

Fuzzy Logic Plane Landing

Group: Eoghan Quinn, Fergus Hunt, Michael Power

Introduction:

We decided to produce a fuzzy implementation of an autopilot system for the specific case of landing a plane. Fuzzy input & output seemed highly relevant to this kind of system because as a plane descends and slows down toward a landing zone, all critical data about the process would be changing almost constantly. Autopilot systems are already responsible for a lot of the flying on modern commercial aircrafts because they are much less error-prone than human pilots. Fuzzifying these systems would simply serve to make the plane ride steadier & more enjoyable for the people on board, which is obviously a priority for commercial airlines (other than Ryanair).

In forming our fuzzy control system, we had to be careful to understand the main influences & goals of successfully landing a plane. We believe the system we've produced would make for consistently comfortable landings, as well as appropriate handling of dangerous edge cases. We think fuzzy logic was very appropriate for this system and are generally happy with how the rules we've produced handle actual data.

Background:

Domain Specific

Autonomous landing of an aircraft is generally a part of the flight that needs to be strongly assisted by human pilot. Over half of all aviation accidents happen at the final approach and landing stages of the flight sequence.^[1]

One such accident was the Tatarstan Airlines Flight 363 which caused the death of all passengers aboard (50 people). The probable cause of the accident was primarily Human Error "with the pilot lack of flying skills in complex spatial positions (Upset Recovery) led to the creation of a large negative overload, loss of spatial orientation and transfer of the aircraft into a steep dive (pitch down to 75°) until the impact with the ground." with the first error that was made was the switching off of the autopilot which caused and exposed the pilot's lack of skill.^[2]

Takagi-Sugeno-Kang vs. Mamdani

For our system we had to choose between the Mamdani and Takagi-Sugeno fuzzy inference methods for our system.

Mamdani Controllers: Rules contain membership functions for both antecedents and consequent.

Takagi-Sugeno Controllers: Rules contain membership functions for antecedents and linear functions in the consequent this makes Takagi more computationally efficient than Mamdani.

The most fundamental difference between Mamdani and Sugeno is the way the crisp output is generated from the fuzzy inputs. While Mamdani uses the technique of defuzzification of a fuzzy output, Sugeno uses a weighted average to compute the crisp output. After some consideration we decided to use the Mamdani due to its intuitive and plausible description of the knowledge which we feel would suit a decision support system such as this. [3]

Why Fuzzy Logic is better suited for this task?

Traditional set logic returns either 1 or 0 as membership of a variable to a set, whereas in fuzzy logic our return value will be a probability of membership between 0 and 1. The reason why fuzzy logic would be best suited to the automatic landing of a plane is because unlike traditional Logic, fuzzy logic will allow us to reason about to what degree does a speed belong to the Fast category in order to implement this type of automation compared to the traditional logic system.

Simulation/model:

Our Inputs & Outputs

For our model, we boiled down the influences of landing a plane to the most basic and substantial factors. Our input variables are height, distance, and speed. We believe other factors such as varying wind speeds could be taken into account through their influence on these inputs. Our output variables are pitch and throttle. Pitch controls the angle of descent and throttle controls the acceleration of the plane. These are the two main factors in changing the location and velocity of the plane. We believe a fuzzy computer system which constantly monitors and adjusts these variables could improve the safety of the landing process.

Variables Specification

Inputs:

Height:

Near Ground (NG)
Small Height (SH)
Medium Height (MH)
Large Height (LH)

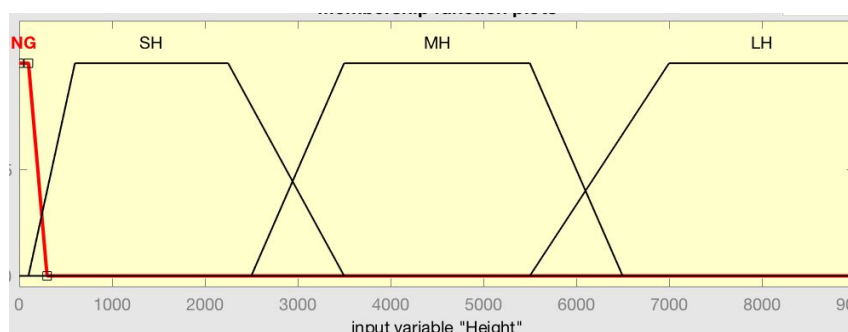
Distance:

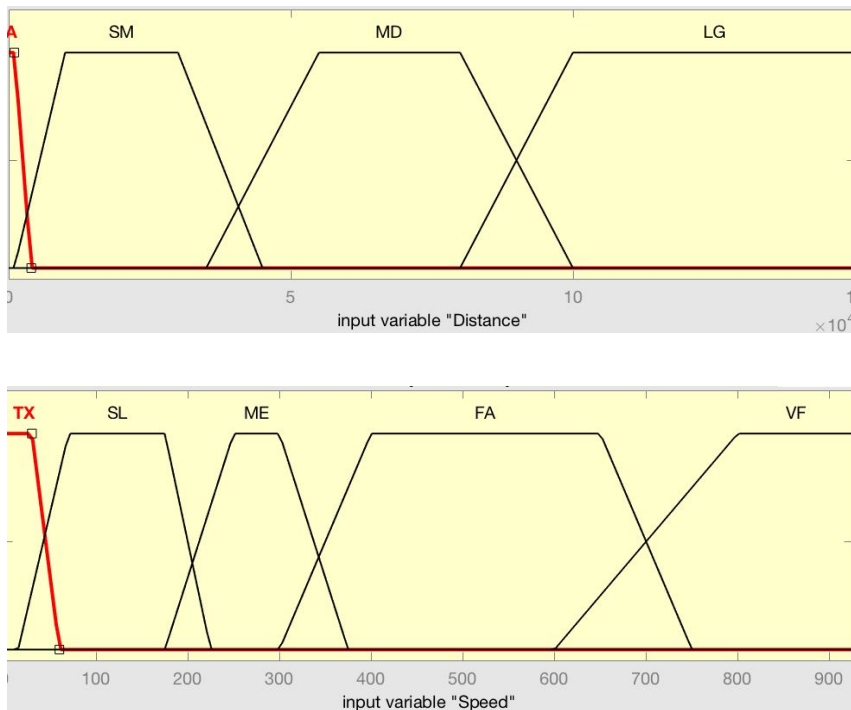
Arrived (A)
Small (SM)
Medium (MD)
Large (LG)

Speed:

Taxing (TX)
Slow (SL)
Medium (ME)
Fast (FA)
Very Fast (VF)

In defining our fuzzy sets, we tried to cover all necessary ranges with as few variables as possible. These inputs can be used to model the plane at all stages of the landing process.





Outputs:

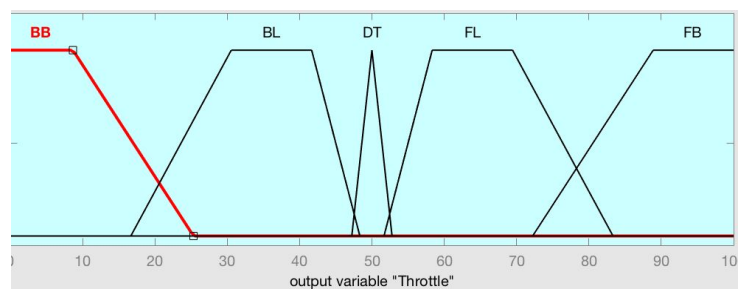
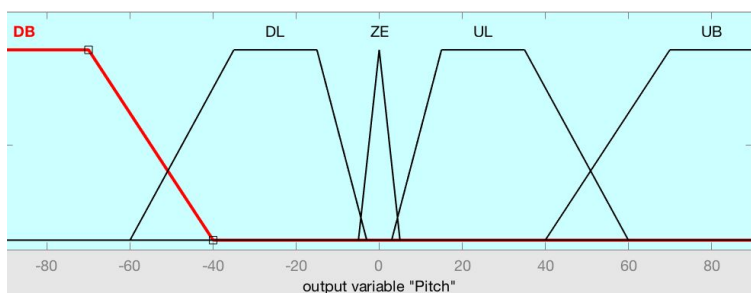
Pitch:

Down Big (DB)
 Down Little (DL)
 Zero (ZE)
 Up Little (UL)
 Up Big (UB)

Throttle:

Back Big (BB)
 Back Little (BL)
 Don't Touch (DT)
 Forward Little (FL)
 Forward Big (FB)

For our outputs, we defined both in terms of how the pilot would tend to perceive their control. This way, they are much easier to interpret which means people would be more able to understand how the system works and thus be more likely to trust it. This also means that it is easier to reason about the system so if any adjustments to the rules are required, they can be easily implemented. Finally the defuzzification process of these outputs is trivial because they can essentially just be translated into percentages of input.



Rules

With our three input variables, we have a total of 80 ($4 \times 4 \times 5$) combinations of inputs values, each mapped onto two output variables for a total of 160 rules. Many of the rules produced by this full process are either obvious or redundant. For example, when a plane is travelling very fast at great height but is already within horizontal landing distance of the runway, both outputs are essentially irrelevant because the plane no longer has any hope of landing anyway and needs to fly on or turn around.

Key rules

The key rules to this system's success are the ones that follow the arc of a perfect landing. Whenever the inputs values are lining up appropriately the output values are only minor adjustments to create a steady comfortable landing. When the inputs values do not line up, the outputs are more aggressive. Here is an example of each type of situation to demonstrate the logic of the system:

1. **Normal:** LH, LG, VF → Pitch: DL Throttle: DT

The plane is far from its destination and beginning the landing procedure as expected. Begin to arc down slowly while maintaining cruising speed for now.

2. **Danger:** NG, MD, FA → Pitch: UB Throttle: BB

The plane is travelling at near-cruising speed while potentially flying just above the ground nowhere near the landing zone. Arc up fast to reach a stable height & slow down to avoid any severe collisions.

3. **Trivial:** LH, A, VF → Pitch: ZE Throttle: DT

The plane has somehow reached the destination horizontally speaking without having even initiated the landing process. Landing is impossible in the short term future so our landing system has nothing to contribute to the situation.

Full List of Rules

A	NG	SH	MH	LH
VF	ZE BB	ZE BB	ZE DT	ZE DT
FA	ZE BB	ZE BB	ZE DT	ZE DT
ME	ZE BB	DL BB	ZE DT	ZE DT
SL	ZE BL	DL BL	ZE DT	ZE DT
TX	ZE DT	ZE DT	ZE DT	ZE DT

SM	NG	SH	MH	LH
VF	UB BB	ZE BB	ZE BB	ZE DT
FA	UL BB	ZE BB	DB BL	ZE DT
ME	UL BL	DL BL	DB DT	ZE FL
SL	UL DT	DL DT	DL DT	ZE FB
TX	ZE DT	ZE FB	ZE FB	ZE FB

MD	NG	SH	MH	LH
VF	UB BB	UB BB	DL BB	ZE DT
FA	UB BL	UB BL	DL BL	DB BB
ME	UB DT	UL DT	DL DT	DB BL
SL	UL FL	ZE FL	ZE FL	DL DT
TX	ZE FB	ZE FB	ZE FB	ZE FB

LG	NG	SH	MH	LH
VF	UB BB	UB BL	UL DT	DL DT
FA	UB BB	UB DT	UL DT	DL FL
ME	UB DT	UB FL	UL FL	DL FB
SL	UL FL	UL FL	UL FB	DB FB
TX	ZE FB	ZE FB	ZE FB	ZE FB

Composition & Defuzzification methods

The method of Composition that we used was Max Composition where “the combined fuzzy subset is constructed by taking the pointwise maximum over all the fuzzy subsets assigned to the output variable by the inference rule.” The method of Defuzzification that we used was the centroid method “relies on using the centre of gravity of the membership function to calculate the crisp value of the output variable.” Both of these process are done automatically within Matlab.

Results & Analysis

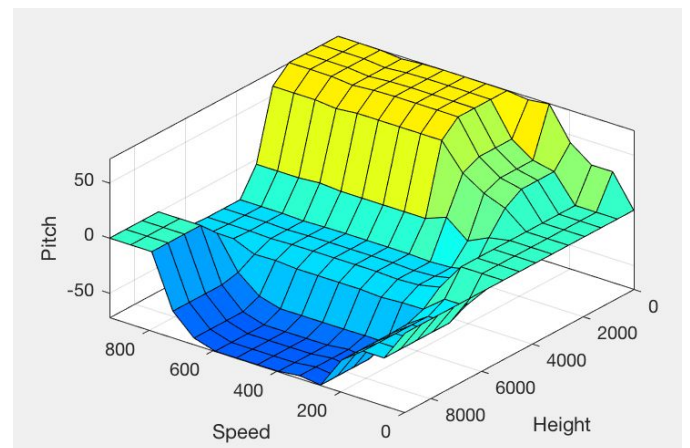
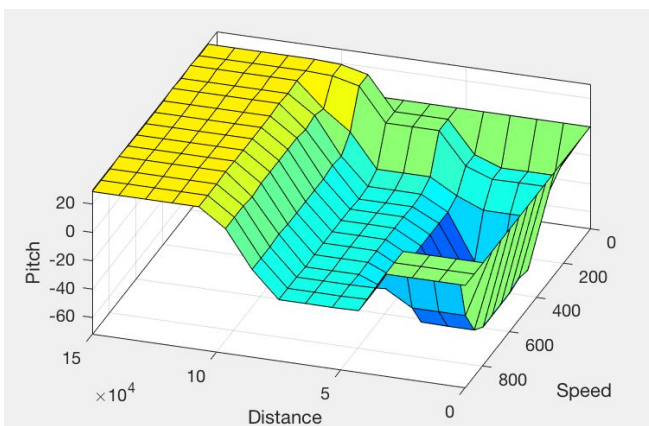
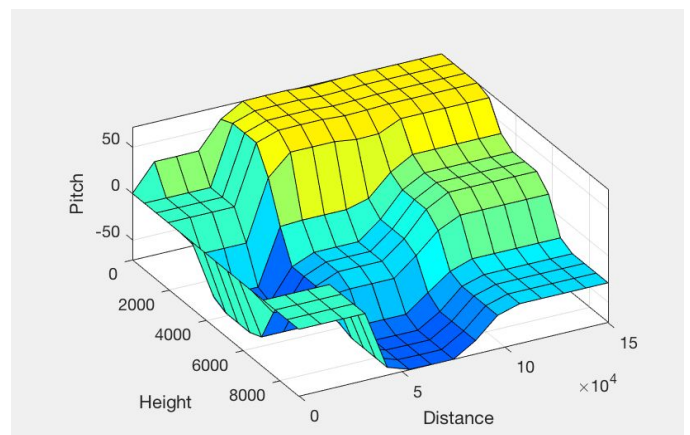
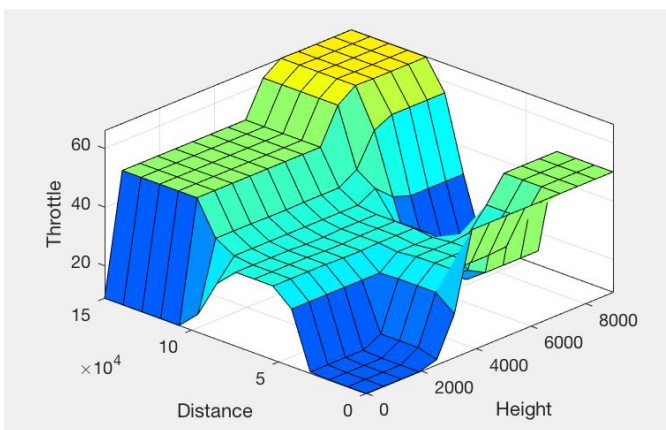
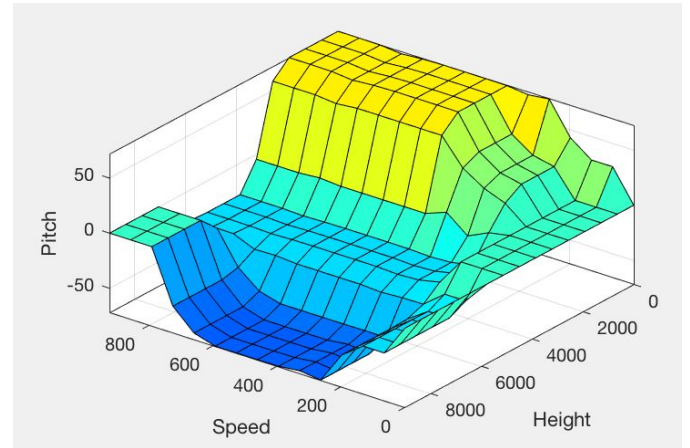
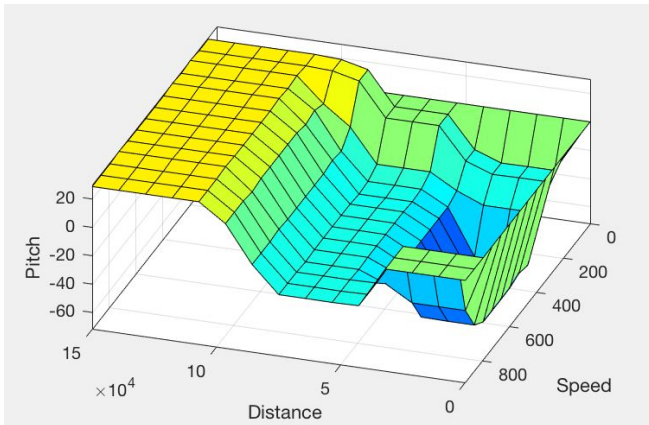
To test our system we created a file with inputs to simulate both normal and abnormal situations, and fed these into the system. The inputs and outputs are shown in the table below.

The first five inputs simulate a normal landing, where the height, distance, and speed decrease proportionally to each other. The outputs for these show that the system will continue the steady descent of the plane.

The rest of the inputs show how the system would correct the plane if it was descending incorrectly. The outputs for these show the plane adjusting to a more correct landing sequence.

Height(m)	Distance(m)	Speed(km/h)	Pitch(°)	Throttle(50 = 0)
8500	130,000	800	-28.77	50
5000	85000	550	-12.02	34.734
1500	30000	225	-29.527	33.597
800	15000	150	-28.77	50
150	1750	75	-1.908	34.04
1000	70000	500	71.901	34.018
1000	10000	200	-29.953	36.074
1000	70000	500	71.901	34.018
1000	10000	700	-3×10^{-15}	10.473
7000	20000	200	-3×10^{-15}	79.03
800	50000	550	70.54	33.708
700	100,000	100	28.77	65.982
8000	100,000	150	-71.901	89.945
4000	8000	350	-68.983	35.068
900	120,000	450	71.901	50
6000	80000	150	-25.072	64.932

Surfaces



Evaluation of Results

These results show that the system works as intended. If the plane is in the correct position for landing, the system will make it land gently, otherwise the system will correct the position of the plane and then continue with the landing.

As seen in the first test input, when the plane is far from the airport and high above the ground, the system will change the pitch down slightly so the plane gets lower and it will keep the speed high as it still needs to go a long distance to get to the airport.

In contrast, when the plane is low off the ground and close to the airport, like in the fifth test input, the pitch will be left mostly alone and the throttle will be pulled back to decrease speed for landing.

If the plane is too low and too fast the system will correct it accordingly. In the eighth test input the plane is 70 kilometers from the airport, but it is only 1000 metres off the ground and going at 500 km/h. The pitch will go up strongly while the throttle is pulled back slightly to correct the position of the plane.

Conclusions

While there are a large amount of variables that go into landing a plane we ultimately feel that our system is justified to use fuzzy logic as a control system for our plane landing, while taking more inputs compared to other studies[4][5] on automatic plane landing using fuzzy logic, there is room to possibly include more inputs such as fuel, and outputs such as the yaw and roll of the plane, but this could overcomplicate the underlying logic of the landing sequence and knowing how many and what inputs are necessary for the system. In order to counteract this an aircraft pilot or someone more familiar within the domain could be incorporated in the design in order to establish the minimum necessary inputs and outputs that would be needed to ensure that all possible outcomes that could affect the landing of the plane are accounted for.

Bibliography

- [1] <https://www.1001crash.com/index-page-statistique-lg-2-numpage-3.html>
- [2] <https://aviation-safety.net/database/record.php?id=20131117-0>
- [3] <https://www.ijser.org/researchpaper/Comparison-of-Mamdani-type-and-Sugeno-type-FIS-for-Water-Flow-Rate-Control-in-a-Rawmill.pdf>

- [4] https://rd.springer.com/content/pdf/10.1007%2F3-540-45631-7_7.pdf
- [5] <http://ieeexplore.ieee.org/document/5354295/?reload=true>
- [6] <http://www.planecrashinfo.com/cause.htm>