

Climbing Fuzzy Inference System – IMAT3406



BOULDERING

Introduction

Climbing is one of the most extreme sports in the world which is becoming popular every day. New and unexperienced climbers are at a high risk for injuries that could have been avoided.

A control method to highlight the degree of success for different climbers can be seen using Mamdani type one fuzzy inference systems. This creates an effective way to determine the chance of success for any climber as well as how many attempts it would take them on average to climb a wall.

This report highlights the issues of climbing and explains how a fuzzy inference system can be used to limit injuries and predict someone's likelihood of completing a climb.

The graphical outputs of these fuzzy inference systems can be found in the Appendix

Literature Review

“Fuzzy logic has two different meanings. In a narrow sense, fuzzy logic is a logical system, which is an extension of multivalued logic. However, in a wider sense fuzzy logic (FL) is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects without crisp, clearly defined boundaries. In such cases, membership in a set is a matter of degree. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its narrower definition, fuzzy logic differs both in concept and substance from traditional multivalued logical systems.”

(Uk.mathworks.com, 2018) And “The fuzzy logic deals with incomplete, inconsistent, uncertain, vague and undecided information with belief.” (Reddy, 2021) describe best what Fuzzy logic is all about. the vagueness and uncertainty of a logical system.

One way to showcase fuzzy logic is within a fuzzy inference system is by using fuzzy “inference rules” (H. Teodorescu and T. Yamakawa, 1993) these inference rules are a way of creating logical steps within a fuzzy inference system by creating a sentence and then using a base of information in order to output a desired variable. E.g. a rule base where Adam is tall and Adam is heavy exist. The inference rules can say “If Adam is tall and Adam is heavy then Adam is big” it can be seen as if “words are used in place of numbers for computing and reasoning” (Zadeh, 1996). This can be done in MATLAB which is the software that I will use to create my fuzzy inference system.

Climbing has a lot of history, Dominic Saul states “Its evolution began in the 19th century in northern England and Italy, predominantly with alpine mountaineering, and continued until the 1950s when the general public began to increase their interest in climbing as a sport. The climbing without use of any aids was born in the beginning of the 20th century” (Dominik Saul, 2019) showcases some climbing history.

“Rock climbing has been steadily growing as a sport and debuted at the 2020 Tokyo Olympics.^{1,2} With the increasing participation, climbing injuries are on the rise.” (Joo Hyung Yoon, 2022), “Acute injuries in rock climbing either come from a fall onto the lower leg or from performing a hard move and injuring the upper extremity” (Volker Schöffl, 2016) and “they strain the passive and active anatomical structures of their hands and fingers to maximum capacity during training or competing” (Christoph Lutter, 2016) are some examples about climbing injuries. This statistic is on the rise because of new rock climbers, and This is a main reason that a Fuzzy inference system would be wanted in this sport. The system will predict how likely someone is to complete a course in a single attempt. Climbing is a sport that has many variations. “Climbing venues can be found both outdoors (e.g., on natural cliffs or mountain rock walls) and indoors in climbing gyms. A climbing gym usually provides a large number of different climbing routes on artificial walls, often constructed from plywood and synthetic holds” (M. Andrić, 2022) quotes that the stereotypical climbing has lots of different variations, hand holds and locations. This makes it more imperative that a FIS is made for this sport. Proposing a Mamdani style FIS which judges how successful the climber will be on a certain wall would make sense to use 2 FISs which are then inherited into a master FIS. The first child FIS would be Climbers Skill and the second being Wall difficulty.

System Overview

Design detail

There are 3 different forms of climbing each with their own individual challenges. To make the Fuzzy Input System (FIS) it would be wise to only one of these three. I have decided to go with Indoor Bouldering as it demonstrates a wider spread of climbing skills as well as not having the restriction of a rope and harness. A climbers' success rate is dictated by two distinct factors, the difficulty of the wall that is attempted and the skill of the climber. For the system to work, I will have to make 3 Fuzzy inference systems. One system will output the climber's skill, one will output the difficulty of the wall and the third will output the success of the climber. The range for each variable will be determined by using the minimum and maximum value that can be found in real life cases.

Climbers' skill:

Climbers' Skill has 4 Inputs. These include BMI (kg/m^2), Ape(cm), Experience (Years), and Strength (repetitions). Climbers' Skill has 1 Output, Climber Skill (degree of success) The ranges for each input are as follows:

Variable	Variable Type	Range	Intervals
BMI(kg/m^2)	Input	12-40	Very High, High, Average, Low, Very Low
Ape Index (cm),	Input	-10 - +11	Very Small, Small, Average, Long, Very Long
Experience (Years)	Input	0-20	Very Inexperienced, Inexperience, Average, Experienced, Very Experienced
Strength (repetitions)	Input	0-200	Very Weak, Weak, Average, Strong, Very Strong
Climber Skill (%)	Output	0-100	Very Low, Low, Medium, High, Very High

Justification of the parameters chosen:

BMI (kg/m^2): BMI stands for Body mass index which is defined by comparing your mass to your height. Within climbing it is advantageous to be tall but also to be light. A low BMI allows for climbers to throw themselves around and pull themselves up without a lot of energy or effort. However, a High BMI will mean that the person has more mass to move around compared to their body weight. Since A BMI of less than 12 is classed as severely underweight and a BMI of 40+ is severe obesity, my ranges I have stated will be between 12-40 as ranges outside this would put the person at a significant health risk. (Center for Disease Control and Prevention, 2022)

Ape Index(cm): A person's ape index is the ratio between their own height and their arm span. To calculate this, you subtract the person's height from their arm span to get a positive, negative or value of 0. If a value is positive, it means the climber's arms are longer and are also beneficial for climbing. The ranges for this are between -10 and +11 as these are the 2 anomalies in the world. (Beale, 2021)

Experience (Years): Someone with experience in climbing has different techniques like body positioning and climbing moves that ordinary starting climbers just wouldn't have. Experience for a climber not only conditions your problem solving of the wall but also allows for greater on the spot thinking when attempting the wall for the first time. The ranges for this start at 0 (new) and 20 as the other extreme for avid regular climbers.

Strength (repetitions): there are a lot of ways to utilize strength in climbing. However, the best way to test someone's strength for climbing is to attempt however many pull ups they can do in a row. This shows their endurance, stamina, and overall strength of the climber. the ranges are set at 0 (can't do a pull up), and 50 being very strong as this benchmark exceeds the marines training where they only must do 10 (Wood, 2008)

Walls difficulty:

Walls difficulty has 3 Inputs. These include Walls Height(m), Hold Distance(cm), and Incline ($^{\circ}$). Walls difficulty has 1 Output, Wall difficulty (degree of success)

The ranges for each input are as follows.

Variable	Variable Type	Range	Intervals
Wall Height(m), Hold	Input	0 – 4.5	Very Small, Small, Average, Tall, Very Tall
Hold Distance(cm),	Input	0-300	Very Small, Small, Average, Tall, Very Tall
Incline ($^{\circ}$),	Input	-90 - 90	Flat, Slightly Inclined, Straight, Very Inclined, Overhang
Wall Difficulty (%)	Output	0-100	Very Low, Low, Medium, High , Very High

Justification of the parameters chosen:

Wall Height: the wall height input measures the horizontal distance of the wall. Although bouldering has no ropes in use, the walls can still be very tall. An Olympic sized bouldering wall has a maximum height of 4.5 meters so I will use this as my limit. This is included as a climbing wall that is not very tall won't be as difficult as a taller wall on average.

Hold distance: this refers to the maximum distance between each hold. This can determine whether a hold is reachable for the climber. as the world record for the largest dyno is less than 3 meters, I have set the range between 0-300cm.

Incline: the incline of the wall is simple vectoral physics. To go up at an angle you'll need both energy going up the wall and energy out of the wall. This is a very big factor for climbers. In this example, a positive number would mean that a part of the wall is protruding. If the incline of the wall is negative that means it is going away from the

wall. The reason that the range is only -90 and 90 is that if it goes over 90 then the climber starts climbing down.

Success of climber:

Success of climber has 2 Inputs. These include Climber Skill (%) and Wall Difficulty (%). Success of climber has 1 Output, Success of climber (%),

The ranges for each input are as follows.

Variable	Variable Type	Range	Intervals
Climber Skill (%)	Input	0-100	Very Low, Low, Medium, High, Very High
Wall Difficulty (%)	Input	0-100	Very Low, Low, Medium, High, Very High
Success of climber (%),	Output	0-100	Very Low, Low, Medium, High , Very High

Justification of the parameters chosen:

Since the previous 2 outputs have been inherited into inputs, the ranges and intervals are unchanged. The Success of the climber output variable follows a similar range and therefore can be interpreted just as well.

Testing and Tuning:

Initial Design in MATLAB:

For my initial declaration of variables, rule sets and fuzzy sets for the climber's skill, wall difficulty and Success of climber are covered in the appendix with fig (1-5). I used trapezoidal membership functions as the left and right of my sets as the variables plateau and therefore incorporated 100% membership association. Triangle membership functions (trimf) were used in the mid intervals to give as most of the fuzzy sets overlap was there and trimf was excellent at displaying this.

My initial design contains 305 tests which is an average number for a project this size. Because the climber's skill inputs all have 5 intervals in their own range, has a total of 161 tests with the wall difficulty having 119 tests and the success of the climber having 25 tests. These are a lot of rules, and it would be best to try and optimize his in any way possible. The fuzzy sets also have their own set boundaries each within a reasonable relation to the other intervals.

Dummy Testing

The dummy testing started by inputting fake information into my excel spreadsheets to see if the desired outcome came out as the correct values. However, I soon realised that I didn't nearly have enough rules as I needed to have. Since all my operations relied on the AND operator, each individual interval for each FIS had to be satisfied. I as all my values have 5 intervals, the operation was simple. The number of rules to make per FIS is the number of intervals to the power of my inputs. So, the maximum the Success of climber was $5^2 = 25$, the Wall difficulty was $5^3 = 125$ and the climber's skill was $5^4 = 625$. These rules are imperative to allow all, and any climbers attempt the wall. Therefore, I updated my rule set to include all these rules with appropriate outputs as well, adjusting the outputs for each of them to get the best result possible. the new rules can be found in the appendix with fig 6. The system then outputted the results from wall difficulty and climbers' skill into the input section in the excel spreadsheet for success of climber. This test proved to be true, and I could happily move on to testing different defuzzification methods with real life. Before that, I found that the Success of the climber needed to be inversed to give an accurate representation. This changed the rule base for success of climber as well, which I have also added to the appendix with fig 7

Real world Data

Real world data of athletes and pro climbers are available to people looking for statistics. However, the FIS was made to input nonathletic and athletic people in. Therefore, the data used will be created to allow a wider scope of people and a broader success rate for new, weak, and overweight people. The data used is in the appendix in figure 8. I have also created a prediction table which estimates the desired outcomes for each FIS based on the rule bases attached to them. This can be found in the appendix with figure 9. This table estimates values for the Climber's Skill, the Wall Difficulty, and the Success of the climber.

Test 1: Climber's Skill

For this test, I will look at which defuzzification method is best for my FIS. Currently the system is only using centroid defuzzification which is relatively reliable, however there are 4 other defuzzification methods available and to find the best one we must try all of them. Because there is 3 FIS within my code, I have decided to test each system within their own test.

Defuzzification:

The data created provides a widespread on each bound of the FIS. Using the Climber's skill FIS, I ran the data against my system using the different Defuzzification methods. The system ran 5 defuzzification types on the climbers Skill which are Centroid, Bisector, Lom, Som, Mom. The outputs of this are displayed in the appendix with figure 10 . The results showed to be positive with every desired outcome achieved bar 1. This is showcased in the comparison table which takes the different outputs from the system, compares them to the expected output and then determines true if the outputs match or false if it didn't.

Outcome:

The climber's skill FIS passed the majority of proved that my FIS was advanced enough to be used in the success of climber. The system Is advanced enough to take in most real-world data and output an estimated value. However, the FIS still needs alteration. Tests that have failed all the tests may be due to the rule base being incorrect or the distribution of the Fuzzy sets are a bit off.

Changes:

The system Didn't require any changes. Although there were some false outputs in the centroid and bisector method, the mom and lom methods didn't have any false data outputs in them and therefore this this method is correct and doesn't need any alteration. However, I found 1 alteration, the intervals for BMI were the wrong way around which may have displayed the rule base in a different way than normal. This will be rectified when I output the final rule base, fuzzy sets, and variable declaration.

Test 2 Wall Difficulty:

For this test, the FIS will be focussing on the wall difficulty section where it will again look at the desired and actual output and determine if and what needs changing. The program will be using the 5 defuzzification methods that were stated prior to determine the degree of relation each of them have. The outputs to the Wall difficulty FIS are displayed in the appendix with figure 11 as well as the desired vs actual output table.

Outcome:

The Wall difficulty FIS passed less test than anticipated. As well as this, the FIS didn't fire 7 pieces of data from the table, this being data test 6 and data test 14-19. The systems rule base will be checked before continuance. Other than these 7 data test the system is functional. The outputs may change after the changes

Changes:

As stated previously, the FIS for Wall Difficulty is 5^3 or 125 rules in the rule base. There are only 119 rules in my system, meaning I was missing 6 rules. After reviewing the existing rules, I have written the last 6 rules and put them into my test list for the program to activate them. However, this has not fixed the reoccurring problem. My fuzzy sets are the next to look at. When I look at the fuzzy sets, there appears to be no relation between parts of the intervals. The system changes its intervals to allow for a bit of relation and to begin the fuzzification step. The updated code and fuzzy set are in the appendix with figure 16.

Outcome 2:

After processing the data through the 5 defuzzification types, the system now accepts the 7 pieces of data as well as the expected outcomes being correct. Like the results of the climbers Skill, there are almost no failures in any of the defuzzification methods and therefore the system can move on to the Success of Climber FIS.

Test 3 – Success of Climber:

Outcome:

After running the tests, the desired outcome for the FIS wasn't correct. The desired outputs were, although sometimes correct, unreliable and I looked for the problem in the system. All the Rules had been made the same way as the other 2 FIS and therefore added up all the values and got as large or as small an answer as possible. Figure 20 in the appendix

Changes:

The rules had to be changed to output an answer if the results were too different. E.g., if climbing skill=1 and wall difficulty = 1. Then the success of climber would be average or 3. This wasn't the way the FIS' rules were set up. The system has been amended to accommodate these changes.

Outcome 2:

This outcome gives us the desired output as we expected. Almost all the tests determined to be true, and the false ones are only a few points off. This concludes the testing and tuning of the FISs as the outputs are as refined as they need to be to produce a crisp and clean value that can be used to calculate the chance of a climber climbing a wall.

Critical Analysis

What started out as a small and simplistic way to showcase the ability of climbing, turned out to be a lot more complex than people thought.

Further testing:

Some further testing that could be achieved could include

1. Using Real world data such as the Olympic bouldering team statistics.
2. Using gaussian membership function with the climbing skill and the wall difficulty
3. Modelling within a different software like Simulink as it has more of a graphical interface.
4. Trying to incorporate the system into an app or software that can be used in real life

Improvements

There are many improvements that I could have undergone. For example

1. Optimize the code by using a Karnaugh maps to simplify the fuzzy rules for each FIS
2. Combine the excel spreadsheets into 1 to enhance file dependability
3. Incorporate different muscle groups rather than just base strength
5. Incorporate different types of climbing holds and surface texture
6. Incorporating the other 2 forms of Olympic climbing (speed climbing and lead climbing)
7. Have an entire FIS on hand grip as out of everything, it is the most important strength in climbing.
8. Use the new terms suggested like mamfis instead of newfis to make the project more sustainable and available in the future.
9. Make all the expected outputs be true in mom, low, som, and bisector

Reflection:

If I was able to retry the project I would probably:

1. Make the system more modular and have no more than 2 inputs per fuzzy logic operator. This avoids the ever-increasing range of rule bases that would get exponentially larger the more inputs per system.
2. Spend more time honing the intervals in to perfect the system.
3. Not waste time creating 625 rules for the rule base. Instead focus on everything else and not try and create and cover all data entries.

Conclusion

To summarize, I have created a fuzzy inference system on the likelihood of a climber being able to climb a wall. In the appendix my results show my progression and final state of the system. Time could be spent more efficiently testing and tuning as opposed to coding. This report can be used as a more extensive and elaborate project.

References

- Beale, A., 2021. *Ape Index Calculator: What's Your Ape Index?*. [Online] Available at: <https://www.99boulders.com/ape-index-calculator> [Accessed 8 October 2021].
- Center for Disease Control and Prevention, 2022. *Calculating BMI Using the English System*. [Online] Available at: https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmimage/page5_2.html [Accessed 3 December 2022].
- Christoph Lutter, A. S. T. H. T. B. V. S., 2016. Pulling Harder than the Hamate Tolerates: Evaluation of Hamate Injuries in Rock Climbing and Bouldering. *Wilderness & Environmental Medicine*, 27(4), pp. 492-499.
- Dominik Saul, G. S. W. L. A. F. S., 2019. Determinants for success in climbing: A systematic review. *Journal of Exercise Science & Fitness*, 17(3), pp. 91-100.
- H. Teodorescu and T. Yamakawa, 1993. A new strategy in fuzzy inference systems and in AI: the selective rules activation (SRA) algorithm. *[Proceedings 1993] Second IEEE International Conference on Fuzzy Systems*, 2(1), pp. 934-937.
- Joo Hyung Yoon, W. A. E. P. N. D. I. C., 2022. Head Injuries in Rock Climbing: A Scoping Review. *Wilderness & Environmental Medicine*, 3(4), pp. 479-487.
- M. Andrić, F. R. a. F. Z., 2022. Sensor-Based Activity Recognition and Performance Assessment in Climbing: A Review. *IEEE Access*, 10(1), pp. 108583-108603.
- Reddy, P. V. S., 2021. Generalized Fuzzy Logic with twofold fuzzy set: Learning through Neural Net and Application to Business Intelligence. *2021 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, pp. 1-5.
- Selvachandran, G., 2021. A New Design of Mamdani Complex Fuzzy Inference System for Multiattribute Decision Making Problems. *IEEE Transactions on Fuzzy Systems*, 29(4), pp. 716-730.
- Uk.mathworks.com, 2018. *What Is Fuzzy Logic?*. [Online] Available at: <https://uk.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html> [Accessed 3 December 2022].
- Volker Schöffl, C. L. D. P., 2016. The “Heel Hook”—A Climbing-Specific Technique to Injure the Leg. *Wilderness & Environmental Medicine*, 27(2), pp. 294-301.
- Wood, R., 2008. *PFT Pull-Up Test*. [Online] Available at: <https://www.topendsports.com/testing/tests/pull-up-pft.htm> [Accessed 9 December 2022].
- Zadeh, L. A., 1996. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2), pp. 103-111.

Appendix:

Figure 1 – Rule base for Climber’s Skill

Figure 2 – Wall Difficulty Rule base

Figure 3 – Success of climber Rule base

Figure 4 - Variable declarations

Figure 5 - Fuzzy set distributions

Revised Rule Base: Figure 6 – Climber’s Skill Rule base

Figure 7 Success of climber Rule Base

Figure 8 – Data sets for testing

Figure 9 – Expected Outcomes

Figure 10 – Defuzzification for the climber’s skill

Figure 11 - Defuzzification for the Wall Difficulty

Figure 12 - Initial Defuzzification for the Success Of Climber

Figure 13 – Climber’s skill expected outcome vs actual outcome

Figure 14 – Wall Difficulty expected outcome vs actual outcome

Figure 15 – Climber’s skill expected outcome vs actual outcome

Figure 16 – Updated Wall difficulty FIS outputs

Figure 17 - updated Wall difficulty FIS vs expected results

Figure 18 – Initial Success of climber

Figure 19 – Success of climber vs expected output

Figure 20 – Updated Success of climber

Figure 22 – updated Success of climber vs expected output

Figure 1 – Rule base for Climber’s Skill

Figure 2 – Wall Difficulty Rule base

Figure 3 – Success of climber Rule base

'1. If (Climbers Skill is Very Low) and (Wall Difficulty is Very Low) then (Success of climber is Very Small) (1)
'2. If (Climbers Skill is Low) and (Wall Difficulty is Very Low) then (Success of climber is Very Small) (1)
'3. If (Climbers Skill is Average) and (Wall Difficulty is Very Low) then (Success of climber is Small) (1)
'4. If (Climbers Skill is High) and (Wall Difficulty is Very Low) then (Success of climber is Moderate) (1)
'5. If (Climbers Skill is Very High) and (Wall Difficulty is Very Low) then (Success of climber is Moderate) (1)
'6. If (Climbers Skill is Very Low) and (Wall Difficulty is Low) then (Success of climber is Very Small) (1)
'7. If (Climbers Skill is Very Low) and (Wall Difficulty is Average) then (Success of climber is Small) (1)
'8. If (Climbers Skill is Very Low) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'9. If (Climbers Skill is Very Low) and (Wall Difficulty is Very High) then (Success of climber is Moderate) (1)
'10. If (Climbers Skill is Low) and (Wall Difficulty is Low) then (Success of climber is Small) (1)
'11. If (Climbers Skill is Average) and (Wall Difficulty is Low) then (Success of climber is Moderate) (1)
'12. If (Climbers Skill is High) and (Wall Difficulty is Low) then (Success of climber is Moderate) (1)
'13. If (Climbers Skill is Very High) and (Wall Difficulty is Low) then (Success of climber is High) (1)
'14. If (Climbers Skill is Low) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'15. If (Climbers Skill is Low) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'16. If (Climbers Skill is Low) and (Wall Difficulty is Very High) then (Success of climber is High) (1)
'17. If (Climbers Skill is Average) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'18. If (Climbers Skill is Average) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'19. If (Climbers Skill is High) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'20. If (Climbers Skill is Very High) and (Wall Difficulty is Average) then (Success of climber is High) (1)
'21. If (Climbers Skill is Average) and (Wall Difficulty is Very High) then (Success of climber is High) (1)
'22. If (Climbers Skill is High) and (Wall Difficulty is High) then (Success of climber is High) (1)
'23. If (Climbers Skill is High) and (Wall Difficulty is Very High) then (Success of climber is High) (1)
'24. If (Climbers Skill is Very High) and (Wall Difficulty is High) then (Success of climber is High) (1)
'25. If (Climbers Skill is Very High) and (Wall Difficulty is Very High) then (Success of climber is Very High) (1)

Figure 4 - Variable declarations

```

a=newtis('Climbers Skill');

a=addvar(a,'input','BMI',[12 40]);
% Populating the 1st input variable with membership functions
a = addmf(a, 'input', 1, 'Very High', 'trapmf', [12 12 15 18]);
a = addmf(a, 'input', 1, 'High', 'trimf', [15 17 21]);
a = addmf(a, 'input', 1, 'Average', 'trimf', [18 21 24 ]);
a = addmf(a, 'input', 1, 'Low', 'trimf', [23 27 31]);
a = addmf(a, 'input', 1, 'Very Low', 'trapmf', [29 33 40 40]);

a=addvar(a,'input','Experience',[0 20]);
% Populating the 2st input variable with membership functions
a = addmf(a, 'input', 2, 'Very Inexperienced', 'trapmf', [0 0 1 2]);
a = addmf(a, 'input', 2, 'Inexperienced', 'trimf', [1 3 5]);
a = addmf(a, 'input', 2, 'Average', 'trimf', [4 7 9 ]);
a = addmf(a, 'input', 2, 'Experienced', 'trimf', [7 10 13]);
a = addmf(a, 'input', 2, 'Very Experienced', 'trapmf', [10 15 20 20]);

a=addvar(a,'input','Strength',[0 50]);
% Populating the 3st input variable with membership functions
a = addmf(a, 'input', 3, 'Very Weak', 'trapmf', [0 0 1 3]);
a = addmf(a, 'input', 3, 'Weak', 'trimf', [2 4 6]);
a = addmf(a, 'input', 3, 'Average', 'trimf', [5 7 13 ]);
a = addmf(a, 'input', 3, 'Strong', 'trimf', [10 15 23 ]);
a = addmf(a, 'input', 3, 'Very Strong', 'trapmf', [18 30 50 50]);

a=addvar(a,'input','Ape Index',[-10 10]);
% Populating the 4st input variable with membership functions
a = addmf(a, 'input', 4, 'Very Small', 'trapmf', [-10 -10 -8 -5]);
a = addmf(a, 'input', 4, 'Small', 'trimf', [-7 -4 -2]);
a = addmf(a, 'input', 4, 'Average', 'trimf', [-3 0 3]);
a = addmf(a, 'input', 4, 'Long', 'trimf', [2 4 6]);
a = addmf(a, 'input', 4, 'Very Long', 'trapmf', [3 8 10 10]);

a=addvar (a,'output','Climbers Skill',[0 100]);
% Populating the 1st output variable with membership functions
a = addmf(a, 'output', 1, 'Very Low', 'trapmf', [0 0 10 20]);
a = addmf(a, 'output', 1, 'Low', 'trimf', [15 25 40]);
a = addmf(a, 'output', 1, 'Average', 'trimf', [30 50 70]);
a = addmf(a, 'output', 1, 'High', 'trimf', [60 70 80]);
a = addmf(a, 'output', 1, 'Very High', 'trapmf', [80 90 100 100]);
b=newfis('Difficulty of Wall');

b=addvar(b,'input',' Wall Height',[0 4.5]);
% Populating the 1st input variable with membership functions
b = addmf(b, 'input', 1, 'Very Small', 'trapmf', [0 0 1 1.5]);
b = addmf(b, 'input', 1, 'Small', 'trimf', [1.25 1.75 2.5]);
b = addmf(b, 'input', 1, 'Average', 'trimf', [2.25 3 3.5 ]);
b = addmf(b, 'input', 1, 'Tall', 'trimf', [3 3.75 4]);
b = addmf(b, 'input', 1, 'Very Tall', 'trapmf', [3.5 4 4.5 4.5]);

```

```

b=addvar(b,'input','Hold Distance',[0 300]);
% Populating the 2st input variable with membership functions
b = addmf(b, 'input', 2, 'Very Small', 'trapmf', [0 0 60 120]);
b = addmf(b, 'input', 2, 'Small', 'trimf', [60 130 150]);
b = addmf(b, 'input', 2, 'Average', 'trimf', [145 170 195 ]);
b = addmf(b, 'input', 2, 'Tall', 'trimf', [160 180 200]);
b = addmf(b, 'input', 2, 'Very Tall', 'trapmf', [200 250 300 300]);

b=addvar(b,'input','Incline',[-90 90]);
% Populating the 3st input variable with membership functions
b = addmf(b, 'input', 3, 'Flat', 'trapmf', [-90 -90 -75 -50 ]);
b = addmf(b, 'input', 3, 'Inclined', 'trimf', [-60 -45 -10]);
b = addmf(b, 'input', 3, 'Straight', 'trimf', [-30 0 30 ]);
b = addmf(b, 'input', 3, 'Very Inclined', 'trimf', [10 45 60]);
b = addmf(b, 'input', 3, 'Overhang', 'trapmf', [60 75 90 90]);

b=addvar (b,'output','Wall Difficulty',[0 100]);
% Populating the 1st output variable with membership functions
b = addmf(b, 'output', 1, 'Very Low', 'trapmf', [0 0 10 20]);
b = addmf(b, 'output', 1, 'Low', 'trimf', [15 25 40]);
b = addmf(b, 'output', 1, 'Average', 'trimf', [30 50 70]);
b = addmf(b, 'output', 1, 'High', 'trimf', [60 70 80]);
b = addmf(b, 'output', 1, 'Very High', 'trapmf', [80 90 100 100]);

c=newfis('Success of climber');

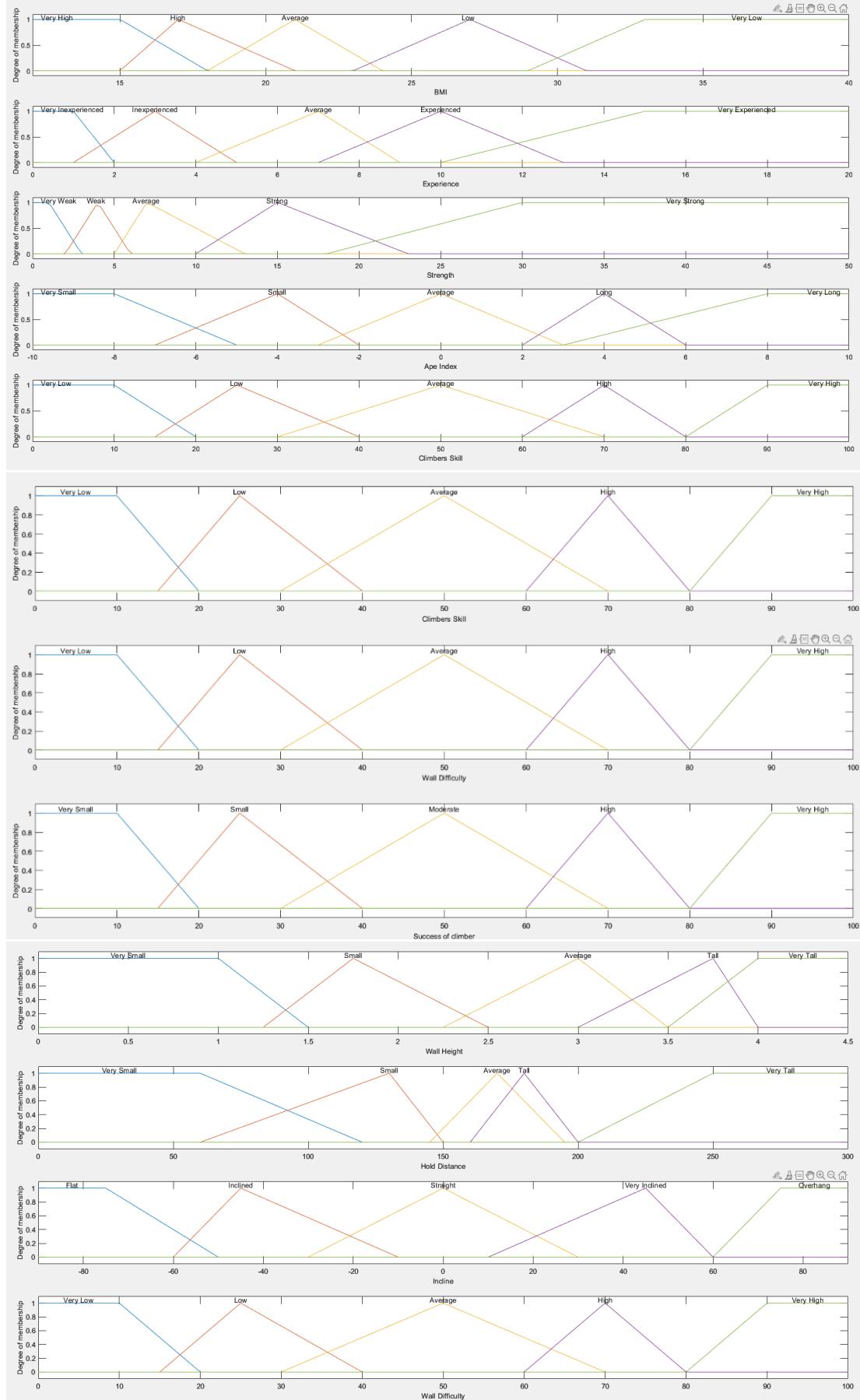
c=addvar(c,'input','Climbers Skill',[0 100]);
% Populating the 1st output variable with membership functions
c = addmf(c, 'input', 1, 'Very Low', 'trapmf', [0 0 10 20]);
c = addmf(c, 'input', 1, 'Low', 'trimf', [15 25 40]);
c = addmf(c, 'input', 1, 'Average', 'trimf', [30 50 70]);
c = addmf(c, 'input', 1, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'input', 1, 'Very High', 'trapmf', [80 90 100 100]);

c=addvar(c,'input','Wall Difficulty',[0 100]);
% Populating the 2st output variable with membership functions
c = addmf(c, 'input', 2, 'Very Low', 'trapmf', [0 0 10 20]);
c = addmf(c, 'input', 2, 'Low', 'trimf', [15 25 40]);
c = addmf(c, 'input', 2, 'Average', 'trimf', [30 50 70]);
c = addmf(c, 'input', 2, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'input', 2, 'Very High', 'trapmf', [80 90 100 100]);

c=addvar(c,'output','Success of climber',[0 100]);
% Populating the 3st output variable with membership functions
c = addmf(c, 'output', 1, 'Very Small', 'trapmf', [0 0 10 20]);
c = addmf(c, 'output', 1, 'Small', 'trimf', [15 25 40]);
c = addmf(c, 'output', 1, 'Moderate', 'trimf', [30 50 70]);
c = addmf(c, 'output', 1, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'output', 1, 'Very High', 'trapmf', [80 90 100 100]);

```

Figure 5 - Fuzzy set distributions



Revised Rule Base: Figure 6 – Climber’s Skill Rule base

Command Window

Figure 7 Success of climber Rule Base

```
'1. If (Climbers Skill is Very Low) and (Wall Difficulty is Very Low) then (Success of climber is Very High) (1)
'2. If (Climbers Skill is Low) and (Wall Difficulty is Very Low) then (Success of climber is Very High) (1)
'3. If (Climbers Skill is Average) and (Wall Difficulty is Very Low) then (Success of climber is High) (1)
'4. If (Climbers Skill is High) and (Wall Difficulty is Very Low) then (Success of climber is Moderate) (1)
'5. If (Climbers Skill is Very High) and (Wall Difficulty is Very Low) then (Success of climber is Moderate) (1)
'6. If (Climbers Skill is Very Low) and (Wall Difficulty is Low) then (Success of climber is Very High) (1)
'7. If (Climbers Skill is Very Low) and (Wall Difficulty is Average) then (Success of climber is High) (1)
'8. If (Climbers Skill is Very Low) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'9. If (Climbers Skill is Very Low) and (Wall Difficulty is Very High) then (Success of climber is Moderate) (1)
'10. If (Climbers Skill is Low) and (Wall Difficulty is Low) then (Success of climber is High) (1)
'11. If (Climbers Skill is Average) and (Wall Difficulty is Low) then (Success of climber is Moderate) (1)
'12. If (Climbers Skill is High) and (Wall Difficulty is Low) then (Success of climber is Moderate) (1)
'13. If (Climbers Skill is Very High) and (Wall Difficulty is Low) then (Success of climber is Small) (1)
'14. If (Climbers Skill is Low) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'15. If (Climbers Skill is Low) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'16. If (Climbers Skill is Low) and (Wall Difficulty is Very High) then (Success of climber is Small) (1)
'17. If (Climbers Skill is Average) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'18. If (Climbers Skill is Average) and (Wall Difficulty is High) then (Success of climber is Moderate) (1)
'19. If (Climbers Skill is High) and (Wall Difficulty is Average) then (Success of climber is Moderate) (1)
'20. If (Climbers Skill is Very High) and (Wall Difficulty is Average) then (Success of climber is Small) (1)
'21. If (Climbers Skill is Average) and (Wall Difficulty is Very High) then (Success of climber is Small) (1)
'22. If (Climbers Skill is High) and (Wall Difficulty is High) then (Success of climber is Small) (1)
'23. If (Climbers Skill is High) and (Wall Difficulty is Very High) then (Success of climber is Small) (1)
'24. If (Climbers Skill is Very High) and (Wall Difficulty is High) then (Success of climber is Small) (1)
'25. If (Climbers Skill is Very High) and (Wall Difficulty is Very High) then (Success of climber is Very Small) (1)
```

Figure 8 – Data sets for testing

BMI	Experience	Strength	Ape Index	Wall Height	Hold Distance	Incline
16	3	5	2	1	30	-90
12	0	0	-10	4.5	300	90
24	1	15	4	0.5	50	0
35	0	0	-4	3.5	150	15
20	10	35	5	2.2	240	-30
15	15	45	8	4	200	45
21	8	20	4	3	40	15
30	0	40	-8	2.4	120	30
28	20	30	0	0.3	50	40
15	2	25	5	1.3	30	-5
24	5	15	-5	2	250	-15
18	12	23	8	1	300	-10
34	16	10	-2	3.3	100	0
31	3	7	1	2.8	20	60
26	0	12	-8	1.7	10	90
14	8	17	3	2.9	80	-60
22	15	33	0	3.2	140	-70
38	0	2	6	1.4	260	-30
13	20	50	10	2.9	170	80

Figure 9 – Expected Outcomes

Data number	Climbers Skill Expected Outcome	Wall Difficulty expected outcome	Success of climber expected outcome
1	Low	Very Low	Very High
2	Very Low	Very High	
3	High	Low	Average
4	Low	Average	Average
5	High	Average	High
6	High	High	Average
7	High	Average	Average

8	Average	Average	Average
9	Very High	Low	Very High
10	Average	Low	Average
11	Average	Average	Average
12	Very High	Average	High
13	High	Average	Average
14	Average	Average	Average
15	Low	Average	Average
16	Average	Low	Average
17	Very High	Low	Very high
18	Average	Average	Average
19	High	Average	Average

Figure 10 – Defuzzification for the climber's skill

Centroid	MoM	LoM	SoM	Bisector
27.125	27	35	19	27
7.516129	5	10	0	7
56.56158	70	77	63	57
26.66667	25	25	25	26
80.53691	70	75	65	79
70	70	70	70	70
77.2364	70	76	64	74
50	50	65	35	50
91.99203	94	100	88	92
50	50	60	40	50
50	50	65	35	50
91.21622	92	100	84	91
70	70	76	64	70
50	50	60	40	50
27.04167	26.5	34	19	27
54.59753	50	60	40	54
91.81633	93.5	100	87	92
50	50	60	40	50
70	70	70	70	70

Figure 11 - Defuzzification for the Wall Difficulty

Centroid	MoM	LoM	SoM	Bisector
7.516129	5	10	0	7
92.48387	95	100	90	93
26.66667	25	25	25	26
54.16134	50	66	34	54
50	50	62	38	50
50	50	50	50	50
50	50	60	40	50
50	50	66	34	50
26.69938	25.5	27	24	26
27.04167	26.5	34	19	27
61.43616	70	75	65	66
50	50	56	44	50
50	50	61	39	50
50	50	50	50	50
50	50	52	48	50
27.04167	26.5	34	19	27
26.9395	26	32	20	27
50	50	64	36	50
54.95029	50	52	48	53

Figure 12 - Initial Defuzzification for the Success Of Climber

E	F	G	H	I
Centroid	MoM	LoM	SoM	Bisector
92.21223	94.5	100	84	92
50	50	50	50	50
50	50	64	37	50
50	50	58	64	50
27.5	50	62	38	50
50	50	50	50	50
50	50	62	38	50
50	50	32	64	50
26.68738	25	27	23	26
50	50	62	38	50
50	50	32	37	50
26.66667	25	29	19	26
50	50	62	38	50
50	50	60	40	50
50	50	62	38	50
50	50	62	38	50
26.69398	25.5	32	20	26
50	50	62	36	50
50	50	52	48	50

Figure 13 – Climber's skill expected outcome vs actual outcome

Data number	Centroid	Mom	Lom	Som	Bisector
1	True	True	True	True	True
2	True	True	True	True	True
3	False	True	True	False	False
4	True	True	True	True	True
5	False	True	True	True	True
6	True	True	True	True	True
7	True	True	True	True	True
8	True	True	True	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	True	True	False	True
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17	True	True	True	True	True
18	True	True	True	True	True
19	True	True	True	True	True

Figure 14 – Wall Difficulty expected outcome vs actual outcome Blue means No Rule was activated

Data number	Centroid	Mom	Lom	Som	Bisector
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	False	False	True
5	True	True	True	True	True
6	False	False	False	False	False
7	True	True	True	True	True
8	True	True	True	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	False	False	False	False
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17	True	True	True	True	True
18	True	True	False	True	True
19	False	False	False	False	False

Figure 15 – Climber's skill expected outcome vs actual outcome

Data number	Centroid	MoM	LoM	SoM	Bisector
1	True	True	True	True	True
2	True	True	True	True	True
3	False	True	True	True	False
4	True	True	True	True	True
5	False	True	True	True	True
6	True	True	True	True	True
7	True	True	True	True	True
8	False	True	True	False	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	True	True	False	True
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17	True	True	True	True	True
18	False	True	True	True	True
19	True	True	True	True	True

Figure 16 – Updated Wall difficulty FIS outputs

Wall Height	Hold Distance	Incline	Centroid	MoM	LoM	SoM	Bisector
1	30	-90	7.516129	5	10	0	7
4.5	300	90	90.98148	95	100	90	91
0.5	50	0	26.66667	25	25	25	26
3.5	150	15	54.16134	50	66	34	54
2.2	240	-30	50	50	62	38	50
4	200	45	70	70	70	70	70
3	40	15	50	50	60	40	50
2.4	120	30	50	50	64	36	50
0.3	50	40	26.69938	25.5	27	24	26
1.3	30	-5	27.04167	26.5	34	19	27
2	250	-15	61.43616	70	75	65	66
1	300	-10	50	50	56	44	50
3.3	100	0	50	50	58	42	50
2.8	20	60	50	50	62	38	50
1.7	10	90	50	50	52	48	50
2.9	80	-60	27.04167	26.5	34	19	27
3.2	140	-70	27.04167	26.5	33	20	27
1.4	260	-30	50	50	64	36	50
2.9	170	80	50	50	55	45	50

Figure 17 - updated Wall difficulty FIS vs expected results

Data number	Centroid	MoM	LoM	SoM	Bisector
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	False	False	True
5	True	True	True	True	True
6	True	True	True	True	True
7	True	True	True	True	True
8	True	True	False	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	False	False	False	False
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17	True	True	True	True	True
18	True	True	False	True	True
19	True	True	True	True	True

Figure 18 – Initial Success of climber

Centroid	MoM	LoM	SoM	Bisector
92.21223	94.5	100	84	92
50	50	50	50	50
50	50	64	37	50
50	50	58	64	50
27.5	50	62	38	50
26.66667	25	25	25	26
50	50	62	38	50
50	50	62	36	50
26.68738	25	27	23	26
50	50	62	38	50
50	50	32	37	50
26.66667	25	29	19	26
50	50	62	38	50
50	50	62	38	50
50	50	62	38	50
26.69674	25	33	20	26
50	50	62	36	50
50	50	55	45	50

Figure 19 – Success of climber vs expected output

Data number	Centroid	MoM	LoM	SoM	Bisector
1	True	True	True	True	True
2	False	False	False	False	False
3	True	True	True	True	True
4	True	True	True	True	True
5	False	False	False	False	False
6	False	False	False	False	False
7	True	True	True	True	True
8	True	True	True	True	True
9	False	False	False	False	False
10	True	True	True	True	True
11	True	True	True	True	True
12	False	False	False	False	False
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17s	False	False	False	False	False
18s	True	True	True	True	True
19s	True	True	True	True	True

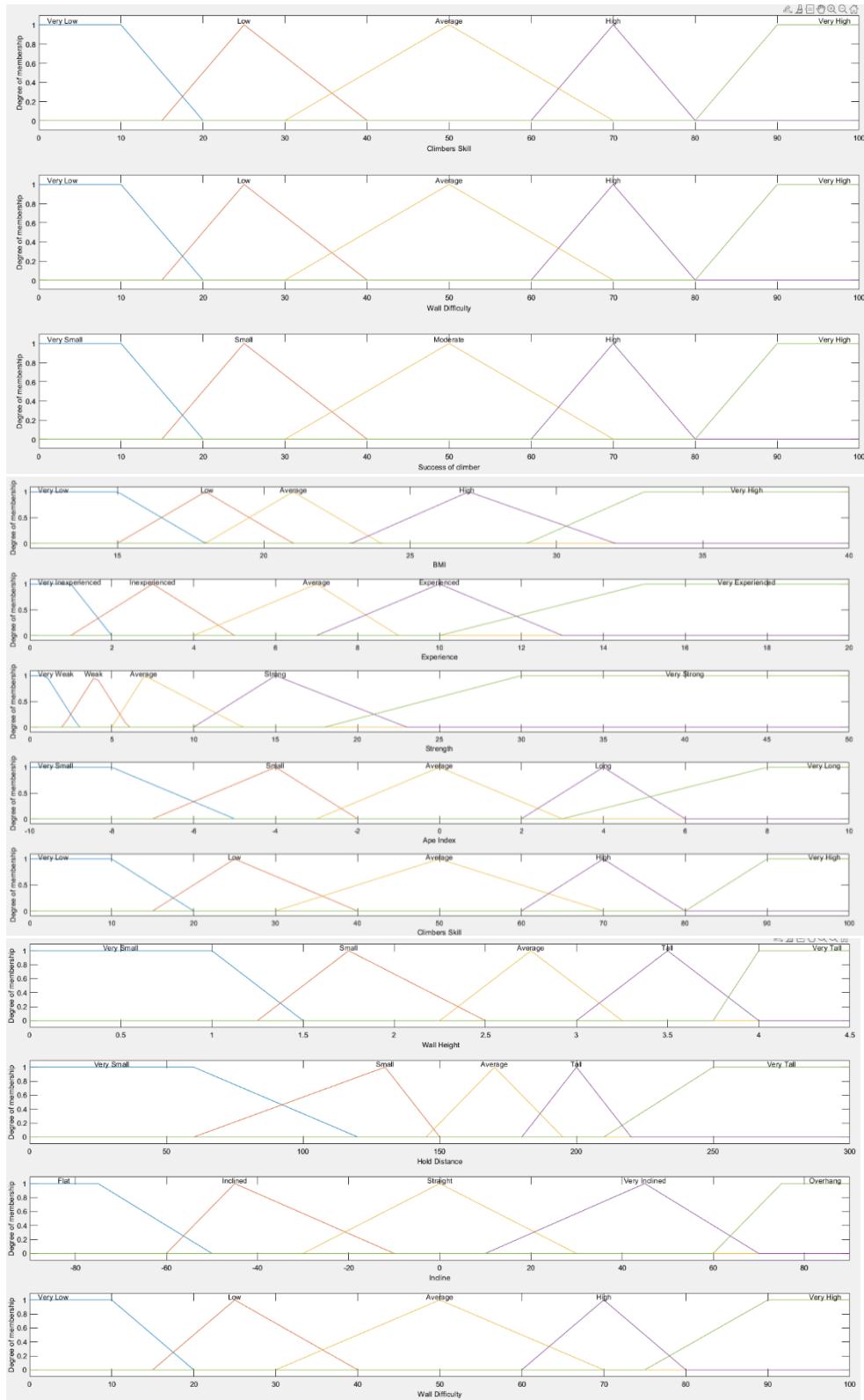
Figure 20 – Updated Success of climber

Centroid	MoM	LoM	SoM	Bisector
50	50	63	38	50
7.516129	5	10	0	7
50	70	77	37	50
50	50	30	38	50
70	50	62	38	50
50	50	50	50	50
50	50	62	38	50
50	50	62	36	50
92.26693	95	100	88	93
	50	50	62	38
	50	50	60	19
	70	70	73	64
	50	50	62	38
	50	50	62	38
	50	50	62	38
92.22315	94.5	100	85	92
	50	50	62	36
	50	50	55	45

Figure 21 – updated Success of climber vs expected output

Data number	Centroid	Mom	Lom	Som	Bisector
1	True	True	True	True	True
2	True	False	False	False	True
3	True	True	False	True	True
4	True	False	False	False	False
5	True	True	True	True	True
6	True	True	True	True	True
7	True	True	True	False	True
8	True	True	True	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	True	True	False	True
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17s	True	True	True	True	True
18s	True	True	True	True	True
19s	True	True	True	True	True

Figure 22 – Final Rule base, variable declarations, and Fuzzy sets



```

a=newfis('Climbers Skill','DefuzzificationMethod','centroid');

a=addvar(a,'input','BMI',[12 40]);
% Populating the 1st input variable with membership functions
a = addmf(a, 'input', 1, 'Very Low', 'trapmf', [12 12 15 18]);
a = addmf(a, 'input', 1, 'Low', 'trimf', [15 18 21]);
a = addmf(a, 'input', 1, 'Average', 'trimf', [18 21 24 ]);
a = addmf(a, 'input', 1, 'High', 'trimf', [23 27 32]);
a = addmf(a, 'input', 1, 'Very High', 'trapmf', [29 33 40 40]);

a=addvar(a,'input','Experience',[0 20]);
% Populating the 2st input variable with membership functions
a = addmf(a, 'input', 2, 'Very Inexperienced', 'trapmf', [0 0 1 2]);
a = addmf(a, 'input', 2, 'Inexperienced', 'trimf', [1 3 5]);
a = addmf(a, 'input', 2, 'Average', 'trimf', [4 7 9 ]);
a = addmf(a, 'input', 2, 'Experienced', 'trimf', [7 10 13]);
a = addmf(a, 'input', 2, 'Very Experienced', 'trapmf', [10 15 20 20]);

a=addvar(a,'input','Strength',[0 50]);
% Populating the 3st input variable with membership functions
a = addmf(a, 'input', 3, 'Very Weak', 'trapmf', [0 0 1 3]);
a = addmf(a, 'input', 3, 'Weak', 'trimf', [2 4 6]);
a = addmf(a, 'input', 3, 'Average', 'trimf', [5 7 13 ]);
a = addmf(a, 'input', 3, 'Strong', 'trimf', [10 15 23]);
a = addmf(a, 'input', 3, 'Very Strong', 'trapmf', [18 30 50 50]);

a=addvar(a,'input','Ape Index',[ -10 10]);
% Populating the 4st input variable with membership functions
a = addmf(a, 'input', 4, 'Very Small', 'trapmf', [-10 -10 -8 -5]);
a = addmf(a, 'input', 4, 'Small', 'trimf', [-7 -4 -2]);
a = addmf(a, 'input', 4, 'Average', 'trimf', [-3 0 3]);
a = addmf(a, 'input', 4, 'Long', 'trimf', [2 4 6]);
a = addmf(a, 'input', 4, 'Very Long', 'trapmf', [3 8 10 10]);

a=addvar(a,'output','Climbers Skill',[0 100]);
% Populating the 1st output variable with membership functions
a = addmf(a, 'output', 1, 'Very Low', 'trapmf', [0 0 10 20]);
a = addmf(a, 'output', 1, 'Low', 'trimf', [15 25 40]);
a = addmf(a, 'output', 1, 'Average', 'trimf', [30 50 70]);
a = addmf(a, 'output', 1, 'High', 'trimf', [60 70 80]);
a = addmf(a, 'output', 1, 'Very High', 'trapmf', [80 90 100 100]);
b=newfis('Difficulty of Wall','DefuzzificationMethod','centroid');

b=addvar(b,'input',' Wall Height',[0 4.5]);
% Populating the 1st input variable with membership functions
b = addmf(b, 'input', 1, 'Very Small', 'trapmf', [0 0 1 1.5]);
b = addmf(b, 'input', 1, 'Small', 'trimf', [1.25 1.75 2.5]);
b = addmf(b, 'input', 1, 'Average', 'trimf', [2.25 2.75 3.25 ]);
b = addmf(b, 'input', 1, 'Tall', 'trimf', [3 3.5 4]);
b = addmf(b, 'input', 1, 'Very Tall', 'trapmf', [3.75 4 4.5 4.5]);

b=addvar(b,'input','Hold Distance',[0 300]);
% Populating the 2st input variable with membership functions
b = addmf(b, 'input', 2, 'Very Small', 'trapmf', [0 0 60 120]);
b = addmf(b, 'input', 2, 'Small', 'trimf', [60 130 150]);
b = addmf(b, 'input', 2, 'Average', 'trimf', [145 170 195 ]);
b = addmf(b, 'input', 2, 'Tall', 'trimf', [180 200 220]);
b = addmf(b, 'input', 2, 'Very Tall', 'trapmf', [210 250 300 300]);

b=addvar(b,'input','Incline',[ -90 90]);
% Populating the 3st input variable with membership functions
b = addmf(b, 'input', 3, 'Flat', 'trapmf', [-90 -90 -75 -50 ]);
b = addmf(b, 'input', 3, 'Inclined', 'trimf', [-60 -45 -10]);
b = addmf(b, 'input', 3, 'Straight', 'trimf', [-30 0 30 ]);
b = addmf(b, 'input', 3, 'Very Inclined', 'trimf', [10 45 70]);
b = addmf(b, 'input', 3, 'Overhang', 'trapmf', [60 75 90 90]);

b=addvar(b,'output','Wall Difficulty',[0 100]);
% Populating the 1st output variable with membership functions
b = addmf(b, 'output', 1, 'Very Low', 'trapmf', [0 0 10 20]);
b = addmf(b, 'output', 1, 'Low', 'trimf', [15 25 40]);
b = addmf(b, 'output', 1, 'Average', 'trimf', [30 50 70]);
b = addmf(b, 'output', 1, 'High', 'trimf', [60 70 80]);
b = addmf(b, 'output', 1, 'Very High', 'trapmf', [75 90 100 100]);

```

```

c=newfis('Success of climber','DefuzzificationMethod','centroid');

c=addvar(c,'input','Climbers Skill',[0 100]);
% Populating the 1st output variable with membership functions
c = addmf(c, 'input', 1, 'Very Low', 'trapmf', [0 0 10 20]);
c = addmf(c, 'input', 1, 'Low', 'trimf', [15 25 40]);
c = addmf(c, 'input', 1, 'Average', 'trimf', [30 50 70]);
c = addmf(c, 'input', 1, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'input', 1, 'Very High', 'trapmf', [80 90 100 100]);

c=addvar(c,'input','Wall Difficulty',[0 100]);
% Populating the 2st output variable with membership functions
c = addmf(c, 'input', 2, 'Very Low', 'trapmf', [0 0 10 20]);
c = addmf(c, 'input', 2, 'Low', 'trimf', [15 25 40]);
c = addmf(c, 'input', 2, 'Average', 'trimf', [30 50 70]);
c = addmf(c, 'input', 2, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'input', 2, 'Very High', 'trapmf', [80 90 100 100]);

c=addvar(c,'output','Success of climber',[0 100]);
% Populating the 3st output variable with membership functions
c = addmf(c, 'output', 1, 'Very Small', 'trapmf', [0 0 10 20]);
c = addmf(c, 'output', 1, 'Small', 'trimf', [15 25 40]);
c = addmf(c, 'output', 1, 'Moderate', 'trimf', [30 50 70]);
c = addmf(c, 'output', 1, 'High', 'trimf', [60 70 80]);
c = addmf(c, 'output', 1, 'Very High', 'trapmf', [80 90 100 100]);

```