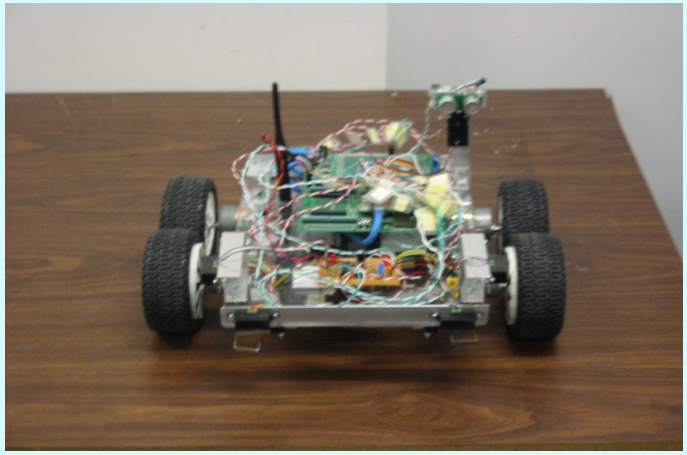
Collaborative Mobile Robot Surveyors

- Sterling Stuart Stein
- Case Cantrell
- Aaron Collver
- Michael Reed
- Chris Meyers
- Sheryl Lau
- Chance Yohman
- William Roberts
- Garret Stephenson
- Robert Malas

Odometry and Motion Control

- Examine the need for error correction
- Develop means for measuring the distance the robot has traveled
- Create algorithm to dynamically adjust the signals sent to the motors to compensate for deviations



Proportional

Integral

Differential

Algorithm (PID)

Proportional

- To respond quicker to deviations.
- Increases overshoot, decrease steady-state error.

Integral

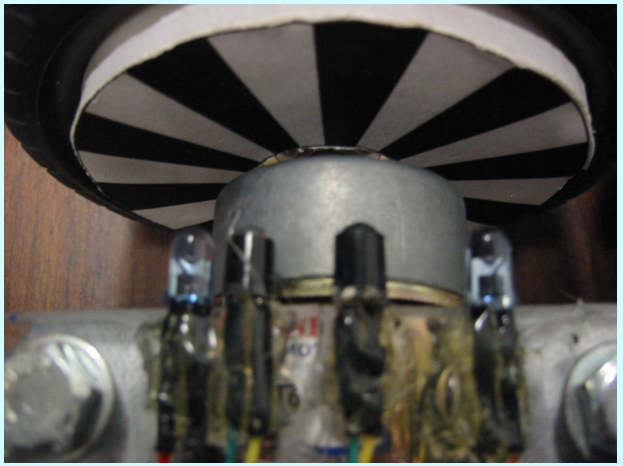
- Decreases rise time like Proportional Control
- Increase overshoot
- Eliminates steady-state error

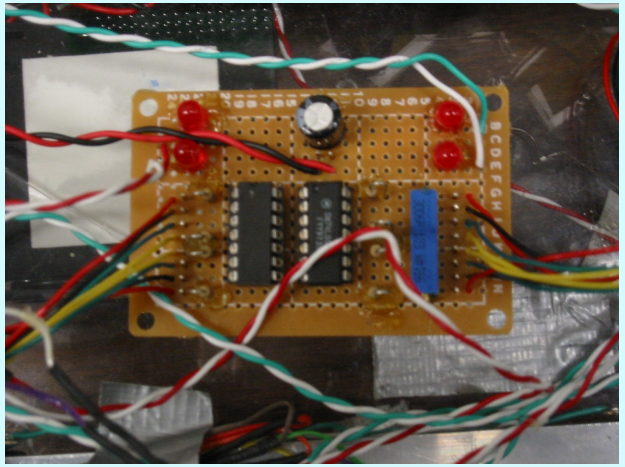
Differential

- Decreases overshoot
- Introduces a small steady-state error

Optical Encoders

- How fast the robot is going is determined by reflected pulses
- The distance traveled can be determined from number of rotations detected
- Direction can be detected based upon which detector is activated first



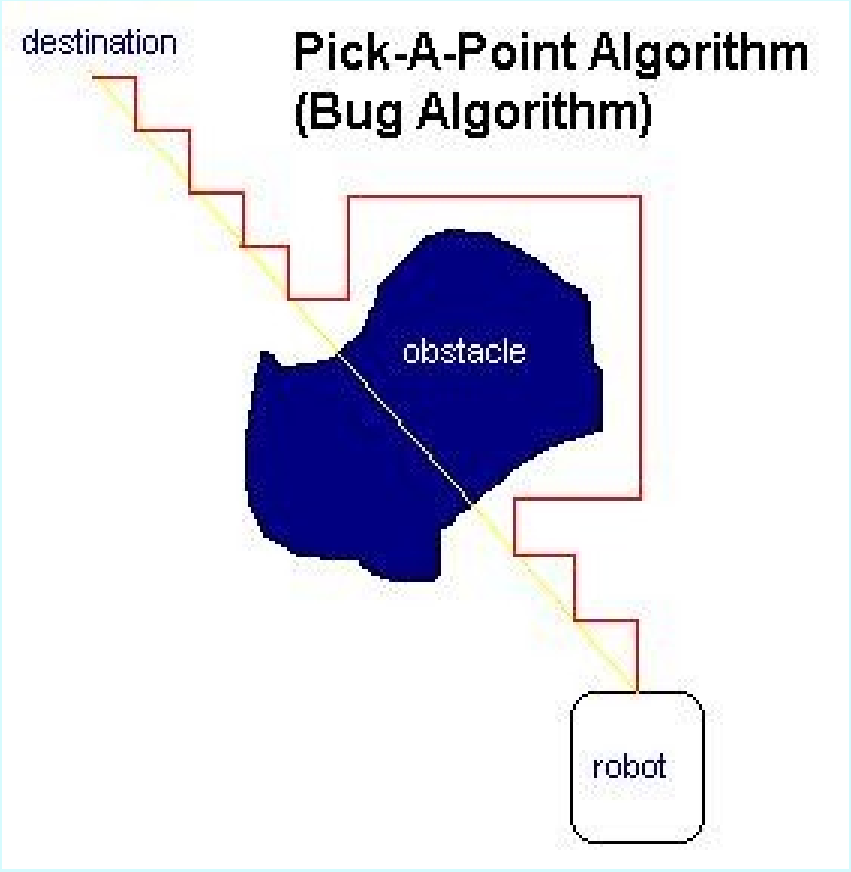


Future Suggestions

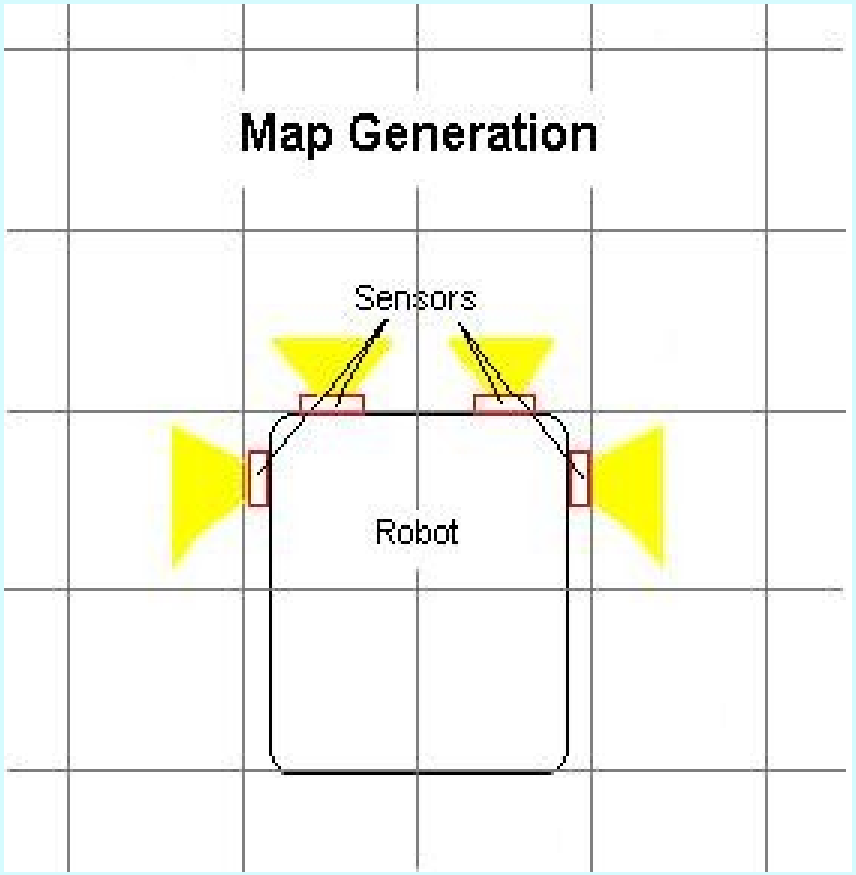
- Tweak PID algorithm for accuracy
- Examine need for 4 sets of optical encoders
- Evaluate the need for other technologies

Pick-a-Point Algorithm

- Moves at 90 degree angles in order to support grid map system
- Moves around obstacles by placing them on its left side
- Uses straight-line path to guide its motion

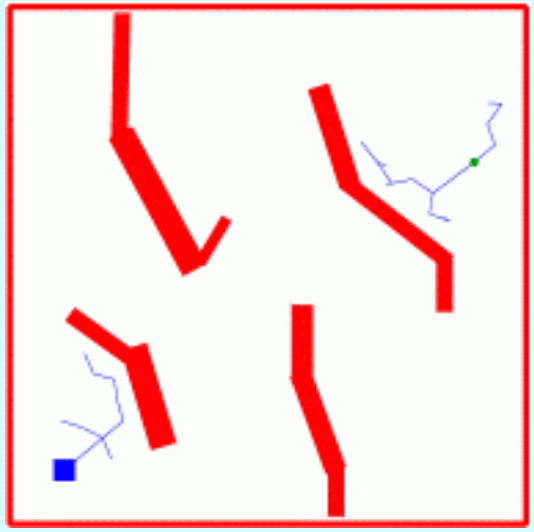


Map Generation



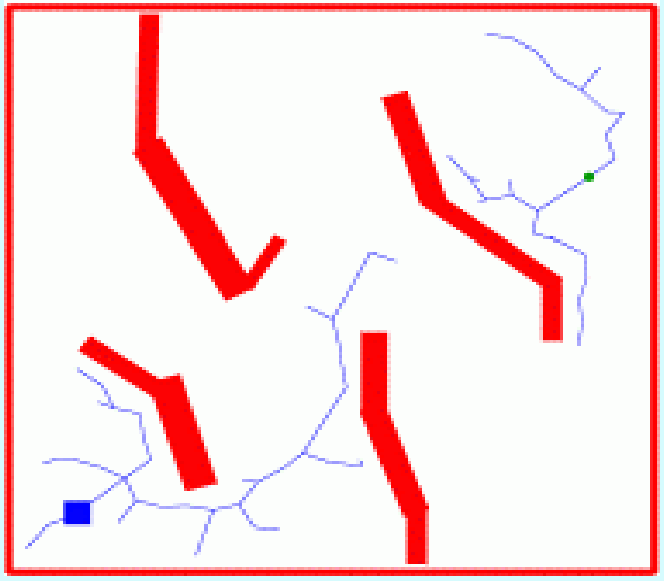
- Map grid is divided into 6” squares
- Two sensors located in the front and one on the front of each side can each view 1 tile of the map.
- Map updates as robot moves and finds objects during Pick-A-Point Algorithm

Path Planning

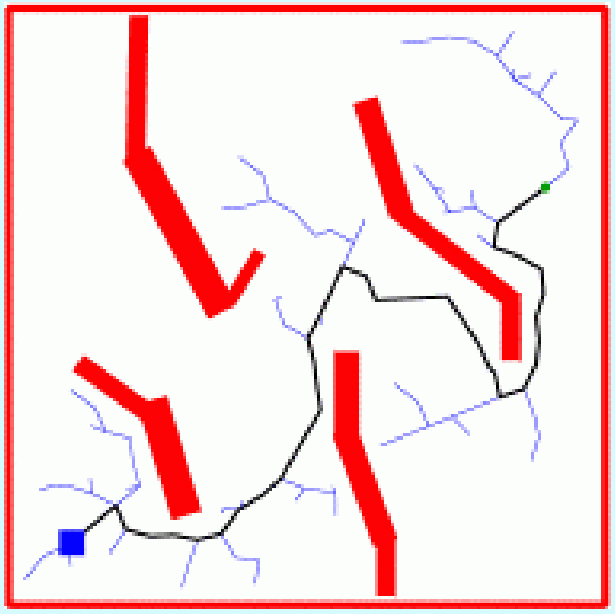


• With a pre-generated map, the robot can move about without worry of running into obstacles.

• First, determine multiple paths for getting around an obstacle.

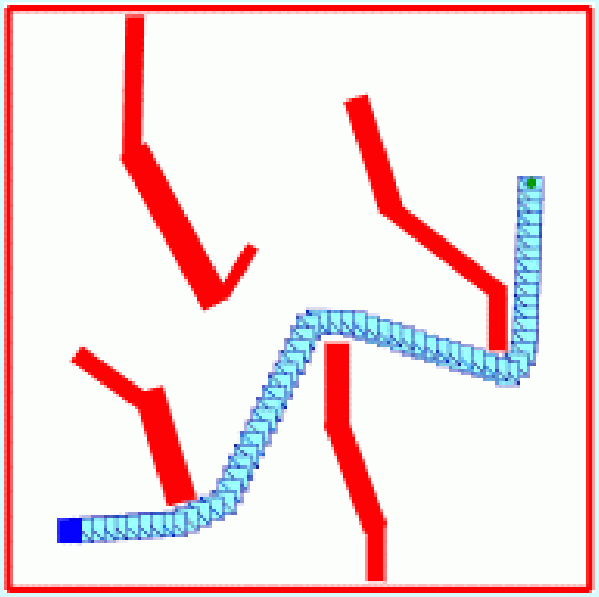


• Second, determine the optimal path around the obstacle path around the obstacle (traverses the least distance on way to destination.)



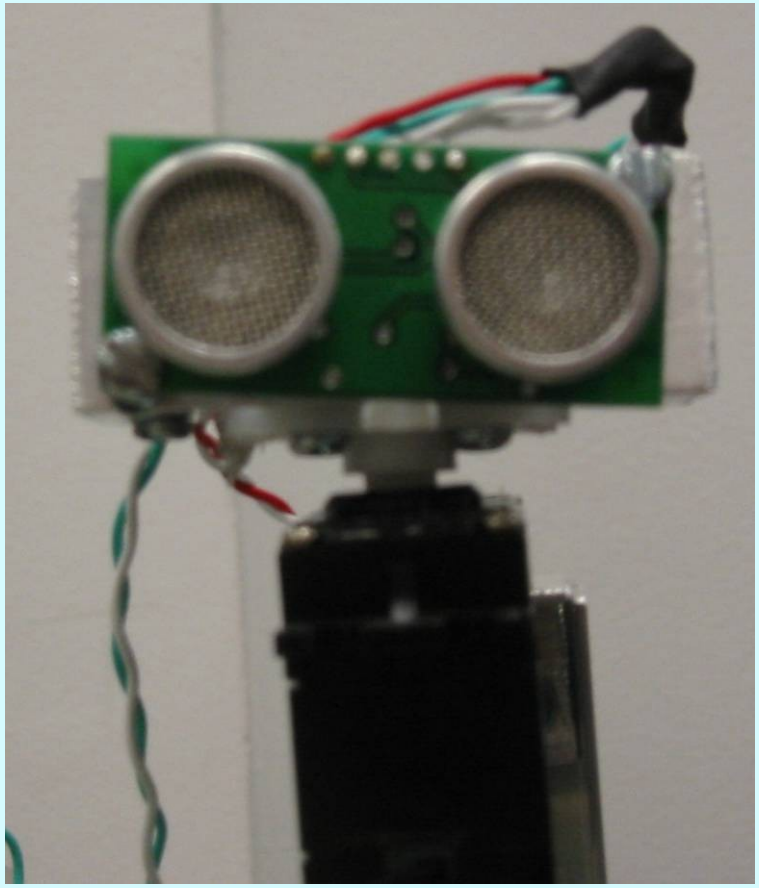
• Third, continue doing so around each obstacle until destination is reached. At this point, the quickest path to the destination around the obstacles has been determined.

• Finally, robot travels the points calculated to its destination.

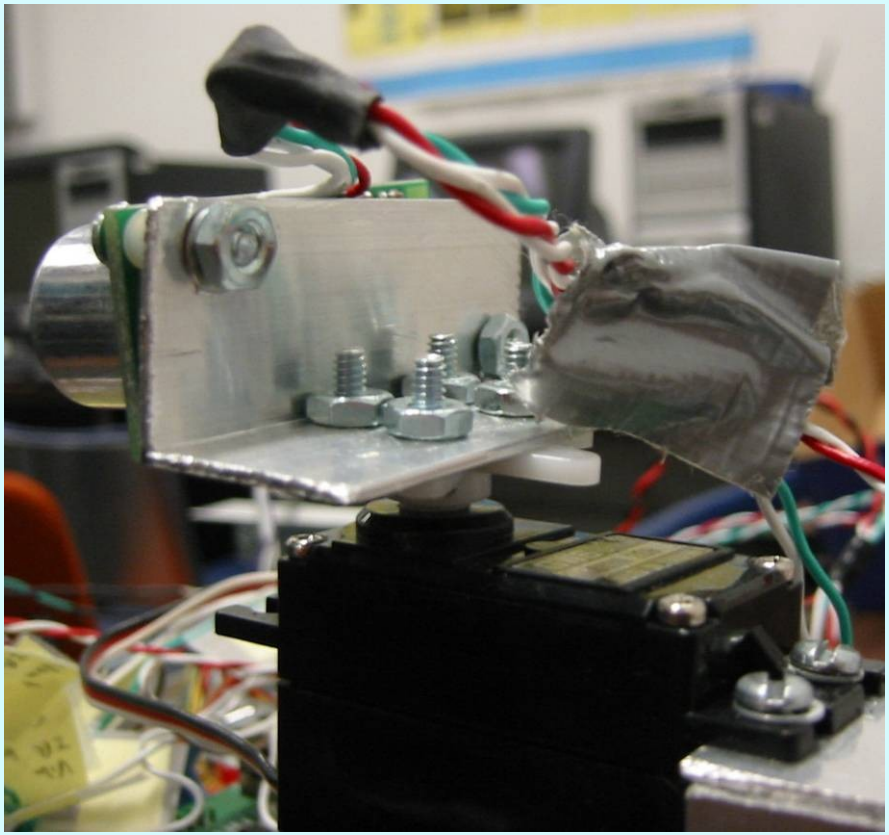


Sonar Mapping

- Our robotic surveyor is equipped with sonar, mounted on a servo motor.
- The sonar has a transmitter, which sends out ultrasonic pings, and a receiver, which detects the ultrasound that is reflected off of objects.
- This sonar is used by the robot to extend it's mapping and location abilities.



- The servo motor is controlled by a modulated-width pulse.
- Different pulse width turns the motor to different angles.



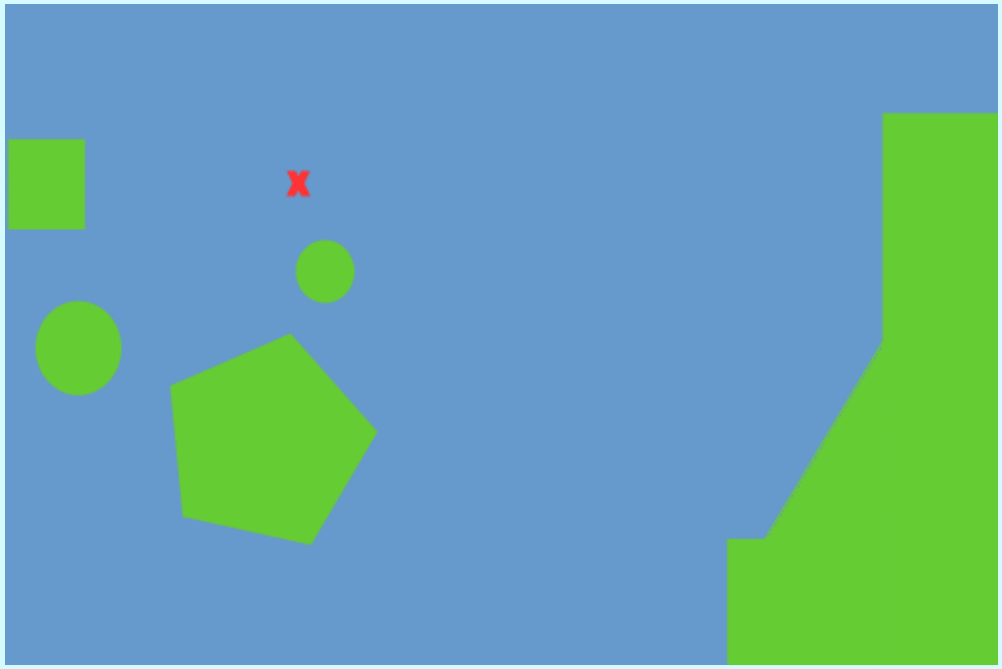
- The sonar is activated with a simple electronic Hi/Low signal.
- Once it is activated, it sends out a single ultrasonic pulse, and waits to receive an echo.
- The robot counts the time required for the return pulse.
- This time can be translated into the distance to the nearest object.

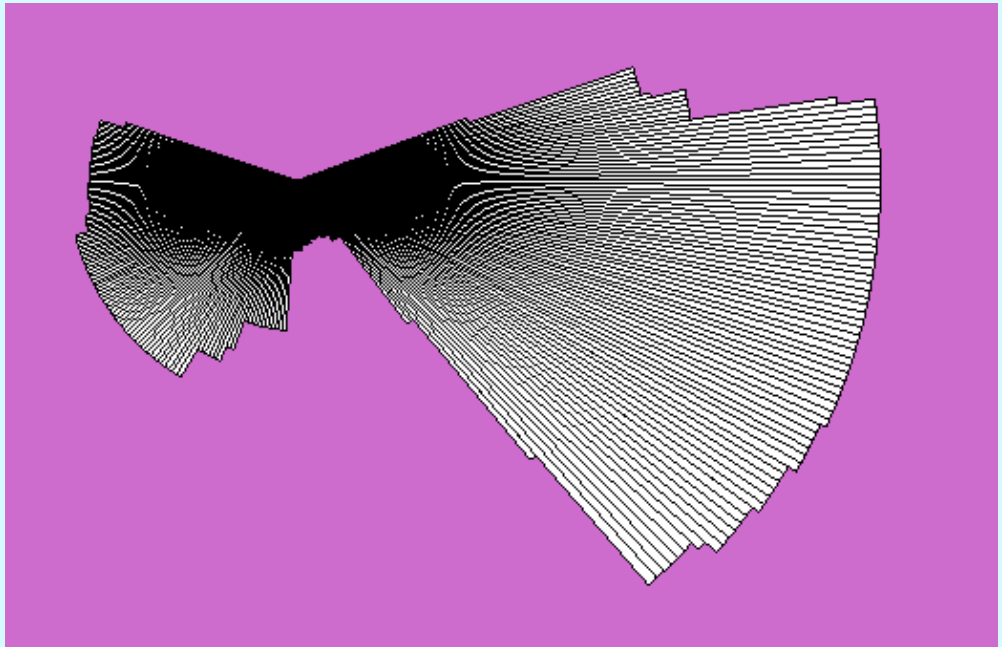
1. In this picture, the robot is at the X-marked place, the green areas represent objects.

2. The robot's base station tells the robot to start a sonar scan.

3. The angle from which an echo is received is not exact, but areas can be ruled out as 'empty' from the data.

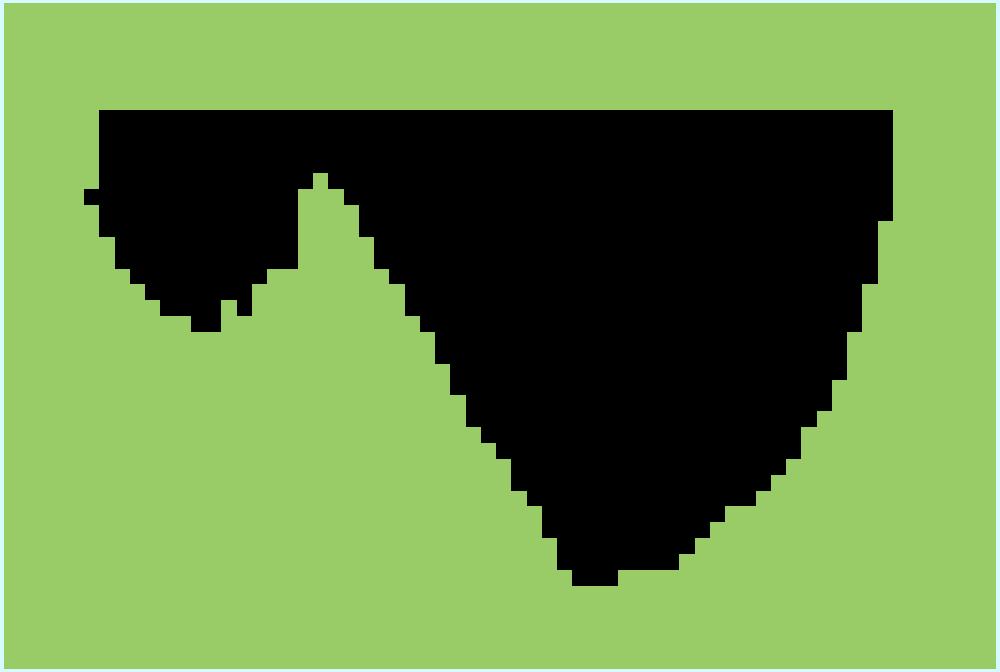
4. These data are collected and used to create a polar map representation of the area around the robot.





5. The polar map from the sonar data is transformed into a Cartesian map.

6. This map is merged with the robot's global map, and is used to help route planning.

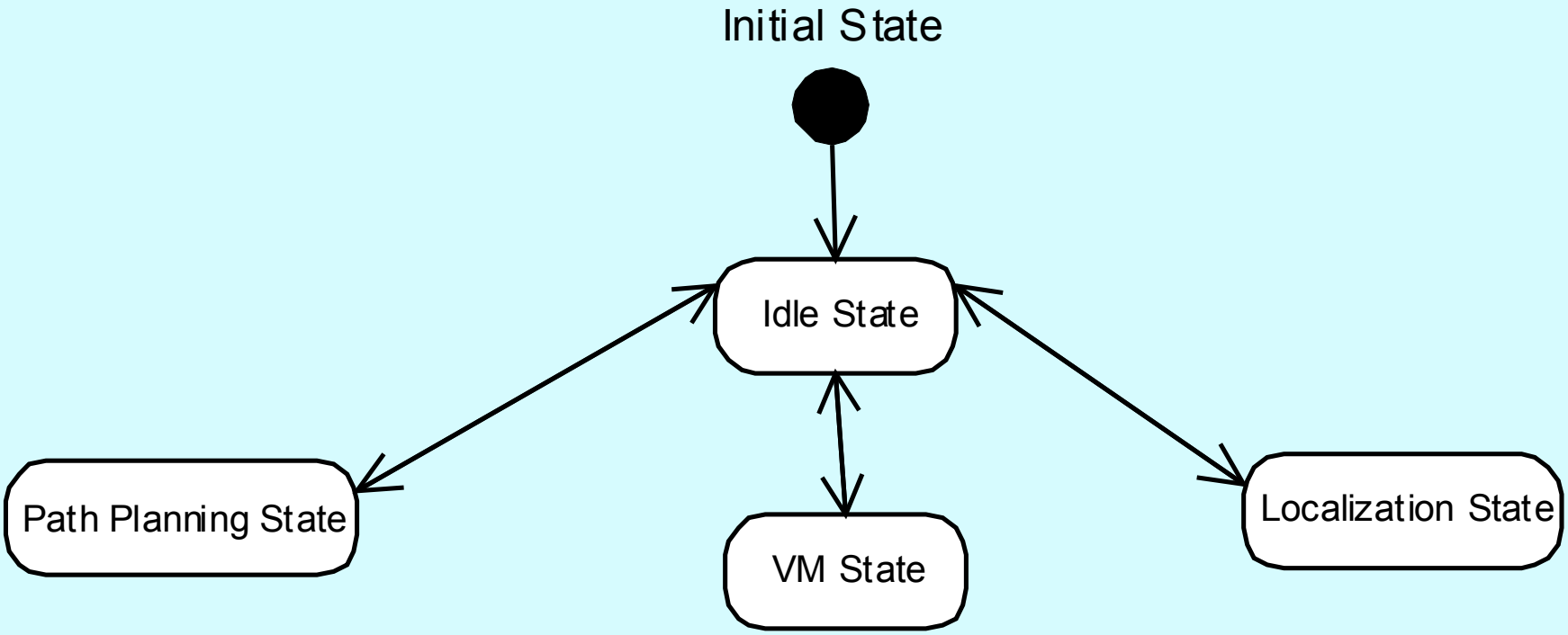


Operating System & Loader design

- The development of a procedure for loading data and applications into RAM through Wireless Ethernet (WiFi)
- Created a Virtual Machine (VM) to incorporate loader and facilitate integration with other groups.
- Written in Dynamic C

The operating system was programmed on a RabbitCore RCM3000

VM Operation Modes



Desired Capabilities of the Virtual Machine

- Direct VM state where the user interacts directly with the VM to upload, download, or select an application to run.

- Direct path planning and self localization states to where the control of the robot is turned over to its client mode