The University of Washington, Computing and Software Systems - Technical Report

Shailja Tyler Folsom
{shailja, tfolsom}@uw.edu

University of Washington
Bothell, WA, 98011

2016

Abstract

The Elcano Project provides a blueprint for building your own experimental automated vehicle using electronics and sensors costing under $1000. A tricycle with an electric helper motor under 750 Watt and top speed under 20 mph is legally a bicycle, and thus street-legal without license, registration or insurance. The long term vision of the project is not a single self-driving vehicle, but a collection of such vehicles that communicate with each other. Automatically such vehicles can take instructions from a roadside traffic management computer that manages a section of roadways. These computers would link together to form a distributed, scalable Traffic Management System; such a system can reduce congestion. In this report we discuss the technology behind the localization of trike. We estimate the position of the vehicle from the previous known position using data from different sensors. We have used the compass, cyclometer, GPS and an optical mouse sensor to develop an accurate localization module for the tricycle. Using the data from the camera mounted on the vehicle, we have shown the development of a lane detection algorithm for this tricycle which forms an integral part of the navigation module for the trike.

1 Introduction to Elcano Project

The Elcano Project is developing low-cost hardware and software kits to convert any vehicle to self-drive. We are concentrating on recumbent tricycles, since that produces a real people mover at a total cost of under $1000. Our kits could also be used in full-sized cars or toy RC cars.

The kits are based on Arduino microcontrollers, and allow you to connect them robustly into a compact, low power package. The open source software is written in C++ and can run on many other platforms. The basic package uses several processors. The open source software is available on Github.

2 Navigation and Localization

2.1 Position Estimate using Dead Reckoning

Dead reckoning is implemented in the three-wheeled bicycle in order to overcome the limitations of GPS technology alone. Satellite microwave signals are unavailable in parking garages and tunnels, and often severely degraded in urban canyons and near trees due to blocked lines of sight to the satellites or multipath propagation. In a dead-reckoning navigation system, the trike is equipped with sensors that know the wheel circumference and record wheel rotations and steering direction. This can be read by the navigation system from the controller-area network bus. We have developed a navigation system which uses a fuzzy algorithm to integrate the always-available sensor data with the accurate but occasionally unavailable position information from the satellite data into a combined position fix.

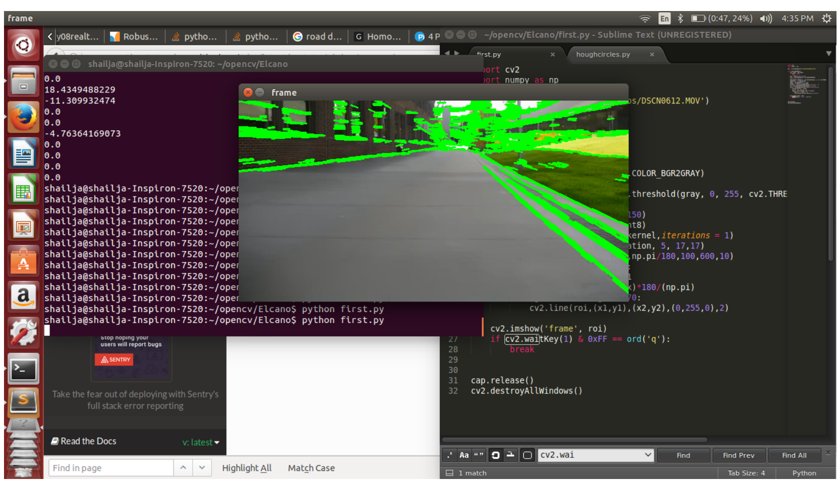
3 Innovative Optical Mouse Based Sensor

Using the optical Mouse ADNS 3080, we developed a sensor to calculate the position of the robot using 2-D displacement vectors. The mouse sensor returns the average movement (in the x and y directions) of surface features that it sees. It does so by identifying texture in two subsequent frames and calculating the distance that the images have been displaced. With proper calibration, the optical mouse based position sensor proved to be a valuable side information in the localization module helping the robot navigate the planned path much more accurately.

4 Conclusions

The navigation and localization module was tested across the UW campus. Overall, we used compass, cyclometer, GPS, optical mouse sensor and camera to estimate the position of the trike. Each sensor had its own limitations which were countered by other sensors. We started with speedometer and compass, using the bearing angle and speed reading we measured the final position after a given time interval. Some of the drawbacks of solely depending on these sensors were that the speeding around curves that require slowing down, rough or loose terrain that could negatively affect the trike control and rainy or wet weather could lose traction. Also, it does not take the direction of the vehicle into consideration. To overcome this, we used the Global Positioning Sensor which gives the latitude and longitude of the trike at a given time. GPS is extremely easy to navigate as the direction of motion can be revealed using it. Sometimes the GPS signals are not accurate due to some obstacles to the signals such as buildings, trees and sometimes by extreme atmospheric conditions such as geomagnetic storms. Seeing the pros and cones of above two methods, we came up with the idea of position estimation using sensor data fusion. We tested out fuzzy number algorithm to estimate the current position of the trike. The results were promising with only 0.04% error.

To eliminate the intersection of the drawbacks of above method, we devised the optical mouse sensor which was used to calculate the position of the robot using 2-D displacement vectors. This sensor regularly updates the information whether the trike is moving or stationary. We also tried to implement lane vision to make our system robust. The raspberry pi camera was mounted on the robot which can be used for various purposes like obstacle avoidance, assistance while turning etc. The lane detection method was implemented to know the position of the robot from right side of the lane. Integrating all the information from various subsystems, we successfully estimated the position of the robot. We could use the lane detection algorithm along with machine learning approach to ensure that the trike is always on the lane. Thus, we represent the localization model using different sensors contributing to the trike information and at the same time countering the drawbacks of each other. One of the results of the lane detection algorithm for the tricycle is shown below



5 Reference

1. Shailja, Tyler Folsom, "Elcano : Localization and Navigation of an Autonomous Tricycle" In: *Technical Report - University of Washington, Computing and Software Systems 2016*