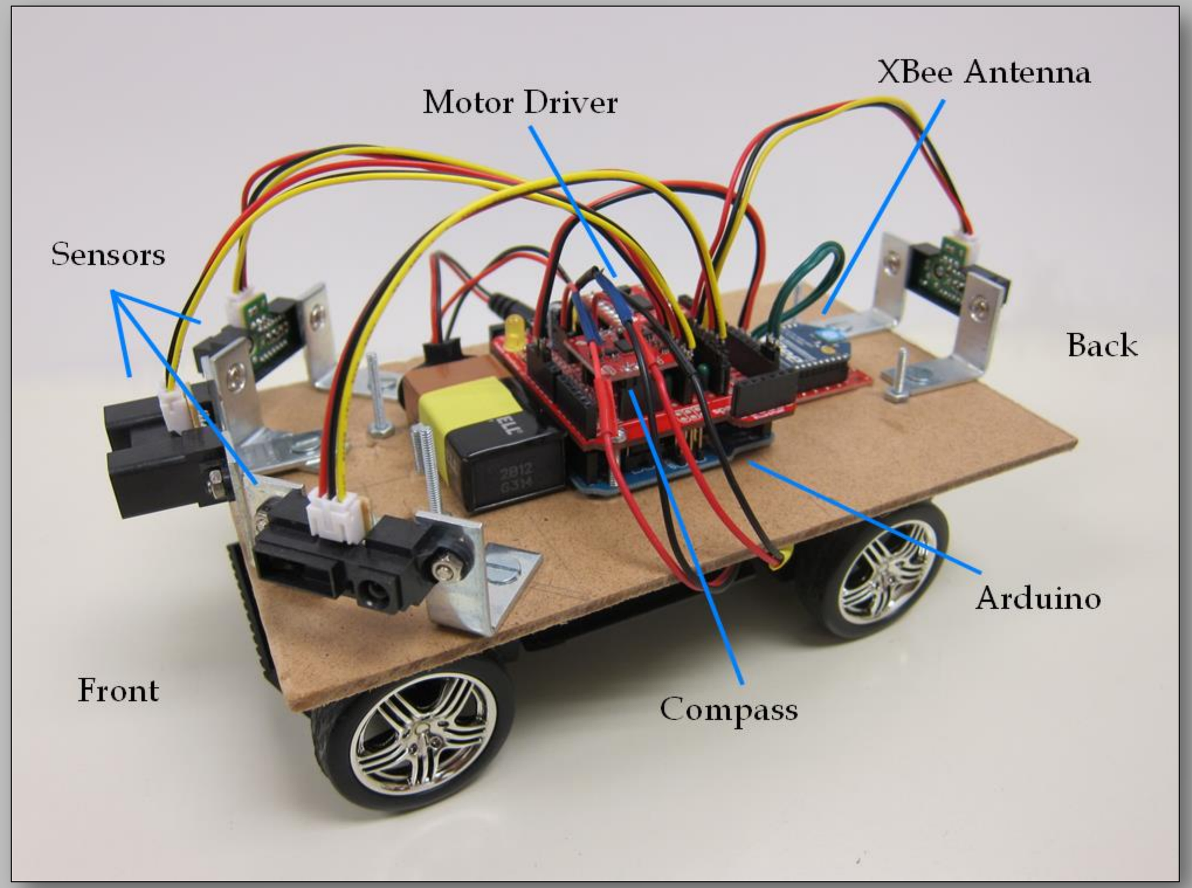
Concurrent Autonomous Vehicle Navigation

by Alex Brooks, Monmouth College; Kendall Lewis, Humboldt State University; Scott Thede, Faculty Advisor, DePauw University

Problem

- *Autonomous* vehicles are driverless or self-driving vehicles.[1] Many systems exist, mainly for driving on streets (highways or local).[2][3]
- This project's goal: come up with a model for multiple vehicles navigating and mapping a specified, flat region without the use of roads or markings
- The region is split up for vehicles to explore separate sections
- The region contains static obstacles, including free-standing objects and walls, that vehicles must detect and avoid
- The result of navigating the region is a coordinate-pair map or image-map indicating where the vehicles travelled and where obstacles exist
- The model is useful for classroom visualization of aspects such as Computer Vision, Robotics, Multi-Agent Systems, and Parallelism/Concurrency.
- The model is also useful for real-world applications such as exploring abandoned or dangerous buildings



Hardware Simulation

- Each vehicle made up of the following components:
  - RC car
  - Arduino microcontroller
  - Dual motor driver
  - Tilt-Compensated Compass
  - XBee Wireless antenna
  - Infrared Proximity Sensors
- Arduino microcontrollers are inexpensive and easy to use
  - The code is stored on each vehicle, allowing it to drive itself
  - Written using the Arduino programming language and environment
- The RC car contains two motors; one responsible for turning, the other for driving
- Both motors controlled by Dual Motor Driver in order to navigate through region
- A Tilt-Compensated Compass determines vehicle's heading relative to Magnetic North
- Each vehicle has a XBee Wireless antenna for communicating with master computer
- Master computer contains a XBee Wireless antenna for signaling a vehicle to stop

Vision

- Computer vision can be simulated through:
  - still-images
  - video
  - ultra-sonic sensors
  - infrared sensors
  - lasers
  - or other options
- Sensors are placed in the locations as seen in Figure 1.
  - Provides an acceptable range of vision
  - Allows for detecting obstacles and preventing collisions
- Collision prevention is made possible by primary function of sensor:
  - Find the distance between the sensor and an obstacle.

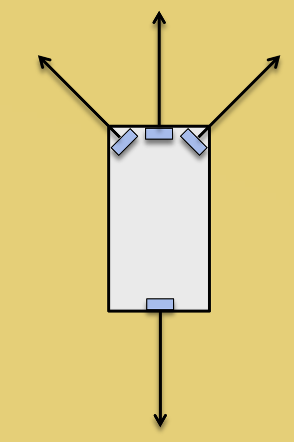
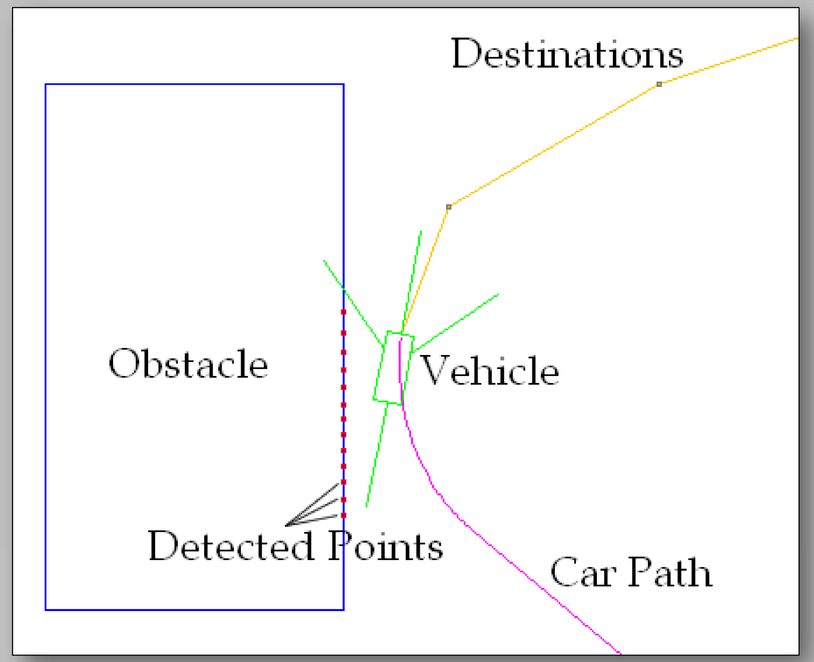


Figure 1

Communication

- Master-slave model used, where central computer (master) has control over all vehicles (slaves)
- A set of destinations is sent to each vehicle prior to navigation
- Communication not explored extensively, but two approaches considered:
  1) Once the set of destinations for a vehicle is empty, send the gathered information about travelled region to master computer
  2) Once a destination is reached, send gathered information to master computer; once information is sent, delete from memory and proceed to next destination



Software Simulation

- Simulation written in Java utilizing:
  - Threads library for concurrency in model
  - Java-Swing library for handling Graphical User Interface (GUI)
- Each vehicle consists of 3 threads:
  - One to calculate positions and visible obstacles
  - One to send vehicle information to the overall model
  - One to update the GUI
- A file containing all properties of region and vehicle is loaded
- Destination points are generated and sent to vehicle
- Vehicles travel their respective paths until simulation is complete

Navigation

- Vehicles are given sets of destination points in order to navigate through a region
- Each vehicle performs a set of instructions to navigate around obstacles if the shortest path is obstructed (See Figure 2)
- The destination is removed if unreachable, and the next one in the set becomes the new destination
- The navigation is complete once the set is empty

Figure 2

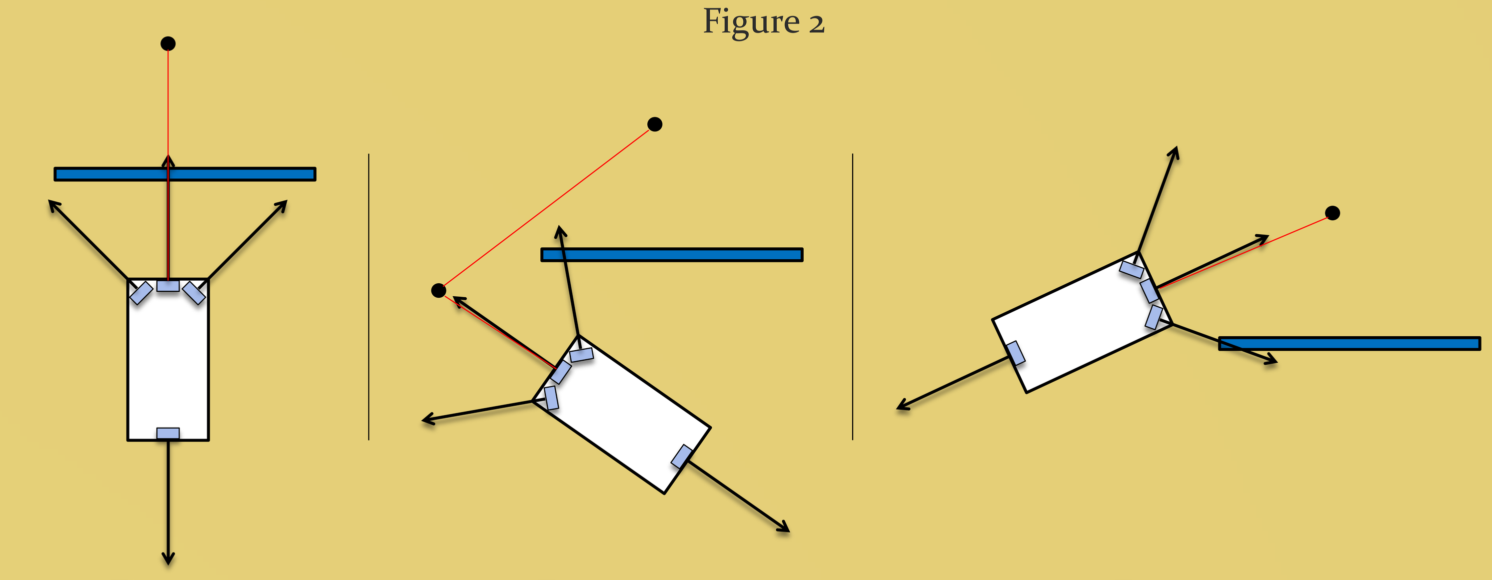


Figure 2.1 Figure 2.2 Figure 2.3

Future Work

Overall:

- Integration of the two simulations to provide better understanding of the overall model
- Simulate over different terrains, incorporate moving objects, and use different types of vehicles such as helicopters or marine vehicles
- Add possibility for simulating in 3-dimensions using video or still-image cameras
  - Will improve understanding of reactions of each vehicle to its surroundings

Software:

- Determine which visible points are a part of the same object and remove redundant/unneeded points
- A better system is needed for allowing multiple vehicles to start in any position of the region

Hardware:

- Incorporate a method for determining the position of a vehicle; two approaches considered:
  1. Determine the speed of a vehicle and use its heading to determine position
  2. Integrate GPS for a more absolute position
- Once positioning is incorporated, communicate positions and visible points to master computer

DEPAUW UNIVERSITY

Acknowledgements

- We would like to thank the National Science Foundation for funding the REU Program, under grant CNS-1156893.
- We would also like to thank DePauw University for hosting our project and providing us with the necessary resources to conduct our research.

References

1. Coppin, B. *Artificial Intelligence Illuminated*. Jones and Bartlett. 2004. pp.545.
2. Kuan, D., Phipps, G., and Hsueh, A. Autonomous land vehicle road following. Proc. 1st International Conference on Computer Vision (London). June 1987. pp. 557-566.
3. Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach* (2e). Prentice Hall. 2003. pp.27.