PI: Dr. Philip K. McKinley
Anthony J. Clark
Jared M. Moore
mckinley@cse.msu.edu

Solving VEX Robot Maze Task with Evolutionary Computation

RET Teacher Participant: Charles Payson
Battle Creek Area Mathematics & Science Center (BCAMSC)
Battle Creek, MI

MICHIGAN STATE UNIVERSITY

Co-PI: Drew Kim

Motivation

As part of the curriculum for the Battle Creek Area Mathematics & Science Center's Engineering Robotics course, students construct a VEX Squarebot 4.0 robot (Figure 1), and program it to navigate a simple maze known as the "labyrinth" (Figure 2). Students learn through these exercises that a program (or *controller*) executes its instructions in sequential order and that battery life can have a significant effect on performance, which leads to learning about using sensors as feedback mechanisms.

Students currently hand-design the controller to perform the task, which means that any structural changes to the maze or the robot must be explicitly dealt with by reprogramming the sequence of moves necessary for successful navigation. Evolutionary computation provides one method of automatically generating solutions to any configuration, enhancing the generalizability of the controller design problem. Removing the hand-designed element from this step enables the engineer to address other aspects of the engineering process while the evolutionary algorithm works towards a solution to the maze task.

Evolution & Initial Results

A model of Squarebot 4.0 was created in a rigid body dynamics simulation environment based on the Open Dynamics Engine (Figure 3). A rule-based, repeating control sequence consisting of 5 unique commands was devised to control the robot. Each command consists of a left and right motor speed as well as a time duration for execution. The commands simply repeat should the five rules execute for less than the simulation time.

The initial experiments show that evolutionary computation is not always a guarantor of an optimal solution. Figure 4 demonstrates the effect of deception, in which the task can have local optima that are difficult to avoid. In order to reach the target destination the controller would have to travel through lower fitness zones to find the global optimum.

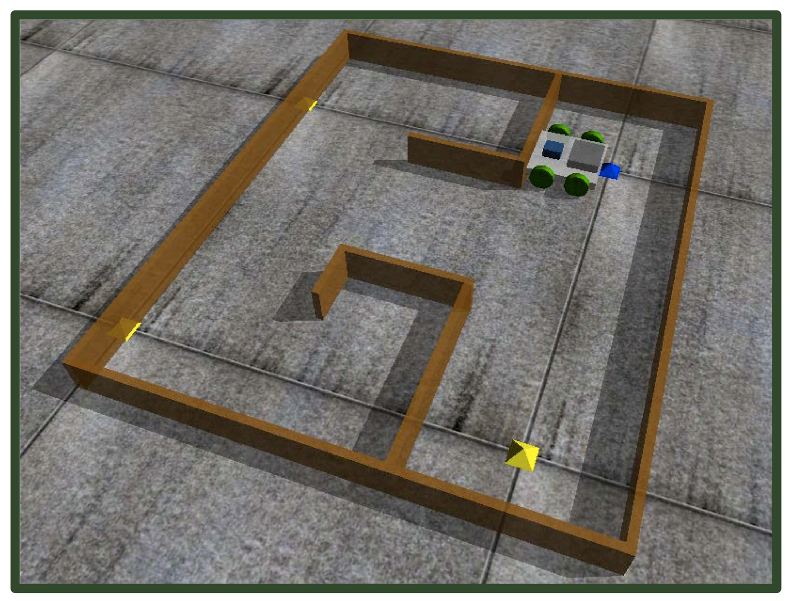


Figure 4

In the first evolutionary experiment, evolved solutions fell into a local optima wherein they were close to the goal, but were ultimately prevented from reaching it due to the presence of a wall. This deceptive trap is a known problem in the maze navigation domain, as mentioned in [1].

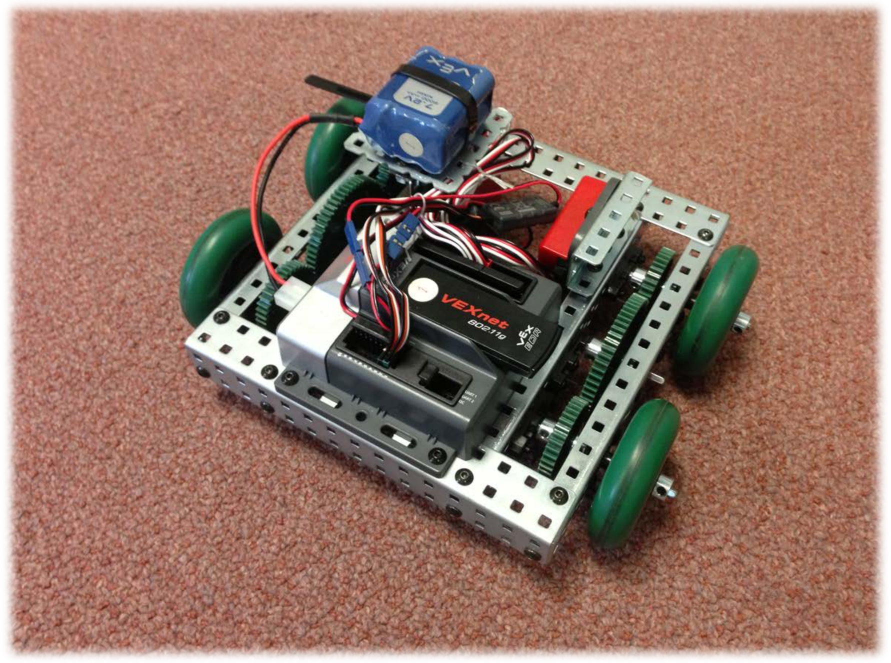


Figure 1: The Squarebot 4.0 design includes a chassis, left and right drive motors, a battery, and a processor unit.

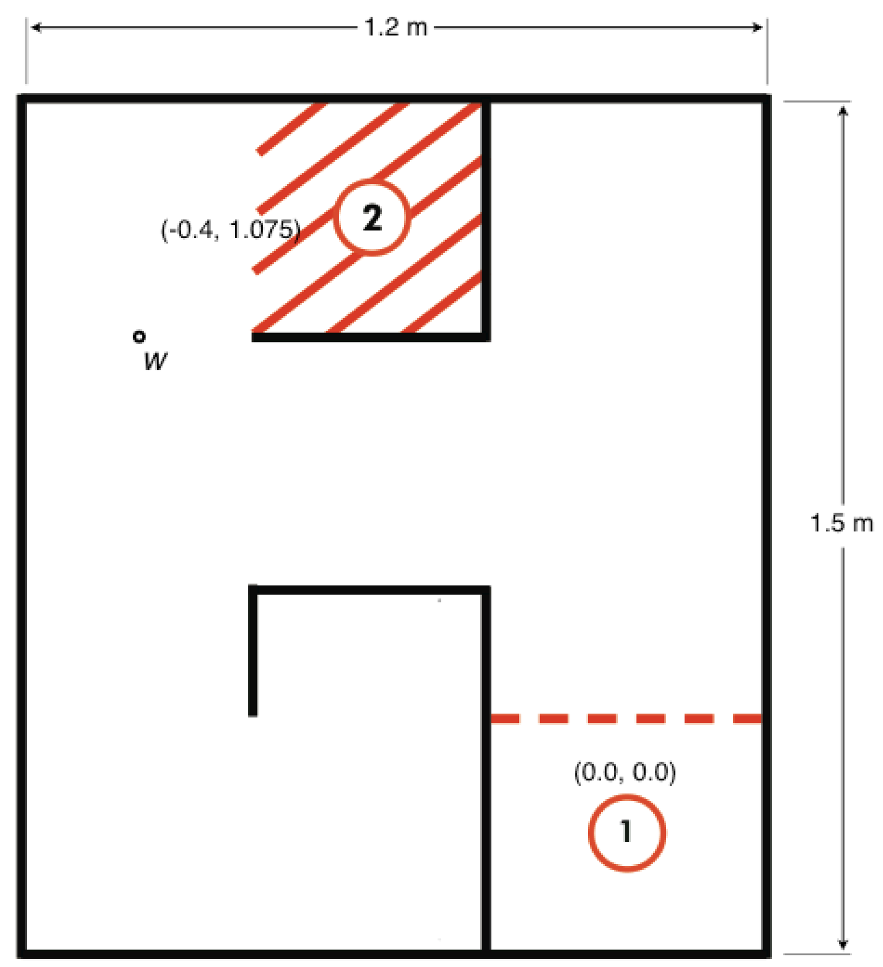
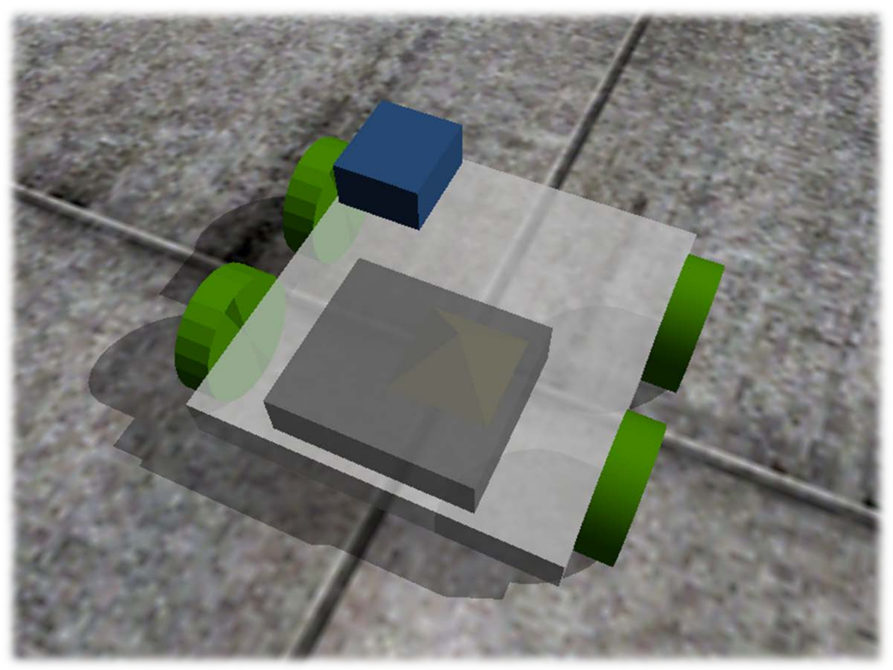


Figure 2: The labyrinth. Robots must move from the start at Point 1 to the end at Point 2 without crossing any of the black boundary lines. The point "*w*" is the waypoint referred to in the second evolution setup.

Figure 3 (below): The Squarebot 4.0 robot as represented in the ODE environment. The masses and dimensions of the major components were modeled after the physical robot.



Second Evolution Setup

To circumvent the deceptive trap, the fitness function was altered to include the waypoint *w* (shown in Figure 2), which is an intermediate target on the way to the endpoint. This gave individuals higher fitness scores for first going away from the endpoint (by going to *w*), and then proceeding to the endpoint. Figures 5 and 6 show the results of rewriting the fitness function.

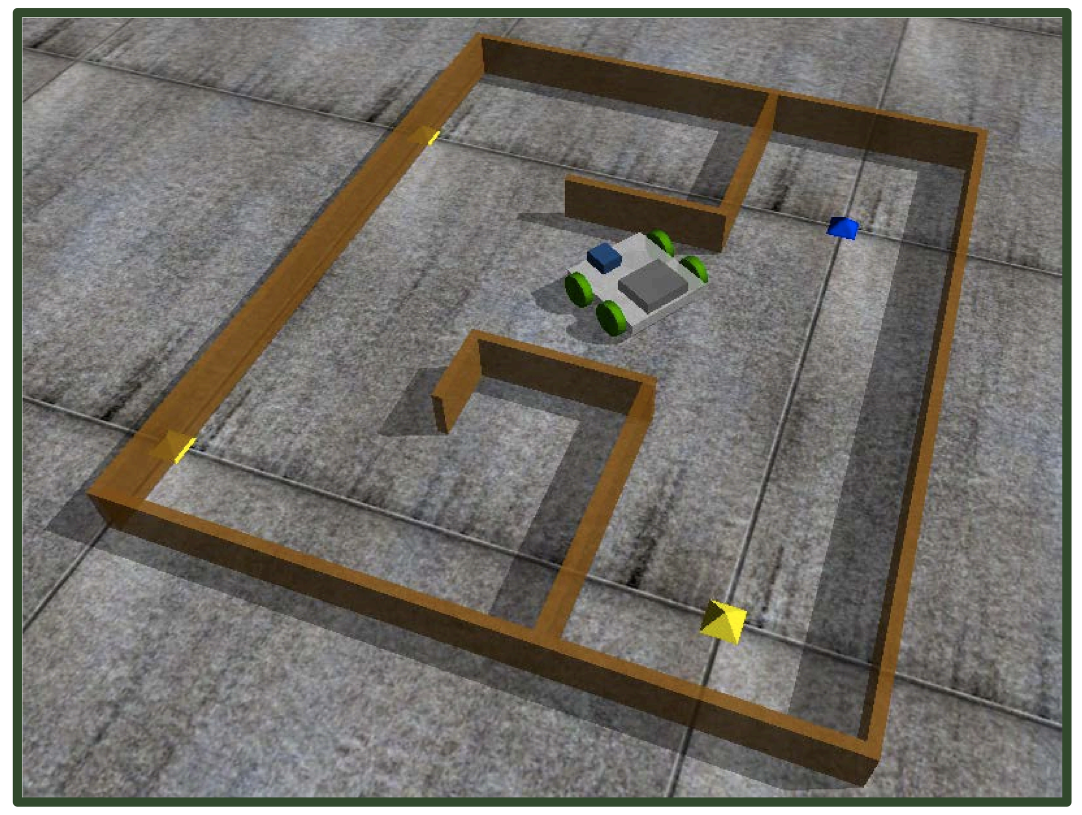


Figure 5

The controller first directs the robot to transit to waypoint *w* from the start point (yellow pyramid in lower right). Waypoint *w* was placed by the user; in future work the waypoint will be chosen automatically by as part of the genetic algorithm.

Once the robot reaches waypoint *w*, the only way to increase fitness is to drive the robot to the endpoint. The fitness function is intended to guide the robot along a path through waypoint *w* to the target position.

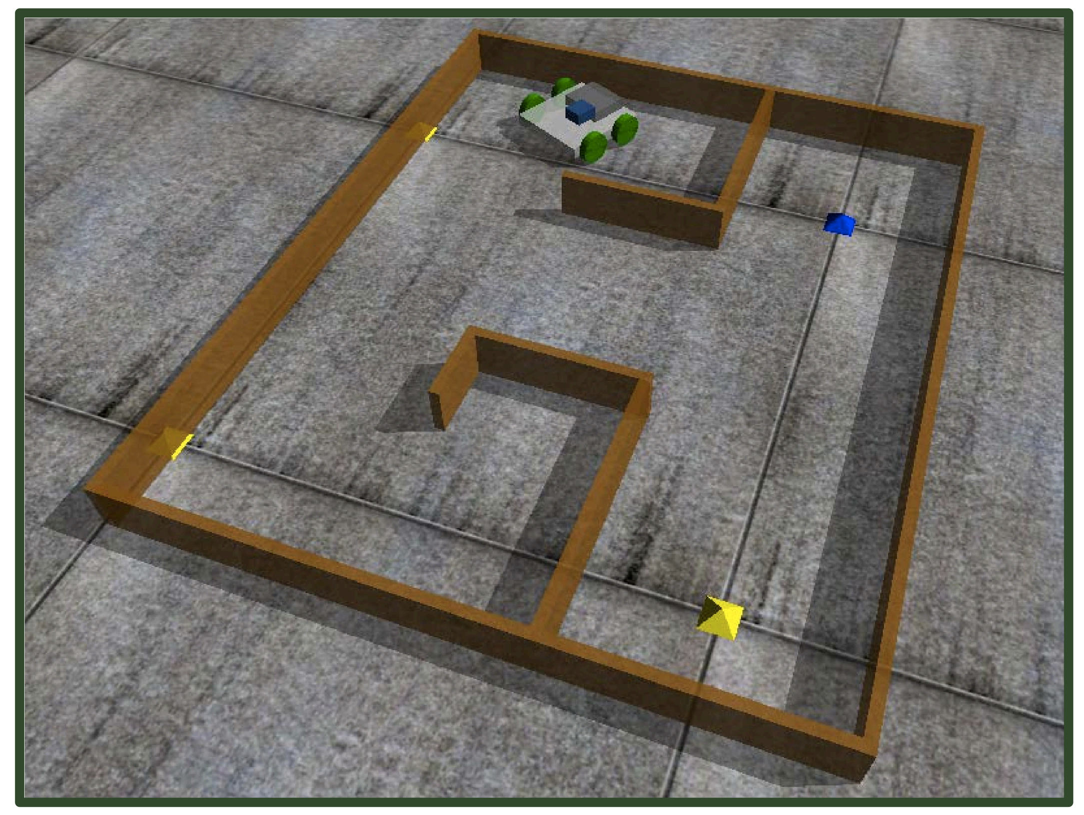


Figure 6

Future Research

Future studies will investigate the incorporation of sensory feedback as well as refining the fitness function. Additionally, morphology will be evolved with control to improve the robot's ability to navigate in varied terrain. For instance, a front wedge can be evolved to enable the robot to climb obstacles.

References

[1] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation, 19(2):189–223, June 2011.

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1300794

DART FOUNDATION

MOTOROLA

Consumers Energy Count on Us