Automatic Braking System Using Fuzzy Logic Controller

Anthony Trinh, Dylan John Sutherland, George David Nicholson, Julian Ryan, Lachlan Foy, Norbert Pop

Introduction

When travelling long distances by car, driver fatigue becomes a major risk factor. Driver assist systems such as cruise control attempt to reduce this risk by automatically managing simple tasks for the driver. Cruise control is beneficial for long driving periods on highways and freeways where a constant speed needs to be maintained but is unable to manage more complex traffic situations on its own. An Adaptive Cruise Control (ACC) system uses sensors to complement basic cruise control so that car speed can be automatically adjusted based on cars and obstacles ahead. We have created an automatic braking controller which reduces car speed when approaching a static obstacle at high speed or getting too close to a leading car. The system uses fuzzy logic to map current vehicle speed and distance from an object in front of the car to a percentage of braking pressure. The effectiveness of our system was modelled though a software simulation graphing the speed reduction and break pressure strength when encountering an obstacle.

Design of the Fuzzy Inference System

In order to create our automatic braking controller we first had to identify the inputs which would be processed by our fuzzifier. We decided our system would be radar based because it works in all weather conditions and can simultaneously provide information on both the distance and speed of neighbouring vehicles. The two inputs, controlled car speed and distance from lead car were chosen to limit the initial scope of our fuzzy controller. These values would be used to calculate a percentage braking force for the car.

The next problem we had to address was identifying the membership functions for the inputs and outputs. Initially we selected three linguistic classifications for each variable. The number of membership functions was increased to five for the vehicle speed to address the typical speed limits which apply on Victorian roads. The output braking power was assigned four functions. The relationships between linguistic variables are shown in the table below

	very_close	**close**	**far**
very_slow	hard	soft	reduce
slow	hard	soft	reduce
medium	hard	soft	nil
fast	hard	hard	nil
very_fast	hard	hard	nil

The relationships in the table can also be written as a set of Fuzzy Propositions:

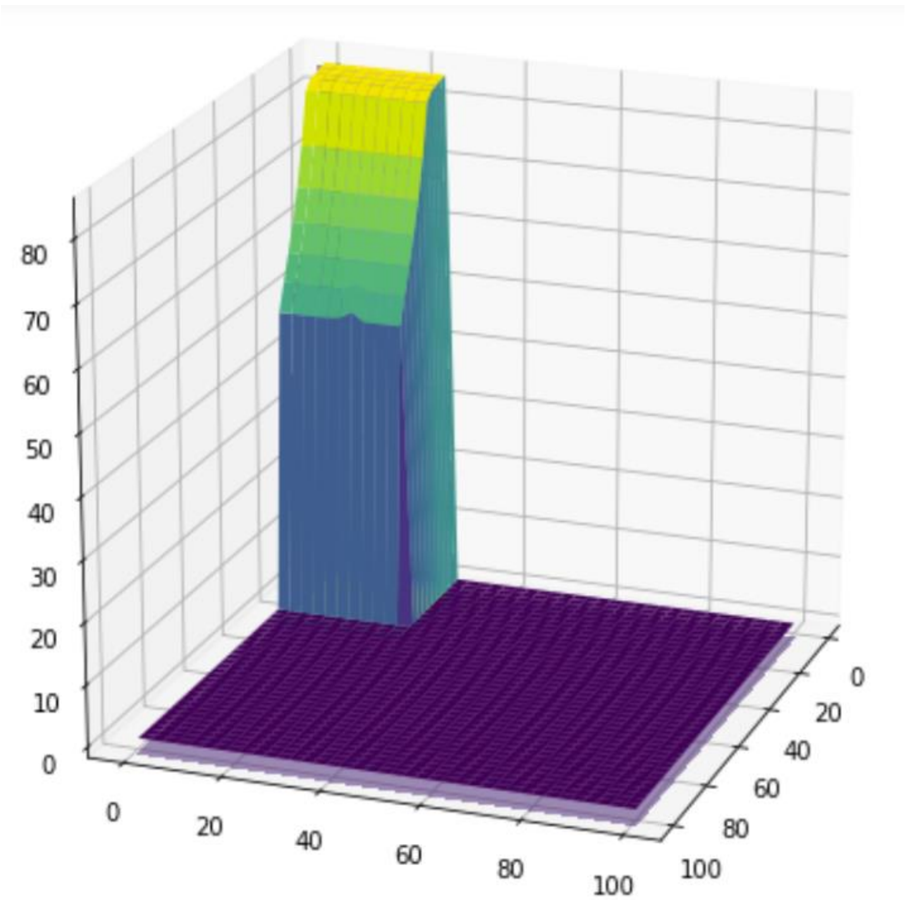
- IF distance is very_close THEN brake hard

- IF distance is close AND speed is slow THEN brake soft
- IF distance is close AND speed is medium THEN brake soft

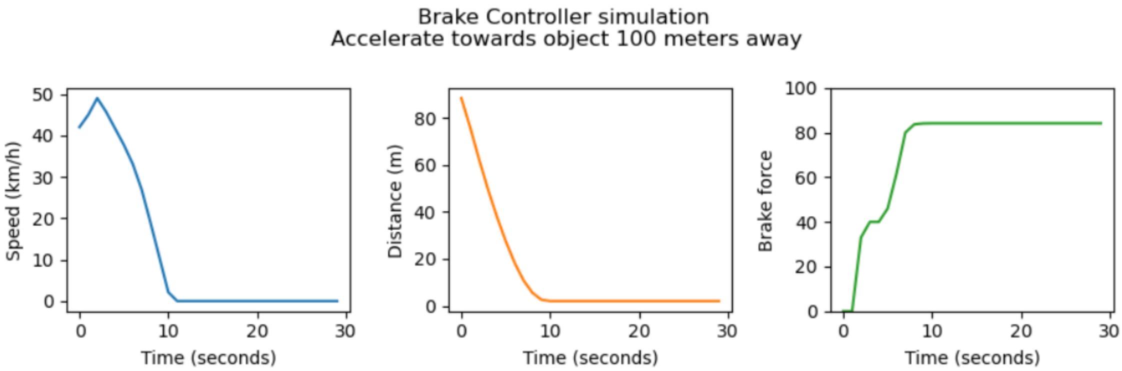
- IF distance is close AND speed is very_slow THEN brake nil/do nothing
- IF distance is far AND speed is medium THEN brake nil/do nothing
- IF distance is far AND speed is fast THEN brake nil/do nothing
- IF distance is far AND speed is very fast THEN brake nil/do nothing

- IF distance is far AND speed is very slow THEN brake reduce
- IF distance is far AND speed is slow THEN brake reduce

After constructing the rules they needed to be implemented into code. We coded our rules in python using the sci-kit fuzzy library though implementation was also done using the Simpful library during the prototyping phase. We used examples from both of these library's documentation in order to inform the construction and visualisation of our outputs. One of our main challenges with implementation was restricting our braking controller to only react in situations where a collision was likely to occur. In other words we wanted an output of zero, or close to zero, braking pressure until both the speed and distance reached above a certain threshold. The output space visualises this output and was a significant tool during testing and adjustment of the rules and membership functions.



Additionally, we investigated how our model might perform if it was part of a car computer system. This was done by creating a basic simulation which created a vehicle going at a constant speed in front of our vehicle. The controlled vehicle accelerated towards the leading vehicle and was brought to a stop by a soft brake pressure transitioning into a hard break pressure as the distance between the vehicles grew smaller. The output of this simulation is shown in the figure below.



Limitations and Constraints

Our automatic braking controller has a number of improvements which could be made to it. Foremost among these would be ensuring the controller is adjusted to the laws of physics. This would require a significant investigation into the forces which effect braking and deceleration. Furthermore, a recognition system would need to be created for driving surfaces so that the brakes would engage earlier on surfaces with less traction. The fuzzy inference system is static with the rules being based on a guess so to obtain the most effective system extensive trial and error is required. An automatic braking controller created by using a deep learning algorithm would produce a more customised output and ultimately be more accurate.

Contributions

Over the time our group have spent working on creating the fuzzy logic controller we have had several meetings in a Microsoft Teams group, as well as discussions through messages. During our initial meetings we each shared our views on which feature we wanted to focus on and identified the need to extend our knowledge of fuzzy logic before we could begin to create the code.

Some members took more time to understand the constraints of the task and the concepts of fuzzy logic, so they spent more time engaging in their own research. We ensured that members who were having difficulty were provided with support by others who had a stronger understanding. This was done through one-on-one meetings where we discussed questions and explained the workings of our code to them. Additionally we ensured they still had an opportunity to contribute by having them help out in areas such as presenting our finding.

As the team captain I oversaw the progress of our group. I compiled the different parts of the report together, worked on testing and improving the fuzzy controller and ensured that everyone in the team was contributing and available for meetings. Lachlan produced the first basic working code with membership functions in sci-kit fuzzy and managed both Norbert and Dylan ensuring the code they produced was compatible. Norbert conceived the idea of demonstrating our fuzzy controller using a simulation and created the code for it as well as engaging in discussions about the code. Anthony contributed a lot to the early solution design, producing the first draft of our rule base and ensuring the theory of fuzzy logic was demonstrated in the code and final solution. Anthony also assisted in the recording and production of our presentation. George, Anthony and Julian collaborated together in a sub-group focused on the effective communication of our solution. This included creating the structure and documents for our presentation and reports. George actively engaged in our team discussions asking questions to ensure we were on the right track as a team and helping to keep team morale up. He produced the preliminary report and assisted in the explanation of fuzzy logic concepts and automatic braking systems. Julian played roles in a number of areas working on producing the early code, contributing to the report and taking charge of the presentation.

References

[1]E. Bassey and K. Udofia, "Modelling of automatic car braking system using fuzzy logic controller", *Nigerian Journal of Technology*, vol. 38, no. 4, p. 1021, 2019. Available: 10.4314/njt.v38i4.27.

[2]Y. Yazicioglu and Y. Unlusoy, "A fuzzy logic controlled Anti-lock Braking System (ABS) for improved braking performance and directional stability", *International Journal of Vehicle Design*, vol. 48, no. 34, p. 299, 2008. Available: 10.1504/ijvd.2008.022581.

[3]N. Basjaruddin, D. Saefudin and I. Nugraha, "Developing Adaptive Cruise Control Based on Fuzzy Logic Using Hardware Simulation", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 6, 2014. Available: 10.11591/ijece.v4i6.6734.

[4]S. Alrubaee, "Automatic Brake System using Fuzzy Logic Contents", 2019. Available: 10.13140/RG.2.2.14398.95043.

[5]J. Naranjo, M. Sotelo, C. Gonzalez, R. Garcia and T. Pedro, "Using Fuzzy Logic in Automated Vehicle Control", *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 36-45, 2007. Available: 10.1109/mis.2007.18 [Accessed 3 April 2021].

[6]"skfuzzy 0.2 docs — skfuzzy v0.2 docs", Pythonhosted.org. [Online]. Available: https://pythonhosted.org/scikit-fuzzy/. [Accessed: 26- Mar- 2021].

[7]S. Spolaor, C. Fuchs, P. Cazzaniga, U. Kaymak, D. Besozzi and M. Nobile, "Simpful: A User-Friendly Python Library for Fuzzy Logic", International Journal of Computational Intelligence Systems, vol. 13, no. 1, p. 1687, 2020. Available: 10.2991/ijcis.d.201012.002.