

VICTORIA UNIVERSITY, ENGR101

AVC Project Report

Jacob Beal

June 9, 2016

CONTENTS

1	Abstract	2
2	Introduction	2
2.1	Motivation	2
2.2	Problem	2
2.3	Scope	2
2.4	Motivation	3
3	Background	3
3.1	Theory	3
3.1.1	Closed Loops	3
3.1.2	PID	3
3.1.3	C and Raspberry Pi	4
3.2	Research	4
4	Methods	4
4.1	Equipment	5
4.2	Procedure	5
5	Results	5
5.1	Data	5
5.2	Interpretation	6
5.3	Discussion	6

6 Conclusion	6
6.1 Assessment	6
6.2 Recommendations	6
7 Bibliography	6
8 Appendix	6
8.1 Weekly Log	6

1 ABSTRACT

The Following Lab Report details how the application of fundamental engineering concepts could be used in such a way to construct and operate an Autonomous Vehicle. This report will detail the design, construction, and programming decision that we made, the final outcome of the project and the end results, and in particular issues that arose, and how this was resolved.

The Robot in question did not complete the entire maze, and did not complete the Third Quadrant.

2 INTRODUCTION

2.1 MOTIVATION

The purpose of the AVC is to demonstrate engineering skills taught in the ENGR101 Labs; and to work as a team to create an autonomous vehicle. This is used to solidify the teams understanding of fundamental Engineering concepts.

2.2 PROBLEM

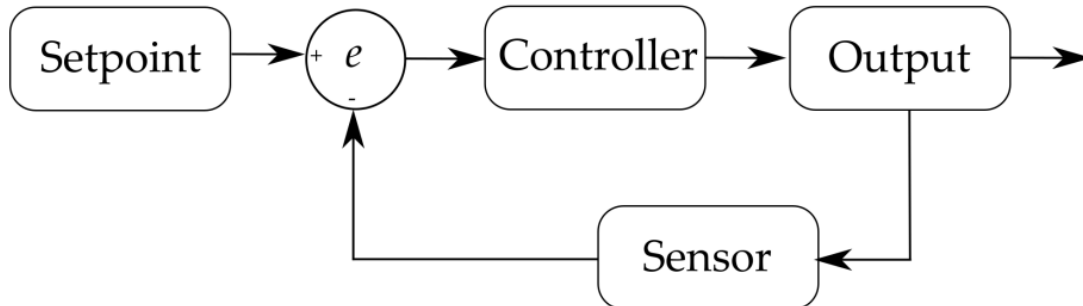
The challenge of a self-driving car is not immediately intuitive; the act of simply “following a line”. This involves computing the relative position of the vehicle compared to the line. The team employed other methods, by tracking the relative position over time, that will be detailed later in the report.

2.3 SCOPE

The scope of this project was using C/C++, the ENGR101 C Library and skills we have learned in the Labs for the project. This includes *PID* Error Correction (Proportional, Integral and Derivative Responses) to follow the white line and other techniques taught in lectures. We are also limited by our budget of \$100 for use at the “*part bazaar*” and the equipment that were provided at the beginning of the project, the Roboshield, motors and Raspberry Pi.

h

General Control System:

 $e = \text{calculate error}$ 

Example of a **closed loop** system: a system that responds to its current state.

Figure 3.1: A Closed loop diagram (Eldridge, 2016)

2.4 MOTIVATION

Understanding how an AVC works and solving a problem is important to the future, with autonomous vehicles increasing in prevalence, and being developed at break neck speed, it is of paramount importance that we understand how they work, and their limitations.

3 BACKGROUND

3.1 THEORY

3.1.1 CLOSED LOOPS

A closed loop system is a system which constantly loops, checking a physical variable. This variable is then computed in such a fashion as to deliver an output which is used in such a method to influence the physical variable the next time the loop is checked.¹

A closed loop system is useful in the “follow the line” section of the AVC where a difference in position of the line and the centre of the camera, which can be used to generate a difference in wheel speed. This difference in wheel speed can cause a turn in the vehicle

3.1.2 PID

PID stands for *Proportional, Integral and Derivative* response to a change in the conditions in a closed loop.

¹Eldridge, 2016.

A proportional response is the current conditions and how it responds to them immediately. i.e. the further the object is from the line, the stronger corrective response is.

An integral response checks over time to see whether the car is on one side too often, and compensates accordingly.

A derivative response is related to how quickly the proportional response changes, and therefore turns the car the other way when it approaches the line.

General PID Control System:

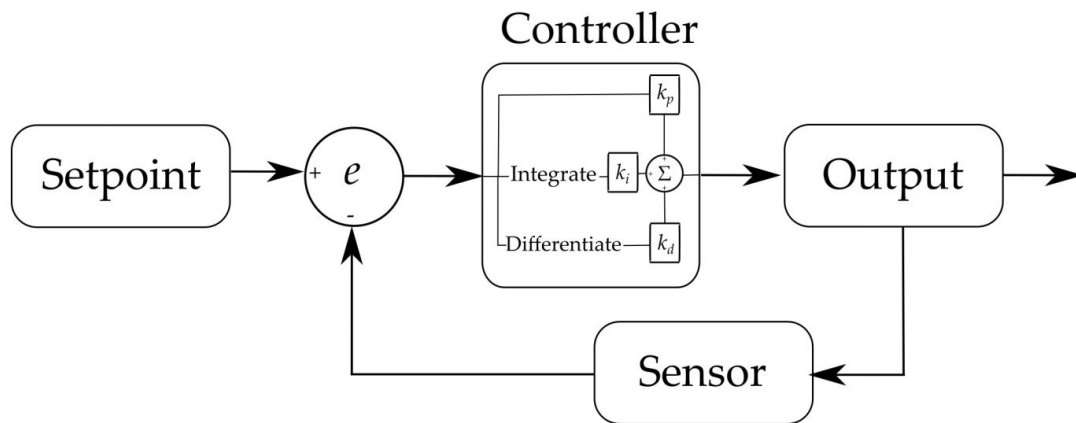


Figure 3.2: A PID system diagram(Eldridge, 2016)

3.1.3 C AND RASPBERRY PI

The project will be written using C++, with an imported C library for the functionality to control the motors and receive input from sensors.

3.2 RESEARCH

Tuning the PID system should be done in the specific order of Proportional, Integral then derivative responses. This is because the integral response can only be turned following the proportional response, so the offset at the end of the sinusoidal activity, and the derivative response merely acts as a dampener²³

4 METHODS

²Metso Minerals Industries, 2016.

³Douglas, 2012.

4.1 EQUIPMENT

The AVC Project was based on a Raspberry Pi and the Roboshield. The Autonomous vehicle was built with a 'PiCam' camera (provided), a pair of motors (provided), 2 'Sharp GP2Y0A41SK0F Analog Distance' sensors, a 3D printed chassis, and half a ping pong ball.

4.2 PROCEDURE

The team first designed the layout of the vehicle. The chosen design is based primarily on the given initial board. From this the motors were mounted at the front (See Fig 4.1), and the rear wheel, a semi-sphere, was chosen. A semi-sphere was chosen so that it had no bias in direction, nor resistance to change in direction, so that it could react better to change in direction.

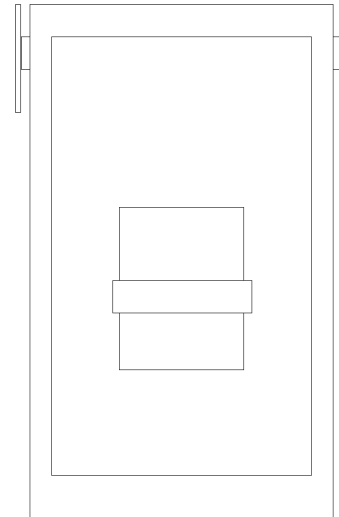


Figure 4.1: A top-down view of the design chosen for the AVC

5 RESULTS

5.1 DATA

The gate at the beginning now opens on request.



Figure 5.1: The Maze used to test the Autonomous vehicle

5.2 INTERPRETATION

5.3 DISCUSSION

The decision to not have a rear wheel, was made because it would give a greater ability to turn the vehicle and make the AVC more responsive to the change in environment.

The co-efficient of proportionality is much higher then derivative which is higher than the integral response, to better respond to the present situation.

The Camera appears to be broken on further expectation right now, so we can't show any results

The networking section, works flawlessly at the present, and causes the bridge to raise and lower as required.

6 CONCLUSION

6.1 ASSESSMENT

6.2 RECOMMENDATIONS

7 BIBLIOGRAPHY

REFERENCES

- Douglas, B. (2012, December 13). Pid control - a brief introduction. Retrieved from <https://www.youtube.com/watch?v=UR0hOmjaHp0>
- Eldridge, J. (2016, May 15). Control systems and image processing for avc. Retrieved May 15, 2016, from http://ecs.victoria.ac.nz/foswiki/pub/Courses/ENGR101_2016T1/LectureSchedule/ENGR101_Lecture20.pdf
- Metso Minerals Industries, I. (2016, May 14). Pid tuning tutorial. Retrieved May 14, 2016, from <http://www.expertune.com/tutor.aspx>

8 APPENDIX

8.1 WEEKLY LOG

WEEK 1

- | | | |
|---|----------|---|
| ✓ | ALL | Construct a Project Plan and Contact Julius |
| ✓ | Rhaz | Organise Meeting |
| ✓ | Jacob | Create GitHub and Establish Facebook Chat |
| ✓ | Andrew | Study SSH for Thursday Meeting |
| ✓ | Mitchell | Study Unit Testing |
| ✓ | Theo | Make a general plan for the chassis |
| ✓ | Julius | Get in contact with the group |

WEEK 2

- ✓ ALL Discuss ideas and start on the AVC Code
- ✓ Rhaz Keep up to date on weekly tasks
- Jacob Robot moving in straight line — appears to work, but haven't tested with a battery yet
- ✓ Andrew Complete the README.md
- ✓ Mitchell Look into the networking code, and how it works
- ✓ Theo Figure out how to use the CAD software
- Julius Show Up — Not turned up — Postponed

WEEK 3

- ✓ ALL Progress update with team members and begin writing progress report
- ✓ Rhaz Robot Opens Gate
- ✓ Jacob Robot Opens Gate
- ✓ Andrew Robot Opens Gate
- ✓ Mitchell Update the README create Hardware updates and assist in the hardware efforts.
- ✓ Theo Create a non-powered wheel case and battery case for the robot
- Julius Show up and contribute to team meetings

WEEK 4

- ✓ ALL Progress update with team members
- ✓ Rhaz Quadrant 1 completed
- ✓ Jacob Quadrant 1 completed
- ✓ Andrew Quadrant 1 completed
- ✓ Mitchell Assist in hardware and pre-reading for quadrant 2
- Theo Attach sensors to robot so it can detect the maze and obstructions — postponed as code not up to that section yet.
- Julius Contribute in some way

WEEK 5

- ✓ ALL Progress update with team members
- ✓ Rhaz PID system functioning
- ✓ Jacob PID system functioning
- ✓ Andrew PID system functioning
- ✓ Mitchell Bug fixing and tinker with PID constants
- ✓ Theo Make battery clip easier
- Julius Contribute in some way

WEEK 6

	ALL	Quadrant 3 complete
	Rhaz	Quadrant 3 complete
	Jacob	Quadrant 3 complete
	Andrew	Quadrant 3 complete
	Mitchell	Quadrant 3 complete
✓	Theo	Attaching sensors to navigate Quadrant 4
	Julius	Contribute in some way