Course: AI, U10M12018.01 Teacher: Li Xiaoan

Project Report

New realization and ideas about Braitenberg Vehicles

Wang Yunze 2018302461

2020-1-8

Abstract

As a classic model, Braitenberg Vehicle has been studied for long in the history of artificial intelligence. However, as the development of AI progresses and the demand of capabilities of robotic designs continues to raise, the traditional Braitenberg Vehicle requires an update. This report presents new realization and inspirations on Braitenberg Vehicles, and proposes extrapolated designs based on the Braitenberg Vehicles.

The related files of this project is hosted on GitHub:
https://github.com/Firefox2100/Braitenberg_Vehicle_AI_Project

Index

1 Introduction

1.1 Background on Braitenberg Vehicle

A Braitenberg vehicle[①] is a concept conceived in a thought experiment by the Italian-Austrian cyberneticist Valentino Braitenberg. It's essentially a robotic platform consists of several sensors and motors, which the motors are directly controlled by the sensors in the simplest vehicles.[②] As the simplest implementation of goal-oriented agents, it serves as a classic model of embodied reflex agents. Based on different configurations and connections between the sensors and the wheels as actuators, it can peform different behaviors named fear, aggression, liking, and love, which correspond to biological positive and negative taxes.

1.2 Different Braitenberg Vehicles

The simplest Braitenberg Vehicle is a platform with only one sensor and one (set) of actuators. The input of that sensor directly stimulates the actuators, so it can only perform certain behaviors: accelerate, decelerate, and halt. It possesses no control of its own course, for ideally, it can only move in one dimension. If for some reason it deviates from previous course, it will stay on the new one. The basic behavior can be described as the tendency or avoidance of certain criteria (heat, light, object, etc.).

More complicated Braitenberg Vehicles have two sensors and two actuators. Each sensor is connected to one actuator, and has positive or negative taxes on the actuators. This more complicated design allows Braitenberg Vehicles to be able to steer, thus to perform multiple other actions. The basic model can be concluded in the following 4 types: fear, aggression, love and exploration.

The Fear vehicle connects the leftmost sensor to the left wheel motor and the rightmost sensor to the right wheel motor. If the sensor detects light, the motor on the same side will accelerate, causing the vehicle to speed up and steer to the opposite direction. To human observer, this would appear as if the vehicle speeds up and escape from the light.

The Aggression vehicle is similar to Fear, except that the sensors are wired to motor at the opposite end of the vehicle. If the sensor detects light, the motor on the other side will accelerate, causing the vehicle to speed up and steer to the direction of the light. That is, Aggression moves towards the light. The closer the vehicle gets to the light, the stronger the increase in speed of the motors, until the vehicle speeds through the light. To human observer, this would appear as if the vehicle is trying to ram the light.

The Love vehicle is similar to Fear, except that the sensors decrease the forward speed of the motor to which they are connected. If the sensor detects light, the motor on the same side will decelerate, causing the vehicle to slow down and steer to the direction of the light. Like Aggression, Love moves towards the light. Unlike Aggression, however, the closer Love gets to the light, the slower it moves. As a result, Love moves towards the light until it reaches the perfect distance and goes into a halt.

The Exploration vehicle has the same crossed wiring of Aggression, except that the sensors decrease the speed of the motors to which they are attached. If the sensor detects light, the motor on the other side will decelerate, causing the vehicle to slow down and steer to the opposite direction, until it's turned away. Like Fear, Exploration avoids the light. However, Exploration slows down in light areas, almost as if it is cautiously exploring the light.

2 State of the art

2.1 The current research about Braitenberg Vehicle

The latest update on the research papers related to Braitenberg Vehicle are:
(The names of the papers are not ordered, and only papers indexed by IEEE after 2019 are listed)

- *Using Reinforcement Learning to Attenuate for Stochasticity in Robot Navigation Controllers*, James Gillespie;[3]
- *Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot*, Dony Hutabarat[4]
- *A chemosensory navigation model inspired by the neural odour processing mechanism in cockroaches*, Alejandro Pequeño Zurro;[5]

It would appear that the current research direction on Braitenberg Vehicle are to use other sensory input (LiDAR is 2D input, traditional proximity sensors are 1D; and chemosensory is another area other than photoelectricity), or to implement more complicated neural network on the system.

2.2 The current application of Braitenberg Vehicle

The most common application of Braitenberg Vehicle, both the traditional "simple" form and the more complicated variant, is to use the principle in robot navigation. The representing papers are:

- *Development of an intelligent and distributed low-cost platform for marine observations*, Christoph Tholen⑥
- *Against the flow: A Braitenberg controller for a fish robot*, Taavi Salumäe;⑦

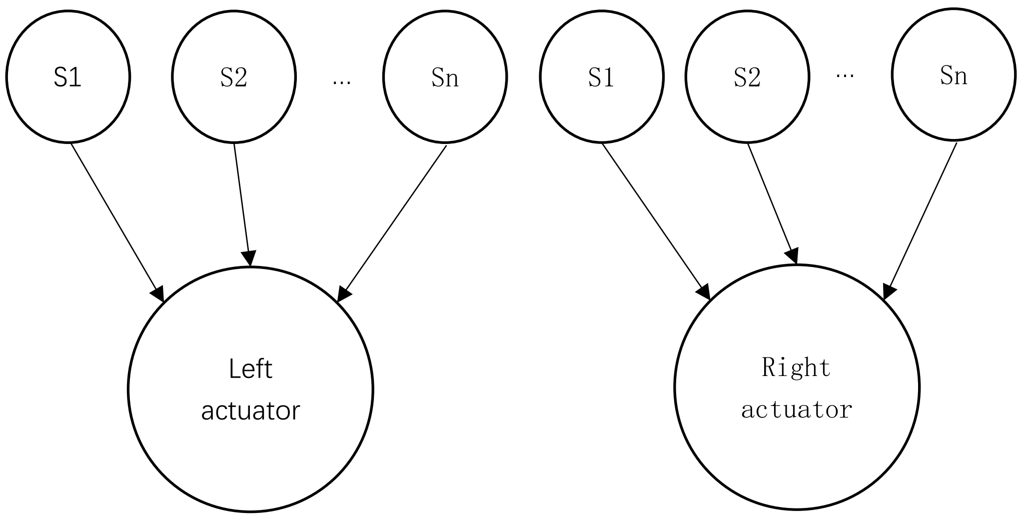
3 Neural Network of Braitenberg Vehicle with multiple sensory input

3.1 Multiple sensory input and the need for it

The traditional Braitenberg Vehicle has only two sensors (for the more complicated version, or Vehicle 2), and two actuators. However, in real-life, the perception of the world is much more complicated than two sensors can handle. So, the implementation of Braitenberg Vehicle in physical world calls for multiple sensory input. In general, there are two different types of multiple sensory input:

3.1.1 Multiple types of sensory input

For a Braitenberg Vehicle, it could have sensory input of different types. The information of the real world may be too complicated for one single type of sensors to perceive. For example, in the requirement of the Koan, there has to be at least two different types of sensors to detect obstacles and other vehicles. Multiple types of sensory input make up multiple nodes in each layer, however the total number of the layers of the NN remains unchanged, given there're two sensors for each type. In representation:



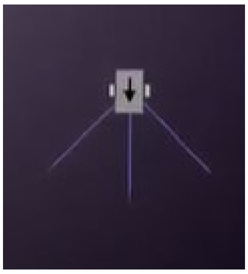
In this model, multiple sensors control one actuator together. This is relatively easy to implement, and can be described as:

$$O = (S_1 w_1 + S_2 w_2 + \cdots + S_n w_n) + c$$

Where O is the output of the actuator, S_n is the sensory input of the connected sensor n, and w_n is the weight assigned to sensor n. c is a constant that can be considered as the "offset" of the output from 0.

3.1.2 Multiple sensors for each type

In the traditional Braitenberg Vehicle implementation, ever sensor corresponds to one actuator, and every actuator correspond to only one sensor for each type. However, as a variant an improvement in real life of Braitenberg Vehicle, there could be more sensors of each type than actuators. This provides a more thorough perspective of the surroundings, and can be seen as a discrete and simplified view of 2D/3D modeling. For example, a Vehicle can have two sensors for distance at its side, and one at its front like this:


⑧

Now the sensor in the middle should be connected (directly or indirectly) to both of the actuators. However, this sensor does more than only to slow down the motors (when programmed to avoid obstacles). If both the left and right sensors detect coming object, it could be either a wall or a doorway. If, in bad luck, the Vehicle is pointed directly toward the doorway, the left and right motors will decelerate at the same rate, causing the Vehicle to stop in front to the doorway. If there's a wall, the Vehicle will do the same. However, there're better solutions for these conditions: if it's a doorway, the Vehicle should go through it without (too much) deceleration; and if it's a wall, the Vehicle should attempt to turn away from it's facing direction to go along with and away from the wall. The third sensor in the middle can distinguish these two conditions and avoid halting of the Vehicle. If we only consider if a tax is positive or negative, we can have the following table:
(0 and 1 at the sensor stands for whether there's an obstacle detected, and * means unchanged, while – means decelerate)

Left sensor	Middle sensor	Right sensor	Left motor	Right motor
0	0	0	*	*
0	0	1	-	*
0	1	0	*	-
0	1	1	-	*
1	0	0	*	-
1	0	1	*	*
1	1	0	*	-

1	1	1	*	-

In the above table, we assume that the Vehicle turns right to avoid frontal collision with an object. Obviously, for both left and right motor, this is linear separable. So, the extra sensor improved the performance (by avoiding slowing down at the doorway) and fixed a probable halting issue (when facing directly to a wall), but the computational cost won't increase too much.

3.2 Multi-layered Neural Network for Braitenberg Vehicle

In the traditional Braitenberg Vehicle, the sensors are directly connected to the actuators. However, when the model becomes more complicated, the sensory input and expected output may become linear inseparable. This would require the Neural Network to be modified into NN with hidden layers. And when the sensory data is too complicated (for example, when using LiDAR to get a 2-Dimension plotting of the circumstance rather than multiple discreate values with proximity sensors), or when the input data contains noise, some specific type of Neural Network may be used to solve a specific problem. [9][10]

Because of the limited computational power an embedded MCU can provided, in my research I didn't focus on this area. However, as an embedded programmer that has some experience on Neural Networks, it's easy to imagine that for a system using LiDAR or other form of 2D/3D input, an early but effective model – DnCNN is suitable for noise dampening and pre-processing the data for control NN.

4 Hardware realization for Braitenberg Vehicle with multiple sensory input

4.1 Requirement analyzation

To further the research of Braitenberg Vehicle, an actual hardware platform is required. The current platforms that can be modified to perform as a Braitenberg Vehicle are mostly not designed as Braitenberg Vehicles, so they have extra input/output ports, slots for other attachments like mechanical arms, yet not enough resource for extended design of Braitenberg Vehicles. Based on this I decided to design a low-cost platform specialized for Braitenberg Vehicles. The design needs to have the following traits:

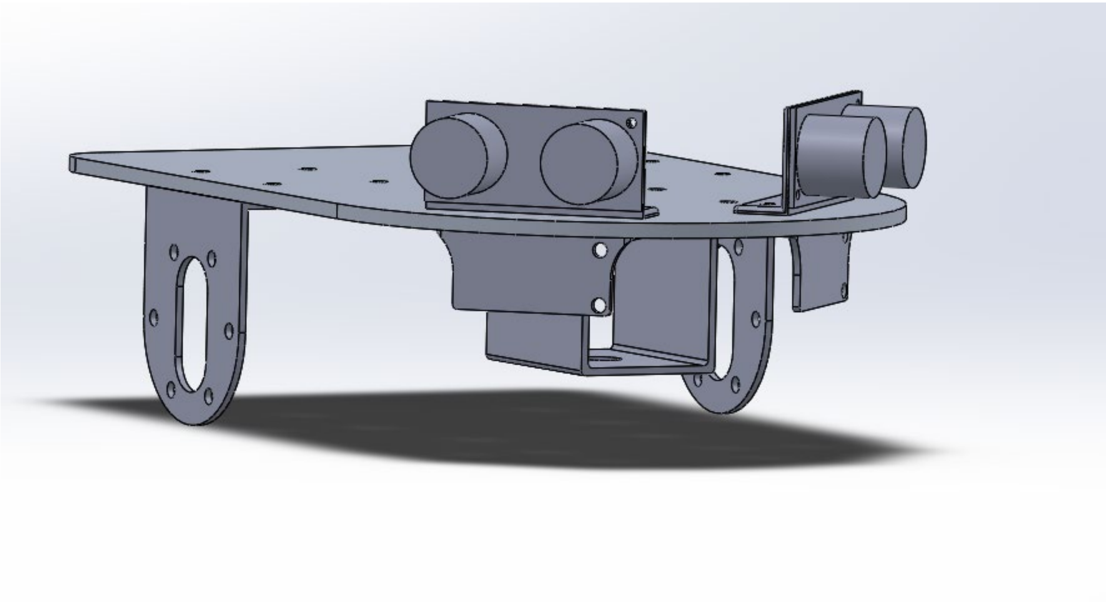
1. Low cost and robust. This design needs to be able to be easily put to mass production with minimal modification, and used by other researchers on this field. If the design is too expensive or too delicate, the final product won't be able to suit different research environments.
2. Extendable. For different purposes this system may need to interface with different

sensors, even actuators. Certain ports and interfaces must be pre-configured and designed for easy integration with other parts.

3. Specialized. Braitenberg Vehicle is a specific type of robot, which doesn't require it to perform in hazardous/complicated environments, nor does it need complicated method to communicate with the researcher (for it is automated). In researches this design is only intended to be a proof of concept, a prototype, so the resources are better used at specialized aspects.

4.2 Mechanical design

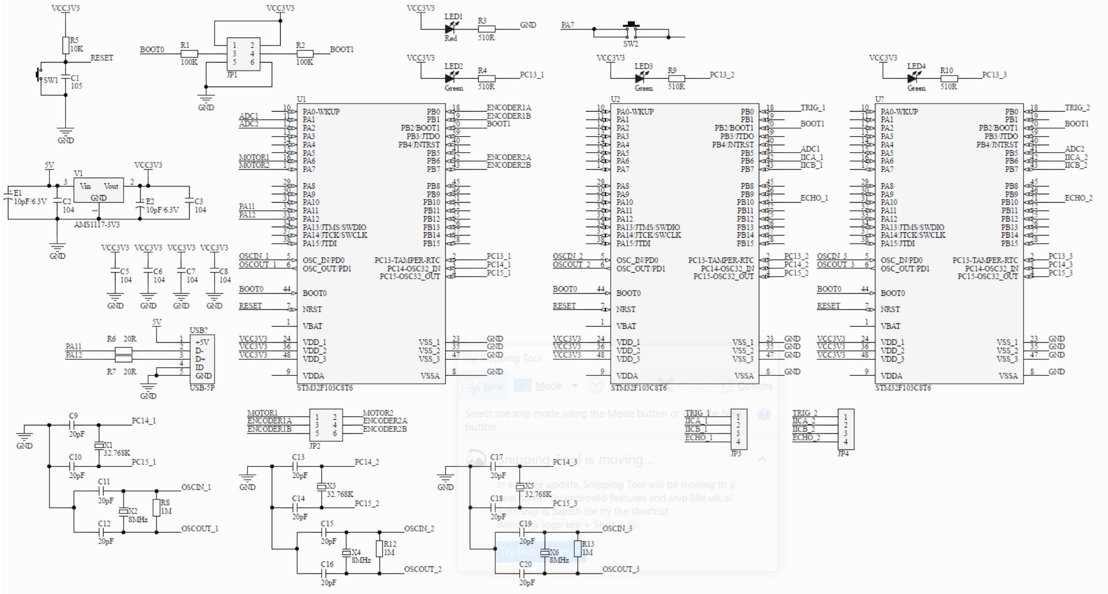
The major design for mechanical part is show in the screen shot below. The detailed CAD file is built with SolidWorks 2019 Pro SP4, and can be found in the GitHub repository.



As can be seen above, this platform has two motor wheels and one omnidirectional wheel for balance. There are two ultrasound sensors above the main chasse, and two light sensors below the chasse. The rest of the space is for the MCUs, PCB boards, battery pack, etc. and is not marked on the CAD file due to the need to adjust their positioning for balancing purpose. The LEDs around the chasse are also not marked, for only practical experiments can determine their positions and brightness.

4.3 Electronic design

The schematics for the core PCB design is drawn with Altium Designer 20.0.1. It's shown in the following screenshot:



The schematic file is not given directly because of copyright and license issues. The PDF version of the schematic is provided instead. As can be seen, there're 3 STM32F103C8T6 MCUs in the design. One is used for motor control and read the encoder feedback, because the resources on STM32F103C8T6 is limited, reading 2 encoders and controlling 2 motors uses up most of the resources. The other 2 MCUs are used to process different sensory input for each actuator. The list of major IC/MCU/modules used are:

Name	Type	Amount
STM32F103C8T6	Micro Controller Unit	3
GY-302 (BH1750FIV)	Illumination sensor	2
HY-SRF05	Ultrasound distance sensor	2
TB6612FNG	Motor driver	1
MG530	Motor encoder	2

4.4 Firmware design

The firmware is developed with Keil µV5.16.0. Be ware that if you want to try to compile the project on your own, you may need a full license for Keil to be able to compile Assembly code as a function in C. The parameters are preset to simulated suitable values, they may not be optimal in an actual physical platform. This design is based on the template provided by OpenEdv[11]. The core functions can be simplified as:

```
while(1)
  {
        adc1=Get_Adc_Average(ADC_Channel_1, 10);
        adc1=Get_Adc_Average(ADC_Channel_2, 10);
        speed_left=calc_speed(adc1);
```

```
        speed_right=calc_speed(adc2);
        move_left(speed_left);
        move_right(speed_right);
        delay_ms(150);
    }
```

This is the core part of main function in the firmware for the MCU that controls the wheels. It first reads in the voltage level provided by the other two MCUs in the form of PWM waves, then calculates the proper speed for the wheels, and calibrate the wheel speed to the desired value.

```
int calc_speed(u16 adc)
{
    int speed = MAX - adc / RATIO;
    return speed;
}
```

This is the function to calculate the speed for each wheel from the input taken from other MCUs.

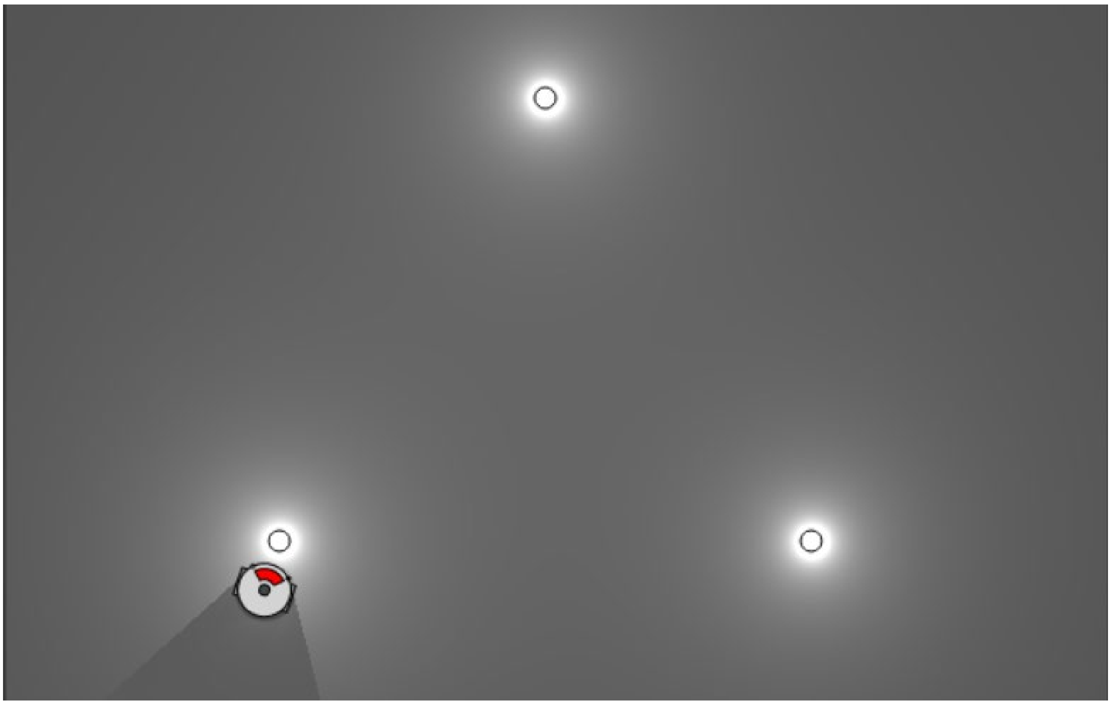
```
void output(u16 illuminance, float distance)
{
    u16 pwmval;
    pwmval = (illuminance*ILL+distance*DIS)*RATIO;

    TIM_SetCompare2(TIM3,pwmval);
}
```

This is the function used by MCUs that process sensor inputs to calculate the proper duty ratio of the PWM output.

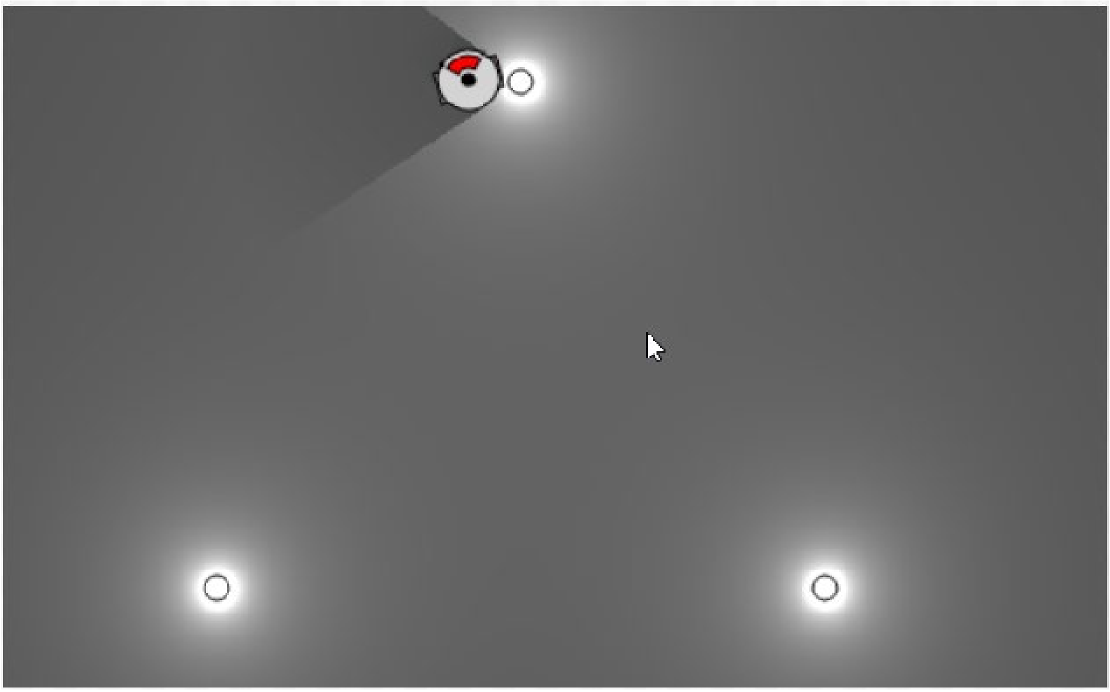
5 Conclusion

5.1 Simulation

Due to copyright problem, I can't provide the software simulation with Keil and Proteus. However, all the schematics and firmware project are ready for simulation, simply import them into the software should do. The behavioral simulation is accomplished with a JS script developed by Harmen de Weerd[12]. I modified his source code, and used the script to find an optimal parameter set for my design. The simulation is shown as follows:



When the Vehicle is programmed to go after the light in “love” model (chase after the other Vehicles)



When the Vehicle is programmed to avoid the light in “exploration” model (avoid obstacles).

5.2 Further extend the model of Braitenberg Vehicles

The classic model of Braitenberg Vehicle has many distinguishing features: the sensors are directly connected to one actuator, uses wheels, etc. However, is it possible to further extend the model of Braitenberg Vehicles?

5.2.1 Braitenberg Vehicles without wheels

In real life, there're many conditions that requires other actuators than wheels. For example, tracks. When dealing with rough terrain or muddy roads, wheels are not as effective as tracks. However, directly replacing the wheel system on a Braitenberg Vehicle might work, but this method can't utilize the unique feature with tracks: it's capability to do spot turns. The configuration of the controlling NN with tracks should allow either or both sides of tracks be able to move in reverse, so that it can avoid obstacles in tight spaces.

Another interesting system is water propeller system. The Braitenberg Vehicle model may also be used on automated ship vessels. In such systems, the Vehicle turns not because of the speed differentiation of different wheels (or propellers, in this case), but because of the steering of the rudder. This means the system can't maneuver in tight space, so would require the controlling system to react in advance. Also, in this case, the actuators are no longer two same parts, but a propeller and a rudder, one for acceleration and one for steering. This changes the fundamental structure of the Neural Network.

5.2.2 Braitenberg Vehicles and clustering algorithm

The driving method of Braitenberg Vehicle indicates that it always tried to go in the direction where density gradient (of the determine information) is pointing. This is very similar with some clustering algorithm. For example, mean shift[13] is a typical algorithm which uses density gradient to find the center of a cluster. However, the traditional method used by Braitenberg Vehicle is simply find the density gradient at a specific point (where the sensor is located), which is neither effective nor robust. By applying some of the clustering algorithm to Braitenberg Vehicle, it's expected that the performance of the Braitenberg Vehicle may improve. I will investigate this aspect further to find out more.

Meanwhile, the principle of the Braitenberg Vehicle can also be applied to clustering algorithms. In algorithms like mean shift, the shifting target circle are independent – they do not affect each other. However, in real life, it's often the case that once a space is occupied (or a subset of data points are determined to belong to one cluster), no other item may occupy that same place (or no other clusters can have these data points). The common way is to use evaluation functions to determine who will occupy which space, or which cluster are these points closer to. However, it may be possible to allow the clusters to "interact" with each other and "compete for" data points, just like the Braitenberg Vehicles when they perform certain tasks. This will be hard to evaluate, but still possible.

6 Personal evaluation and contribution

Technically speaking, I did everything in this project, from information gathering, to modeling, programming, and simulating. There wasn't much of "group work", as one of my teammates is not very motivated to help, and the other one doesn't have enough background on this to

help.
If I had more time, I would have further investigated some interesting aspects mentioned above, like using data clustering technique to power a Braitenberg Vehicle. However, this would take time, and in this assignment (which happens to be in the same time with many other projects and exams) time is not in massive supply. I estimate that for me to actually accomplish something to contribute to the advancement of data clustering and robot guidance (I have worked on data clustering in image processing and video tracking before), it would take at least two months. So instead I focused on hardware platform realization of Braitenberg Vehicle, and designed a low-cost prototype. In my spare time, I'll continue to look into these questions, and hopefully this isn't just a dead end for me, and one day I'll be able to accomplish something on this.

7 References

---

① V. Braitenberg, Vehicles. Experiments in synthetic psycology. The MIT Press, 1984.
② Wikipedia: https://en.wikipedia.org/wiki/Braitenberg_vehicle
③ J. Gillespie, I. Rañó, N. Siddique, J. Santos and M. Khamassi, "Using Reinforcement Learning to Attenuate for Stochasticity in Robot Navigation Controllers," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 705-713, doi: 10.1109/SSCI44817.2019.9002834.
④ D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 197-202, doi: 10.1109/ICTS.2019.8850952.
⑤ A. P. Zurro, I. Rañó and D. Shaikh, "A chemosensory navigation model inspired by the neural odour processing mechanism in cockroaches," 2019 IEEE International Conference on Cyborg and Bionic Systems (CBS), Munich, Germany, 2019, pp. 17-23, doi: 10.1109/CBS46900.2019.9114486.
⑥ C. Tholen, L. Nolle, T. A. El-Mihoub, O. Zielinski and J. Wellhausen, "Development of an intelligent and distributed low-cost platform for marine observations," OCEANS 2019 - Marseille, Marseille, France, 2019, pp. 1-6, doi: 10.1109/OCEANSE.2019.8867466.
⑦ T. Salumäe, I. Rañó, O. Akanyeti and M. Kruusmaa, "Against the flow: A Braitenberg controller for a fish robot," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, 2012, pp. 4210-4215, doi: 10.1109/ICRA.2012.6225023.
⑧ YouTube video: Robot Simulator for Braitenberg Vehicles and Other Small Robots (https://www.youtube.com/watch?v=W_X07gZuqog)
⑨ J. Gillespie, I. Rañó, N. Siddique, J. Santos and M. Khamassi, "Using Reinforcement Learning to Attenuate for Stochasticity in Robot Navigation Controllers," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 705-713, doi: 10.1109/SSCI44817.2019.9002834.
⑩ D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 197-202,

doi: 10.1109/ICTS.2019.8850952.

[11] http://www.openedv.com/

[12] http://www.harmendeweerd.nl/braitenberg-vehicles/

[13] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," in IEEE Transactions on Information Theory, vol. 21, no. 1, pp. 32-40, January 1975, doi: 10.1109/TIT.1975.1055330.