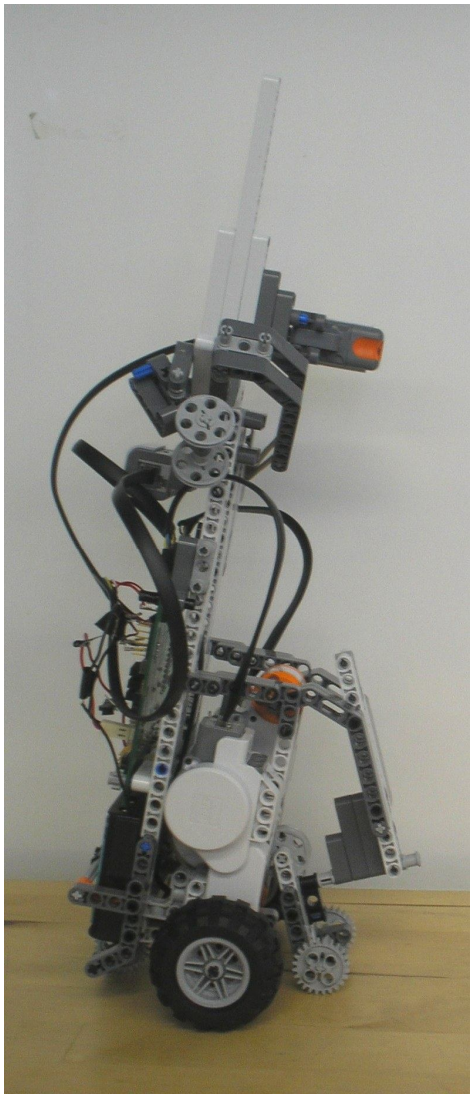
MIE 443 Contest Report



Group #: 2
Group Name: A.S.H Design Group
Group Members: Alejandro Martinez
Steven Weinberg
Harshvardhan Sukthankar

Contest Description & Rules

Six cans were to be pushed out of an 80 cm diameter circle using an autonomous robot made of LEGO bricks, utilizing LEGO sensors and controlled by the MIE 443 microprocessor. The cans are located 60 cm away from the centre of the field and spaced 60 degrees from each other with respect to the center of the circle. A smaller circle of diameter 16 cm was centered within the larger circle. The robot used for pushing the cans was to be initially contained within the 16 cm dia. circle. The robot had 2 minutes to push all the cans out of the circle.

Objectives

- Conforming to all constraints and contest regulations
- Pushing all the cans out of the outer circle in the least amount of time
- Making a robust design
- Making the design adaptable to further changes if necessary

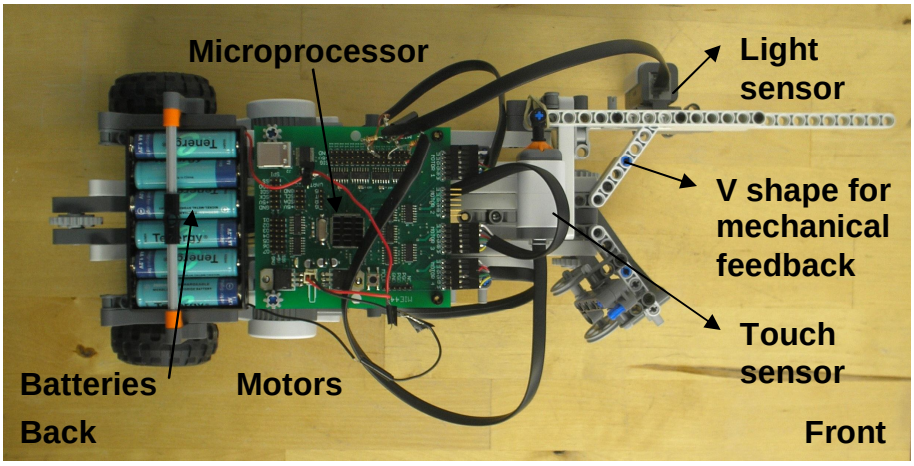
Concept Selection

Our initial idea was to use a robot that consisted of a cardboard cylinder with a diameter of 16cm and a height of 1m. The robot would be placed in the 16cm inner circle and then it would fall over by using an eccentric weight inside the cylinder. Then a wheel on the side of the cylinder would move it 40 cm so that it was inside the outer circle. When by using the eccentric weight previously mentioned it would move one way 40cm and the opposite way 80cm, displacing all the cans out of the circle. Upon being informed that a cardboard cylinder could not be used in the design we changed our design. This idea came from the work of Karl Sims, a computer scientist that devised a simulation called “blocky creatures” where for the first time both morphology and control systems of simulated robots was chosen using a genetic algorithm. In one of those simulations the robots had to take control of a cube in the shortest amount of time and one of the most common “solutions” was a robot that simply fell over and then started moving.

The initial design that was built was a line following robot that would fit inside the inner circle. The robot would be powered by two motors, for the left and right wheels, and would use the light sensor to find and follow the line. The idea was to initially align the robot towards the first can and push it towards the edge of the outer circle. Upon reaching the outer circle the robot would turn right and move towards the next can then following the outer circle and repeating until all the cans were pushed out.

The problem with this design is that we had to use dead reckoning to go back to the line each time the robot pushed a can. After some trials with a prototype we determined that dead reckoning was not reliable or fast enough to put the robot at the desired position and angle with respect to the line so that it could start following it. We also had to implement a PD controller for the robot to follow the line. We also didn’t have a good way of making the robot find the line if it lost track of it. We came to the decision of again changing our design.

Our revised concept was a modified robot with a longer length. In order to comply with the contest regulations, the robot was designed to start in a vertical configuration initially, and with the use of a third motor, lower itself relatively slowly into a horizontal position, instead of letting it drop down, to lower the chances of the robot breaking. The articulated wheel on the front of the robot serves both as a guide to lower the robot with two wheels in contact with the ground and once it is in the horizontal position, only one wheel touches the ground. Upon reaching its horizontal configuration, the robot would rotate clockwise, sweeping its "arm" that is attached to the touch sensor. When the arm hits a can, the touch sensor is activated and the robot moves forward until the can is pushed outside the field. The field boundary is detected with the use of a light sensor. When detecting the boundary, the robot would move backward into the centre circle, and then repeat the process until all six cans were pushed out of the field. We called our robot Rodney 1 in honor of Rodney Brooks, the inventor of subsumption architecture based robotics and current CEO of iRobot. Rodney 1's chassis consisted of two motors for turning the left and right wheels under the chassis. The chassis also carried the battery back and the microprocessor. A bar is slotted across the batteries to prevent them from falling out or being displaced during motion.



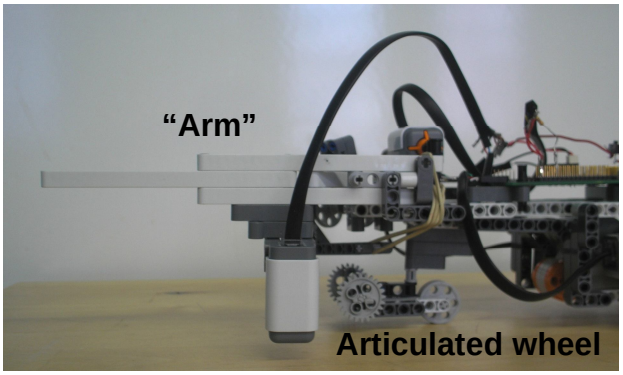


Fig 1: Rodney 1's components

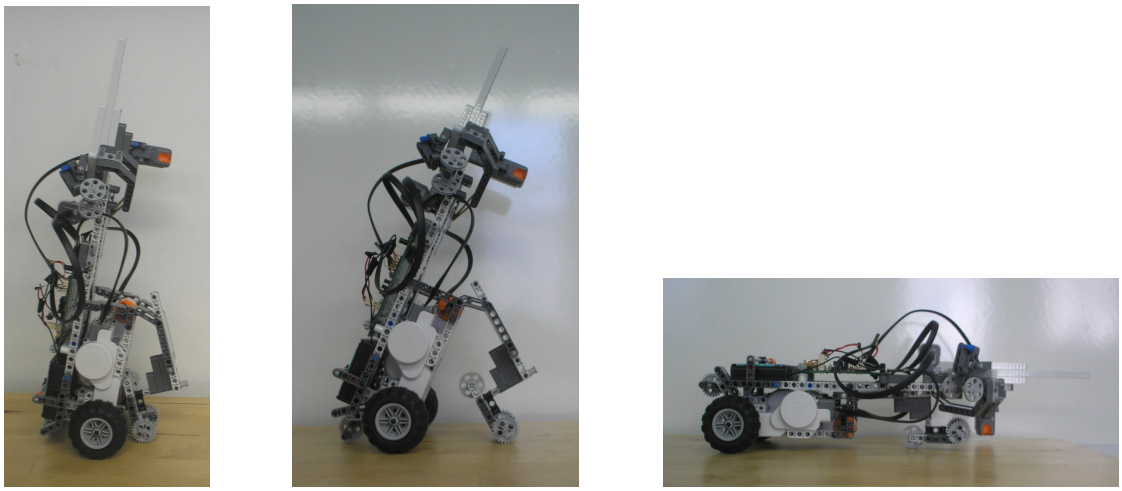


Fig 2: Rodney 1's moving from vertical to horizontal. The articulated wheel can be seen guiding the robot down with two wheels in contact with the ground and then when in a horizontal position only one wheel touches the ground

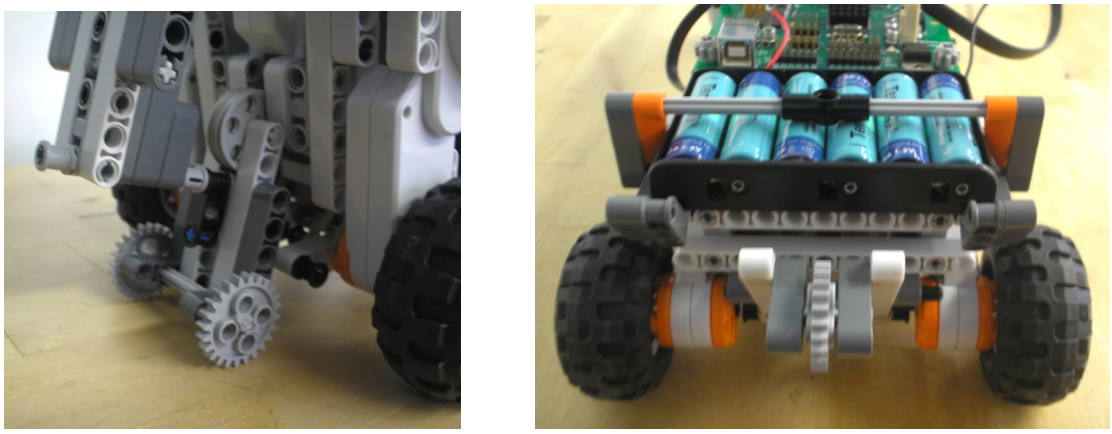


Fig 3; Detail view of the articulated wheel in the vertical position with two wheels touching the ground and a detail view of the battery holder that holds the batteries in a fixed position

Advantages over our previous designs and the competition

Our latest design lowered the amount of estimates we had to make to move the cans down to two; the radius of the outer circle and the angle between the cans measured from the center of the outer and inner circles. By estimate we mean any constant in the robot's program that represents a quantity in the outside world. In order to have no steady state errors due to dead reckoning, and a simpler program, we used feedback from the light sensor to find out where the outer circle is, and form the touch sensor to know where the next can is. We substituted an estimate that when used repeatedly leads to steady state errors

with a measurement and an estimate. Initially we thought we could only use a v shape in front of the robot to get feedback from the cans themselves about their position, a kind of mechanical feedback like fins in a rocket, but after some trials we concluded it wasn't reliable enough. Not only does this setup reduce the steady state error of the variables we are measuring, but it also allows our robot to push out cans from any position within the outer circle and the circle traced by the front of the robot as it rotates, without considering the length of the arm. It also allowed us to make a simpler program that is behavior based as shown below:

Pseudo-code

Move back the distance traveled when the robot goes from vertical to horizontal
Move motor with articulated guide wheel 90 degrees forward
Start of infinite loop
    If the touch sensor is activated move forwards
    If the light sensor detects the outer circle move back a distance equal to the radius of the outer circle
    Else rotate clockwise
Go back to start of infinite loop

Conclusion and possible improvements

Our robot was quantitatively the best performer since it took the least amount of time, 10 seconds, to push all of the cans out while moving out of the inner circle. It is also the most reliable since it can push cans from any position within the outer circle and the circle traced by the front of the robot as it rotates. An improvement that could decrease steady state errors even further could be to add another sensor on the back of the robot to locate the inner circle. When brainstorming our group thought of most of the configurations that participated in the competition but we came to the wrong conclusion that an arm that sweeps the cans outwards was not possible because of the torque needed from the motors. We should have tested that design. Still this wasn't a good enough design because it stayed inside the inner circle. The only design we didn't think of in advance was the one that followed the inside line. As explained when we talked about our second design, using dead reckoning to find the desired position and angle with respect to the line to then follow it is not reliable enough even with PD control to follow the line afterwards. The bad performance of that design confirmed our experiments. Our robot proves that an appropriate morphology that lowers the variables the robot has to estimate and the same amount of sensors to eliminate steady state errors before using those estimates makes the robot extremely robust and as a consequence very reliable.