ParkingBot

University of Florida
Department of Electrical and Computer Engineering
EEL 5666
Intelligent Machines Design Laboratory

Rohan Pais

TAs:
Sara Keen
Adam Barnett
Mike Pridgen

Instructors:
Dr. A. Antonio Arroyo
Dr. Eric M. Schwartz

Table of Contents

Abstract

The ***ParkingBot*** is a robotic vehicle. The ***ParkingBot*** has four wheels. Steering is done at the front wheels using an Ackerman steering principle arrangement. A servo motor is used to control the steering. Power for moving either forward or backward will be sent to the rear wheels via a differential. A hacked servo motor is used for this.
When the ***ParkingBot*** is left at a parking lot it will move to an empty parking space and park itself. When the robot receives the call signal it maneuvers itself out of the parking space and goes to the parking lot exit.

Introduction:

The ParkingBot will use its four wheel platform to move around in a model parking lot which is made for the purpose of the demo. There is an overhead camera which monitors the parking lot and finds an empty parking space. It then sends the address of this parking space to the ParkingBot mobile platform. On receiving this address the ParkinBot then moves to the empty parking space and parks itself.

Platform:

Mobile Platform

The mobile platform is called the ParkingBot. It has four wheels. The main components are the steering assembly, the drivetrain, onboard electronics.

The steering is an Ackerman Principle steering. The Ackerman concept is to have all four wheels rolling around a common point during a turn. This is done to prevent slip in the front wheels. Using this geometry both wheels don't turn through the same angle. The steering linkages are actuated by a servo motor.

The drive train consists of a hacked servo motor which powers the car. This power is sent to the back wheels via a differential. The differential is required to facilitate easy turning. As the ParkingBot need to be able to maneuver itself into a tight empty parking space a differential is required.

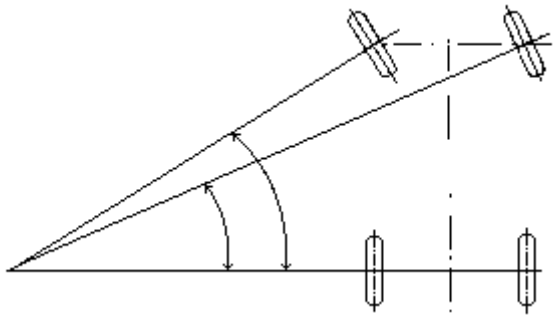
The onboard electronics consists of the Maveric IIb board. It has the Atmel 128 microcontroller on it. This board is used to get information from the onboard sensors, control the actuators(motors) and communicate with a pc at the parking lot.

Components On The Mobile Platform

Mechanical Devices

Steering Assembly

The steering is an Ackerman Principle steering. The Ackerman concept is to have all four wheels rolling around a common point during a turn. This is done to prevent slip in the front wheels. Using this geometry both wheels don't turn through the same angle.



The steering linkages are actuated by a servo motor.

Differential

The differential has three jobs:

- To aim the engine power at the wheels
- To act as the final gear reduction in the vehicle, slowing the rotational speed of the transmission one final time before it hits the wheels
- To transmit the power to the wheels while allowing them to rotate at different speeds.

The differential is required to facilitate easy turning by allowing each of the rear wheels to turn at different speeds. As the ParkingBot to be able to maneuver itself into a tight empty parking space a differential is required.

Electronics:

The most important electronic component is the Maveric IIb board. It is an ATMEL 128 microcontroller board made by BD micro. This board is designed with functionality in mind and should be usable for most robotics projects.

Bluetooth Module

In addition I have also used a Bluetooth module to establish communication between the Maveric board and a laptop. The idea was to try and make full use of a computers processing capabilities and to take advantage of MATLAB's functionality for image processing. Hence communication was necessary. For this purpose I have used Sparkfuns Bluetooth DIP Module WRL-0846. This device receives data sent from my pc.

Parking Lot

For the demo it is required that I have a model parking lot in addition to the ***ParkingBot***. This parking lot will have six parking spaces. These parking spaces will be monitored by overhead camera. It detects an empty parking space and then sends this information to the ***ParkingBot***. Image processing is used to find the empty parking space. The address of this empty parking space is sent to the ParkingBot.

Actuation

The robot has one hacked servo motor and one unhacked servo. The servo is used to actuate the front steering mechanism and the dc motor is used to move the robot either forward or backward.

For steering the front wheels I have used a micro servo. This is the GSW micro servo. This is a very tiny servo and produces a small torque of 20 ounce-inches. However this is more than sufficient to actuate the steering mechanism.

The hacked servo is a 70 ounce-inch torque producing servo. It was hacked for continuous rotation. Basically hacking makes a servo motor into a geared dc motor. The advantage of this is that the existing servo motor circuitry can be used to drive this motor and hence there is no requirement for a motor driver circuit board to be made.

Sensors

Sonar

The sensors onboard the ParkingBot are for collision avoidance. I have used sonars for this. The principle is very simple, when the sonar detects an object closer than five inches from the front of the vehicle it triggers an interrupt which stops all other operations and stops the vehicle where it is. The sonar module I used is the Maxbotix EZ1. This is a great module which has three operating modes. I have used the analog mode of operation. The only drawback with this device is that it can't detect anything closer than six inches from it. For this reason I had to mount the sonar further back on the platform.

Webcam:

This is the most important sensor used for my robot. It is literally the eyes of the robot. The webcam is used to detect an empty parking space in the parking lot. The image from the webcam is sent to MATLAB for image processing. An empty space is detected and the ParkingBot is made to go to this space.

Working and Navigation

Detection Of An Empty Parking Space

The overhead webcam views the entire parking lot. The image of the parking lot is taken and cropped at each parking space. Then to each of these images a filter is applied. This filter gives an output image which is either black or white depending on whether the space is occupied. This filter is made based on color detection. It is an almost failsafe method of detecting if the space is occupied or not. The only condition where this method will fail is if the vehicle parked in it is exactly the same color as the parking lot floor (there is a very slim chance of this happening).

Navigation

Path

Based on predefined waypoints there is a path obtained for each of the parking spaces. These six paths are stored in a multidimensional array (20 x 2 x 6). Then based on the parking spot which is free, the corresponding path is pulled up and sent to the traking and navigation algorithm

Navigation

The car moves in the parking lot using vision based navigation. There is a bright LED in the front of the vehicle. This bright intensity spot is detected in the image coordinate system. Hence the coordinates of the front of the vehicle is known. These coordinates are then compared with the path coordinates. Then using rather COMPLEX vector analysis the vehicle is sent a signal to move in the desired direction. Below is the algorithm for the entire vision based navigation.

1. Create a video adaptor for MATLAB to recognize the webcamera.

2. Start the video input object. This stores all the frames acquired into the memory buffer.

3. Run the function in which the user defines the waypoints. For this I have used the ginput (graphical user input) function to define the waypoints. This is necessary because the camera is not fixed and hence the waypoints cannot be hard coded.

4. Analyze an image frame to detect an empty parking space.

5. Run the tracking function and navigation function.

6. Send data. This sends data to the Bluetooth module. Here, Matlab writes data to the Bluetooth serial com port. The DIP module takes care of the sending the data to the board.

Conclusion

There are many advantages to vision based navigation and hence I see large scope in this project. However, mono vision (single camera based) has a drawback due to the fact that depth perception is not possible with this method. A possible add on to this is to use stereo vision for navigation.