Final Report

MIE444: Mechatronics Principles

Team 3

Primary Lab Session 1

Name	Student Number
Zian Zhuang	1002449870
Siyan He	1001335021
Shing Chak Samuel Wong	1002172123
Xu Deng	1003268126

Submission date: December 15th, 2019

Executive Summary

The performance of the designed robot could be seen via https://drive.google.com/drive/folders/1v2XIXEBMU4g2E_n-ukBZUjfHMCAWBa1G?usp=sharing, which demonstrates how the objectives were achieved for each milestone. To complete the tasks of obstacle avoidance, object pick-up and delivery, maze solving, and localization, the rover was designed to have two layers of platforms to hold electrical and mechanical components. Three omni-wheels were mounted on the bottom layer for movement, controlled by three DC motors with encoders. A total amount of seven ultrasonic sensors were used with six for maze navigation and one for object detection. The readings from the sensors could identify the distances from obstacle, which were used for most of the code developments. The gripping mechanism were 3D-printed and controlled by two servo motors for the motion of gripping and lifting. A close-and-lift command was given for the gripper at the beginning of the maze solving strategy until the loading zone was reached.

The overall performance of the robot met the requirements as assigned. The rover could detect and avoid the obstacles for most circumstances. However, due to the limited time of testing, the rover could not perfectly avoid collisions with walls while solving the maze. A function of block detection was developed. When the rover arrives at the loading zone, it searches the distances' differences of front sensors at both top and bottom layers as it rotates. However, the ultrasonic sensors could not be fully relied on. When the block is at an unusual angle or the rover is too far from the block, the rover fails to detect object. Thus, the location of the block was manually selected and placed for Milestone 3. The gripping mechanism functioned as desired for the first run of Milestone Three but failed during the second run. This situation happened since there were design flaws for the gripping mechanism, which resulted that the grippers were not tightened enough to hold the block. Localization functioned with an error during the first run and performed flawlessly during the second trial. By integrating the probability logic from Matlab, the rover could identify its location and signal with a LED light when it reaches the loading zone or the delivery point.

In conclusion, the designed robot achieves the requirements, but did not function as desired when all of the features were integrated. Further code development and more testing are required for improvements.

1. Detailed Rover Design

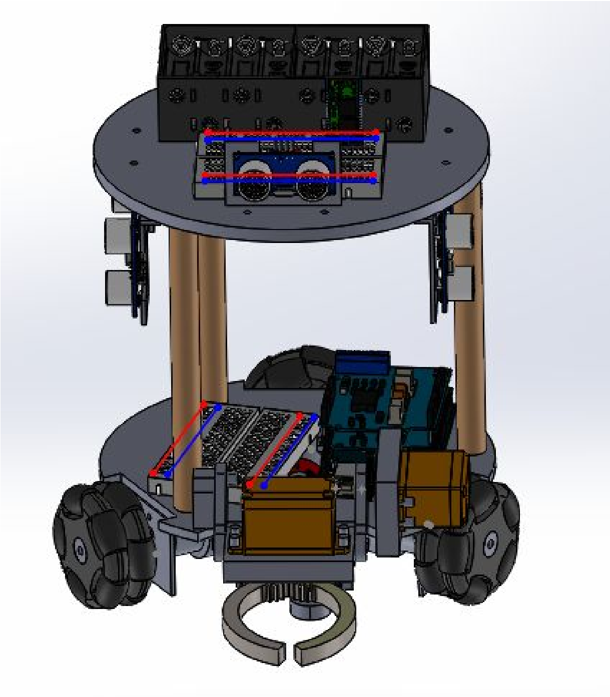


Figure 1. Front View of Final Rover Design

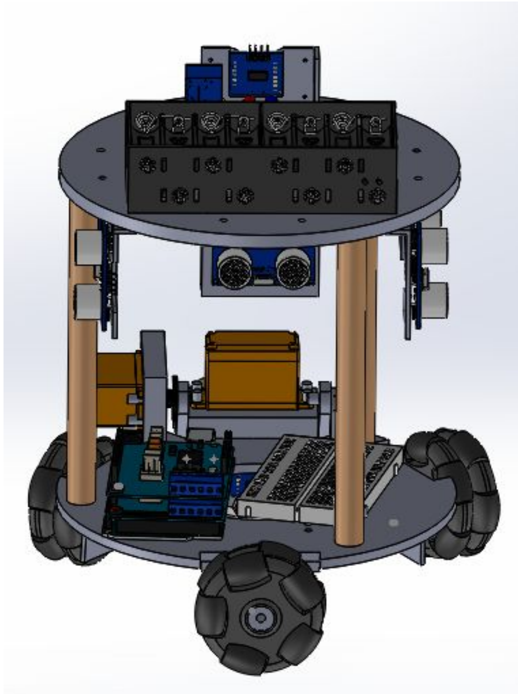


Figure 2. Back View of Final Rover Design

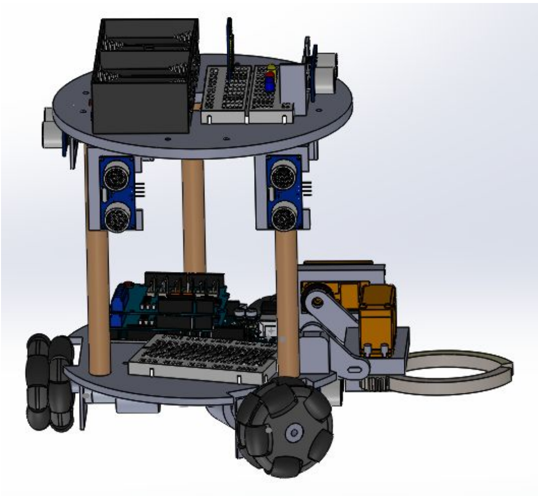


Figure 3. Side View of Final Rover Design

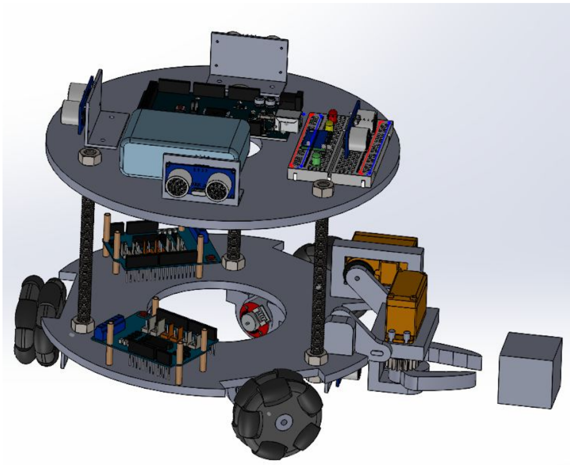


Figure 4. Proposed Rover Design

The final design of the rover is similar to the proposed design as shown in Figure 1. Some features were modified to meet the requirements.

1.1 Power and Control

Initially, a 12-volt rechargeable battery was the power supply. Through the process of rover testing, the provided battery often failed to power the entire rover to achieve maximum performance. An example is when motors were powered, it would lead to a voltage drop at Arduino, which caused Arduino to self-rest and couldn't perform the remaining tasks. Thus, the rechargeable battery was replaced by two separate packages of six 1.5-volt battery to supply the power for motors and Arduino respectively. 2 DC-DC converters were used to ensure proper amounts of voltage to be 6 volts and 7 volts, as 6 volts meets the power rating for motors and 7 volts is the minimum voltage required for Arduino. For motor control and communication with Arduino, two L298N motor drives were used initially but later found difficult to provide desired motor power. To achieve similar power distribution for each motor and ease wiring debugging, two L298P motor shields were used instead, and the pins of the arduino board were modified to ensure an adequate amount of PWM ports.

1.2 Ultrasonic Sensors Configuration

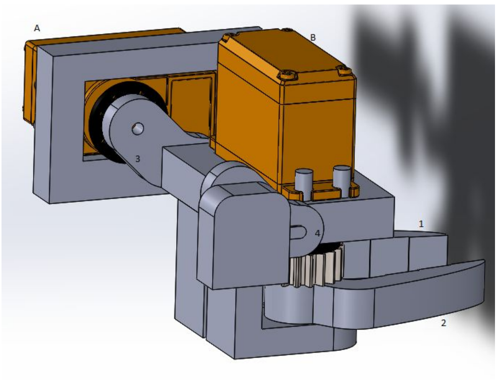
Four ultrasonic sensors were proposed to be installed at the top layer of the rover for detection of the front, back, left, and right sides. For the bottom layer, an ultrasonic sensor was also installed for object detection. However, after the first milestone, it showed that there were blind spots between the gaps of the sensors. Therefore, the number of the ultrasonic sensor was increased to two with equally-spaced distances from the centre for both left and right sides of the rover as shown in Figure 3. The addition of side sensors were used to ensure if the rover had entered or passed the intersection areas. It also helped to adjust rover’s heading direction by aligning both sensors on the side with the wall. The number and the position of the other sides’ sensors remain the same. The front top and bottom sensors are aligned in the same vertical plane with the purpose of object detection by sensing a distance difference of the object and wall.

1.3 Rover Movement

The principle of the rover movement remains the same as the proposed design. Three omni-wheels are installed at the bottom layer with a separate degree of 120^{o}. The left and the right wheels are driven while moving straight, all wheels are driven clockwise or counter-clockwise for self-rotating. Back wheel is also driven when minimum heading angle adjustment is needed.

1.4 Gripper Mechanism

The original design of the gripping mechanism is demonstrated in Figure 5. The gripper is lifted up while navigating to ensure clearance required for the bottom sensor. When the rover arrives at the pick-up location, Motor A drives linkages 3 and 4 to move down. Motor B drives spur gears to move linkages 1 and 2 towards each other for gripping. Once the block is gripped, the gripper restores to its original position. Same principles apply while unloading the object. For the final design, the geometry and the dimension of the gripper were modified to allow a larger gripping space and larger contact areas as shown in Figure 6. The length of the gripper was increased from 62mm to 79mm.



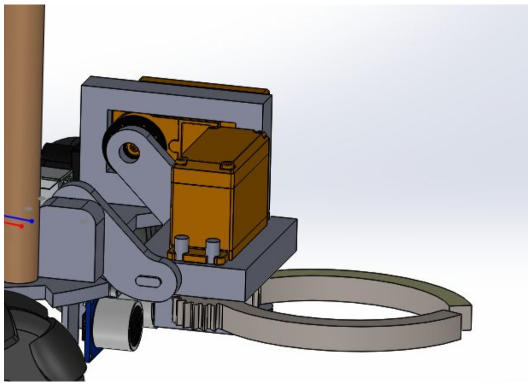


Figure 5. The Original Gripper Mechanism *Figure 6. The Final Gripper Mechanism*

1.5 Direction Calibration

A compass with gyro sensors was specified to obtain the initial direction of where the rover was facing from the proposal. As the testing was conducted, it was found that the compass provided readings with errors due to the influence of other magnetic electronics. Since the error was generated randomly, it was hard to filter out the undesired data. To address this kind of problem, the use of the compass was abandoned and replaced by new maze-solving and orientation initialization algorithm which was generated with the use of the ultrasonic sensors.

2. Maze Solving Strategy

The overall detection for the rover is based on the seven ultrasonic sensors with six for navigation and one for detection. The walls are detected by the top six sensors to prevent any collision between the walls and the rover. The load is detected by sensing a large difference between the top front sensor and the bottom front sensor.

To solve the maze, the robot follows specific steps of commands. Since the rover would be randomly placed in the maze, the first task for the robot was to adjust itself to be parallel to the walls for further actions. This is achieved by checking the sensors on the top layer constantly and rotating itself until the two sensors of the left or the right side detect an identical distance. The initial facing direction was found by sensing the longest distance of the four sides of the rover. At this step, the initial set up was achieved and ready to run the maze solving strategy.

The maze solving strategy consists of a left-hand-rule and a right-hand-rule. The left-hand-rule set the priority of movement to be turning left when sensing a front wall exists in a pre-set distance. The right-hand-rule operates the opposite way, turning right is the priority command of execution. The choice of rule depends on the location in the maze. The pocket was defined to be 3-sided-wall location. When the rover senses three walls exist around it at the starting point, it then follows right-hand-rule. If the starting point is not a pocket, it follows the left-hand-rule until it reaches a pocket location to switch to the right-hand-rule. Since the configuration of the maze is fixed. the overall running strategy was generated to minimize the running distance. As such, the error of each function could be reduced.

For the indication of localization, three LEDs was used to show the different status of the rover. When the red LED is on, it means that the loading zone has been arrived. The yellow LED light indicates that the rover is in the process of finding block. Once the blue LED is on, it shows that the robot has arrived at the loading zone, and the block has been unloaded.

During the movement of solving the maze, the moving forward function set a distance of three inches by calculating encoder pulses. The rover keeps tracking the steps and the changing of the walls. The probability of reaching the loading zone is generated and updated remotely in Matlab through bluetooth as the rover moves. The rover would indicate that the loading zone is arrived by flashing the orange light when the probability of the loading zone is greater than 0.65.

The desired function of finding the block is to rotate 90 degrees clockwise or counterclockwise after each step of moving forward when arriving loading zone in two different paths. After the block is picked up, it keeps moving around the loading zone in the same turning direction until it senses that the distances reading from front and left are small, which there are only two circumstances as shown with red colour in Figure 4. Four hard-coded paths are built for each of the circumstances to reach the four unloading points.

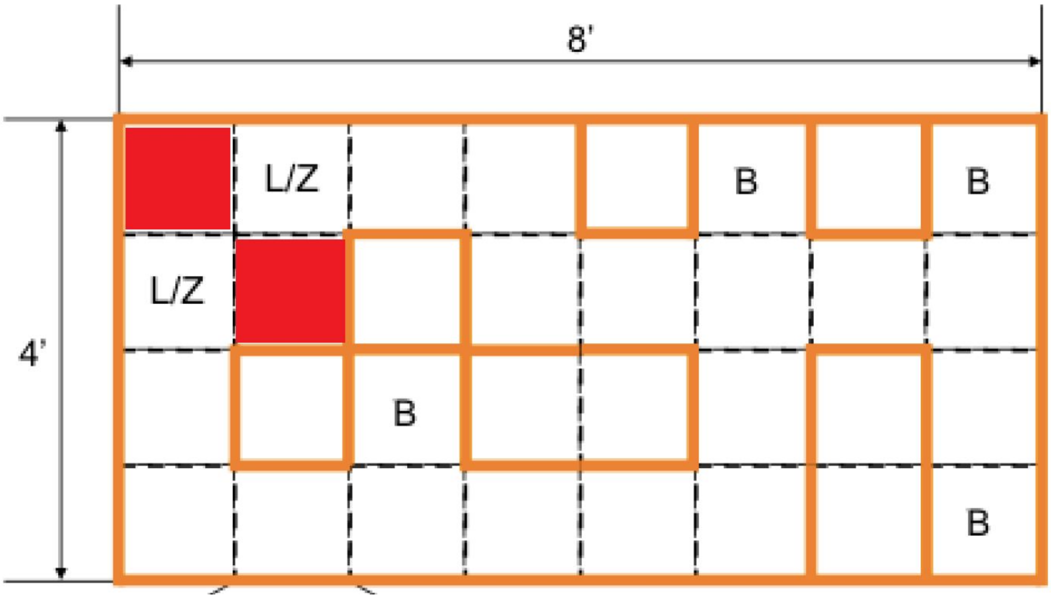


Figure 4. Indication of Two Cases

Each of the paths was calculated to have the exact amount of steps to reach the unloading points. After the selected path is uploaded, the rover follows the command and arrives at the designated point with blue light turns on which indicated that the load is successfully delivered.

3. Final Result

Our rover made attempts in all three milestones. Three major areas will be focused and analyzed in this section. They are obstacle avoidance, localization and autonomous pick-up and delivery of the load.

3.1 Obstacle avoidance

Multiple hits occurred during both runs in Milestone 1. In the first run, rover did not seem avoiding obstacle properly. Hit occurred right after it started. Furthermore, wheels were stuck easily to the wall and they did not have enough traction to draw the rover out of the situation. Multiple manual assistance were required. In the second run, adjustments on motors and codes were made and that certainly improved the performance of obstacle avoidance. It did go better than the previous run. Hits still happened but it did complete a 20-blocks run. It turned properly and avoided collision. The major modification between the runs was the codes. A more complex approach was used in the first run. Rover was attempted to do drifting with motor encoders, in order to keep itself at the middle of the lane. However, it did not work at all so hits occurred. Codes were modified during the break. It was replaced by a more simple and straight-forward approach. Rover was simply driven by motors without using the encoders. Motors were calibrated to make straight run. Turns were made with the feedback from the ultrasonic sensors. It worked better significantly.

In Milestone 3, rover did a good job in avoiding obstacles. A function was used in this milestone. It simply tries to keep distance between the rover and the walls. The motors were calibrated to keep the rover away from the walls. It was based on the drifting function in Milestone 1 but with a more simple approach.

Therefore, practical solution for improvement in obstacle avoidance is to simplify the codes. Our team has overthought the situation and should take a step back and do it in the most straight-forward way. Besides, more testing should be conducted in Milestone 1. A lot of adjustments were made right before the runs so testing and calibration were not done properly due to time-constraint. After Milestone 1, testing was completed and codes were modified and that resulted in a greatly improved performance in obstacle avoidance.

3.2 Localization

Localization was used in Milestone 3. During the first run, it did not perform properly. It hit the wall once so errors were present in locating itself in the maze. The possibility changed so it did not recognize itself arriving at the loading zone. During the second run, localization performed flawlessly. It localized itself at the entrance of the loading zone, with an LED light indication. The overall performance of localization is satisfactory.

In order to improve the performance further, more testing could be done prior to milestone runs. Testing can reduce the occurrence of errors.

3.3 Autonomous pick-up and delivery of the block

In Milestone 2, block pick-up and delivery were completed with the assistance of Bluetooth. In Milestone 3, it has to be completed by the itself autonomously. During the first run, it did pick up the block successfully and delivered it to the assigned unloading point. However, in the second run, the block could not being picked up by the rover. There was an error of the gripping mechanism. Before the second run, the gripper was broken and it was only fixed temporarily with tape. As a result, the positioning of the gripper was not correct. It did not grip the block firmly, therefore, the block slipped out of the gripper during the delivery in the run. Delivery of the block was not completed.

Sudden damage of mechanism is something that we did not foresee. However, the consequence was something we could avoid. In order to improve the performance, a back-up plan should be made prior to the accident. Moreover, it can also be improved by a more solid design. A solid design of the gripping mechanism can avoid breakage.

The overall performance of the autonomous pick-up and delivery is acceptable. The coding of the pick-up and delivery was good. However, the hardware did not provide enough support to complete the tasks. The performance can be improved by a risk management plan.

4. Discussion

The rover consists of various parts. The CAD drawings made the mechanism of the rover and the gripper. The circuit provided power and connection to all the devices. The coding provided a logic for the rover to drive and to complete the tasks. Thus, they are all irreplaceable.

The best feature of the rover is the circuit. It was stable at all time. Wiring and power was never an issue in testing. They were reliable and completing the tasks flawlessly. The circuit was the least time-consuming part of the rover. The process of making the circuit was effortless.

The worst feature of the rover is code execution complexity. Issues were present in the codes. Those issues made the rover not performing well in both Milestone 1 and 3. Testing should be done more frequently to lower the chances of errors happening. However, the errors were understandable due to various reasons. Firstly, there were time-constraint in coding and testing codes. The schedule was packed so not much time was left for testing. Code debugging was time-consuming and that was the most difficult part of the project. Secondly, none of our team is proficient in writing codes. Most of the time was spent on writing code and debugging. That made our team process inefficient and led to undesirable results eventually.

Summarizing the experience learnt from the project, a good job delegation is essential. This can maximize the efficiency of a team. A time-consuming task should be done the sooner the better. A project management will be useful. Besides, testing should be conducted more frequently. Most of the time the performance of the codes are unpredictable. Bugs and errors always exist. The only way to resolve them is testing. Last but not least, do not overthink a problem. The obstacle avoidance in Milestone 1 was certainly overthought by our team. It was supposed to be a simple task, but a complex method was used and that resulted in time-wasting and poor performance. Time was wasted and more modification was needed in order to rectify the errors previously made. The cost of this could be unbearable.

5. Cost Analysis

This section lists the costs of components consisting of the rover and calculates the total cost.

Table 1. Costs of components consisting of the rover

Part Name	Description	Number of Parts	Price (CAD $)	Total Cost (CAD $)
Electrical Components				
Ultrasonic Sensor	To detect walls and objects	7	1.07	7.49
Arduino Mega 2560	To store program	1	17.8	17.8
USB programming cable	To connect Arduino to computer to upload program/ connect with serial port	1	1.7	1.7
Motor shield, L298P	To control motor speed/direction	2	6.05	12.1
AA battery case with 6 slots	To contain batteries	2	1.14	2.28
AA battery pack	Power supply	2	14	28
MG995 servo motor	To drop/lift and open/grip gripper	2	4.39	8.78
Electrical Tape	To bundle wires	1	1.2	1.2
DC-DC converter	To convert power	2	5.92	11.84
HC05 bluetooth module	To remote control	1	4.78	4.78
Jumper wires	To connect pins	Miscellaneous	0.15-0.16	5
Mini-breadboard	To provide pins	2	1.86	3.72
LED light	To signal	3	0.65	1.95
6V DC Motors (with encoder)	To drive wheels and record steps	3	12.99	38.97
Miniature Toggle Switch Single	To switch on/off of power supply	2	0.29	0.58

Pole				
Mechanical Components				
Birch Dowel S	To support the plates	1	0.25	0.25
3D printed parts	For various uses, support, grip, etc.	Miscellaneous	1.15	1.15
Laser cut parts	To support other components	Miscellaneous	12.86	12.86
Omni Wheels	To run	3	9.12	27.36
Fasteners	To fasten and assemble	Miscellaneous	1	1
Total				187.75

The total material cost for the rover is $187.75 which is under $250.

6. Contribution Table

Table 2. Distribution of project work

	Zian Zhuang	Siyan He	Shing Chak Samuel Wong	Xu Deng
RFP (20%)	25%	25%	25%	25%
CAD (10%)	10%	55%	0%	35%
Hardware acquisition (2%)	50%	40%	0%	10%
Electronics wiring (8%)	55%	15%	20%	10%
Coding (15%)	70%	10%	10%	10%
Testing (15%)	50%	25%	10%	15%
Final report - Design	10%	10%	10%	70%

Strategy (10%)				
Final report - Discussion (10%)	0%	0%	100%	0%
Presentation (10%)	5%	85%	5%	5%
Overall	32.4%	26.8%	18.6%	22.25%