York University

Vex Robot Design Report

Group C – Team 31

ENG 1000 – Introduction to Engineering Design
Section E
April 12, 2014

Won Mo Jung 213056031
Ariel Laboriante 212951984
Tyler Rajaram 213084520
Ezra Schwartz 211519485

Table of Contents

i. List of Figures

1.0 Introduction

Group 31 consists of four members: Andy Jung, Ariel Laboriante, Ezra Schwartz and Tyler Rajaram. Within the first week of the Vex Robotics prototyping activity in ENG 1000, the tasks were split up evenly within our team. Ariel was responsible for creating the Creo models. Andy, who has the most experience in robotics, was assigned to lead the team during the building process of the robot. Ezra was in charge of programming the robot using the easyC software, and Tyler was assigned to collect photos, screenshots and test results, as well as create the PowerPoint presentation. The team worked well together and each of our members played an invaluable role throughout the robot design process.

MEMBER:	Ariel L.	Andy J.	Ezra S.	Tyler R.
TASK:	Creo Models	Robot Design	EasyC Coding	Documentation

2.0 Design Description

2.1 Three-Wheel Design

Due to the most success during testing, the three-wheel configuration was selected for the final design. This consisted of two large wheels in the back, each powered by a separate motor, and a single medium-sized wheel in the front. This lead wheel was left to freely rotate an equal distance to what the back wheels travelled. This was supposed to eliminate the problem caused by having two different sized wheels with the same rotational speed. One rotation of the back wheel would try to move the car farther than one rotation of the medium because of the different circumferences.

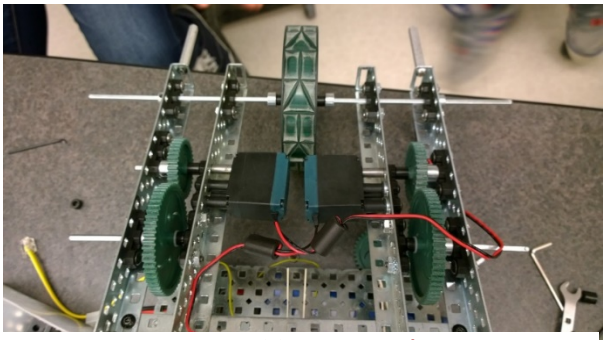

Figure 1 Bottom view of final design (without back wheels)

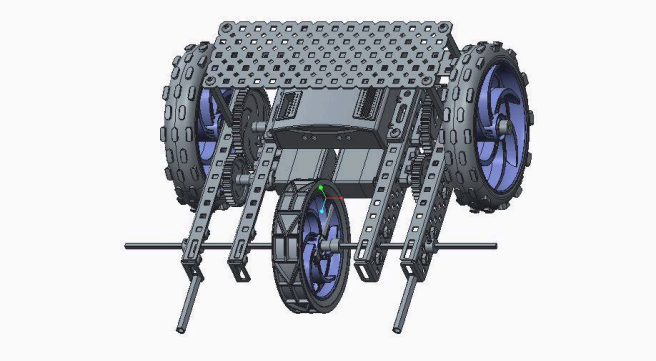

Figure 2 Early Creo model of the final design

Using the lead rolling wheel was a much more efficient solution than trying to adjust the rotating speed of wheels using different sized gears. There is always internal friction between the gears, and the more gears used, the more energy lost to friction. For that reason, the design limited the use of gears to two per motor. From our previous designs, we noticed that the use of compound gears was not very effective, as the more gears we used, the greater the difference there was in rotational speed for each side of the robot.

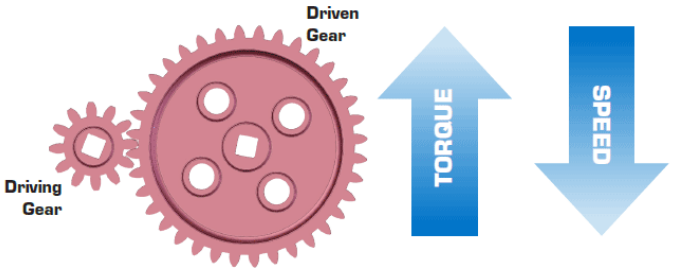

Figure 3 Visual of gear configuration used in design

A medium-sized, 36-toothed gear was connected to each motor, which powered the larger 60-toothed gear attached to each large wheel axle. The purpose of this was to give each back wheel enough torque to climb the various ramps and to maneuver the different obstacles within the course. The large wheels had more grip than the other wheel sizes, and therefore needed the extra torque to drive. The resulting speed would be slow, but the wheels would not slip.

The use of three wheels rather than four saved the design some weight, which allowed the use of the heavier wheels. Also, there were less points of contact between the vehicle and floor, and therefore less energy lost to friction because of the loss of a wheel. This design should allow for quicker turns because of the middle wheel. The heavy wheels distributed the weight lower and reduced the risk of the vehicle flipping over itself.

Although this design would decrease our robot's speed, the design should allow the vehicle to maneuver through the course easier and lower the risk of getting stuck at some point in the course. By having the two rear wheels power the vehicle, it should travel in a much straighter path than a four-wheel drive design. The smaller wheels had a smaller contact surface area than the larger wheels; the wider wheels are able to roll in a straight path for a much longer time and distance. Furthermore, there are only two wheels pushing the chassis and therefore a lower chance that one side will try to move faster than the other, causing a change in direction.

2.2 Remote-Control Features

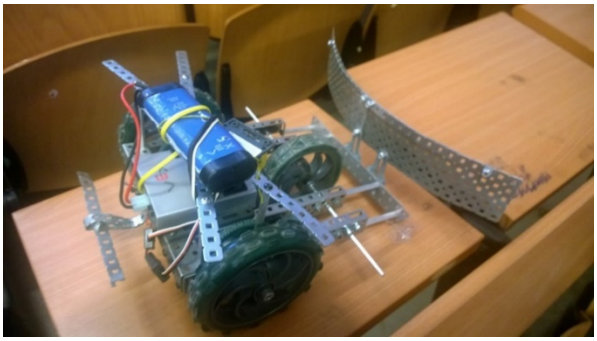

Figure 4 Final design of remote control vehicle

For the remote control section of the competition, it was a necessity for our design to have certain key features to optimize performance. The slow, powerful design, which provided a lot of torque, helped pushed the ball and wooden block quite easily. The manual design included an arm with a raised curved plate, which would be used to either push or roll the ball up the ramp. This plate was designed to be high enough not to interfere with the side guards all around the edges of the course. Below the curved plate was a flat rail used to push the block off the gangplank section. This arm added length to the vehicle, which would help when going down the final ramp by reducing the chance of spinning, overturning or falling out of the course. Most of the weight is at the back of the vehicle, which provided risk of flipping over when pushing the ball up the ramp. As a precaution, a metal safety piece (the tail feature) was attached at the back of the robot to stop the robot from completely flipping over.

2.3 Autonomous Features

For the autonomous robot competition, the top quality of this design was the straight path of movement. With the use of multiple switches and preplanned turns, the robot should be able to avoid most parts of the course where it could either fall off or get stuck.

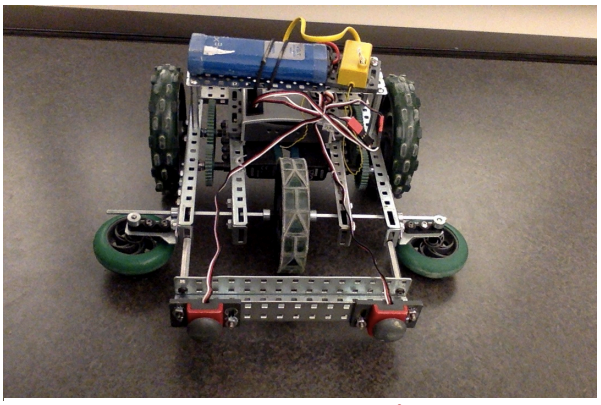

Figure 5 Autonomous robot design (without the limit switches)

On the front rail, two separate bumper switches were used on either side. By using two switches instead of one, the car could detect impact with a wall faster, and spend less time hitting a corner of the chassis into a wall. When either bumper switch is triggered, the vehicle should go back and turn either left or right depending on the number of prior hits. By using two bumpers, we were also able to make the robot straighten itself out along the wall. Our code allowed the robot to continue moving forward (for 200 milliseconds) after the collision with the wall is detected. This allowed our robot to square up with the wall, and the turns became more consistent.

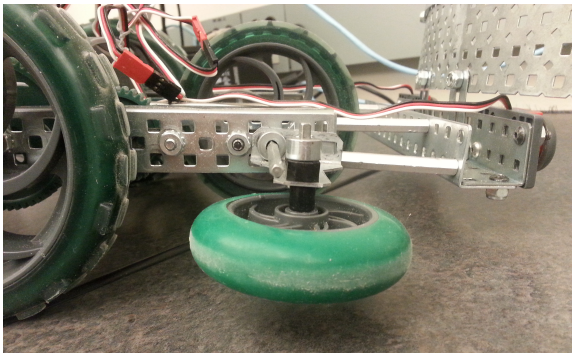

Figure 6 Side-wheel robot feature

The design had two horizontally placed small wheels on both sides of the lead wheel. Their purpose was to align the vehicle parallel to the edges of the course when the vehicle moved towards a side. Since there were sections of the course that were missing edge guards, this was a clever solution and it greatly lowered the risk of exiting the course. They also stopped the rear wheels from getting too close to the sides when moving forward, which could cause performance issues. The horizontal wheels would help the vehicle take the straightest paths through the course, and prepare the vehicle best before making a left or right turn.

On both sides, in between the small wheel and bumper switch was a limit switch. Its purpose was to prevent the vehicle from getting stuck at the sections where the edge guards were missing. Moreover, the horizontal wheels could only work if the vehicle moved towards the edge at a small angle; too large of an angle and it would just get stuck. In contrast, the limit switches would only get activated if this angle with the wall was too large. When activated, the vehicle would stop, straighten the robot out accordingly, and then continue forward. We were inspired by Lucas' group to add the limit switches, but our coding, which enabled the robot to first move backwards (and get out of the "sticky" situations) and then slightly turn to become straight again, enabled our robot to run continuously through the track during test runs.

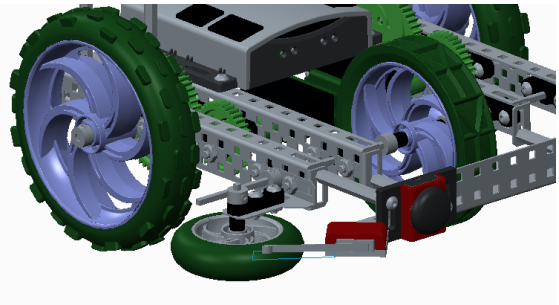

Figure 7 Creo model showing the location of the limit switch