UNIVERSITY OF WATERLOO
Mechatronics Engineering

FINAL DESIGN REPORT

MTE 380 Mechatronics Engineering Design Workshop

Prepared by
Daniel Dworakowski
Rae Jeong
Nhat Le
Ji-Won Park
Fan Zhang
3B Mechatronics Engineering

March 29, 2017

University of Waterloo,
Waterloo, ON N2L 3G1

March 29, 2017

Dr. Bill Owen

Associate Director of First-Year Engineering
Department of Mechanical and Mechatronics Engineering

University of Waterloo,
Waterloo, ON N2L 3G1

Dear Dr. Owen:

Group 12, consisting of D. Dworakowski, R. Jeong, N. Le, J. Park, F. Zhang, is submitting this report for review.

Members are responsible to work collaboratively and competitively, applying the design process to fulfill the projects goal. The goal this term involves designing and building a device to go over a wall and delivering a package. The detailed design analysis, selection of material and parts, and final CAD drawings are outlined in this report. The purpose of this report is to present the detailed design and the analysis of the robot.

In this project, the group aims to study and apply the design process in a multidisciplinary environment. Members will make decisions and evaluate trade-offs by applying knowledge from different courses. Likewise, students will develop further interpersonal skills, such as leadership and teamwork. This engineering challenge will give group members the experience they need to go through a full design process. This ultimately provides an opportunity to produce a useful and interesting product.

Group 12 believes that this project provides an opportunity to gain the skills required to be an effective engineer. To that end, all the students in group 12 are aware of the University of Waterloos academic integrity policy 71 student discipline and have been and will be following it throughout this project. Non-original ideas will be credited and cited to the original author.

Sincerely,

Daniel Dworakowski   Rae Jeong   Nhat Le   Ji-Won Park   Fan Zhang

Executive Summary

Citizens live in the region between the inner and outer walls, making them vulnerable to attacks from monsters beyond the wall. The inner city needs a safe, efficient, and autonomous method to deliver supplies over a wall to those in need in case of emergency. This report outlines the design analysis and parts selection to build a device capable of solving this problem. The report explores design analysis in the fields of mechanical, electrical and controls engineering of a robot capable of jumping over obstacles. Engineering design analysis techniques such as stress analysis, differential dynamics modelling, kinematic analysis and simulations facilitate analysis and evaluation of design choices. Project management techniques such as work breakdown structure divides a complex problem into small and solvable tasks. Using the bill of materials, a budget of 500 Canadian dollars is requested. Responsibility of design analysis and calculations are shared by discussing the strengths and interests of each team member. Gantt charts aid in prioritizing those tasks to meet important milestones along the project. Given the progress of the project so far, estimates were made for important milestones to project completion, including a construction check in week 9, with completion being projected in week 12. Currently, an initial prototype has been created and tested to ensure the viability of the project. In conclusion, the team makes detailed engineering design decisions that are either theoretically guaranteed to perform correctly or are experimentally tested to ensure the proper functionality.

Table of Contents

List of Figures

List of Tables

1 Introduction and Background

1.1 General Background

1.2 Problem Forumulation

1.2.1 Problem Definition

1.2.2 Desired Functions and Goals

1.2.3 Constraints

1.2.4 Objectives

1.2.5 Design Selection Criteria

1.3 Review of the Selected Design

2 Project Management

2.1 Schedule

2.2 Budget

2.2.1 Bill of Material

2.2.2 Cost analysis

3 Final Mechanical Design

3.0.1 Key Features

3.0.2 Design Exploration

3.0.3 Testing

3.0.4 Predicted Results Vs. Actual

3.0.5 Lessons Learned

4 Final Electrical Design

4.1 General Electrical Design

4.1.1 Key Features

4.1.2 Design Exploration

4.1.3 Testing

4.1.4 Predicted Results Vs. Actual

4.1.5 Lessons Learned

5 Final Control Design

This section will provide a detailed overview of the final implementation of the robot control architecture. When compared to the previous report, the changes have mainly involved the controller design as well as the finalization of the pole location algorithm.

5.1 Robot Position and Velocity Control

The overall design of the Robot's position and velocity controller has changed to better take into account external disturbances that original in the mechanical design. Originally, the design called for two PID controllers driving each of the wheels individually as shown in Figure 1.

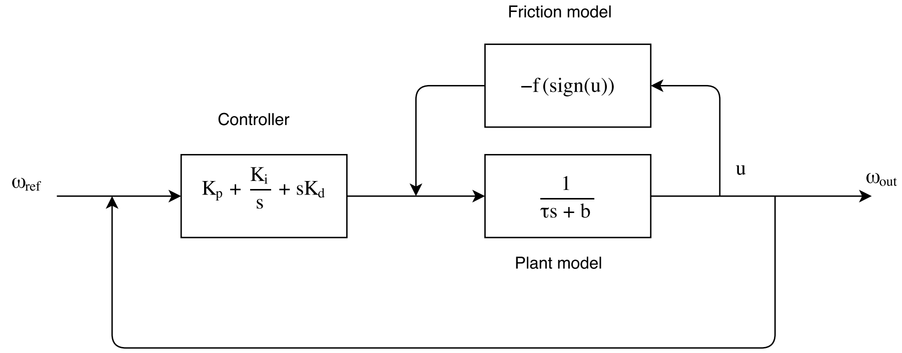

Figure 1: Controller for an individual Wheel.

The individual speed where determined through kinematic equations related individual wheel speeds to the robot's overall speed and angular velocity. Figure 2 shows the kinematic model of the robot. Specifically, r describes the radius of curvature of the path, L the length of the robot, ω_l, ω_r describes the rotation rate of the left and right wheel respectively. V_l, V_r describe the linear velocities of each of the wheels. R is the radius of the wheels and θ describes the rotation about the origin.

With this model the following equations can be derived, full details are found in the previous report. V and ω describe the global tangential and rotational velocities desired. It is important to note the assumptions that were made obtain these results. Mainly, that no slip exists between the wheels and the ground, and that all components within the drive train are always in contact.

$$\omega_l = \frac{2V - \omega L}{2R}, \omega_r = \frac{2V + \omega L}{2R}$$

$$V = \frac{R}{2}(\omega_l + \omega_r), \omega = \frac{R}{L}(\omega_r - \omega_l)$$

It was assumed if the wheels were individually capable of maintaining speed the robot overall would be able to follow a straight track accurately. However, this left the issue that the true variable that

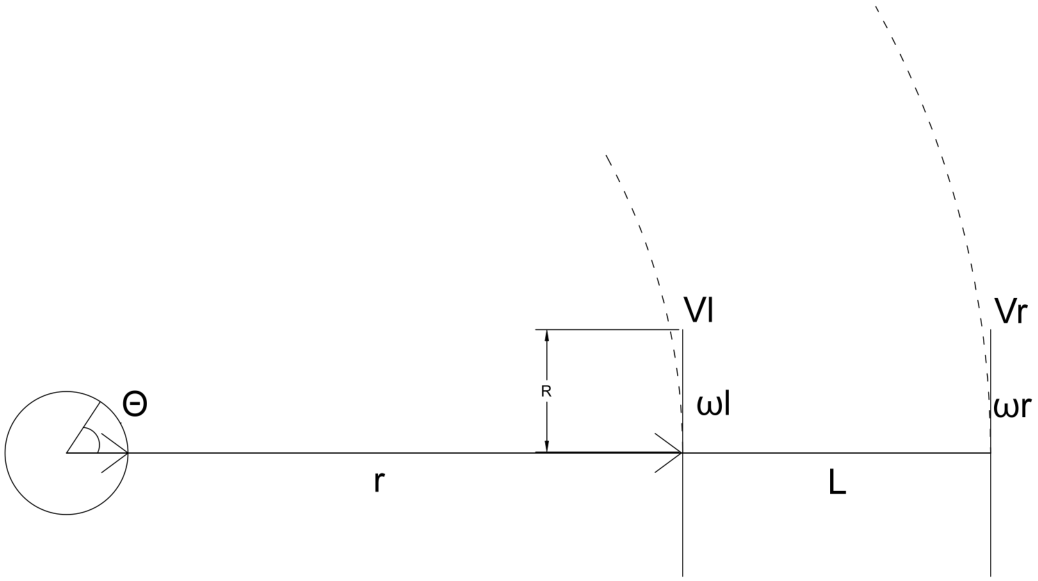

Figure 2: Kinematic model of the robot.

was desired to be controller was open loop, and error from either motor in terms of speed would result in the integration of error. The process that was used to overcome this issue is described below.

5.1.1 Design Exploration

During initial testing it was found to be difficult to achieve straight line driving on the platform. Through analysis of the individual wheel's speed over time, it was determined that while on average the individual speed's were very similar, bias' existed due to disturbances from construction. There are multiple root causes of this, being differing alignment on bevel gears, imperfect shafts, as well as differing motor characteristics. Overall, this caused significant turning while driving in a straight line.

A key observation made is as to now, the yaw of the robot, is being controlled in open loop. The obvious solution to this is to implement another controller around the yaw of the robot. Firstly, the following equation was derived through integration with a zero initial condition assumption.

$$\theta_r = \int \omega_r dt$$

$$\theta_l = \int \omega_l dt$$

$$\theta = \frac{R}{L}(\theta_r - \theta_l)$$

Given that there is a measure of the current angle, a PID controller can be defined to ensure that the desired angle is being outputted. The block diagram of the model is described in Figure 3.

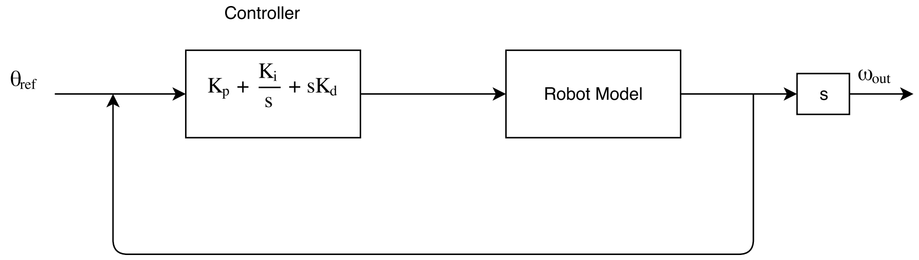

Figure 3: Robot yaw controller.

Given the correction rotation rate as the output of the controller, corrections to the yaw can be made by changing the desired yaw being used to calculate the individual wheel speeds. In order to maintain generality the reference angle is defined as follows where Δt is the time step between updates.

$$\theta_{ref} = \sum \omega \Delta t$$

This ensures that the robot would achieve an angle as if it were following the desired rate of rotation over time. It is important to note that the kinematic model decouples the tangential and rotational velocities, which allowed for minimal change to the previous implementation.

During testing it was noted that the same tuning of the controller parameters was unable to achieve stability in both the jumping and pole searching orientations of the robot. In order to combat this a gain schedule was implemented to adaptively change the tuning based on the orientation of the robot.

5.1.2 Key Features

By using the full kinematic model of the system it is possible to control both the speed as well as the angular velocity of the robot through a simple interface. Moreover, since any path can be generated using a tangential and angular velocity, the implemented design is theoretically capable of physically replicating any desired trajectory with little development effort. Given the modifications, and a full model of the system, theoretical guarantees can be made about the accuracy of the position of the robot at any time. However, it is important that all times the assumptions of the system must remain valid otherwise significant tracking error will occur as described later.

5.1.3 Testing

During end to end testing of the robot the assumptions that were made in order to formulate the controller were strained. One of the first issues that was noticed was that after impact from landing, there was a possibility that the bevel gears within the robot were decoupled from their shafts. That is, the torque being transferred from the motor was not reaching the wheels themselves. However, presented itself as one wheel spinning slower than the other, which resulted in confusion, and doubt in the controller implementation. Moreover when the orientation of the robot in the jumping state, there were instances where the wheel did not touch the ground resulting in the robot spinning in place. These issues show the sensitivity of the algorithm to the initial assumptions that were made. Without moving the encoders to measure the output shaft or purchasing an accurate IMU to measure the overall yaw, these issues cannot be solved in software.

Another result that was found from testing, is that when immediately commanding the motors to drive at a high speed, there was a high transient error that would not be corrected for in the length of the course (< 2 (m)). However, this error did not exist at low speed. This lead to the conclusion that a velocity profile would act as the simplest solution to prevent any error from existing. While this was the implemented solution, it is important to note that the yaw controller itself was tuned for somewhat slower response to minimize oscillation. By increasing the performance of the controller it would be possible to achieve a better overall response with a much faster ramp up time.

A final experiment indicated that the overall system was capable of positioning the robot better than 1 (cm) accuracy in terms of distance. Along with the high rotational accuracy, this meant that the half of the course before jumping over the wall was largely a function of human placement. Barring an orientation algorithm against the wall before jumping, localization within the course was not feasible and therefore not pursued. This prevented perfect pose within the course at all times which presented itself during the competition when the above assumptions were violated.

Another key victory of the design was that the robot was capable of achieving the above results with significantly damaged wheel shafts. The ends of the wheels had large horizontal displacement which would have presented as large disturbances to the controller. It is likely that because the selected tuning for the controllers were not aggressive the system was less sensitive to these disturbances and was able to remain stable and accurate.

5.1.4 Lessons Learned

One of the key lessons learned from the implementation described is to ensure that the overall design is less susceptible to measurement errors. In this case, by either using an IMU to measure the pose, or by having the measurement done on the output, the described controller could have prevented many of the issues that took several hours of debugging. Alternatively, by implementing localization code on the robot it would have been possible to achieve the same result without having to change the robot physically. However, this is prohibitive given the limitations of the Arduino mega.

In terms of control, a key take-away is that different orientation can cause significant dynamic changes to the system. and that significantly differing gains are required for stability.

Lastly, while the describe system achieves high accuracy and robustness a large amount of time was devoted to tuning the controllers to achieve these results. That is, a better method of tuning, perhaps by using the serial to set the tuning values would be beneficial.

5.2 Pole Location Algorithm

5.2.1 Key Features

5.2.2 Design Exploration

5.2.3 Testing

5.2.4 Predicted Results Vs. Actual

5.2.5 Lessons Learned

6 Competition Result Analysis

7 Conclusions

8 Forth Year Design Project Proposal

References