Comparing Navigation Methods with Boe-Botics

Alexander Davies, Jessy Gardner, Ethan Hurley, Brad Marion, Conor Mauro, Gianna Muto, Alexis Steelman, and Greta Waitz

Appalachian Regional Commission/Oak Ridge National Laboratory Math-Science-Technology Institute 2020

Introduction

Navigation is an essential part of every robot's movement. Each robot has a different job and navigates in unique ways to meet its goals. Through countless trials, errors, and several navigation techniques, routing is not the same for any robot. At times, a robot may find itself in a changing environment or a set environment. These implications require programming the Boe-Bot to be able to adapt and complete these tasks with accuracy. Their full potential is only a few lines away. This can all be solved by an investigation of how they are used. This poster will discuss the pros and cons of four types of navigation and their probable solutions.

Terminology

- **Dead reckoning**: A way of navigation that works by coding an exact path using exact measurements into the code without allowing it to adapt on its own
- **Tactical Navigation**: A form of programmed movement that uses direct contact using prongs or whiskers to detect surfaces it comes into contact with, then it will use pre-programmed directions on how it will react
- **Whiskers**: A metal wire that gets easily pushed back into a pin head to send a pulse to the robot to move and correct direction
- **Infrared Sensors**: A sensor that, when programmed correctly and placed on the front of the robot, can prevent a robot from colliding with another object
- **PhotoTransistor**: A sensor that works by sending a signal or pulse whenever it detects a certain amount of light
- **Computer Coding:** A system using programming to influence a robot's decisions; coding can be used to turn lights on/off, movement, noise, and more
- **Resistors:** Electronic components that will reduce the amount of electricity to an object
- **Multimeter**: A tool that is used to measure the voltage, current, ohms, and check various electronic components

Background

Boe-Bot

The robot that was used for this project was the Boe-Bot, an intermediate driving robot, or simply a robot with wheels. Building and wiring the robot was a learning experience with many trials and errors.

Coding

Any robot can do anything with the right amount of code, but we kept it simple in the world of robotics. First, the robots were programmed to play a reaction game by using LEDs. Then, a traffic light was simulated. The robots were not all just lights though, they also had wheels, so they were able to make simple shapes and even auto-correct themselves using some navigation methods.

Materials

Materials used:

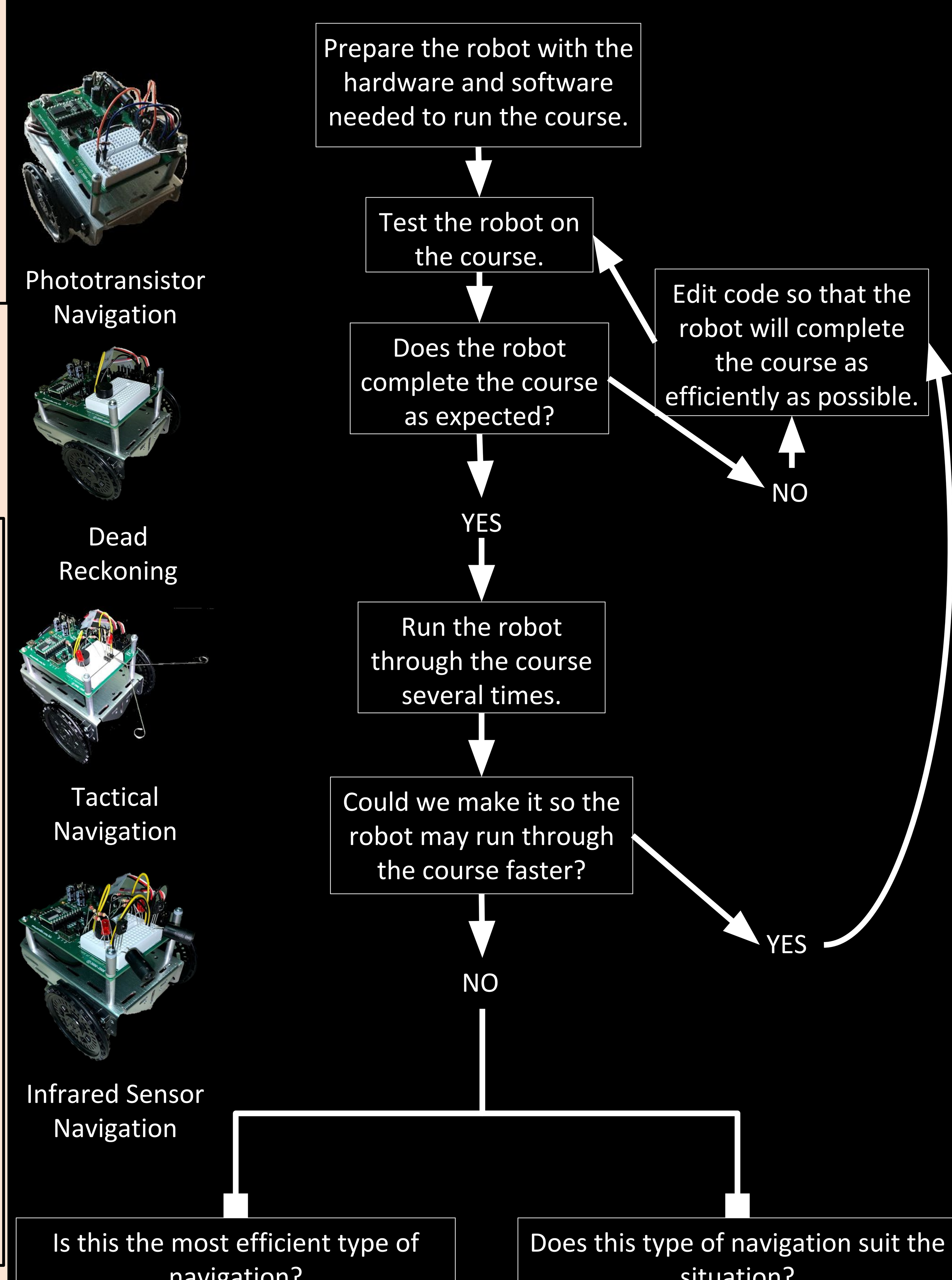
- Parallax Boe-Bot Robot Kit USB
- 220,470,1k,2k,2.2k,4.7k, and 10k ohm resistors
- Blue, red, green, yellow, and white LEDs
- Phototransistor, infrared LEDs, and switches
- Sensors, whiskers, tape, generic obstacles
- Etekcity MSR-A600 multimeter, ruler, and a protractor.

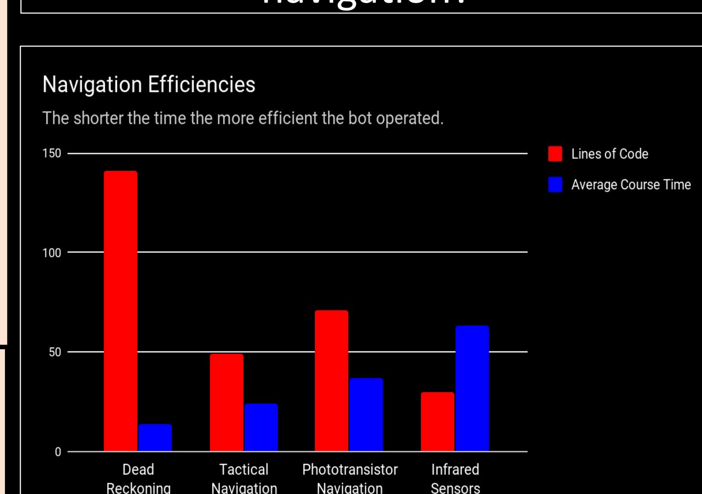
Paths used:

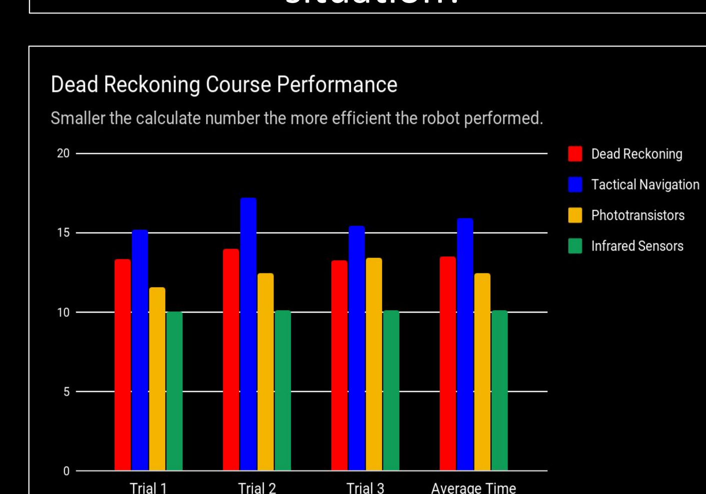
-Circling Technique
-Turns based on angles
-Equilaterals
-Polygons

Methods

Dead Reckoning, Tactical Navigation, Infrared Sensors, and Phototransistor Navigation







Results

- Dead Reckoning:
  - The average course time was 13.503 seconds.
  - There were 143 lines of code in the program.
  - The ratio of course time to lines of code was 0.094.
- Tactical Navigation:
  - The average course time was 15.916 seconds.
  - There were 40 lines of code for Tactical Navigation.
  - The ratio of course time to lines of code was 0.398.
- Infrared Sensor Navigation:
  - The average course time was 10.090 seconds.
  - There were 65 lines of code for Infrared Sensor Navigation.
  - The ratio of course time to lines of code was 0.155.
- Phototransistors
  - The average course time was 12.453 seconds.
  - There were 65 lines of code for phototransistor Navigation.
  - The ratio of course time to lines of code was 0.192.

Conclusions

To conclude the comparison of navigation methods with the Boe Bot, it is evident that there are several ways for a robot to get around. The software and hardware setup of each method is critical to the completion of a robot's task. By understanding the details of robotics, one could decide on which method of navigation to use in order to reach a goal. Trial and error is a major aspect of navigation in robotics, so it is important to find the best technique, or method to use under the correct circumstances.

Method Conclusions:

Upon analyzing the results, we discovered that:

- Dead Reckoning was an efficient method, but had over 100 lines of code. It is best used on set courses where it does not need to make decisions as it navigates.
- Tactical navigation (with the whiskers) is not as fast as Dead Reckoning, but requires fewer lines of code. It makes a few decisions as it navigates, but not as accurate as other methods.
- Infrared Sensor navigation was the most accurate. The number of lines of code is in the middle of the Dead Reckoning and Tactical navigation numbers. It makes decisions based on what it senses, not touches. It would be best used on a course that requires quick changes and adaptability to complete.
- Phototransistor navigation was faster than Dead Reckoning. It does not take as many lines of code, but it is less accurate as any light source can influence its navigation.

Part of a code used for basic navigation:

```
' {$STAMP BS2}
' {$PBASIC 2.5}

'Constants
Left       CON  13
Right      CON  12
LeftLED    CON  11
RightLED   CON  6
R_right    CON  21
R_left     CON  22
P_right    CON  21
P_left     CON  21
V_right    CON  21
V_left     CON  21

'Variables
Counter    VAR   Word
time       VAR   Word

'Main
  DEBUG "program start", CR

'Foward
time = 100
'GOSUB Forward

PAUSE 500

'Backward
```

Acknowledgements

A HUGE thank you to Curtis Holmes, Andy Rayfield, Stacey Murray, and Chris Nelson. We appreciate the opportunity provided to us by Oak Ridge National Laboratory, Oak Ridge Associated Universities, and the Appalachian Regional Commission.