, sensor data and map data. However, one of the disadvantages is that this software that comes with the robot is very complex. It requires extensive study of its existing libraries and functions.

3.2 Background on robot control flow chart design

This section describes the development process of the client ↔ server architecture for robot control and for localization and navigation. As mentioned earlier, one of the ways to establish wirelessly communication between the mobile robot and a remote computer is to use ArNetworking library. The ArNetworking library provides data requests and updates between client and server applications via TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) and robot communication is based on using the packet concept.

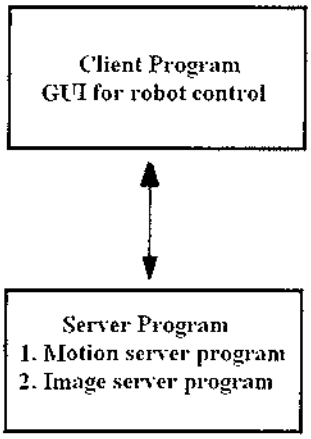

Figure 13: The Client Server architecture for robot localization and navigation

The server side contains the image server program and the motion server program. This program is written in VC++ programming language. The motion program controls the robot's movements and obstacle detection. It is created for collecting motion commands from the client program and to execute these motion commands accordingly. And the image server program encodes the image taken by the framegrabber of the server into a run length encoded image and transmits over the network using the ArNetworking library. The source code of the image server program is provided with the ARIA software that comes with the mobile robot. The client and server control flow charts can be represented by Figures 14 and 15 respectively.

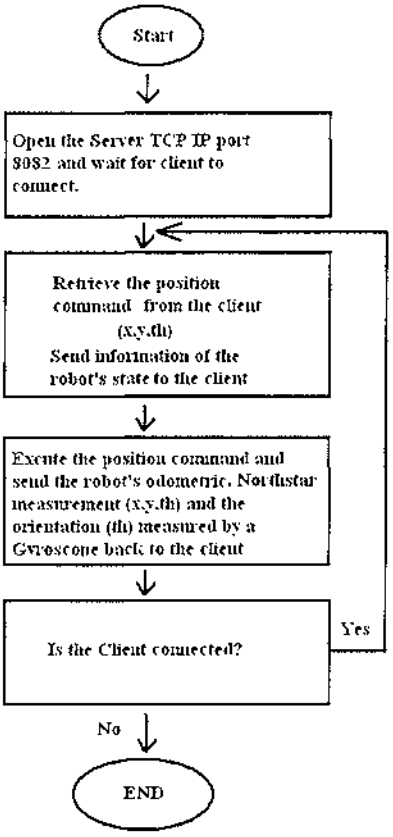

Figure 14: The server program control flow chart for localization

Figure 14 shows the server program establishes connection and connects to the robot simulator or the real root and then opens port number 8082 and wait for a connection from the client program. Once a connection is established with the client program, the server program starts an infinite loop that sends the client program information of the robot's state. These include the robot's position, heading direction, translational and rotational velocity. If the server program has been interrupted or connection with the client program has been disconnected during the read operation, the client program will not be able to receive the information of the robot's state and both programs will have to be restarted.

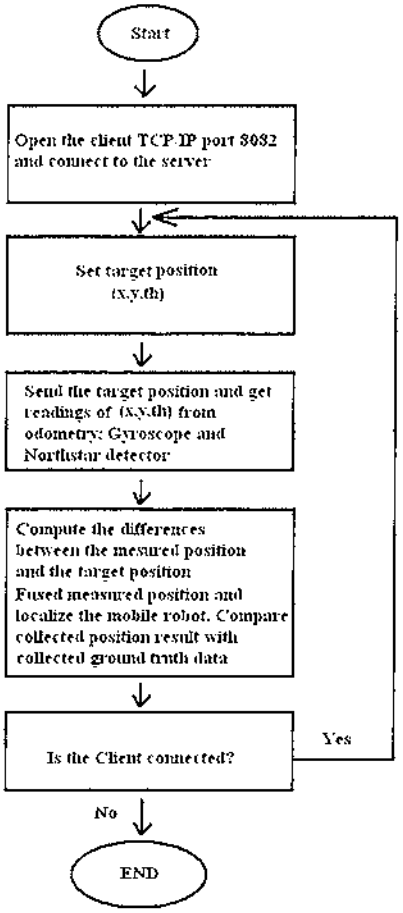

Figure 15: The client program control flow chart for localization

The client program is mainly developed to send motion commands and control algorithm instructions to the server program and to retrieve the Odometric, gyroscope, Northstar detector measurements from the server program which is running on the robot. The client program is also written in VC++ programming language. The program has been divided into several class files, each designed with a purpose to handle a different task. And to achieve this, the program is multi-threaded by using multiple worker threads to operate concurrently and to receive the robot's state information and the external sensor's information. The entire Client program can be separated into the following five classes:

Client: This establishes socket communication with the motion server and image server running on the mobile robot.

Map: This will be display on the simulator on the GUI (Figure 16), it shows the robot's position and heading direction information in two dimensional Cartesian coordinate display.

Camera: This camera class is developed with the ARIA (Advanced Robotics Interface for Applications) software that comes from the manufacturer of the robot model. This class handles and establishes connection to the camera and display an encoded image received from the image server. This camera class is mainly included in the program for future research purposes.

Robot: This class is actually the program version of the physical robot. It is an abstraction of the real robot. It includes the robot's properties for display.

Localization and Navigation class: This class integrates all classes and implements the user interface with the program. Two separated worker threads are created to receive the robot's state information and the external sensor's information such as the readings form the Gyroscope and the Northstar system. And the main thread of the client program waits for user events and sends out motions commands to the motion server program accordingly.

3.3 The developed GUI for robot control

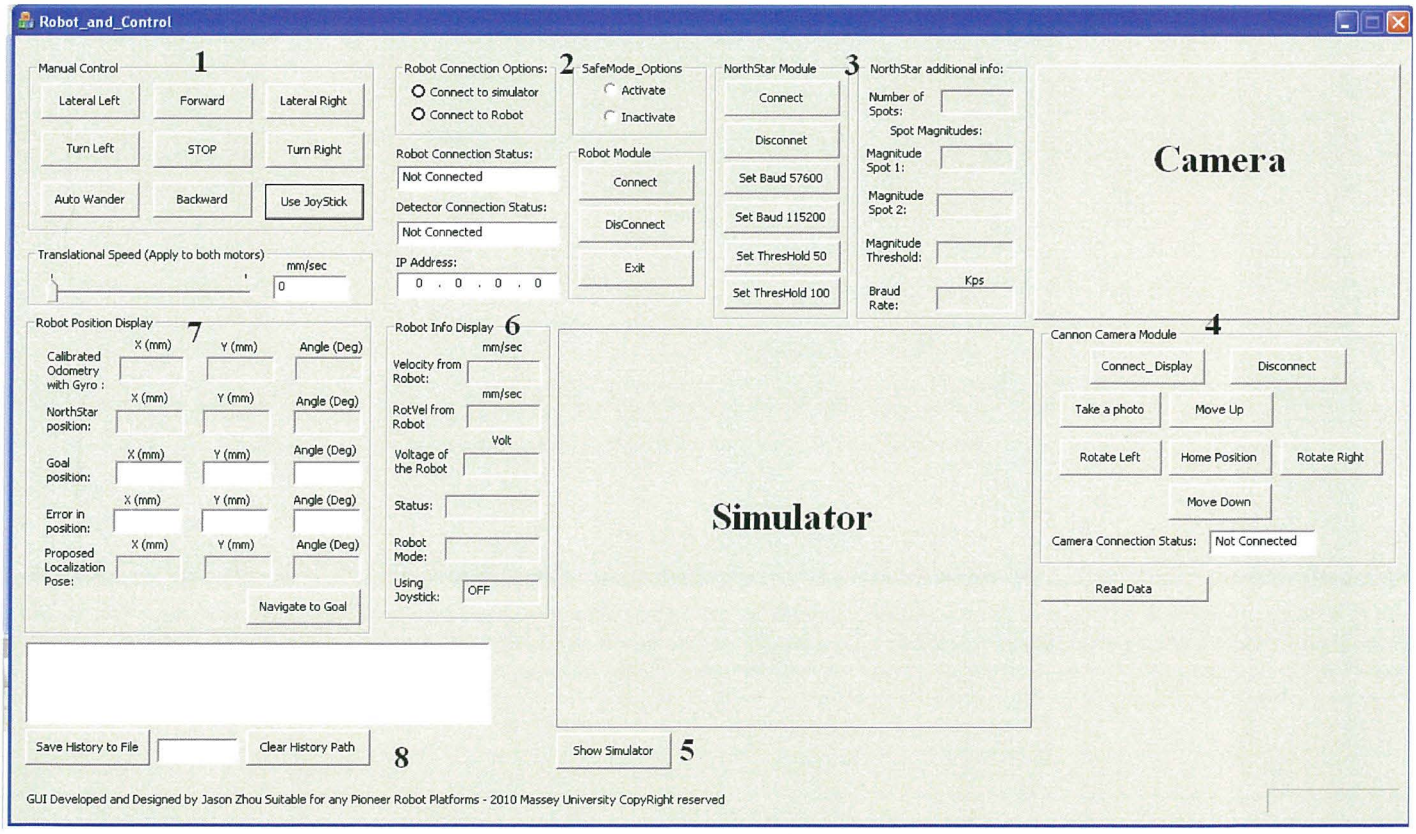

Figure 16: The developed GUI for robot control

The robot and control GUI (Graphical User Interface) is developed using MFC (Microsoft Foundation Class) application VC++ programming language for simulation of the mobile robot and real robot interaction. One of the main purposes for the design of this GUI is to make simulation of the mobile robot before experiments on the real mobile robot for safety purpose. The developed GUI has many different features and it includes many sub-modules for testing purposes. Each of the labeled modules is explained as follow:

Module 1 shows the manual control option of the robot, with each action button performed the mobile robot in the simulator also moves accordingly. The translational speed track bar allows the user to choose the travelling speed. This module also allows the user to use the joystick option to manually navigate the robot from one location to another. There is also an "auto wander" button, when this button is selected it overwrites all manual control commands and navigate the robot autonomously on its own with the specified translation speed selected using the track bar.

Module 2 shows there are two robot connection options available. The first option connects the GUI to the robot simulator and the status of the robot which will be displayed in module 5 and the second option connects and establishes wireless connection with the real mobile robot P3DX. Only one of the two options can be selected at a time. This module includes the safe mode option, when this is activated, the mobile robot will not bump into obstacles in its path. Essentially, it uses the eight sonar sensors installed on the front to sense its distance from an object. When the estimated distance is within 100mm, the mobile robot will paused for one second and move 200mm backward and navigate itself away from the obstacle. Also the micro-switches installed around the mobile robot employed the same navigation algorithm provided with the ARIA software.

Module 3 shows the section where initial connection to the Northstar Detector is started. The user can set control parameters such as the baud rate and Threshold when establishing connection with the detector. The spot magnitudes, the details on the number of spots and the specified baud rate are also displayed in this area.

Module 4 shows image processing of the images captured by the camera. When the connection of the camera is established, this module allows the user to take photos of the environment and tilt the direction of the camera. There is a need to develop software which calculates the robot position data from the image processing results that can be used for simulation as well as the real mobile robot.

Module 5 shows the prior simulation before real robot interaction.

Module 6 displays robot info on translation velocity, rotational velocity, voltage, operation status, robot mode of the mobile robot.

Module 7 shows the estimated robot position and orientation is given by (X, Y, Θ), these figures are computed using different methods and differences are noted. The goal position can also be specified by the user and the differences in position estimated by the developed position estimator are displayed. The pose results displayed in module 7 are also displayed into the output window of module 8.

Module 8 shows when initial connection to the simulator, the real robot or Northstar detector is established, data streaming into the remote computer is displayed into the output window. Essentially, the history of pose coordinate result from different methods used is displayed. The user can specify the filename and click the "save history to file" button to create a text file with collected pose results for further analysis.

4 Proposed approaches and results

In this section, methods of testing and calibration, experiments and results are presented and explained in details.

4.1 The development process

The majority of the project was mainly software based but also includes some hardware interfacing. The general design steps of the project can be described by Figure 17 as shown below.

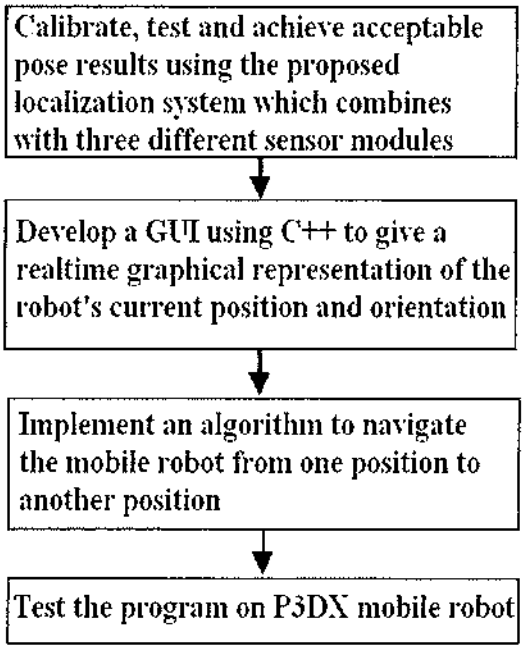

Figure 17: The design process flowchart of the project

4.2 Calibration methods used to improve odometry

Experiments with the Pioneer p3dx shows that localization using the un-calibrated odometry method that relies on velocities estimated by the encoders can produce error up to 15% in the position estimates. Throughout testing, we have also found that a significant portion of systematic error comes from the radius of the wheels and the wheelbase. This is mainly due to when turning, the given axle length (D) of the mobile robot's wheelbase do not pivot at the centre between the two wheels. Therefore the effective axle length changes slightly during a turn. The changes of parameter (D) affects the accuracy of the yaw rate computed in Equation 7 using the velocities returned from the encoders, this causes the accumulation of orientation errors to grow without bound over time. And eventually the position estimates returned from the un-calibrated odometry method is substantially different comparing to the ground truth position of the mobile robot.

In our experiments, the following calibration procedure is conducted to compensate systematic errors associated with the wheelbase and the wheel radius differences of the Pioneer P3dx mobile robot. Figure 18 illustrates how the calibration method is implemented into the robot's encoder model. Similar to the methods used in article [1, 3, 5], an additional calibration parameter (&) is introduced to compensate the wheelbase error incurred when turning the mobile robot.

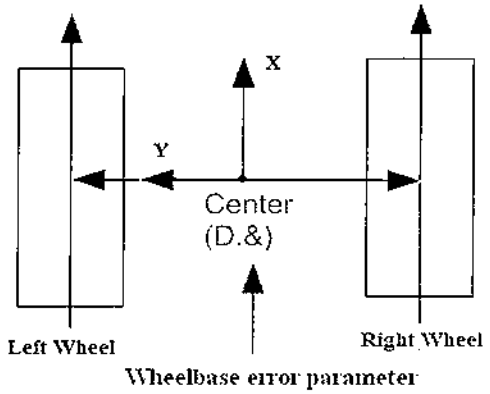

Figure 18: The wheel layout diagram

A few assumptions were made when testing the Pioneer Pd3x model with the calibrated Odometry method. To name a few, when finding parameter (&) the radius of the two wheels (R) are assumed to be the same, angular velocities for both left and right wheels are assumed to be equivalent to

$W_L.R_L$ and $W_R.R_R$. The calibration parameter (&) for the robot's wheel base is assumed that $(D >> D.\&)$. Therefore, the yaw rate computed by the velocities of the robot model is now represented by the following equation:

$$\dot{\theta} = \frac{W_R.R_R - W_L.R_L}{D.\&} \rightarrow \frac{V_R - V_L}{D.\&} \tag{8}$$

To determine the optimum value for the calibration factor (&), a trail and error method is adopted. Essentially, ten different readings are taken from the orientation estimated by velocities returned from the encoders, and the actual orientation angle is measured using a protractor. Differences are compared and noted. E.G. if the difference in the measured angle is five degrees on average and by referring to Equation 8, it allows us to determine the value of (&). The improvement and effectiveness of this calibrated odometry method is presented in the experiments and results section.

4.3 Calibration method to improve Gyroscope readings

To further improve estimate of the yaw rate $(\dot{\theta})$ in the x-y plane, a gyroscope with a range of $-180°/s$ to $+180°/s$ is placed on the platform of the mobile robot P3dx for estimating orientation. Rotation angle measured from three different sensor modules (Northstar system, calibrated odometry and gyro) are compared, eventually heading angle data returned from the gyroscope is fused with the calibrated odometry measurement and the measurement of the Northstar system using the Kalman filter algorithm.

Throughout testing, before we can use the measurement from the Gyroscope, it is important to calibrate its reading and to verify its error model so that its obtained reading can be improved. As stated in [5, 53, 54], the random bias and the scale factor error are the dominant causes of gyroscope errors. Generally, there is some offset caused by manufacturing imperfections and the gyroscope signal is not always maintained at zero even though there is no input and more importantly this bias error would reduce the accuracy and affects the performance of the gyroscope. The random bias error can be modeled as a constant and it can be represented by the following differential equation:

$$\frac{dx(t)}{dt} = 0 \tag{9}$$

The scale factor error can be modeled as the first order Markov process and it can be represented by the following equation:

$$\dot{\psi}_{gyro} = (S + B_S).\Omega + B_{rb}(1 - e^{\frac{t}{T}}) + v(t) \qquad (10)$$

Where (Ω) is the true angular rate, (S) is the gyroscope scale factor, B_{rb} is the random bias error, B_S is the gyroscope scale factor error and finally $v(t)$ is the white noise. The Gyroscope error can be represented by the following equations using the perturbation methods. Where parameters (V) and (ω) are the white noise.

$$\delta\dot{\psi} = \delta B_S \Omega + \delta B_{rb}(1 - e^{\frac{t}{T}}) + v \qquad (11)$$

$$\delta\dot{B}_S = -\frac{1}{T_S}\delta B_S + \omega$$

$$\delta\dot{B}_{rb} = 0$$

In our experiments, our primary focus is to minimize and to reduce the random bias error of the gyroscope. To achieve this, before using the gyroscope for estimating orientation, a set of its measurements has been collected to obtain the mean value which is about $(0.3\,°/s)$. These measurements are taken when the mobile robot is stationary. Results show this mean value doesn't change significantly with time when testing the gyro with the Pioneer mobile robot P3DX. Therefore each time a reading is collected from the gyroscope the bias value of $0.3\,°/s$ is deducted from the gyro signal.

4.4 Propose approach to improve speed estimation from encoder readings

Experiments with the P3dx mobile robot revealed that localization which relies on the un-calibrated odometry method can produce error up to 15% in the position estimates. And referring to section 2.1.3 in the literature reviews, we believe that by improving speed estimation from encoder readings would improve the accuracy of localization methods that rely on the velocities returned by the encoders of the mobile robot. And as stated in [1 – 7], some of the main reasons for this could be the limited precision of the encoders, the low sampling frequency of the encoder's values and the inaccessibility to raw data which can be used to compute the angular velocities of the wheels. Therefore, in this project we have also developed an approach to improve real time speed estimation from the readings of encoder through varying sampling time. It is shown that with a proper selection of the sampling time considering the trend of actual speed, the effects of inconsistent position measurements and potential errors caused by encoder imperfections on speed estimation are reduced.

As mentioned and discussed in the literature reviews, the classical speed estimation method involved in taking the difference of position readings of two discrete sampling instants, and then having it divided by the sampling time to obtain the estimated speed. The position signal can be obtained from the number of pulses counted at the sampling instant using the following equation:

$$\theta_K = \frac{N}{R} \cdot 2 \cdot \pi \tag{12}$$

Where N is the number of counted pulses and R is the resolution of the encoder pulses per revolution.

In discrete time, the speed can be derived by the following equation:

$$\Omega_K = \frac{\theta_K - \theta_{K-1}}{T_s} = \frac{2\pi(N_K - N_{K-1})}{RT_s} \tag{13}$$

Where T_s is the sampling time, θ_K and θ_{K-1} are the position readings, and N_K and N_{K-1} are the pulse counts at the neighbouring instants.

If $\Delta\theta$ is the error in position readings, then the error in speed estimation can be represented by the following:

$$\Delta\Omega = \frac{2.\Delta\theta}{T_s} \tag{14}$$

Error amplification effects of the sampling time are obvious from the above equation. It seems that the bigger the sampling time, the better in terms of reducing errors. This is only one sided conclusion, as the sampling time must be chosen considering the resolution of the encoder and the actual speed. When the speed is low, it must be long enough to make sure no pulses are missed, but when the speed is high, it should be set at a reasonable length to avoid saturation in readings. Our approach is to set up a mechanism in which the sampling time is adjusted according to the actual speed through either of the following equations:

$$T_N = K_{EXP}.e^{\frac{-w}{w_0}} \tag{15}$$

$$T_N = T_{s0} - K_L.w \tag{16}$$

$$T_N = \frac{K_{NL}}{w}.w_{n-1} \tag{17}$$

Where T_N is the sampling time, w_0 is the maximum speed and K_{EXP}, K_L and K_{NL} are the tuning parameters, T_{s0} is the initial sampling time and w is the speed estimated at the previous sampling instant. They all show the trend that the higher the speed, the smaller the sampling time. This method works for the encoders with different resolutions. The main task remaining is how to choose the tuning parameters and the initial sampling time in practice.

4.5 Experiment to improve speed estimation from encoder readings

In this section, an experimental set-up to implement our proposed approach to improve speed estimation presented in 4.4 will be described. Experiment results are also presented to verify its effectiveness and its improvement over the classical method.

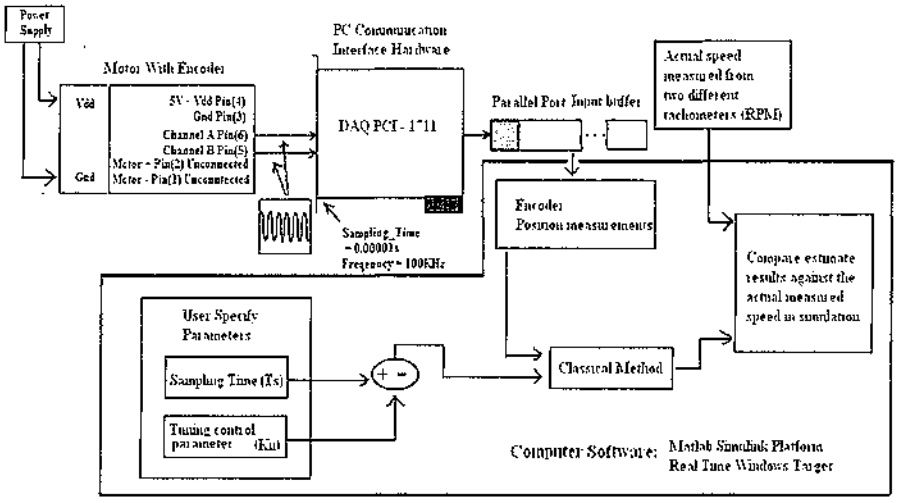

Figure 19: The overall configuration of the speed improvement experiment

The set up of the testing platform as shown in Figure 19 consisted of a DC motor mounted with a rotary incremental encoder on the motor shaft. The encoder output is connected to the PC interface PCI-1711 data acquisition (DAQ) card that provides multiple measurement and control functions. A variable power supply is used to adjust the input voltage applied to the motor, which in turn runs at different speeds. In this experiment 3V and 5V are used as inputs to produce two speeds. A tachometer is used to capture the actual speed for comparison purpose. The initial sampling period is set to be $1ms$. The incremental optical encoder of the motor also provides a resolution of 12.265625×10^{-3} radian / pulse.

The following table includes the tuning parameters used in experiment. These selected figures are empirically tested until the best possible speed estimation results at both low and high speed regions have been achieved.

Speed / Input Voltage	K_{EXP}	K_L	K_{NL}	T_S (sec)	W_O (Average Max Speed in RPM)
At Low Speed / (3V)	0.0024	1.27E-06	0.001	0.001	1620
At High Speed / (5V)	0.003	1.65E-08	0.0009	0.001	2866

Table 1: The control parameters used on the proposed method

4.5.1 Results obtained at low speed using the proposed method

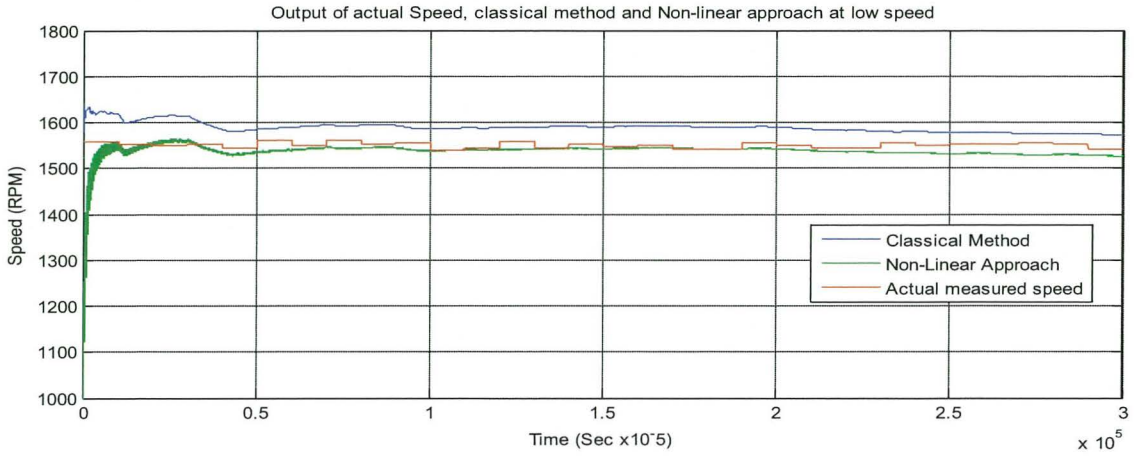

Figure 20: Speed estimation using the Classical method and the non-linear method at low speed

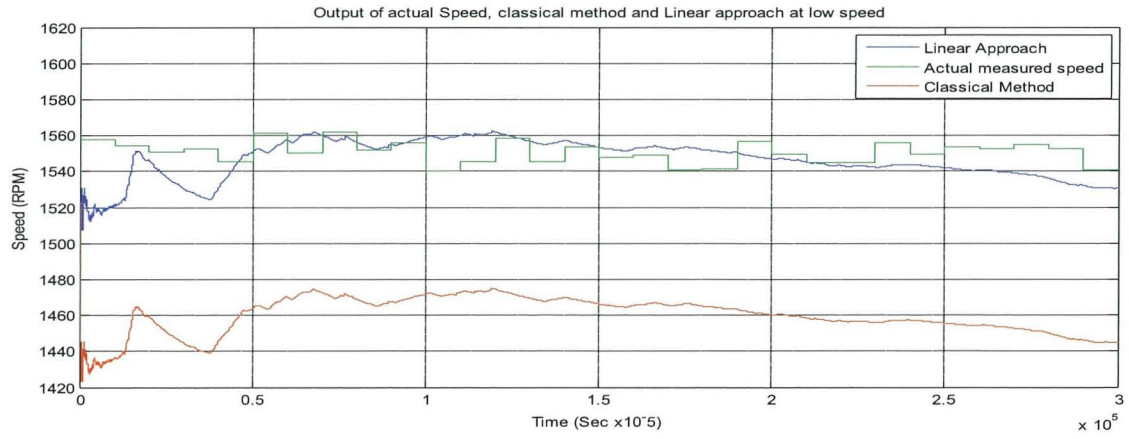

Figure 21: Speed estimation using the Classical method and the linear method at low speed

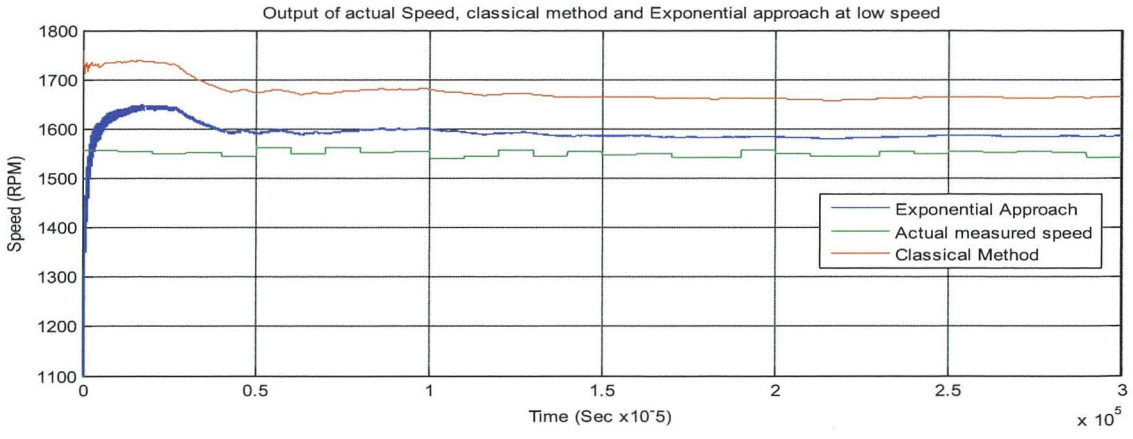

Figure 22: Speed estimation using the Classical method and the exponential method at low speed

4.5.2 Results obtained at high speed using the proposed method

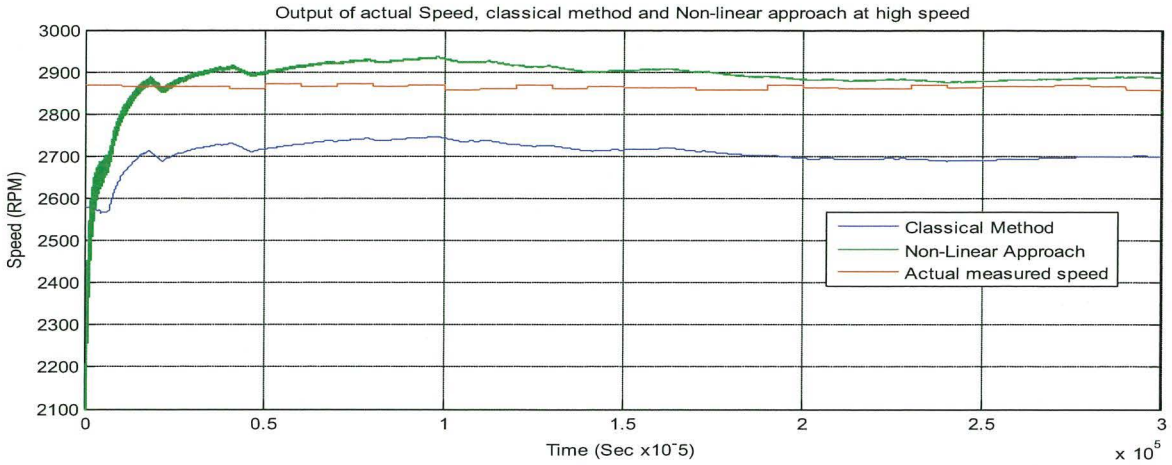

Figure 23: Speed estimation using the Classical method and the non-linear method at high speed

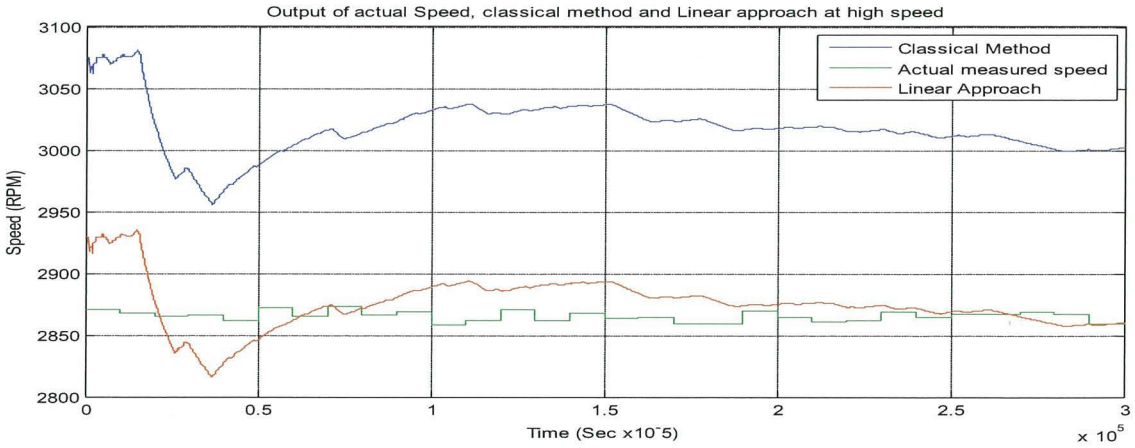

Figure 24: Speed estimation using the Classical method and the linear method at high speed

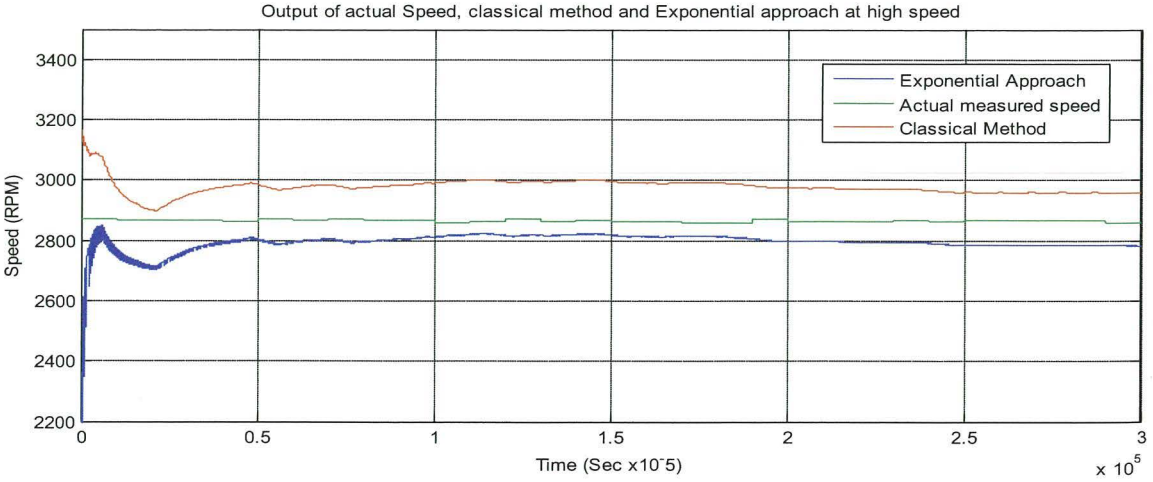

Figure 25: Speed estimation using the Classical method and the exponential method at high speed

The experiment results show that our proposed method produces the smallest possible error covariance. The speed estimation has improved significantly by implementing the proposed linear approach presented in Equation 16. The output converges to the measured reference speed in roughly 0.5 second. And the reference speed measured by the external optical tachometer is used to compare against the obtained measurement from the three approaches of our proposed method. And to validate and to confirm the advantage and effectiveness of our proposed method, Equation 18 is used to check the relative speed error percentage of the proposed method at both high and low speed regions.

The relative speed error percentage equation can be derived by the following equation:

$$Error\% = 100(\frac{\hat{W}_K - W_K}{W_K}) \tag{18}$$

Whereas (W_K) is the reference speed measured by the tachometer and ($\hat{W}_K$) is the estimated speed of the proposed method. Firstly, Equation 18 is applied onto the classical method to work out the relative speed error percentage and the same procedure is applied to the three approaches of our proposed method. Comparison is made and the differences are noted.

Relative speed Error % (Min to Max)	Classical method	Linear Approach	Non-linear Approach	Exponential Approach
Error percentage at low speed	3.5% to 5%	1% to 1.2%	1.2% to 1.5%	2.5% to 3%
Error percentage at high speed	4% to 5.5%	1% to 1.1%	1.1% to 1.4%	2.8% to 3%

Table 2: Summary of the relative speed error percentage for the methods used

The summary of relative speed error on Table 2 shows that all three proposed approaches improved the relative accuracy of speed estimation over the classical method. And the linear approach out of the three cases produced the minimal relative speed error at both low and high speed regions. It can be concluded that the proposed linear approach speed estimator performs significantly better than the classical method with minimal speed error produced at both low and high speed regions.

4.6 Kalman Filter

This section discussed the basic and the purpose of Kalman filter and its implementation for sensory integration used for localization of mobile robots.

The Kalman filter technique is mainly selected due to its ease of implementation and its ability to update multiple pose data continuously. Kalman filter was developed by Rudolf E. Kalman in 1960. It is essentially a recursive state estimator similar to the recursive computation of the least square algorithm which uses the output as its input. It was initially developed as a solution to solve the discrete data linear filtering problem. Furthermore, Kalman filter has been the subject of extensive research and application, especially in the field of localization, navigation and system integration [15 - 19].

The general configuration of a Kalman filter governed by a linear dynamic system is often represented in a state space model as shown by the following two equations:

$$\text{Process Model: } X_{n-1} = A.X_n + B.U_n + W_n \tag{19}$$

$$\text{Measurement Model: } Z_n = H.X_n + V_n \tag{20}$$

A Kalman Filter is consisted of the following three stages:

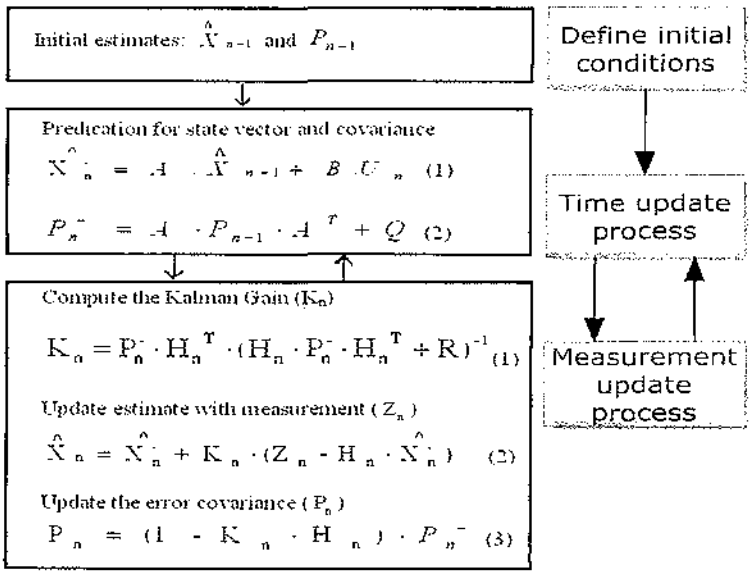

Figure 26: The direct Kalman filter algorithm

Once the system is defined correctly, an improved estimate of the system state vector (X) and its covariance (P) can be computed base upon the information in a new observation. However, when the system model (A) is inaccurately or incompletely determined, the Kalman filter will be affected by the inconsistent data points with the chosen system model and eventually lead to inaccurate estimations. The Extended Kalman Filter (EFK) is formulated for nonlinear state estimation. However, significant estimation error may occur, due to the propagation of uncertainty through the nonlinear system. Methods that are developed to simultaneously to estimate the state vector and the noise parameters of the system have suboptimal performance and are potentially unstable [18, 19].

4.6.1 Implementing the Kalman Filter algorithm

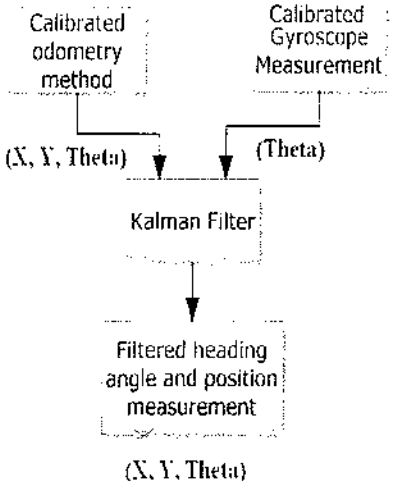

Figure 27: Integration of the calibrated odometry method and the calibrated gyroscope

The Kalman filter algorithm is implemented to enable integration of the calibrated odometry and the calibrated gyro. The integration of the two measurements is called the Gyrodometry method. This method relies on the calibrated odometry position measurement most of the time and substituted the data from a gyro when odometry and gyro data differ substantially. Experiments with the Pioneer p3dx robot shows that localization using the Gyrodometry method can produce $4.5\% - 6.5\%$ error in the position estimates.

The following procedure is implemented to combine the measurement from the calibrated odometry method and the calibrated gyro.

The measurement Vector can be represented by the following:

$$Z_K = \begin{bmatrix} V_L \\ V_R \\ \dot{\theta} \end{bmatrix}$$

Whereas (V_L) and (V_R) are the translational velocities of the left and right wheels.

Basically, the yaw rate $(\dot{\theta})$ measured by the Gyroscope is used instead of the yaw rate returned from the calibrated odometry method. Thus, the prediction Vector can be represented by the following:

$$\hat{Z}_K = \begin{bmatrix} \hat{V}_L \\ \hat{V}_R \\ \hat{\dot{\theta}} \end{bmatrix}$$

$$r = Z_K - \hat{Z}_K$$

Whereas:

(r) is the residual that is the actual measurement (Z_K) minus the measurement estimate $(\hat{Z}_K)$

The time update (prediction) procedure can be represented by the following equations:

$$\hat{X}_K^- = A.\hat{X}_{K-1}$$

$$P_K^- = A.P_{K-1}.A^T + Q \qquad (21)$$

Whereas:

(A) is the system matrix, in this case it has a dimension of $[3x3]$. The system matrix (A) after applied the calibrated odometry in Equation 6 is given as follow:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/D.\& & 1/D.\& & 0 \end{bmatrix}$$

The measurement update procedure can be represented by the following equations:

First work out the Kalman Gain (K_K)

$$K_K = P_K^-.(P_K^- + R)^{-1} \tag{22}$$

$$\hat{X}_K = \hat{X}_K^- + K_K.r \tag{23}$$

$$P_K = (1 - K_K).P_K^- \tag{24}$$

R is the measurement error matrix and (P) is the error covariance matrix.

The final stage is to determine the values for (Q) matrix which is the system noise covariance matrix and (R) the measurement noise covariance matrix. The (Q) matrix is also used as a tuning parameter. The values collected for the R matrix are based on the sensor specifications on both optical encoder and the Gyroscope as well as based on empirical observations through experiments. The following (Q) and (R) are used in our experiments:

$$Q = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1^{-4} \end{bmatrix} \qquad R = \begin{bmatrix} 0.952 & 0 & 0 \\ 0 & 0.952 & 0 \\ 0 & 0 & 10^{-5} \end{bmatrix}$$

4.7 Propose techniques for localization

This section presents two different approaches developed in our proposed method for localization and details on the two approaches are discussed, results are also presented to verify the validity of each method.

Figure 28: Testing the completed localization system with the mobile robot

The completed localization system is tested with mobile robot P3dx. It carries the Northstar detector and a calibrated gyro on its platform as it navigates on the floor space in our laboratory. And the developed localization system can successfully provide information of the robot's current location and orientation relative to this indoor environment as presented in Figure 28.

Throughout experiments with the Northstar system, with only one projector installed we've found that the detector's optimum field of view to be within two meters from the origin (the area between the two IR spots from the projector). And when the detector is within two meters of the origin, the Northstar is reasonably accurate. Obtained readings at this distance were correct to within an average value of 6 centimeters and 4 degrees. Inaccuracies of up to 9cm in measured position and 11 degrees of heading angle error are obtained at two meters distance. At 2.5 meters from the origin, readings obtained from the detector deviate up to 40 centimeters and 22 degrees from actual

position. We've proved that position measurements taken at over two meters from the origin are unreliable and not useful.

To overcome such limitation with the Northstar system, we have proposed two different approaches in our proposed method for localization and each approach consisted of the relative pose measurement obtained from the Gyrodometry method and the absolute pose measurement obtained from the Northstar system. Furthermore, the Kalman filter is used for integration of sensory data returned by the three sensor modules implemented in each approach. And the conditional statements are used to analyze and apply the collected pose measurement for localization accordingly. The overall structure of the two proposed approaches is presented in Figure 29 and Figure 30 respectively.

4.7.1 Propose method one

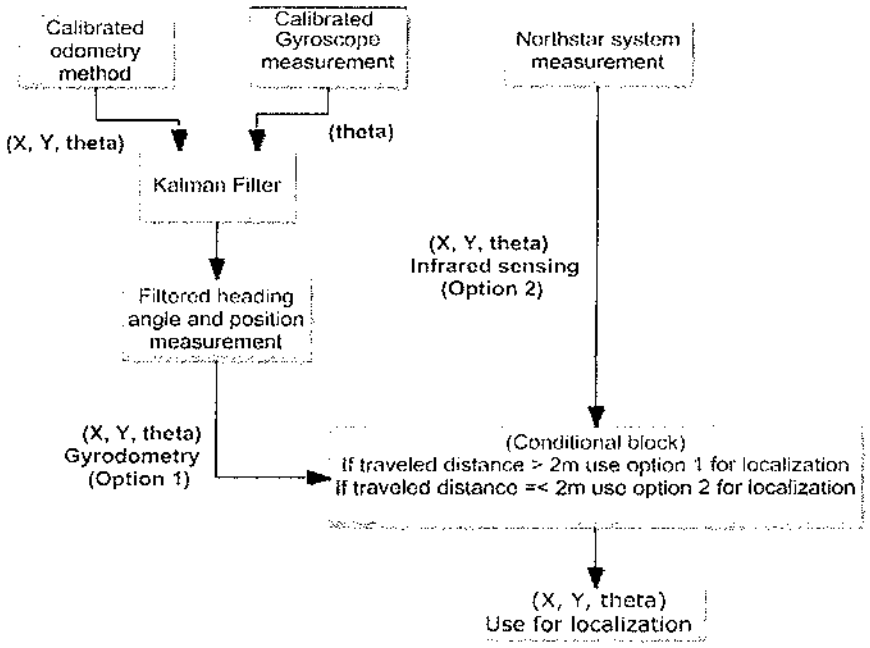

Figure 29: Overall structure of proposed method 1 used for localization

The implementation of the proposed method one used for localization is shown in Figure 29. This approach consisted of two different options for localization, the Gyrodometry method has been specified as option one and the Northstar system has been specified as option two for localization. This proposed algorithm basically uses one of the two localization options when a certain condition is met as specified in the conditional block. The travelled distance is used as the primary criteria to determine which one of the two localization methods is to be used. The travelled distance of the mobile robot is measured based on the distance of itself from the home reference position. Essentially, the conditional block consisted of two different conditional statements. Condition one states if the mobile robot is equal or within two meters of travelled distance from the home reference position (which is inside the detector's optimum field of view) then use only the Northstar position measurement and heading angle data for localization. Whereas, condition two states if the mobile robot has travelled past two meters from its home reference position (which is outside the detector's optimum field of view), then pose data returned by the Gyrodometry method are used for localization.

4.7.2 Propose method two

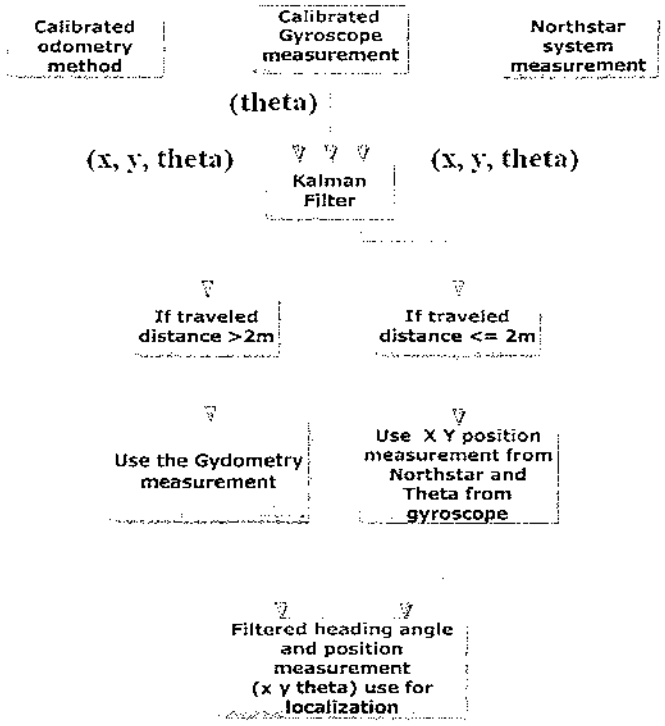

Figure 30: Overall structure of proposed method 2 used for localization

The implementation of the proposed method two used for localization is shown in Figure 30. Essentially, the Northstar system measurement is fused with calibrated wheel odometry measurement and the calibrated gyro measurement via a Kalman filter.

Throughout testing, we have proved that the accuracy of the position measurements from the Northstar system depends on the distance between the detector and the origin. When the detector moves along one axis (x or y), the readings of orientation (θ) and the other axis will eventually deteriorate. With the detector located 2.5 meters from the origin, position and heading angle readings became so unreliable. Collected results for both axis (x, y) returned by the detector would deviate up to 90cm centimeters from the computed mean value. It was noted that the detector also had problem determining the sign $(+/-)$ for the heading angle value being collected for a given position when it is located 1.5 meters or more from the origin. Thus, the approach presented in method two replaced the heading value (θ) returned by the Northstar system with the calibrated gyro measurement when the travelled distance of the mobile robot is within two meters from the home reference position.

4.8 Autonomous navigation implementation:

The main focus of my project is on localization of a mobile robot but it is also necessary to test the developed localization system by employing a navigation control algorithm to move the robot from a defined home position to a desired target position. In order to achieve this, the mobile robot must be able to estimate its own position with respect to its environment. The navigation algorithm could be programmed once a localization system has been successfully implemented.

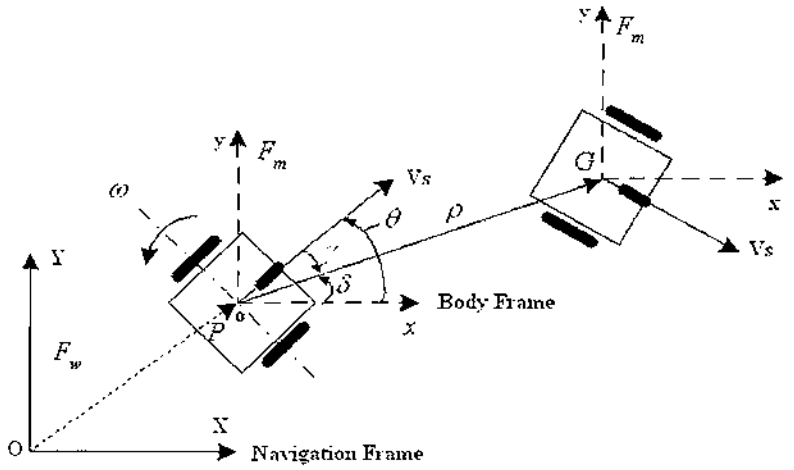

Figure 31: Shows the body and navigation frame of the mobile robot

Figure 31 shows the coordinates (OXY) is the navigation frame and (oxy) is the body frame of the mobile robot and the body frame's origin is located at the center of the mobile robot. The pose of the robot are in the form of (x, y, θ), where x and y are the Cartesian coordinates and (θ) is the current heading angle of the robot in a 2D plane.

The goal of navigation control is to move the robot from its original posture to its target posture. The system model of the navigation control task can be described in polar coordinates by the following equations:

$$\dot{\rho} = -v\cos\gamma, \quad \dot{\gamma} = -\omega + v\frac{\sin\gamma}{\rho}, \quad \dot{\delta} = v\frac{\sin\gamma}{\rho} \tag{25}$$

$$\rho = \sqrt{(x_g - x)^2 + (y_g - y)^2}$$

$$\delta = \tan^{-1}\left(\frac{y_g - y}{x_g - x}\right)$$

$$\gamma = \delta - \theta$$

Whereas x_g and y_g are the desired position of the mobile robot in the 2D plane.

Looking at the Equation 25, it can be seen that the current pose coordinates obtained from the localization system are used for calculating the following:

The distance to destination (ρ),

The angle to the desired destination (δ),

The error angle or the difference between the current angle and desired angle (γ)

Once the pose coordinate results have been collected and read, the distance to destination (ρ) can be calculated by the following code segment from the method 'ReadPoseValues( )':

```
DistanceToDestination = Sqrt(((DestinationXpose - (CurrentXpose + 65)) *
(DestinationXpose - (CurrentXpose + 65))) + (((DestinationYpose - 25) - (CurrentYpose + 65)) * ((DestinationYpose -
25) - (CurrentYpose + 65))));
```

The added and subtracted values are the values related to the geometry of the mobile robot's shape.

The angle to the desired destination (δ) is calculated by the following code segment in the program:

```
Destination_Angle =  (Atan2(((DestinationYpose - 25) - (CurrentYpose + 65)),
                                (DestinationXpose - (CurrentXpose + 65))) * (180 /PI));
```

The error angle (γ) is calculated by the following code segment in the program:

```
ErrorAngle = double (DestinationAngle - HeadingAngle);
```

The navigation controller equation, with the proved of Lyapunov and Barbalat, it can be represented by the following two equations:

$$v = k_1 \rho \cos\gamma \tag{26}$$

$$\omega = k_2 a_2 \gamma + \frac{k\gamma \sin 2}{2a\gamma}(a_2\gamma + a_3\delta)$$

The tuning parameters are k_1 and k_2 which need to be adjusted accordingly to achieve optimal performance. If the tuning factors are chose correctly, it will effectively make:
$\rho \to 0, \gamma \to 0, \delta \to 0$, hence the robot will reach its target position. Once the target has been reached the robot will stop, unless there is still another target position then the robot will proceed to this next target position.

The tuning factor of (1.8) is used as K_1 and K_2, but the method described above can be adjusted to suit different robot designs. The code used for trajectory planning basically first defines how long the mobile robot has to wait at each position if it is following a path, this delay duration is based upon the value set in the program by the user. Throughout testing, the delay duration at each position is set to 20 seconds to allow sufficient time for collecting ground truth data for comparison. The main algorithm firstly normalized the error angle parameter between -90 and 90 degrees for ease of calculation. And then the trajectory calculations are performed so the velocities are determined on each wheel. After this, the obtained velocities for each wheel are updated to the mobile robot's motors.

4.9 Experiments and Results on developed techniques for localization

This section presents the experiment results of the two different developed methods used for localization. Also two different experiments have been performed to observe the accuracy of all methods used for localization.

4.9.1 Experiment one

In experiment one, the reliability of the localization results obtained from the Northstar system, the Gyrodometry method and the two proposed methods were checked with a relatively ad-hoc approach. By using a metal 1-meter ruler along with a plastic protractor, a grid was drawn on the floor-space within the localization space. In addition, as shown in Figure 32, a 0.5 meter square grid is labeled over the floor area for measuring ground truth data for comparison at the position the mobile robot is located.

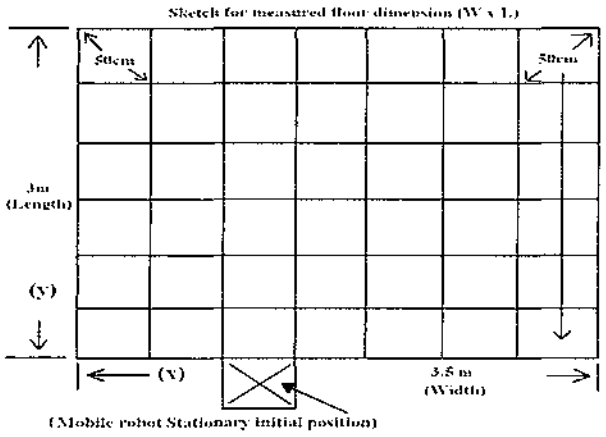

Figure 32: The grid of 0.5 meter square that was placed one the floor

As shown in Figure 32, an x-axis and a y-axis are defined on the floor of the laboratory with the home reference position directly within the optimum detection field of the Northstar detector sensor which is placed on top of the mobile robot. The origin (reference position) was taken to be at the approximate (0,0) position on the floor. A three meter y-axis was drawn on the floor along the -90 degree direction. Likewise, a three meter x-axis was drawn perpendicular to this y-axis. Once these axis lines had been defined, a three-meter line dissecting both the x and y axis was drawn at a 45 degree angle. This setup is presented in Figure 33.

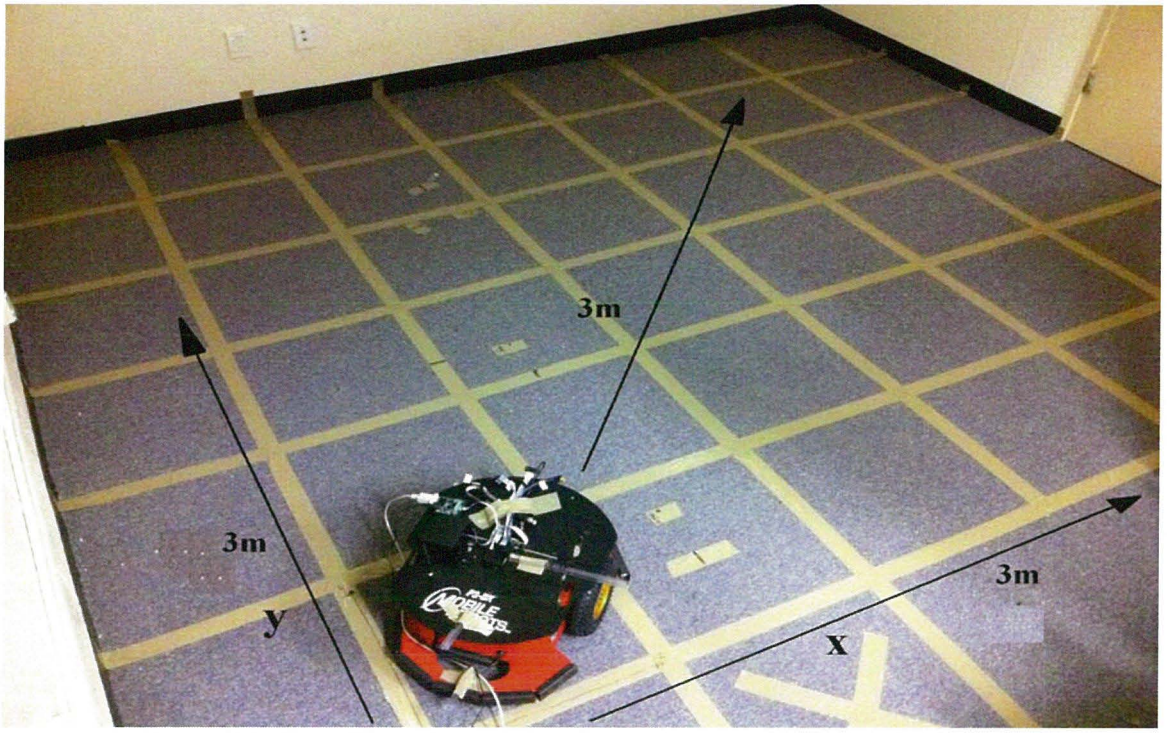

Figure 33: Testing method for taking pose measurements

During this experiment, 20 data points were collected for each method at each position. To do this with the Northstar system, the N-Point average window provided by Northstar as shown in Figure 34 is used to acquire the required number of points and use statistical measures to compute the mean, standard deviation and variance of these pose results. And as for the other localization methods, the collected pose data are exported to an excel spread sheet so the mean value can be computed.

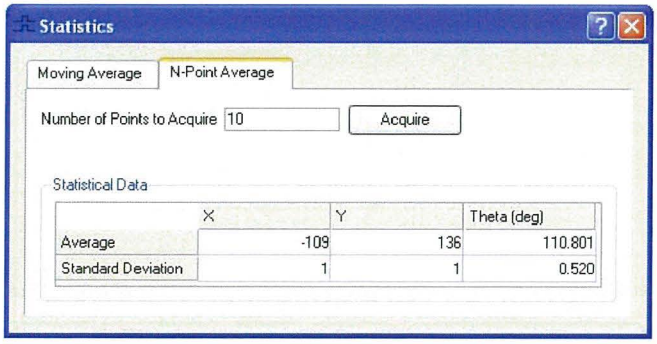

Figure 34: The N-point average window provided by Northstar

The next step was to align the robot with the y-axis and take pose measurements in intervals of 10cm moving along the positive y-axis direction. Pose measurements were taken in the same procedure in intervals of 10cm along the defined x-axis direction and 45 degree dissecting line.

The results measured in our laboratory are displayed in Figures 35 - 40. Throughout testing, for each position multiple values are recorded and the absolute difference between the mean of the measured position and the true value is also provided and displayed in Figures 35 – 40.

Figure 35 is a plot of different methods used for comparison and Figure 36 is a plot of the two developed methods y-displacement versus true y-displacement. These results were obtained by using the measurement defined on the floor-space to be the true measurements and taking the y value from the pose result of the three different methods to be the estimated measurements. A similar approach was taken to obtain the results for Figure 37 and 38 for the x axis. The results for Figure 39 and 40 were acquired by taken both an x and y measurements and using Pythagoras theorem to determine the measured hypotenuse distance to the measurement point. These calculated values were matched against the supposed true value as marked on the floor-space.

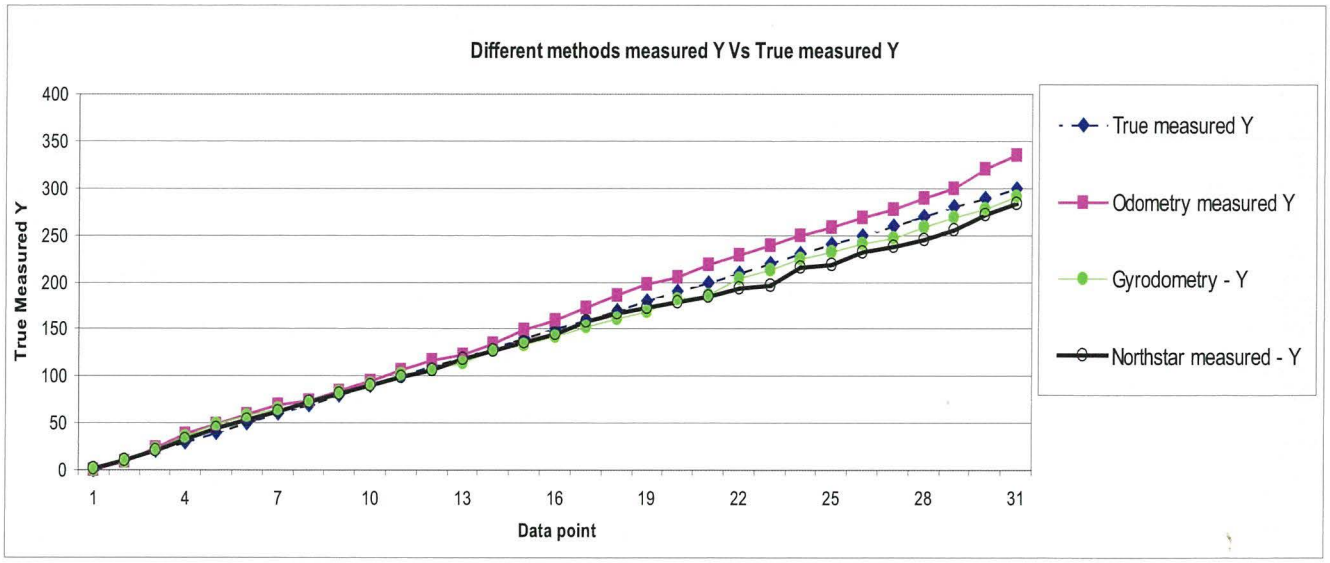

Figure 35: Different methods measured y-displacement vs true y-displacement

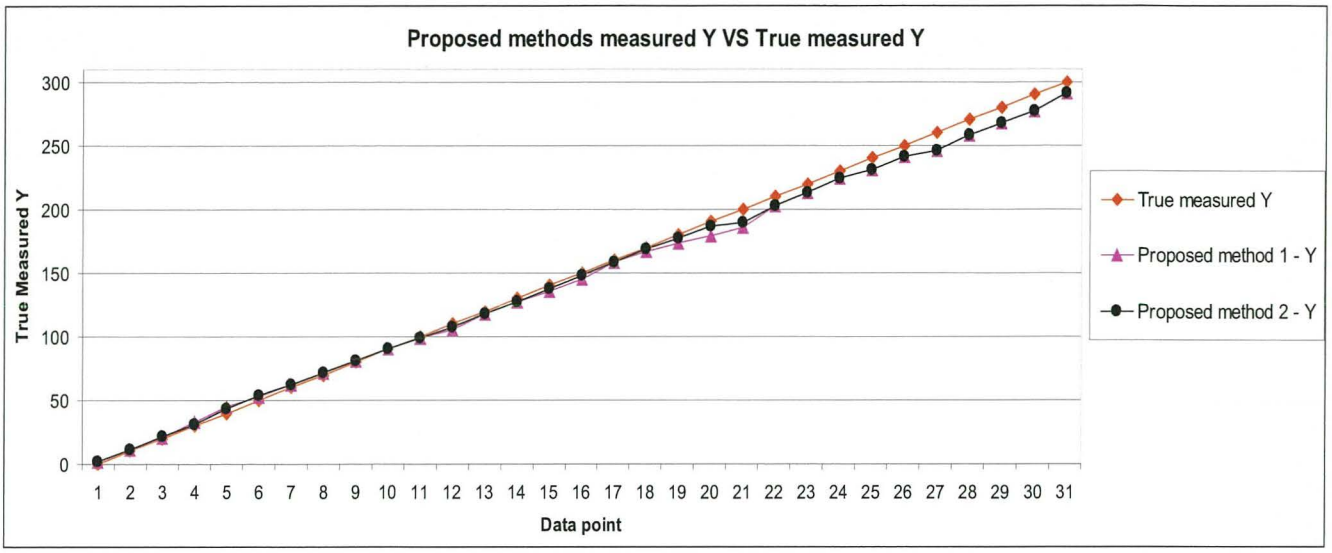

Figure 36: Proposed methods measured y-displacement vs true y-displacement

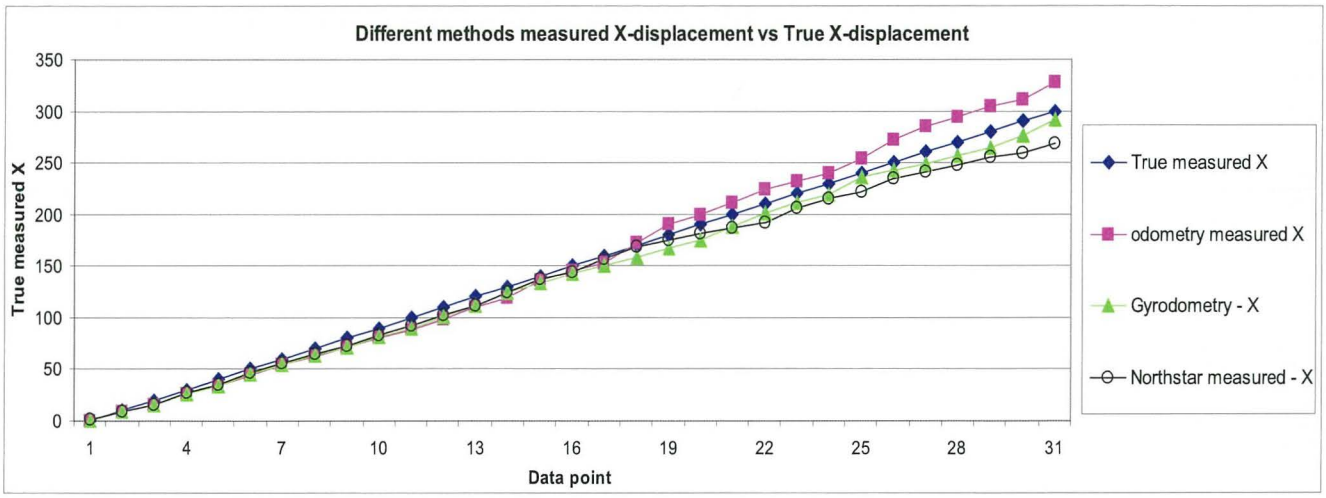

Figure 37: Different methods measured x-displacement vs true x-displacement

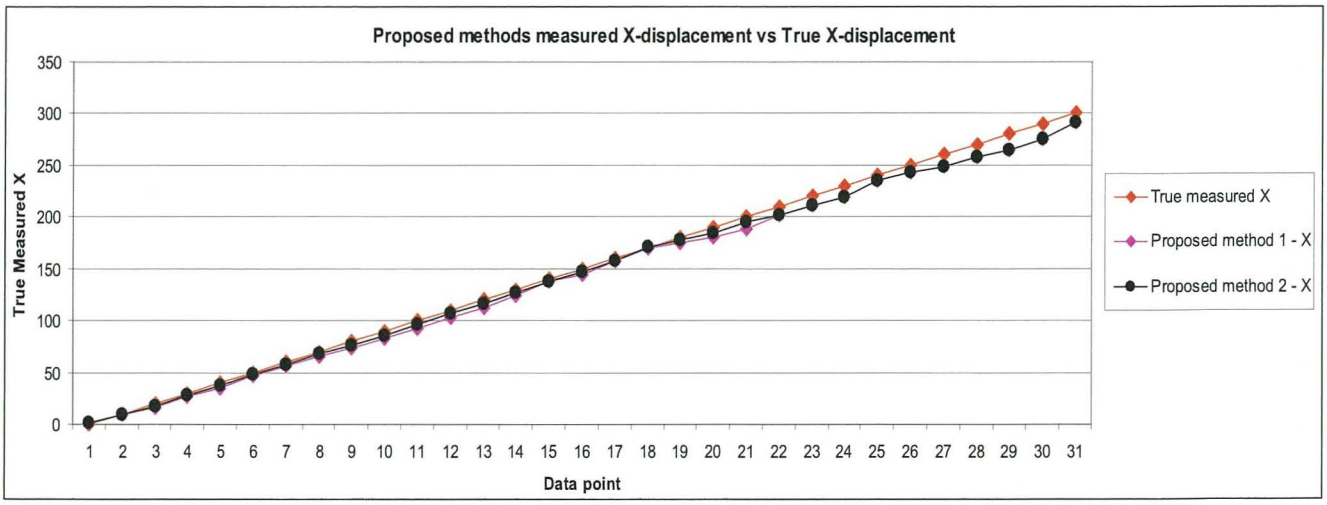

Figure 38: Proposed methods measured x-displacement vs true x-displacement

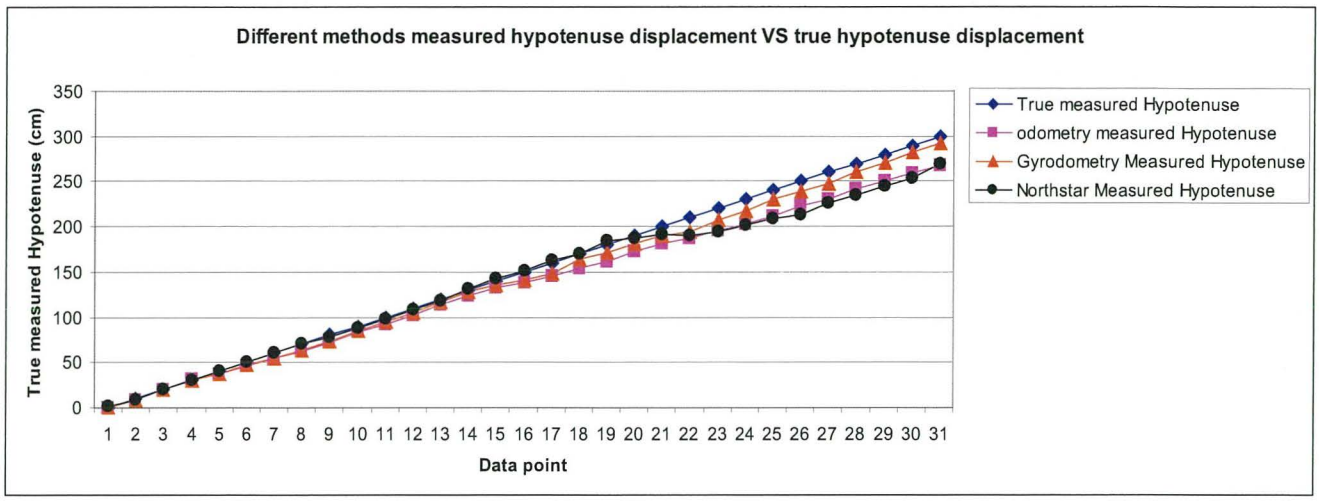

Figure 39: Different methods measured hypotenuse displacement vs true hypotenuse displacement

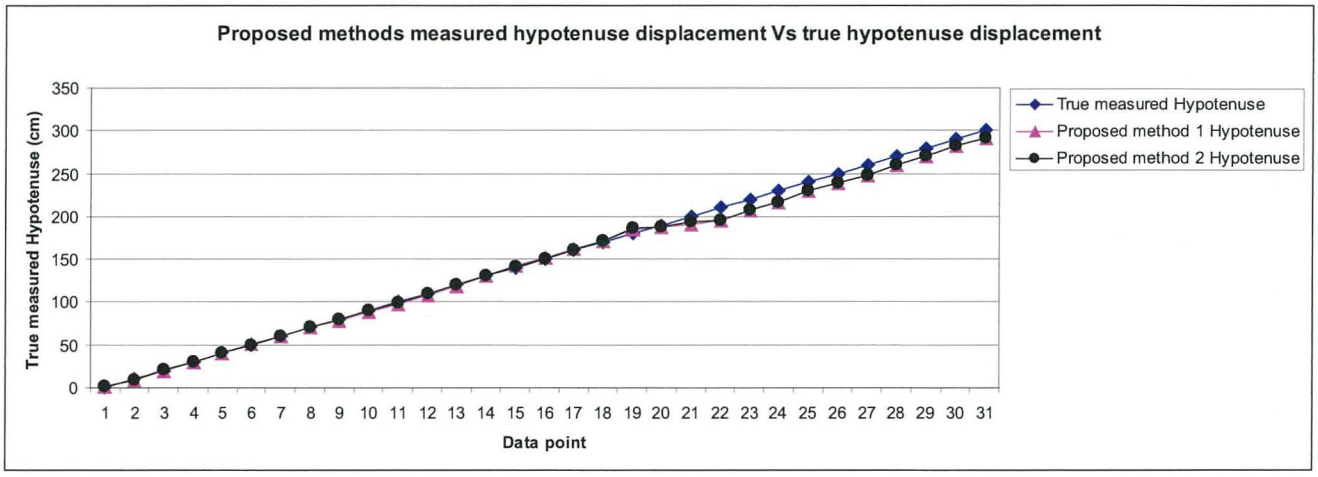

Figure 40: Proposed methods measured hypotenuse displacement vs true hypotenuse displacement

Figures 36, 38 and 40 demonstrated the effectiveness of the two developed methods for localization. The experiment results gathered show that both approaches in our proposed localization method achieved relatively linear results which match the true measured results to acceptable levels. Results also showed that the Northstar system pose measurement errors started to increase after the mobile robot has travelled past 1.8 m distance from the localization origin.

The results collected from the Gyrodometry method are not as linear as the Northstar system for the measurements taken in the first two meters of distance from the reference home position. Throughout testing, we have concluded that the accuracy of position estimates from the Northstar system depends on the distance between the detector and the origin. As the mobile robot carrying the detector moves along one axis direction, the values for the orientation will eventually degrade.

4.9.2 Experiment two

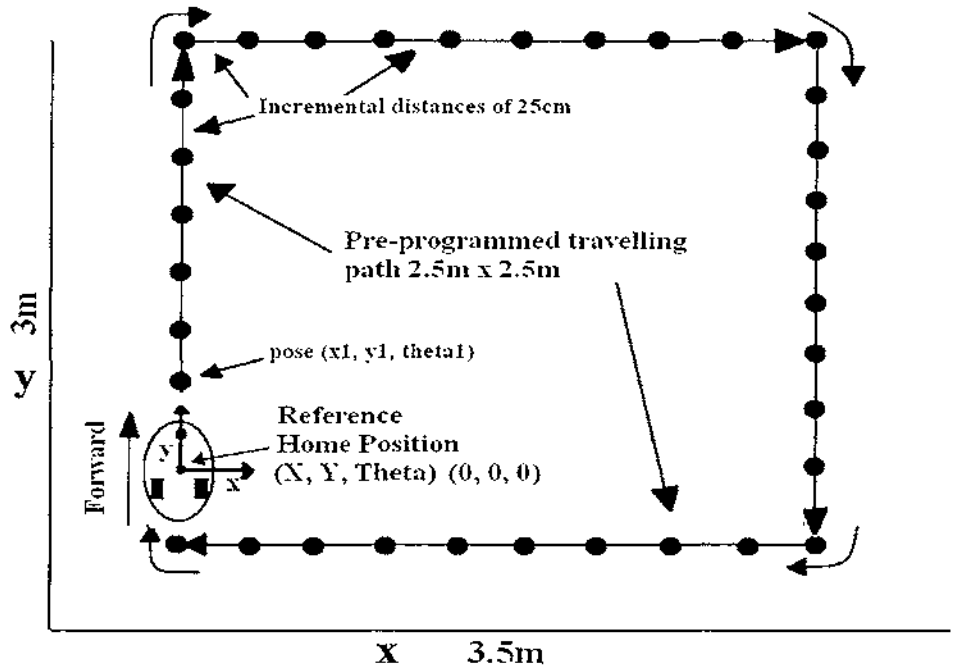

Figure 41: Testing procedure for experiment two

The main purpose of experiment two is to test the implemented autonomous navigation algorithm with the mobile robot. And the preprogrammed travelling path as shown in Figure 41 is performed to observe the accuracy of the first and second approach in our proposed method localization.

The testing procedure in experiment two involved in starting the mobile robot at a reference home position of our lab room and navigates it to the predetermined points along the lab floor. The mobile robot travels at an incremental distances of 25cm and when it has arrived at each point, the robot would stopped for 20 seconds, this delay duration is used to provide sufficient time for recording pose data from the odometry method, Gyrodometry method, Northstar system measurements, the proposed localization method one and two. At each position the actual pose data measured from the ground was also collected for comparison and to evaluate the effectiveness of the two proposed methods used for localization. Results obtained from each method are presented in Table 3. Throughout testing, the mobile robot can either autonomously navigate to a single point or navigate through a path of points. It also was noted that further tuning on the autonomous navigation algorithm is required to improve speed to destination.

4.9.3 Experiment Results

In this section, the error comparison of the five different techniques used in experiment one and two are presented in Table 3.

$$\text{Error } \% = \frac{(\text{Actual measured position}) - (\text{Estimated position})}{\text{Total distance travelled}} \quad (27)$$

The obtained error percentage on each method is computed based on Equation 27. The actual position (ground truth data) is measured using a metal one meter ruler and a protractor. In every experiment the start position (home reference position) and heading angle of the mobile robot is set at (0, 0, 0). All measurements are in centimeters. The maximum length of travel was limited to the maximum length of the floor in our laboratory. In this case, we have set up the floor dimension to be a 3m x 3.5m area. The travel speed of the mobile robot is set to $^{0.1}/_{s}$.

Experiment number	Odometry method (Min to Max)	Gyrodometry Method (Min to Max)	Northstar localization system (Min to Max)	Proposed method 1 (Min to Max)	Proposed method 2 (Min to Max)
1	5.5 - 8%	4.5 - 5%	3 – 4% within 2m 10% at 3m	3.5 – 4.5%	3% - 4%
2	5.5 – 15%	5.5 - 6.5%	4.5 - 10%	4.5 - 6.5%	4% - 4.5%

Table 3: Summary table for error comparison of five different methods used for localization

Table 3 summaries the results of the two experiments conducted using five different localization techniques. Results show that the mobile robot can not reply on the un-calibrated odometry method even for small distances. It can produce error up to 8 % in the positions estimated in experiment one and 15% in experiment two.

As discussed earlier, the optimum field of view of the Northstar detector is merely two meters. Localization using the Northstar within this region provides maximum position error of 4.5% in the position estimated. But pose measurements which are measured outside this region are fairly inaccurate. The Northstar system can produce error up to 10% in the positions estimated at three meters from the origin in experiment one and two.

Experiments with the Pioneer p3dx robot show that localization using our proposed method two achieved the lowest error at 3% (Minimum) to 4% (Maximum) in the position estimates in experiment one and 4% (Minimum) to 4.5% (Maximum) in the position estimates in experiment two. The obtained results also proved that the second approach in our proposed method for localization performs the best and has achieved minimal error percentage in the position estimates out of the five methods tested in the two experiments.

5. Conclusions and future work

In conclusion, the localization system was implemented successfully and allows further development. The Northstar system measurements and Gyrodometry measurements were successfully integrated into the robot control computer program developed to display the robot's position and orientation. This computer program interacts with the simulator or the P3dx mobile robot test-platform so that manual and autonomous control was achieved. The pose results can be recorded and sent to a txt file for storage and further analysis.

The Kalman filter also proved to provide an accurate position and heading angle estimator in places that Northstar was reliable and Gyrodometry method was not, and also in places where Northstar was unreliable and Gyrodometry method was.

Two different approaches have been developed for achieving reliable indoor localization. Both methods combined with the relative position measurement obtained from the Gyrodometry method and the absolute position measurement obtained from the Northstar system to provide better and more accurate pose measurement. Extensive experimental results also proved both proposed methods to be very effective in reducing odometry errors due to both systematic and non-systematic errors such as those caused by wheel base error and uneven floor or by bumps. Overall, the developed localization system also provides a robust and efficient method to navigate the mobile robot autonomously in an indoor environment.

As a recommendation for future work, trajectory planning around objects that are close to the mobile robot can be performed. If the mobile robot can detect objects near it or it knows where the objects are that it has to avoid than paths can be planned in the navigation algorithm to move the mobile robot around the obstacles. Also we could look into expanding the field of view of the Northstar detector by using an array of projectors to enlarge the area for indoor localization of mobile robots. But installation of these sensors would be a costly process. By implementing the proposed localization system which is fused with the Gyrodometry method allows for less frequent absolute position updates from the Northstar system. Thus, fewer optical beacons (Northstar projectors) are required for a given travel distance.

6. References

[1] P. Goel, Stergios I. Roumeliotis and Gaurav S. Sukhatme, *"Robust localization using relative and absolute position estimates"*. In **Proceedings of the *1999* IEEE International Conference on Intelligent Robots and Systems**, pages 1134-1140, March, 1999.

[2] J. Borenstein and L. Feng. "*Measurement ad Correction of Systematic Odometry Errors in Mobile Robots*". In **Proceedings of the *1996* IEEE International Conference on Robotics and Automation,** pages 869-881, 1996.

[3] J. Borenstein and L. Feng. "*Gyrodometry: A new method for combining data from gyros and odometry in mobile robots*". In **Proceedings of the *1996* IEEE International conference on robotics and automation. Minneapolis, Minnesota.** April 1996.

[4] Á. Gutiérrez, A. Campo, F. C. Santos, F. Monasterio-Huelin and M.Dorigo. "*Social Odometry: Limitation Based Odometry in Collective Robotics*". In ***2009*** **International Journal of Advanced Robotic System, Vol. 6, no. 2,** pages 129-136, 2009.

[5] K. Park, D. Chung, H. Chung and J. G. Lee. "*Dead reckoning navigation of a mobile robot using an indirect Kalman filter*". In **Proceedings of the *1996* IEEE/SICE/RSJ International conference on multisensor fusion and integration for intelligent systems.** 1996.

[6] E. Papadopoulos and M. Misailidis. "*On Differential Drive Robot Odometry with Application to Path Planning*". In **Proceedings of the *2007* European Control Conference,** pages 5492-5499, 2007.

[7] N. Doh, H. Choset and W. Kyun Chung, *"Accurate Relative Localization Using Odometry"*. In **Proceedings of the *2003* International Conference on Robotics and Automation**, Taipei, Taiwan, September 14-19, 2003.

[8] Y. Yutaka, P. Paolo, B. Joe, M. Mario, D. Enrico, G. Luis, O. Jim and K. Niklas, *"Optical sensing for robot perception and localization"*, Evolution robotics, Inc.

[9] Q. Alexander, H. Hockett II, A. Veluchamy, M. Anderson, *"Acquisition of 2-D Ground truth data in multi-robot experiments"*, University of Alabama, Tuscaloosa, AL, 35487 United States.

[10] R.J.E. Merry. *"Speed and acceleration estimation for optical incremental encoders"*. **In Journal of Mechatronics** 20, 2010, 20-26.

[11] M.F. Benkhoris and M. Ait-Ahmed. *"Discrete speed estimation from a position encoder for motor drives"*. **In Proceedings of the 1996 IEEE Power Electronics and Variable Speed Drives.** Conference Publication No. 429, September 23-25, 1996.

[12] Brown R.H., Schneider S.C. and Mulligan M.G., 1992, *"Analysis of algorithms for velocity estimation from discrete position versus time data"*, **IEEE Trans. On Industrial Electronics.** 39, 11 to 19

[13] Merry R, van de Molengraft MJG, Steinbuch M. *"Error modeling and improved position estimation for optical incremental encoders by means of time stamping"*. America Control Conference 2007:3570–5.

[14] G. Liu. *"On speed estimation using position measurements"*. **In Proceedings of the 2002 American Control Conference, Anchorage,** USA, May 8-10, 2002.

[15] Y. Buchnik and R. Rabinovici. *"Speed and position estimation of brushless DC motor in very low speeds"*. **In Proceedings of the 2004 IEEE convention of Electrical and electronics engineers in Israel,** Sep 6-7, 2004.

[16] H. Kim and S. Sul. *"A new motor speed estimator using Kalman Filter in low-speed range"*. **In Proceedings of the 1996 IEEE transactions on industrial electronics,** vol. 43, NO. 4, August 1996.

[17] J. Yoo, T. Park, S. Kim, N. Kim and J. Yoo. *"Speed estimation of an IM using Kalman filter algorithm at ultra-low speed region"*. **In Proceedings of the 1997 IEEE international electric machines and drives conference,** Korea, May 18-21, 1997.

[18] R. E, Kalman. *"A New Approach to Linear Filtering and Prediction Problems"*. **In Transaction of the ASME 1960 Journal of Basic Engineering,** pages 35-45, March, 1960.

[19] C. Hul, W. Chen, Y. Chen and D. Liu. *"Adaptive Kalman Filtering for Vehicle Navigation"*. **In Journal of Global Positioning Systems 2003** Vol. 2, No. 1, pages 42-47, 2003.

[20] Saito K., Kamiyama K., Ohmae T. and Matsuda M., 1988, "A microprocessor-controlled speed regulator with instantaneous speed estimation for motor drives", IEEE Trans. On Industrial Electronics. 35, 95 to 99

[21] Kenji Kubo, Masahiko Watanabe, Fusaaki Kozawa, Kyouichi Kawasaki, "*Disturbance Torque Compensated Speed Observer for Digital Servo Drives*", Conf. Rec., IEEE/IAS Ann. Mtg., ppll82-1187,1990.

[22] Evolution Robotics. (2008). *Evoluation Robotic Northstar Localization System.* Retrieved from website: http://www.evolution.com/products/northstar/

[23] Evolution Robotics. (2008). *The Northstar localization system configuration mode.* Retrieved from website: http://www.evolution.com/products/northstar/works.masn

[24] S. Patnaik. *"Robot cognition and Navigation"*. An experiment with mobile robots, Springer Berlin Heidelberg, New York, 2007.

[25] J. Laaksonen. "*Mobile robot localization using sonar ranging and WLAN intensity maps*". Lappeenranta University of Technology, Department of Information Technology, 2007.

[26] J. Kim, J.G. Lee, G.I. Jee, and T.K. Sung. "*Compensation of Gyroscope Errors and GPS/DR integration.*" Volume 1. Position Location and Navigation Symposium, IEEE, April 1996. Atlanta, USA.

[27] Betke, M. and Gurvits, L., 1994, "*Mobile Robot Localization Using Landmarks.* " 1994 Intemutional Conference on Intelligent Robots and Systems(IROS '94). Munich, Germany, Sept. 12-16, pp. 135-142.

[28] B. Barshan, and H. F. Durrant-Whyte, 1994, *"Orientation Estimate for Mobile Robots Using Gyroscopic Information."* 1994 International Conference on Intelligent Robots and Systems. Munich, Germany, Sept. 12-16, pp. 2243-2248.

[29] J. Borentein, and L. Feng, "*Correction of Systematic Dead-reckoning Errors in Mobile Robots.*" 1995 International Conference on Intelligent Robots and Systems. Pittsburgh, PA, Aug. 5-9, pp.569-574.

[30] L. Kleeman, 1992, "*Optimal Estimation of Position and Heading for Mobile Robot Using Ultrasonic Beacons and Dead-reckoning.*" In Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, May 12-14, pp. 2582-2587.

[31] J. Vaganay, M. J. Aldon, and A. Fourinier, 1993, "*Mobile Robot Attitude Estimation by Fusion of Inertial Data.*" In Proceedings of IEEE International Conference on Robotics and Automation, Atlanta, GA, May 2-7, pp. 3243-3248.

[32] C.M. Wang. *"Location estimation and uncertainty analysis for mobile robots."* In Proceedings of IEEE International Conference on Robotics and Automation, IEEE, April 1988. Philadelphia, USA. Pages 1231–1235.

[33] S.I. Roumeliotis, G.S. Sukhatme, and G.A. Bekey. *"Fault detection and identification in a mobile robot using multiplemodel estimation."* Volume 3, pages 2223–2228. In Proceedings of IEEE International Conference on Robotics and Automation, IEEE, May 1998. Leuven, Belgium.

[34] H. Durrant-Whyte and J. Leonard. *"Mobile robot localization by tracking geometric beacons."* In Proceedings of IEEE Transaction on Robotics and Automation, 7:376–382, 1991.

[35] B. Barshan, and H.F. Durrant-Whyte. *"Inertial navigation systems for mobile robots."* In Proceedings of IEEE Transactions on Robotics and Automation, 11:328–342, 1995.

[36] M.A. Batalin, G.S. Sukhatme and M. Hattig. *"Mobile Robot Navigation using a Sensor Network."*

In Proceedings of the 2004 IEEE International conference on Robotics and Automation, New Orleans, LA. April 2004.

[37] S. Se, D. Lowe, and J. Little. *"Local and Global Localization for mobile robots using visual landmarks."* In Proceedings of the 2001 IEEE/RSJ International conference on intelligent robots and systems, Vancouver, BC. 2001.

[38] S. Se, D. Lowe, and J. Little. *"Vision-based mobile robot localization and mapping using scale-invariant features."* In Proceedings of the IEEE International Conference on Robotics and Automation (ZCRA), pages 2051-2058, Seoul, Korea, May 2001.

[39] W. Burgard, D. Fox, and S. Thrun. *"Active mobile robot localization."* In Proceedings of the International Joint Conference on Artificial Intelligence, Nagoya, Japan, August 1997.

[40] L. Feng, Y. Fainman, and Y. Koren. *"Estimate of Absolute Position of Mobile Systems by Opto-electronic Processor."* In Proceedings of the IEEE Transactions on Man, Machine and Cybernetics, Vol. 22, No. 5, pp. 954-963.

[41] J. Borenstein, *"Experimental Results from Internal Odometry Error Correction With the OmniMate Mobile Platform."* In Proceedings of the IEEE Transactions on Robotics and Automation, July 1996.

[42] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe, *"Mobile robot positioning sensors and techniques."* Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249.

[43] Jesse, N. Real World Applications of Computational Intelligence. *"Autonomous Mobile Robots – From Science Fiction to Reality."* Vol.179, pp.197-219, 2005.

[44] Graham, B., & McGowan, K., (2004). *"Build Your Own All-Terrain Robot."* McGraw-Hill.

[45] Liu, J.X., (2005). *"Mobile Robots New Research."* Nova Science Publishers, Inc.: New York.

[46] Cuesta, F.,& Ollero, A., (2005). *"Intelligent mobile robot navigation."* Springer, London.

[47] Bräunl.T., (2). *"Embedded Robotics: mobile robot design and applications with embedded systems."* Nova Science Publishers, Inc.: New York.

[48] Thuilot, B., D'andrea Novel, B., & Micaellli, A. (1996). *"Modeling and feedback control of mobile robots equipped with several steering wheels."* In the proceeding of the IEEE Transactions on Robotics and Automation, 12(3), 375–390.

[49] Elliott, D. Kaplan, (1996). *"Understanding GPS: principles and applications."* Boston: Artech House.

[50] K. Moore, and J. Zhou, (2009). "Indoor navigation control of a mobile robot part 1: Localization and Communications." Massey University, Wellington.

[51] J. Zhou, and K. Moore, (2009). "Indoor navigation control of a mobile robot part 2: Robot and Control." Massey University, Wellington.

[52] J. Dixon, and O. Henlich, (1997). *Mobile robot navigation.* Retrieved from website: http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/

[53] P. S. Maybeck, *"Stochastic models, estimation, and Control Volume 1."* Academic Press. 1979

[54] A. Lawrence, *"Modern Inertial Technology - Navigation, Guidance, and Control."* Springer-Verlang. 1993