METU
Electrical and Electronics Engineering Department

EE 496

2020-Spring Term

HW3

Introduction to Computational Intelligence

Onur Selim KILIÇ
2169530

1. Introduction

In this homework, an example fuzzy system is implemented and tested for different cases. I have used "skfuzzy" library to conduct some of the important parts of the fuzzy systems together with the given "plague.py" code.

2. Plague v1

For this part, we do not have any information with regard to the disappearance rates. So, by using only the current percentage of the infected bots, the percentage of the infected bots is fixed to 60%.

2.1. Set Partitioning

An essential part of the fuzzy systems is partitioning sets. The selection of the sets are very crucial to have a proper convergence point, a faster convergence speed and a smaller overshoot. Although, there is not a precise rule for the set partitioning, one can find a favourable set partitioning method experimentally. I have tried a number of sets to see the effect of selecting different sets. I have tried trapezoidal, triangular and Gaussian membership functions. In the end, the best partitioning method for this part has turned out to be the following set partitions.

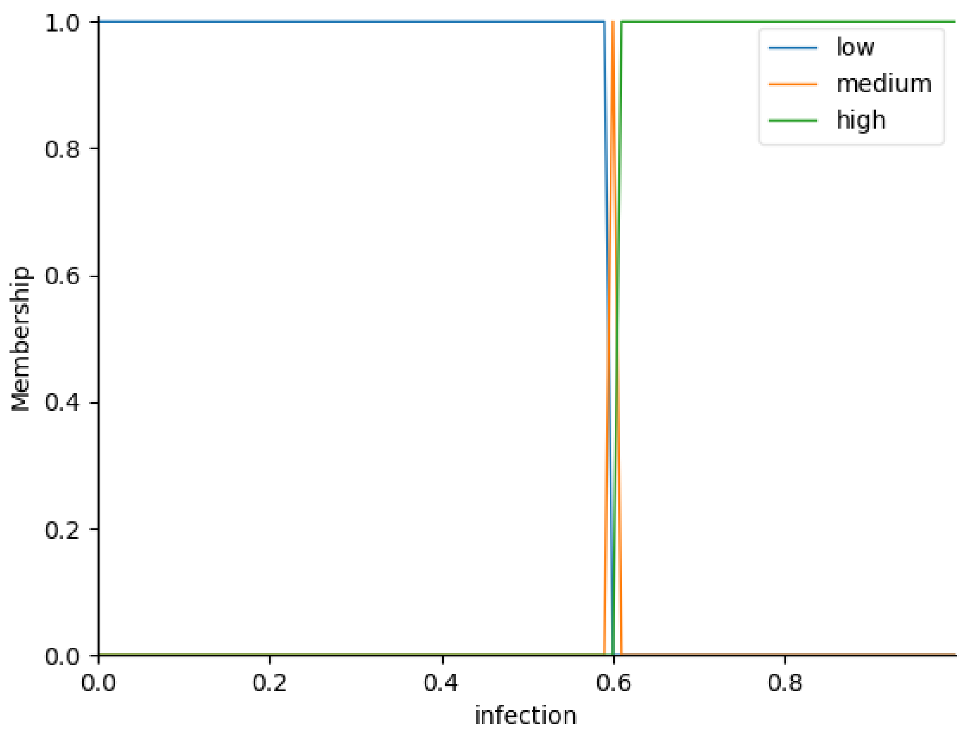

Figure 1: The membership functions for the infected percentage of the population which is found experimentally.

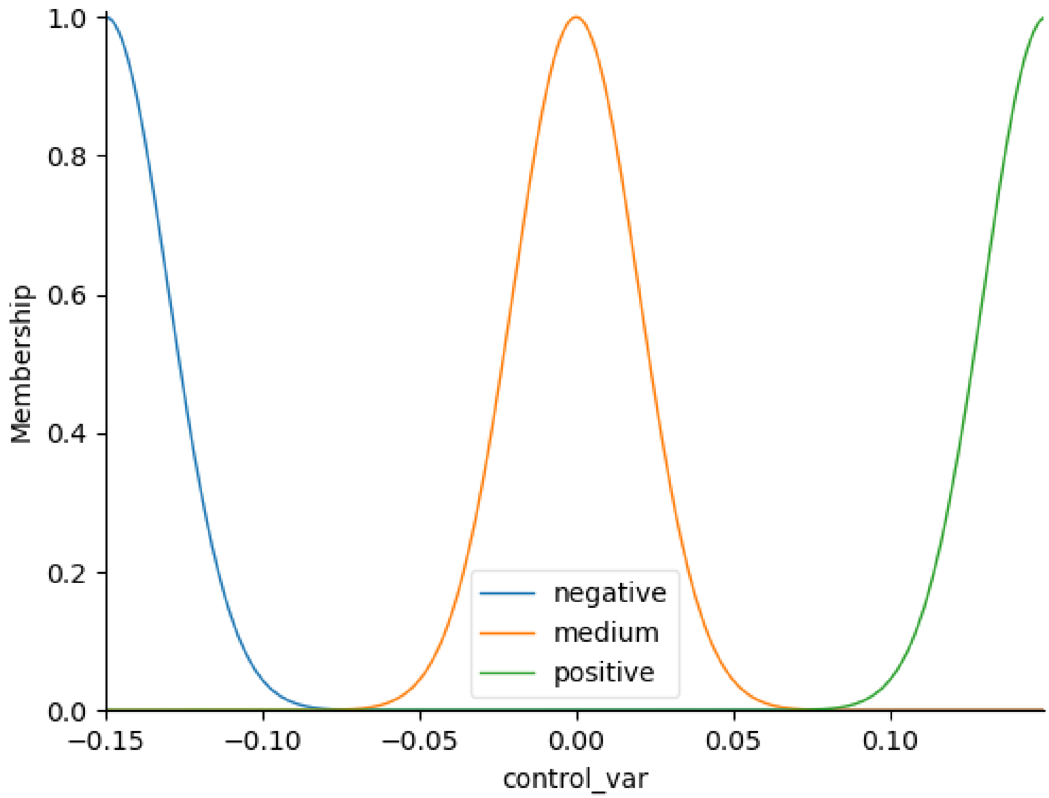

Figure 2: Experimentally found membership functions for the control variable of the fuzzy controller.

While I was selecting the membership functions, I have assigned the “medium infection” to the interval [0.59-0.61] in a triangular fashion. By doing so I have aimed to have a convergence point of 0.6. The infection rates smaller than 0.6 is assigned to “low infection”. While I do so, I have used a trapezoidal membership function which can be seen clearly in Figure 1. Finally, the intervals greater than 0.6 is assigned to “high infection” in a trapezoidal fashion.

For the “control_var”, I have used 3 Gaussian membership functions with 0.02 variance. The negative control variable is located around -0.15, the positive control variable is located around +0.15 and the medium control variable is located around 0.00. By selecting these endpoints for the control variable, I was able to speed up the convergence of the fuzzy controller to the 0.6 percent infection of the population. Consequently, the cost of the fuzzy system has also decreased compared to other set partitioning methods. But this choice for the set partitioning have caused some oscillations after the fuzzy system reached to point 0.6. In my design I have accepted small oscillations, namely oscillations smaller than 0.001. Therefore, with a very small convergence error, I was able to reach to point 0.6 with a very small convergence time.

These partitioned sets together with the fuzzy control rules construct the fuzzy controller and they stabilize the percentage of the infected population at the point 0.6 as it is desired in the homework.

2.2. Fuzzy Control Rules

The control rules of the fuzzy system are also crucial for the fuzzy control to know which operation it should conduct for a particular input.

Table 1: The control rules for the plague v1 fuzzy controller.

Infection Percentage	**Control Variable**
Low	Positive
Medium	Medium
High	Negative

A simple method is used for this fuzzy control rules that is when the infection percentage is low, control variable is set to be positive to increase the infection percentage of the system. Similarly, when the infection percentage is high, the control variable is set to be negative to decrease the infection percentage. Finally, for medium infection percentage's (the interval [0.59-0.61]) the control variable is set to medium values which are mainly very close to zero.

With the help of these fuzzy control rules, the fuzzy controller was able to reach to a stable point of 0.6 (Or a very closer point). The related test results will be given in the following parts of this chapter.

2.3. Fuzzification and Defuzzification Interface

For the Fuzzification process, the current situation of the plague is obtained from the given code "plague.py". Then according to the membership functions that are explained in part 2.1 the membership values for the infection are calculated. Together with the membership values for infection, membership function for control variable and the control rules, the fuzzy controller determines the a value for the control variable. This is mainly the Defuzzification process and it can be observed in the Figure 3. For the Defuzzification process, the centre of gravity method is chosen. That is firstly, the three member ship values for low, medium and high are found. Then with these membership values, the centre of the gravity for the control variable is determined. In Figure 3, one can see the determined control variable for an infection of .60048. As it can be seen, the value of the control variable is very close to 0.

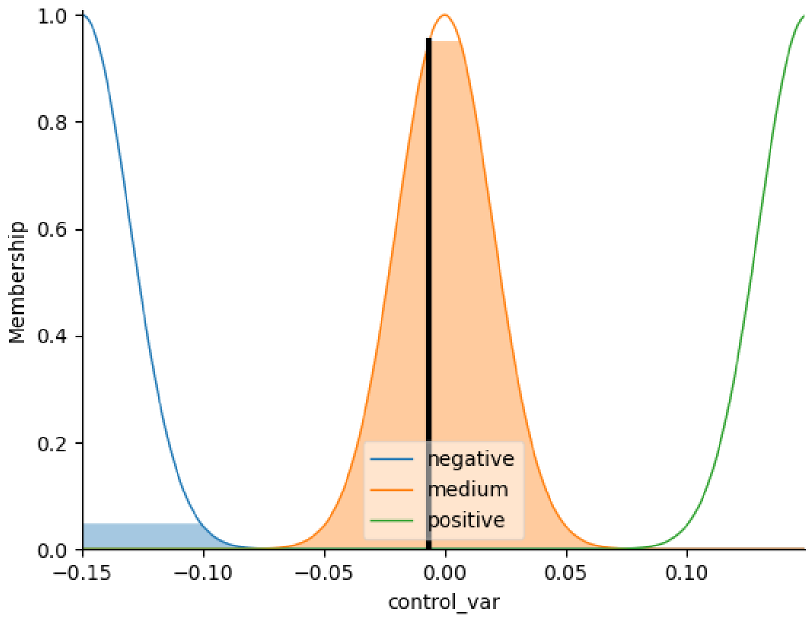

Figure 3: An example Defuzzification operation for an infection of 0.60048.

2.4. Simulation

In this part of the document, the simulation results are provided. As I have stated before, I have tried many different partitioning methods for both the control variable and the infection membership functions. In this part, I will be representing the best result out of many trials.

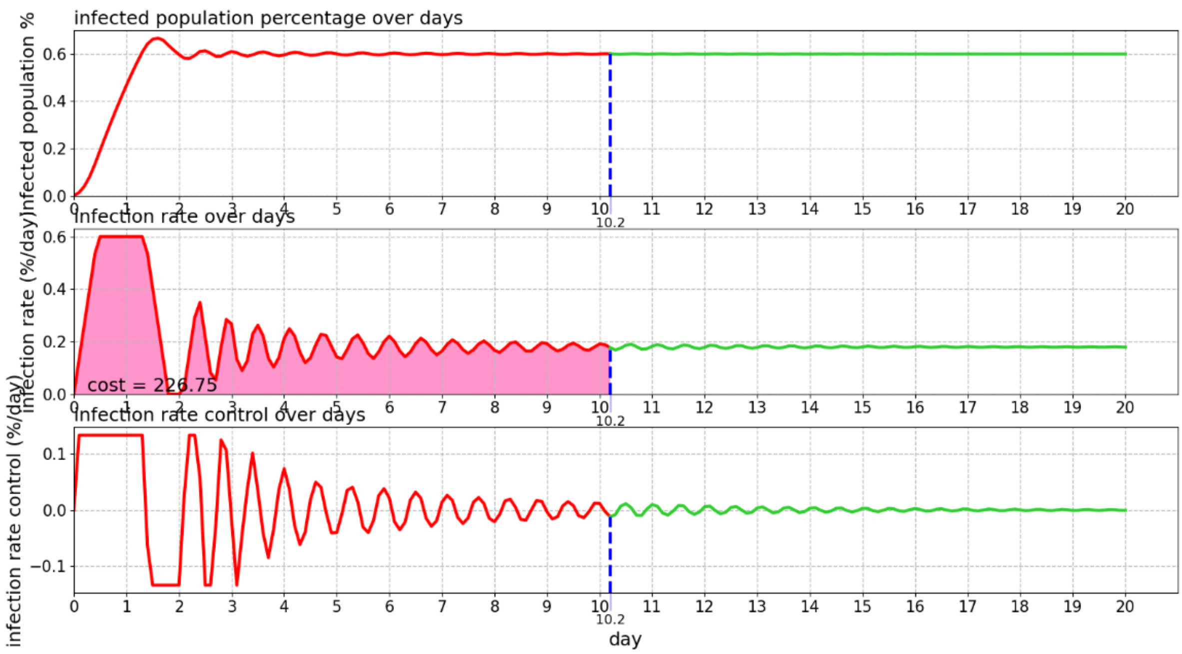

Figure 4: The simulation results for plague v1.

For the calculation of the convergence time, I have used a tolerance of 0.001 infection. As it can be seen from Figure 4, the infected population percentage is very close to 0.6 after day 10.2. Up until that day, a total cost of 226.8 is spent. While I calculated the cost, I have considered every 0.1 day as the time in which we use the current infection rate. Since every 1% out of 60% costs 1 for one day, the total cost is calculated as the following;

- First the numbers in the array infection rate array is added up, up until the fuzzy controller reaches steady state.
- Since every number in the infection rate array is operating for 0.1 days, the sum is multiplied by 0.1.
- The prewritten code projects infection rates of 0-60% range to 0-0.6, so to have a unit cost for 1%, I multiplied the sum with 100.

After the above mentioned operations are done, the total cost turned out to be 226.8.

As it can be seen in the top plot, there is overshoot for the infection population. The results found in this simulation will be compared with the results of the second simulation in following chapters of this document.

3. Plague v2

In addition to the Plague v1, we have the information with regard the infection rate over days. So, I will try to make use of this information to decrease the convergence time and overshoot.

3.1. Set Partitioning

As it is stated before, in this part we have the information with regard the infection rate in this part. And also, the infected population membership function will compose of 5 different curves instead of 3 as it is in Plague v1.

Similar to the Plague v1, I have chosen the membership functions experimentally. In Figure 5, 6 and 7, one can see the membership function for infected population percentage, infection rate and the control variable. Similar to the Plague v1, I have chosen the membership functions to have a convergence point of 0.6. As it can be seen from Figure 5, the infected population rates from 0.5 to 0.7 is chosen to be medium. Additionally, the infected population rates above 0.6 is considered to be high and the infected population rates below 0.6 is considered to be low. With these configuration, the fuzzy controller converges to 0.6.

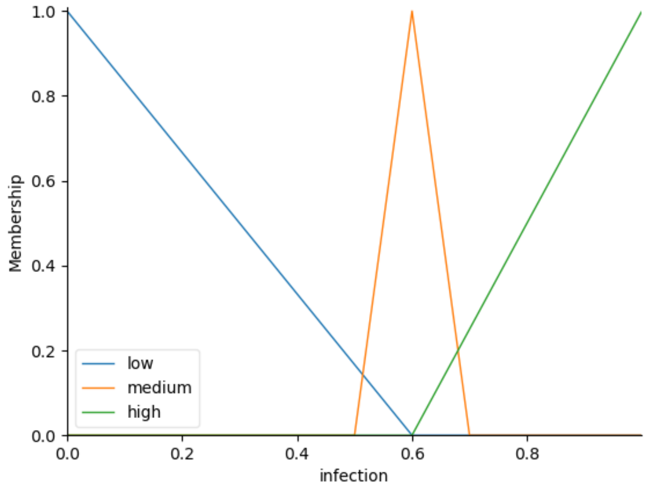

Figure 5: Membership functions for infected population rates.

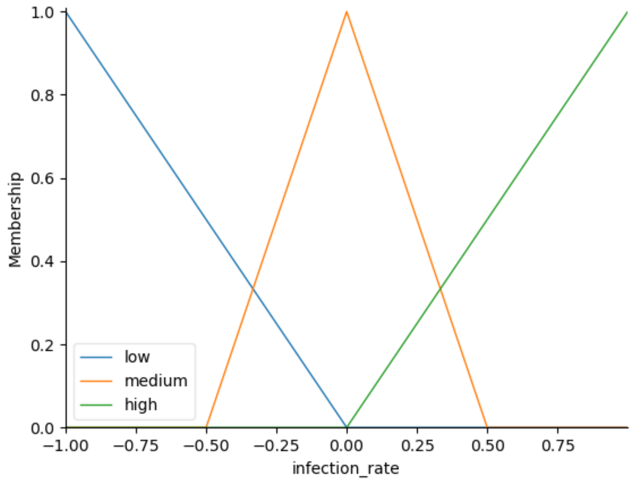

Figure 6: Membership functions for infection rates.

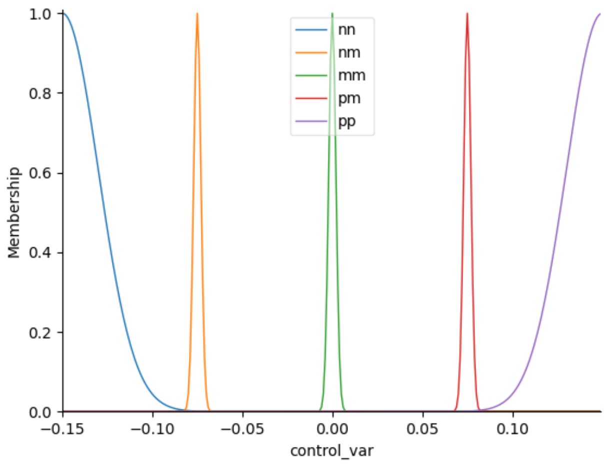

Figure 7: Membership functions for the control variable.

I have used three triangular membership functions for the infection rates which can be seen in Figure 5. By doing so, the fuzzy controller can easily understand the current trend in the infection rate.

Finally, I have used 5 Gaussian membership function for the control variable. As it can be seen the variance of the left most and the right most Gaussian membership functions are a little larger than others. This way, I have tried to speed up the convergence of the fuzzy controller.

3.2. Fuzzy Control Rules

Table 2: The control rules for the plague v2 fuzzy controller.

Infection Percentage	The Operant	Infection Rate	Control Variable
Low	OR	Low	Positive-Positive(PP)
Low	OR	Medium	Positive-Medium(PM)
Low	OR	High	Medium-Medium(MM)
Medium	OR	Low	Positive-Medium(PM)
Medium	OR	Medium	Medium-Medium(MM)
Medium	OR	High	Negative-Medium(NM)
High	OR	Low	Medium-Medium(MM)
High	OR	Medium	Negative-Medium(NM)
High	OR	High	Negative-Negative(NN)

A similar method to the one which is used in plague v1 is also used for the plague v2. But this time, we have also the infection rate membership values which I can use to decrease the overshoot and increase the convergence speed. As it can be seen I have used an “OR” operant in my control rules. So the control rule selects the larger membership value between the infection rate and the infected population percentage membership values. I have also tried to use “AND” operant (for this case, the fuzzy controller selects the smaller membership value out of the two) but this case turned out to be better.

3.3. Fuzzification and Defuzzification Interface

For the Fuzzification process, a total of 9 membership values are calculated. Then using these 9 membership values together with the 9 fuzzy control rules, the new value for the control variable is calculated. Since I have used "OR" operant in my rules, the fuzzy rule selects the membership value which is higher. Similar to the Plague v1, I have used centre of gravity approach for the Defuzzification. One can see an illustration of the determined control variable in Figure 7.

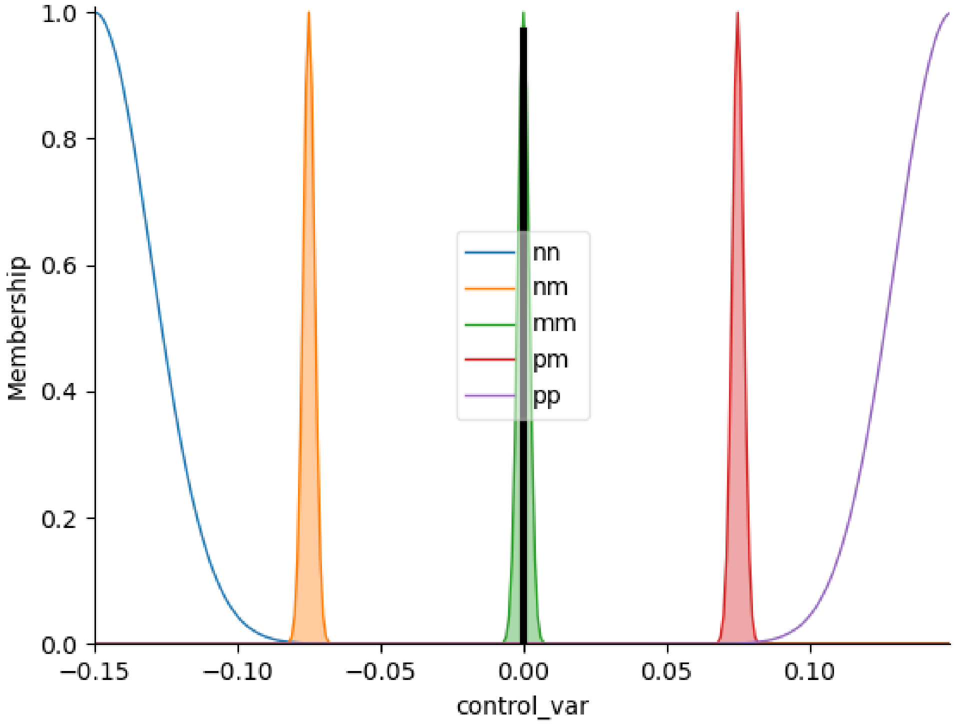

Figure 8: An illustration of Defuzzification operation for plague v2.

In Figure 8, the value for the control variable is determined when the infected population percentage is 0.600027. As it can be seen it so close to 0 and it is closer to compared to Plague v1 steady state case.

3.4. Simulation

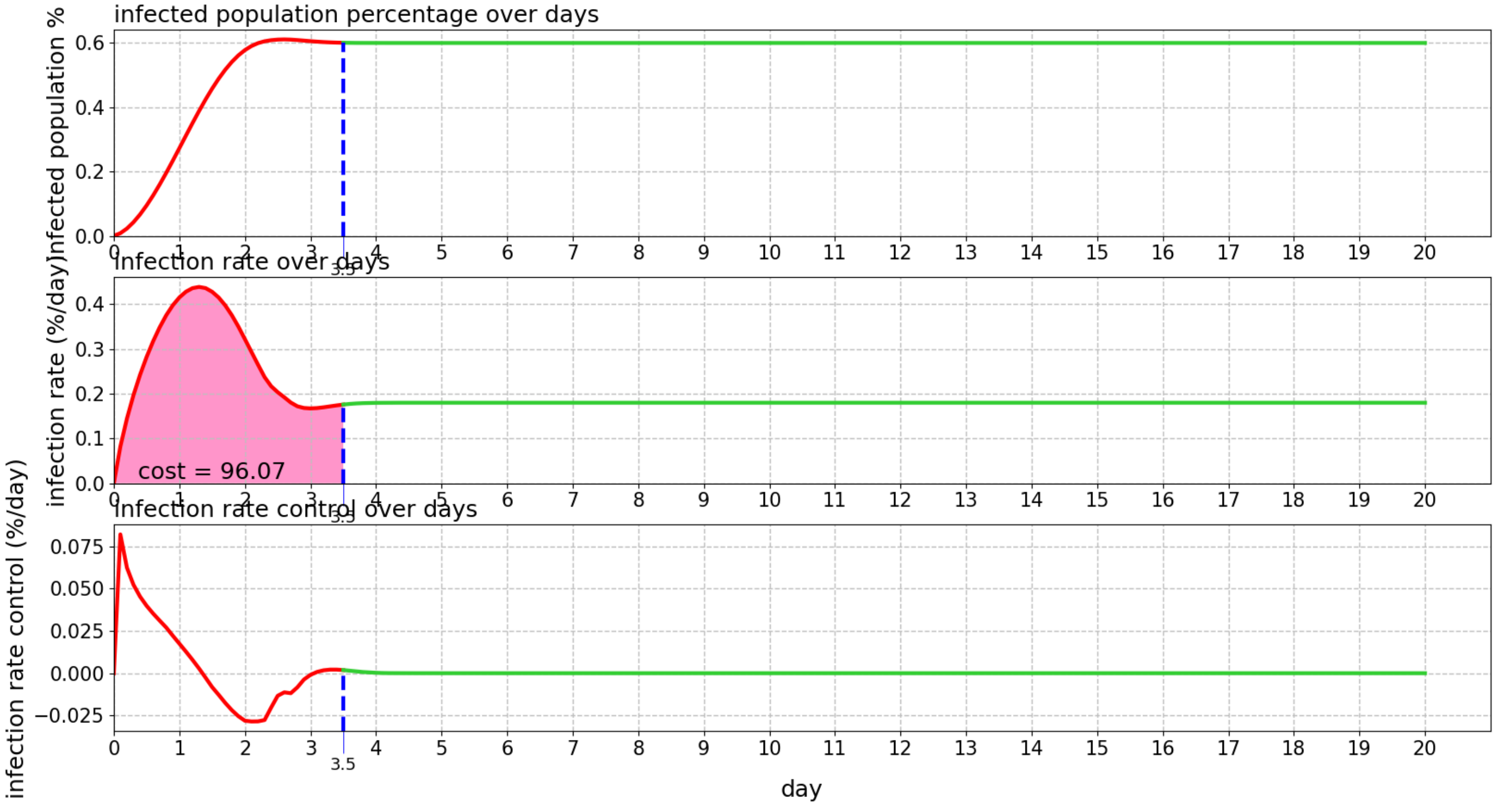

Figure 9: The simulation results for Plague v2.

One can see the simulation results that are obtained for the Plague v2. The cost of the fuzzy controller turned out to be 96.07 and the convergence time for the fuzzy controller is 3.5 days. As it can be seen from the Figure, there is almost no overshoot and the fuzzy controller converges very quickly.

As I have stated before, I have made many trials with many different membership functions and control rules. This is the one, which worked in the most efficient way.

The cost and the steady state day calculations are similar to the ones that are conducted in Plague v1.

3.5. Comparison

In this part of the document I will compare the performances of the two plague versions in terms of convergence speed, overshoot and the cost.

Table 3: The comparison of the two Plague versions in terms of related parameters.

Parameters	Plague v1	Plague v2
Cost	226.75	96.07
Convergence Day	10.2	3.5
Overshoot	0.4233	0.1102
Convergence Point	0.6000415393161597	0.5999999999999996

As it can be seen from table 3, Plague v2 is superior in terms of cost, convergence day, overshoot and convergence point. This is actually expected because, in Plague v2 we also have the information with regard to the infection rate. So, knowing the current infection rate and changing the control variable together with this knowledge has fastened the convergence time a lot. Consequently, the cost of the fuzzy controller has also decreased.

Similarly, knowing the infection rate and selecting the control variable with this knowledge, I was able to decrease the overshoot also. Finally, the convergence point of the system is now so much closer to the point 0.6 because we have two additional membership functions for the control variable. These two additional membership functions helps the fuzzy controller to have a more stable steady state condition.

All of these differences are actually expected because we are using additional information about the system to change our control variable. In other words, it is always better to have additional information about the system we are dealing with.