Contents

AndroBot Documentation

Robotics club, IITK

July 2017

1 Aim of the project

The aim of the project is simple and self explanatory from the name of the project, which is to exploit the features of an android smart phone to guide and control a robot. These features include the various sensors of the phone and the android platform itself for processing and controlling the robot.

2 Abstract and Objectives

The completed bot will have features such as Remote Control (through Internet or Wifi), Indoor WiFi Navigation(Coordinate mapping).

3 Modules

Different modules of the project are-

3.1 Android Team

The goals of this part of the project are as follows:-

- Configuring the gyroscopic sensor, compass and accelerometer for easy movement
- Basic movements of the bot and and controlling it using a microcontroller
- Indoor navigation using wifi
- Using camera for long range remote control through another phone over internet
- Integration of all the modules on an interface for simultaneous working

3.1.1 Objectives completed

- Extracted calibrated data from sensors of android Smartphone namely Camera, Proximity, WiFi, Accelerometer, Ambient Light, GyroScope.
- Serial Communication between Android Phone and Arduino Uno
- Basic Movement of Bot and remote control (using TeamViewer app)
- Extracted Wifi Signal for each available SSID.
- Server - Client communication using bluetooth and Wifi hotspot
- Live feed of camera added in the control app
- Automated movement towards wifi router by using signal strength
- Finding the location of bot using wifi navigation and coordinates

3.1.2 Problems faced and their solutions

- Interacting with the hardware of the smartphone is very difficult and requires deeper understanding of the software interface.

The developer website of android provides a good reference for the above. A lot of queries are also answered on the stackoverflow website.

- One of the major problems faced was the serial communication between the android app and arduino.

  This was solved by including additional usb libraries. GitHub was really helpful in this matter.

- To help the wifi navigation team taking wifi data, we made an app for data management.

3.1.3 Unresolved Problems, if any

- The data points provided for location and Wifi Signal Strength are less even after increasing their number resulting in some significant error.
- The orientation sensor, i.e. magnetometer, is very slow in its data updating which makes it very complicated to use it for precise angle rotation in its least count. It has been tested on the bot and the errors are very large.
- Accelerometer could not be used to measure the speed and distance by integrating over time because of its abrupt and sudden changes. It was also tried on the bot and gave significant errors.
- Image processing could not be done due to large processing and time required for it.

3.1.4 Future Work and timeline

- Calibrate movement of Bot for controlling distance travelled
- If above objectives are completed then implement voice recognition, image processing etc. in our bot.
- Refining the bot movements further.
- Solving the live feed transfer to remote device on low internet.
- Solving usb callback problem so as to provide better processing through android.

3.1.5 Links for code and tutorials of what you have learnt

```
Many members were new to programming on Java, Arduino and xml.
Java beginners-
    https://www.youtube.com/watch?v=Hl-zzrqQoSE&list=PLFE2CE09D83EE3E28
    https://www.tutorialspoint.com/java/
    https://www.javatpoint.com/java-tutorial
```

```
    https://www.udemy.com/java-tutorial/
    programmingknowledgeblog.blogspot.com/p/jave-code.html

Java advanced-
    Java in a nutshell - O'Reilly


Arduino-
    https://www.youtube.com/playlist?list=PLZfay8jtbyJt6gkkOgeeapCS_UrsgfuJA


Xml introduction-
    https://www.youtube.com/watch?v=n-y-YHVZSwk&list=PLsyeobzWxl7oanwaonj3iV3rxfTdEA2gC


Android Studio Basics-
    https://www.youtube.com/watch?v=09zfRaLEasY&list=PLZfay8jtbyJt6gkkOgeeapCS_UrsgfuJA
    https://www.youtube.com/watch?v=fCxzA9_kg6s&list=PLA567CE235D39FA84

Android App Development
Basic-
    https://in.udacity.com/course/android-development-for-beginners--ud837/
    http://programmingknowledgeblog.blogspot.in/2015/02/android-tutorial-for-beginners-6.htm

Advanced-
    https://www.udacity.com/course/new-android-fundamentals--ud851
```

3.2 Indoor localisation using Wifi

3.2.1 Aim of Project

Our target of project is to determine the position of our bot called Androbot. It means we have to find the x and y coordinates of the bot.

3.2.2 Abstract and Objective

As we have mentioned we have to find the coordinates of our bot we decided to do it by Wifi Localization Technique. So first, we establish five routers in the arena where we move our bot. Our bot will measure wifi signal strength from each routers in terms of RSSI values (Received Signal Strength Indicator). After knowing RSSI values from each router we have to apply a suitable algorithm which will calculate the coordinates. We decided to do it by fingerprinting method which has two stage calibration and navigation stage. In the calibration stage we have to take 9-10 data of RSSI values for each coordinate from every router in the arena. In the navigation stage we fed these data into an algorithm so that if we input RSSI data of five router in any

arbitrary coordinate it will give the coordinates of the point. we have four algorithm but we have to determine which gives minimum error.

But wait, There is a good amount of error in it. So instead of having sole dependence on signal strengths we also have to determine its position by knowing its previous state or position and also its velocity and acceleration before calculating the current state. All these data to be fed into a suitable algorithm which will give most accurate position of current state. We decided to apply Kalman Filter to achieve this. We have to write the code of kalman filter in an android app. The app takes data from the phone wifi adapter and take the readings of accelerometer, gyroscope, rotation vector sensor of the phone. The app will use all these data and kalman filter algorithm to determine the coordinates of the bot.

First we have to understand what is Kalman Filter. Kalman filter predicts final position of the bot by considering the position obtained from wifi signal strength and the position obtained by kinematics. The filter gives weight-age to both these position according to errors in calculating these positions and finally calculate the position of the bot. When we solve Bayesian Filter functions by using Gaussian distribution function the solution which we get is known as Kalman Filter.

We make our algorithm for kalman filter and bot movement. We decide to make our bot autonomous which means we give the desired location of where we want to go and it will reach that location automatically. So is here is what our algorithm does. At first it will locate its position by wifi signals. Then it calculates the angle of the direction in which it has to go from x-axis. Then we will calculate the orientation of bot that is the angle which it makes with the x-axis. Then, we calculate required degree to turn the bot by taking difference between these two angles. Then we will call arduino to make the bot turn up-to required degree. Then we make the bot to move in the new direction for two seconds and meanwhile calculating acceleration from accelerometer sensor in phone to know the position by kinematics. Then we switch off the arduino.Then we calculate the position of bot by wifi signals. After it, we feed both these position in kalman filter to predict the actual position of bot. Now we get our position. Then, again we again find the angle of direction bot has to go from x-axis by knowing the target points given already. Then, again we calculate the orientation of our bot from x-axis by orientation sensor of mobile. Then repeat the same process until we reach our target region. Here region is the circle of 1 unit radius and target point as centre.

3.2.3 Work done so far

- We have read about kalman Filter algorithm.
- we did the calibration stage in which we we took 10 data of RSSI values for each coordinate in the arena from five routers manually.(we took 2500 readings in total).

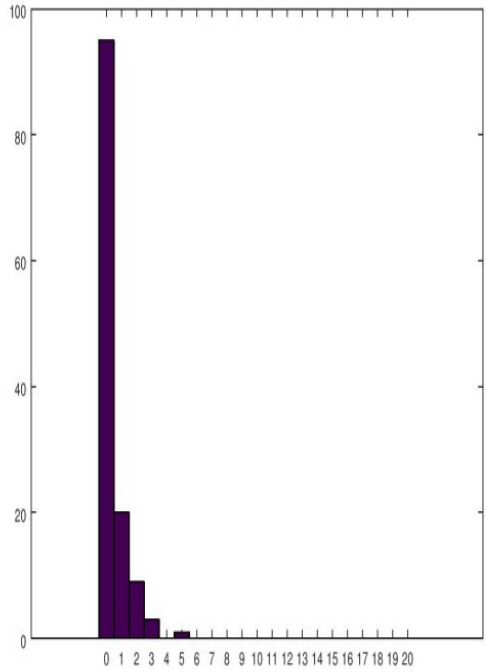
- We studied Octave language to apply the fingerprinting algorithm.
- We studied the four algorithms from research papers to get the coordinates in navigation stage. Those are namely-
    1. k nearest neighbour
    2. weighed k nearest neighbour
    3. maximum likelihood
    4. Euclidean distance
- We did the navigation stage by coding four algorithms to check has better accuracy.We did coding in octave and also build histograms of each algorithm for error. Weighed k nearest neighbour gave minimum error.
- We read object oriented programming in java to build an app to apply kalman filter.
- We wrote the code of Kalman Filter and bot movement in java.
- We finally took 10 readings in each and every point in arena. We have 64 points in our arena. We used 8 readings as our training data and 2 readings as testing point. Now we have 128 points to test. We plotted the histogram in which x-axis shows error distance means distance between actual and predicted point by wifi. The y axis shows the frequency of error out of total 128 testing points
- We combine our code in android to develop an app to run the bot.

3.2.4 Problems faced and Solution

- We had to test and find the best algorithm that will give us the most accurate prediction about the co-ordinate of the bot. For this we tried four algorithms which include: 1. k nearest neighbour 2. weighed k nearest neighbour 3. maximum likelihood 4. Euclidean distance

  We found that for fingerprinting method 'weighed knn' gave best results.
- In fingerprinting method we have to take the readings manually at each and every point. It requires much labour. So we made the app to take readings and complete the task of taking readings in one hour Otherwise it would take 5-6 hours.
- The accelerometer of the android phone is not giving correct values. It is giving random values which was not fixed.We tried to solve the problem but it was not solved. So we decided to put the acceleration zero and velocity constant. We calculate the average velocity of bot manually and feed into the code.

Figure 1:



3.2.5 Unresolved Problems

The accelerometer of the android phone is not giving correct values. It is giving random values which was not fixed.

3.2.6 Future aspects

We can have different paths for bot which will make the bot to reach its destination in shortest possible time and also avoid obstacles.

3.2.7 Links of code and tutorial

For Java: https://www.javatpoint.com/java-oops-concepts

For android: https://developer.android.com/guide/topics/sensors/sensors_motion.html, http://thenewboston.com

For KALMAN FILTER:
https://www.youtube.com/watch?v=CaCcOwJPytQlist=PLX2gX-ftPVXU3oUFNATxGXY90AULiqnWT

For Octave: https://www.youtube.com/watch?v=MBEDRv3L9Sklist=PLHj4uM4X8Oso_M2yN6Fg0_Mp219z6W

3.3 Mechanical and Electronics

Responsible to make design of bot and electrical circuits.

3.3.1 Objectives of Mechanical and Electronics Team

- To make design and the body of bot.
- To make elctronic circuit for the bot movement and connetivity with android phone.
- Using ultrasonic sensors for obstacle avoiding.

3.3.2 Work Done/ Objective Completed

- Design and manufacturing of bot completed.
- Circuit of bot is completed and has been tested for bot movement.
- Obstacle avoiding mechanism is ready and has been tested
- We also tested the bot to run it by giving commands from the mobile.

3.3.3 Problems Faced and Their Solutions

3.3.4 Problems

- At the starting we had very less exposure to Inventor fusion.
- Problems faced in flanging and assembling the parts
- Faced problem in giving design a real form due to many curves in it.
- Problems faced in making dxf file for cutting
- Little idea about micro-controller here arduino.
- No idea about how to use ultrasonic sensors.
- Some problem faced while bending the flat parts of bot.
- Fixing the dc-motors in bot.

3.3.5 Solutions

- Attended Workshop on Inventor and practiced a model given by our club coordinator.
- Saw the tutorial video on flanging and assembling on youtube and practiced it.
- Removed non-necessary curves.
- Took help from seniors and also saw youtube videos to make dxf file.
- Attended workshop conducted by electronics club on arduino and Atmega and also googled on net.
- Looked into site arduino.cc.

3.3.6 Links

- http://www.instructables.com/id/Use-your-android-phone-sensors-on-thearduino-/
- http://www.instructables.com/id/Arduino-Powered-Autonomous-Vehicle/
- http://www.instructables.com/id/How-to-use-the-L293D-Motor-Driver-ArduinoTutorial/
- https://mobiforge.com/design-development/sense-and-sensor-bility-accessmobile-device-sensors-with-javascript
- https://lh4.googleusercontent.com/-EgQK-VJRL54/TgHt3Sw1AxI/AAAAAAAAAPQ/Padivqv49Y

3.4 Team Members

- Sandarsh Gupta
- Manas Rawat
- Suraj Verma
- Ravi Prakash Tripathi
- Ayush Singh
- Sudhanshu Bansal
- Satyam Bhartiya
- Rishav Kumar