Introduction

Novel coronavirus disease (COVID-19) was firstly identified in 2019 in Wuhan, China and it spread all around of the world in a short amount of time. Currently, more than 500 million cases and 6 million deaths are reported. Luckily, some vaccines have been introduced. Countries are hoping to have a herd immunity after a specific percentage of vaccination. However, there is not enough vaccine available for every country to vaccinate everybody.

The point of this story is creating a herd immunity and controlling it with limited amount of vaccine. In this homework, you will create a herd immunity by vaccinating people. Your aim is to vaccinate 60% of the people in your country and maintain that percentage. In this way, you will protect 100% of the people from the virus with limited amount of vaccine.

You vaccinate people to get them immune to the virus. Assume that you have conducted the required research on the relation between the vaccination rate and birth-death rates as well. Hence, you are able control the vaccination rate in terms of percentage per day. In that manner, setting vaccination rate to some $x\,\frac{\%}{day}$ means that the percentage of the vaccinated people will be increased by x; namely, if the current percentage were p then the percentage after a day would be $p + x$. On the other hand, your research has revealed that there exists a failure rate for the vaccine owing to the nature of it. Assume that the failure rate is proportional to the percentage of the vaccinated people. In other words, the more people get vaccinated, the more rapidly failed your vaccination will be. You can also measure that rate in terms of percentage per day. Therefore, there is an effective vaccination rate which is the difference between the vaccination rate controlled by you and the failure rate. Your task is to provide proper control for the vaccination rate so that the percentage of the vaccinated people is to come to the equilibrium at 60% as quickly as possible. You will perform this task via Fuzzy Control! You see that coming...

You do not need to worry about the aforementioned dynamics of the vaccination to provide proper controls. You will be provided with a class named *Vaccination* in the *vaccination.py* file under *HW3* folder in *ODTUClass* course page. With that class, you are able to take measurements to determine your control and apply your control to vaccine the people. The measurements are the current vaccinated percentage and effective vaccination rate. You are able to control the daily vaccination rate by increasing or decreasing the current vaccination rate. For instance, if your control output were c and the current vaccination rate were r, then the updated vaccination rate would be $c + r$. Once you apply your control by calling the related method of the Vaccination class, the vaccination rate will be updated, the effective vaccination rate will be determined and the percentage of the vaccination population will be updated with those rates for 0.1 day (2.4 hours), internally. Then you can observe the updated vaccinated population percentage and effective vaccination rate by simply calling the related class method. Note that you are observing changes within 0.1 day intervals. The flow of the controller is depicted in Fig. 1 together with the sample codes to be used in your implementations.

Input Variables and Control Variable

The input variables are the two measurements which are the current percentage of the vaccinated people,

$$\pi \in [0, 1]\,,$$

and the current effective vaccination rate,

$$\dot{\pi} \in [-1, 1]\,.$$

The control variable,

$$\delta \in [-0.2, 0.2]$$

is the vaccination rate to be added to the current vaccination rate. Note that the control variable allows to increase or decrease the vaccination rate and its maximum magnitude is limited to 0.2.

While providing your control to the Vaccination class, you do not need to consider whether the percentage will exceed 1 or drop below 0. Those are internally handled within the class. Similarly, you do not need to consider whether the vaccination rate will exceed 1 or drop below -1 as well.

1 Vaccination v1

In this version of the vaccination process, you have not researched the failure rates yet. Thus, you can only measure the current percentage of the vaccinated people. In this manner, design a fuzzy controller which takes π as the input and provides δ as the output to vaccinate 60% of the people and maintain that percentage.

1.1 Set Partitioning

Partition the sets where the measurement and the output lie into **3** fuzzy sets. Namely, you are to represent $[0, 1]$ and $[-0.2, 0.2]$ with three fuzzy sets for each. You may experiment on different partitioning strategies by examining the performances. Decide your favorite partitioning and plot your fuzzy partitions for both sets in Python and include them in your report. Explain how you decide that partitioning.

1.2 Fuzzy Control Rules

Considering your partitions, list your control rules.

1.3 Fuzzification and Defuzzification Interface

Implement your controller in Python. Explain how you implement the fuzzification and defuzzification interface.

1.4 Simulation

Using your controller and the Vaccination class, vaccinate people for at least 20 days. In other words, perform at least 200 iteration of control. You may increase the number of iterations if you fail to observe the equilibrium.

Now, you are not sure whether your vaccine behaves efficiently. You benefit from herd immunity when your reach 60% of vaccination and maintain that percentage. Until then, unit vaccination percentage is considered to have unit cost. You will compute the total cost until the equilibrium. Your performance measures will be that vaccination cost and the number of days passed until the equilibrium. To compute those measures:

1. Estimate the iteration number where the vaccination comes to an equilibrium at 60%. You can use *vaccinated_percentage_curve_* attribute of the Vaccination class to obtain the percentage curve.
2. Compute the vaccination cost until the equilibrium. You can use *vaccination_rate_curve_* attribute of the Vaccination class to obtain the curve of the infection rate consumed during the spread process.
3. Visualize your vaccination process by using the *viewVaccination* method of the Vaccination class. You should provide iteration number you find in Step 1 and the cost you computed in Step 2 to that function in order to properly visualize the vaccination process. Add that plot to your report.

2 Vaccination v2

You decide to conduct research on the failure rates to develop your vaccination process after you have observed the behavior of the previous version of your vaccination process. You want to decrease the overshoot and increase the convergence rate to the equilibrium. You have made your research and now you can measure the current effective vaccination rate which is the difference between the controlled vaccination rate and the failure rate. In this manner, design a fuzzy controller which takes π and $\dot{\pi}$ as the inputs and provides δ as the output to vaccine 60% of the people and maintain that percentage.

2.1 Set Partitioning

Partition the sets where the measurements lie into **3** fuzzy sets. Namely, you are to represent $[0, 1]$ and $[-1, 1]$ with three fuzzy sets for each. Partition the output set into **5** fuzzy sets $[-0.2, 0.2]$. You may experiment on different partitioning strategies by examining the performances. Decide your favorite partitioning and plot your fuzzy partitions for both sets in Python and include them in your report. Explain how you decide that partitioning.

2.2 Fuzzy Control Rules

Considering your partitions, list your control rules. You may experiment on different set of rules by examining the performances. Decide your favorite set of rules and list them. Explain how you decide that rule set.

2.3 Implementation and Simulation

Repeat 1.3 and 1.4 for this version of the vaccination process.

2.4 Comparison

Considering the performance measures and the behavior of the percentage curves, compare two versions of the vaccination. State the differences and discuss.