Location guidance of robots using local positioning system

Rahul Kumar1*, Shubham Sarangi1, Anurag Shukla1, Meghana Gupta Arakere2

1Dept. of Mechanical Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu (603203)
2Dept. of ECE, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu (603203)

*Corresponding author: rs15041678@gmail.com

Abstract. The concept of this project is to achieve precision control of locomotive robots in a vast facility (industrial, universities, and hospitals) by designing a Local Positioning System for that particular facility and defining the coordinates of the robot using that system. This is achieved by setting up cameras at specific points around the facility and by fixing a beacon on the robot. The movement of the beacon is tracked by the cameras and the location is controlled using a control system, the information being passed to the robot via RF waves. This can achieve greater control than one using GPS as the robot is mapped only along a confined area.

1. Introduction

The inertial navigation system is used to guide a wheeled robot inside facilities like warehouses, universities and industrial level buildings in order to perform specific payload task. This device is a land based navigation system using beacons to define the local positioning system, and vision inertial navigation to navigate the robot through the campus. Many of the research papers have been using the approach of uploading an environmental simulations to their device while others used number of captured images to compare them with the real time images being captured. The major problems were to create landmarks and path milestone in a large number with definite positions for navigation.

This problem has been overcome with this project by providing the robot with real time tracking systems using beacons which reduce the time delay and hence give more accurate results.

This paper will talk about the major parts in the project, how and why they are being used.

2. Project

2.1. Navigation System- Components

The navigation systems include the beacons, local GPS and obstacle avoidance systems. [1]

2.1.1. Beacons
A beacon, simply put is a radio transmitter. It’s kind of like a lighthouse: it repeatedly transmits a single signal that other devices can see. Instead of emitting visible light, though, it broadcasts a radio signal that is made up of a combination of letters and numbers transmitted on a regular interval of approximately 1/10th of a second. A device like a smart phone, GPS or Wi-Fi can “see” a beacon once it’s in range, much like sailors looking for a lighthouse to know where they are.

The beacon sends out its ID numbers about ten times every second (sometimes more, sometimes less, depending on its settings). A nearby device, like your phone, GPS or Wi-Fi picks up that signal (RF Transmitter and Receiver is used). When a dedicated component recognizes it, it links it to an action or piece of content stored in the cloud or live time data and displays it to the user. You can “teach” your component how to react to a beacon signal by developing using third-party tools.

The beacons being used here are traditional RF emitters and RF detectors. They are used for the detection of the waves. The robot will be installed with a set of RF beacon and detector. This system will be emitting a set of frequencies of specific bandwidth and hence defining the real time location of robot to the computation unit. [4]

2.1.2. Local GPS

A **local positioning system** (LPS) is a navigation system that provides location information in all weather, anywhere within the coverage of the network, where there is an unobstructed line of sight to three or more signaling beacons of which the exact position on earth is known. A special type of LPS is the real-time locating systems; which also allows real-time tracking of an object or person in a confined area such as a building.

A virtual local GPS is created with the help of a set of RF emitters and detectors. This system consists of 4 RF emitters and detectors which create a real time connection with the robot. The RF waves emitted by the robot are detected and hence computed to locate the position of the beacon and hence show you the real time tracking. [5]

2.1.3. Air Caster

Air caster is kind of pneumatic hovercraft. It is designed to create a thin flow of Omni directional flow of air which makes the body float few millimetres above the ground. It uses a diaphragm (which creates a torus shaped area inside it) which is inflated with the help of a continuous flow of compressed air, this inflation causes a sealed contact with ground which is continuously destroyed by the forces of more air at centre of torus. It creates an easy movement across the ground without any considerate amount of friction. It can hence carry up to 500 kgs of weight in a completely stable state with an ease in movement in any direction. Although considering its limitation to translate in irregular surfaces we consider the ground of working area to be considerably smooth.
2.1.4. NODEMCU (ESP8266)

NodeMCU is a Wi-Fi module we will be using to connect the beacons and form the Local GPS system. NodeMCU has a component called the ESP8266 which is the Wi-Fi module. It connects to the network by either LAN or Global Networks. It works as an individual module just like Arduino. Since it does not have a specific coding platform, we are using Arduino coding platform for NodeMCU. It is connected to the LAN using an application called the BLYNK which can be accessed by any smartphone or computer.

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the C++ or Python scripting language.

2.2. Obstacle Avoidance

The only obstacles that we will be getting in the platform of the robot will be mobile, for instance a human worker or a forklift. For obstacle avoidance at low cost, a ultrasound sensor can be used which when detects a mobile obstacle, will start a feedback loop and hence stop the robots motion until the removal of obstruction. We use the ultrasonic sensor to detect and avoid the obstacles. [6]

The Ultrasonic Sensor sends out a high-frequency sound pulse and then checks the time as to how long it would take for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).
The speed of sound is approximately 341 meters (1100 feet) per second in air. The ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object. It uses the following mathematical equation:

**Distance = Time x Speed of Sound divided by 2**

2.3. **Navigation System - Working**

This system is connected by a bidirectional beacon system. As we have discussed previously, the RF waves emitted by the robot are detected by the RF detectors which are virtually placed on approximate alignment of the robot. Each of the detectors placed on the edges take up the beacon signals and provide the alignment of gyroscope.

When we need to locomote the robot from one place to another, the RF emitter in local GPS which are connected to the beacons will send the data of the next position and by using the method of dead reckoning, the next position of the robot on the basis of previous alignment and position will be determined before the robot actually goes to that position hence providing it with real time tracking and reducing time delay. [2][3]

![Figure 2. Basic Layout](image)

We also use a novel and new vision-aided navigation system which is based on an IMU and a stereo camera. Motion estimation is performed in real time at up to 30 frames per second and the integration of the sensors is robust. By robust, we imply that VO is allowed to fail and be restarted at any moment. This has been implemented because of its important practical advantages:

1. The camera trajectory does not need to be registered within the global coordinate system;
2. In case of failure, the update is set to have infinite uncertainty and the VO is simply restarted;
3. Expensive uncertainty propagation over time is avoided since the uncertainty estimation is only required for the motion update.
2.4. Navigation systems - Coding

We have to connect the beacons and ultrasonic sensor to the Arduino and then connect them to the Wi-Fi network. We will consider the Wi-Fi network to be LAN network which is the most used type of network. [7][8]

The code to connect the beacon to NodeMCU is as follows:

1. Install Arduino and open it.

2. Go to File > Preferences

3. Add http://arduino.esp8266.com/stable/package_esp8266com_index.json to the Additional Boards Manager URLs.

4. Go to Tools > Board > Boards Manager

5. Type in esp8266

6. Select version 2.0.0 and click on Install (must be version 2.0.0!)
Figure 4. Audrino snapshot

7. Go to File > Preferences

8. Open the folder path under More preferences can be edited directly in the file
Figure 5. Audrino snapshot

9. Go to packages > esp8266 > hardware > esp8266 > 2.0.0 > tools > sdk > include

10. Open user_interface.h with a text editor

11. Scroll down and before #endif add following lines:

```c
typedef void (*freedom_outside_cb_t)(uint8 status);
int wifi_register_send_pkt_freedom_cb(freedom_outside_cb_t cb);
void wifi_unregister_send_pkt_freedom_cb(void);
int wifi_send_pkt_freedom(uint8 *buf, int len, bool sys_seq);
```
Figure 6. Audrino snapshot

don't forget to save!

12. Go to the SDK_fix folder of this project
13. Copy ESP8266Wi-Fi.cpp and ESP8266Wi-Fi.h
14. Paste these files here packages > esp8266 > hardware > esp8266 > 2.0.0 > libraries > ESP8266WiFi > src
15. Open esp8266_beacon > esp8266_beacon.ino in Arduino
16. Select your ESP8266 board at Tools > Board and the right port at Tools > Port
17. Depending on your board you may have to adjust the Tools > Board > Flash Frequency and the Tools > Board > Flash Size. I use a 160MHz flash frequency and a 4M (3M SPIFFS) flash size.
18. Upload!

The code to connect the ultrasonic sensor to Nodemcu is:

```c
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```
#define TRIGGERPIN D1

#define ECHOPIN D2

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "Your auth token";

// Your Wi-Fi credentials.

// Set password to "" for open networks.

char ssid[] = "your wifi SSID";

char pass[] = "your password";

WidgetLCD lcd(V1);

void setup()
{

// Debug console

Serial.begin(9600);

pinMode(TRIGGERPIN, OUTPUT);

pinMode(ECHOPIN, INPUT);

Blynk.begin(auth, ssid, pass);

// You can also specify server:

//Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);

//Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8442);

lcd.clear(); //Use it to clear the LCD Widget

lcd.print(0, 0, "Distance in cm"); // use: (position X: 0-15, position Y: 0-1, "Message you want to print")

// Please use timed events when LCD printintg in void loop to avoid sending too many commands
void loop()
{
    lcd.clear();
    lcd.print(0, 0, "Distance in cm"); // use: (position X: 0-15, position Y: 0-1, "Message you want to print")
    long duration, distance;
    digitalWrite(TRIGGERPIN, LOW);
    delayMicroseconds(3);
    digitalWrite(TRIGGERPIN, HIGH);
    delay(12);
    digitalWrite(TRIGGERPIN, LOW);
    duration = pulseIn(ECHOPIN, HIGH);
    distance = (duration/2) / 29.1;
    Serial.print(distance);
    Serial.println("Cm");
    lcd.print(7, 1, distance);
    Blynk.run();
    delay(3500);
}

4. Conclusion

This type of navigation system has a very great advantage in industrial areas and warehouse with high production and low human proximity, as it has a faster feedback response which will help in faster transfer of payloads from one part of industry to other and can be operated from mobile areas.
5. References

[1] Kawabata K, Doi M, Chugo D, Kaetsu H and Asama H, 2007 Vehicle Guidance System using Local Information Assistants, *Distributed Autonomous Robotic Systems*. 6 , 85-94

[2] Suzuki T, Uehara T, Kawabata K, Kurabayashi D, Paromtchik I E and Asama H 2003 Indoor Navigation for Mobile Robot by Using Environment-Embedded Loca Information Management Device and Optical Pointer, *Field and Service Robotics*.12 ,41-49

[3] Golledge R G , Klatzky  R L , Loomis J M, Speigle  J and Tietz  J 1998  A geographical information system for a GPS based personal guidance system, *International Journal of Geographical Information Science*.12 ,727-749

[4] https://kontakt.io/beacon-basics/what-is-a-beacon/

[5] https://en.wikipedia.org/wiki/Local_positioning_system

[6] http://arduino-info.wikispaces.com/Ultrasonic+Distance+Sensor

[7] (source: https://github.com/esp8266/Arduino)

[8] https://www.hackster.io/helloworld1997/ultrasonic-sensor-with-blynk-and-nodemcu-50c074