Design of Robotics Vehicle with Autonomous Navigation and Obstacle Avoidance

DEPARTMENT OF
Engineering

TCU COLLEGE OF SCIENCE & ENGINEERING

Authors: Bao Thach, Chris Prasai, Sam Adams

Faculty Advisor: Professor Morgan Kiani (Engineering Department)

SciCom Let's Talk Science

In our project, a control-theory based algorithm would be employed to develop a small electric vehicle that can self-navigate through an unknown course to arrive at the desired location while avoiding obstacles and walls. This project is an extension of our successful project funded last year, in which we were able to operate a partially autonomous car to run around a location and generate a virtual map. Our team expects to grant the car full autonomy like a self-driving car and let it travel through a relative abundance of places to create computer models of critical infrastructures without the help of humans. The success of this project will have a broad impact on society. First, this capability would be useful in self-driving cars, which allow drivers to spend their time more productively instead of driving to work or assist disabled people. Second, the car can generate a simulated model of places that help to analyze unknown locations. Finally, the project can surely create a platform for future TCU engineering students to learn about self-driving car technology and machine learning. This project is expected to succeed due to the achievements we gained from the previous project.

The algorithm will be written in Python/ROS, controlled by Raspberry Pi 3, and tested on a walled course constructed by us. It should be able to navigate a course without having already driven through it. Another special feature is that the car will also precisely arrive at a pre-determined location.

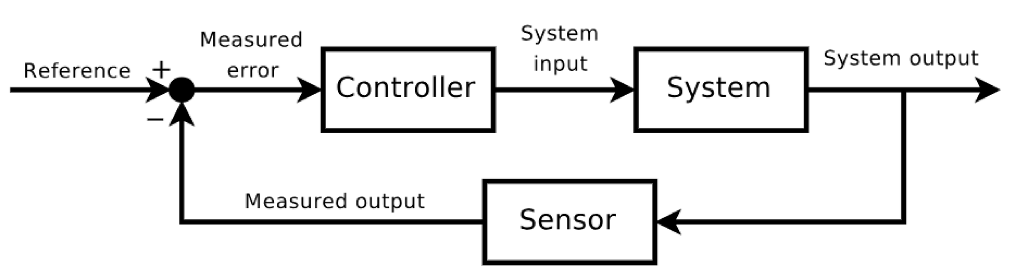
Objectives

Motivation: To design, manufacture, and program a small robotic vehicle to perform autonomous tasks

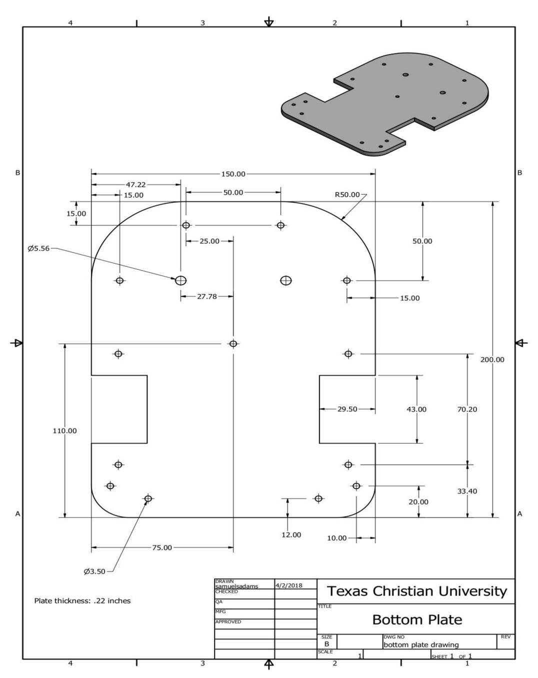
Goal: Develop and implement control theory based algorithms to enable the car to reach a desired location while simultaneously avoiding obstacles.

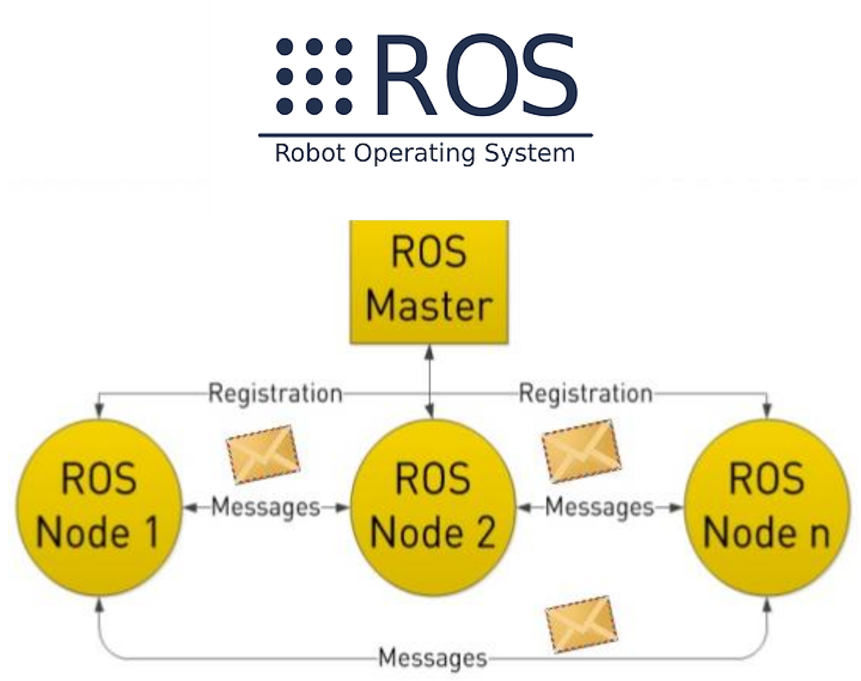
Background

- ***Control theory:*** a closed-loop model used to control the behavior of an object in real time. Control theory is an attempt to assess and respond to changes in inputs to produce desired outputs.

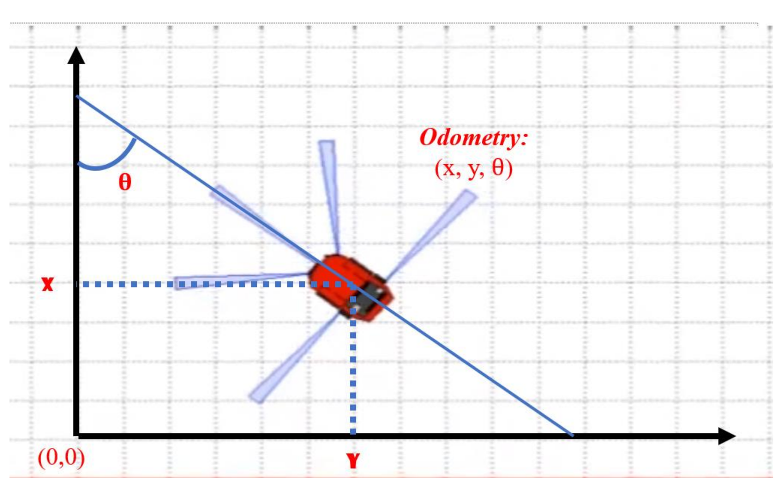


- ***ROS:*** is a software framework allowing us to write applications which operate robotic hardware. ROS provides useful services such as hardware abstraction, implementation of commonly used functionality, message-passing between processes, and package management. It is one of most common frameworks among the robotics community.
- ***Design of robot housing:*** The robot was designed through the use of AutoCAD's Inventor software to create dimensioned drawings to be manufactured through the help of laser cutting, 3D-printers, and TCU's machine shop





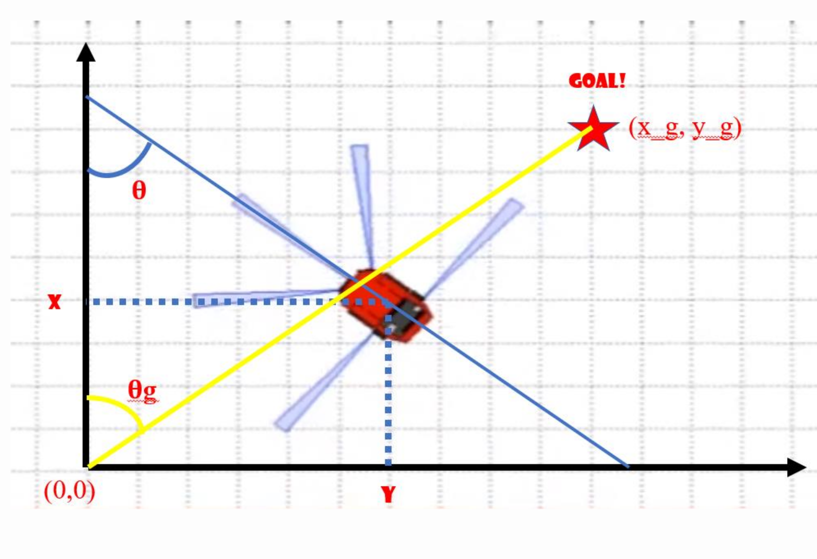
- ***Odometry:*** is the use of data from motion sensors to create an imaginary spatial grid. Odometry estimates an object's location over time with three components: vertical distance (x), horizontal distance (y), and heading angle (θ)

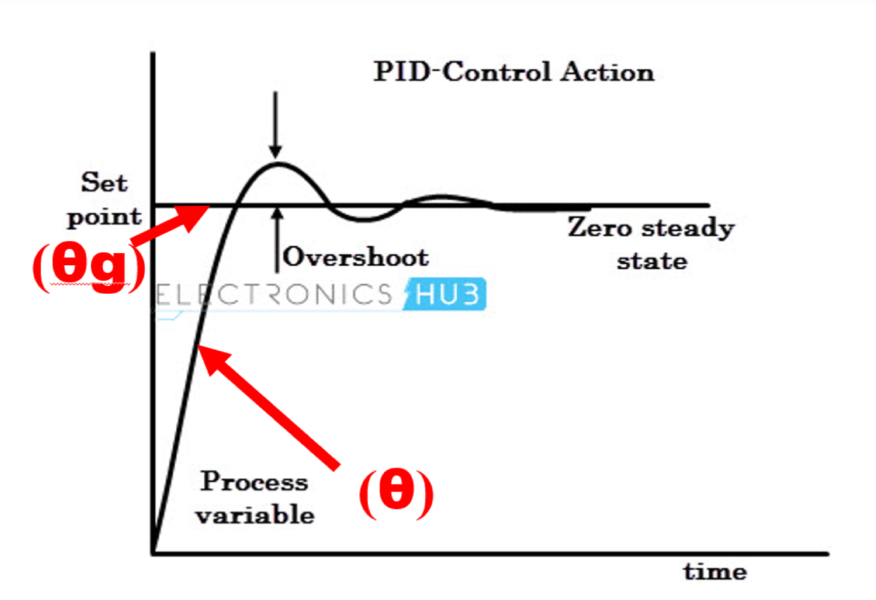


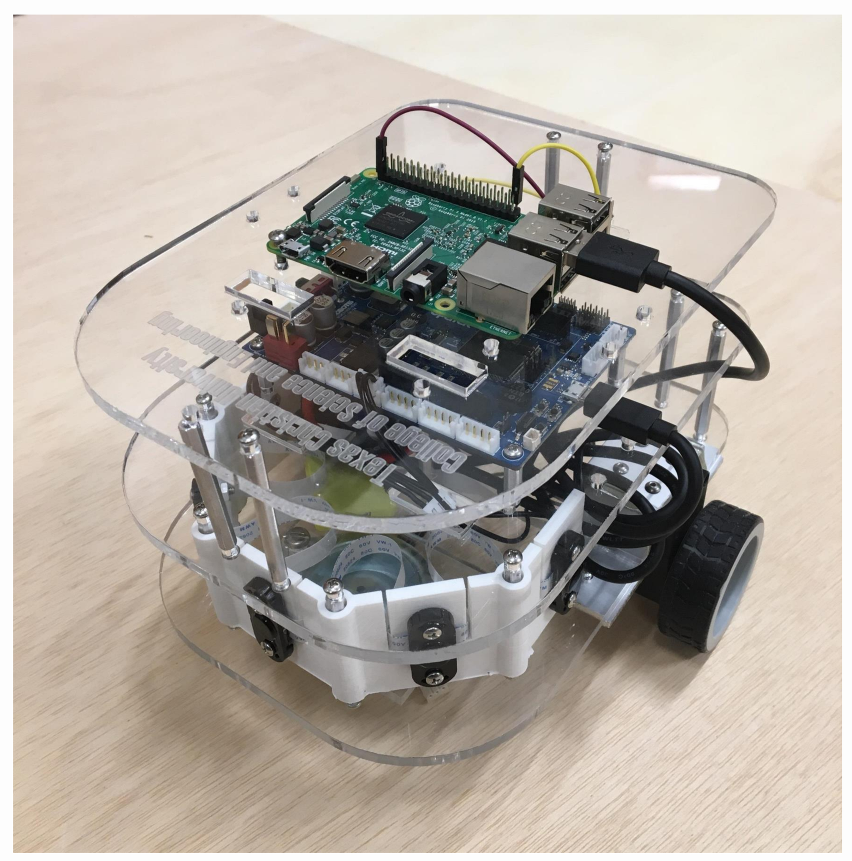
Navigation Algorithms

❖ Go-to-goal Algorithm:

1. Calculates the heading (angle), θg, to the goal location (xg, yg)
2. Calculates the error between θg and the current heading of the robot, θ
3. Feeds the error to the PID controller. PID determines the angular velocity the car should run at
4. θ gets closer to θg over time. Stop the car when its (x,y) coordinate matches with (xg, yg)

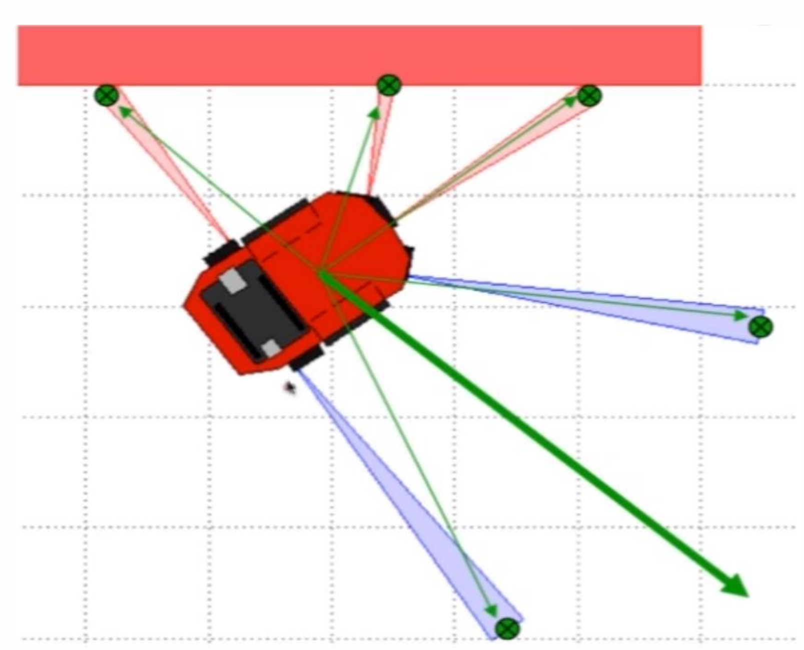


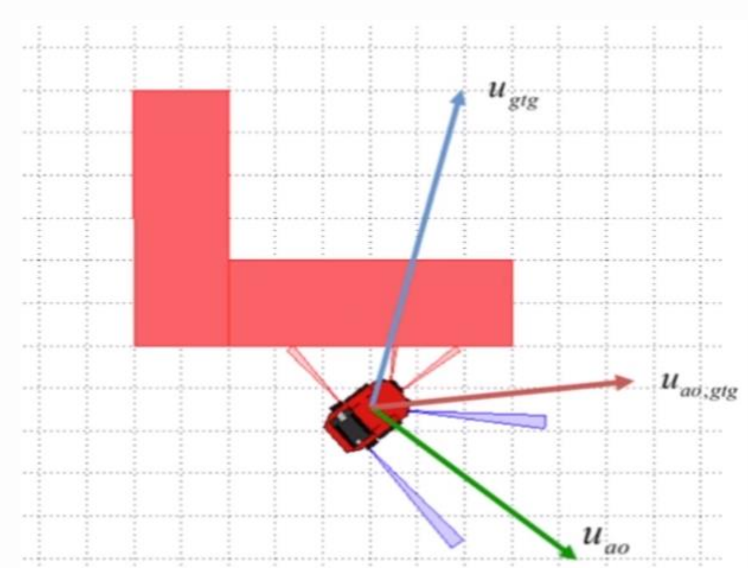




❖ Obstacle Avoidance Algorithm:

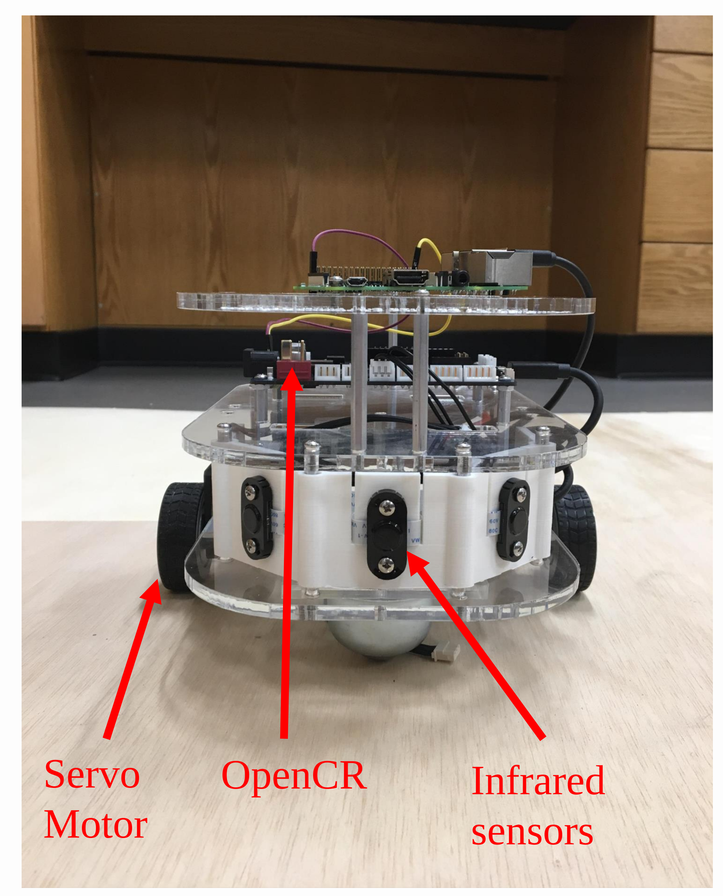
1. Transforms infrared sensor measurements to vectors relative to the rotational center of the car
2. Translates the relative vectors to the global coordinate
3. Sum of these vectors results in a new direction vector which points away from detected obstacles.
4. The resultant direction vector defines the new heading angle which is used by the go-to-goal algorithm to steer the car around the obstacle

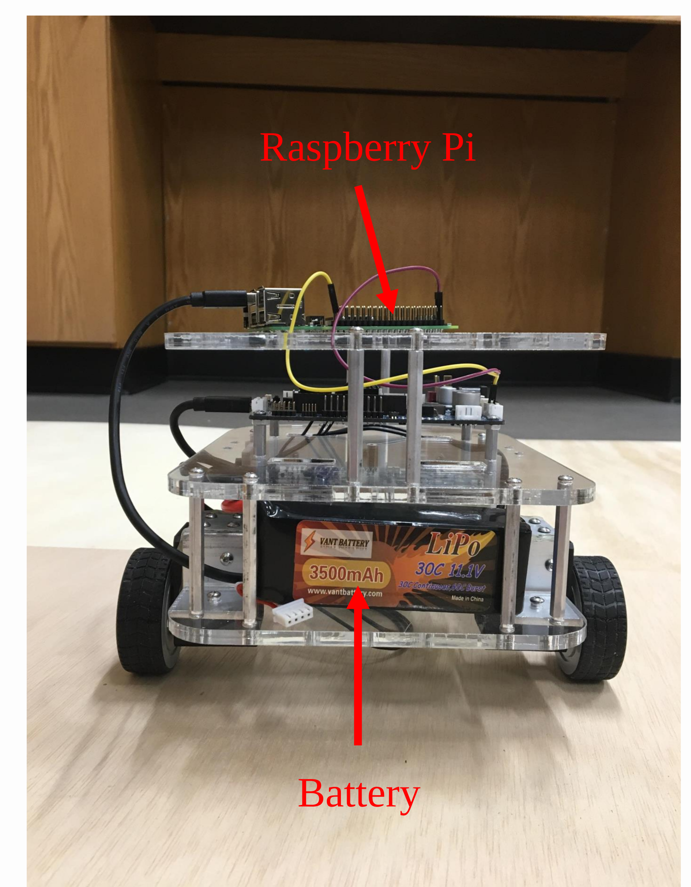




How it works

- ***Lithium battery:*** a light weight, low cost, high power density type of battery. It gives power to the computer and the controller board.
- ***Raspberry Pi:*** a low cost, credit-card sized computer. The Pi is essentially the brain of our robot where all sensor data is sent to and computation occurs.
- ***OpenCR:*** the main controller board which receives commands from the Raspberry Pi to control all motors.
- ***Dynamixel Servo Motor:*** Actuators with 360 degree control allows it to operate as the vehicle's wheels. These "smart" actuators are able to perform specific command velocities dictated by the operating system's controller node.
- ***Infrared Sensors:*** Infrared sensors are positioned on the car to detect obstacles within 2 meters' distance of the car. The sensors update their distance readings rapidly and relay this information via a ROS node to the central processing nodes for the vehicle to alter its route accordingly.





Conclusions/Results

- A robotic vehicle was designed and programmed to perform the autonomous tasks of traveling to a desired coordinate location while also avoiding obstacles.
- Through the use of the Robotic Operating Software (ROS), a network of communicating nodes was established to use sensor input and motor feedback to allow control theory based algorithms to direct the robot's movements and avoid obstacles.

Acknowledgements

The team is thankful to Dr. Kiani for her support and mentorship in developing this project, and for her dedication to ensuring its success. We are also thankful to Ms. Tammy Pfrang for her assistance with logistics and resolving technical issues. The machine shop also provided a us a great support in manufacturing our robot. Additionally, we are grateful to the TCU College of Science and Engineering and SciCom for their dedication to undergraduate research opportunities.