Aircraft Landing Control Using Fuzzy Logic and Neural Networks

Elvira Lakovic
Intelligent Embedded Systems
elc10001@student.mdh.se

Damir Lotinac
Intelligent Embedded Systems
dlc10001@student.mdh.se

ABSTRACT

Fuzzy logic is well known and widely used today in control systems. Neural networks are also used in a wide variety of systems. Combination of these two different approaches should have many beneficial properties for solving many different problems, but still this combination is not used as much as it could be used. In this paper we explore possibilities for application of two different fuzzy and one neuro-fuzzy controller for aircraft landing problem. Simplified model of aircraft landing problem is used. The description of the model we utilized for this paper has already been published, but we constructed more than just one type of fuzzy controller and tested them. The tests were conducted on simulation, and in this paper we provide analysis of these results.

Keywords
Fuzzy controller, Neuro-Fuzzy controller, ANFIS controller, Aircraft landing controller

1. INTRODUCTION

Fuzzy systems can utilize approximate reasoning and can be used in decision making and mechanical control systems like air conditioning, cars, ships, robotic arms, in industrial control processes and many other types of application. Fuzzy logic is applicable in many different areas, for vide variety of problems. Engineering [1], problems in medicine, biology [2] are just a few fields where fuzzy logic is successfully applied already. Usage of fuzzy systems is recommended for very complex processes, where simple mathematical model cannot be derived. That is one major advantage of fuzzy logic, because the designer does not need to know everything about the system before starting the work, and simplicity provides solutions to what were once unsolvable problems. Fuzzy logic defines operations for modifying sets and allows elements to partially belong to the set, which offers a lot more flexibility. The reason for the popularity of fuzzy logic lies in the application of fuzzy sets, which give much greater flexibility than a regular set of numbers.

On the other hand, the popularity of neural networks is reflected in learning opportunities through training, which enables the system to adjust its weights in order to achieve better results. Inspired by biological nervous system, many researches explored neural networks; approach for information processing that is not based on algorithms. Neural networks are modeling the brain as a continuous nonlinear dynamic system in connected architecture that should mimic the mechanism of the brain in the simulation of intelligent behavior. Such a connection is replacing a symbolic representation of the structure with distributed representation in the form of weight between the massive set of interconnected neurons (or processing units). They do not require critical decision-making in their algorithms.

Both systems based only on fuzzy logic and systems based purely on neural networks, have their advantages and their limitations in different aspects. In this paper we will explore the fuzzy logic based system for aircraft landing described in [1] and improvement possibilities. Aircraft landing is the process during which the aircraft descends from certain altitude, and touches the ground. The aircraft landing system is supposed to control the vertical speed during the aircraft landing process, so that the aircraft touches the ground "smoothly" but descends rapidly. If the aircraft touches the ground with high negative vertical speed, that is considered a crash landing, and should be avoided by the controller. The model suggested in [1] is a simplified model of aircraft landing and that model will be the basis for designing our fuzzy controllers. The aim of this work is to evaluate the behavior of two different types of fuzzy controllers, and the behavior of the ANFIS (Adaptive Neuro-Fuzzy Inference System) controller.

In this paper Section 2 contains description of fuzzy and neuro-fuzzy inference systems that we used. Section 3 describes the work related to fuzzy and neuro-fuzzy controllers and their application in aircraft landing problem. In Section 4 we explain details about fuzzy controller design, while in Section 5 we provide information on our ANFIS solution. Section 6 explains how we trained and tested the ANFIS solution, and how the datasets for these purposes were generated. Section 7 contains the results of simulations, for all three different controllers using different parameters. In Section 8 is a short summary and conclusions based on results from Section 7.

2. FUZZY AND ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS

2.1 Fuzzy Inference System

Many types of fuzzy inference systems exist and are used today. In this section we will briefly describe two types of fuzzy inference systems that have been used to model the described aircraft landing system and whose performance is evaluated in this paper. These two types are: Mamdani-type fuzzy inference system and Sugeno-type fuzzy inference system. In this paper, we also use and evaluate performance and modeling features of ANFIS.

The inputs to fuzzy inference system are crisp and are in a certain range. Inputs are then fuzzified using membership functions. Rules are then evaluated using fuzzy reasoning and "and" and/or "or" and "implication" operators. The combination of rules is done using aggregation process. In this paper we used three types of aggregation for Mamdani-type fuzzy controller, which are supported by Matlab [3]:

- "Maximum"
- "Probabilistic or"
- "Sum" (The sum of output sets of each rule);

Output of a controller that is used to solve a problem like aircraft landing should be a crisp number. Since the aggregation step gives us fuzzy set as the output, that output should be defuzzified. In defuzzification step we use three different methods supported by Matlab for Mamdani-type fuzzy controller [3]:

- Center of area or centroid (COA)
- Bisector
- Middle of maximum (MOM)

Mamdani-type fuzzy inference system used in this paper is supposed to have output functions as fuzzy sets. These sets are than subject to defuzzification process. Of course it is possible to use a singleton fuzzy set, which than represents a single value, for these type of sets, defuzzification is not needed, therefore these sets are sometimes more efficient than distributed fuzzy sets.

While Mamdani uses distributed fuzzy sets on the output, Sugeno uses exact numbers on the output. Sugeno uses either linear or constant output types. The linear type should give "smoother" results but requires more tuning. For the Sugeno-type controller we compared two different defuzzification methods which are supported by Matlab for Sugeno-type fuzzy controller [10]:

- Weighted sum ("wtsum")
- Weighted average ("wtaver")

2.2 Adaptive Neuro-Fuzzy Inference System

Ordinary fuzzy inference systems use predetermined set of values which specify membership functions. Usually these values are manually typed in and are fixed, thus the inference system depends on user's understanding and interpretation of variables in the model. Combination of neural networks and fuzzy inference systems enables adaptation of fuzzy inference systems.

ANFIS which is used in this paper, by its structure, is equivalent to Sugeno-type inference system (Figure 1.), and it can use two different learning techniques. ANFIS utilizes these learning techniques to adapt the fuzzy inference system to the data that is used during the training. The end result should be the fuzzy inference system that has membership functions which are more suitable as a possible solution to the problem. Backpropagation is a supervised learning method for neural networks. For backpropagation to function properly, activation function of each node should be differentiable. Since convergence with this algorithm is very slow, another algorithm can be used for ANFIS learning. Hybrid algorithm utilizes both least square and error backpropagation method. ANFIS uses learning techniques and has structure of the neural network, but the learning process adapts the fuzzy inference system, which is the product of ANFIS and therefore produces a possible solution to the problem, which is improved using training data. For practical reasons we will not describe ANFIS in more detail here, but more information can be found in [8].

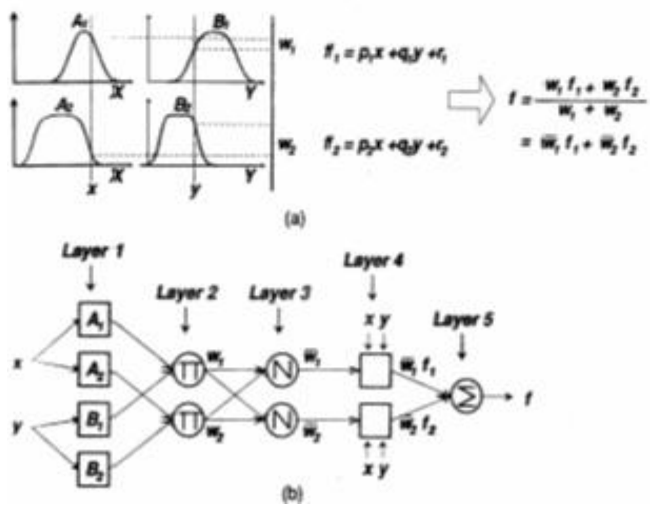

Figure 1. (a) Two-input first-order Sugeno fuzzy model with two rules; (b) Equivalent ANFIS architecture [8]

3. RELATED WORK

Model of system for aircraft landing described in [1] is the basis for our work presented in this paper. The described system controls aircraft vertical speed during the final leg of the landing. That model does not consider the influence of exterior forces on the aircraft. The vertical speed is manipulated using the control force, which is the output from the fuzzy controller. The inputs are current vertical speed and current height, based on those inputs controller is supposed to produce control force that will land the aircraft. There is no information on performance of different kinds of fuzzy systems designed as suggested by Ross [1].

Different types of fuzzy, neuro-fuzzy and other controllers have been compared as solutions for vide variety of problems. Mallek et al. [4] evaluated PID, Neuro, Neuro-PID and ANFIS-PID controllers for automatic landing. Their model of the problem included different wind patterns. Two levels have been defined: desired and acceptable. ANFIS based controller satisfied desired conditions, which none of the other controllers did. Juang and Chio [5] also evaluated performance and robustness of multilayered fuzzy-neural network as the solution to the similar problem in order to improve performance of conventional automatic landing systems. They used "Backpropagation through Time" algorithm to train the network. Their simulations show that fuzzy systems are capable of improving the safety of conventional automatic landing systems even in hostile environments such as severe air turbulence.

Livchitz et al. [6] constructed an "automated fuzzy-logic-based expert system for unmanned aircraft landing". This fuzzy system controlled three-dimensional flight path, vertical velocity and angular altitudes for safe landing. Using fuzzy logic they combined "elements of human pilot with landing knowledge". This system has been successfully tested both on simulations and in "real-life" aircraft landing, which shows the potential of fuzzy logic in this field.

None of these papers discusses the influence of different defuzzification and aggregation strategies. Larkin [7] evaluates the performance and effectiveness of the model associating it with varied parameters and different defuzzification methods. The defuzzification methods evaluated by Larkin were COA and MOM, but bisector has not been evaluated in that particular paper. Different methods of aggregation have not been evaluated in that paper either.

4. FUZZY CONTROLLER DESIGN

The design of both Mamdani and Sugeno-type fuzzy controller has been done according to [1], this means that fuzzy rules, fuzzy membership functions, have been modeled accordingly. The system proposed in [1] is Mamdani-type fuzzy inference system. Although it is not clearly stated, we can conclude this based on output "control force" membership functions which are all distributed. The simplifications regarding time and mass are also consistent with [1]:

$$\Delta t = 1.0\ (s)$$
$$m = 1.0\ (lb)$$

Since we have these simplifications then vertical velocity is calculated as in [1]:

$$v_{i+1} = v_i + f_i \ (1)$$

Vertical velocity changes only if control force is applied. Negative control force directs the aircraft to the ground while positive control force directs the aircraft from the ground.

Height changes according to [1]:

$$h_{i+1} = h_i + v_i \ (2)$$

If current vertical velocity is negative, than aircraft will descend, if it is positive it will increase altitude, and if it is zero, then aircraft will remain at the same altitude.

All three different types of controllers have two inputs: height in feet and vertical velocity in feet per second. The output is also the same variable for all three types of controllers: control force in pound-force. Control force can change current vertical velocity, and change of vertical velocity, changes rate of descent or climb, thus affects current height (Figures: 2, 3).

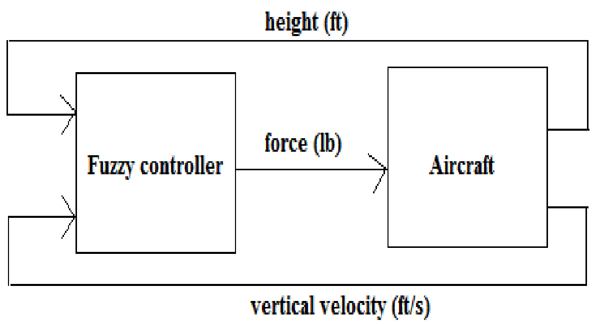

Figure 2. Fuzzy aircraft landing control system

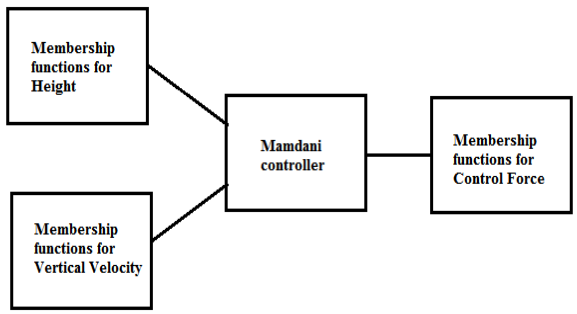

Figure 3. Inputs and output for Mamdani-type fuzzy controller

4.1 Membership Functions

Membership functions for Mamdani-type fuzzy controller are consistent with [1] (Figures: 4, 5, 6.).

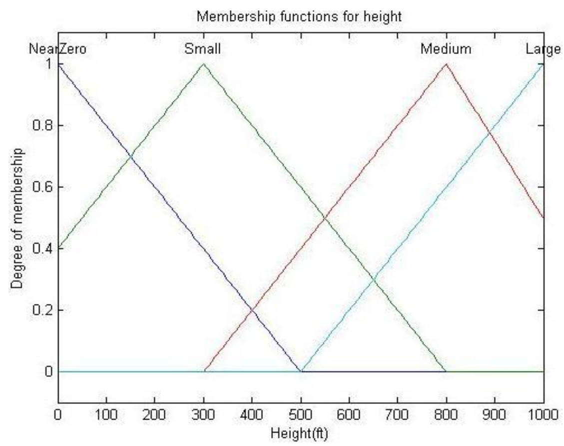

Figure 4. Height membership functions

Membership functions for height are all 'trimf' (triangle membership function) type, and are symmetrical in shape like the membership functions for vertical velocity and Mamdani output membership functions.

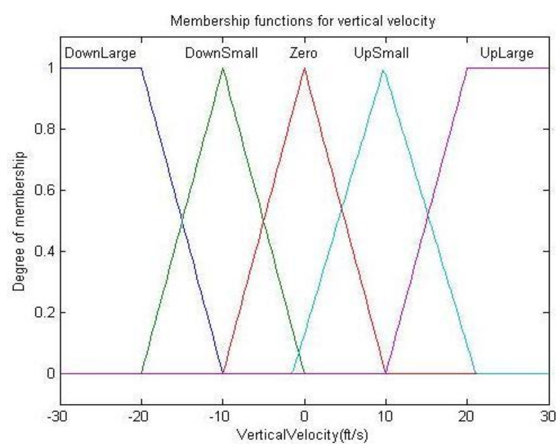

Figure 5. Vertical velocity membership functions

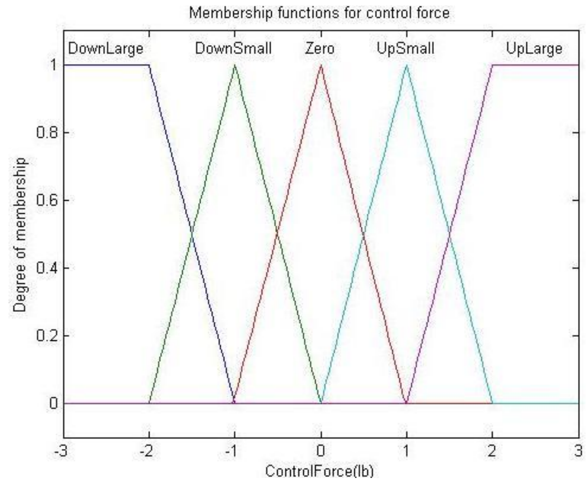

Figure 6. Control force membership functions

Control force and vertical velocity use ‘trimf’ and ‘tramf’ (trapezoid membership functions).Membership functions for Mamdani and Sugeno-type inference systems are the same for input variables. Sugeno-type controller uses different membership functions for the output control force. All Sugeno output membership functions are linear type. The parameters for these membership functions are manually picked.

4.2 Fuzzy Rules

Fuzzy rules are usually represented using the compact graphical form Fuzzy Associative Memory table (FAM table). FAM table for this problem has two dimensions since there are two inputs (height and vertical velocity). Our FAM table is the same as in [1] since we used the same set of rules during evaluation of controllers (Table 1.).

	Velocity				
Height	**DL**	**DS**	**Zero**	**US**	**UL**
L	Z	DS	DL	DL	DL
M	US	Z	DS	DL	DL
S	UL	US	Z	DS	DL
NZ	UL	UL	Z	DS	DS

Table 1. FAM table [1]

In the FAM table each column represents membership function for vertical velocity. DL is “Down Large”, DS is “Down Small”, US is “Up Small” and UL is “Up Large”. Each row is marked by membership functions of the height input. L is “Large”, M is “Middle”, S is “Small” and NZ is “Near Zero”. Values in the table are corresponding control force outputs for each possible (in this model) combination of vertical velocity and height. Z is “Zero”, DS is “Down Small”, DL is “Down Large” and UL is “Up Large”. FAM table shows that there are 20 rules, which is correct since there are four membership functions for height and five membership functions for vertical velocity. Same rules are used both by Mamdani and Sugeno-type controller. ANFIS controller does not use this rule base; it constructs its own rule base.

4.3 Control Surface

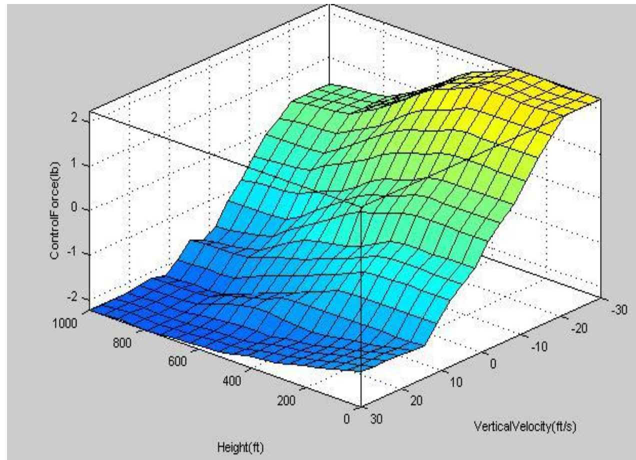

Figure 7. Control surface for Mamdani-type fuzzy inference system

Control surface graphically represents all possible inputs and outputs, which is in this case three-dimensional, since we have two inputs and one output. Control surface for this problem shows graph of control force in function of vertical velocity and height, for all values in range of both inputs (Figure 7.). We can see possible landing situations from this surface. For example, the graph shows that the control force is large for negative large vertical velocity and small height.

5. ANFIS CONTROLLER DESIGN

ANFIS controller can use manually made Sugeno-type fuzzy controller, and improve it using the neural network. This particular ANFIS controller does not use the Sugeno-type fuzzy controller that we developed earlier. Since ANFIS controller has some limitations for the starting Sugeno fuzzy model. According to [9] these limitations are:

- The Sugeno-type system has to be first or zero order
- Each rule should be weighted
- Rules cannot be shared, which means that each rule has to have its own output membership function
- Should have a single output, use weighted average defuzzification process and all output membership functions can be either all linear or all constant

For construction of the starting Sugeno-type fuzzy controller we used grid-partitioning of the input dataset. Using grid-partitioning it is possible to specify generate rules and use a small number of linguistic variables. However if the input dataset is very regular, grid-partitioning may not be able to produce good ruleset for the corresponding input data. This way we specify the number of membership functions for each input, output and input membership function type and the starting Sugeno-type fuzzy controller is generated, and satisfies all constraints for the start inference system. In this case we used 4 and 5 membership functions for the inputs. This number of membership functions enabled us better comparison between ANFIS and ordinary fuzzy inference systems, since we use the exact same number for input variables in construction of ordinary fuzzy inference systems. The type of membership functions for the input is ‘trapmf’, which is not the same as in ordinary fuzzy inference systems, but ‘trapmf’ is used for some membership functions there. For ANFIS ‘trapmf’ should be better than ‘trimf’, since it has four parameters and ANFIS should be able to adjust it better. All output membership functions are set to linear, the same as for ordinary Sugeno-type fuzzy inference system.

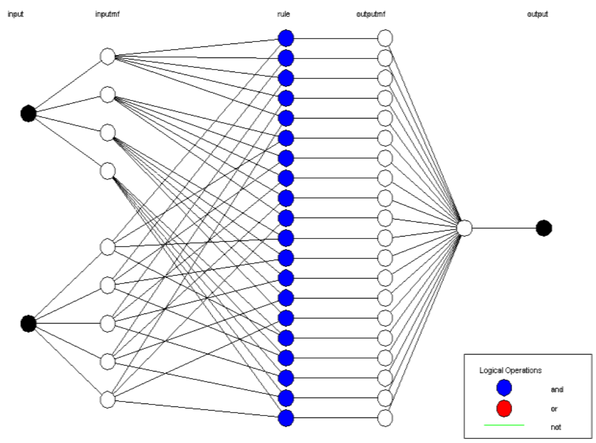

Figure 8. Generated ANFIS structure

After we generated the starting fuzzy inference system (Figure 8.), this system is trained and tested using the neural network, which will be described in the next section.

6. ANFIS CONTROLLER TRAINING AND TESTING

During the training ANFIS controller adjusts its weights, so that the output error is minimized. Output error is measured by comparing ANFIS output for the current input data, with the correct output for that particular input data. Training is conducted in epochs. One epoch represents one pass through all data in the input dataset. Error backpropagation, and hybrid algorithm which uses least square method with error backpropagation, are supported. In our case hybrid algorithm is used.

Since there were some simplifications in the model, first we had to generate the correct and representative dataset for this model. To generate correct output data we need accurately and completely defined mathematical model. Mathematical model of the problem in [1] is not completely defined.

Since “The desired downward velocity is proportional to the square of height” [1], we say that vertical velocity equals square of height multiplied by some coefficient of proportionality:

$$v = k \cdot h^2 \ (3)$$

Out of this equation we can easily get coefficient of proportionality k as:

$$k = \frac{v}{h^2} \ (4)$$

Since coefficient of proportionality would change as the vertical speed and height change, we had to fix it to starting values of vertical speed and height. Based on this we constructed two different datasets. One dataset is constructed for negative vertical velocity and dataset for positive vertical velocity. These two datasets were later merged into one dataset. Datasets for training and testing are then generated from the whole dataset. For training we take every second record as we do for testing, but with different start records.

6.1 Dataset for Negative Vertical Velocity

The negative vertical velocity means that the aircraft is decreasing its height (altitude). Vertical velocity for this dataset changes according to equation (3). Coefficient of proportionality is calculated with these starting parameters:

$$v = -30\,(\frac{ft}{s})$$
$$h = 1000\,(ft)$$

Height changes according to equation (2) (Figure 9.).

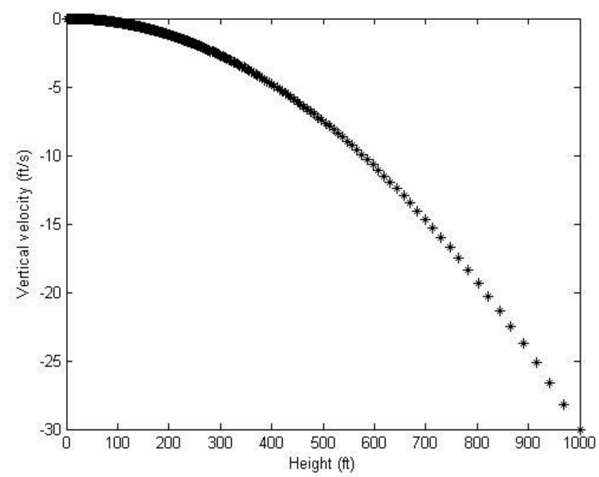

Figure 9. Dataset for negative vertical speed for ANFIS

The correct control force output is calculated according to (1):

$$f_i = v_{i+1} - v_i$$

The control force is calculated as height changes from 1000 (ft) to 1 (ft). Since the vertical velocity is large negative at first, height will decrease faster at first, and control force will be large in those areas. As the aircraft descends, vertical velocity will approach to the zero because of control force applied to the aircraft, which enables ANFIS to learn “the smooth” landing if the start speed is negative large and altitude is high. Dataset for negative vertical speed also enables ANFIS to learn the desired behavior in the area where vertical velocity is negative low and altitude is low. There are few possibilities for ANFIS to learn about desired behavior if the vertical velocity is negative large and altitude is high, or if vertical velocity is negative low and altitude is high.

6.2 Dataset for Positive Vertical Velocity

Positive vertical velocity means that the aircraft is increasing its altitude. In our model we don't want the aircraft to increase altitude, since our goal is aircraft landing. What we want is to decrease the vertical speed until it becomes negative, then based on dataset for negative vertical velocity ANFIS should land the plane in a proper manner. Calculation of control force is the same as for negative vertical velocity dataset, but vertical velocity and height have to be calculated differently. We want the ANFIS to learn that at high altitudes and with large positive vertical velocity, the control force should be negative large. When the vertical velocity is small positive and altitude is low, the control force should be negative small, so that the aircraft does not suddenly change the vertical velocity to negative at low altitudes and thus crush the aircraft. Height change data is taken from the dataset for negative vertical velocity, to enable ANFIS to learn this (Figure 10.).

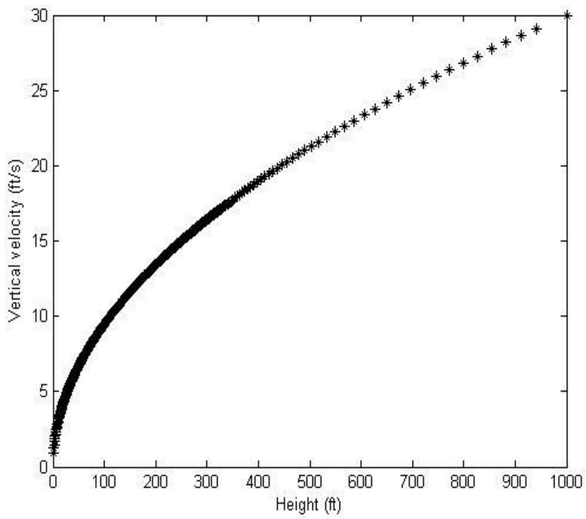

Figure 10. Dataset for positive vertical speed for ANFIS

Vertical velocity changes by formula (6), so that we can get the appropriate path which enables us to calculate control force that we need for suitable decrease of vertical velocity:

$$v = \sqrt{\frac{h}{k}} \quad (6)$$

Coefficient of proportionality is also different and is shown at (7):

$$k = \sqrt{\frac{h}{v^2}} \quad (7)$$

After generating both datasets were merged into one and then training dataset was generated (Figure 11.)

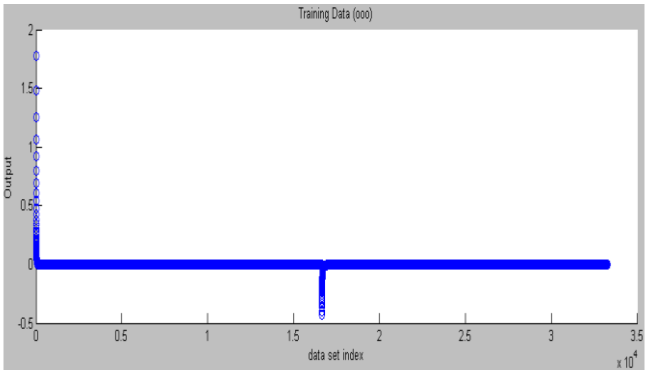

Figure 11. Complete ANFIS training dataset

The training error at the last 100th epoch was: 0.00010213 (Figure 12.).

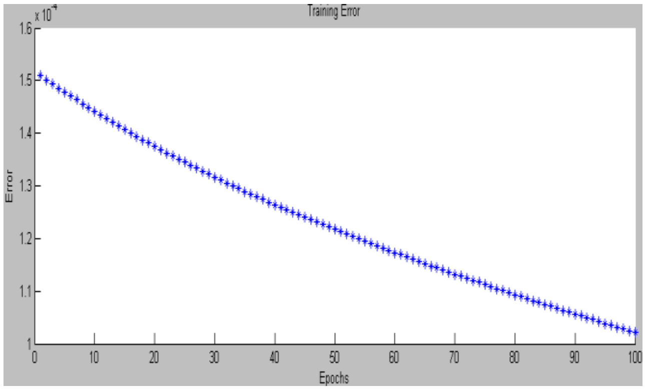

Figure 12. Training error for 100 epochs

6.3 Testing of the ANFIS Controller

The ANFIS controller produced is tested using the test dataset which has also been produced using the merged, complete dataset (Figure 13.).

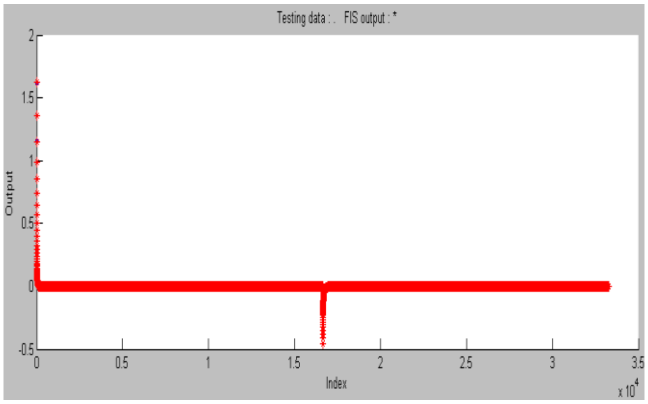

Figure 13. Testing of the ANFIS controller

7. RESULTS

Using Matlab, we simulated the aircraft landing using Mamdani, Sugeno and ANFIS controllers. The results were then compared. For Mamdani-type fuzzy controller we compared: centroid, bisector and MOM defuzzification methods, using "maximum" aggregation method (Figure 14.).

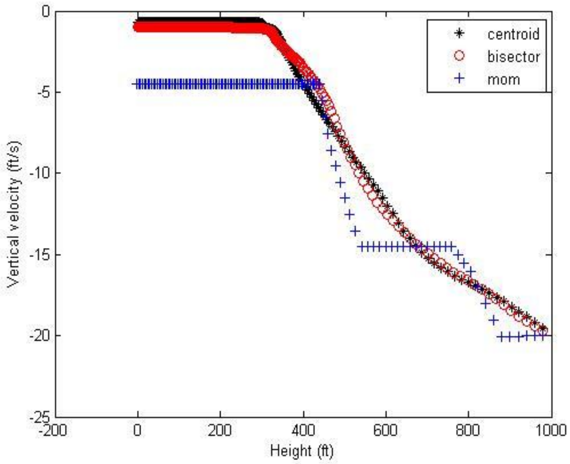

Figure 14. Different defuzzification methods for Mamdani-type fuzzy controller

Using centroid defuzzification method, we then compared three different aggregation strategies for Mamdani. These methods are: "maximum", "sum", "probabilistic or" (Figure 15.). The start points for both of these tests are:

$$h = 1000\ (ft)$$

$$v = -20\ (\frac{ft}{s})$$

The same start point values are used in example in [1].
After evaluating different defuzzification methods, we evaluated different aggregation methods for Mamdani-type fuzzy controller (Figure 15.). For Sugeno-type fuzzy controller, we just evaluated different defuzzification methods: "wtaver" or weighted average and "wtsum" or weighted sum (Figure 16.). We then compared the Mamdani and Sugeno controller with the mathematical model for the example stated in [1]. For Mamdani we used centroid defuzzification and "maximum" aggregation, Sugeno controller used "wtsum" defuzzification method (Figure 17.). ANFIS controller has also been subject to testing. We compared ANFIS controller's performance to mathematical model (Figure 18.). Many test runs are not on the figures for practical reasons, but during the tests, we discovered one undesired property of our ANFIS controller. If the vertical velocity is too high and ANFIS does not reduce it until 1000 (ft), it will start to increase vertical velocity (Figure 19.).

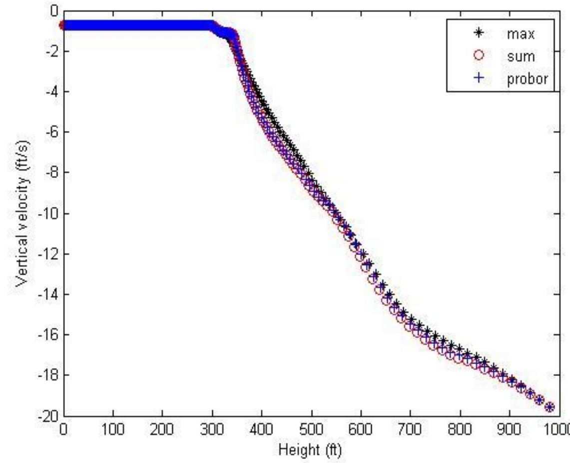

Figure 15. Different aggregation methods for Mamdani-type fuzzy controller

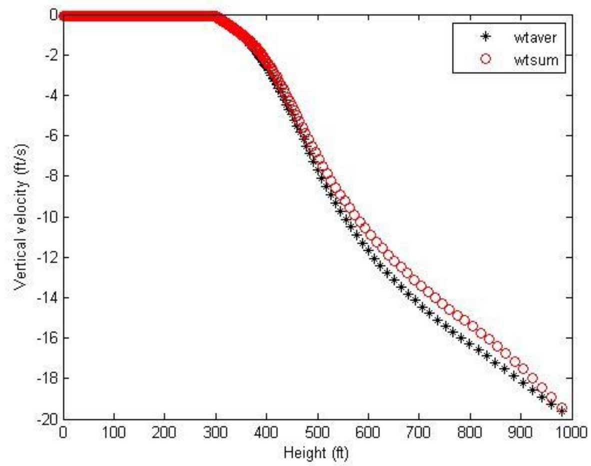

Figure 16. Different defuzzification methods for Sugeno-type fuzzy controller

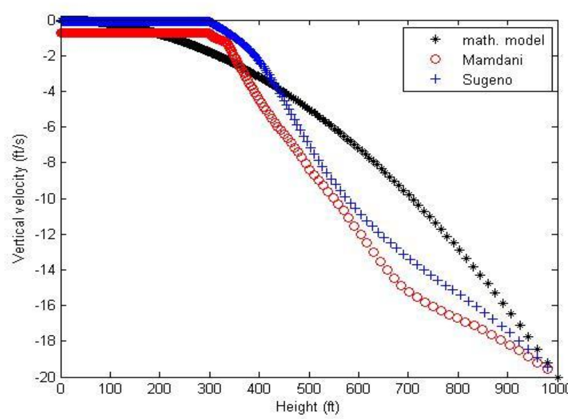

Figure 17. Mamdani and Sugeno controllers compared to the mathematical model

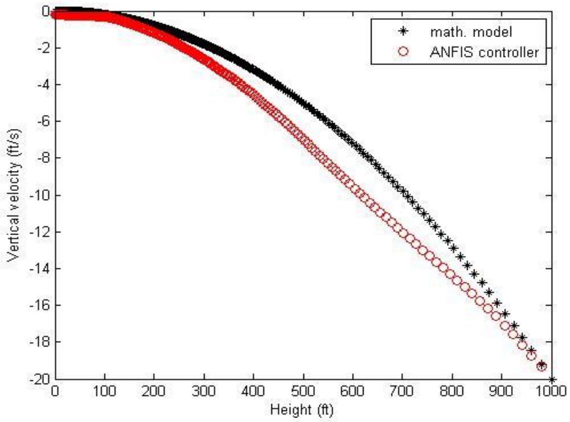

Figure 18. ANFIS controller compared to the mathematical model

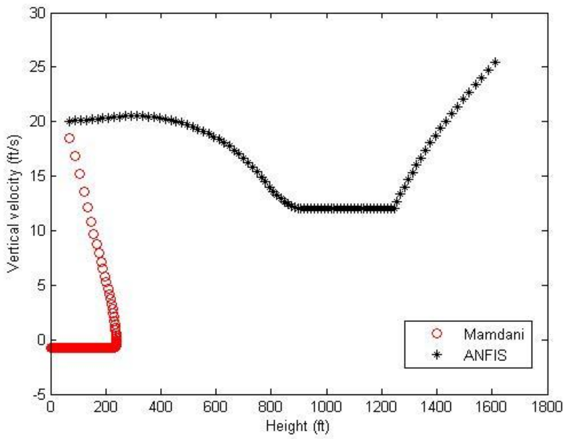

Figure 19. ANFIS controller unable to recover if it reaches 1000 feet and has positive vertical velocity

8. CONCLUSIONS

In this paper we compared two fuzzy and one neuro-fuzzy controller as a solution to an aircraft landing control problem. The paper addresses the vertical velocity during the descent. Design used for the fuzzy and ANFIS controller, has been thoroughly described in Sections 4 and 5. Some problems emerged during the design and training of ANFIS controller. Since we did not use manually made Sugeno-type fuzzy inference system as starting point in ANFIS training, we had to generate one using grid partitioning. This means that the inference system can only know what is presented in the dataset. Following the data trends in the mathematical model that we made for this system, ANFIS could not learn all rules that we had in manually made Mamdani and Sugeno inference systems. However, with the merged dataset we trained ANFIS through 100 epochs, which resulted with very small training and later on very small testing error.

The results of simulation for different types of defuzzification for Mamdani-type fuzzy controller (Figure 14.) show us that MOM method is significantly worse option in this case than "centroid" or "bisector" that are very close. To decide which one of these methods performs better in this case we would need to employ statistic methods, since it is not obvious which one is actually better. We would also need to use statistics if we want to know which aggregation method is better and to decide which defuzzification method for Sugeno-type inference system is better, since these are also close (Figures 15, 16). Compared to the mathematical model, both Mamdani and Sugeno controllers show significant discrepancy. Of all controllers, ANFIS controller follows the mathematical model most accurately (Figure 18.), although it has certain problems. ANFIS controller starts to increase vertical velocity if the aircraft has positive vertical velocity at 1000 (ft). This leads us to conclusion that when neuro-fuzzy inference systems are modeled using data only, we can have erroneous systems as product. These errors are reflection of data imprecision and lack of data.

In future work we will try to employ statistics in comparison of controller outputs, which should enable us to know which defuzzification and which aggregation methods are actually better as a solution to this particular problem. We will also manually prepare the start Sugeno-type fuzzy inference system, for ANFIS training and check the results of that inference system.

REFERENCES

[1] Ross, T.J. *Fuzzy logic with engineering applications*. John Wiley & Sons, Ltd, West Sussex, 2004.

[2] Dourado, A. *Biological systems theory: an introduction*. Retrieved: October 16, 2010, from University of Coimbra: http://www.fis.uc.pt/data/20052006/apontamentos/apnt_1354_5.pdf

[3] MathWorks: *Fuzzy logic toolbox documentation*, Retrieved October 16,2010, from MathWorks: http://www.mathworks.com/help/toolbox/fuzzy/fp351dup8.html#bq3s7wb

[4] Malaek ,S. M. B., Sadati N, Izadi H, Pakmehr M, *Intelligent autolanding controller design using neural networks and fuzzy logic,* in Control Conference, 2004. 5th Asian

[5] JUANG J.G., CHIO J.Z. *Fuzzy modeling control for aircraft automatic landing system*, International Journal of Systems Science, Vol. 36, No. 2, 10 February 2005, Pages 77–87

[6] Livchitz M., Abershitz A., Soudak U, Kandel A. *Development of an automated fuzzy-logic-based expert systemfor unmanned landing,* Fuzzy Sets and Systems, Volume 93, Issue 2, 16 January 1998, Pages 145-159

[7] Larkin L.I., *A fuzzy logic controller for aircraft flight control*, in Decision and Control, 1984. The 23rd IEEE Conference on. Pages 894 - 897

[8] Jang R., Sun C., and Mijutani E., *Neuro-fuzzy and soft computing,* Pearson Education, 2004

[9] MathWorks: *Fuzzy logic toolbox documentation*, Retrieved October 16,2010, from MathWorks: http://www.mathworks.com/help/toolbox/fuzzy/fp715dup12.html

[10] MathWorks: *Fuzzy logic toolbox documentation*, Retrieved October 16,2010, from MathWorks: http://www.mathworks.com/help/toolbox/fuzzy/fuzzy.html