Design and development of Intelligent Mobile Robots (IMRs) for disaster mitigation and firefighting

Seventh Deliverable

Report submitted on 18th January 2018

PI: Dr. Muhammad Bilal Kadri

Co-PI: Dr. Tariq Mairaj Rasool Khan

Table of Contents

Chapter 1. Introduction

This report is the seventh deliverable for the National ICTR&D funded research project titled *"Design and development of Intelligent Mobile Robots (IMRs) for disaster mitigation and firefighting"*.

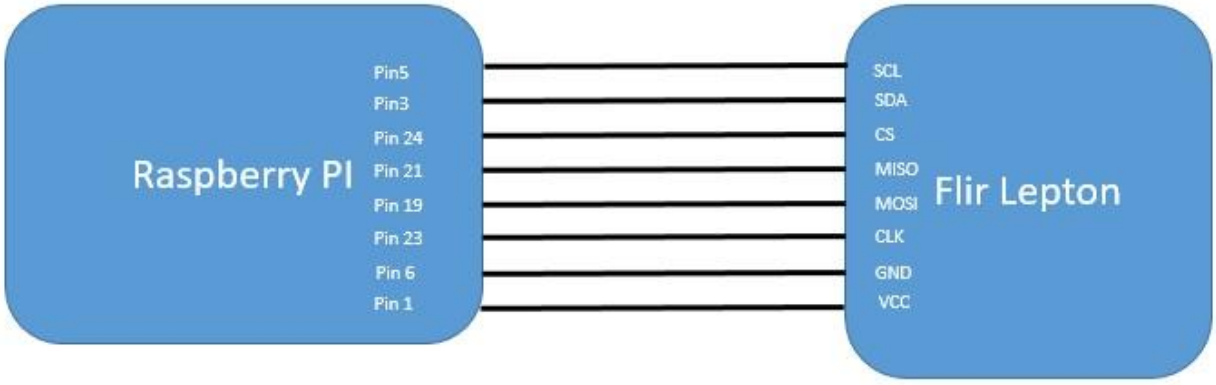
Chapter 2. Fusion of sensor information from Thermal Imaging Camera and UT sensors for robot navigation

2.1 Integration of Thermal Imaging Camera FLIR Lepton 2.5 with radiometry mode.

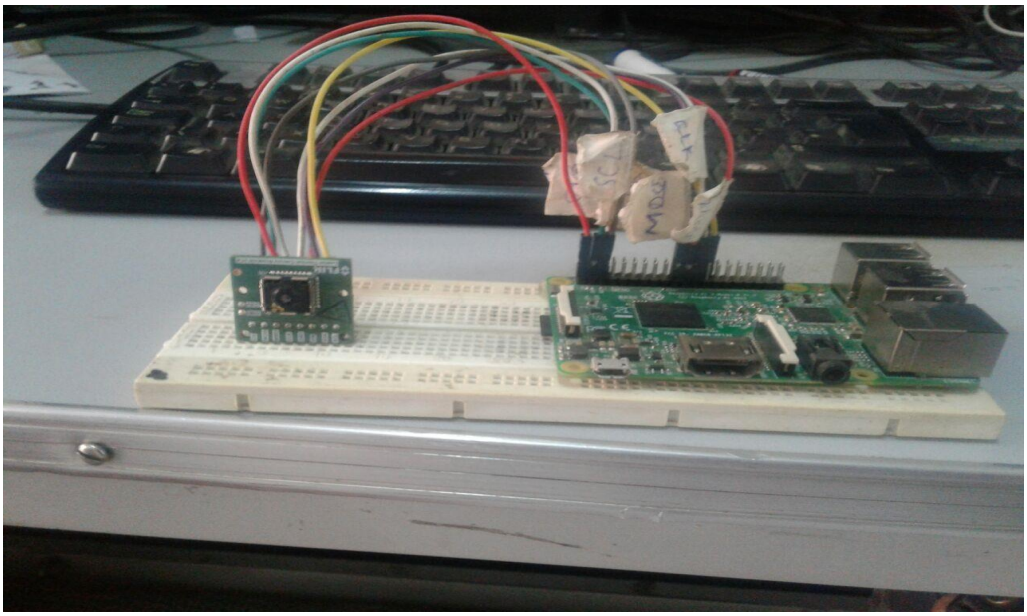
Radiometric enable FLIR 2.5 is built upon its predecessor FLIR 2.0. It comes with Long Wave Infrared (LWIR). It can generate temperature values of 4800-pixel values with +/- 5 ˚C.

Just like Lepton 2.0, the 2.5 core has a resolution of 80x60 pixels in an 8.5 x 11.7 x 5.6 mm package - it will fit into the same socket as Lepton 2.0 and existing Lepton 2.0 hardware will support 2.5 but firmware will likely need updating to utilize radiometry. As with Lepton 2.0, the 2.5's frame rate is 9 Hz with a spectral response wavelength range of 8 - 14 microns (nominal)

2.1.1 Interfacing of FLIR Lepton with Raspberry PI 3



Pin configuration of FLIR Lepton with Raspberry PI 3



Experimental Setup at NDT Lab

2.2 Experimental Setup at NDT Lab

The tasks are done in two steps:

a) Python Script (attached at Appendix 2.1) was developed and a function implemented to convert raw values into temperature values
b) The raw 2D array was published and subscriber node was able to extract the image with temperature values from the node. Other processing like extracting of temperature densities are also done on Subscriber node. (ROS Package attached at Appendix 2.2)

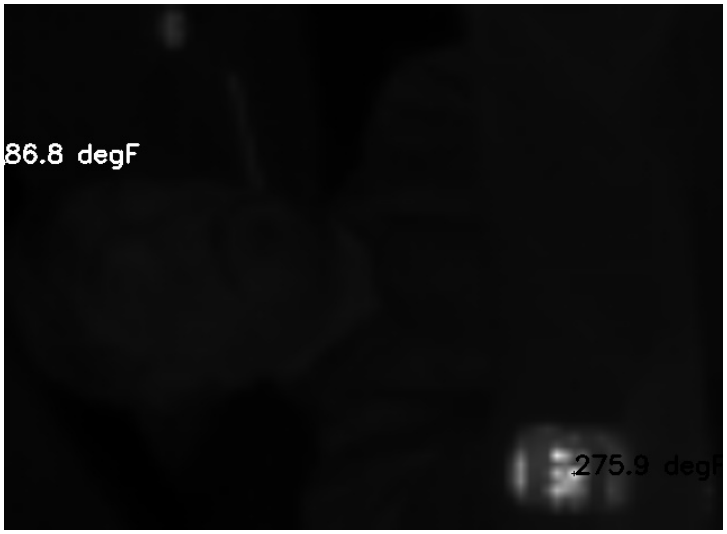
2.2.1 Task 1.

Firstly it captures 60 x 80 pixel values (raw values) a 2D array. A function is made to convert raw values into temperature values. Maximum and minimum values are noted while the rest are ignored only for this task. After that, it converts the values into normalize values. After that, it converts the 2D normalize array into the image and the noted temperature values are shown in the image. Hotter to colder as shown in White to black respectively.

Images captured

Above image shows the thermal image of a heated lighter. Max value is 327 deg Fahrenheit and Min Value is at the top with 95 deg Fahrenheit

Above image shows the thermal image of a heated lighter after a few seconds. Max value is 271 deg Fahrenheit and Min Value is at the top with deg 90 Fahrenheit.



Above shows the thermal image of a heated lighter after a few seconds. Max value is 271 deg Fahrenheit and Min Value is at the top with 86.8 deg Fahrenheit.

2.2.2 Task 2

The same process is done but now Task is divided into two nodes.

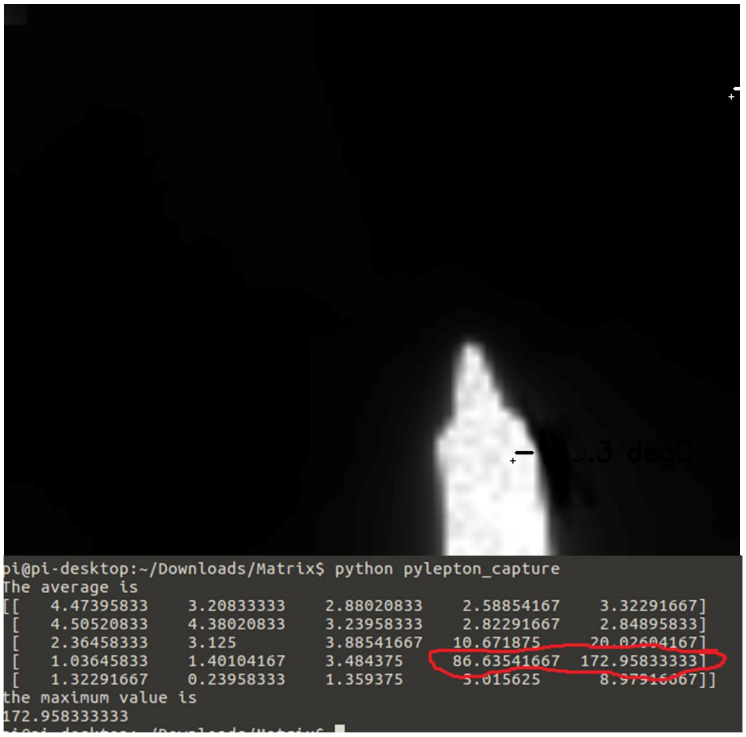
On Publisher node a python script captures the 60 x 80 pixel values (raw values) a 2D array. A function is made to convert raw values into temperature values. Maximum and minimum values along with raw values are published.

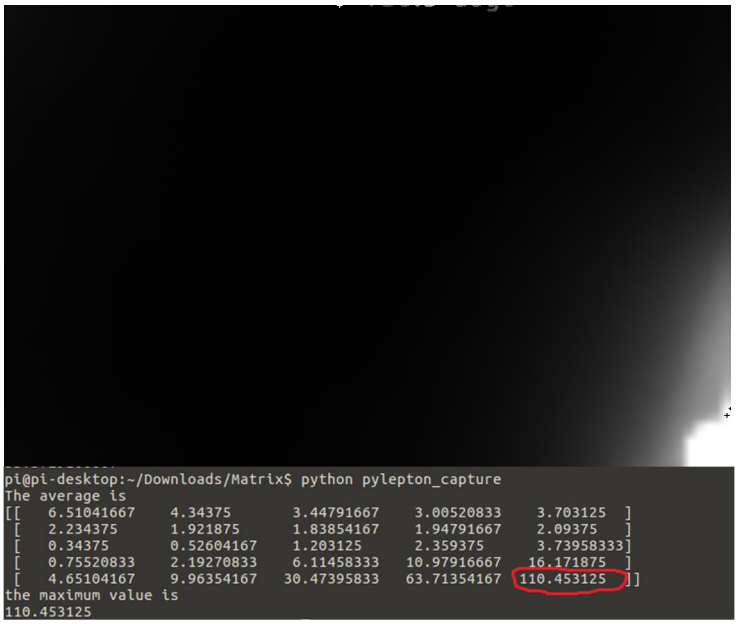
On Subscriber node, it converts the values into normalize values. After that, it converts the 2D normalize array into the image and the noted temperature values are shown onto the image. Hotter to colder as shown in White to black respectively.

In task 3, temperature densities are calculated by taking the average of 12*16 pixels. By this, a 5x5 Matrix is generated that shows the heat density.

Images captured

Images captured of heated soldering iron. Values inside red highlighted area shows the maximum heat distribution in 5x5 matrix





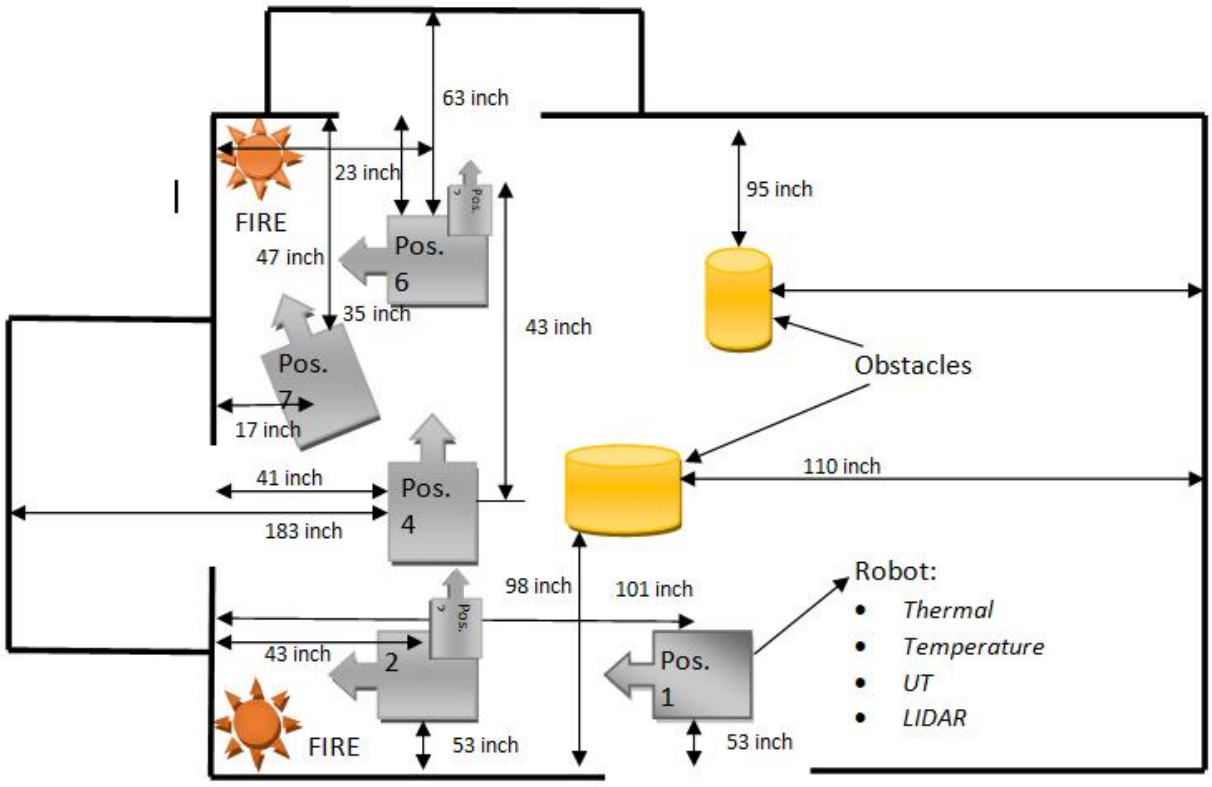
```
pi@pi-desktop:~/Downloads/Matrix$ python pylepton_capture
The average is
[[   6.51041667    4.34375       3.44791667    3.00520833    3.703125  ]
 [   2.234375      1.921875      1.83854167    1.94791667    2.09375   ]
 [   0.34375       0.52604167    1.203125      2.359375      3.73958333]
 [   0.75520833    2.19270833    6.11458333   10.97916667   16.171875  ]
 [   4.65104167    9.96354167   30.47395833   63.71354167  110.453125  ]]
the maximum value is
110.453125
```

2.3 Data Collection at Test Site PAF KIET

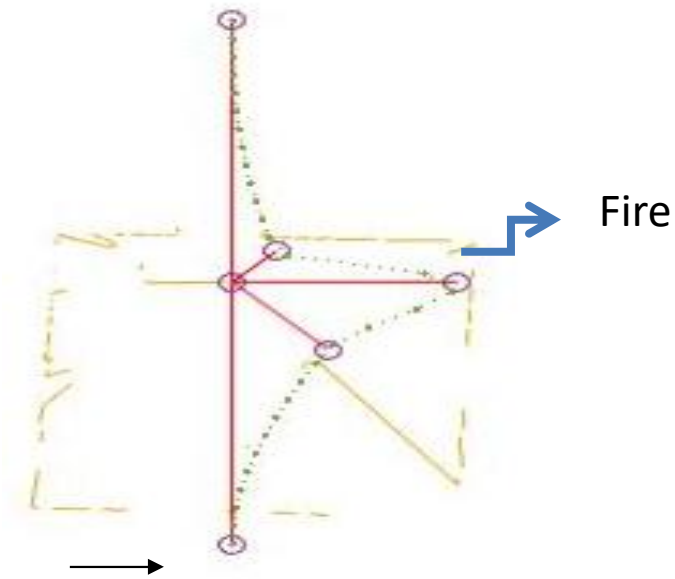
On 6th December 2017, team NUST went to PAF KIET to test the behavior of sensors at fire condition. Fire was set at two different corners of the test site. Obstacles were also placed to mimic the real conditions. The site is shown below

Site Picture

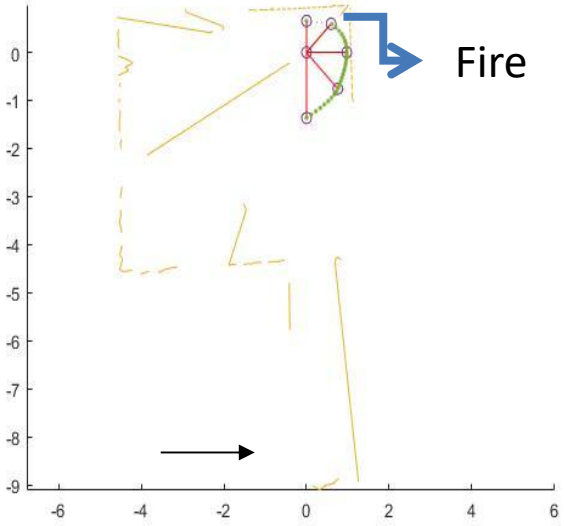
The data was acquired using Thermal, UT, Temperature and LIDAR sensors. The robot was placed at 5 different locations. Positions of Robot and direction at the test site is shown according to the following 2D Map. The measurement for all the positions has also taken to get the error from sensors data.



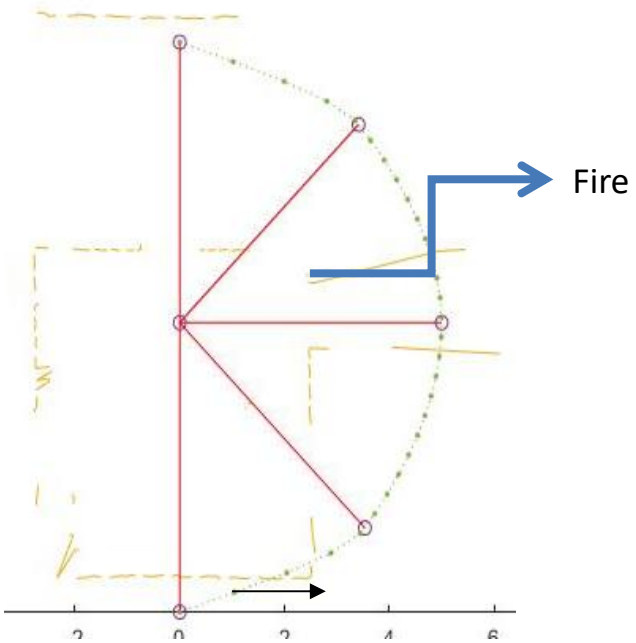
Plotting of Lidar and UT Sensor Data at different positions are shown below. The robot was mounted with 5 UT sensor. UT data was interpolated with cubic spline to get continues values.



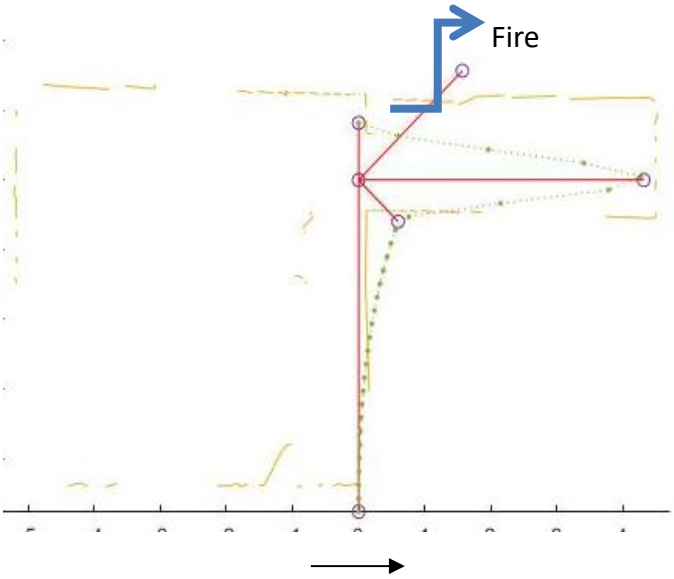
Position 1



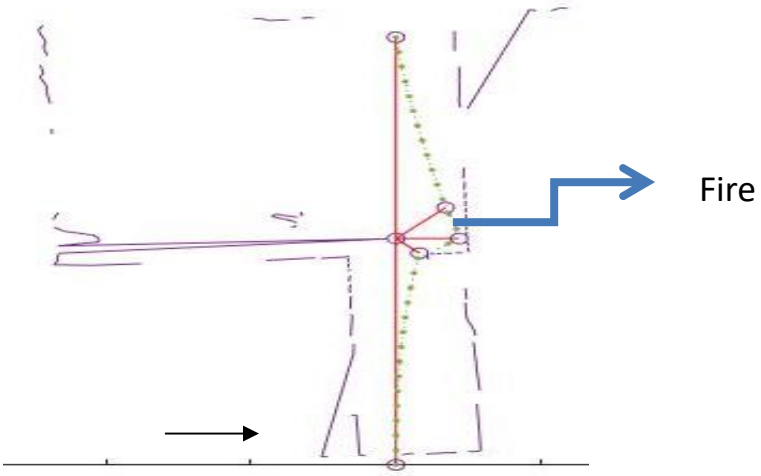
Position 2



Position 3



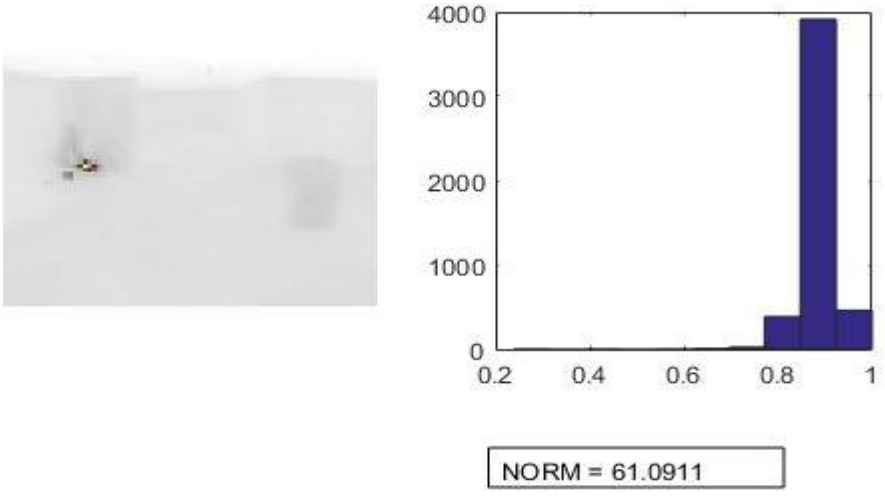
Position 4



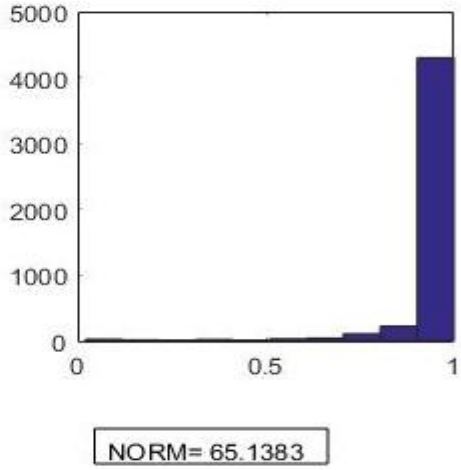
Position 5

2.4 Thermal Images and their distributions

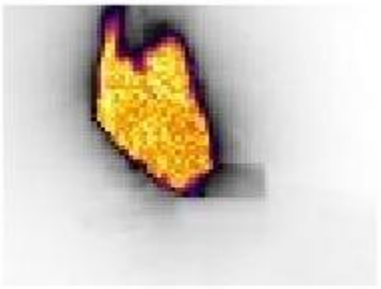
Thermal images were gathered at different distances from the fire. The image that captured from a maximum distance shows the lower content of fire. However thermal image that captured very to fire shows the maximum content of higher temperature. In order to build the relationship between the distance of robot with fire based on thermal results, a histogram of the thermal image has been plotted. The next we will find the probability based on these histograms. Thermal images with their respective histograms are shown below.

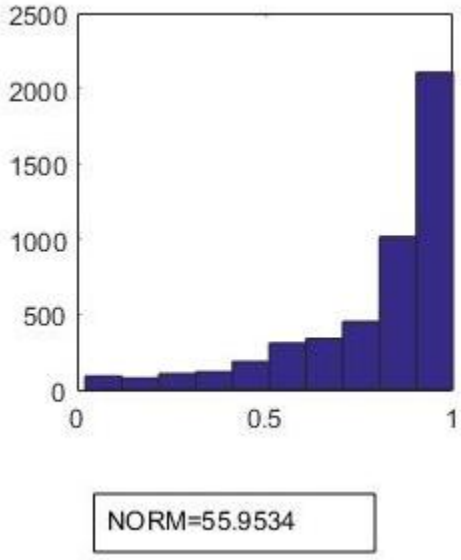


Position 1

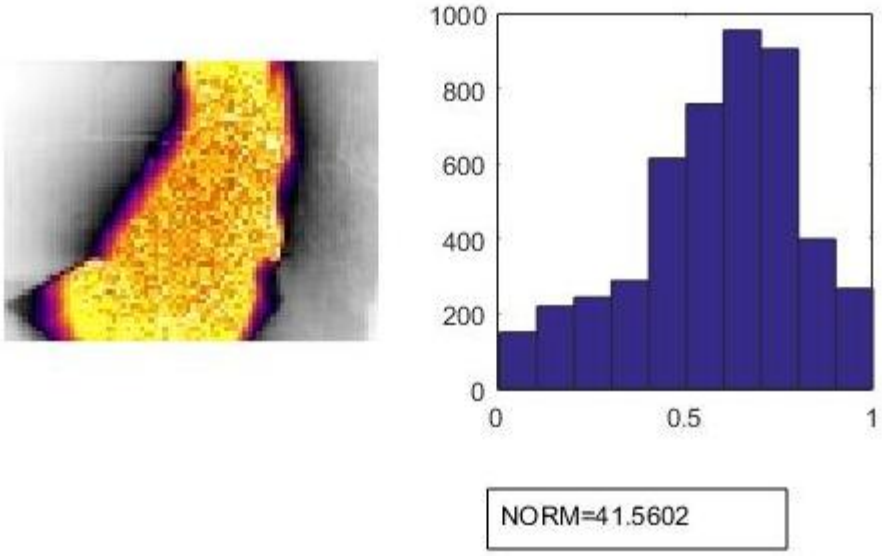


Position 2

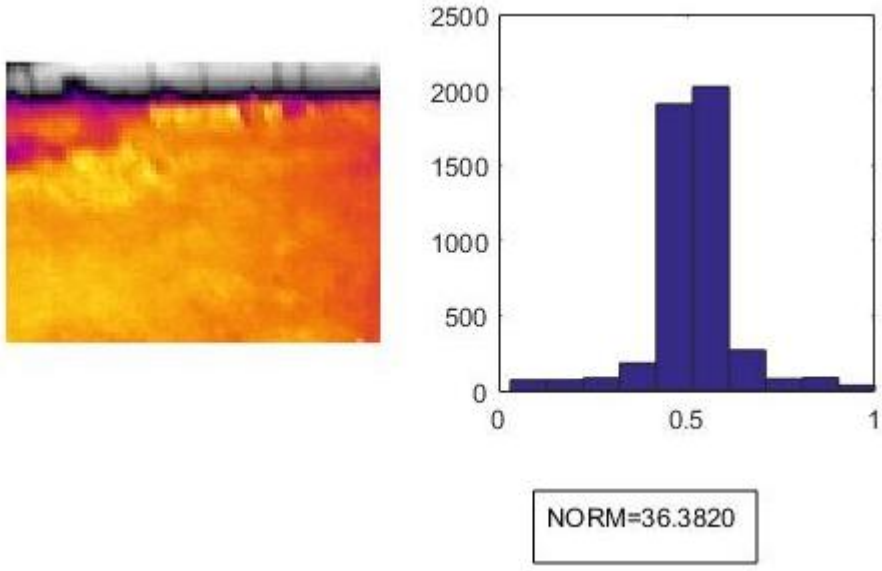




Position 3



Position 4



Position 5

2.5 Conclusion

It is evident from the test site data that LIDAR is prone to errors in fire, light and extreme temperatures. In order to build a robust navigation scheme we should fuse the sensor information of Thermal Imaging Camera and Ultrasonic Sensors.

Chapter 3. Robot formation control in lab based test environment

3.1 Overview of the lab test environment

As previously developed, the Mubassir vision based system can be efficiently used to track position and orientation of a unique marker in the test area. Lab environment tests for formation control were carried out by running the centralized algorithm in MATLAB on a PC. Zumo 32U4 robots are being used as the substitute of the full sized robot. The algorithm running on MATLAB gets feedback from the Mubassir vision system over LAN. However, movement commands to the robots are issued via a wireless serial link using the Zigbee protocol.

The layered structure of control scheme for the overall movement can be explored to test different approaches at various levels.

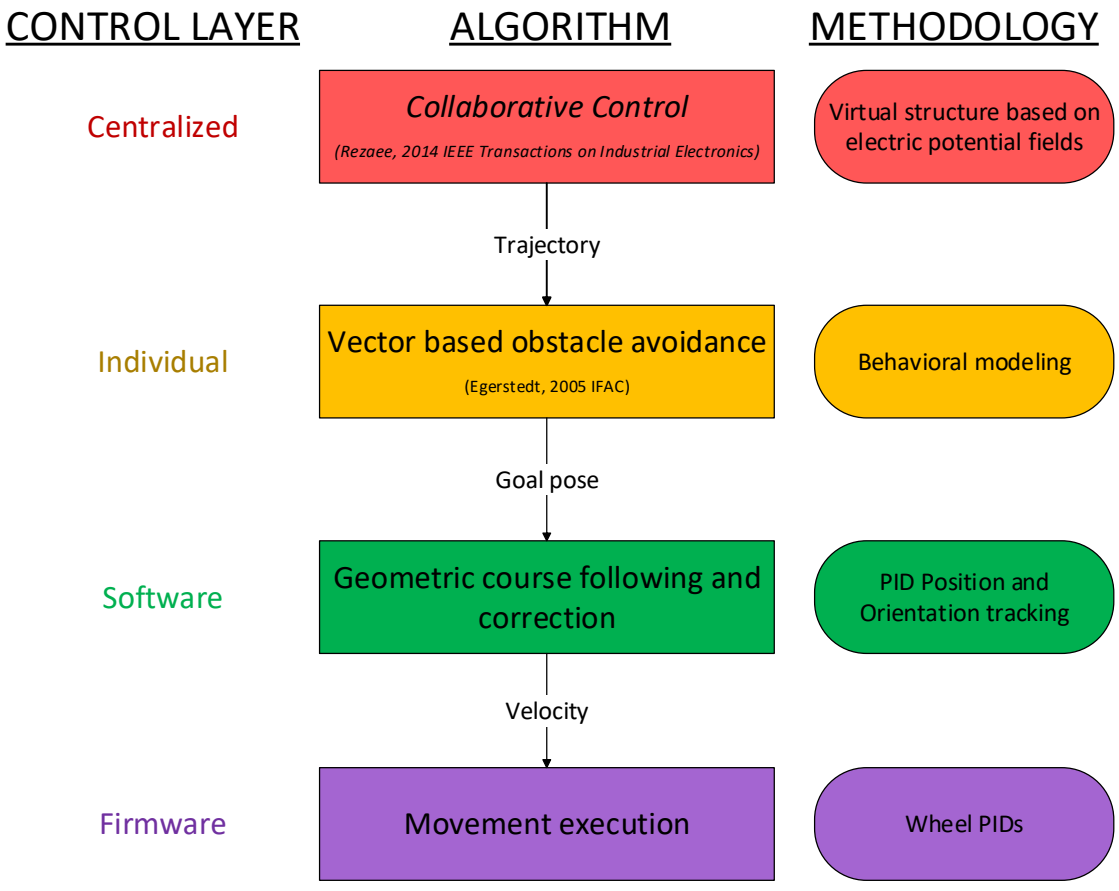


Figure 1 - Levels if control for robot movement

3.1.1 Vision System (Mubassir) feedback

The Mubassir system has been ported to use ROS messages for better compatibility with the rest of the system. Therefore, a ROS MATLAB node has to be run on the host PC where the algorithm is to be tested. Once the Mubassir RaspberryPi ROS node is started, it publishes the the X,Y position and orientation, theta, on the ROS topic. This is then subscribed by the ROS MATLAB node thus storing the latest available pose of the Zumo robots in the designated variables at an update rate of 5 Hz.

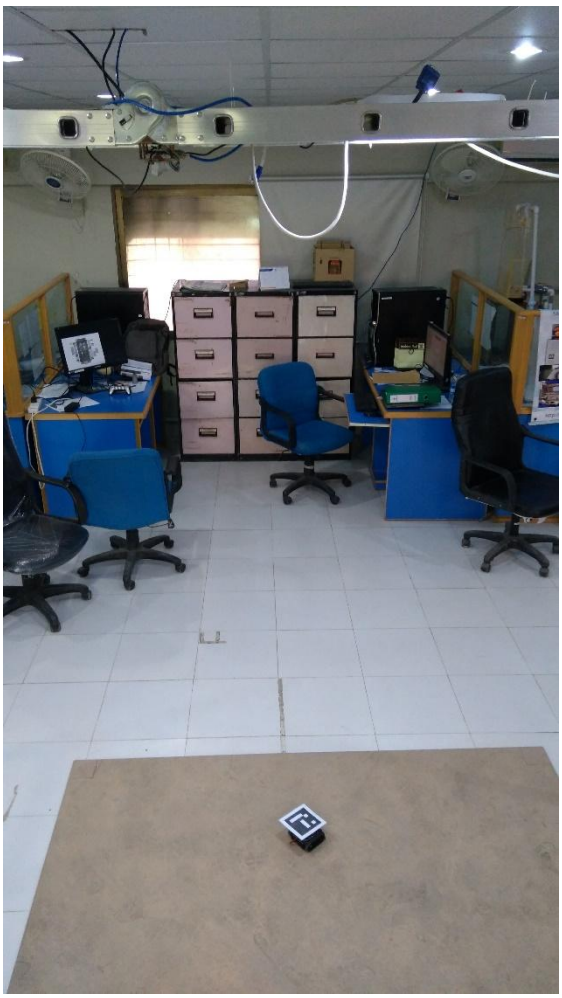


Figure 2 - Zumo and Mubassir in test area without obstacles. Axes: X (left) Y (down)

3.1.2 Firmware PID for speed control via Zigbee link

Previously, only the left and right motor PWM values were sent in the frame to the robots. But since all robots may have different response at times, battery level and slippage can also affect the reliable movement of the robot. To ensure that the remote speed commands are effectively applied, the Arduino on the Zumo 32U4 was reprogrammed. Xbee module was already interfaced with three Zumo robots for testing formation algorithm.

The firmware now includes a PID controller to maintain the desired wheel speeds sent in cm/sec from the base station. The controller uses encoder counts from the wheels to get feedback and maintain the desired speed of each motor. A significant improvement of this change can be seen as the robot is now able to move in a straight line which was not possible in the open loop scenario. As the speed commands are being received at a refresh rate of 5 Hz, the robots can be relied upon to traverse the desired trajectory without getting out of synchronization.

3.2 Trajectory following tests

3.2.1 Go-to-goal behavior

The first test was to run a go-to-goal algorithm in this setup. This simple algorithm calculates the rotation required and applies a fixed speed to reach a goal point in the X-Y plane. With further additions, it has been shown to follow a predefined trajectory by giving successive points as goal. The Mubassir system feedback is plotted in MATLAB in real time.

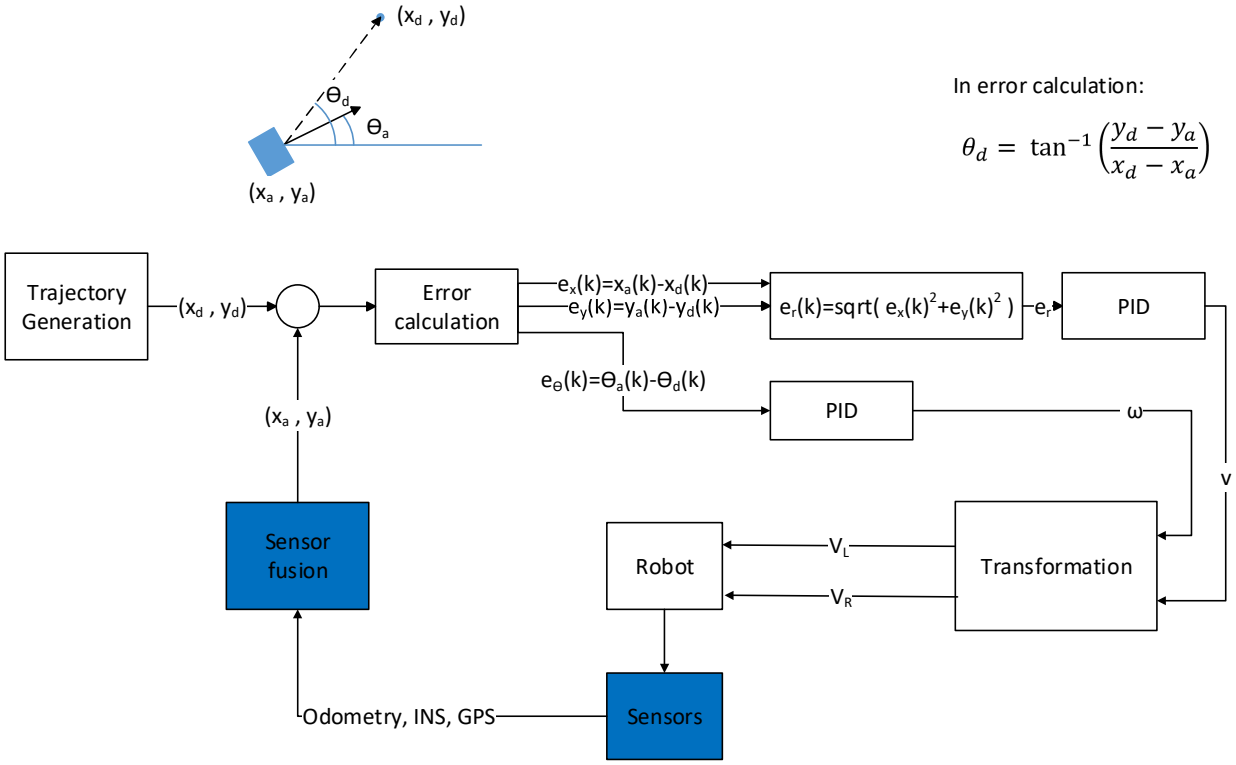


Figure 3- Trajectory following algorithm (go-to-goal)

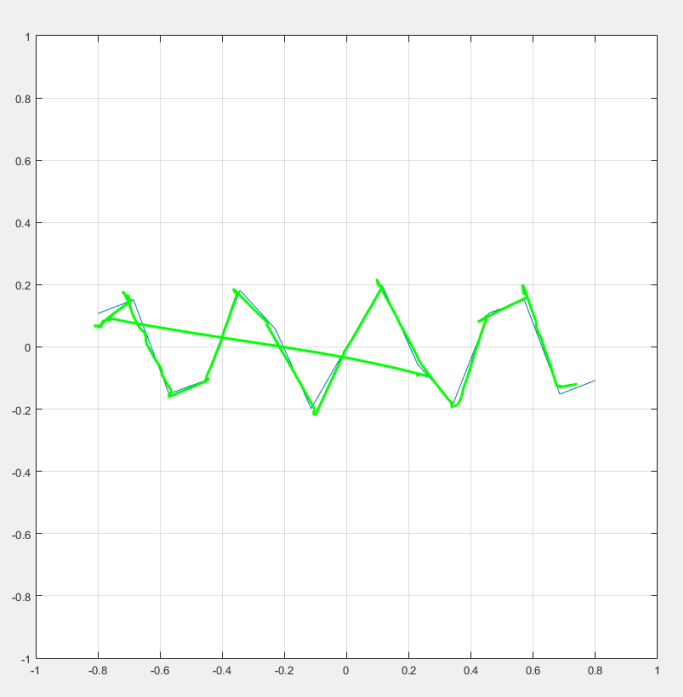


Figure 4 - Following points on a predefined zig zagged trajectory (Axes in meters)

3.2.2 Multiple robots with time bound trajectory points

In order to test whether multiple robots can function in the environment where the algorithm is running on MATLAB and the wireless channel is used for transmitting control commands for 3 robots, a trajectory following test was made as shown. It was visually seen that robots move simultaneously without jerks or delays in command execution.

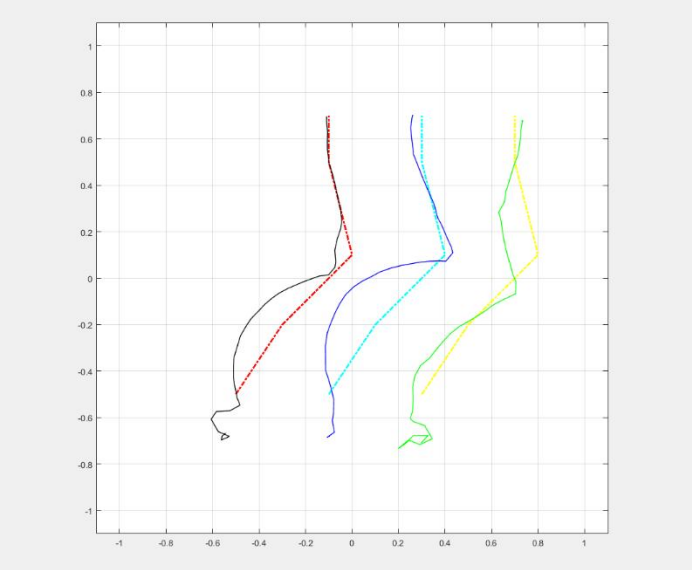


Figure 5 – Multiple robots following trajectories simultaneously

3.3 Cooperative control scheme based on virtual potential fields

As investigated during the 2nd quarter of the project, the method given in "A Decentralized Cooperative Control Scheme With Obstacle Avoidance for a Team of Mobile Robots" by Hamed Rezaee & Farzaneh Abdollahi was taken up as the first algorithm to test in the test environment. The original scheme contains point robots that can move in any direction in the X-Y plane i.e. they are holonomic.

The scheme has to be tailored first to work with differential drive robots. This is accomplished in the first step by providing a goal point based on the resulting forces of the virtual potential field. In this way, the existing framework of go-to-goal can be used without doing extensive calculations which convert attractive and repulsive forces into linear and angular velocities of for the robot.

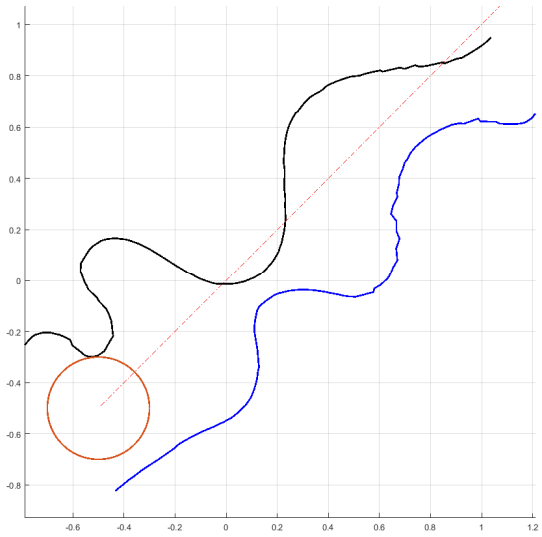


Figure 6 – Two robots moving in a formation using potential field method

The experiment designed to test the formation control can be listed as follows (see figure above for the resulting plot from vision feedback, Mubassir):

1. Two robots are expected to maintain a formation i.e. to maintain a fixed distance with each other (shown as a circle at the beginning of the trajectory) while moving in a straight line.
2. The straight line path (dotted red line) is defined by a virtual robot moving independently in the diagonal direction in the test area.
3. The two Zumo robots (black and blue) are attracted by the virtual robot (to follow the path) and repelled by each other to maintain a trajectory as well as formation.

The results show that the robots do behave as expected to about moving charges in a potential field. However, since the non-holonomic nature allows movement only in limited degrees of freedom, the robots overshoot their desired positions. As it takes some time to “turn away” from repulsion, the effect can be seen as oscillations around the straight path.

This result signifies that the technique is indeed feasible for formation control but needs modifications to incorporate the non-holonomic behavior of the robots.

3.3.1 A-priori known obstacle avoidance

The additional idea proposed by Rezaee and Abdollahi using virtual potential fields is that of avoiding an obstacle whose position is known in advance. The obstacle produces a rotating field around it which produces force impacts on the robot to “go-around” it.

A similar experiment was designed in which the robots are again expected to move diagonally across the test area but an obstacle is placed in the way. The circular magnetic field takes the direction to align with direction of approach of the robot. The results of the plots from Mubassir feedback are shown in the following figure.

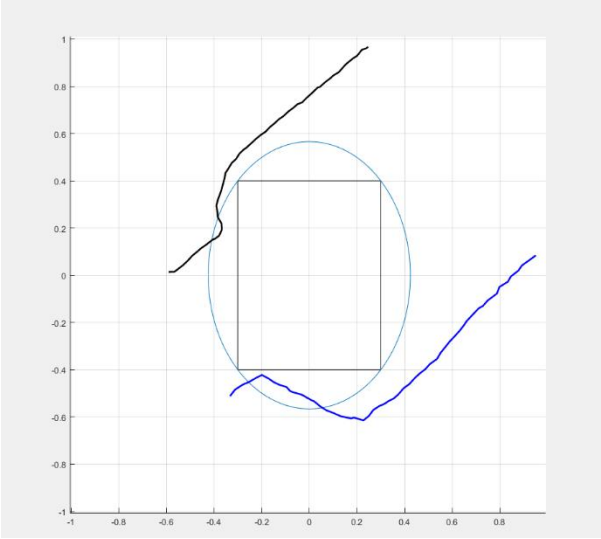
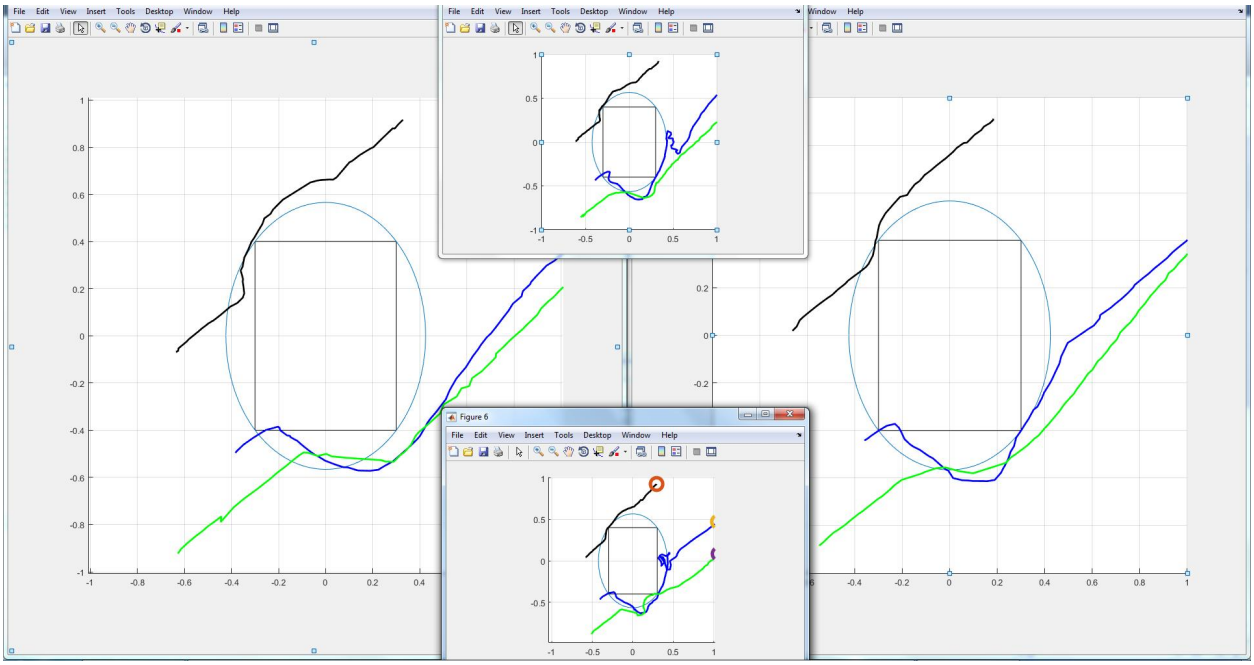


Figure 7 - Avoidance by potential field generated by obstacle

Further tests with three robots were conducted for repeatability and performance shown in the following figure



3.4 Vector based obstacle avoidance using local sensors

The shortcoming of the previous approach is evidently that position of the obstacle has to be defined a priori. For the discovery of obstacles, the distance sensors on the robot will be used in the real world scenario. In the lab environment, the Zumo in-built Infrared proximity sensors were used to test the avoidance algorithm.

This approach given by Magnus Egerstedt forms a resulting target direction vector for the robot based on the detection of obstacles from its sensors. The location and direction of sensors is used to complete the coordinate transformation to identify the exact direction of the obstacle. From this direction, an obstacle avoidance normal vector is formed. It is checked whether the obstacle avoidance goal and global goal direction are coinciding. In case the obstacle is in sight, the local direction is modified to go towards the global goal as well as turn away from the obstacle. When the robot is alarmingly close to an obstacle, the obstacle avoidance vector overrides the goal vector until such time the obstacle moves out of view.

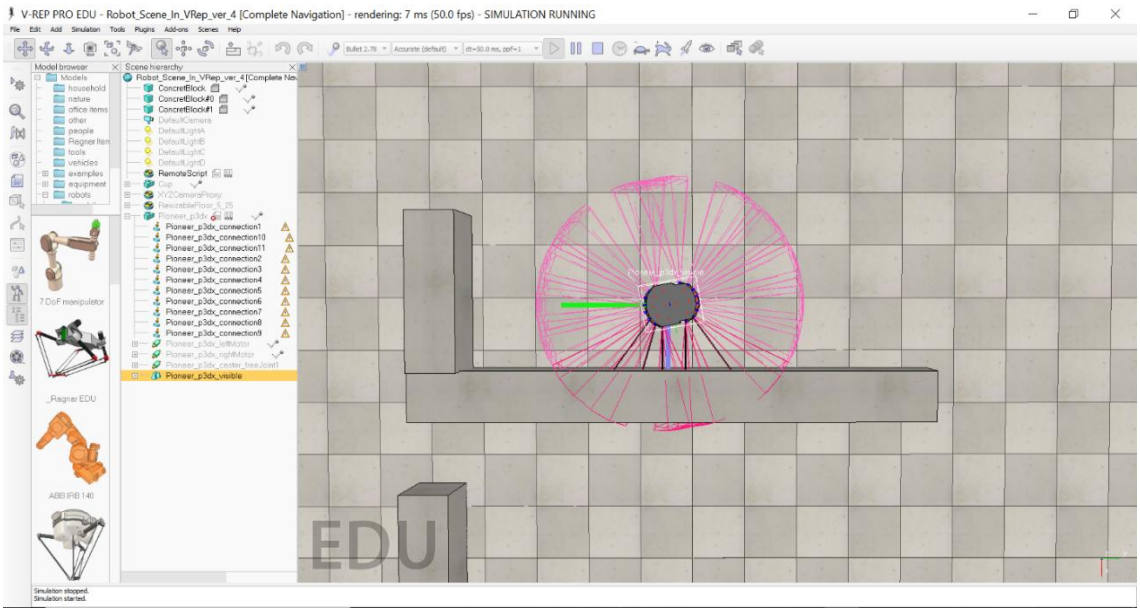


Figure 8 - Simulation in VRep environment

3.4.1 Zumo IR proximity sensors and interface

To emulate a distance sensor, the proximity sensor values have to be mapped onto different levels. For lab tests, three levels of obstacle detection are defined. Using these coarse/discrete distance levels, a proof of the concept can be verified experimentally.

Firstly, a receipt of sensor values from the individual robots over the Zigbee network at the base station MATLAB was first established. The sensor values are received at an update rate of 4 Hz due to limitation caused by time taking nature of the sensor reading Zumo library routines. The values received in MATLAB serially are updated by a callback function into global variables and thus accessible to the central algorithm.

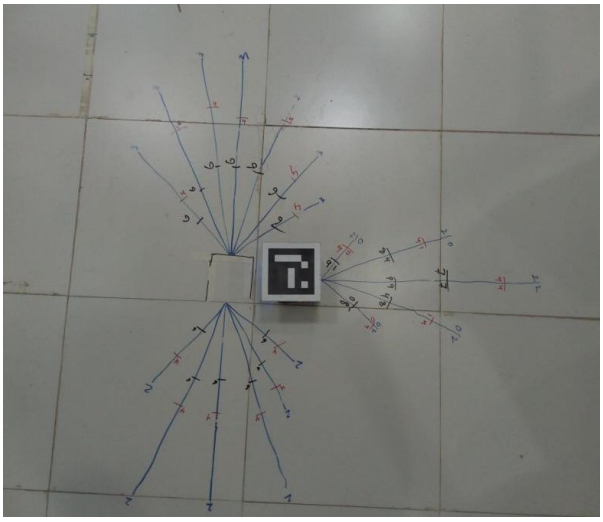


Figure 9 - Testing sensors sensitivity at distances and angles

The Mubassir system plotting algorithm was extended to mark obstacles detected by each sensor is marked on the test area along with the color showing how close to the robot was the obstacle detected.

3.4.2 Results with single robot

It is expected that the presence of multiple robots will cause interference of the sensors as well as the possibility that another robot may be detected as an obstacle. The results of similar experiment for reaching a goal point at the opposite diagonal end with a walls of obstacles in the center of the test area is shown from Mubassir's vision feedback.

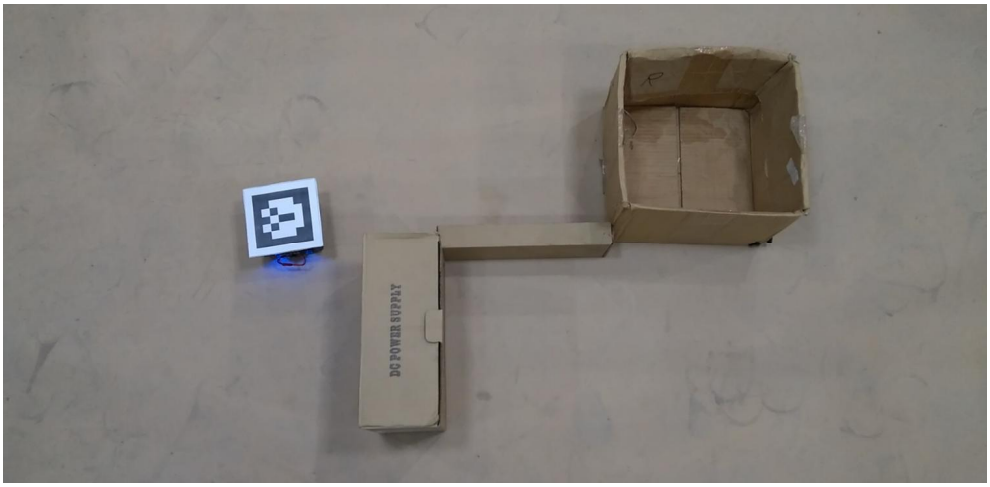


Figure 10 - Zumo navigating around the obstacles

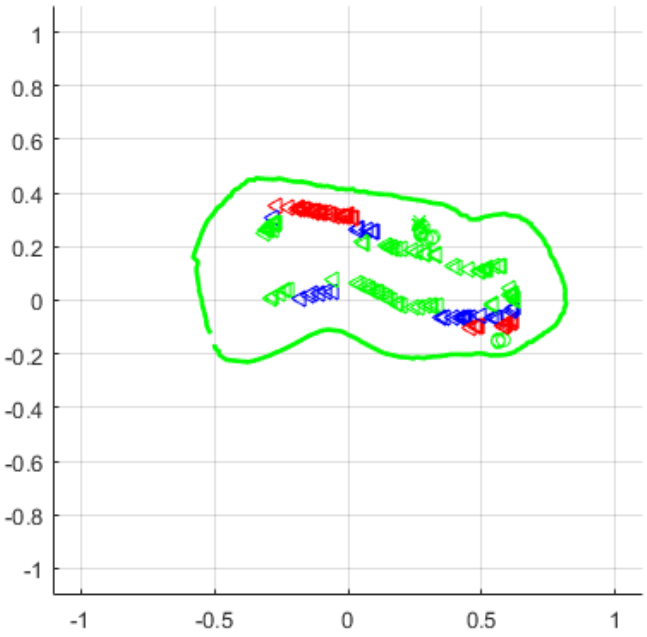


Figure 11 - Plot of practical run around a horizontal wall (showing points on which obstacles were detected)

It can be seen that the algorithm performs well in avoiding simple obstacles i.e. when approached from the front or side. However, certain conditions may occur where the robot may get stuck in a repeated loop of jumping between obstacle avoidance and global goal following modes.

3.5 Limitations of tested algorithms

The main limitation of potential fields method is that the obstacle locations and dimensions must be known in a priori. Obstacle avoidance discovered by local sensors will need a lot of mathematical effort to model.

The vector based method is meant to be applied on a single robot without any formation algorithm. First, multiple robots will be run simultaneously with independent behavior too see if the can avoid one another as well as obstacles. Then, a method to utilize both techniques will need to be formulated.

3.6 Way forward

A hybrid strategy for combining formation control with obstacle avoidance needs to developed for the overall goal. For this reason, a proposed model is given as state flow diagram which will be first explored in simulation and if feasible, verified experimentally.

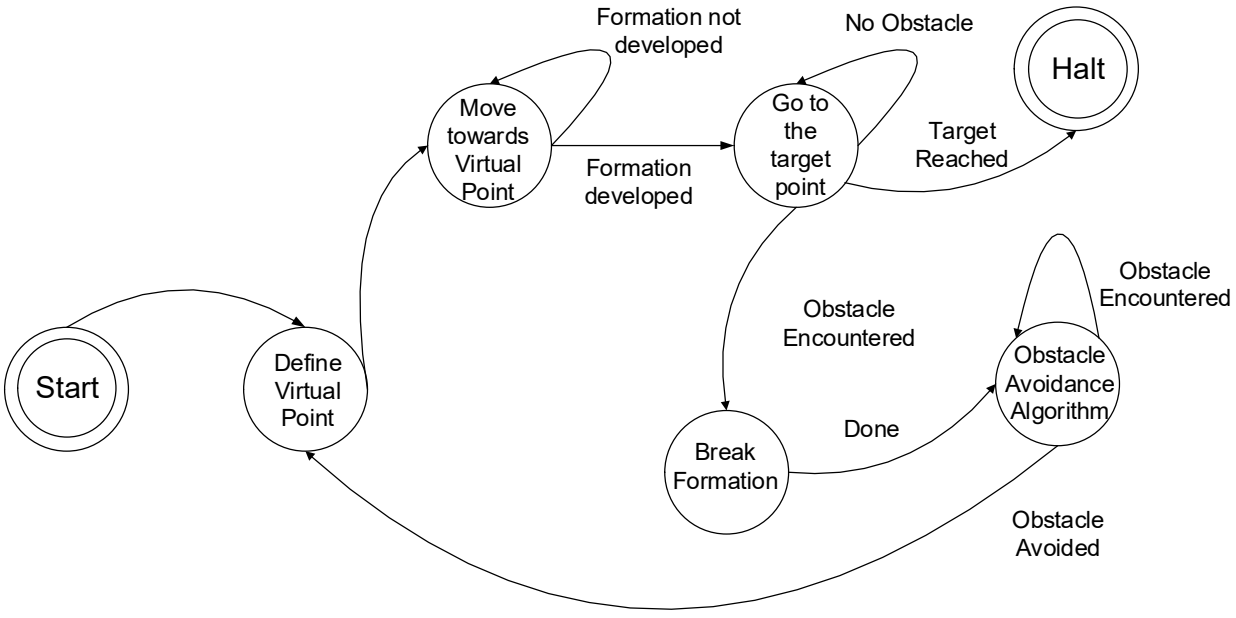


Figure 12 - Proposed FSM for algorithm switching

Chapter 4. Working performance of IMR and improvements in the design of the mechanical structure of IMR

4.1 Initial design of the IMR

For the purpose of this project, several IMRs, Intelligent Mobile Robots have been designed, each catering to a specific task and built to test a specific capability and functionality, in terms of different sensors and different methods of locomotion. The need for this experimentation has stem from the fact that there is no standard model or practice for designing such robots, which are able to douse fires and traverse to locations which are difficult or dangerous for humans. We have developed all the robots in house utilizing as much local resources and talent as we can, including keeping close check on the financial feasibility and the later production of such robots in future for commercial applications.

4.2 IMR Robot Models

The Following are the different IMR Robot Models; 2 robots were made with wheeled differential drive while the third IMR robot is based on belt driven design. The 2 wheeled robots are actually iterations of the same design with refined functionality and improvement in the overall design form the first model of the IMR.

4.2.1 WHEELED IMR ROBOT

The wheeled IMR Robot model is a differential drive robot with skid drive, referring to both wheels of the same side moving due to a single motor. The wheeled robot has been tested extensively on the off road and rough terrain around PAF KIET Main Campus grounds and has been found to be very stable, working on highly uneven and inclined surfaces as well as on different types of terrains, road, mud, gravel etc.

There are 2 different models made of the Wheeled IMR Robot, the second model improves upon many of the issues faced in the first model. The main difference in the wheel size and the motor selection, the motors used in the first variant seemed to to be under powered and hence the second model has bigger motors, providing a higher torque at lower RPMs.

Figures 4.1 to 4.4 show the CAD diagrams of the wheeled IMR Robots, including inner and outer renders and also a top view, showcasing where the ultrasonic sensors are placed with respect to the robot body.

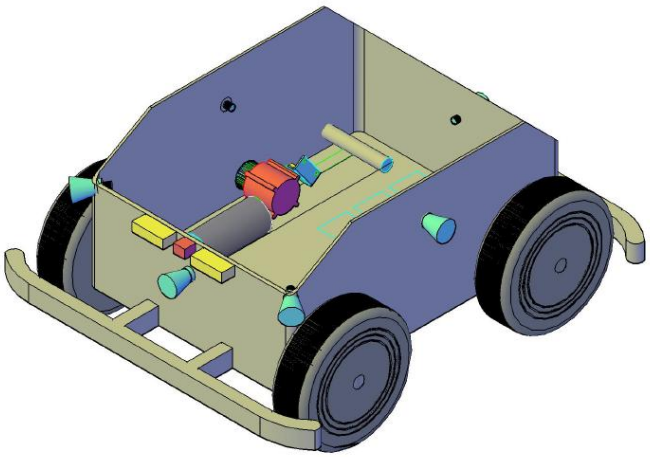


Figure 4.1: IMR Wheeled Robot – V1, inner view

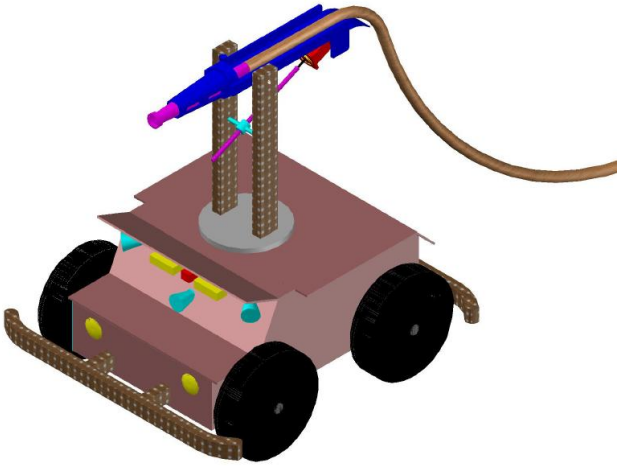


Figure 4.2: IMR Wheeled Robot – V1, complete render with fire hose

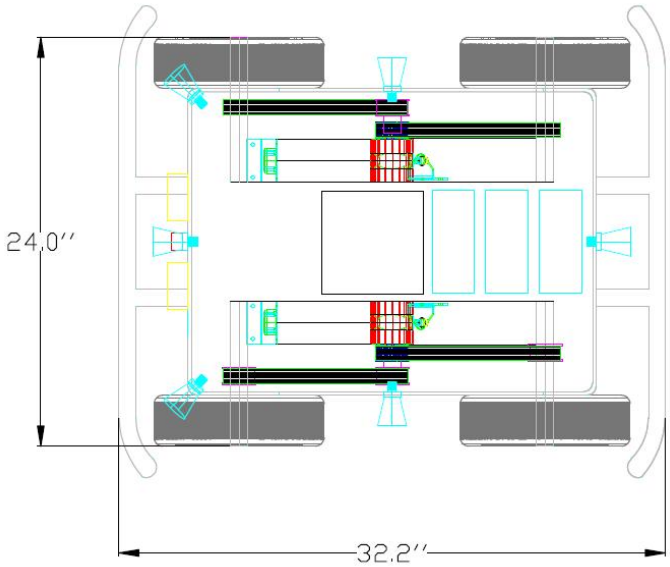


Figure 4.3: IMR Wheeled Robot – V1, Top View with outside dimensions

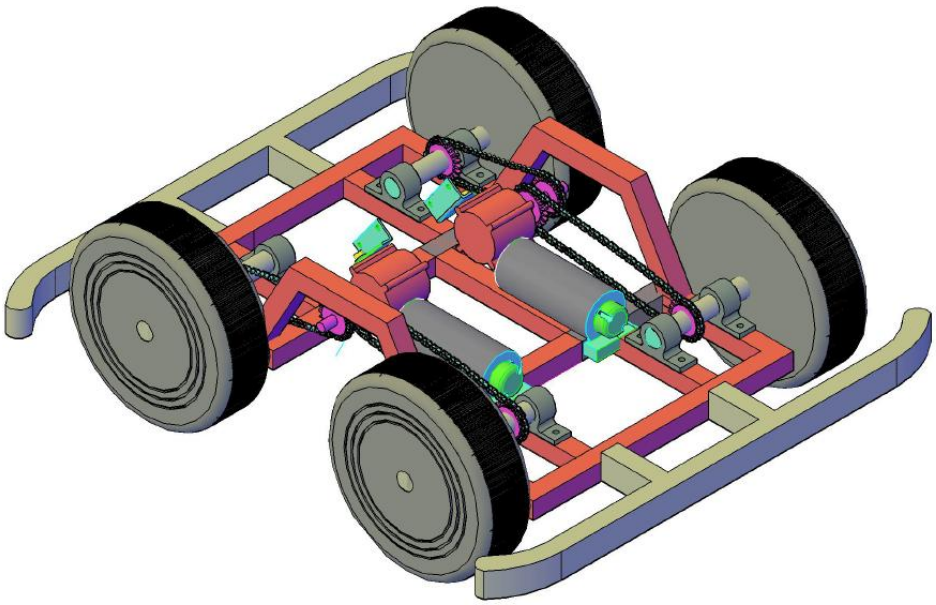


Figure 4.4: IMR Wheeled Robot – V2, inner chassis and motor and wheel assembly

4.2.2 BELT DRIVEN IMR ROBOT

The belt driven robot has been the biggest challenge so far as the manufacture of the belt has provided different issues to tackle. The belt driven robot also required the most powerful motors of the three robots as it has the highest surface contact area and hence the highest friction, however this also results in better traction and stable drive once it starts moving a greater payload capacity. Figures 4.5 and 4.6 show the robot body, with a basic render, this robot will be attached with a Robot arm instead of a fire hose to be used as a way to clear debris and move items from one place to another. The same can be used for security related applications too, i.e. bomb disposal etc.

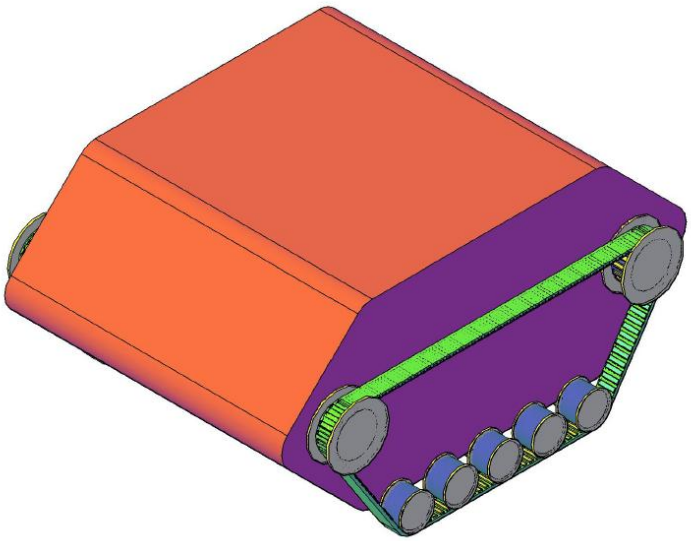


Figure 4.5: IMR Belt Robot – outer render with basic mechanism

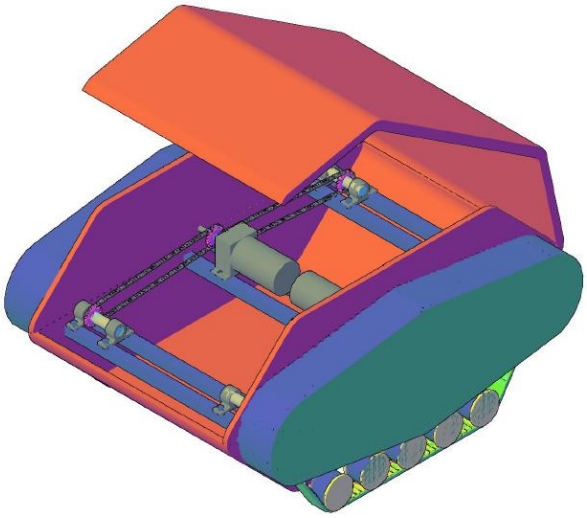


Figure 4.6: IMR Belt Robot – internal mechanism and chassis

4.3 Robot Comparison Matrix

In the initial project proposal, a robot comparison matrix was included which detailed specifications and capabilities of some of the robots in use at the time. We have done a similar study with the current robots from IMR Lab and a PAROSHA GO SAFER Robot by a German company.

Robot Specifications Comparison IMR LAB Matrix					
Description	**Unit**	**Parosha GOSAFER**	**IMR Robot 1**	**IMR Robot 2**	**IMR Robot Belt**
Length	mm	1325	735	785	420
Width	mm	700	600	655	690
Height	mm	485	530	470	420
Underbelly Clearance	mm	110	65	90	100
Track	mm	600			585
Track Width	mm	130			70
Wheel Dia.	mm		250	295	

4.4 On Ground Testing of Robots

The 3 robots were tested at PAF KIET Main Campus ground and the test site constructed there, the test procedure involved taking the robots from the workshop to the test area by remote, and then traversing the test site and the terrain around it, the test site and the accompanying grounds are shown in figure 4.7 and 4.8

Figure 4.7: IMR Test Site at PAF KIET Main Campus, Korangi Creek

Figure 4.8: Still from a Robot TEST Video, PAF KIET Grounds and the robot

During the tests it was observed that the robots will need manual intervention as and when necessary as sometimes the sensors don't provide reliable information and the current suite of sensors is NOT necessarily accurate enough to localize in a world map, as the GPS sensors we have, have tolerance in 5 to 10 meters. However once the robot has been steered even roughly towards the target the autonomous mode can kick in and the robot can orient itself and get closer to the target co-ordinates as needed.

The tests also included firing water from the hose of the robot, this was accomplished by having a water pump send water from a reservoir to the robot via the attached fire hose and the robot was able to aim the nozzle towards a target upto 40 feet away without losing aim or stability, the robot was also tested to be able to pull the filled water hose along its path, as the weight is so much that it usually requires several fire fighters to carry the hose near the seat of fire. Figure 4.9 shows the IMR Wheeled robot v1 with water action.

Figure 4.9: IMR Wheeled robot, v1, Water Tests

4.5 Problems and challenges faced in the initial testing

Majority of the problems faced can be categorized as either hardware issues where the manufacturing and selection has had problems or that the robot problems have stem from lack of precision sensors and instruments. Some of the major problems faced are:

a) Selection of Motors and Availability
b) Water Proofing of Robot Chassis
c) Heat/Thermal Insulation and Proofing
d) Sensor Capabilities differ indoor and outdoor
e) Network and Communication Issues
f) Power Circuit for Motor Drive
g) Manufacture of Drive Belt

4.6 Proposed modifications in the design of the IMR

Majority of the problems faced while working with IMR robots were resolved by modifications to the designs, these modifications included changing of the power motors, electronic circuitry re-design with single PCB for all sensor interfacing and proper cabling for reliable wiring of the sensors and actuators.

4.6.1 Power Circuit for Motor Drive

One of the main issues faced was power circuit for driving of motors, as changing the motors on the robots, resulted in higher power motors which the initial circuits were not capable to drive safely, we have tested several different techniques for the solution of this issue, including

- Relay based Motor Drive
- High Current Motor Drive ASICs (BTS7960)
- Solid State Relay based Motor Drive
- Hybrid Drive, Relay+MOSFET Based

The final solution we have determined is to use the BTS7960 on the IMR Wheeled Robot v1, which has the least powerful motors and then use the Hybrid approach for the other 2 robots.

4.6.2 New PCBs and Connectors

In order to make the system more reliable, several changes were made to the electronics circuitry, the primary change is using IDC connectors and Ribbon Cables for interfacing of off-board sensors and also using newly designed PCBs which house all the relevant circuitry and sensors in a small area.

Chapter 5. Issues faced in the robot localization

5.1 Introduction:

Robot localization is a key problem in mobile robotics .Today many R&D activities designed to conduct using a robot requires a system for proper localization of a robot. These include the applications such as navigation of a robot to a desired path and robot tracking.

Various sensors are available for providing position estimates of a moving object such as a robot. Most of the sensors when used alone are not capable of providing an accurate object localization solution. Also there are issues when some sensors such as GPS systems that are considered as a reliable source, become unavailable in an indoor environment. Other sources of inaccuracies in position measurements are due to sensor errors such as bias and sensor noise. This chapter discusses the major issues that are encountered in the determination of accurate location of a robot, such as:

a. **Errors in GPS Signals**
b. **Integration Errors in Dead-Reckoning Sensors**
c. **The Need for Sensor Fusion**
d. **Sensor Data Transformations**
e. **Wheel Slippage Errors in Odometers**
f. **Availability of GPS Signals**
g. **Possibility of Localization in Indoor as well as Outdoor Environments**

These issues are described as under.

5.2 Issues in Robot Localization:

5.2.1 Errors in GPS Signals:

The Global Positioning System GPS is widely used as a positioning system in a variety of applications. However, the GPS signals possess errors that are Gaussian in nature. Therefore, a filtering system is required such as *Kalman Filtering* in order to smooth out the randomness in GPS based measurements and provide a reliable estimate of position of an object such as a robot.

5.2.2 Integration Errors in Dead-Reckoning Sensors:

Various sensors data can be manipulated in order to obtain position. These sensors include IMU based sensors such as accelerometer, gyroscopes and a wheel sensor such as odometer. These sensors can be easily mounted on the robot of which position is to be measured. The output from these sensors has to processed in order to obtain position information. Using accelerometers, the position can be obtained by double integrating the position values. Similarly, the position can be obtained from odometer by integrating the measured velocity. The process of integration in order

to obtain the position of an object is called *dead-reckoning*. Due to integration, the errors in position measurements due to many noise sources in these dead-reckoning sensors are also increased.

5.2.3 The Need for Sensor Fusion:

Due to error sources described above sensor fusion of various sensors is required in order to produce a reliable position estimate of a moving object. This involves collecting data from various sensors and combining them using a sensor fusion algorithm, such as Kalman Filtering.

5.2.4 Sensor Data Transformations:

In order to implement a Kalman filter using the available sensors data, it is important to transform the data coming out of various sensors. The Kalman filter for filtering the GPS based position data using dead-reckoning sensors requires this transformation.

The dead-reckoning sensors that are used in localization scheme such as IMU and odometer register object movement information in the body frame. This is the frame that is attached to the body of the sensor, commonly known as the 'b-frame'. Whereas the global positioning system such as the GPS system provides the object localization data in the form of geodetic co-ordinates, i.e, latitude, longitude and height in the units of degrees, degrees and meters respectively. In order to obtain the robot position data from these units to a common world co-ordinate system, for instance, East North Up co-ordinate system, ENU, North East Down, NED co-ordinate system or ECEF co-ordinate system various transformations are required. For the computation of these transformation, the orientation information in the form of ***'Euler Angles'*** are needed which is obtained using gyroscope data.

5.2.5 Wheel Slippage Errors in Odometers:

In a localization system using wheel odometers, the errors due to wheel slippage in velocity and hence position measurements registered by an odometer cannot be ignored. The wheel slippage is the relative motion between a wheel and terrain surface. The wheel slippage occurs when a moving object such as a robot is obstructed by an obstacle. The wheel continuously moves registering false velocity measurements whereas the robot is not moving. These velocity measurements lead to false position measurements.

5.2.6 Availability of GPS Signals:

GPS signals provide position information but in certain cases, the GPS signals are blocked and hence position data also becomes unavailable. The GPS signals may not be available due to following reasons:

1. GPS outages

2. Satellite Blockage

The satellite signal is blocked in indoor environments such as buildings and other encapsulated spaces such as tunnels. In such cases, sensor fusion algorithms such as that integrate GPS data with other sensors such as IMU or wheel sensor will not operate and some other mechanism for localization has to be designed that can provide robot localization information.

5.2.7 Possibility of Localization in Indoor as well as Outdoor Environments:

As discussed above, a GPS based positioning system is capable of providing positioning data in environments where GPS is available, i.e, outdoor environments. In case of indoor environments, GPS would not work other sensors such as odometer and INS are able to provide position information in both the presence and the absence of GPS signals but these sensors do not guarantee long-term accuracies in position measurements.

The proposed localization scheme is designed to handle all these localization issues. This scheme is summarized as follows:

5.3 The Proposed Robot Localization Scheme:

The proposed localization scheme is based on usage of Artificial Intelligence-AI such as *Artificial Neural Networks (ANN)*. The localization scheme aims to provide a complete localization solution that can be used for both indoor and outdoor positioning applications using various available sensors including *GNSS* system such as *GPS* and *dead-reckoning* sensors such as *INS* and *odometer*. The localization scheme utilizes kalman filters for fusing various sensor data in order to provide a filtered signal output from the GPS during GPS signal availability. In the absence of GPS ANN is used to train and predict the position estimates.

This localization scheme works in two modes of operations: the ***update*** mode and the ***prediction*** mode. These names have been chosen according to the functionality of *ANN*. In the update mode, the ANN is trained on the object motion data provided by sensors such as accelerometer, gyroscopes and odometer. The weights of the MLP based ANN are updated using the sensor data. When the GPS signal is lost, the trained ANN switches to prediction mode and provides predicted estimates of position of a robot based on the input data provided by the same sensors.

Chapter 6. Commercialization Plan

Presentation attached.